	<b>UNIVERSIDAD FRANCISCO DE PAULA SANTANDER OCAÑA</b>			
	<u>Documento</u>	<u>Código</u>	<u>Fecha</u>	<u>Revisión</u>
	<b>FORMATO HOJA DE RESUMEN PARA TRABAJO DE GRADO</b>	<b>F-AC-DBL-007</b>	<b>10-04-2012</b>	<b>A</b>
	<u>Dependencia</u>	<u>Aprobado</u>		<u>Pág.</u>
	<b>DIVISIÓN DE BIBLIOTECA</b>	<b>SUBDIRECTOR ACADEMICO</b>		<b>1(206)</b>

### RESUMEN - TESIS DE GRADO

<b>AUTORES</b>	<b>JESSICA PATRICIA GAMBOA CARRASCAL</b>		
<b>FACULTAD</b>	<b>INGENIERÍAS</b>		
<b>PLAN DE ESTUDIOS</b>	<b>INGENIERÍA DE SISTEMAS</b>		
<b>DIRECTOR</b>	<b>ALVEIRO ALONSO ROSADO GOMEZ</b>		
<b>TÍTULO DE LA TESIS</b>	<b>DISEÑO DE UN METODO AGIL DE DESARROLLO DE SOFTWARE BASADO EN XP, SCRUM, OPENUP Y VALIDADO CON LA HERRAMIENTA DE ANALISIS 4-DAT PARA MEJORAR LA CALIDAD DE LOS PROYECTOS DESARROLLADOS POR LOS GRUPOS DE GESTION DE SOFTWARE DE LA UFPSO.</b>		
<b><u>RESUMEN</u></b> <b><u>(70 palabras aproximadamente)</u></b>			
<p>ESTA INVESTIGACIÓN PROPUSO EL DISEÑO DE UNA ALTERNATIVA ÁGIL PARA EL DESARROLLO DE SOFTWARE QUE INTEGRARA LAS MEJORES PRÁCTICAS DE XP, SCRUM Y OPENUP. SE COMENZÓ DEFINIENDO Y CONOCIENDO LOS ROLES, CICLO DE VIDA Y ARTEFACTOS DE LOS TRES MÉTODOS, DESDE ESOS TRES CRITERIOS SE HIZO UN MAPEO ENTRE SUS CARACTERÍSTICAS, PARA CREAR A PARTIR DE ELLAS LA NUEVA PROPUESTA. PARA DETERMINAR EL GRADO DE CUMPLIMIENTO DEL MANIFIESTO ÁGIL, EL MÉTODO FUE EVALUADO CON LA HERRAMIENTA DE ANÁLISIS Y COMPARACIÓN 4-DAT</p>			
<b>CARACTERÍSTICAS</b>			
<b>PÁGINAS: 206</b>	<b>PLANOS:</b>	<b>ILUSTRACIONES: 16</b>	<b>CD-ROM: 1</b>



**DISEÑO DE UN METODO AGIL DE DESARROLLO DE SOFTWARE BASADO  
EN XP, SCRUM, OPENUP Y VALIDADO CON LA HERRAMIENTA DE  
ANALISIS 4-DAT PARA MEJORAR LA CALIDAD DE LOS PROYECTOS  
DESARROLLADOS POR LOS GRUPOS DE GESTION DE SOFTWARE DE LA  
UFPSO.**

**JESSICA PATRICIA GAMBOA CARRASCAL**

**UNIVERSIDAD FRANCISCO DE PAULA SANTANDER  
FACULTAD DE INGENIERÍAS  
INGENIERÍA DE SISTEMAS  
OCAÑA  
2015**

**DISEÑO DE UN METODO AGIL DE DESARROLLO DE SOFTWARE BASADO  
EN XP, SCRUM, OPENUP Y VALIDADO CON LA HERRAMIENTA DE  
ANALISIS 4-DAT PARA MEJORAR LA CALIDAD DE LOS PROYECTOS  
DESARROLLADOS POR LOS GRUPOS DE GESTION DE SOFTWARE DE LA  
UFPSO.**

**JESSICA PATRICIA GAMBOA CARRASCAL**

**Trabajo de grado presentado como requisito para obtener el título de Ingeniero de  
Sistemas**

**Director**

**ALVEIRO ALONSO ROSADO GOMEZ**  
**Magister en Gestión aplicación y desarrollo de software**

**UNIVERSIDAD FRANCISCO DE PAULA SANTANDER  
FACULTAD DE INGENIERÍAS  
INGENIERÍA DE SISTEMAS  
OCAÑA  
2015**

## **DEDICATORIA**

*Con la satisfacción del deber cumplido, dedico este trabajo de grado especialmente a Dios y a mis padres Marlene y Alberto, quienes fueron pilar fundamental durante toda mi carrera y quienes me impulsaron y ayudaron a cumplir esta meta.*

*Así mismo, a mis hermanos Jhonni y Jhojan, quienes incondicionalmente me han respaldado en cada paso que doy.*

## **AGRADECIMIENTOS**

Doy agradecimiento primero a Dios por darme la sabiduría y la gracia para afrontar los retos y dificultades que se presentaron durante el desarrollo de este trabajo, por ayudarme a lograr este gran sueño de terminar mi proyecto de grado con satisfacción.

Sin duda alguna a mis padres, porque sin ellos esto no sería posible, ya que gracias a ellos obtuve el apoyo moral, psicológico, económico y espiritual. Al igual que a toda mi familia que han sido un apoyo constante.

A mi Alma Mater la Universidad Francisco De Paula Santander Ocaña, por permitir nutrirnos de conocimiento como seres humanos y como profesionales durante todos estos años.

Al Magister Albeiro Rosado Gómez, quien me dio la oportunidad de pertenecer a este proyecto de investigación y que dentro de las materias que me dictó me fue guiando y corrigiendo en la producción de mi tesis, Por su esfuerzo, motivación, experiencia y ayuda dentro del proceso de desarrollo del trabajo.

A mis compañeros que me han brindado su ayuda y han compartido sus conocimientos y que de una u otra manera han contribuido al desarrollo de este logro.

Por último a todos los profesores que durante toda la carrera, pusieron su granito de arena en mi formación, por sus enseñanzas, correcciones, sus consejos, por guiarme a convertirme en persona de bien e incentivarme a mejorar el futuro del país.

## CONTENIDO

	Pág.
<u>INTRODUCCION</u>	16
<u>1. DISEÑO DE UN METODO AGIL DE DESARROLLO DE SOFTWARE BASADO EN XP, SCRUM, OPENUP Y VALIDADO CON LA HERRAMIENTA DE ANALISIS 4-DAT PARA MEJORAR LA CALIDAD DE LOS PROYECTOS DESARROLLADOS POR LOS GRUPOS DE GESTION DE SOFTWARE DE LA UFPSO.</u>	17
1.1 <u>PLANTEAMIENTO DEL PROBLEMA</u>	17
1.2 <u>FORMULACION DEL PROBLEMA</u>	19
1.3 <u>OBJETIVOS</u>	19
1.3.1 Objetivo general.	19
1.3.2 Objetivos específicos	19
1.4 <u>JUSTIFICACION</u>	19
1.5 <u>DELIMITACIONES</u>	22
1.5.1 Geográficas	22
1.5.2 Temporales	22
1.5.3 Conceptuales	22
1.5.4 Operativas	22
2. <u>MARCO REFERENCIAL</u>	23
2.1 <u>MARCO HISTORICO</u>	23
2.1.1 Origen de la ingeniería del software	23
2.1.2 Origen de las metodologías tradicionales	23
2.1.3 Origen de las metodologías ágiles	23
2.1.4 Estado del arte	23
2.2 <u>MARCO CONCEPTUAL</u>	25
2.2.1 Software	25
2.2.2 Ingeniería del software	25
2.2.3 Dimensional Analytical Tool (4-DAT)	26
2.2.4 Métodos ágiles	26
2.2.5 Desarrollo de software	26
2.2.6 Metodologías tradicionales	26
2.2.7 Las metodologías tradicionales (formales)	26
2.2.8 Programación extrema	26
2.2.9 SCRUM	27
2.2.10 OPENUP	27
2.3 <u>MARCO TEORICO</u>	28
2.4 <u>MARCO LEGAL</u>	28
3. <u>DISEÑO METODOLOGICO</u>	36

3.1	<a href="#"><u>TIPO DE INVESTIGACIÓN</u></a>	36
3.2	<a href="#"><u>POBLACIÓN</u></a>	36
3.3	<a href="#"><u>MUESTRA</u></a>	36
3.4	<a href="#"><u>RECOLECCIÓN DE LA INFORMACIÓN</u></a>	37
3.5	<a href="#"><u>ANÁLISIS DE LA INFORMACIÓN</u></a>	37
4.	<a href="#"><u>PRESENTACION DE RESULTADO</u></a>	42
4.1	<a href="#"><u>DEFINICIÓN Y CARACTERÍSTICAS DE LOS MÉTODOS AGILES XP SCRUM Y OPENUP</u></a>	55
4.1.1	Extreme Programming	55
4.1.2	Scrum	63
4.1.3	OpenUp	78
4.2	<a href="#"><u>COMPARACIÓN ENTRE FASES, ROLES Y ARTEFACTOS DE XP, SCRUM Y OPENUP</u></a>	90
4.2.1	Ciclo de vida XP, SCRUM Y OPENUP	90
4.2.2	Roles XP, SCRUM Y OPENUP	90
4.2.3	Artefactos XP, SCRUM Y OPENUP	93
4.3	<a href="#"><u>PROPUESTA DE UN NUEVO MÉTODO ÁGIL A PARTIR DE LAS CARACTERÍSTICAS PROPIAS DE XP, SCRUM Y OPENUP, INTEGRANDO LOS PUNTOS EN COMÚN QUE EXISTAN ENTRE ELLAS Y ADICIONANDO CARACTERÍSTICAS QUE LO COMPLEMENTEN PARA DEFINIR LA ESTRUCTURA QUE TENDRÁ EL NUEVO MÉTODO</u></a>	94
4.4	<a href="#"><u>EVALUACIÓN DEL MÉTODO ÁGIL UTILIZANDO LA HERRAMIENTA DE COMPARACIÓN Y ANÁLISIS 4-DAT</u></a>	112
4.4.1	Evaluación de MAF con 4-DAT	115
4.4.2	Alineación de MAF con la UFPSO	121
5.	<a href="#"><u>CONCLUSIONES</u></a>	145
6.	<a href="#"><u>RECOMENDACIONES</u></a>	146
	<a href="#"><u>BIBLIOGRAFIA</u></a>	147
	<a href="#"><u>REFERENCIAS DOCUMENTALES ELECTRONICAS</u></a>	149
	<a href="#"><u>ANEXOS</u></a>	153

## LISTA DE FIGURAS

	Pág.
Figura 1. Ranking de métodos más usados según versiónOne	20
Figura 2. Modelo en cascada	43
Figura 3. Modelo en espiral	44
Figura 4. Modelo de prototipos	45
Figura 5. Ciclo de vida RUP	46
Figura 6. Desarrollo y prototipado incremental	51
Figura 7. Ciclos en XP	62
Figura 8. Vista general de SCRUM	65
Figura 9. Ejemplo de Burndown chart	72
Figura 10. Ciclo de vida SCRUM	73
Figura 11. Vista general método ágil18	78
Figura 12. Ciclo de vida OPENUP	88
Figura 13. Roles MAF	97
Figura 14. Artefactos MAF	109
Figura 15. Fases de MAF	111
Figura 16. Herramientas y técnicas de desarrollo de software	126



## LISTA DE TABLAS

	Pág.
Tabla 1. Rankig de agilidad	21
Tabla 2. Muestra	37
Tabla 3. Evaluación y comparación con la segunda dimensión	118

## LISTA DE CUADROS

	Pág.
Cuadro 1. ¿Cómo calificaría usted su nivel de conocimiento en métodos ágiles?	37
Cuadro 2. ¿Qué métodos ágiles ha empleado?	38
Cuadro 3. ¿En qué medida su método adoptado ha mejorado el rendimiento de sus equipos de trabajo?	39
Cuadro 4. ¿En qué medida su método adoptado ha mejorado su tiempo de entrega?	39
Cuadro 5. ¿En qué medida su método adoptado ha mejorado sus costos?	39
Cuadro 6. ¿En qué medida su método adoptado ha mejorado la calidad de su producto final?	40
Cuadro 7. ¿Principal problema que tuvo al adoptar los métodos ágiles?	40
Cuadro 8. ¿Usara métodos ágiles en un futuro?	41
Cuadro 9. Artefactos RUP	47
Cuadro 10. Los principios de los métodos ágiles	53
Cuadro 11. Diferencias entre métodos ágiles y tradicionales	54
Cuadro 12. Comparación entre las bases de los métodos ágiles y las de los procesos de desarrollo por planes	55
Cuadro 13. Formato para historias de usuario	56
Cuadro 14. Formato caso prueba de aceptación	59
Cuadro 15. Task card (Tareas de ingeniería)	59
Cuadro 16. Formato tarjeta CRC	60
Cuadro 17. Historias de usuario SCRUM	70
Cuadro 18. Plantilla product backlog	70
Cuadro 19. Ejemplo de un sprint backlog	71
Cuadro 20. Artefactos OPENUP	80
Cuadro 21. Relación entre ciclos de vida	91
Cuadro 22. Relación entre roles	92
Cuadro 23. Cuadro de relación existente entre artefactos	93
Cuadro 24. Plantilla del product backlog	99
Cuadro 25. Formato para historias de usuario	99
Cuadro 26. Formato caso prueba de aceptación	100
Cuadro 27. Task card (Tareas de ingeniería)	101
Cuadro 28. Historias de usuario Vs casos de uso	102
Cuadro 29. Plantilla sprint backlog	104
Cuadro 30. Plantilla libro de arquitectura	105
Cuadro 31. Libro de arquitectura y diseño	105
Cuadro 32. Plantilla prueba de desarrollo	108
Cuadro 33. Mapeo Artefactos-rol	110
Cuadro 34. Mapeo Fase-Artefacto	112
Cuadro 35. Primera dimensión 4-DAT	113
Cuadro 36. Segunda dimensión 4-DAT	114
Cuadro 37. Tercera dimensión 4-DAT	115
Cuadro 38. Cuarta dimensión 4-DAT	116

Cuadro 39. Evaluación y comparación con la primera dimensión	116
Cuadro 40. Evaluación y comparación con la tercera dimensión	119
Cuadro 41. Evaluación y comparación con la cuarta dimensión	120
Cuadro 42. Product Backlog	126
Cuadro 43. Historias de usuario crear login	127
Cuadro 44. Historias de usuario migrar base de datos	127
Cuadro 45. Historias de usuario Consultar usuarios externos en las bases de datos	128
Cuadro 46. Historias de usuario Asignar Cita	128
Cuadro 47. Historias de usuario Modificar cita	128
Cuadro 48. Historias de usuario Cancelar cita	129
Cuadro 49. Historias de usuario Confirmar Asistencia	129
Cuadro 50. Historias de usuario Crear usuarios administradores del sistema	129
Cuadro 51. Historias de usuario Actualizar usuarios internos	130
Cuadro 52. Historias de usuario Consultar usuarios internos	130
Cuadro 53. Historias de usuario Eliminar usuarios Internos	130
Cuadro 54. Historias de usuario Recuperar Contraseña	131
Cuadro 55. Historias de usuario: Generar Reportes	131
Cuadro 56. Historias de usuario: Generar Indicadores de Oportunidad	131
Cuadro 57. Historias de usuario: Consultar Citas	132
Cuadro 58. Historias de usuario: Generar reportes por usuarios externos	132
Cuadro 59. Historias de usuario: Generar reportes por usuarios externos	133
Cuadro 60. Sprint Backlog	133
Cuadro 61. Libro de arquitectura y diseño	135
Cuadro 62. Prueba de desarrollo 001	142
Cuadro 63. Prueba de desarrollo 002	142
Cuadro 64. Prueba de desarrollo 003	142
Cuadro 65. Prueba de desarrollo 004	143
Cuadro 66. Prueba de desarrollo 005	143
Cuadro 67. Prueba de desarrollo 006	144
Cuadro 68. Prueba de desarrollo 007	144
Cuadro 69. Prueba de desarrollo 008	144
Cuadro 70. Prueba de desarrollo 009	145
Cuadro 71. Prueba de desarrollo 0010	145

## LISTA DE ANEXOS

	Pág.
Anexo A. Cronograma de actividades	156
Anexo B. Encuesta 1	158
Anexo C. Encuesta 2	160
Anexo D. Manual de usuario	162

## **RESUMEN**

Esta investigación propuso el diseño de una alternativa ágil para el desarrollo de software que integrara las mejores prácticas de XP, Scrum y OpenUP. Se comenzó definiendo y conociendo los roles, ciclo de vida y artefactos de los tres métodos, desde esos tres criterios se hizo un mapeo entre sus características, para crear a partir de ellas la nueva propuesta. Para determinar el grado de cumplimiento del manifiesto ágil, el método fue evaluado con la herramienta de análisis y comparación 4-DAT, la cual arrojó como resultado que el nuevo método cumple con todos los requisitos para considerarse ágil y por lo tanto estructurar los elementos necesarios para el desarrollo de software

## INTRODUCCION

En la actualidad el desarrollo de software ha ido escalando a grandes pasos en muchas organizaciones. El desarrollo de software es la conceptualización, análisis, diseño y construcción de aplicaciones de software a través del uso de procesos, métricas y herramientas, en un amplio rango de arquitecturas y tecnologías, con el objetivo de contribuir a la eficiencia de los procesos estratégicos, operativos y de soporte de una organización.<sup>1</sup>

Los métodos con que se realizan este tipo de proyectos han ido evolucionando para lograr un mejor desempeño de los equipos y un producto mejorado y con más calidad. La última tendencia son los métodos ágiles, que son métodos de desarrollo de software basados en el desarrollo iterativo e incremental, se divide en iteraciones, genera el trabajo en equipo y promueve las comunicaciones cara a cara en vez de la documentación, permite a los equipos desarrollar software de forma más acertada con las necesidades del cliente y responder mejor a los cambios que puedan surgir a lo largo de un proyecto de desarrollo.<sup>2</sup> Es primordial para el éxito de los proyectos de este tipo, conocer las nuevas tendencias y la variedad de métodos que existen para obtener un mejor trabajo.

En este proyecto de investigación a partir de tres métodos ágiles como lo son XP, SCRUM y OPENUP, se obtuvo un nuevo método que fusiono las mejores características de los tres métodos mencionados para mejorar la calidad de los proyectos desarrollados por los grupos de gestión de software de la UFPSO, el método creado se evaluó por la herramienta de análisis y comparación 4-DAT, para comprobar que cumplía con los requisitos para considerarse ágil y además de esto se hizo un caso práctico aplicando el método diseñado a un proyecto de desarrollo de software hecho en dicha institución.

---

<sup>1</sup> INTELLEGO Desarrollo de software (s.l.) [En línea]. Mar. 2010. [Citado el 25 noviembre de 2014.] Disponible en internet en: <http://www.intellego.com.mx/es/sd-desarrollo-de-software>

<sup>2</sup> Alaimo, Diego Martín. Proyectos ágiles con Scrum : flexibilidad, aprendizaje, innovación y colaboración en contextos complejos . - 1ª ed. - Ciudad Autónoma de Bueos Aires : Kleer, 2013. EBook.

# 1. DISEÑO DE UN METODO AGIL DE DESARROLLO DE SOFTWARE BASADO EN XP, SCRUM, OPENUP Y VALIDADO CON LA HERRAMIENTA DE ANALISIS 4-DAT PARA MEJORAR LA CALIDAD DE LOS PROYECTOS DESARROLLADOS POR LOS GRUPOS DE GESTION DE SOFTWARE DE LA UFPSO.

## 1.1 PLANTEAMIENTO DEL PROBLEMA

A través del tiempo el desarrollo de software ha ido escalando y mejorando, existiendo una gran variedad de procesos para su solución. Una metodología no es más que un conjunto de procedimientos, técnicas, herramientas y documentos que tienen como fin ayudar a los desarrolladores de software en tratamiento de sus proyectos.<sup>3</sup>

Lo anterior quiere decir que una metodología encierra varios aspectos que hacen que se pueda desarrollar software de manera más organizada, rápida y efectiva.

Los métodos ágiles de desarrollo de software son usados como **técnicas** para acelerar los procesos del ciclo de vida del desarrollo a través de herramientas, técnicas y artefactos que enfatizan las comunicaciones cara a cara en vez de la documentación.

La variedad de métodos ágiles produjo que los desarrolladores se vean enfrentados a la difícil decisión de elegir la que mejor conlleve al éxito del proceso con un menor número de errores posibles, sin embargo no existe hoy un método que permita utilizar lo mejor de estos métodos con el fin de optimizar el tiempo entre fases, especificar los artefactos necesarios y mejorar el proceso de desarrollo de software.

En la actualidad se viene teniendo una gran acogida de los métodos ágiles en empresas que desarrollan software, es así que un estudio realizado en el 2013 por versionOne<sup>4</sup> a un grupo de empresas de este sector, encontró que el 88% están implementando métodos ágiles en el desarrollo en sus proyectos, este resultado comparado con el 84% del año anterior muestra como los métodos ágiles tienen una gran aceptación y crecimiento en pequeñas y grandes empresas; además el estudio permite determinar que el 19% de las empresas que empezaron usando este método de desarrollo lo ha seguido aplicando durante los últimos 5 años lo que refleja que son estables, completas y una buena opción para gestionar proyectos.

Pero al ser estos métodos la tendencia en la actualidad, se puede deber a que se están aplicando como excusa para no construir artefactos, esto quiere decir que quienes los implementan se basan en que son métodos que no requieren documentación y que en cada

---

<sup>3</sup> CONCYTEC Revista de la Facultad de Ingeniería Industrial 13(1): 70-74 (2010) UNMS ISSN: 1560-9146 (Impreso) / ISSN: 1810-9993 (Electrónico) (s.l.) [En línea]. Mar. 2010. [Citado el 25 noviembre de 2014.] Disponible en internet en: [http://revistas.concytec.gob.pe/scielo.php?pid=S1810-99932010000200009&script=sci\\_arttext](http://revistas.concytec.gob.pe/scielo.php?pid=S1810-99932010000200009&script=sci_arttext)

<sup>4</sup> VERSIÓN ONE. metodologías ágiles en el desarrollo en sus proyectos (s.l.) [En línea]. Mar. 2010. [Citado el 25 noviembre de 2014.] Disponible en internet en: <http://www.versionone.com/>

iteración se pueden analizar los requerimientos, lo cual puede ser un error porque va a comprometer el proyecto en el futuro, ya que si no está escrito sino en la mente de los desarrolladores va a generar confusión y se perdería la trazabilidad y escalabilidad de la aplicación.

Una de las críticas hechas a los métodos ágiles, emana de una carta de Steven Rakitin a la revista Computer, bajo la rúbrica “El Manifiesto genera cinismo”. Refiriéndose a la estructura opositiva del Manifiesto, Rakitin expresa que en su experiencia, los elementos de la derecha (vinculados a la mentalidad planificadora) son esenciales, mientras que los de la izquierda sólo sirven como excusas para que los hackers escupan código irresponsablemente sin ningún cuidado por la disciplina de ingeniería. Así como hay una lectura literal y una estrecha del canon de CMM (modelo de calidad específico para desarrollo de software), Rakitin practica una “interpretación hacker” de propuestas de valor tales como “responder al cambio en vez de seguir un plan” y encuentra que es un generador de caos. La interpretación hacker de ese mandamiento sería algo así como: “¡Genial! Ahora tenemos una razón para evitar la planificación y codificar como nos dé la gana”.<sup>5</sup>

Bajo este panorama se puede ver que existe una controversia entre lo ágil, visto como flexible y desordenado y por otro lado lo ágil como la alternativa a las estructuras rígidas con una jerarquización desproporcional de artefactos, roles y funciones. Es necesario la creación de nuevos métodos que integren lo ágil con lo tradicional, mejorando la flexibilidad en las fases, roles y artefactos, pero respetando la estructura propuesta por los métodos tradicionales que han dado solución en muchos proyectos exitosos hasta este momento.

Este nuevo modelo planteado puede ser aplicado en proyectos grandes y pequeños, ya que cuenta con todo lo necesario para el desarrollo como lo son artefactos, fases y roles. Le dará a los desarrolladores una nueva alternativa completa y comprobada por la herramienta para el análisis de metodologías 4-DAT. Dicha herramienta facilita la evaluación de métodos ágiles mediante cuatro dimensiones: alcance de la metodología, caracterización de la agilidad, valores ágiles (Manifiesto Ágil) y caracterización del proceso de software. La herramienta está diseñada para comparar y analizar los métodos ágiles.

Este método está dirigido a cualquier empresa que desee utilizarlo, por lo que se realizó una prueba base con los grupos Desarrolladores de software de la Universidad Francisco de Paula Santander; conformados por los integrantes de proyecto GENESIS, los ingenieros pertenecientes a la división de sistemas y los directores de los grupos de investigación vinculados al programa de ingeniería de sistemas.

Se anexa a este documento una encuesta realizada a dichos grupos (Anexo B) que dará una idea más clara acerca de que tan usados son los métodos ágiles en la Universidad.

---

<sup>5</sup> METODOLOGIASAGILES. Carta de Steven Rakitin a la revista Computer (s.l.) [En línea] (s.f) [Citado el 25 noviembre de 2014.] Disponible en internet en: <http://metodologiasagiles.wikispaces.com/Cr%C3%ADtica+de+a+las+Metodolog%C3%ADas+%C3%81gile>



## **1.2 FORMULACION DEL PROBLEMA**

¿Se puede crear una metodología ágil a partir de las fases y artefactos propuestos XP, SCRUM, OPENUP que pueda ser seguida en proyectos de desarrollo de software?

## **1.3 OBJETIVOS**

**1.3.1 Objetivo general.** Diseñar un método ágil de desarrollo de software basada en XP, SCRUM, OPENUP y validada con la herramienta de análisis 4d (4-dat) para mejorar la calidad de los proyectos desarrollados por los grupos de gestión de software de la UFPSO

**1.3.2 Objetivos específicos.** Proponer un nuevo método ágil a partir de las características propias de XP, SCRUM, OPENUP, integrando los puntos en común que existan entre ellas y adicionando características que lo complementen para definir la estructura que tendrá el nuevo método.

Evaluar el método ágil propuesto para comprobar si cumple con los criterios que tienen los métodos ágiles utilizando la herramienta de comparación y análisis 4-DAT.

## **1.4 JUSTIFICACION**

El desarrollo de software es una actividad muy común y que en la actualidad requiere soluciones rápidas y confiables, es por eso que en respuesta a dicha necesidad se ha creado el uso de métodos que proporcionan una serie de pasos para mejorar la calidad y la productividad.

Existen diferentes tipos de métodos para el desarrollo de software, están los métodos tradicionales, los métodos híbridos y los métodos ágiles.

Los métodos ágiles son un tipo de método basado en el desarrollo “iterativo e incremental”<sup>6</sup> y que generan respuesta en un corto lapso de tiempo, son flexibles a cambios y minimizan la documentación.

El uso de métodos ágiles para el desarrollo de proyectos de software es muy importante ya que proporciona una gran cantidad de ventajas a los equipos de desarrollo entre las que se encuentran: disminución de documentación, mejora la relación con el cliente ya que está involucrado totalmente en el proyecto, promueve el trabajo en equipo con sus reuniones diarias, mejora la calidad del producto ya que no pasa a la siguiente fase hasta que no se hayan solucionado todos los errores.

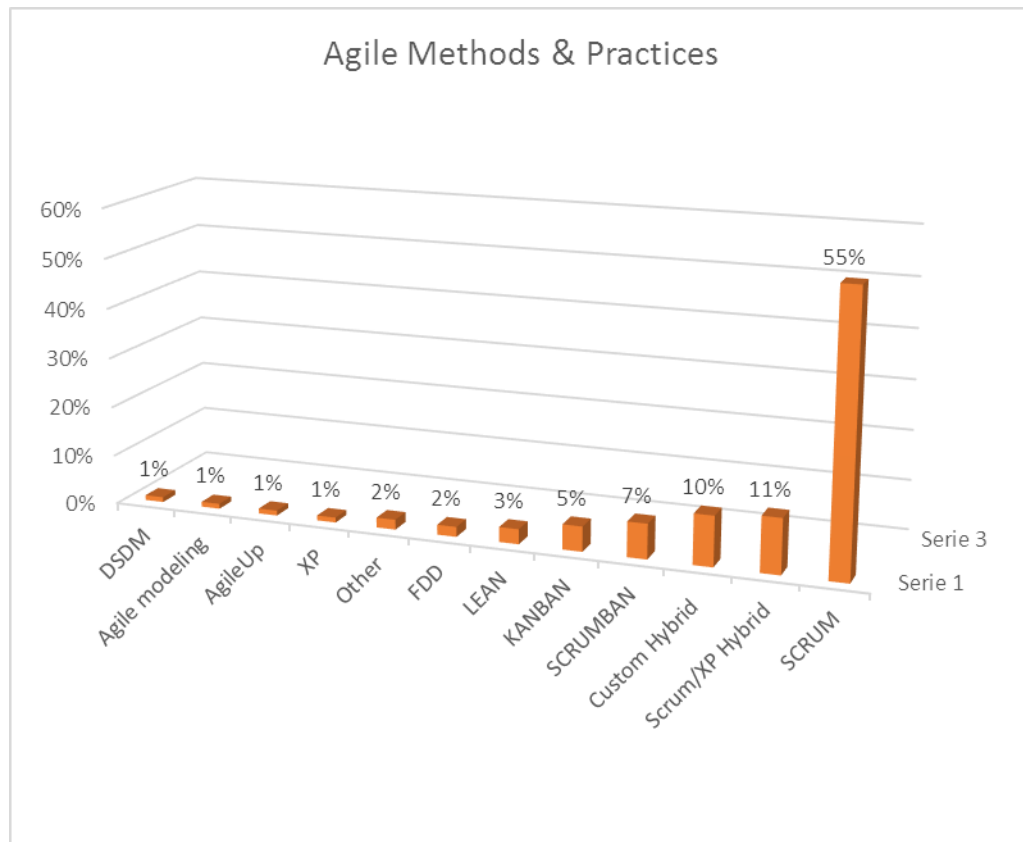
Se decidió trabajar sobre estos métodos ya que tienen mayor acogida, encuestas realizadas por VersionOne aplicada a varios canales de comunidades desarrollo de software arrojo

---

<sup>6</sup> PROYECTOSAGILES. Iterativo e incremental (s.l.) [En línea]. (s.f). [Citado el 25 noviembre de 2014.] Disponible en internet en: <http://www.proyectosagiles.org/desarrollo-iterativo-incremental>

como resultado que SCRUM y XP son las más utilizadas como nos muestra la figura 1 donde se puede observar que Scrum domina el ranking.<sup>7</sup>

**Figura 1.** Ranking de métodos más usados según versiónOne



**Fuente.** Adaptado de VersioOne. Métodos y prácticas ágiles. Disponible en: [www.versionone.com](http://www.versionone.com)

Esto también se puede ver en la tabla 1 obtenida de “Agile Software Development Ecosystems”<sup>8</sup> en la que se compara algunos métodos ágiles en base a: vista del sistema como algo cambiante, tener en cuenta la colaboración entre los miembros del equipo y características más específicas de la propia metodología como son simplicidad, excelencia técnica, resultados, adaptabilidad, etc.

7 VERSIONONE, Scrum y XP (s.l.) [En línea]. Mar. 2010. [Citado el 25 noviembre de 2014.] Disponible en internet en: <http://www.versionone.com/pdf/2013-state-of-agile-survey.pdf>

8 HIGHSMITH, J. "Agile Software Development Ecosystems". Addison-Wesley. 2002.

Tabla 1. Rankig de agilidad

	Cmm	Asd	Crystal	Dsdm	Fdd	Ld	Scrum	Xp
Sistema como algo cambiante	1	5	4	3	3	4	5	5
Colaboración	2	5	5	4	4	4	5	5
Características metodología (CM)								
Resultados	2	5	5	4	4	4	5	5
Simplicidad	1	4	4	3	5	3	5	5
Adaptabilidad	2	5	5	3	3	4	4	3
Excelencia técnica	4	3	3	4	4	4	3	4
Prácticas de colaboración	2	5	5	4	3	3	4	5
<b>Media CM</b>	2.2	4.4	4.4	3.6	3.8	3.6	4.2	4.4
<b>MEDIA TOTAL</b>	1.7	4.8	4.5	3.6	3.6	3.9	4.7	4.8

**Fuente.** Adaptado de: HIGHSMITH, J. "Agile Software Development Ecosystems". Addison-Wesley. 2002.

Como se puede apreciar en la tabla 1 OpenUP no aparece en ninguna de las encuestas y esto se debe a que aunque no es muy conocida se decidió inclinarse por ella ya que posee mayor número de artefactos que un método ágil y menos que un método tradicional, lo cual hace que sume gran importancia ya que uno de los inconvenientes que poseen los métodos ágiles es que no se hace un buen uso de los artefactos, además de esto emplea plantillas para el desarrollo de estos lo cual hace que se mantenga mayor organización en el proyecto. Dicho método fue creado ya que IBM vio la necesidad de crear una versión más ágil de la metodología RUP (RATIONAL UNIFIED PROCESS)<sup>9</sup>, conserva sus características esenciales y tiene un enfoque iterativo e incremental.<sup>10</sup>

Según algunos expertos que hacen parte de IBM (International Business Machines Corp), OpenUP muestra que un proceso ágil es disciplinado, a diferencia de lo que muchos piensan sobre el desarrollo ágil sea sinónimo de "desarrollo desordenado." Adoptar un proceso ágil significa la adopción de las mejores prácticas de gestión y técnicas con el sentido común y mucha disciplina.<sup>11</sup>

Este proyecto planeo desde un punto de vista solucionar los problemas que genera el mal uso de los métodos ágiles, con respecto a la documentación y generación de artefactos. Por otra parte busca plantear un método ágil que permita seguir un marco de referencia para el desarrollo de software, cuya finalidad sea combinar la flexibilidad propuesta por el

<sup>9</sup>METODOLOGÍA RUP (RATIONAL UNIFIED PROCESS) Disponible en línea en: <http://procesosdesoftware.wikispaces.com/METODOLOGIA+RUP>

<sup>10</sup> Página oficial OpenUp disponible en línea en: <http://epf.eclipse.org/wikis/openup/index.htm>

<sup>11</sup> IBM. OpenUP: Un proceso ágil. En línea < [http://www.ibm.com/developerworks/br/rational/local/open\\_up/>](http://www.ibm.com/developerworks/br/rational/local/open_up/>). Citado 27 de julio 2015.

manifiesto ágil con la documentación necesaria y apropiada para planear y soportar el desarrollo. Finalmente este método se evaluó a con la herramienta comparativa Dimensional Analytical Tool (4-DAT), para comprobar que cumplía como método ágil.

Después de hacer la evaluación con esta herramienta se hizo una prueba base aplicando dicho método a un proyecto realizado en la Universidad Francisco de Paula Santander, para confirmar que cumplía con las expectativas las cuales son: contar con documentación del proyecto, tener un desarrollo ágil, flexible, especificar y realizar de manera clara los requerimientos del usuario. (Véase caso de estudio pág. 121)

## **1.5 DELIMITACIONES**

**1.5.1 Geográficas.** El desarrollo de la investigación se realizará en la Universidad Francisco de Paula Santander Ocaña sede la granja ubicada vía al Algodonal, PBX 5690088.

**1.5.2 Temporales.** El tiempo necesario para el desarrollo de este proyecto de investigación, será de un (1) semestre académico, que corresponde aproximadamente a seis (6) meses calendario.

**1.5.3 Conceptuales.** Esta investigación tiene como conceptos fundamentales las metodologías ágiles: XP, SCRUM Y OPENUP y la herramienta de análisis y comparación Dimensional Analytical Tool (4-DAT) correspondientes al área de desarrollo de software.

**1.5.4 Operativas.** Las posibles dificultades que se pueden encontrar en este proyecto son:

El modelo presentado no supere las exigencias de la herramienta 4-DAT

## 2. MARCO REFERENCIAL

### 2.1 MARCO HISTORICO

**2.1.1 Origen de la ingeniería del software.**<sup>12</sup> El termino ingeniería de software surgió a partir de 1968, su objetivo era resolver la llamada crisis del software que tuvo lugar desde finales de 1960 hasta mediados de 1980, esto hacía referencia a los problemas que se tenían al desarrollar nuevos proyectos de software. Ya que iban apareciendo nuevo hardware y sistemas y aplicaciones más complejos lo que hacía que el software no lo soportara y generaran pérdidas y discordia por parte del cliente.

El término “ingeniería de software” se atribuye a Edsger Dijkstra quien lo utilizo en su libro *The Humble Programmer* y fue usado en el año 1968 durante una reunión de la OTAN por la crisis del software, que con el tiempo ha venido evolucionando en cuanto a técnicas y mejores prácticas para las distintas fases de la construcción de software.

**2.1.2 Origen de las metodologías tradicionales.**<sup>13</sup> A partir de los años 80 se empezó a proponer que la mejor manera para desarrollar un sistema de software era mediante una planificación rigurosa del proyecto soportada por herramientas CASE (Ingeniería de Software Asistida por Computador), estas metodologías eran “pesadas” en planificación, diseño y documentación, lo cual absorbía gran parte del tiempo destinado al desarrollo del sistema.

**2.1.3 Origen de las metodologías ágiles.**<sup>14</sup> A partir de 1990 y con el descontento por los métodos de desarrollo de software existentes que tenían muchas limitaciones y eran robustos y pesados se empezó a usar el término metodología ágil que permitían centrarse en el software y no en la documentación, además se ajustaban a cambios de requerimientos en cualquier fase del proyecto sin tener que rehacer todo lo hecho había algún cambio. A partir de esto han nacido diferentes tipos de métodos ágiles todos con procesos y fases diferentes.

**2.1.4 Estado del arte.**<sup>15</sup> Las metodologías ágiles suman gran importancia al desarrollo de software, con ellas se ha logrado no solo que el cliente interactúe más en el desarrollo del proyecto sino que también el equipo de trabajo sea unido y solucionen los problemas que se van presentando de manera rápida. Una de las grandes ventajas de las metodologías ágiles es su gran adaptabilidad a los cambios de requerimientos en cualquier fase del proyecto lo que hace que se vuelva más llamativo para quienes lo empiezan a conocer.

---

<sup>12</sup> ASTEASUAIN Fernando. Lenguaje de Modelado Unificado UML, GRADI. 2009

<sup>13</sup> Pressman, R. (1998). INGENIERÍA DEL SOFTWARE UN ENFOQUE PRACTICO. Mexico: Mc Graw Hill.

<sup>14</sup> E. KENDALL, Kenneth y E. KENDALL, Julie. Análisis y diseño de sistemas. Pearson Education, México, 2005

<sup>15</sup> Stamelos, Ioannis y Sfetsos, Pinagiotis. Agile Software Development Quality Assurance. Information Science Reference. USA, 2007.

Algunos estudios seleccionan los métodos ágiles por diferentes criterios, se ve como están documentadas y la manera y el porcentaje en que son utilizadas.

Las clasificaciones son:<sup>16</sup>

La metodología con mayor presencia en Internet.

La metodología mejor documentada.

Metodologías certificadas y con training.

Metodologías con comunidades.

Metodología más utilizada por empresas. Presencia empresarial.

Metodología más utilizada en proyectos software.

Todo este hace que se haga fácil escoger el uso de una en particular aunque no quiere decir que sea la mejor.

El uso de métodos ágiles es apoyado en gran porcentaje ya que minimiza costos, aumenta productividad, corrige errores en cada iteración y logra un gran grado de satisfacción en el cliente, una encuesta realizada a un grupo de organizaciones desarrolladoras de software a tiempo completo nos confirma lo anterior.

Se puede ver los resultados que arrojó a continuación:

- 84.7% de los encuestados calificaron su conocimiento Agile como promedio o por encima. Se tiene clasificado estos encuestados como "bien informados" para el propósito de la encuesta
- 49% sostenía que los costes se redujeron o se reducen significativamente
- 46% declaró que los costos se mantuvieron sin cambios
- 95% afirmó que no había ningún efecto o una reducción de costes
- 5% declaro que procesos ágiles tuvieron un efecto negativo en el costo.
- 93% indica que la productividad era mejor o significativamente mejor
- 88% indica que la calidad era mejor o significativamente mejor
- 83% afirmó que la satisfacción de negocios era mejor o mucho mejor, encuestados conocedores eran mucho más a favor de los procesos ágiles. Sólo 1,8% de los encuestados conocedores encontraron degradaciones de productividad, pero esto aumentado a 3,1% cuando se toma entre todos los encuestados.”

También se debe resaltar el uso de la herramienta 4-DAT para el análisis de métodos ágiles, existen comparaciones evaluativas entre XP y SCRUM utilizando la herramienta de Dimensional Analítica (4-DAT), basado en cuatro perspectivas de caracterización: los de alcance, la agilidad, los valores ágiles y proceso de software. Un informe generado con la ayuda de 4-DAT ayudará a las organizaciones en la toma de decisiones acerca de la selección o la adopción de un método ágil.

Es una herramienta completa que nos muestra 4 dimensiones:

---

<sup>16</sup> Tinoco, O. Rosales, P. & Salas J. (2010). Criterios de selección de metodologías de desarrollo de software. Sistema e informática , 1, 80.

En la primera se ve el método de caracterización donde se hacen interrogantes tales como el tamaño del equipo, tamaño del proyecto, estilo del código, etc.

La segunda dimensión es la agilidad que posee características tales como flexibilidad, velocidad, delgadez, etc.

La tercera dimensión hace referencia a los "valores ágiles", dimensión tres, es un conjunto de seis valores: cuatro de ellos son proporcionados por el Manifiesto Ágil, con el quinto valor ágil proporcionada por Koch. El sexto valor "mantener el costo efectivo proceso" fue propuesto en Qumer y Henderson-Sellers sobre la base de su estudio de los diversos métodos ágiles. Dimensión tres examina el apoyo de los valores ágiles en diferentes prácticas de los métodos ágiles.

La dimensión cuatro tiene dos componentes principales del proceso de software: proceso de ingeniería de producto y proceso de gestión de procesos. El primero tiene un período de tres nuevas categorías: proceso de desarrollo, proceso de gestión de proyectos y procesos de apoyo.<sup>17</sup>

Los métodos ágiles ofrecen un sin número de características una de las principales es la sencillez tanto en su aprendizaje como en su aplicación en un equipo de desarrollo. Esto conlleva a que cada día sean usadas un poco más.

En febrero de 2001, tras una reunión celebrada en Utah-EEUU, nace el término "ágil" aplicado al desarrollo de software. En esta reunión participan un grupo de 17 expertos de la industria del software, incluyendo algunos de los creadores o impulsores de metodologías de software. Su objetivo fue esbozar los valores y principios que deberían permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto. Se pretendía ofrecer una alternativa a los procesos de desarrollo de software tradicionales, caracterizados por ser rígidos y dirigidos por la documentación que se genera en cada una de las actividades desarrolladas. A partir de esta reunión nace el manifiesto ágil, que será descrito más adelante.

## **2.2 MARCO CONCEPTUAL**

**2.2.1 Software.**<sup>18</sup> Un software es un programa de ordenador, formado por una serie de instrucciones con un propósito en específico.

**2.2.2 Ingeniería del software.**<sup>19</sup> La ingeniería del software es una disciplina de la ingeniería cuya finalidad es definir métodos, etapas, técnicas y mejores prácticas para la creación de software.

---

<sup>17</sup> A Qumer, B Henderson-Sellers, An evaluation of the degree of agility in six agile methods and its applicability for method engineering

<sup>18</sup> CAMPDERRICH FALGUERAS, Benet. Ingeniería del Software. Barcelona, España, UOC, 2003.

<sup>19</sup> CAMPDERRICH FALGUERAS, Benet. Ingeniería del Software. Barcelona, España, UOC, 2003.

**2.2.3 Dimensional Analytical Tool (4-DAT).** Es una herramienta diseñada para comparar y analizar métodos ágiles.

**2.2.4 Métodos ágiles.**<sup>20</sup> Son métodos de desarrollo de software basados en el desarrollo iterativo e incremental, se divide en iteraciones, genera el trabajo en equipo y promueve las comunicaciones cara a cara en vez de la documentación.

Los métodos ágiles impulsan generalmente una gestión de proyectos que promueve el trabajo en equipo, la organización y responsabilidad propia, un grupo de buenas prácticas de ingeniería de software que brindan una entrega rápida de software de alta calidad, y un enfoque de negocios que alinea el desarrollo con las necesidades del cliente y los objetivos de la compañía.<sup>21</sup>

Ventajas de los métodos ágiles	Desventajas de los métodos ágiles
Mejoran la satisfacción del cliente	Falta de documentación
Mejora la motivación e implicación del equipo de desarrollo.	Problemas derivados de la comunicación oral
Ahorrar tanto tiempo como costes	Restricciones en cuanto a tamaño de los proyectos
Mayor velocidad y eficiencia	
Alertar rápidamente tanto de errores como de problemas	

**2.2.5 Desarrollo de software.** El desarrollo de software es la conceptualización, análisis, diseño y construcción de aplicaciones de software a través del uso de procesos, métricas y herramientas, en un amplio rango de arquitecturas y tecnologías, con el objetivo de contribuir a la eficiencia de los procesos estratégicos, operativos y de soporte de una organización.

**2.2.6 Metodologías tradicionales.** Las metodologías tradicionales empezaron a surgir en 1980 con ellas se pretendía ayudar a los profesionales indicando pautas para realizar y documentar cada una de las tareas de desarrollo de software.

**2.2.7 Las metodologías tradicionales (formales)** se focalizan en documentación, planificación y procesos. (Plantillas, técnicas de administración, revisiones, etc.).

**2.2.8 Programación extrema.**<sup>22</sup> La programación extrema o eXtreme Programming (XP) hace parte de las metodologías ágiles o ligeras de desarrollo de software, formulado por Kent Beck, con las principales características de ser de entregas limitadas o pequeñas, semana de trabajo de 40 horas, cliente en el sitio e involucrado en todo el proceso, y la

<sup>20</sup> Abrahamsson, P., Salo, O., Ronkainen, J., Warsta, J. "Agile software development methods Review and analysis". VTT Publications. 2002.

<sup>21</sup> Rivadeneira Molina, Silvia. METODOLOGÍAS ÁGILES ENFOCADAS AL MODELADO DE REQUERIMIENTOS. En: Revista de informe científicos técnicos. ICT-UNPA-57-2013 ISSN: 1852 - 4516

<sup>22</sup> MANN, Mik. Ingeniería del Software. Mikel Angoar.



programación realizada en parejas para así agilizar el tiempo de desarrollo del software, generando poca documentación a diferencia de las metodologías tradicionales sin dejar de ser de buena calidad.

XP posee una serie de artefactos tales y como son:

**Historias de Usuario.**<sup>23</sup> Representan una breve descripción del comportamiento del sistema, emplea terminología del cliente sin lenguaje técnico, se realiza una por cada característica principal del sistema, se emplean para hacer estimaciones de tiempo y para el plan de lanzamientos, reemplazan un gran documento de requisitos y presiden la creación de las pruebas de aceptación. Estas deben proporcionar sólo el detalle suficiente como para poder hacer razonable la estimación de cuánto tiempo requiere la implementación de la historia, difiere de los casos de uso porque son escritos por el cliente, no por los programadores, empleando terminología del cliente. "Las historias de usuario son más "amigables" que los casos de uso formales".

Las Historias de Usuario tienen tres aspectos:

**Tarjeta.** En ella se almacena suficiente información para identificar y detallar la historia.

**Conversación.** Cliente y programadores discuten la historia para ampliar los detalles (verbalmente cuando sea posible, pero documentada cuando se requiera confirmación)

**Pruebas de Aceptación.** Permite confirmar que la historia ha sido implementada correctamente.

**Tarjetas CRC (Clase - Responsabilidad – Colaborador).** Estas tarjetas se dividen en tres secciones que contienen la información del nombre de la clase, sus responsabilidades y sus colaboradores. En la siguiente figura se muestra cómo se distribuye esta información.

**2.2.9 SCRUM.**<sup>24</sup> Es un método ágil que puede ser aplicado a casi cualquier proyecto; Sin embargo, este método es más comúnmente utilizado en el desarrollo de software. El proceso de Scrum es adecuado para los proyectos que cambian rápidamente de requisitos. Progresa a través de una serie de iteraciones llamadas sprints, que duran de uno a cuatro semanas. Este modelo sugiere cada sprint comienza con una breve reunión de planificación y concluye con una revisión.

Sus artefactos son: Product backlog, Sprint backlog y Burndown Chart.

**2.2.10 OPENUP.**<sup>25</sup> OpenUP es un proceso unificado que aplica enfoques iterativos e incrementales dentro de un ciclo de vida estructurado. OpenUP abraza una filosofía

---

<sup>23</sup> E. KENDALL, Kenneth y E. KENDALL, Julie. Análisis y diseño de sistemas. Pearson Education, México, 2005.

<sup>24</sup> Disponible en: <https://www.scrumalliance.org/>

<sup>25</sup> Disponible en: [epf.eclipse.org/wikis/openup/](http://epf.eclipse.org/wikis/openup/)

pragmática, ágil, que se centra en la naturaleza colaborativa de desarrollo de software. Puede ser utilizado en una amplia variedad de proyectos.

Por otra parte Un artefacto en OpenUp se considera a todo aquello que una tarea necesita para realizar su función, o bien la produce o modifica. Los distintos roles existentes son los encargados de crear y actualizar los artefactos.

## **2.3 MARCO TEORICO**

## **2.4 MARCO LEGAL**

**Propiedad intelectual en Colombia.** La Propiedad Intelectual es la denominación asignada a la protección jurídica del Estado sobre bienes inmateriales específicos; estos últimos hacen referencia a toda creación del talento o del ingenio humano, dentro del ámbito científico, artístico, literario, industrial o comercial. Así, mediante la regulación de diversos instrumentos como la Constitución Política, leyes, decretos y tratados internacionales se procura la defensa del intelecto.

La Constitución Política de 1991, en su artículo 61 dispone:

“El Estado protegerá la propiedad intelectual por el tiempo y mediante las formalidades que establezca la ley”.

Es debido aclarar que si bien es una protección jurídica, las demás disposiciones concernientes al tema no realizan una denominación conceptual y literal de la Propiedad Intelectual, sino que hacen referencia a los dos tipos de protección que la componen, a saber: los Derechos de Autor y la Propiedad Industrial.

La diferencia fundamental entre estos dos tipos de propiedad intelectual, es que mientras los Derechos de Autor protegen el medio en el que va la creación y el ingenio artístico, durante toda la vida del autor más un tiempo adicional, la Propiedad Industrial protege la idea sólo en el caso de que ésta tenga una aplicación industrial, por un tiempo más corto y limitado que asegura su explotación económica exclusiva. En ambos casos, una vez transcurrido el tiempo de protección otorgado por el Estado, las creaciones pasan a ser de Dominio Público, siempre teniendo que ser reconocido su creador.

A continuación se adelanta la descripción de las disposiciones jurídicas dictadas en la materia, las cuales aplican para todo el territorio nacional, siempre teniendo claro que estas se sustentan en disposiciones internacionales que intentan una protección similar, propendiendo por la cooperación de los diversos países en la determinación de una aplicación uniforme.

Contexto Normativo de los Derechos de Autor

Ley 23 de 1982. Constituida como la ley “sobre Derechos de Autor”, los sujetos protegidos por dicho cuerpo normativo, serán los autores de obras literarias, científicas y artísticas, los cuales gozarán de protección para sus obras en la forma prescrita en esta. En adición, comprende a los intérpretes o ejecutantes, a los productores de programas y a los organismos de radiodifusión, en sus derechos conexos a los del autor; los causahabientes, a título singular o universal, de los titulares, anteriormente citados; y a la persona natural o jurídica que, en virtud de contrato obtenga por su cuenta y riesgo, la producción de una obra científica, literaria o artística realizada por uno o varios autores en las condiciones previstas en el artículo 20 de esta Ley.

Los derechos de autor recaen sobre las obras científicas literarias y artísticas, las cuales comprenden todas las creaciones del espíritu en el campo científico, literario y artístico, cualquiera que sea el modo o forma de expresión y cualquiera que sea su destinación (artículo 2). La protección otorgada al autor tiene como título originario la creación intelectual, sin que se requiera registro alguno. De esta manera, queda claro cómo se protege exclusivamente la forma o medio por el cual las ideas del autor son descritas, explicadas o ilustradas, mas no son objeto de protección las ideas, contenidos conceptuales, ideológicos o técnicos como tal.

En el artículo 3, los derechos de autor comprenden para sus titulares las facultades exclusivas:

De disponer de su obra a título gratuito u oneroso bajo las condiciones lícitas que su libre criterio les dicte.

De aprovecharla, con fines de lucro o sin él, por medio de la imprenta, grabado, copias, molde, fonograma, fotografía, película cinematografía, ideograma, y por la ejecución, recitación, representación, traducción, adaptación, exhibición, transmisión, o cualquier otro medio de reproducción, multiplicación, o difusión conocido o por conocer.

De ejercer las prerrogativas, aseguradas por esta Ley en defensa de su "derecho moral".

Siguiendo con la protección del autor, según el artículo 21, ésta perdura durante toda su vida, más 80 años después de su muerte. En caso de ser una obra literaria, científica o artística cuyo titular sea una persona jurídica o una entidad oficial o cualquier institución de derecho público, se considerará que el plazo de protección será de 30 años contados a partir de su publicación (artículo 27).

Según el artículo 29 (modificado por el artículo 2 de la ley 44 de 1993), la protección consagrada por la presente ley a favor de los artistas, intérpretes y ejecutantes, de los productores de fonogramas y de los organismos de radiodifusión, será de ochenta años a partir de la muerte del respectivo titular, si este fuere persona natural; si el titular fuere persona jurídica, el término será de treinta años a partir de la fecha en que tuvo lugar la interpretación o ejecución o la primera fijación del fonograma, o la emisión de la radiodifusión.

Por otra parte, de vital importancia resulta la subdivisión que realiza la ley en cuanto al contenido del derecho de autor, pues comprende tanto los derechos morales como los patrimoniales. En primer lugar, los derechos morales consisten en el reconocimiento de la paternidad del autor sobre la obra realizada y el respeto a la integridad de la misma. En el artículo 30, el autor tendrá sobre su obra un derecho perpetuo, inalienable, e irrenunciable para:

- Reivindicar en todo tiempo la paternidad de su obra y, en especial, para que se indique su nombre seudónimo cuando se realice cualquiera de los actos mencionados en el artículo 12 de esta Ley.

A oponerse a toda deformación, mutilación u otra modificación de la obra, cuando tales actos puedan causar o acusen perjuicio a su honor o a su reputación, o la obra se demerite, y a pedir reparación por esto.

A conservar su obra inédita o anónima hasta su fallecimiento, o después de él cuando así lo ordenase por disposición testamentaria.

A modificarla, antes o después de su publicación.

A retirarla de la circulación o suspender cualquier forma de utilización aunque ella hubiere sido previamente autorizada.

Según su parágrafo 1, los derechos anteriores no pueden ser renunciados ni cedidos. Los autores al transferir o autorizar el ejercicio de sus derechos patrimoniales no conceden sino los de goce y disposición a que se refiere el respectivo contrato, conservando los derechos consagrados en el presente artículo.

En segundo lugar, los derechos patrimoniales consisten en la facultad de aprovechar y disponer económicamente la obra, motivo por el cual, según el artículo 72 de la mencionada ley, el derecho patrimonial del autor se causa desde el momento en que la obra o producción, susceptible de estimación económica y cualquiera que sea su finalidad, se divulgue por cualquier forma o modo de expresión.

En todos los casos en que los autores o las asociaciones de autores, celebren contratos con los usuarios o con las organizaciones que los representen, respecto al derecho de autor, por concepto de ejecución, representación, exhibición y en general, por uso o explotación de las obras protegidas por la presente Ley, serán las tarifas concertadas en los respectivos contratos, las que tendrán aplicación, siempre que no sean contrarias a los principios consagrados por la misma (artículo 73).

Así mismo, el artículo 78 es claro en disponer que la interpretación de los negocios jurídicos sobre derechos de autor sea siempre restrictiva. En este sentido, no se admite el

reconocimiento de derechos más amplios de los expresamente concedidos por el autor en el instrumento respectivo.

Respecto a este tema, el artículo 20, modificado por el art. 28 de la Ley 1450 de 201115, disponía: “cuando uno o varios autores, mediante contrato de servicios, elaboren una obra según plan señalado por persona natural o jurídica y por cuenta y riesgo de ésta, solo percibirán, en la ejecución de ese plan, los honorarios pactados en el respectivo contrato. Por este solo acto, se entiende que el autor o autores transfieren los derechos sobre la obra, pero conservarán las prerrogativas consagradas en el artículo 30 de la presente Ley, en sus literales a), y b)”.

Actualmente, con la modificación mencionada anteriormente, el artículo 20 relativo a propiedad intelectual en cumplimiento de un contrato de prestación de servicios o de un contrato de trabajo, en su nueva redacción dispone: “En las obras creadas para una persona natural o jurídica en cumplimiento de un contrato de prestación de servicios o de un contrato de trabajo, el autor es el titular originario de los derechos patrimoniales y morales; pero se presume, salvo pacto en contrario, que los derechos patrimoniales sobre la obra han sido transferidos al encargado o al empleador, según sea el caso, en la medida necesaria para el ejercicio de sus actividades habituales en la época de creación de la obra. Para que opere esta presunción se requiere que el contrato conste por escrito. El titular de las obras de acuerdo a este artículo podrá intentar directamente o por intermedia persona acciones preservativas contra actos violatorios de los derechos morales informando previamente al autor o autores para evitar duplicidad de acciones.”

La modificación incorporada por el artículo 28 de la Ley 1450 de 2011 al artículo 20 de la Ley 23 de 1982 pone fin al debate antes existente sobre la presunción de cesión de derechos en los contratos de prestación de servicios que constan por escrito. Este tema ya había sido objeto de aclaración por vía de jurisprudencia por parte de la Corte Constitucional, al decidir la exequibilidad del artículo 20 de la Ley 23 de 198216, donde el alto tribunal se pronunció en el sentido en que, para que opere la presunción señalada, se requiere que el contrato conste por escrito. Adicionalmente, si el “encargante” o empleador quiere disponer de los derechos sobre la obra para otro tipo de actividades, más allá del giro ordinario de sus negocios, deberá obtener una autorización adicional del autor y, si es del caso, pagarle una remuneración adicional por concepto de derechos de autor”.

Debe hacerse claridad en que las distintas formas de utilización de una obra son independientes entre sí, pues pueden presentarse diversas formas, tales como: la reproducción, traducción, adaptación, etc. Sobre estas formas, el autor o titular de los derechos patrimoniales puede cederlos, lo cual implica entregarlos a otra persona, o en su defecto autorizar su uso, donde si bien se da un permiso, no se ceden los derechos, y se consideran aspectos como el tiempo, la cobertura y la retribución económica.

En materia de licencias que se pueden otorgar, en el artículo 58 dispone que cualquier persona natural o jurídica podrá pedir a la autoridad competente, una vez expirados los plazos que se fijan en el presente artículo, una licencia para reproducir y publicar una

edición determinada de una obra en forma impresa o en cualquier forma análoga de reproducción.

No se podrá conceder ninguna licencia antes de que expire uno de los plazos siguientes, calculados a partir de la primera publicación de la edición de la obra sobre la que se solicite dicha licencia:

Tres años para las obras que traten de ciencias exactas y naturales, comprendidas las matemáticas y la tecnología.

Siete años para las obras de imaginación, como las novelas, las obras poéticas, gramáticas y musicales y para los libros de arte.

Cinco años para todas las demás obras.

Como el artículo 67 lo dispone, la licencia llevará consigo, en favor del titular del derecho de reproducción, una remuneración equitativa y ajustada a la escala de derecho que normalmente se pagan en el caso de licencias libremente negociadas entre los interesados en el país y los titulares de los derechos de reproducción en el país del titular del derecho a que se refiere dicha licencia.

Ley 44 de 1993. Esta ley, “por la cual se modifica y adiciona la Ley 23 de 1982 y se modifica la Ley 29 de 1944”, realiza cambios importantes en la ley de Derechos de Autor como: El artículo 2, modifica el artículo 29 de la Ley 23 de 1982, ampliando el término de los derechos conexos consagrados cuando el titular sea persona jurídica, pasando de un término de protección de 30 a 50 años, contados a partir del último día del año en que se tuvo lugar la interpretación o ejecución, la primera publicación del fonograma o, de no ser publicado, de su primera fijación, o la emisión de su radiodifusión.

El artículo 6 dispone que todo acto en virtud del cual se enajene el Derecho de Autor, o los Derechos Conexos así como cualquier otro acto o contrato vinculado con estos derechos, deba ser inscrito en el Registro Nacional del Derecho de Autor como condición de publicidad y oponibilidad ante terceros.

A su vez, el artículo 67 adiciona el artículo 2 de la Ley 23 de 1982 de la siguiente manera: "Los derechos de autor se reputan de interés social y son preferentes a los de los intérpretes o ejecutantes, de los productores de fonogramas y de los organismos de radiodifusión, y en caso de conflicto primarán los derechos del autor".

Decisión Andina 351 de 1993. Mediante esta decisión de la Comisión del Acuerdo de Cartagena, aplicable dentro del ordenamiento interno colombiano, “Por la cual se adopta el régimen común sobre derechos de autor y conexos”, se afirma en su artículo 1 que las disposiciones de esta Decisión tienen como finalidad reconocer una adecuada y efectiva protección a los autores y demás titulares de derechos, sobre las obras del ingenio, en el

campo literario, artístico o científico, cualquiera que sea el género o forma de expresión y sin importar el mérito literario o artístico ni su destino.

Es por esto que cada País Miembro deberá conceder a los nacionales de otro país una protección no menos favorable que la reconocida a sus propios nacionales en materia de Derecho de Autor y Derechos Conexos. En concordancia con el objeto de protección de esta decisión, según el artículo 4, este recae sobre todas las obras literarias, artísticas y científicas que puedan reproducirse o divulgarse por cualquier forma o medio conocido o por conocer. Siendo así, queda protegida exclusivamente la forma mediante la cual las ideas del autor son descritas, explicadas, ilustradas o incorporadas a las obras. En lo referente a los titulares de los derechos, se presume autor, salvo prueba en contrario, la persona cuyo nombre, seudónimo u otro signo que la identifique, aparezca indicado en la obra. Sin embargo, para el artículo 9, una persona natural o jurídica, distinta del autor, podrá ostentar la titularidad de los derechos patrimoniales sobre la obra de conformidad con lo dispuesto por las legislaciones internas de los Países Miembros. En adición, para el artículo 10 las personas naturales o jurídicas ejercen la titularidad originaria o derivada, de conformidad con la legislación nacional, de los derechos patrimoniales de las obras creadas por su encargo o bajo relación laboral, salvo prueba en contrario. De acuerdo con lo previsto en el artículo 13, el autor o, en su caso, sus derechohabientes, tienen el derecho exclusivo de realizar, autorizar o prohibir:

La reproducción de la obra por cualquier forma o procedimiento.

La comunicación pública de la obra por cualquier medio que sirva para difundir las palabras, los signos, los sonidos o las imágenes.

La distribución pública de ejemplares o copias de la obra mediante la venta, arrendamiento o alquiler.

La importación al territorio de cualquier País Miembro de copias hechas sin autorización del titular del derecho.

La traducción, adaptación, arreglo u otra transformación de la obra.

Por último, dentro de las disposiciones más significativas de su contenido, el artículo 60 argumenta que los derechos sobre obras que no gozaban de protección conforme a las normas legales nacionales anteriores a la presente Decisión, por no haber sido registradas, gozarán automáticamente de la protección reconocida por ésta, sin perjuicio de los derechos adquiridos por terceros con anterioridad a la entrada de vigencia de la misma, siempre que se trate de utilidades ya realizadas o en curso en dicha fecha.

### **Tipos de licencias de software libre**

**GPL.** La Licencia Pública General de GNU es una licencia libre, sin derechos para software y otro tipo de obras.

Las licencias para la mayoría del software y otros trabajos prácticos están diseñadas para quitarle a usted la libertad de compartir y modificar esos trabajos. Por el contrario, la Licencia Pública General de GNU pretende garantizarle la libertad de compartir y

modificar todas las versiones de un programa - para asegurarse de que sigue siendo software libre para todos sus usuarios. Nosotros, la Fundación para el Software Libre, usamos la Licencia Pública General GNU para la mayoría de nuestro software, sino que también se aplica a cualquier trabajo realizado de la misma manera por sus autores. Usted también puede aplicarla a sus programas.

Cuando hablamos de software libre, estamos refiriéndonos a libertad, no a precio. Nuestras Licencias Públicas Generales están diseñadas para asegurarnos de que tenga la libertad de distribuir copias de software libre (y cobrar por ello si lo desea), que reciba el código fuente o que pueda conseguirlo si lo quiere, de que pueda modificar el software o usar fragmentos de él en nuevos programas libres, y de que sepa que puede hacer estas cosas.

Para proteger sus derechos, necesitamos evitar que otros le nieguen estos derechos o pedirle que renuncie a ellos. Por lo tanto, usted tiene ciertas responsabilidades si distribuye copias del software, o si lo modifica: responsabilidades de respetar la libertad de los demás.

Por ejemplo, si distribuye copias de un programa, ya sea gratuitamente a cambio de una contraprestación, debe transmitir a los destinatarios los mismos derechos que usted recibió. Debe asegurarse de que ellos también reciben, o pueden conseguir, el código fuente. Y debe mostrarles estas condiciones de forma que conozcan sus derechos.

Los desarrolladores que usan la GPL GNU protegen sus derechos de dos formas: (1) imponen derechos al software, y (2) le ofrecemos esta licencia que le da permiso legal para copiar, distribuir y / o modificar la misma.

Para la protección de los desarrolladores y autores, la GPL expone claramente que no hay ninguna garantía para este software libre. Por el amor de ambos, usuarios y autores, la GPL establece que las versiones modificadas ser identificadas como tales, por lo que cualquier problema no sea atribuido por error a los autores de las versiones anteriores.

Algunos dispositivos están diseñados para negar al usuario para instalar o ejecutar versiones modificadas del software que usan internamente, aunque el fabricante sí pueda hacerlo. Esto es fundamentalmente incompatible con el objetivo de proteger la libertad de los usuarios para cambiar el software. El patrón de abuso sistemático ocurre en el área de productos para particulares de usar, que es precisamente donde es más inaceptable. Por lo tanto, hemos diseñado esta versión de la GPL para prohibir la práctica de esos productos. Si apareciesen problemas similares en otros ámbitos, estamos dispuestos a extender esta disposición a aquellos dominios en futuras versiones de la GPL, según sea necesario para proteger la libertad de los usuarios.

Por último, todo programa está constantemente amenazado por las patentes de software. Los Estados no deberían permitir que las patentes restrinjan el desarrollo y uso de software en ordenadores de uso general, pero en los que sí, queremos evitar el peligro especial de que las patentes aplicadas a un programa libre puedan hacerlo propietario. Para evitar esto,



la GPL establece que las patentes no se puede utilizar para hacer que el programa no-libre. Los términos exactos y las condiciones para la copia, distribución y modificación.

**AGPL.** La Licencia Pública General de Affero (en inglés Affero General Public License, también Affero GPL o AGPL) es una licencia copyleft derivada de la Licencia Pública General de GNU diseñada específicamente para asegurar la cooperación con la comunidad en el caso de software que corra en servidores de red.

La Affero GPL es íntegramente una GNU GPL con una cláusula nueva que añade la obligación de distribuir el software si éste se ejecuta para ofrecer servicios a través de una red de ordenadores.

La Free Software Foundation recomienda que el uso de la GNU AGPLv3 sea considerado para cualquier software que usualmente corra sobre una red

**LGPL.** La Licencia Pública General Reducida de GNU, o más conocida por su nombre en inglés GNU Lesser General Public License (antes GNU Library General Public License o Licencia Pública General para Bibliotecas de GNU), o simplemente por su acrónimo del inglés GNU LGPL, es una licencia de software creada por la Free Software Foundation que pretende garantizar la libertad de compartir y modificar el software cubierto por ella, asegurando que el software es libre para todos sus usuarios.

Esta licencia permisiva se aplica a cualquier programa o trabajo que contenga una nota puesta por el propietario de los derechos del trabajo estableciendo que su trabajo puede ser distribuido bajo los términos de esta "LGPL Lesser General Public License". El "Programa", utilizado en lo subsecuente, se refiere a cualquier programa o trabajo original, y el "trabajo basado en el Programa" significa ya sea el programa o cualquier trabajo derivado del mismo bajo la ley de derechos de autor: es decir, un trabajo que contenga el Programa o alguna porción de él, ya sea íntegra o con modificaciones o traducciones a otros idiomas.

Otras actividades que no sean copia, distribución o modificación no están cubiertas en esta licencia y están fuera de su alcance. El acto de ejecutar el programa no está restringido, y la salida de información del programa está cubierto sólo si su contenido constituye un trabajo basado en el Programa (es independiente de si fue resultado de ejecutar el programa). Si esto es cierto o no depende de la función del programa.

### **3. DISEÑO METODOLOGICO**

Para comenzar el desarrollo de este proyecto, se empezó por describir y conocer las características esenciales de XP, SCRUM y OPENUP tales como son ciclo de vida, artefactos, roles y particularidades de cada una, a partir de esto se comparó cada uno de estos métodos para concluir que tienen en común y definir cuál de sus características sería la mejor al momento de crear un nuevo método, luego se definió el método y fue evaluado con la herramienta de análisis y comparación 4-DAT para saber si cumple con los requisitos para ser un método ágil.

#### **3.1 TIPO DE INVESTIGACIÓN**

En este proyecto se utilizó la investigación descriptiva, ya que a través de la descripción exacta de las fases, artefactos y roles se podrán identificar las pautas a seguir para el desarrollo de una nueva metodología ágil, y de igual manera al definir la herramienta 4 DAT podremos validar la metodología creada.

#### **3.2 POBLACIÓN**

Según Tamayo y Tamayo, "La población se define como la totalidad del fenómeno a estudiar donde las unidades de población poseen una característica común la cual se estudia y da origen a los datos de la investigación".

La población que se tuvo en cuenta para la realización de este proyecto fueron los grupos Desarrolladores de software de la Universidad Francisco de Paula Santander; conformados por los integrantes de proyecto GENESIS, los ingenieros pertenecientes a la división de sistemas y los directores de los grupos de investigación vinculados al programa de ingeniería de sistemas, ya que estos son los principales actores en el uso de estos métodos, lo que permite conocer el grado de aceptación del método planteado.

#### **3.3 MUESTRA**

Tamayo y Tamayo, afirma que la muestra "es el grupo de individuos que se toma de la población, para estudiar un fenómeno estadístico".

Para el desarrollo de esta investigación se utilizó el muestreo no probabilístico, en el cual no se usa el azar, sino el criterio del investigador, quien define si la muestra es representativa o no; lo cual permitió escoger una muestra representativa para la investigación teniendo en cuenta la línea de profundización del encuestado.

Para el caso de este proyecto el tamaño de la muestra fue el mismo que la población dado que las personas encuestadas están comprendidos como se muestra en la tabla 2.

**Tabla 2.** Muestra

<b>Población</b>	<b>Cantidad</b>
GENESIS	6
GRUPOS DE INVESTIGACION	3
DIVISION DE SISTEMAS	4
Total	13

**Fuente.** Autor del proyecto

### 3.4 RECOLECCIÓN DE LA INFORMACIÓN

La recolección de la información se hizo mediante el uso de libros, documentos científicos, estudios y las bases de datos que nos ofrece la biblioteca de la Universidad Francisco de Paula Santander Ocaña.

En este caso se realizaron dos encuesta a los grupos desarrolladores de software en la Universidad Francisco de Paula Santander Ocaña para obtener un estimado sobre el uso de métodos ágiles en dicha institución y el manejo de los proyectos de desarrollo de software (Ver anexo B y anexo C)

### 3.5 ANÁLISIS DE LA INFORMACIÓN

Se hizo un estudio comparativo entre las fases, roles y artefactos de XP, SCRUM y OPENUP.

Además de esto se realizó una encuesta (Anexo B), la cual se hizo para saber en qué porcentaje los grupos de desarrollo de software de la Universidad Francisco de Paula Santander usaban o tenían conocimiento acerca de los métodos ágiles a partir de los resultados obtenidos se decidió crear un método que cubriera la necesidad existente. Después de crear dicho método se realizó una segunda encuesta (Anexo C), con el fin alinear el nuevo método con los usados en la UFPSO.

En total fueron 12 las personas encuestadas, dicha encuesta arrojó los siguientes resultados:

En cuanto al nivel de conocimiento de métodos ágiles se tiene lo siguiente:

**Cuadro 1.** ¿Cómo calificaría usted su nivel de conocimiento en métodos ágiles?

Nivel de conocimiento en métodos ágiles.		
	Numero de respuestas	Equivalente en porcentaje
Muy limitado	0	0
Limitado	6	50%
Medio	6	50%

Extenso	0	0
Muy extenso	0	0

**Fuente.** Autor del proyecto

Lo cual quiere decir que quienes desarrollan software en la UFPSO no poseen un buen grado de conocimiento acerca de los métodos ágiles lo que genera una desventaja ya que estos aportan características importantes a los proyectos tales como: son flexibles, sus proyectos son subdivididos en proyectos más pequeños, incluyen comunicación constante con el cliente, son altamente colaborativos y se adaptan mejor a los cambios.<sup>26</sup>

Para continuar se indago acerca de que métodos ágiles habían empleado estos son los resultados, cabe resaltar que es una pregunta con múltiples respuestas, aquí se seleccionaban todos los métodos que aplicarían:

**Cuadro 2.** ¿Qué métodos ágiles ha empleado?

Métodos ágiles empleados	
	Numero de respuestas
Scrum	5
XP	5
FDD	0
ASD	0
CRYSTAL	0
OPENUP	0
DSDM	0
LSD	0
TDD	0
No lo se	0
No uso métodos ágiles	5
Otro	1

**Fuente.** Autor del proyecto

En el cuadro 2 se logra apreciar que aquellos que dijeron tener conocimiento en el uso de métodos ágiles utilizaban dos de los trabajados en este trabajo XP y SCRUM otro porcentaje aún no ha empezado a desarrollar software con ninguno de estos métodos. Lo cual favorece el método a crear ya que dos de los métodos que lo componen son estos mismos.

Se continúa preguntando sobre en qué medida ha mejorado el rendimiento en los grupos de trabajo que desarrollan los proyectos el resultado es el siguiente:

<sup>26</sup> Ghosh, S. (2012). Systemic comparison of the application of EVM in traditional and agile software project [Internet]. Disponible desde <http://pm.umd.edu/files/public/documents/studentpapers/2011/EVM%20in%20Waterfall%20and%20Agile%20Software%20Project%20by%20Sam%20Ghosh.pdf> [Acceso Junio 1, 2013].

**Cuadro 3.** ¿En qué medida su método adoptado ha mejorado el rendimiento de sus equipos de trabajo?

Rendimiento en equipos de trabajo		
	Numero de respuestas	Equivalente en porcentaje
Empeorado	0	0
Sin cambios	2	18%
Mejorado	5	41%
No lo se	5	41%

**Fuente.** Autor del proyecto

El análisis cualitativo de esta pregunta se hizo a manera de resumen al finalizar la pregunta presentada en el cuadro 6. Después de preguntar sobre rendimiento sobre equipos, se sigue con cuanto mejora el tiempo de entrega cuando se usan métodos ágiles los resultados son los siguientes:

**Cuadro 4.** ¿En qué medida su método adoptado ha mejorado su tiempo de entrega?

Tiempo de entrega		
	Numero de respuestas	Equivalente en porcentaje
Empeorado	0	0
Sin cambios	1	8%
Mejorado	6	50%
No lo se	5	42%

**Fuente.** Autor del proyecto

El análisis cualitativo de esta pregunta se hizo a manera de resumen al finalizar la pregunta presentada en el cuadro 6. La siguiente pregunta hace énfasis a que tanto han mejorado los costos en los proyectos realizados estas son las respuestas:

**Cuadro 5.** ¿En qué medida su método adoptado ha mejorado sus costos?

Costos		
	Numero de respuestas	Equivalente en porcentaje
Empeorado	0	0
Sin cambios	5	42%
Mejorado	1	8%
No lo se	6	50%

**Fuente.** Autor del proyecto

El análisis cualitativo de esta pregunta se hizo a manera de resumen al finalizar la pregunta presentada en el cuadro 6. También se evaluó cuanto mejora la calidad del producto al utilizar métodos ágiles para esto se tiene:

**Cuadro 6.** ¿En qué medida su método adoptado ha mejorado la calidad de su producto final?

Calidad del producto		
	Numero de respuestas	Equivalente en porcentaje
Empeorado	0	0
Sin cambios	0	0
Mejorado	7	59%
No lo se	5	41%

**Fuente.** Autor del proyecto

En conclusión para las respuestas de los cuadros 3, 4, 5 y 6. Se puede decir que aquellos que aplican métodos ágiles están conscientes que estos mejoran el rendimiento, tiempo de entrega y la calidad del producto, elementos tales como responder a los cambios más que seguir estrictamente un plan, estilo de desarrollo y cultura de negocio están totalmente cubiertos, Mike Cohn<sup>27</sup>, un reconocido promotor de las metodologías ágiles, resalta el empoderamiento de los equipos pequeños alineados para alcanzar objetivos de negocio, así como también la capacidad de estos metodos para entregar productos de alta calidad en menor tiempo, obteniendo mayor satisfacción en los clientes. Según un estudio realizado por VersionOne en el 2011<sup>28</sup>, entre los beneficios principales de la adopción de metodos ágiles está: la habilidad para gestionar cambio de prioridades en el proyecto (84%), mejorar la visibilidad del proyecto (77%) y aumentar la productividad (75%). Esto quiere decir que al pretender crear un nuevo método ágil con todas sus características se cubrirían estos inconvenientes y mejora el producto en todos los sentidos.

Se continúa preguntando cual fue o fueron los inconvenientes principales al adoptar los métodos ágiles, esta pregunta fue de selección múltiple:

**Cuadro 7.** ¿Principal problema que tuvo al adoptar los métodos ágiles?

Principal problema al adoptar los métodos ágiles			
	Numero de respuestas	de	Equivalente en porcentaje
Falta de experiencia en métodos	3		25%

<sup>27</sup> Mountain Goat Software. 2009. [Citado el 25 noviembre de 2014.] disponible en: <http://www.mountaingoatsoftware.com/topics/transitioning-to-agile>

<sup>28</sup> [http://www.versionone.com/state\\_of\\_agile\\_development\\_survey/11/](http://www.versionone.com/state_of_agile_development_survey/11/)

agiles		
Poca administración en métodos agiles	3	25%
Filosofía y cultura en la organización	4	34%
Poco entrenamiento en métodos agiles	2	16%
Seguir usando la metodología anterior (Modelo de cascada)	0	0
Ningún problema	0	0

**Fuente.** Autor del proyecto

Según el cuadro 7 el principal problema que tuvieron las personas encuestadas al adoptar los métodos agiles es la cultura a la que están acostumbrados, además de eso la falta de experiencia y la poca administración de los métodos agiles.

Por último se indago acerca de si quienes utilizan estos métodos los seguirán usando a futuro estos fue lo que dijeron:

**Cuadro 8.** ¿Usara métodos agiles en un futuro?

Seguirá usando métodos agiles		
	Numero de respuestas	Equivalente en porcentaje
Si	6	50%
No	0	0
No lo se	6	50%

**Fuente.** Autor del proyecto

Con la realización de esta encuesta se puede concluir que el 50% tiene poco conocimiento en lo que son los métodos agiles, por lo cual no han implementado ninguno de ellos, mientras que otro 50% afirma tener conocimiento básico y haber trabajado con alguno de ellos, como se ve en los cuadros 3 y 4 hay mejora en el tiempo de entrega y la calidad del producto características importantes de estos métodos, el método planteado a continuación surge como respuesta a las falencias existentes en la Universidad Francisco de Paula Santander Ocaña, en cuanto a su desconocimiento de los métodos agiles y al mal manejo que le dan quienes lo usan por su poca información al respecto, se recomienda a los grupos de desarrollo de software de la UFPSO empaparse más de este tema y usar en la gestión de sus proyectos este tipo de procesos ya que pueden mejorar sus trabajos.

#### 4. PRESENTACION DE RESULTADO

Esta investigación tuvo como finalidad crear un método ágil de desarrollo de software basado en XP, SCRUM, OPENUP y validado con la herramienta de análisis 4-dat para mejorar la calidad de los proyectos desarrollados por los grupos de gestión de software de la UFPSO.

**Definir y conocer los conceptos, roles, fases y artefactos de los métodos ágiles que serán evaluados.** El siguiente apartado busca definir lo que es una metodología y dar una introducción al mundo de los métodos tradicionales y los métodos ágiles dando énfasis a XP, SCRUM Y OPENUP además de conocer sus conceptos se hará un análisis detallado de sus roles, ciclo de vida, artefactos y otras particularidades que cada uno posee.

Con esto se busca que el lector pueda adentrarse y entender mejor lo que se busca con el proyecto y porque se decidió usar estos tres métodos.

**Metodología.** Cualquier enfoque aceptado para analizar, diseñar, implementar, probar, mantener y evaluar un sistema de información.<sup>29</sup>

Un método para el desarrollo de software establece un orden y de esa manera se desarrollara el proyecto y ayuda a la forma en que se va a construir.

Lo anterior quiere decir que una metodología encierra varios aspectos que hacen que se pueda desarrollar software de manera más organizada, rápida y efectiva.

Existen muchos tipos de métodos para el desarrollo de software, están **los métodos tradicionales, los métodos híbridos y los métodos ágiles.**

Diferencia entre metodología y método:<sup>30</sup>

**Metodología:** conjunto de procedimientos basados en principios lógicos que se usan para lograr una serie de objetivos en una investigación científica.

**Método:** es un conjunto de actividades, técnicas y acciones, diseñadas para lograr un objetivo de manera más fácil.

El término método se utiliza para el procedimiento que se emplea para alcanzar los objetivos de un proyecto y la metodología es el estudio del método.

---

<sup>29</sup> E. KENDALL, Kenneth y E. KENDALL, Julie. Análisis y diseño de sistemas. Pearson Education, México, 2005.

<sup>30</sup> Diferencias encontrado en internet en: <https://sites.google.com/site/metodololgdelainvestig/modulo-0-2/deferencias-entre>



**Métodos tradicionales**<sup>31</sup>. El desarrollo de software no es una tarea fácil prueba de ello es el sin número de propuestas que existen para su tratamiento, hasta hace unos años el desarrollo de software era dirigido por los llamados métodos tradicionales, estos imponen una disciplina de trabajo mucho más rigurosa con el fin de conseguir un producto mejor elaborado y más eficiente. Hacen énfasis en un proceso detallado, se centran en definir menudamente roles, actividades, artefactos, herramientas y documentación. Entre las metodologías tradicionales se pueden citar:

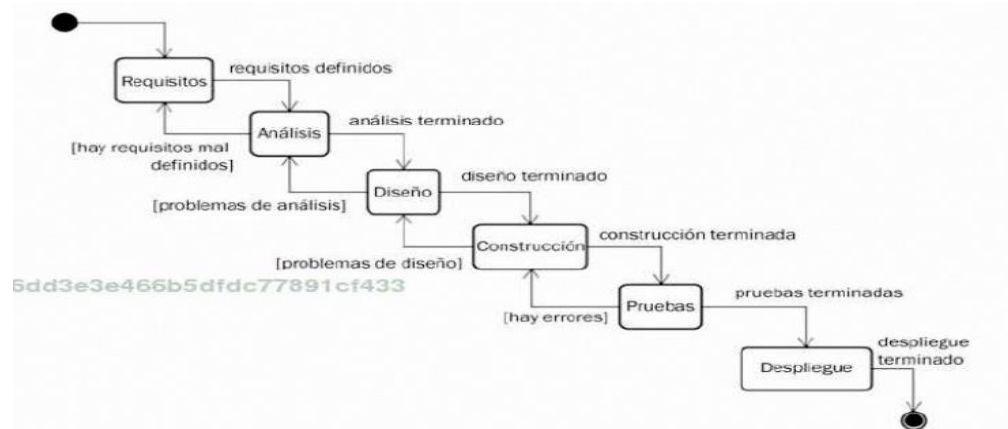
- RUP (Rational Unified Procces)
- MSF (Microsoft Solution Framework)
- Win-Win Spiral Model
- Iconix

Algunos modelos de desarrollo de software que se encuentran son:

**Modelo en cascada.** En este método se mantiene una secuencia ordenada, el trabajo de una entrada previa es la entrada del siguiente proceso lo que hace que sea muy ordenado y puntual con sus fechas de entrega, los requerimientos poseen gran estabilidad y se puede planear con anticipación.

La figura 2 que se muestra a continuación contiene las fases que utiliza el modelo en cascada para desarrollar sus proyectos.

**Figura 2.** Modelo en cascada.<sup>32</sup>



**Fuente.** Fontelo Carlos, METODOLOGIA DE DESARROLLO DE SOFTWARE Y UML, Alfaomega, México (2005)

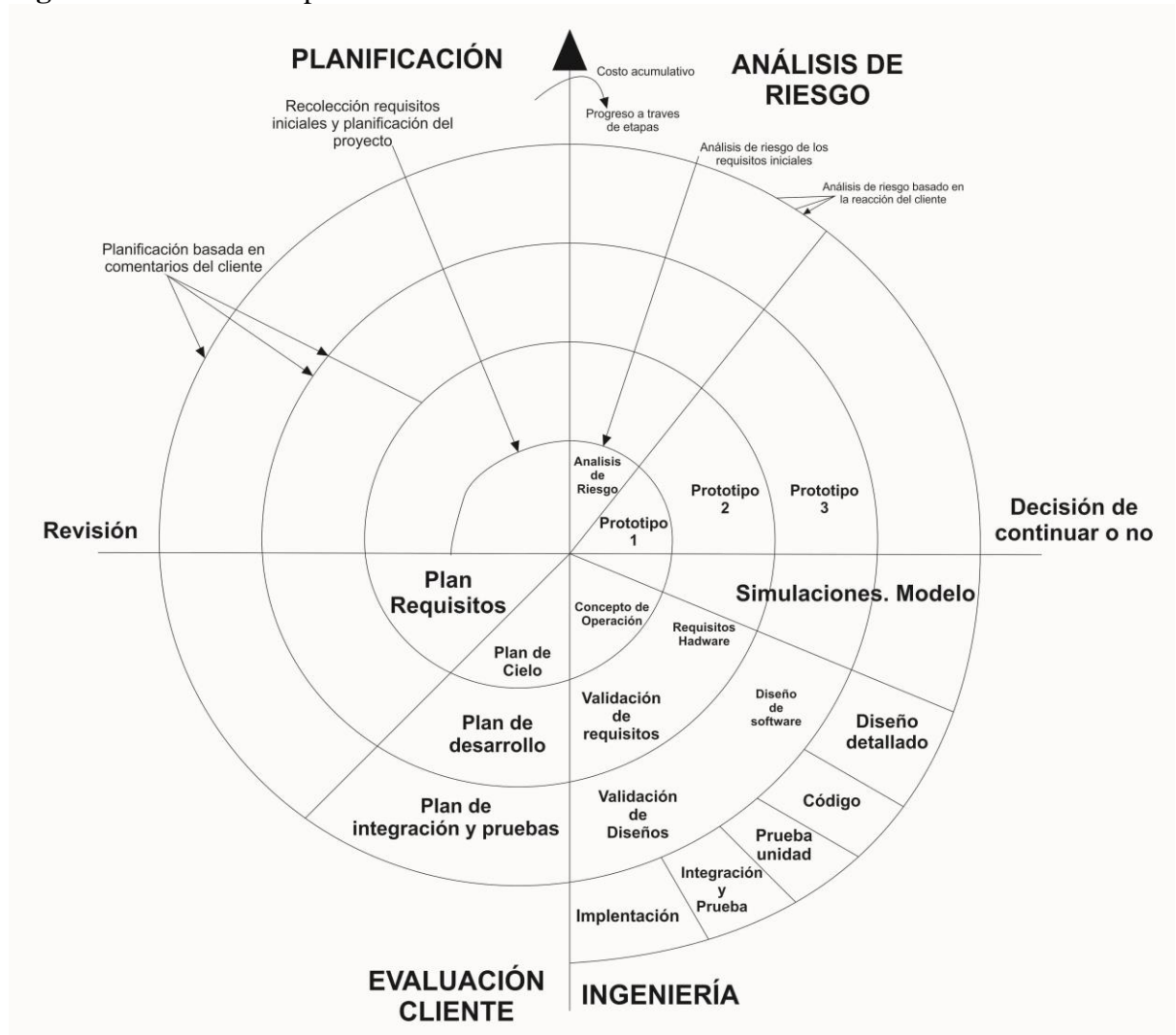
<sup>31</sup> Brito Acuña Karenny, (2009), “SELECCIÓN DE METODOLOGÍAS DE DESARROLLO PARA APLICACIONES WEB EN LA FACULTAD DE INFORMÁTICA DE LA UNIVERSIDAD DE CIENFUEGOS”.

<sup>32</sup> Fontelo Carlos, METODOLOGIA DE DESARROLLO DE SOFTWARE Y UML, Alfaomega, México (2005).

**Modelo en espiral.** Este modelo culmina cada fase con una evaluación de riesgos y un prototipo, con esto se logra saber si debe quedarse en esa fase para corregir errores o pasar a la siguiente, en este método a medida que el costo aumenta los riesgos disminuyen y se pueden incluir otros métodos en cada iteración.

La figura 3 que se muestra a continuación explica cada fase del modelo en espiral y de todas las actividades que se realizan en cada una.

**Figura 3.** Modelo en espiral <sup>33</sup>



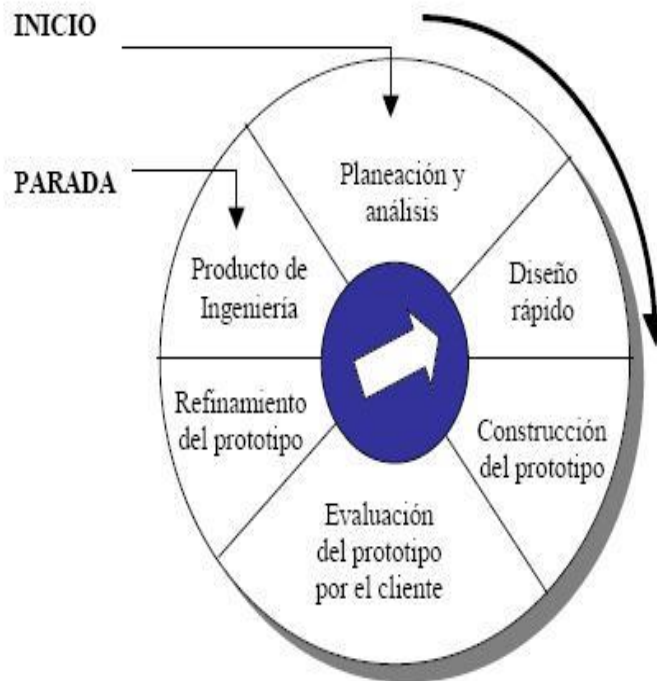
**Fuente.** Amo Fernando, Martínez Loic & Segovia Francisco. Introducción a la ingeniería de software, modelos de desarrollo de programas. Delta publicaciones. Madrid (2005).

<sup>33</sup> Amo Fernando, Martínez Loic & Segovia Francisco. INTRODUCCION A LA INGENIERIA DE SOFTWARE modelos de desarrollo de programas. Delta publicaciones. Madrid (2005).

**Prototipos:** es un método menos formal de desarrollo en este los clientes y analistas se reúnen para sacar los objetivos y requerimientos a partir de esto se planea una iteración para hacer un prototipo este se entrega y es evaluado por los participantes.

La figura 4 que se muestra a continuación muestra en qué fase inicia el modelo en prototipos como se va desarrollando y la fase en que para.

**Figura 4.** Modelo de prototipos<sup>34</sup>



**Fuente.** Zachman, John.(1999), El modelado de las empresas: la arquitectura Zachman, Zachman Institute for Framework Advancement. Estados Unidos.

Para aclarar un poco el contexto se dará un informe detallado de alguno de estos métodos para establecer mejor una comparación con cada uno de los métodos ágiles que se están usando en este caso se definirá RUP (Rational Unified Procces).

**RUP (Rational Unified Procces).**<sup>35</sup> En junio del 1998 se lanza Rational Unified Process RUP (Proceso unificado racional), siendo un método de desarrollo iterativo promovido por la compañía Rational Software, que fue comprada por IBM. El método RUP especifica, principalmente, la constitución del equipo y las escalas de tiempo, así como un número de

<sup>34</sup> Zachman, John.(1999), El modelado de las empresas: la arquitectura Zachman, Zachman Institute for Framework Advancement. Estados Unidos.

<sup>35</sup> Jacobson, I. Booch, G. Rumbaugh, J. "El proceso unificado de desarrollo software". Editorial Addison Wesley

modelos de documento. El enfoque práctico de RUP describe las buenas prácticas de ingeniería de software que se recomiendan para su uso en desarrollo de sistemas software<sup>36</sup>. Es una plataforma de procesos de desarrollo de software que le brinda a los usuarios guías personalizadas de procesos para todo el equipo de desarrollo. RUP organiza los proyectos en términos de disciplinas y fases, cada una tiene una o varias iteraciones. Como contiene iteraciones el flujo de trabajo puede variar con el tiempo y al ser iterativo se pueden disminuir los riesgos en forma temprana y continua, con un proceso demostrable y releases<sup>37</sup> ejecutables.

Los procesos RUP consideran las tareas y el horario midiendo la velocidad de las iteraciones concerniente a sus iteraciones iniciales. Esas iteraciones tempranas se enfocan sobre la arquitectura del software.

### **Ciclo de vida RUP<sup>38</sup>**

**Inicio.** En esta fase se especifican los objetivos del ciclo de vida del proyecto y las necesidades de cada participante, se identifican los casos de uso, se hace un catálogo de requisitos y se estima el presupuesto del proyecto.

**Elaboración.** En esta fase se brinda una arquitectura sólida en cuanto a requerimientos y planes estables, refinamiento e los objetivos de la fase anterior, casos de uso, análisis, diseño, definición. Al final de la fase se realiza un análisis para determinar los riesgos y se evalúan los gastos hechos contra los originalmente planeados.

**Construcción.** En esta fase se desarrollan, integran y verifican todos los componentes de la aplicación, también se construye el sistema de información al final de esta fase deben haber versiones de prueba y se debe compilar también una versión de entrega.

**Transición.** En esta fase se corrigen los errores y se agregan los rasgos propuestos, se hace la documentación respectiva y se hace preparación del producto para su entrega, se llama transición porque se transfiere a las manos del usuario, pasando del entorno de desarrollo al de producción.

La figura 5 muestra el ciclo de vida de RUP con sus fases y sus flujos de trabajo.

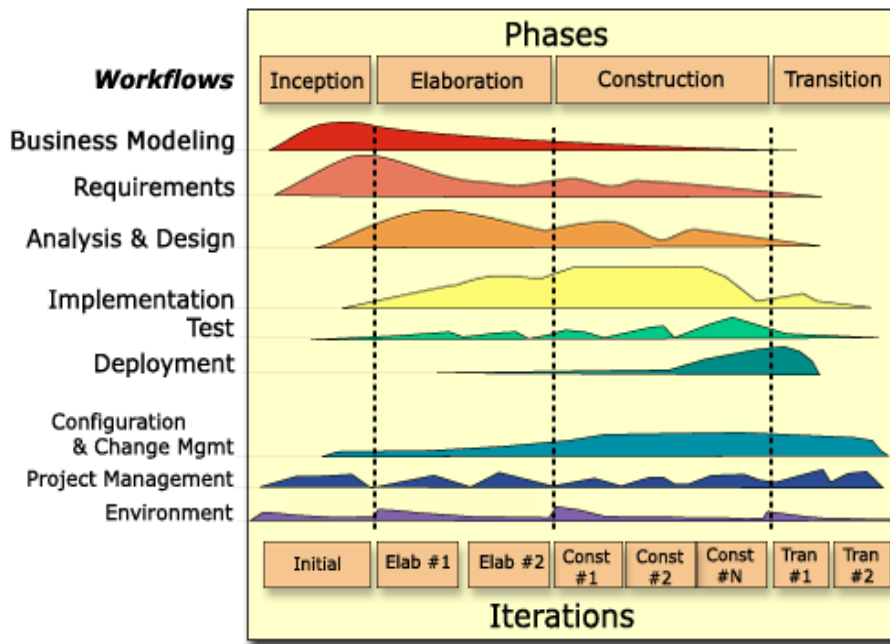
---

<sup>36</sup> Egas, Luis. Jativa, Juan. Evolución de las Metodologías de Desarrollo de la Ingeniería de Software en el Proceso la Ingeniería de Sistemas Software y Determinación de una metodología adaptable orientada a una organización pequeña. Repositorio digital ESPE. 2014

<sup>37</sup> Jojooa - tecnología, marketing y crm. (Noviembre de 2014). Definición de Release. Obtenido de <https://sites.google.com/site/jojooa/analisis-de-sistemas/definicion-de-release-que-es-un-release>

<sup>38</sup> Kroll, P. Krutchten, F. The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP. Pearson Education Inc.

**Figura 5.** Ciclo de vida RUP<sup>39</sup>



**Fuente.** RUP Summary, Imagen tomada de: <http://projects.staffs.ac.uk/suniwe/project/rup.html>

**Roles en RUP<sup>40</sup>.** Un rol define un comportamiento y una responsabilidad asignada a un individuo, las responsabilidades de un rol son llevar a cabo un conjunto de actividades y artefactos. Estos roles son divididos entre todos los que hacen parte del grupo de trabajo que definen las tareas de cada uno y los resultados que se esperan de ellos (artefactos).

A continuación se detalla cada rol, en RUP cada uno de ellos está dividido en varios tipos de sub-roles que serán definidos como se muestra:

**Analistas:**

**Analista de procesos de negocio:** descubre todos los casos de uso del negocio.

**Diseñador del negocio:** detalla un único conjunto de caso de usos del negocio.

**Analista de sistema:** descubre todos los requisitos de los casos de uso.

<sup>39</sup> Amaya, Y. (Noviembre de 2014). Metodologías ágiles en el desarrollo de aplicaciones. Obtenido de: <http://projects.staffs.ac.uk/suniwe/project/rup.html>

<sup>40</sup> IBM. Understanding RUP roles. Disponible en: <http://www.ibm.com/developerworks/rational/library/apr05/crain/>

**Especificador de requisitos:** detalla un conjunto único de los requisitos de los casos de usos.

**Desarrolladores:**

**Arquitecto de software:** decide que tecnologías usar para la arquitectura del software.

**Diseñador:** detalla el análisis y diseño de los casos de uso.

**Diseñador de interfaz de usuario:** diseña y construye la interfaz del usuario.

**Diseñador de base de datos:** diseña y construye la base de datos del sistema.

**Implementador:** es responsable de desarrollar y probar los componentes, de acuerdo a los estándares adoptados para el proyecto, y la integración de los subsistemas.

**Integrador:** combinan los elementos entregados por el implementador. Un integrador es también responsable de planear la integración, que ocurre en los niveles del subsistema y de sistema.

**Gestores:**

**Jefe de proyecto:** crea el plan para el negocio.

**Jefe de control de cambios:** se encarga de mantener al tanto a todos los miembros del equipo de todos los cambios realizados en el sistema.

**Jefe de configuración:** proporciona toda la infraestructura y el ambiente la administración de la configuración al equipo del desarrollo de producto.

Apoya la actividad del desarrollo de producto de modo que los desarrolladores y los integradores tengan espacios de trabajo apropiados para construir y para probar su trabajo, y de modo que todos los artefactos estén disponibles para la inclusión en la unidad del despliegue según lo requerido.

**Jefe de pruebas:** es el responsable de definir los métodos de prueba y asegurarse del éxito de su implementación.

**Jefe de despliegue:** supervisa la implementación de todas las unidades de despliegue.

**Ingeniero de procesos:** su función es diseñar, poner en marcha y ejecutar todo lo necesario para obtener lo mejor del sistema.

*Apoyo:*

**Documentador técnico:** mantiene la información generada durante el proceso de desarrollo con un adecuado procesamiento que permita la calidad en el mantenimiento de la misma

**Administrador de sistema:** implementa, configura, mantiene, monitoriza, documenta y asegura el correcto funcionamiento del sistema.

**Especialista en herramientas:** encargado de buscar, instalar y manejar las herramientas que necesite el sistema para desarrollarse.

**Artista gráfico:** describe gráficamente lo que se pide.

*Especialista en pruebas:*

**Especialista en Pruebas (tester):** asegura que la prueba está completa y realizada por los motivadores adecuados.

**Analista de pruebas:** selecciona las pruebas que se necesitan basado en los motivadores.

**Diseñador de pruebas:** decide que pruebas deben ser automatizadas y crea el manual de automatizaciones.

*Otros roles:*

**Stakeholders:** son los que pueden afectar o ser afectados por las actividades desarrolladas.

**Coordinación de revisiones:** mantiene una lista de tareas que se le asignan al revisor

**Revisor técnico:** es el encargado de revisar que todos los equipos tengan buen funcionamiento.

**Artefactos en RUP<sup>41</sup>.** En cada fase del ciclo de vida de un proyecto RUP se definen una serie de artefactos los cuales son:

---

<sup>41</sup> Carrillo, A. Metodología RUP De Ingeniería Del Software.

**Cuadro 9.** Artefactos RUP



**Fuente.** Adaptado de: Carrillo Anay. Metodología RUP De Ingeniería Del Software.

El desarrollo de software es una actividad que tiene mucha importancia y que en la actualidad requiere soluciones rápidas y confiables, es por eso que en respuesta a dicha necesidad se ha creado el uso de métodos que proporcionan una serie de pasos para mejorar la calidad y la productividad.

Como se describe anteriormente ante la necesidad de mejorar la calidad y productividad se crearon los métodos tradicionales cada uno de ellos poseen ciclo de vida, fases, artefactos y roles que ayudan a solucionar los requerimientos del usuario, disminución de documentación, mejora la relación con el cliente ya que está involucrado totalmente en el proyecto, promueve el trabajo en equipo con sus reuniones diarias, mejora la calidad del producto ya que no pasa a la siguiente fase hasta que no se hayan solucionado todos los errores.

A partir de los métodos tradicionales y tras su evolución se realizan algunas mejoras que dan paso a los métodos ágiles son un método basado en el desarrollo “iterativo e incremental”<sup>42</sup> y que generan respuesta en un corto lapso de tiempo, son flexibles a cambios y minimizan la documentación.

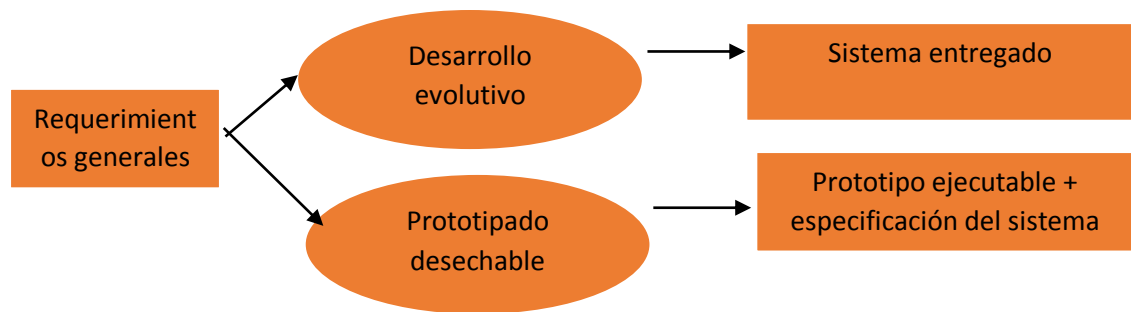
<sup>42</sup> PROYECTOSAGILES. Iterativo e incremental (s.l.) [En línea]. (s.f). [Citado el 25 noviembre de 2014.] Disponible en internet en: <http://www.proyectosagiles.org/desarrollo-iterativo-incremental>



**Métodos ágiles.** A comienzo de los años 90 surgieron nuevas propuestas para desarrollar software en tiempo corto y flexible a los cambios, estas propuestas serán conocidas con el nombre de métodos ágiles, Los métodos ágiles de desarrollo de software hoy en día son usados como técnicas para acelerar los procesos del ciclo de vida del desarrollo a través de herramientas, técnicas y artefactos que enfatizan las comunicaciones cara a cara en vez de la documentación.

En la figura 6 se encuentra un esquema de lo que es el desarrollo prototipado incremental<sup>43</sup> que es el que utilizan los métodos ágiles en ellos hay entradas y salidas.

**Figura 6.** Desarrollo y prototipado incremental <sup>44</sup>



**Fuente.** Ian Sommerville. (2005). INGENIERIA DEL SOFTWARE, séptima edición. Madrid: Pearson educación, S.A.

En febrero de 2001, tras una reunión celebrada en Utah-EEUU, nace el término “ágil” aplicado al desarrollo de software. Se pretendía ofrecer una alternativa a los procesos de desarrollo de software tradicionales, caracterizados por ser rígidos y dirigidos por la documentación que se genera en cada una de las actividades desarrolladas.

A partir de esta reunión nace el manifiesto ágil, un documento que resume la filosofía ágil:

**Manifiesto ágil.** El manifiesto ágil surgió en una reunión celebrada en febrero de 2001, en Utah-EEUU, donde participaron 17 expertos de la industria del desarrollo de software, esta reunión tenía como objetivo diseñar los valores y principios que debían permitir a los equipos de desarrollo de software realizarlo de manera más ágil y respondiendo de manera rápida a los cambios que se pudieran tener.

Tras esta reunión se creó *The Agile Alliance*<sup>45</sup>, una organización, sin ánimo de lucro, dedicada a promover los conceptos relacionados con el desarrollo ágil de software y ayudar

<sup>43</sup>SIDAR. (Enero de 2015). Prototipado. Obtenido de <http://www.sidar.org/recur/desdi/traduc/es/visitable/tecnicas/Prototyping.htm>

<sup>44</sup> Ian Sommerville. (2005). INGENIERIA DEL SOFTWARE, séptima edición. Madrid: Pearson educación, S.A.

<sup>45</sup> Agile alliance. (Marzo de 2015). Agile alliance. Obtenido de [www.agilealliance.com](http://www.agilealliance.com)

a las organizaciones para que adopten dichos conceptos. El punto de partida es fue el Manifiesto Ágil, un documento que resume la filosofía "ágil".

En dicha reunión se afirmó que el desarrollo de software debía regirse por una serie de valores y principios que se verán a continuación:

### **Valores del Manifiesto ágil<sup>46</sup>**

A los individuos y su interacción, por encima de los procesos y las herramientas.

El software que funciona, por encima de la documentación exhaustiva.

La colaboración con el cliente, por encima de la negociación contractual.

La respuesta al cambio, por encima del seguimiento de un plan.

### **Principios del Manifiesto Ágil**

Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.

Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.

Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.

Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.

Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.

El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.

El software funcionando es la medida principal de progreso.

Los procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.

La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad.

La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.

Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.

A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.

Según el libro INGENIERIA DEL SOFTWARE, séptima edición de Ian Sommerville los métodos ágiles poseen una serie de principios que se muestran a continuación en el cuadro 10 a la izquierda se encuentra cada principio en orden y a la derecha una breve descripción de cada uno.

---

<sup>46</sup> Cunningham, W. (Febrero de 2015). Manifiesto por el Desarrollo Ágil de Software. Obtenido de <http://agilemanifesto.org/iso/es/manifesto.html>

**Cuadro 10.** Los principios de los métodos ágiles

Principio	Descripción
<b>Participación del cliente</b>	Los clientes deben estar implicados fuertemente en el proceso de desarrollo. Su papel principal es proporcionar y priorizar nuevos requerimientos del sistema y evaluar las iteraciones del sistema.
<b>Entrega incremental</b>	El software se desarrolla en incrementos, donde el cliente especifica los requerimientos a incluir en cada incremento.
<b>Personas, no procesos</b>	Se deben reconocer y explotar las habilidades del equipo de desarrollo se les debe dejar desarrollar sus propias formas de trabajar, sin procesos formales, a los miembros del equipo.
<b>Aceptar el cambio</b>	Se debe contar con que los requerimientos del sistema cambian por lo que el sistema se diseña para dar cabida a estos cambios.
<b>Mantener la simplicidad</b>	Se deben centrar en la simplicidad tanto en el software a desarrollar como en el proceso del desarrollo. Donde sea posible se trabaja activamente para eliminar la complejidad del sistema.

**Fuente.** Adaptado de: Somerville, I. Ingeniería del software séptima edición. Pearson educación S.A. México, 2005.

Entre los métodos tradicionales y ágiles existen grandes diferencias ya que un método tradicional es muy poco flexible y plano estas consistían en un sin número de documentación y procesos largos, en respuesta a esto surgen los métodos ágiles que simplifican la práctica y aseguran la calidad del producto.

A continuación se muestran algunas de las diferencias que existen entre los métodos pesados y los métodos ágiles (cuadro 11), en donde la columna izquierda habla de las características de un método ágil y en la columna derecha las de un método tradicional, estas diferencias no se refieren sólo al proceso en sí, sino también al contexto de equipo y organización que es más favorable a cada uno de estas filosofías de procesos de desarrollo de software.<sup>47</sup>

---

<sup>47</sup> Letelier P, (2006), Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP), Técnica Administrativa, Buenos Aires, 5, recuperado de: [http://www.cyta.com.ar/ta0502/b\\_v5n2a1.htm](http://www.cyta.com.ar/ta0502/b_v5n2a1.htm)

**Cuadro 11.** Diferencias entre métodos ágiles y tradicionales. <sup>48</sup>

Método ágil	Método tradicional
Pocos artefactos. El modelado es prescindible, modelos desechables.	Más artefactos. El modelado es esencial, mantenimiento de modelos.
Pocos roles, mas genéricos y flexibles	Mas roles, más específicos
No existe un contrato tradicional, debe ser bastante flexible	Existe un contrato prefijado
Cliente es parte del equipo de desarrollo (además in-situ)	El cliente interactúa con el equipo de desarrollo mediante reuniones
Orientada a proyectos pequeños. Corta duración ( o entregas frecuentes ) equipos pequeños (<10 integrantes) y trabajando en el mismo sitio	Aplicables a proyectos de cualquier tamaño, pero suelen ser especialmente efectivas usadas en grandes y con equipos posiblemente dispersos
La arquitectura se va definiendo y mejorando a lo largo del proyecto	Se promueve que la arquitectura se defina tempranamente en el proyecto
Énfasis en los aspectos humanos: el individuo y el trabajo en equipo	Énfasis en la definición del proceso: roles, actividades y artefactos
Basadas en heurísticas provenientes de prácticas de producción del código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Se esperan cambios durante el proyecto	Se espera que no ocurran cambios de gran impacto durante el proyecto

**Fuente.** Letelier P, (2006), Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP), Técnica Administrativa, Buenos Aires, 5, recuperado de: [http://www.cyta.com.ar/ta0502/b\\_v5n2a1.htm](http://www.cyta.com.ar/ta0502/b_v5n2a1.htm)

También se dice que un método de desarrollo de software funciona mejor cuando se aplica a situaciones con características muy específicas, a esta clase de situaciones las llama "home ground" (bases) del método de desarrollo de software. En el cuadro 12 que posee tres columnas en las cuales se definen respectivamente de izquierda a derecha el área de desarrollo y la diferencias que existen desde el punto de vista ágil y por último el tradicional. se puede observar la comparación entre las bases de los métodos ágiles y las de los procesos de desarrollo por planes o "planeados" (plan-driven).<sup>49</sup>

<sup>48</sup> Letelier P, (2006), Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP), Técnica Administrativa, Buenos Aires, 5, recuperado de: [http://www.cyta.com.ar/ta0502/b\\_v5n2a1.htm](http://www.cyta.com.ar/ta0502/b_v5n2a1.htm)

<sup>49</sup> Blanco P, Camarero J, Fumero A, Werleski A, Rodriguez P, (2009); Metodología de desarrollo ágil para sistemas móviles Introducción al desarrollo con Android y el iPhone, recuperado de: [http://www.adamwesterski.com/wp-content/files/docsCursos/Agile\\_doc\\_TemasAnv.pdf](http://www.adamwesterski.com/wp-content/files/docsCursos/Agile_doc_TemasAnv.pdf)

**Cuadro 12.** Comparación entre las bases de los métodos ágiles y las de los procesos de desarrollo por planes<sup>50</sup>

Área	Método ágil	Métodos clásicos
<b>Desarrolladores</b>	Colaborativos, unidos , ágiles y entendidos	Orientados al plan con una mezcla de habilidades
<b>Clientes</b>	Son representativos y se les entrega poder	Mezcla niveles de actitud
<b>Requerimientos</b>	En gran parte emergentes y con rápidos cambios	Conocibles tempranamente y bastante estables
<b>Arquitectura</b>	Diseñada para los requerimientos actuales	Diseñada para los requerimientos actuales y los del futuro próximo
<b>Refactorización</b>	Económica	Costosa
<b>Tamaño</b>	Productos y equipo pequeños	Productos y equipos más grandes
<b>Premium</b>	Valor rápido	Alta seguridad

**Fuente.** Blanco P, Camarero J, Fumero A, Werleski A, Rodriguez P, (2009); Metodología de desarrollo ágil para sistemas móviles Introducción al desarrollo con Android y el iPhone, recuperado de: [http://www.adamwesterski.com/wp-content/files/docsCursos/Agile\\_doc\\_TemasAnv.pdf](http://www.adamwesterski.com/wp-content/files/docsCursos/Agile_doc_TemasAnv.pdf)

Se pudo ver y reconocer las claras diferencias que existen entre los métodos ágiles y los métodos tradicionales, se detalló el concepto de lo que es un método ágil, se identificaron sus valores y el manifiesto ágil que son un conjunto de principios que rigen estos métodos. Para mantener un orden en el desarrollo de los capítulos serán enumerados de la siguiente manera:

Definición y características de los métodos ágiles XP, SCRUM y OPENUP

Comparación entre fases, roles y artefactos de XP, SCRUM Y OPENUP

Propuesta de un nuevo método ágil a partir de las características propias de XP, SCRUM y OPENUP, integrando los puntos en común que existan entre ellas y adicionando características que lo complementen para definir la estructura que tendrá el nuevo método.

Evaluación del método ágil utilizando la herramienta de comparación y análisis 4-DAT.

#### **4.1 DEFINICIÓN Y CARACTERÍSTICAS DE LOS MÉTODOS ÁGILES XP, SCRUM Y OPENUP**

A continuación se definirán los métodos ágiles utilizados en este proyecto, se conocerá a fondo su concepto, ciclo de vida, artefactos, roles y particularidades de XP, SCRUM Y OPENUP. Se decidió hacer énfasis en ciclo de vida, artefactos y roles ya que son la base

<sup>50</sup> Blanco P, Camarero J, Fumero A, Werleski A, Rodriguez P, (2009); Metodología de desarrollo ágil para sistemas móviles Introducción al desarrollo con Android y el iPhone, recuperado de: [http://www.adamwesterski.com/wp-content/files/docsCursos/Agile\\_doc\\_TemasAnv.pdf](http://www.adamwesterski.com/wp-content/files/docsCursos/Agile_doc_TemasAnv.pdf)

fundamental de un método ágil de igual manera se hablara de las características particulares que posee cada uno para tener un concepto completo.

Se empezara hablando del método ágil:

**4.1.1 Extreme Programming.** La programación extrema, extreme programming o también conocida como XP, surge en la época de los 90, creada por Kent Beck y considerada como ágil, por sus aportes al manifiesto.

XP como cualquier método ágil busca reducir el uso exhaustivo de documentación por lo cual orienta su trabajo directamente al objetivo, y se basa en las relaciones cara a cara y la velocidad de reacción ante los cambios.

Este método fomenta el trabajo en equipo ya que la programación se ejecuta en parejas uno que escribe el código y otro que lo evalúa, además el cliente forma parte del equipo lo que hace que se corrijan errores de manera más rápida. Por esta razón Forma parte de los métodos ágiles que centran sus prioridades en las personas y no en los procesos.<sup>51</sup>

Kent Beck quien fue el padre de la programación extrema afirmo en una de sus libros que a pesar de que sabíamos que todo en proyecto de desarrollo de software cambiaba el problema era la incapacidad que teníamos para adaptarnos a los cambios constantes.

**«Todo en el software cambia. Los requisitos cambian. El diseño cambia. El negocio cambia. La tecnología cambia. El equipo cambia. Los miembros del equipo cambian. El problema no es el cambio en sí mismo, puesto que sabemos que el cambio va a suceder; el problema es la incapacidad de adaptarnos a dicho cambio cuando éste tiene lugar.» Kent Beck.**<sup>52</sup>

#### Ventajas

Desarrollo iterativo e incremental.  
Corrección de errores antes de añadir una nueva funcionalidad.  
Programación organizada.  
Menos errores al final del proyecto.

#### Desventajas

Es más recomendable en proyectos cortos.  
Altos costos en caso de fallar.

---

<sup>51</sup> Anaya, A. (2010). A propósito de programación extrema XP (eXtreme Programming). 2010, de UNIVERSIDAD CARLOS III DE MADRID Sitio web:<https://www.google.com.co/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&cad=rja&uact=8&ved=0C CoQFjAC&url=http%3A%2F%2Fzew.googlecode.com%2Fsvn%2Ftrunk%2Fdoc%2Fformacion%2Fprogramacionextrema.doc&ei=H13eVKGAMsezggS9r4PwAg&usg=AFQjCNEy6WmAiT8knGyo7XpKVI98IPvCTg&sig2=yGly78iJfqi-y8QVN3Enew&bvm=bv.85970519,d.eXY>

<sup>52</sup> Kent, B.. (1999). (Extreme Programming Explained: Embrace Change). EEUU: Addison-Wesley.

## **Roles XP<sup>53</sup>**

**Cliente.** El cliente es el responsable de dirigir el proyecto, es quien define su alcance y objetivos. Escribe las historias de usuario y las pruebas funcionales para validar su Implementación. Además, asigna la prioridad a las historias de usuario y decide cuáles se Implementan en cada iteración centrándose en aportar mayor valor al negocio.

**Programador.** El programador escribe las pruebas unitarias y produce el código del sistema. Son los encargados de transformar las historias de usuario en código.

**Encargado de pruebas (Tester).** Ayuda al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las Herramientas de soporte para pruebas.

**Encargado de seguimiento (Tracker).** Proporciona realimentación al equipo. Verifica el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, para mejorar futuras estimaciones. Realiza el seguimiento del progreso de cada iteración. Xp posee una cualidad importante que es la velocidad del equipo, que se define como el tiempo ideal estimado para las tareas frente al tiempo real dedicado, para esto el encargado pregunta uno por uno a los desarrolladores cuantas tareas ha culminado, esto ayuda a controlar el flujo del proyecto.

**Entrenador (Coach).** Es responsable del proceso global. Debe proveer guías al equipo de forma que se apliquen las prácticas XP y se siga el proceso correctamente. El objetivo de un entrenador es que el equipo comprenda las directrices de XP. No se trata de que sean solamente lecciones teóricas, si no que se trata de dar ejemplo y propone ideas para mejorar.

**Consultor.** Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto, en el que puedan surgir problemas.

**Gestor (Big boss).** Es el vínculo entre clientes y programadores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas. Su labor esencial es de coordinación.

## **Artefactos XP:<sup>54</sup>**

**Historias de usuario.** Las historias de usuario son escritas por el cliente, por lo cual no emplean un lenguaje técnico en ellas, sustituyen a los casos de usos, las historias de usuario deben proporcionar en detalle lo que el sistema debe realizar, deben llevar detalles mínimos

---

<sup>53</sup> Canós, J. Letelier, P. Penadés C. Metodologías Ágiles en el Desarrollo de Software. DSIC -Universidad Politécnica de Valencia

<sup>54</sup> Extreme Programming. (Febrero de 2015). The Rules of Extreme Programming. Obtenido de <http://www.extremeprogramming.org/rules.html>

para que así los programadores puedan realizar una estimación poco riesgosa del tiempo que les llevara desarrollarla. Cuando llegue el momento de la implementación, los desarrolladores dialogarán directamente con el cliente para obtener todos los detalles necesarios.

Las historias de usuarios deben poder ser programadas en un tiempo entre una y tres semanas. Si la estimación es superior a tres semanas, debe ser dividida en dos o más historias. Si es menos de una semana, se debe combinar con otra historia. A continuación en el cuadro 13 podemos ver el formato de las historias de usuario que utiliza XP.

**Numero:** número de formato.

**Nombre de historia de usuario:** es el requisito que se establece con el usuario.

**Usuario:** nombre del cliente.

**Iteración asignada:** número de la iteración a la que corresponde.

**Prioridad en negocio:** se establece este requerimiento qué lugar ocupa en el proyecto y se le da una prioridad con respecto a esto.

**Puntos estimados:** puntos asignados según la prioridad.

**Riesgo en desarrollo:** este riesgo es puesto por los miembros del equipo dependiendo de la prioridad que tenga.

**Descripción:** se describe el requerimiento detalladamente.

**Observaciones:** aclaraciones que se hacen para no confundir al cliente o al equipo de desarrollo.

**Cuadro 13.** Formato para historias de usuario

Historia de Usuario	
Numero:	Nombre de historia de usuario:
Modificación (o extensión) de Historia de Usuario (Nro y Nombre):	
Usuario:	Iteración Asignada:
Prioridad en Negocio: (Alta/ Media/ Baja)	Puntos Estimados:
Riesgo en Desarrollo: (Alto/ Medio/ Bajo)	Puntos Reales:
Descripción:	
Observaciones:	

**Fuente.** Autor del proyecto

Las Historias de Usuario tienen tres aspectos:

**Tarjeta.** En ella se almacena suficiente información para identificar y detallar la historia.

**Conversación.** Cliente y programadores discuten la historia para ampliar los detalles (verbalmente cuando sea posible, pero documentada cuando se requiera confirmación)



**Pruebas de Aceptación.** Permite confirmar que la historia ha sido implementada correctamente.

Los cuadros 14 y 15 se ilustran los formatos correspondientes a estos aspectos:

**Cuadro 14.** Formato caso prueba de aceptación

Caso de Prueba de Aceptación	
Código:	Historia de Usuario (Nro y Nombre):
Modificación (o extensión) de	
Descripción:	
Condiciones de Ejecución:	
Entrada / pasos de ejecución:	
Resultado Esperado:	
Evaluación de la Prueba:	

**Fuente.** Autor del proyecto

**Cuadro 15.** Task card (Tareas de ingeniería)

Tarea de Ingeniería	
Numero Tarea:	Historia de Usuario (Nro y Nombre):
Nombre de Tarea:	
Tipo de Tarea:	Puntos Estimados:
Desarrollo / Corrección / Mejora / Otra (especificar)	
Fecha de inicio:	Fecha Fin:
Programador Responsable:	
Descripción:	

**Fuente.** Autor del proyecto

**TARJETAS CRC (clase-responsabilidad-colaborador).** Estas tarjetas se dividen en tres partes: el nombre de la clase, sus responsabilidades y colaboradores.

**Cuadro 16.** Formato tarjeta CRC

<b>Nombre de la clase</b>	
Responsabilidades	Colaboradores

**Fuente.** Autor del proyecto

**Nombre de la clase:** crea un vocabulario para discutir un diseño. Debemos encontrar un conjunto nombres consistente y evocativo que describa a nuestros objetos en el contexto de nuestra aplicación.

**Las responsabilidades:** identifican tareas que se deben resolver. Se expresan mediante un puñado de frases simples, cada una con un verbo activo.

**Colaboradores:** "no object is an island" Todos los objetos establecen relaciones con otros. Llamamos objetos colaboradores a aquellos que recibirán mensajes o enviarán mensajes al objeto actual, con objeto de satisfacer una responsabilidad.

Una clase es cualquier persona, cosa, evento, concepto, pantalla o reporte. Las responsabilidades de una clase son las cosas que conoce y las que realizan, sus atributos y métodos. Los colaboradores de una clase son las demás clases con las que trabaja en conjunto para llevar a cabo sus responsabilidades.

En la práctica conviene tener pequeñas tarjetas de cartón, que se llenarán y que son mostradas al cliente, de manera que se pueda llegar a un acuerdo sobre la validez de las abstracciones propuestas.

Los pasos a seguir para llenar las tarjetas son los siguientes:

- Encontrar clases
- Encontrar responsabilidades
- Definir colaboradores
- Disponer las tarjetas

Para encontrar las clases debemos pensar qué cosas interactúan con el sistema (en nuestro caso el usuario), y qué cosas son parte del sistema, así como las pantallas útiles a la aplicación (un despliegue de datos, una entrada de parámetros y una pantalla general, entre otros).

Una vez que las clases principales han sido encontradas se procede a buscar los atributos y las responsabilidades, para esto se puede formular la pregunta ¿Qué sabe la clase? y ¿Qué hace la clase? Finalmente se buscan los colaboradores dentro de la lista de clases que se tenga.

### **Ciclo de vida XP:<sup>55</sup>**

**Exploración.** En esta fase los clientes plantean aun no tan detalladamente las historias de usuario, y de igual manera el grupo de trabajo se va familiarizando con las herramientas que va a utilizar. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo.

---

<sup>55</sup> Extreme Programming. (Febrero de 2015). Extreme Programming A gentle introduction. Obtenido de <http://www.extremeprogramming.org/rules.html>

La fase de exploración dura pocas semanas o pocos meses dependiendo del alcance del proyecto.

**Planificación de la entrega (reléase).** En esta fase se establece la prioridad que tiene cada historia de usuario, a partir de esto los programadores deciden cuanto es el tiempo estimado para la realización de esta y realizan un cronograma en conjunto con el cliente y los resultados de su primera entrega, el tiempo no debe exceder los dos meses.

**Iteraciones.** En esta fase se incluyen varias iteraciones antes de la primera entrega. La programación que se determinó en la fase de planificación se divide en varias iteraciones, en esta fase la primera iteración crea la arquitectura del sistema, esto se hace seleccionando las historias de usuario que cumplan con estos requisitos para la arquitectura del sistema.

El cliente es el encargado de escoger las historias para cada iteración, al final de la última iteración el sistema estará listo para ser entregado y seguir a la fase de producción.

**Producción.** En esta fase se realizan pruebas adicionales y revisiones del rendimiento del sistema antes de ser mostrado al cliente.

En esta fase pueden encontrarse nuevos cambios por lo cual se debe decidir si deben ser incluido o no, el tiempo de cada iteración pueden ser reducido de tres a una semana.

Después que el sistema es entregado al cliente se continúa con la siguiente iteración pero se mantiene en ejecución el sistema.

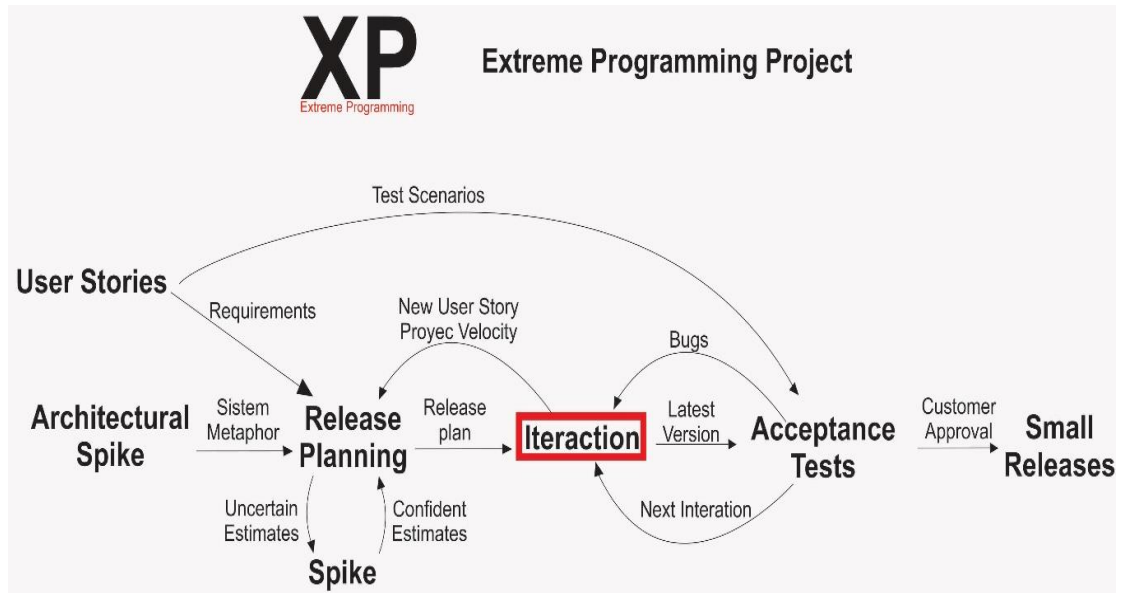
**Mantenimiento.** Mientras la primera entrega se encuentra en producción, XP debe mantener el sistema funcionando al mismo tiempo que se van desarrollando las nuevas iteraciones, para realizar esto el cliente debe ir realizando tareas de soporte que se le asignan, en esta fase se puede requerir cambio de personal y cambio en la estructura.

**Muerte del proyecto.** En esta fase el cliente no tiene más historias de usuario para incluir, esto hace que se deban satisfacer otras de sus necesidades como son rendimiento del sistema y confiabilidad del mismo. En esta fase se genera la documentación y no se realizan más cambios en la arquitectura.

La muerte del sistema también puede ocurrir cuando no satisface lo que el cliente esperaba o cuando no hay más presupuesto para mantenerlo.

La figura 7 muestra el ciclo de vida de XP:

**Figura 7.** Ciclos en XP.<sup>56</sup>



**Fuente.** Extreme Programming: A gentle introduction disponible en: <http://www.extremeprogramming.org/>

### **Particularidades:**

**Valores XP.** Xp posee una serie de valores propios de cualquier método ágil, los cuales son:

**La comunicación.** En XP la comunicación es uno de los valores más importantes la interacción es tan fuerte que el cliente es considerado parte del equipo. Existe un ambiente colaboración entre los miembros del equipo.

**La simplicidad.** Este valor hace parte de todo el ciclo de XP, va desde lo más sencillo que es la funcionalidad como la pide el cliente, hasta la simplificación del código mediante la refactorización del mismo, este método no realiza actividades complejas solo efectúa lo que el cliente le pide de la forma más sencilla.

**La retroalimentación.** Se presenta en todo en todas las etapas del proyecto y se maneja en ambos sentidos, tanto del equipo de trabajo al cliente para brindarle información de cómo va el sistema, como del cliente al equipo para darle aportes para la construcción del proyecto.

<sup>56</sup> Extreme Programming. (Febrero de 2015). Extreme Programming A gentle introduction. Obtenido de <http://www.extremeprogramming.org/rules.html>

**El coraje.** El equipo debe estar preparado para cualquier inconveniente que se presente a lo largo del proyecto, así como debe tener fuerza para exponer cualquier problema que se presente.

**Prácticas XP** <sup>57</sup>. A partir de los valores se definen una serie de prácticas, son doce reglas que se plantean las cuales son muy simples y prácticas y sirven como guía para el equipo de trabajo, a continuación se detalla cada una de ellas:

**El desarrollo está dirigido por pruebas.** Antes de realizar cualquier parte del código es necesario que esta tenga su respectiva parte de pruebas. El programador realiza pruebas a los módulos del sistema para determinar su comportamiento mientras que el cliente con ayuda del tester se encarga de diseñar las pruebas de aceptación que sirven para confirmar que las historias de usuario se hayan implementado correctamente.

**El juego de la planificación.** Desde que comienza el proyecto tanto el equipo de trabajo como el cliente debe tener claro lo que se hará, en el transcurso del proyecto se realizan reuniones que sirven para organizar tareas ideas que provengan por parte de ambos.

**Cliente In-situ.** El cliente o un representante del mismo deben permanecer en el sitio donde se desarrolla el proyecto para que aclare cualquier duda que tenga el programador.

**Programación en parejas.** XP propone que exista un par de programadores por equipo, para que haya menos errores en la codificación y se aumente su calidad. Mientras uno desarrolla el código el otro lo ejecuta.

**Entregas pequeñas.** En XP se hacen entregas constantes y completas de cada módulo funcionando bien, de tal manera que el cliente siempre tendrá parte de la aplicación funcionando.

**Refactorización.** El código se realiza de forma permanente y constante para depurarlo y simplificarlo, buscando la forma para mejorarlo. Se realiza durante todo el desarrollo del proyecto.

**Integración continua del código.** El código de los módulos debe ser integrado a cortos plazos de tiempo (No más de un día). Esto facilita que se corrijan los errores de codificación e integración que se presenten.

**Diseño simple.** Solo se realiza lo necesario para que el diseño cumpla con los requerimientos que pidió el cliente.

**Utilización de metáforas del sistema.** Para que el equipo entienda mejor los términos del sistema se usan metáforas como una forma de universalizar el lenguaje del sistema.

---

<sup>57</sup> Canós, J. Letelier, P. Penadés C. Metodologías Ágiles en el Desarrollo de Software. DSIC -Universidad Politécnica de Valencia

**Propiedad colectiva del código.** El código no es manejado ni conocido por una sola persona, sino por cualquier miembro del equipo esto permite que cualquiera de ellos realice mejoras necesarias.

**Convenciones del código.** La aplicación de estándares de programación al código fuente de la aplicación permite que cualquier persona pueda entender y realizar modificaciones al código del sistema.

**No trabajar horas extras.** Con esta práctica se pretende utilizar al máximo el rendimiento y energía del programador.

### **Administración:**

**Reuniones diarias de seguimiento.** Los grupos de trabajo no superan 20 personas, en XP se realizan reuniones diarias de seguimiento el objetivo de tener reuniones diarias es mantener la comunicación entre el equipo, y compartir problemas y soluciones. En la mayoría de estas reuniones, gran parte de los participantes simplemente escuchan, sin tener mucho que aportar.

Para no quitar tiempo innecesario del equipo, se sugiere realizar estas reuniones en círculo y de pie. Solo se trabaja 40 horas por semana.

La Programación Extrema es una metodología de desarrollo de software que se basa en la simplicidad, la comunicación y la retroalimentación o reutilización del código desarrollado (reciclado de código). XP es uno de los métodos ágiles más usados posee una serie de características que hacen que quien lo usa desarrolle sus proyectos de manera ágil y con gran calidad, es muy específico en cada uno de los pasos que sigue como se vio anteriormente.

A continuación se definirá el segundo método utilizado en este proyecto SCRUM, sus fases, roles, artefactos y particularidades.

**4.1.2 Scrum.** SCRUM es un método ágil que puede ser aplicado a casi cualquier proyecto; Sin embargo, este método es más comúnmente utilizado en el desarrollo de software. Su proceso es adecuado para los proyectos que cambian rápidamente de requisitos. SCRUM progresa a través de una serie de iteraciones llamadas sprints, que duran de uno a cuatro semanas.

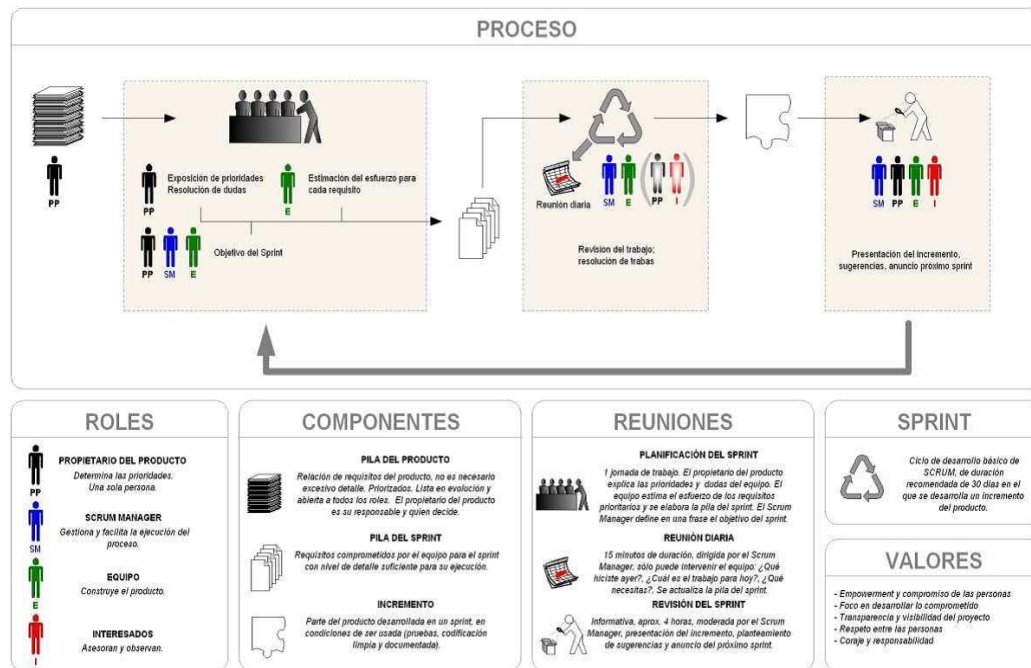
El modelo SCRUM sugiere cada sprint comienza con una breve reunión de planificación y concluye con una revisión<sup>58</sup>

---

<sup>58</sup> MOUNTAINGOATSOFTWARE. SCRUM (s.l.) [En línea]. Mar. 2010. [Citado el 25 noviembre de 2014.] Disponible en internet en: <http://www.mountaingoatsoftware.com/agile/scrum>

A continuación la figura 8 muestra una vista general de todo el proceso SCRUM, sus roles, componentes, reuniones, valores y los sprints. Todo esto será explicado más adelante.

**Figura 8.** Vista general de SCRUM<sup>59</sup>



Fuente. Agile development. Scrum. Disponible en: <https://santimacnet.wordpress.com/page/25/>

### Roles SCRUM:<sup>60</sup>

**Scrum Master.** Es un rol de administración que debe asegurar que el proyecto se está llevando a cabo de acuerdo con las prácticas, valores y reglas de SCRUM y que todo funciona según lo planeado También interactúa con el cliente y los gestores.

El Scrum master ayuda al dueño del producto de diferentes maneras tales como:

- Encontrar técnicas para crear y modificar el product backlog de la mejor manera.
- Ayudar al equipo Scrum a entender porque es necesario contar con elementos claros y precisos en el product backlog
- Entender la planificación del producto de manera empírica
- Asegurar que el product owner sepa como priorizar los elementos del product backlog para maximizar el valor.

El Scrum master ayuda al equipo de desarrollo de la siguiente manera:

<sup>59</sup> Agile development. Scrum. Disponible en: <https://santimacnet.wordpress.com/page/25/>

<sup>60</sup> Schwaber, K. Sutherland, J. La Guía de Scrum. Scrum.Org and ScrumInc. 2014

Guía al equipo para que sea organizado y multifuncional  
Ayuda al equipo a crear productos con alto valor  
Elimina elementos que impidan el avance del proyecto  
Guía al equipo de desarrollo en las prácticas que no han sido adoptadas o entendidas por completo

El Scrum master ayuda a la organización de la siguiente manera:

Liderar y guiar a la organización en la adopción de SCRUM  
Motivar los cambios que incrementan la productividad en el equipo  
Ayudar a los empleados y a otros a entender y llevar a cabo SCRUM.

**Product Owner.** Es el responsable del proyecto, administra, controla y comunica la Backlog List. Es el responsable de encontrar la visión del producto y reflejarla en la Backlog List. Generalmente esta persona puede ser el Product Manager, Marketing, Internal Customer, etc. Se encarga de escribir las ideas del cliente y las ordena con prioridad. . El product owner debe reducir todos los riesgos del proyecto y debe lograr que cada miembro del equipo asista a las reuniones diarias que se realizan. El dueño del producto es el único responsable de gestionar el product backlog la gestión de esta lista incluye:

Expresar claramente los elementos de la lista  
Ordenar por prioridad los elementos de la lista para alcanzar los objetivos de la mejor manera posible.  
Optimizar el valor del trabajo desempeñado por el equipo de desarrollo  
Asegurarse que la lista sea clara para que pueda ser entendida por todos  
El dueño del producto es una única persona y para que pueda realizar bien su trabajo, toda la organización debe respetar sus decisiones. Las decisiones del dueño del producto se ven reflejadas en el contenido y en la priorización de la lista del producto.

**Scrum team.** Es el equipo del proyecto que tiene la autoridad para decidir cómo organizarse para cumplir con los objetivos de un Sprint. Sus tareas son: Effort Estimation (Estimar Esfuerzo), crear el Sprint Backlog, revisar la Product Backlog List y sugerir obstáculos que deban ser removidos para cumplir con los items que aparecen. Típicamente es un equipo de entre 5 y 10 personas cada una especializada en algún elemento que conforma los objetivos a cumplir, por ejemplo: Programadores, Diseñadores de Interfaz de usuario, etc. La dedicación de los miembros del equipo debería ser full-time con algunas excepciones. La membresía solo puede cambiar entre sprints (no durante). Los equipos de desarrollo tienen las siguientes características:

Son multifuncionales y cuentan con las habilidades necesarias para crear un incremento del producto.

SCRUM no reconoce títulos para los miembros del equipo de desarrollo todos son desarrolladores independientemente del trabajo que realicen.



No se reconocen sub-equipos en los equipos de desarrollo, no importan los dominios particulares que requieran ser tenidos en cuenta como lo son pruebas o análisis de negocio. Los miembros del equipo de desarrollo pueden tener habilidades especializadas en áreas específicas pero la responsabilidad recae sobre todo el equipo, es un todo.

**Customer.** Es el cliente y quien participa en las tareas involucradas en el product backlog.

**Management.** Es el responsable de tomar las decisiones finales, acerca de estándares y convenciones a seguir durante el proyecto. Participa en la selección de objetivos y requerimientos y en la selección del Scrum Owner. Tiene la responsabilidad de controlar el progreso y trabaja junto con el Scrum Master en la reducción de la Product Backlog List.

**Artefactos SCRUM**<sup>61</sup>. El artefacto principal en SCRUM es el producto, el objetivo de este método es tener un producto que pueda ser funcional al final de cada sprint.

**Product Backlog.** Es la lista de todo el trabajo que se desea realizar en el proyecto. Esta lista es priorizada por el Product Owner y puede ser actualizada y revisada cuando sea necesario. La mejor manera de crear un product backlog en SCRUM es con historias de usuario que son descripciones breves descritas desde la perspectiva del usuario.

Seguidamente se muestran las plantillas que utiliza SCRUM para las descripciones que hace el usuario de los requerimientos del sistema (historia de usuario) y la plantilla para el product backlog, antes de esto se explica cada campo de las plantillas para entender mejor la manera en que se utilizan.

### **Plantillas para Historias de usuario:**

**Identificador (ID) de la historia.** Código que identifica unívocamente a la historia en el Proyecto que se esté desarrollando. El formato debe ser elegido por el equipo.

**Rol.** Es el rol que está desempeñando el usuario cuando utiliza la funcionalidad que se está describiendo. Debe ser lo más específico posible, describiendo el rol o actor que se está desempeñando. El enunciado puede escribirse como se sigue: Yo como un [Rol], Desempeñando el rol de [Rol], Como un [Rol], entre otros. Por ejemplo: Yo como cliente registrado. Desempeñando el rol de cliente registrado. Como un cliente registrado.

**Característica / funcionalidad.** Representa la función que el rol quiere o necesita hacer en el sistema que se está desarrollando. Puede diferenciarse entre acciones obligatorias u opcionales, utilizando la palabra puede o necesita para describir la acción. Por ejemplo: Necesito realizar búsquedas de productos por categorías. Puedo seleccionar una categoría para ver el número de productos que tiene asociado.

---

<sup>61</sup> Alaimo, D. Proyectos ágiles con Scrum: flexibilidad, aprendizaje, innovación y colaboración en contextos complejos . - 1ª ed. - Ciudad Autónoma de Buenos Aires: Kleer, 2013. EBook.

**Razón / resultado.** Lo que el rol necesita lograr al ejecutar la acción. Este es el resultado de ejecutar la acción desde el punto de vista del rol. Este punto puede ser opcional, pues la historia puede documentarse sólo con la definición del rol y la acción (sin definir la consecuencia).

**Número (#) de escenario.** Número (ejemplo 1, 2, 3 ó 4), que identifica al escenario asociado a la historia.

**Criterio de aceptación (título).** Describe el contexto del escenario que define un comportamiento. Por ejemplo, si se toma el ejemplo de búsquedas de productos por categoría, un posible ejemplo pudiera ser: Categoría sin productos asociados.

**Contexto.** Proporciona mayor descripción sobre las condiciones que desencadenan el escenario.

**Evento.** Representa la acción que el usuario ejecuta, en el contexto definido para el escenario.

**Resultado / comportamiento esperado.** Dado el contexto y la acción ejecutada por el usuario, la consecuencia es el comportamiento del sistema en esa situación.

#### **Plantilla para el product backlog:**

**Identificador (ID) de la historia de usuario.** Código que identifica a la historia de forma unívoca, una vez asignado, no debe ser re-usado en otra historia, ni siquiera si la historia es descartada. El código identifica la historia en otros documentos, como por ejemplo la plantilla de historias de usuario.

**Enunciado de la historia.** Nombre de la historia, el cual debe ser el mismo que se utiliza en otros documentos. Se puede utilizar el formato siguiente. Como un [Rol], Necesito [Descripción de la Funcionalidad], con la finalidad de [Razón o Resultado].

**Alias.** Título de la historia alternativo a la descripción, que servirá para identificar más fácilmente la historia sin tener que repetir todo su enunciado. Se puede utilizar por ejemplo el nombre de la funcionalidad o requerimiento que se pretende desarrollar.

**Estado:** Identifica los posibles estados de la historia durante su ciclo de vida:

*Vacío.* La historia fue identificada pero aún no ha sido asignada a una iteración.

*Planificada.* La historia fue asignada a una iteración y aún no ha comenzado su ejecución. Puede tener este estado incluyendo en la iteración donde está planificado ejecutarla (pero que aún no ha comenzado).

*En Proceso.* La historia fue seleccionada por el equipo y está en proceso de desarrollo (en ejecución).

*Hecho* (Donde). La historia fue desarrollada. Es importante clarificar la definición de “Hecho” con el equipo de trabajo. “Hecho” no sólo incluye el desarrollo sino la integración y pruebas integrales del Software. Una historia hecha puede presentarse al dueño de producto para sus pruebas de aceptación.

*Descartada*. Se determinó que la historia ya no es relevante, su contenido se incluyó en otro grupo de historias o fue cancelada.

**Dimensión / esfuerzo.** Medida del esfuerzo (tamaño) que implica desarrollar la historia, existen distintos métodos para medirlo, un ejemplo es los “puntos de historia” una medida de complejidad no necesariamente relacionado con jornadas o días. Otra forma de medirlo es con días o jornadas ideales.

**Iteración (Sprint).** Iteración o Sprint al que se asigna la historia. Esta asignación puede cambiar en cada iteración donde se haga la revisión de la pila de producto (ProductBacklogReview), según las prioridades indicadas por el dueño de producto. Por medio de este campo se puede crear un “Plan de Salidas a Productivo” (Release Plan).

**Prioridad.** Siguiendo el marco de trabajo ágil y Scrum, se le deben asignar prioridades a las historias, según las instrucciones del dueño de producto (ProductOwner). De esta forma pueden ordenarse. Las historias de mayor prioridad deben ser las que agregan más valor al negocio, y deben ser originadas en sus necesidades.

**Comentarios.** Comentarios o detalles relacionados que expliquen la historia. Para definiciones de mayor longitud deben usarse documentos externos, por ejemplo la plantilla de historias de usuario.

**Cuadro 17.** Historias de usuario SCRUM<sup>62</sup>

Identificador (ID) de la Historia	Enunciados de la historia				Criterios de aceptación			
	Rol	Característica / Funcionalidad	Razón / Resultado	Número (#) de Escenario	Criterio de Aceptación (Título)	Contexto	Evento	Resultado / Comportamiento esperado

**Fuente.** La web sobre gerencia de proyectos de informática, software y tecnología, disponible en: [www.pmoinformatica.com](http://www.pmoinformatica.com)

**Cuadro 18.** Plantilla product backlog <sup>63</sup>

Identificador (ID) de la Historia	Enunciado de la historia	Alias	Estado	Dimensión / Esfuerzo	Iteración (Sprint)	Prioridad	Comentarios

**Fuente.** La web sobre gerencia de proyectos de informática, software y tecnología, disponible en: [www.pmoinformatica.com](http://www.pmoinformatica.com)

<sup>62</sup> La web sobre gerencia de proyectos de informática, software y tecnología. (Marzo 2015) disponible en: [www.pmoinformatica.com](http://www.pmoinformatica.com)

<sup>63</sup> La web sobre gerencia de proyectos de informática, software y tecnología. (Marzo 2015) disponible en: [www.pmoinformatica.com](http://www.pmoinformatica.com)

**Sprint Backlog.** El Scrum Team decide que tareas son necesarias para alcanzar el sprint goal. Estas tareas se denominan Sprint Backlog Items: subconjunto de requerimientos del Product Backlog que se implementarán durante el sprint

El Sprint Backlog puede cambiar durante el sprint:

El equipo puede agregar nuevas tareas cuando sea necesario con el objetivo de alcanzar el sprint goal,

El equipo puede eliminar tareas innecesarias, Pero: el Sprint Backlog sólo puede ser actualizado por el Scrum Team

A continuación en el cuadro 19 se muestra un ejemplo de lo que es un sprint backlog con su respectiva plantilla

**Cuadro 19.** Ejemplo de un sprint backlog<sup>64</sup>

Requisito	Tarea	Qui en	Estado (No iniciada / En progreso/ Completa da)	Día:	1	2	3	4	5	6	7	8
				Hor as:	11 20	1 0	107 6	104 8	104 0	103 2	102 0	992
				Pen dientes	8	8						
Requisito A	Tarea 1	José Laura	Completa da		16 4	8						
Requisito A	Tarea 4	Laura	Completa da		4 8							
Requisito A	Tarea 5	Ana	Completa da		16 8	8	4 8					
Requisito A	Tarea 3	José Ana	Completa da		16	1 6	16	8				
	Tarea 2	José	Completa da									
	Tarea 6		Completa da									
	Tarea 7		Completa da									
Requisito B	Tarea 8	Ana Laura	Completa da		16 8	1 6	16 8	16 8	16 8	16 8	6 8	4
Requisito B	Tarea 9	Laura	En progreso			8						
Requisito C	Tarea 10	José	No iniciada		4	4	4	4	4	4	4	4

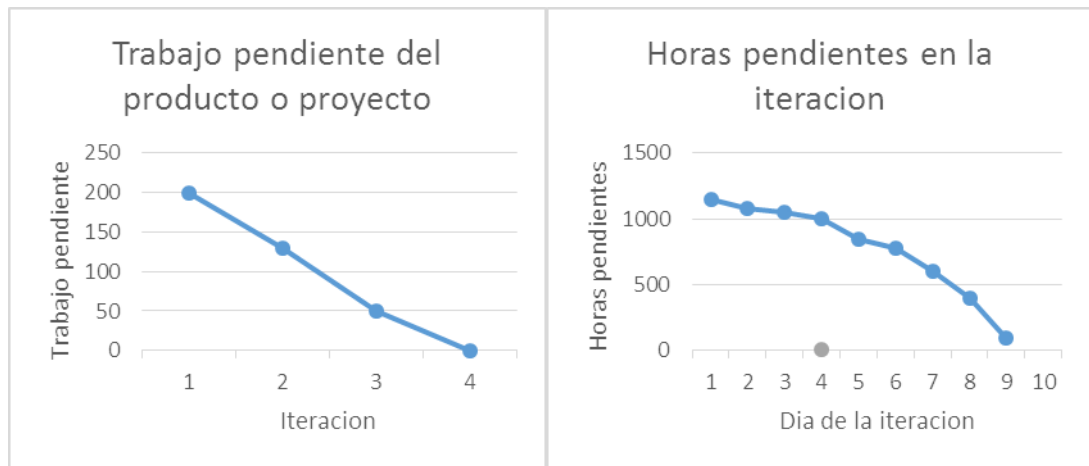
**Fuente.** Como gestionar proyectos Scrum, disponible en: <http://www.proyectosagiles.org/lista-tareas-iteracion-sprint-backlog>

<sup>64</sup> Como gestionar proyectos Scrum, (Enero 2015) disponible en: <http://www.proyectosagiles.org/lista-tareas-iteracion-sprint-backlog>

**Burndown Chart.** Es un gráfico que mide la cantidad de requisitos en el Backlog del proyecto pendientes al comienzo de cada sprint. Dibujando una línea que conecte los puntos de todos los sprints completados, se podrá ver el progreso del proyecto.

La figura 9 muestra un par de gráficos (Burndown chart) el primero a la izquierda es el grafico trabajo pendiente del producto o proyecto el cual muestra cada iteración con respecto al trabajo pendiente, el segundo a la derecha muestra las horas pendientes por cada iteración:

**Figura 9.** Ejemplo de Burndown chart<sup>65</sup>



**Fuente.** Como gestionar proyectos Scrum, disponible en: <http://www.proyectosagiles.org/graficos-trabajo-pendiente-burndown-charts>

### Ciclo de vida:<sup>66</sup>

**Planeamiento.** En esta fase se establece lo que se quiere en el proyecto. Consta de una serie de tareas las cuales son. Escritura de la visión, el presupuesto, el sprint backlog del producto inicial y los ítems estimados, también la arquitectura de alto nivel, el diseño exploratorio y prototipos. Las tareas a realizar en esta parte son.

**Definir el proyecto.** Se indica de forma clara el propósito del proyecto, no es necesario que sea tan detallado pero sí que el equipo entienda cuales son las necesidades del producto y del cliente.

**Definir “Terminado”.** Marcará el punto en el que la tarea se considera terminada.

<sup>65</sup> Como gestionar proyectos Scrum. (Enero 2015) disponible en: <http://www.proyectosagiles.org/graficos-trabajo-pendiente-burndown-charts>

<sup>66</sup> Schwaber, K. Sutherland, J. La Guía de Scrum. Scrum.Org and ScrumInc. 2014

**Definición del product backlog inicial.** Esta es una lista de funcionalidades marcada y priorizada por el product owner que al desarrollarla e implementarla cumpla las especificaciones.

**Definición de los entregables.** Una vez realizada la lista es necesario establecer criterios para hacer pequeñas entregas del producto.

**Montaje.** En esta fase se definen más requerimientos, además se le da prioridad a algunas tareas para la primera iteración. Las actividades que se manejan en esta fase son. Planificación, diseño exploratorio y prototipos.

**Desarrollo.** En esta fase se pone a prueba un sistema listo para su entrega en una serie de iteraciones de treinta días (sprints), las actividades en esta fase son. un encuentro de planeamiento 24 de sprints en cada iteración, la definición del product backlog, los estimados y los encuentros diarios.

**Liberación.** En esta fase se desarrolla operacionalmente el producto. Las actividades son documentación, entrenamiento, mercadeo y venta.

La figura 10 muestra la imagen correspondiente al ciclo de vida de SCRUM:

**Figura 10.** Ciclo de vida SCRUM



**Fuente.** Scrum alliance. (Enero de 2015). Scrum alliance. Obtenido de [www.scrumalliance.com](http://www.scrumalliance.com)

Un sprint consiste en:

**Planificación del sprint.** Se define el product backlog que es una lista con los requerimientos organizados priorizadamente y puede ser modificado constantemente. En

cada iteración (sprint) el product backlog es revisado por el equipo. Se planifica el primer sprint. En la reunión al inicio del sprint se obtiene el sprint backlog que contiene un listado de tareas y el objetivo principal del sprint.

**Seguimiento del sprint.** Se realizan reuniones diarias y se hace un seguimiento para ver si las tareas están siendo realizadas.

**Revisión del sprint.** Finalizado el sprint se evalúa para ver si cumplió con todas las tareas, se presentan resultados finales y se sugiere presentar un prototipo.

### **Particularidades:**<sup>67</sup>

**Valores SCRUM.** Todo método ágil posee una serie de valores que hace que el equipo logre un equilibrio entre sí, es por eso que SCRUM maneja los siguientes valores:

**Enfoque.** El equipo de trabajo se centra en un solo objetivo a la vez para poder entregar un buen trabajo.

**Coraje.** Se trabaja en equipo lo cual hace que se tenga más valentía para afrontar nuevos desafíos.

**Franqueza.** A medida que avanza el proyecto se expresan fortalezas y preocupaciones que se tienen para así poder lograr un mejor resultado.

**Compromiso.** Existe un control sobre las actividades lo cual asegura el éxito.

**Respeto.** A medida que se desarrollan proyectos y se comparten victorias se le otorga respeto a este método.

**Proceso.**<sup>68</sup> SCRUM posee una serie de eventos predefinidos que permiten crear puntualidad y minimizar las reuniones no definidas en SCRUM. Todos los eventos son bloques de tiempos también llamados *time-boxes* de tal modo que tiene una duración máxima. Una vez que un sprint comienza la duración es fija por lo cual no puede acortarse ni alargarse. Los demás eventos pueden terminar antes de lo establecido siempre y cuando hayan logrado su objetivo, se deben asegurar que el tiempo empleado no tenga desperdicio en el proceso.

**Sprint.** Es el núcleo de SCRUM, es un *time-boxes* de un mes o menos durante el cual se crean incrementos del producto “terminados”, utilizable y potencialmente desplegable. Cada nuevo sprint comienza inmediatamente el sprint previo es terminado. Los Sprints contienen y consisten en reunión de la planificación del sprint (Sprint Planning Meeting), los scrums diarios (Daily scrums), el trabajo de desarrollo, la revisión del sprint (Sprint review) y la retrospectiva del sprint (Sprint retrospective).

Durante el sprint:

---

<sup>67</sup> Palacio, J. Flexibilidad con Scrum. Safe creative. 2007

<sup>68</sup> Scrum. (Febrero de 2015). Formations gestion de projet Agile avec SCRUM. Obtenido de <http://www.scrumguides.org/index.html>



Durante el sprint no se pueden cambiar los miembros del equipo ni el sprint backlog, lo único que se podría hacer es cancelarlo por motivos de fuerza mayor.

Durante el sprint se realizan diariamente los scrum Daily Meetings, en ellas participan todos los miembros de equipo y sirven para analizar el avance del trabajo que se realizó durante el día. El scrum master hace de moderador de estas reuniones.

Al finalizar el sprint, se realiza una reunión llamada Sprint Review Meeting, en ella se obtiene lo que se aprendió y se deja registrado en un artefacto denominado sprint retrospective.

**Cancelación de un sprint.** Un sprint puede ser cancelado antes de que el tiempo estimado se cumpla. Solo el product owner o dueño del producto tiene derecho a cancelar el sprint, aunque puede hacerlo bajo la influencia de los interesados del equipo, el equipo de desarrollo o el scrum master.

Un sprint se puede cancelar si su objetivo queda obsoleto, en general un sprint debe cancelarse sino tiene sentido seguir con él. Cuando un sprint es cancelado se deben revisar todos los elementos del product backlog que se hayan terminado, si alguna parte del trabajo está bien hecha el product owner normalmente lo acepta, las cancelaciones de los sprints normalmente consumen recursos ya que todos deben reagruparse en otra reunión de planificación para empezar uno nuevo y esto genera pérdida de tiempo y valor.

**Reunión de planificación del sprint (Sprint Planning Meeting).** En esta reunión se planea el trabajo que va a ser realizado en el sprint, la reunión tiene una duración de ocho horas para un sprint de un mes, para sprints más cortos dura menos, en esta reunión participa todo el equipo y el scrum master es el encargado de que los asistentes entiendan el propósito. La reunión de planificación responde a este par de preguntas.

¿Qué puede entregarse en el incremento resultante del sprint que comienza?  
¿Cómo se conseguirá hacer el trabajo necesario para entregar el incremento?

**Objetivo del sprint (Sprint Goal).** Es una meta establecida para el sprint que puede ser implementada en el product backlog, da al equipo una guía de lo que se está construyendo en el incremento, el objetivo del sprint hace que el equipo trabaje en conjunto y que todos mantengan en mente el mismo. Con el fin de satisfacer el objetivo del sprint se implementa la funcionalidad y la tecnología.

**Scrum diario (Daily scrum).** Es una reunión que tiene una duración de 15 minutos y se hace diariamente con el fin de que el equipo de desarrollo sincronice sus actividades y cree un plan para las siguientes 24 horas. Esto se hace revisando el trabajo del último scrum diario, esta reunión se realiza a la misma hora y en el mismo lugar cada día para reducir la complejidad. El Equipo de Desarrollo usa el Scrum Diario para evaluar el progreso hacia el Objetivo del Sprint y para evaluar qué tendencia sigue este progreso hacia la finalización del trabajo contenido en la Lista del Sprint. El Scrum Diario optimiza las posibilidades de

que el Equipo de Desarrollo cumpla el Objetivo del Sprint. El Scrum Master se asegura de que el Equipo de Desarrollo tenga la reunión, pero el Equipo de Desarrollo es el responsable de dirigir el Scrum Diario. Durante esta reunión cada miembro del equipo explica lo siguiente:

¿Qué hice ayer que ayudó al Equipo de Desarrollo a lograr el Objetivo del Sprint?

¿Qué haré hoy para ayudar al Equipo de Desarrollo a lograr el Objetivo del Sprint?

¿Veo algún impedimento que evite que el Equipo de Desarrollo o yo logremos el Objetivo del Sprint?

Los Scrum Diarios mejoran la comunicación, eliminan la necesidad de mantener otras reuniones, identifican y eliminan impedimentos relativos al desarrollo, resaltan y promueven la toma de decisiones rápida, y mejoran el nivel de conocimiento del Equipo de Desarrollo. El Scrum Diario constituye una reunión clave de inspección y adaptación.

**Revisión del sprint (Sprint Review).** Es una reunión de una duración de 4 horas para sprints de un mes, el scrum master se encarga de que lo que se dice sea entendido por todos, esta reunión tiene como objetivo explorar el incremento realizado y si es necesario reajustar el product backlog, quienes hacen parte de esta reunión ayudan para determinar que más podría hacerse para optimizar el valor, es una reunión informal no de seguimiento y la exposición del incremento tiene como fin facilitar retroalimentación de información y fomentar la colaboración.

Esta reunión de revisión de sprint lleva los siguientes elementos:

Los asistentes son el Equipo Scrum y los interesados clave invitados por el Product owner  
El Product owner explica qué elementos de la Lista de Producto se han “Terminado” y cuales no se han “Terminado”;

El Equipo de Desarrollo habla acerca de qué fue bien durante el Sprint, qué problemas aparecieron y cómo fueron resueltos esos problemas;

El Equipo de Desarrollo demuestra el trabajo que ha “Terminado” y responde preguntas acerca del Incremento;

El Product owner habla acerca de la Lista de Producto en el estado actual. Proyecta fechas de finalización probables en el tiempo basándose en el progreso obtenido hasta la fecha (si es necesario);

El grupo completo colabora acerca de qué hacer a continuación, de modo que la Revisión del Sprint proporcione información de entrada valiosa para Reuniones de Planificación de Sprints subsiguientes.

Revisión de cómo el mercado o el uso potencial del producto podría haber cambiado lo que es de más valor para hacer a continuación; y,

Revisión de la línea de tiempo, presupuesto, capacidades potenciales y mercado para la próxima entrega prevista del producto.

**Retrospectiva de Sprint (Sprint Retrospective).** Sirve para crear un plan de mejoras para el equipo. Se realiza después de la revisión del sprint y antes de la siguiente reunión de

planificación del sprint. Su duración es de tres horas para sprints de un mes, el scrum master participa en la reunión como miembro del equipo ya que la responsabilidad recae sobre él.

El propósito de la Retrospectiva de Sprint es:

Inspeccionar cómo fue el último Sprint en cuanto a personas, relaciones, procesos y herramientas

Identificar y ordenar los elementos más importantes que salieron bien y las posibles mejoras

Crear un plan para implementar las mejoras a la forma en la que el Equipo Scrum desempeña su trabajo.

Al final de esta reunión el equipo debe haber identificado mejoras que se implementaran en el siguiente sprint.

**Administración.** Reunión diaria de 15 minutos del equipo de coordinación para integrarse. Cada dos semanas a un mes se puede ver el software real y decidir si este es el software que se desea en producción o continuar y mejorar el mismo en una próxima iteración.

De manera general se puede decir entonces que **SCRUM** es un proceso para el desarrollo ágil, que da una referencia general para la construcción de software.

El equipo SCRUM está conformado por tres roles: el product owner, scrum master y varios team members. El product owner es quien, a partir de los requisitos planteados por clientes y usuarios, establece la lista de requisitos llamada product backlog, y les asigna prioridades. SCRUM organiza el desarrollo del producto en ciclos o iteraciones llamadas Sprints. La idea principal es que un Sprint establezca objetivos funcionales al principio del mismo y que lo que se pidió esté finalizado al terminar el Sprint. Antes de comenzar un sprint el product owner evalúa y define con el resto del equipo que requisitos o ítems del product backlog habrá que desarrollar en el sprint y a partir de esto se contruye el sprint backlog, partiendo de requisitos para generar tareas. Esto quiere decir que el product backlog usa como ítems los requisitos mientras el sprint backlog usa tareas.

El proceso a seguir dentro de SCRUM corresponde a una serie de iteraciones denominadas sprints en las cuales se construye el producto de acuerdo a los requerimientos existentes en el Product Backlog. Al finalizar cada sprint, se debe poder mostrar a los interesados, el producto construido hasta al momento. Ya sea en el sprint 1 (prototipo) o en el sprint N, los involucrados en el proyecto deben poder tomar decisiones sobre el progreso del mismo luego de una real demostración por parte de quienes lo construyen. Por lo anterior, los requerimientos listados en el Product Backlog serán dinámicos ya que pueden cambiar y/o tener más o menos prioridad cada vez que se incrementa la funcionalidad del producto.

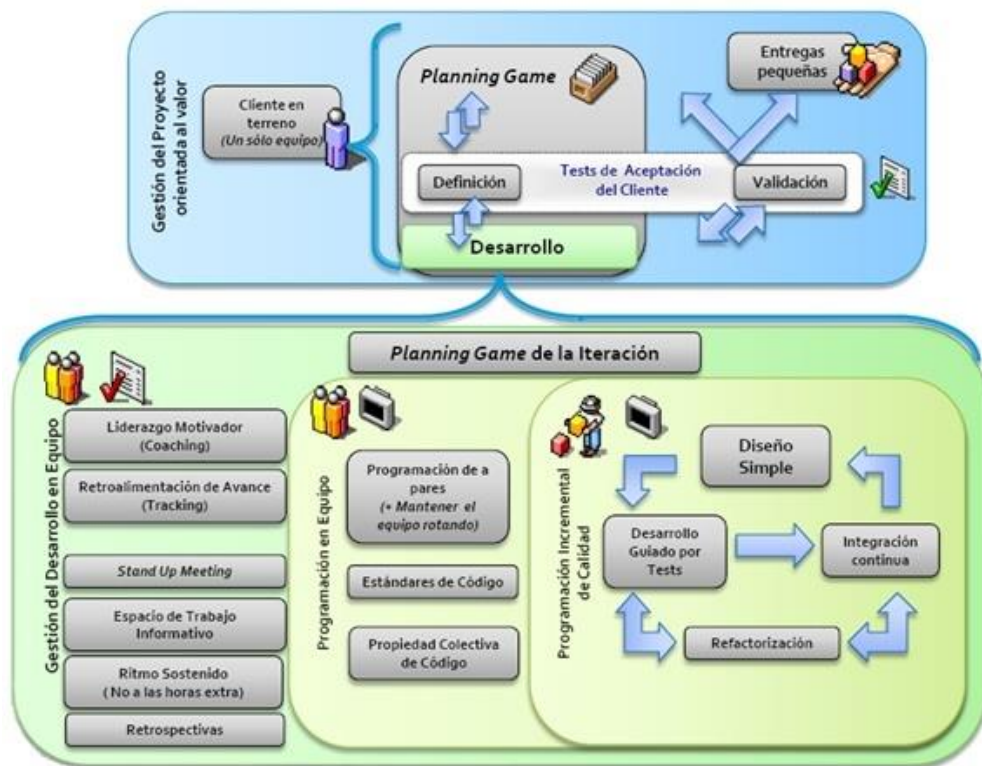
En cuanto a las métricas hay un reporte típico de scrum, denominado **burndown chart** consiste en hacer un seguimiento sobre la base del trabajo que falta por hacer.

A continuación se definirá el tercer método utilizado en este proyecto OPENUP, sus fases, roles, artefactos y particularidades.

**4.1.3 OpenUp<sup>69</sup>.** OPENUP es un proceso unificado que aplica enfoques iterativos e incrementales dentro de un ciclo de vida estructurado<sup>70</sup>. OPENUP se enfoca en algo real y objetivo, es ágil y se centra en la naturaleza colaborativa de desarrollo de software. Puede ser utilizado en una amplia variedad de proyectos.<sup>71</sup>

La figura 11 da una vista general de los componentes de un método ágil como lo es OPENUP.

**Figura 11.** Vista general método ágil<sup>72</sup>



**Fuente.** Vista general de un método ágil. Disponible en: <http:// analisis1daid.wikispaces.com/Metodolog%C3%ADas+%C3%81giles>

<sup>69</sup> EPF.ECLIPSE. OpenUP (s.l.) [En línea]. Mar. 2010. [Citado el 25 noviembre de 2014.] Disponible en internet en: <http://epf.eclipse.org/wikis/openup/>

<sup>70</sup> [http://www.ub.edu.ar/catedras/ingenieria/ing\\_software/ubftecwwdfd/cicloproy/ciclo.htm](http://www.ub.edu.ar/catedras/ingenieria/ing_software/ubftecwwdfd/cicloproy/ciclo.htm)

<sup>71</sup> EPF.ECLIPSE. OpenUP (s.l.) [En línea]. Mar. 2010. [Citado el 25 noviembre de 2014.] Disponible en internet en: <http://epf.eclipse.org/wikis/openup/>

<sup>72</sup> <http:// analisis1daid.wikispaces.com/Metodolog%C3%ADas+%C3%81giles>

## **Roles OPENUP:**

**Ciente.** Representan a los individuos o entidades cuyas necesidades serán cubiertas por el proyecto.

**Analista.** Es el encargado de interactuar con el cliente y sus necesidades, su papel principal es entender lo que el cliente desea para así establecer la prioridad de los requisitos identificados.

**Arquitecto.** Es el encargado de desarrollar la arquitectura del software, lo que incluye tomar las decisiones técnicas clave que limitan el diseño general y la implementación del proyecto.

**Desarrollador.** Se encarga del diseño e implementación y pruebas unitarias de una parte del sistema, también integra los elementos que hacen parte de una solución.

**Tester.** Es el encargado de diseñar, identificar, implementar y ejecutar las pruebas que sean necesarias para que el sistema funcione de la mejor manera, además debe documentar los resultados obtenidos.

**Gestor del proyecto.** Es el encargado dirigir la proyección del proyecto junto con el equipo y los clientes. Así mismo debe lograr que el equipo se mantenga cumpliendo con sus tareas.

**Desarrollador del curso.** Es el encargado de crear materiales de capacitación ya sea para usuarios finales del sistema o al personal de apoyo a la producción que son responsables del mantenimiento del sistema.

**Ingeniero de implementación.** Es el responsable ante el gestor de despliegue para el despliegue seguro de uno o más lanzamientos en el entorno de producción. Ayuda a prevenir la introducción de código malo o no probado en producción.

**Gestor de despliegue.** Este rol es responsable de la gestión y la programación de lanzamientos sincronizados a través de uno o más programas y / o líneas de productos.

**Escritor técnico.** Un escritor técnico es el que ayuda a que un miembro del equipo de desarrollo complete la documentación necesaria para el propietario del producto, los usuarios finales, y personal de apoyo para comprender y ser capaz de utilizar las características entregadas.

**Entrenador.** Ofrece capacitación ya sea a usuarios finales del sistema o al personal de apoyo a la producción que son responsables del mantenimiento del sistema.

**Ingeniero de procesos.** Este rol se encarga de que los miembros del equipo no se vean obstaculizados en su trabajo.

**Especialista en herramientas.** Es el encargado de brindar asistencia técnica al proyecto, esto incluye la selección, adquisición y configuración de las herramientas.

**Artefactos OPENUP**<sup>73</sup>. Un artefacto en OpenUp se considera a todo aquello que una tarea necesita para realizar su función, o bien la produce o modifica. Los distintos roles existentes son los encargados de crear y actualizar los artefactos.<sup>74</sup>

OPENUP es un método más completo en cuanto artefactos para cada actividad que se desarrolla durante el proyecto existirá un artefacto, la siguiente tabla muestra cada artefacto su descripción y el rol que lo ejecuta, cuando se habla de información que requiere, hace referencia en que algunos de estos artefactos necesitan unas plantillas, o una información específica para su ejecución, en algunos casos no es necesaria este tipo de información. El cuadro 20 muestra los artefactos con su descripción, quien lo ejecuta y la información necesaria.

**Cuadro 20.** Artefactos OPENUP

Nombre del artefacto	Descripción	Rol	Información que requiere.
<b>Plan del proyecto</b>	<p>En este artefacto se reúne toda la información necesaria para gestionar el proyecto a nivel estratégico.</p> <p>En este artefacto se describe cómo se organiza el proyecto, e identifica lo que se seguirá en las prácticas. Además, define los parámetros para el seguimiento de los avances del proyecto, y especifica los objetivos de alto nivel de las iteraciones y sus hitos.</p> <p>El plan del proyecto permite a los interesados y a otros miembros del equipo entender el panorama general. Actualizar el plan con la frecuencia necesaria, por lo general al final de cada iteración, con el fin de reflejar los cambios en las prioridades y necesidades, así como registro de las</p>	Gerente del proyecto.	<p>Introducción</p> <p>Organización del proyecto.</p> <p>Prácticas y mediciones del proyecto.</p> <p>Hitos y objetivos del proyecto.</p> <p>Despliegue.</p> <p>Lecciones aprendidas.</p>

<sup>73</sup> EPF.ECLIPSE. OpenUP (s.l.) [En línea]. Mar. 2010. [Citado el 25 noviembre de 2014.] Disponible en internet en: <http://epf.eclipse.org/wikis/openup/>

<sup>74</sup> EPF.ECLIPSE. OpenUP (s.l.) [En línea]. Mar. 2010. [Citado el 25 noviembre de 2014.] Disponible en internet en: <http://epf.eclipse.org/wikis/openup/>

Cuadro 20. (Continuación)

	lecciones aprendidas del proyecto.		
<b>Proceso definido del proyecto</b>	<p>Este artefacto describe el proceso que el proyecto debe seguir para producir los resultados esperados.</p> <p>Un proceso del proyecto normalmente describe o hace referencia a los siguientes elementos:</p> <p>*Qué procesos y políticas de la organización deben ser atendidas.</p> <p>*Qué proceso estándar, en su caso, está siendo adoptado por el proyecto.</p> <p>*Cualquier adaptación del proceso estándar o las desviaciones de los mandatos políticos.</p> <p>*Aprobaciones para cambios.</p> <p>*Información de las directrices de que el proyecto quiere utilizar, además de la información contenida en el proceso principal.</p>	Ingeniero de procesos	<p>introducción de propósito alcance</p> <p>Definiciones, Acrónimos, y Abreviaturas</p> <p>Referencias de visión de conjunto</p> <p>Ciclo de vida</p> <p>Modelo</p> <p>Flujo de trabajo</p> <p>Productos de trabajo</p> <p>Informes</p> <p>Roles</p> <p>Directrices y procedimientos específicos del proyecto</p>
<b>Construcción</b>	<p>Esta versión de trabajo del sistema o parte del sistema es el resultado de poner la aplicación a través de un proceso de construcción (típicamente un script de construcción automatizada) que crea una versión ejecutable, o uno que se ejecuta. Esta versión ejecutable tendrá típicamente una serie de ficheros de soporte que también se consideran parte de este artefacto.</p>	Developer	No requiere ninguna plantilla.
<b>Libro de arquitectura</b>	<p>En este artefacto se describe la arquitectura del software. Proporciona una lista donde se observan los problemas que pueda presentar la arquitectura y las decisiones que se han tomado con respecto a diseños, modelos y códigos.</p> <p>Este artefacto debe informar a los miembros del equipo como se organizan para que ellos sepan cómo adaptarse a las necesidades del sistema.</p>	Arquitecto	<p>propósito</p> <p>Metas de arquitectura y filosofía</p> <p>Suposiciones y dependencias</p> <p>Requisitos de gran importancia arquitectónica</p> <p>Las decisiones, las limitaciones, y justificaciones</p>

Cuadro 20. (Continuación)

			Mecanismos de arquitectura abstracciones clave Capas o marco arquitectónico vistas de arquitectura
<b>Diseño</b>	<p>Este artefacto describe la realización de la funcionalidad que el sistema requiere y sirve como una abstracción del código fuente. Este producto describe los elementos que compondrán el sistema al implementarse.</p> <p>Mientras que la arquitectura se centra en las interfaces, los patrones y las decisiones clave, el diseño da cuerpo a los detalles técnicos en preparación para la ejecución, o como parte de la implementación.</p>	Developer	<p>estructura de Diseño subsistemas Patrones Realización Requisitos.</p>
<b>Visión</b>	<p>La visión contiene un resumen de los requisitos básicos previstos para el sistema. La visión sirve como entrada para la comunicación de la fundamental "qué y por qué" para el proyecto y proporciona una estrategia contra la cual todas las decisiones futuras se pueden validar.</p> <p>La visión debe reunir a los miembros del equipo detrás de una idea y darles el contexto para la toma de decisiones en el área de requisitos. La visión debe ser visible para todos en el equipo.</p>	Analista	<p>introducción Posicionamiento Las descripciones de las partes interesadas Descripción del producto Otros Requisitos del producto</p>
<b>Glosario</b>	<p>Este artefacto define los términos importantes utilizados en el proyecto, con eso se aclara el vocabulario que se maneja.</p> <p>Los propósitos de este artefacto son: *Grabar todos los términos del proyecto para que todos tengan un entendimiento común de ellos. *Lograr coherencia mediante la</p>	Analista	No requiere ninguna plantilla.



Cuadro 20. (Continuación)

	<p>promoción del uso de la terminología común en todo el proyecto. *Proporcionar términos importantes para el equipo de análisis y diseño.</p>		
<b>Prueba desarrollador</b>	<p>Este artefacto cubre todos los pasos para validar un aspecto específico de un elemento de aplicación con ello se asegura que el elemento funciona según lo especificado. La prueba debe ser documentada de manera que es evidente tras la finalización de la prueba si el elemento de aplicación se ha ejecutado correctamente.</p>	Desarrollador	No requiere ninguna plantilla.
<b>Caso de uso</b>	<p>Este artefacto captura el comportamiento del sistema para producir un resultado viable de valor para aquellos que interactúan con el sistema. Los casos de uso se usan para: entender el comportamiento del sistema, diseñar elementos que apoyan ese comportamiento, identificar los casos de prueba, planificar y evaluar el trabajo y escribir la documentación del usuario.</p>	Analista	<p>Nombre. Breve descripción. Flujo de eventos. Escenarios claves. Requisitos especiales. Requisitos. Post-condiciones. Puntos de extensión.</p>
<b>Modelo de caso de uso</b>	<p>Este artefacto captura un modelo de las funciones previstas en el entorno del sistema y sirve como un contrato entre el cliente y el equipo.  Este artefacto presenta una visión general del comportamiento previsto del sistema. Es la base para un acuerdo entre los interesados y el equipo del proyecto en lo que respecta a la funcionalidad prevista del sistema. También guía diversas tareas en el ciclo de vida de desarrollo de software.</p>	Analista	No requiere ninguna plantilla.
<b>Requisitos del sistema</b>	<p>Este artefacto se encarga de capturar los atributos y limitaciones que tiene el sistema de calidad. También</p>	Analista	<p>introducción Requisitos funcionales a</p>

Cuadro 20. (Continuación)

	<p>captura los requisitos funcionales de todo el sistema.</p> <p>Este artefacto se usa con el fin de: describir los atributos de calidad del sistema, y las limitaciones que las opciones de diseño deben satisfacer para entregar los objetivos de negocio, para capturar los requerimientos funcionales que no se expresan como casos de uso, para evaluar el tamaño, coste, y la viabilidad del sistema propuesto y para entender los requisitos de nivel de servicio para la gestión operativa de la solución</p>		<p>todo el sistema</p> <p>Calidades</p> <p>Interfaces del sistema</p> <p>Reglas de Negocio</p> <p>cumplimiento del Sistema</p> <p>documentos del sistema</p>
<b>Documentación del producto.</b>	<p>Este artefacto proporciona documentación sobre el producto para el propietario. Se crea la documentación del producto para el beneficio de la división de marketing de una organización, el director del programa, y aquellas personas que deben evaluar el valor comercial de un sistema en particular.</p> <p>Este es un aspecto a menudo pasado por alto de ejecución equipo de desarrollo. El equipo debe tener en cuenta que los clientes de un lanzamiento en particular por lo general no son técnicos propios, pero sí requieren un conocimiento bastante detallado de cómo funciona su producto y cómo cumple declararon metas y necesidades del negocio.</p>	Escritor técnico	No requiere ninguna plantilla.
<b>Documentación de soporte</b>	<p>En este artefacto se usan documentos para dar información y soporte a quien la requiera. Existen 3 niveles de apoyo si el primer nivel no es capaz de solucionar alguna duda para al segundo nivel y así hasta el tercero.</p>	Escritor técnico	<p>Manuales de usuario con instrucciones de trabajo, descripciones de procesos y procedimientos</p> <p>Comunicaciones , entregables</p> <p>capacitación y</p>

Cuadro 20. (Continuación)

			transferencia de conocimiento Manuales de apoyo y operaciones Información de servicio, incluyendo Ayuda guiones Escritorio
<b>Materiales de formación</b>	Este artefacto es la unión de todos los materiales necesarios para capacitar a los usuarios finales y el personal de apoyo a la producción en las características y funcionamiento interno de un producto.	Desarrollador del curso	folletos de ayudas de trabajo Tutoriales Demos en línea viñetas de video Los ejercicios de laboratorio Cuestionarios Materiales del taller, etc.
<b>Documentación del usuario.</b>	Son los documentos que pueden ser utilizados por los usuarios finales de un sistema o producto en particular. Este tipo de documentación normalmente está escrita de una manera que permite a los usuarios del sistema encontrar fácilmente la información que necesitan para utilizar el producto	Escritor técnico	Manuales de usuario. Tutoriales. Ayuda en línea. Instrucciones de instalación.
<b>Instrumentos</b>	Este artefacto de trabajo son las herramientas necesarias para apoyar el desarrollo de software.	Especialista en herramientas.	No requiere ninguna plantilla.
<b>Plan de backout</b>	Este artefacto sirve para definir los criterios y procedimientos que deben seguirse en caso de que una liberación del producto deba ser revertida.	Ingeniero de implementación.	
<b>Plan de despliegue.</b>	Un plan de despliegue se utiliza para documentar toda la información que necesitan los ingenieros de despliegue para desplegar un lanzamiento con éxito.	Ingeniero de implementación	El alcance de la liberación y una visión general de las capacidades que se desplegarán

Cuadro 20. (Continuación)

			<p>El momento y dependencias para implementar los componentes para varios nodos</p> <p>Los riesgos o problemas relacionados con el lanzamiento basan en una evaluación de riesgos</p> <p>La organización del cliente, las partes interesadas, y la comunidad de usuarios finales que se verán afectados por la liberación</p> <p>La persona o personas que tienen la autoridad.</p>
<p><b>Comunicaciones de lanzamiento.</b></p>	<p>Este artefacto proporciona información a las partes interesadas de que un producto (o subconjunto) se ha colocado en la producción.</p>	<p>Ingeniero de implementación</p>	<p>Si bien no existe un formato prescrito por el artefacto de comunicaciones de liberación, cada comunicado debe indicar los mecanismos de entrega preferida (por ejemplo, la notificación beeper, llamadas telefónicas, un anuncio a un sitio web de la</p>

Cuadro 20. (Continuación)

		<p>liberación interior, vivir o presentaciones pregrabadas por la alta dirección, etc.) y en general, responder a preguntas como: ¿Quiénes son las partes (stakeholders) que estén interesados en saber que un comunicado de la producción ha tenido lugar? ¿Qué específicamente (características, funciones, componentes) se ha colocado en la producción?</p>
--	--	---

Fuente. Eclipse. (Febrero de 2015). OpenUp. Obtenido de <http://epf.eclipse.org/wikis/openup/>

### Ciclo de vida OpenUp:<sup>75</sup>

**Fase de inicio.** En esta fase las necesidades del cliente son expuestas y convertidas en objetivos del proyecto. También se definen. el ámbito, los límites, el criterio de aceptación, los casos de uso críticos, una estimación inicial del coste y un boceto de la planificación.

**Fase de elaboración.** En esta fase se define el alcance y la arquitectura del sistema. Se elabora un plan donde se establezcan los requisitos, el proceso de desarrollo, las herramientas, la infraestructura a utilizar y el entorno de desarrollo. Al final de esta fase se debe tener claro, los casos de uso, los actores, la arquitectura y un prototipo ejecutable.

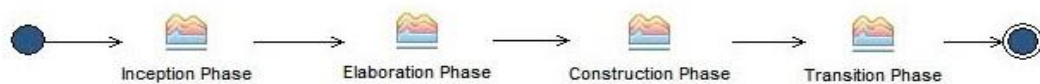
**Fase de construcción.** En esta fase se deben realizar, integrar y probar todos los componentes y funcionalidades del sistema que falten por probar. Estos incrementos ejecutables deben ser desarrollados de forma rápida sin que se pierda la calidad.

<sup>75</sup> EPF.ECLIPSE. OpenUP (s.l.) [En línea]. Mar. 2010. [Citado el 25 noviembre de 2014.] Disponible en internet en: <http://epf.eclipse.org/wikis/openup/>

**Fase de transición.** En esta fase el producto está maduro y se introduce a los usuarios, consta de subfases de pruebas de versiones beta, pilotaje y capacitación de los usuarios finales y de los encargados del mantenimiento del sistema. Dependiendo de la respuesta obtenida por los usuarios puede ser necesario realizar cambios en las entregas finales o implementar alguna funcionalidad más.

Seguidamente se muestra la figura 12 correspondiente al ciclo de vida de OPENUP:

**Figura 12.** Ciclo de vida OPENUP<sup>76</sup>



**Fuente.** EPF.ECLIPSE. OpenUP (s.l.) [En línea]. Mar. 2010. Disponible en internet en: <http://epf.eclipse.org/wikis/openup>

### **Particularidades:**

**Principios básicos**<sup>77</sup>. OPENUP cuenta con una serie de principios básicos que lo fundamentan los cuales son:

Balancear las prioridades competitivas para maximizar el valor de los participantes del proyecto.

Desarrollar una arquitectura al inicio del proceso, con la finalidad de minimizar riesgos y planificar el desarrollo.

Evolucionar para mantener retroalimentación continua en aras de una mejora constante.

Promover la colaboración para alinear los intereses comunes, difundir el conocimiento sobre el proyecto y generar un entorno de trabajo saludable.

**Elementos.** OPENUP se organiza en dos dimensiones: Contenido metodológico y contenido procedimental. El contenido metodológico es el que define elementos metodológicos tales como disciplinas, tareas, artefactos y procesos, independientemente de

<sup>76</sup> <http://epf.eclipse.org/wikis/openup/>

<sup>77</sup> Ríos S, Hinojoza C & Delgado R.. (2013). APLICACIÓN DE LA METODOLOGIA OPENUP EN EL DESARROLLO DEL SISTEMA DE DIFUSIÓN DE GESTIÓN DEL CONOCIMIENTO DE LA ESPE. 2013, de La Escuela Politécnica del Ejército Sitio web: <http://repositorio.espe.edu.ec/bitstream/21000/6316/1/AC-SISTEMAS-ESPE-047042.pdf>

cómo se usen estos o se combinen. El contenido procedimental, por el contrario, es donde se aplican todos estos elementos metodológicos dentro de una dimensión temporal, pudiéndose crear multitud de ciclos de vida diferentes a partir del mismo subconjunto de elementos metodológicos.

Los elementos que forman parte de OPENUP son los siguientes:

**Disciplinas.** Las disciplinas que hacen parte de OpenUp son las siguientes. Requisitos, arquitectura, desarrollo, pruebas, gestión de proyecto, gestión de la configuración y de cambio.

**Tareas.** Una tarea es una parte del trabajo que debe ser realizada por un rol pero al mismo tiempo el encargado de realizarla puede recibir la ayuda de otro rol. En esto se basa la naturaleza colaborativa de OpenUp.

**Artefactos.** Un artefacto en OpenUp se considera a todo aquello que una tarea necesita para realizar su función, o bien la produce o modifica. Los distintos roles existentes son los encargados de crear y modificar los artefactos.

**Procesos.** Los procesos toman los elementos metodológicos y los relacionan entre si dentro de secuencias temporales que satisfacen las necesidades de distintos tipos de proyectos

### **Administración.**

OPENUP es un proceso iterativo cuyas iteraciones se distribuyen a través de cuatro fases: Inicio, Elaboración, Construcción, y Transición.

Cada fase podrá tener tantas iteraciones como se requiera dependiendo del grado de novedad del dominio de negocio, de la tecnología a ser utilizada, de la complejidad de la arquitectura de la solución y del tamaño del proyecto, entre otros factores. Para que los equipos de trabajo planeen sus iteraciones de forma rápida, el OPENUP provee plantillas con estructuras para la división del trabajo (EDT) en cada iteración y una plantilla de EDT para todo el proceso considerado de principio a fin.

Las iteraciones pueden tener duraciones variables dependiendo de las características del proyecto. Iteraciones de un mes son recomendadas típicamente ya que este periodo de tiempo provee:

Una cantidad de tiempo razonable para que los proyectos entreguen incrementos considerables en funcionalidad.

Retro alimentación temprana y frecuente por parte de los usuarios.

Administración a tiempo de los riesgos y problemas encontrados durante el curso del proyecto.

OPENUP está diseñado para ofrecer guía en el proceso de desarrollo en proyectos pequeños:

Equipos de 3 a 6 personas  
3 a 6 meses de trabajo

En este capítulo se investigó lo que es una metodología, el concepto de una metodología tradicional y un método ágil, presentando las características de las metodologías tradicionales y una comparación entre tradicional vs ágil, de igual manera se detallaron los métodos ágiles XP, SCRUM Y OPENUP, se pudo descubrir y entender sus roles, artefactos, ciclo de vida y particularidades.

#### **4.2 COMPARACIÓN ENTRE FASES, ROLES Y ARTEFACTOS DE XP, SCRUM Y OPENUP**

Como se vio anteriormente estos tres métodos tienen en común una serie de características como lo son el ciclo de vida, los roles y los artefactos, en este capítulo se busca comparar cada uno de ellos para así descubrir que similitudes tienen.

Los cuadros 21, 22 y 23 planteados a continuación están organizados de tal manera que se pueda identificar claramente la relación que existe entre los tres métodos ágiles, en la primera columna a la izquierda se describe la característica general y después en cada método se identifica la parte a la cual corresponde.

**4.2.1 Ciclo de vida XP, SCRUM Y OPENUP.** Cada método ágil utiliza un ciclo de vida para el desarrollo del proyecto esto lo hace para definir etapas específicas y controlar la organización y manejo del proyecto.

En el cuadro 21 se muestra la relación que existe entre el ciclo de vida que utiliza cada método.

**Cuadro 21.** Relación entre ciclos de vida

<b>CICLO DE VIDA</b>			
<b>RELACION EXISTENTE</b>	<b>XP</b>	<b>SCRUM</b>	<b>OPENUP</b>
Se describen objetivos, se asigna prioridad, se realiza la arquitectura.	Fase de exploración de la planificación de la entrega (reléase)	Fase de planeamiento Fase de montaje	Fase de inicio de elaboración
Desarrollo del producto y	Fase de iteraciones	Fase de desarrollo	Fase de construcción



Cuadro 21. (Continuación)

entregable de prueba.	de	Fase de producción	de		
Liberación del producto, realización de documentación	del	Fase muerte del proyecto.	del	Fase de liberación	Fase de transición

**Fuente.** Autor del proyecto

El cuadro anterior muestra la relación que existe entre los ciclos de vida de XP, SCRUM y OPENUP, al compararlas se puede observar que las dos primeras etapas de cada uno de ellos realizan actividades similares; lo cual consiste en definir objetivos, requisitos o casos de uso, asignar prioridad, se establece el tiempo de desarrollo de cada uno de ellos, y se hace un prototipo de la arquitectura, en la fase siguiente se tiene el desarrollo del sistema y un entregable que se pone a prueba mientras SCRUM y OPENUP lo realiza en una sola fase XP lo separa en dos que son iteraciones y producción.

Después de haber definido objetivos y demás actividades de la primera fase SCRUM y OPENUP libera el producto, capacita a los usuarios finales y realiza la debida documentación. En cuanto XP antes de realizar esto asigna al cliente tareas de soporte y si es necesario se hace cambio de estructura luego si procede con la fase final.

**4.2.2 Roles XP, SCRUM Y OPENUP.** Un equipo de desarrollo debe tener diferentes expertos con funciones y responsabilidades específicas dentro de un grupo, esto permite lograr un producto con mayor calidad y que al presentarse cualquier inconveniente sea solucionado inmediatamente por quien esté a cargo.

A continuación el cuadro 22 muestra la relación existente entre cada uno de los roles de cada método ágil.

**Cuadro 22.** Relación entre roles

ROLES				
RELACION EXISTENTE	XP	SCRUM	OPENUP	
Programación y desarrollo del producto	Desarrollador	Equipo de desarrollo	Arquitecto y desarrollador.	
Requisitos del sistema	Cliente	Product owner Customer	Cliente	
Guía de proceso	Coach	Scrum master	Gestor del proyecto	
Seguimiento del desarrollo de trabajo.	Tracker	Product owner	Ingeniero de implementación y arquitecto	
Vinculo cliente-	Big boss	Scrum master	Analista	

Cuadro 22. (Continuación)

equipo	Tester		
Pruebas de funcionamiento	Tester	Equipo de desarrollo	Tester Gestor de despliegue

**Fuente.** Autor del proyecto

El cuadro 22 muestra la relación entre cada uno de los roles de XP, SCRUM y OPENUP, al compararlos se puede observar que OPENUP maneja un número mayor de roles a diferencia de XP y SCRUM que son menos explícitos. Esto no quiere decir que algún método tenga roles innecesarios, cada uno tiene un fin característico y puede ser comparado con otro de cualquier método diferente pero que cumpla la misma función, con esto se hace referencia a lo siguiente:

En cuanto a la programación y desarrollo del producto XP tiene a cargo al desarrollador, SCRUM maneja estos procesos en el equipo de desarrollo, el cual está conformado por un grupo de profesionales encargados de manejar las aplicaciones, mientras que en OPENUP los encargados de manejar esto son el desarrollador y el arquitecto.

Para establecer los requisitos del sistema los tres roles tienen un cliente, en XP el cliente es el encargado de establecer las historias de usuario y asignarles una prioridad, en SCRUM se denomina el product owner que es el encargado de gestionar la product backlog y en OPENUP el cliente expone sus necesidades y el equipo se encarga de sacar los requisitos.

Cada uno de los métodos tiene un encargado de exponer los valores de cada uno de ellos, prácticas y reglas y hacer que se cumplan en el caso de XP el responsable es el coach, en SCRUM está el scrum master quien además cumple otras funciones y en OPENUP este papel es desempeñado por el gestor del proyecto.

Cuando se refiere al seguimiento del proceso estos métodos tienen un rol asignado en XP el tracker es el encargado de hacer seguimiento a cada iteración, SCRUM el product owner realiza esto y en OPENUP quien realiza esto es el ingeniero de implementación y el arquitecto.

Debe existir un vínculo entre el cliente y el equipo de desarrollo, este rol se encarga de coordinar y lograr que lo que pide el cliente sea cumplido, el encargado de esto en XP es el big boss y el tester quien también realiza otras funciones, en SCRUM es el scrum master y en OPENUP es el analista.

Al proyecto que se desarrolla deben realizarle una serie de pruebas de funcionamiento antes de ser entregado para esto XP dispone del tester que ejecuta pruebas regularmente, SCRUM esto lo hace el equipo de desarrollo y en OPENUP los encargados son el tester y el gestor de despliegue.

OPENUP es mucho más organizado y va más allá de XP y SCRUM en cuanto a definir roles para la documentación y capacitación de cómo debe usarse el producto final, para esto cuenta con un escritor técnico, un desarrollador del curso y un entrenador.

#### 4.2.3 Artefactos XP, SCRUM Y OPENUP:

Un artefacto es todo aquello que se necesita para que una tarea sea realizada, el cuadro 23 muestra de que manera maneja cada método sus artefactos y de cuál es la relación que existe entre ellos.

**Cuadro 23.** Relación existente entre artefactos

ARTEFACTOS			
RELACION EXISTENTE	XP	SCRUM	OPENUP
Explicación de Lo que el producto o sistema debe realizar	Historias de usuario	Producto backlog	Plan del proyecto Proceso definido del proyecto Libro de arquitectura Diseño Visión Caso de uso Modelo de caso de uso Requisitos del sistema
Culminación y cumplimiento de requisitos.	Pruebas de aceptación	Sprint backlog	Prueba desarrollador Plan de despliegue

**Fuente.** Autor del proyecto

El cuadro 23 muestra la relación que existe entre cada uno de los artefactos de XP, SCRUM y OPENUP, al compararlos se puede observar que cada uno de ellos utiliza artefactos totalmente diferentes pero igualmente funcionales mientras XP utiliza historias de usuario, SCRUM maneja una lista del producto y OPENUP casos de uso.

En cuanto a la parte de la culminación y el cumplimiento de los requisitos propuestos XP y OPENUP utiliza una serie de pruebas mientras que SCRUM lo hace mediante el sprint backlog.

Para finalizar la parte de las similitudes se puede decir que los métodos ágiles poseen características esenciales que los hace particulares, pero al analizarlos como se hizo en este capítulo, se puede notar que cada uno de ellos tiene un proceso y unas fases para planificar

el desarrollo del proyecto que aunque no son iguales en cantidad o que no tienen el mismo nombre tienen como finalidad establecer un patrón para el desarrollo del proyecto.

Otra característica es que para el desarrollo de los requerimientos cada método usa diferente tipo de escenarios, pero en cada uno de ellos se especifica claramente lo que se va a realizar y lo que desea el usuario.

Por otra parte cada método cuenta con un equipo de trabajo y aunque en algunos hay mayor número de integrantes, que en otros en cualquier caso siempre habrá alguien que desempeñe un rol específico en el desarrollo del proyecto.

Por otra parte y de la misma manera en que se logró encontrar las similitudes entre los tres métodos se puede decir que en cada una de sus etapas existen ciertas diferencias que se expresan a continuación:

En cuanto a roles XP tiene un consultor que es un miembro externo al equipo con conocimientos en un tema específico del cual puedan surgir problemas en algún momento, SCRUM posee un management que realiza tareas de revisión y seguimiento de el sprint backlog y por ultimo OPENUP que tiene un participante para cada tarea a realizar como ingeniero de procesos, escritor técnico, especialista en herramientas entre otros, por lo cual se diferencia totalmente de los otros métodos.

Al hablar de artefactos XP para el desarrollo de sus historias de usuario usa tareas de ingenierías plantillas que se necesitan para hacerle seguimiento al desarrollo mientras que SCRUM a diferencia de ellos dos usa un artefacto llamado burndown chart que es una gráfica que muestra el progreso en el desarrollo de los requisitos por ultimo OPENUP es más completo en cuanto artefactos y posee documentación del usuario, del producto, un plan de backup, visión, comunicaciones de lanzamiento y demás.

Las fases del ciclo de vida son similares para los tres métodos.

#### **4.3 PROPUESTA DE UN NUEVO MÉTODO ÁGIL A PARTIR DE LAS CARACTERÍSTICAS PROPIAS DE XP, SCRUM Y OPENUP, INTEGRANDO LOS PUNTOS EN COMÚN QUE EXISTAN ENTRE ELLAS Y ADICIONANDO CARACTERÍSTICAS QUE LO COMPLEMENTEN PARA DEFINIR LA ESTRUCTURA QUE TENDRÁ EL NUEVO MÉTODO.**

A continuación se describe la creación de un nuevo método ágil a partir de XP, SCRUM y OPENUP. Estos tres métodos serán fusionados para extraer lo común y lo mejor de ellos y así lograr un nuevo procedimiento de desarrollo de software ágil que aparte de integrar las características de esos tres métodos, cumplirá con los valores que estipula el manifiesto ágil como se desarrolló en el segundo capítulo en donde se estableció la relación existente entre cada uno de los roles, ciclo de vida y artefactos de cada método se podrán organizar las nuevas características.

## **Método propuesto: MAF (Métodos ágiles fusionados)**

Este nuevo método propuesto (En adelante MAF), es un método de desarrollo ágil, basado en una serie de prácticas y valores extraídos de XP, SCRUM y OPENUP. Tiene como objetivo mejorar la productividad y calidad en los proyectos realizados por los desarrolladores de software y así mejorar su desempeño en el uso de técnicas ágiles.

MAF tiene la adopción de tres métodos ágiles. Tiene un ciclo de vida iterativo e incremental<sup>78</sup>, que produce que el proyecto sea mejorado sucesivamente, MAF busca que los grupos de gestión de software manejen marcos de desarrollo ágil, tengan buena actitud frente a cambios repentinos del sistema, trabajen en equipo y manejen varios procesos en cada etapa de desarrollo. Este nuevo método tiene una serie de propósitos que son cumplidos a través de la consecución de los siguientes objetivos:

Mejorar la calidad de los proyectos desarrollados por los grupos de gestión de software.

Mostrar a estos grupos la ventaja que se tiene al desarrollar software usando métodos ágiles.

Dar una guía que les facilite el desarrollo de sus proyectos.

Promover los valores planteados por el manifiesto ágil como lo son la colaboración con el cliente, entregas de software funcional y la respuesta al cambio.

**Roles MAF.** Los roles planteados en esta propuesta se obtienen a partir de la anterior comparación de los tres métodos y en donde se establece la relación existente entre ellos, con eso se puede concluir cuales son los roles comunes y por lo cual serán utilizados en MAF. Cada integrante del equipo tiene un fin general lograr que cada iteración se desarrolle con éxito.

Ahora se muestran los roles que son utilizados por MAF que nacen de la fusión de las características comunes entre XP, SCRUM y OPENUP que se pueden ver en el cuadro 22, se decidió usar esos nombres para el mejor entendimiento de quien desee utilizar MAF para el desarrollo de su proyecto.

**Cliente.** Es el encargado de establecer los requisitos del sistema y de definir sus objetivos, hará parte del equipo de trabajo y es quien debe asignar prioridad a cada historia de usuario y decidir cuál de estas se implementara en cada iteración.

**Desarrollador.** Será el encargado de la programación del producto, diseña la arquitectura del sistema y ejecuta pruebas unitarias. Transforma las historias de usuario en código.

**Facilitador.** Es el encargado de dar guías al grupo para que cumpla con cada uno de los requisitos establecidos por MAF, el objetivo de un facilitador es que el equipo entienda y

---

<sup>78</sup> PROYECTOSAGILES. Iterativo e incremental (s.l.) [En línea]. (s.f). [Citado el 25 noviembre de 2014.] Disponible en internet en: <http://www.proyectosagiles.org/desarrollo-iterativo-incremental>

manee las directrices establecidas en el nuevo método y no solo dando lecciones teóricas sino proponiendo ejemplos.

**Delegado.** Es el vínculo entre el cliente, desarrolladores y el equipo en general. Se encarga de ayudar al cliente a escribir sus historias de usuario su labor principal es coordinación.

**Tester.** Es el encargado de realizar pruebas de funcionamiento constantemente, realiza pruebas que denoten el avance del equipo y se los comunica. Es el encargado de dar seguimiento al proceso de desarrollo del sistema, realiza pruebas en cada iteración para que así no se tenga ningún fallo.

La figura 13 nos muestra un bosquejo del conjunto de los roles usados por MAF.

**Figura 13.** Roles MAF



**Fuente.** Autor del proyecto

**Artefactos.** Un artefacto es un modelo a seguir en el desarrollo de software que se utiliza para realizar tareas específicas.<sup>79</sup>

A continuación se muestran los artefactos que usara MAF, estos serán documentados para lograr que el método tenga las ventajas mencionadas anteriormente, el uso de plantillas hará que sea más fácil para los grupos de desarrollo la realización de las actividades correspondientes.

En el capítulo anterior se destacaron dos características especiales que ayudan a definir la relación de artefactos entre los tres métodos a partir de esto se concluirá cuáles son los artefactos necesarios para que MAF logre una mejor especificación al desarrollar software.

Documentar las fases de desarrollo en un proyecto de desarrollo de software es un punto realmente importante, esto tiene como propósito eventuales correcciones, usabilidad, mantenimiento futuro y ampliaciones del sistema.<sup>80</sup>

<sup>79</sup> Scott Ambler. “Artifacts for Agile Modeling: The UML and beyond”, 2003

<sup>80</sup> Pressman, R. (2006). Ingeniería del software: Un enfoque práctico . México: McGraw-Hill.

El realizar documentación hace que sea más fácil realizar mantenimiento a un sistema, además de esto facilita la resolución de problemas y mantiene un sistema ordenado. Es fundamental disponer de la documentación si se pretende compartir el software.

Otro aspecto importante, Según Pressman, es que el diseño del software es realmente un proceso de muchos pasos pero que se clasifican dentro de uno mismo. En general, la actividad del diseño se refiere al establecimiento de las estructuras de datos, la arquitectura general del software, representaciones de interfaz y algoritmos. El proceso de diseño traduce requisitos en una representación de software.

El diseño es el primer paso en la fase de desarrollo de cualquier producto o sistema de ingeniería. De acuerdo con Pressman, el objetivo del diseño es producir un modelo o representación de una entidad que se va a construir posteriormente.<sup>81</sup> Según esto el diseño es la representación general del software, es por eso que deben especificarse muy bien los requisitos y crear un artefacto completo que abarque lo que es el diseño para así poder tener un mejor resultado.

Según lo aprendido en los capítulos anteriores se puede destacar tres elementos importantes al momento de usar artefactos en un método ágil:

Explicación de lo que el producto o sistema debe realizar.  
Culminación y cumplimiento de los requisitos.  
Documentación final.

El objetivo de utilizar los artefactos que serán usados a continuación es crear un método que use un número de artefactos que cubran con todas las necesidades para lograr un buen resultado, XP al igual que SCRUM usa un pequeño número de artefactos mientras que OPENUP es más completo en cuanto a ellos, es por eso que se hizo un análisis detallado de los tres métodos que se encuentra en el cuadro 23, para así lograr que MAF integre los mejores artefactos y cumpla con todo lo que se necesita para al final tener un método ágil que contenga los artefactos necesarios y así permitir seguir un marco de referencia para el desarrollo de software, cuya finalidad sea combinar la flexibilidad propuesta por el manifiesto ágil con la documentación necesaria y apropiada para planear y soportar el desarrollo.

A partir de esto y de los artefactos vistos y que son usados por XP, SCRUM y OPENUP, MAF usará la siguiente lista de artefactos:

Product backlog.  
Historias de usuario.  
Sprint backlog.  
Libro de arquitectura y diseño.  
Prueba de desarrollo.

---

<sup>81</sup> Pressman, R. (1998). Ingeniería del software: Un enfoque práctico . México: McGraw-Hill.

Documentación del producto.  
Documentación del usuario.

Los tres primeros artefactos que se muestran a continuación hacen parte de la fase de planificación que será explicada en detalle próximamente, se debe aclarar que para esta parte del proyecto se usaran artefactos de XP y SCRUM, con el fin de lograr que MAF tenga un mejor orden y además de esto la planificación y desarrollo de los requerimientos sea más precisa, para ser más claros se puede decir que los requerimientos del sistema son listados en el product backlog, los proyectos se realizan en una serie de ITERACIONES o SPRINTS, al product backlog se le asigna una prioridad y se seleccionan las tareas que se desarrollaran en cada sprint, luego de crear el product backlog se describirá cada requerimiento tras realizar esto las tareas son removidas al sprint backlog.

Un Sprint es un bloque de tiempo (time-box) de un mes o menos durante el cual se crea un incremento de producto “Terminado”, utilizable y potencialmente desplegable. Es más conveniente si la duración de los Sprints es consistente a lo largo del esfuerzo de desarrollo. Cada nuevo Sprint comienza inmediatamente después de la finalización del Sprint previo.<sup>82</sup>

**Product backlog.** Es la lista de todo el trabajo que se desea realizar en el proyecto. Esta lista es priorizada por el cliente y puede ser actualizada y revisada cuando sea necesario. Se decide usar este artefacto ya que complementa las historias de usuario y aporta mayor organización al momento de desarrollar el proyecto.

A diferencia de SCRUM solo se planteara una lista de requisitos para después continuar con las historias de usuario. El cuadro 24 muestra la plantilla que debe ser usada al momento de realizar el product backlog,

**Cuadro 24.** Plantilla del product backlog

ID-req	Requerimientos	Prioridad	Fecha de entrega

**Fuente.** Autor del proyecto

**Historias de usuario.** Las historias de usuario son escritas por el cliente, en ellas se hace un análisis detallado de lo que el sistema debe realizar, cada uno de los requisitos propuestos en el product backlog será especificado en las historias de usuario.

Las historias de usuarios deben poder ser programadas en un tiempo entre una y tres semanas. Si la estimación es superior a tres semanas, debe ser dividida en dos o más historias. Si es menos de una semana, se debe combinar con otra historia.

---

<sup>82</sup> Disponible en: <http://www.scrumguides.org/index.html>



A continuación en el cuadro 25 podemos ver el formato de las historias de usuario que será utilizado por MAF:<sup>83</sup>

**Cuadro 25.** Formato para historias de usuario

Historia de Usuario	
ID-his	Nombre de historia de usuario:
Modificación (o extensión) de Historia de Usuario (Nro y Nombre):	
Iteración o sprint:	
Usuario:	
Prioridad en Negocio: (Alta/ Media/ Baja)	
Descripción:	
Observaciones:	

**Fuente.** Autor del proyecto

Cuenta con un ID-his de formato que es el ID de la plantilla, además de eso la plantilla debe tener un nombre de la historia de usuario que será establecido por el cliente, un campo llamado usuario que es el respectivo nombre del cliente, como se ha dicho anteriormente cada adelanto es una iteración por lo tanto esta plantilla debe llevar establecido la iteración a la cual pertenece la historia que se está desarrollando, se debe dar una prioridad con respecto al requerimiento que se hará dependiendo su importancia en el proyecto.

Para finalizar se describe el requerimiento detalladamente y se hacen las observaciones necesarias para que todo se claro tanto para el cliente como para el equipo.

Las Historias de Usuario aparte de llevar su propia plantilla posee tres aspectos que van en conjunto con ellas, al realizar una historia de usuario se deben realizar también estos tres aspectos:

**Tarjeta.** En ella se almacena suficiente información para identificar y detallar la historia.

**Conversación.** Cliente y programadores discuten la historia para ampliar los detalles (verbalmente cuando sea posible, pero documentada cuando se requiera confirmación)

**Pruebas de Aceptación.** Permite confirmar que la historia ha sido implementada correctamente.

Cada prueba de aceptación para comprobar que la realización de cada historia de usuario a sido exitosa usa una plantilla que se detalla a continuación en la cuadro 26:

---

<sup>83</sup> Artefactos, disponible en internet en: <http://www.abacovirtual.edu.pe/chiclayo/filedocente/140318-CEI2012203MB-000029-19072012-173247.pdf>

**Cuadro 26.** Formato caso prueba de aceptación

Caso de Prueba de Aceptación	
Código:	Historia de Usuario (ID-his y Nombre):
Modificación (o extensión) de	
Descripción:	
Condiciones de Ejecución:	
Entrada / pasos de ejecución:	
Resultado Esperado:	
Evaluación de la Prueba:	

**Fuente.** Autor del proyecto

Debe contar con un código que la identifica además de eso lleva el nombre y el ID de la historia de usuario a la cual se está evaluando, debe llevar un campo llamado modificación o extensión de donde se coloca el requisito de la historia de usuario, cuenta con unas condiciones de ejecución que son las que se deben cumplir para que el sistema se realice, además debe llevar unos pasos de ejecución que detallen las condiciones anteriores, por último se da el resultado de la prueba y la evaluación final.

Las task card se usan para describir las tareas que se realizan sobre el proyecto. Las tareas pueden ser de: desarrollo, corrección, mejoras, etc. Estas tareas tienen relación con una historia de usuario; se especifica la fecha de inicio y de fin de la tarea, se nombra al desarrollador responsable de cumplirla y se describe que se tratara de hacer en la tarea.

Estas plantillas deben llevar un ID que identifique la tarea, el ID y nombre de la historia de usuario de la cual se están describiendo las tareas, se le asigna el nombre a la tarea y el tipo de tarea que es si es de desarrollo, una corrección, una mejora, si es de otro tipo de especificarse, se deben asignar puntos estimados dependiendo de la prioridad de la tarea, se le asigna una fecha de inicio y una fecha de fin, se coloca el nombre del miembro del equipo que debe realizarla y se hace una descripción de la misma.

A continuación se muestra la plantilla que se necesita para realizar las tareas:

**Cuadro 27.** Task card (Tareas de ingeniería)

Tarea de Ingeniería	
Numero Tarea:	Historia de Usuario (ID-his y Nombre):
Nombre de Tarea:	
Tipo de Tarea:	Puntos Estimados:
Desarrollo / Corrección / Mejora / Otra (especificar)	Fecha Fin:
Fecha de inicio:	
Desarrollador Responsable:	
Descripción:	

**Fuente.** Autor del proyecto

El análisis de requerimientos es una parte crucial de un producto de software es la primera etapa para crearlo, aunque el cliente sabe lo que quiere debe tenerse experiencia para saber que requerimientos están incompletos o son contradictorios. Estos requisitos deben especificarse y con esto se describe el comportamiento que debe tener el software una vez este desarrollado, gran parte del éxito del proyecto depende de la identificación de las necesidades del negocio y la interacción con los clientes. Entre las técnicas utilizadas para la recolección de requisitos están los casos de uso y las historias de usuario siendo los primeros más rigurosos y formales y los segundos más ágiles e informales.<sup>84</sup>

Las historias de usuario colaboran a un ambiente ágil son la evolución de los casos de uso ya que buscan reducir la documentación y fomentar el trabajo en equipo, las historias de usuario apoyan un número de principios del manifiesto ágil:<sup>85</sup>

Se debe aceptar que los requisitos cambian en cualquier momento, incluso en la etapa final de desarrollo, esto como ventaja competitiva del cliente.

Tanto el cliente como los desarrolladores tienen que trabajar en conjunto diariamente durante el desarrollo del proyecto.

La manera rápida de transmitir la información es mediante la conversación cara a cara.

Los requerimientos Ágiles son representados como “Historias de Usuario” (Userstories), la cual es una pequeña descripción del requerimiento del usuario descrito en un lenguaje lo más aproximado a sus jerga diaria. Se hace un énfasis muy fuerte a que este tipo de comunicación esté orientada al usuario final y que sea fácil de entender. Estas historias definen alcances fáciles de identificar para poder efectuar planes iterativos para definición, revisión y confirmación de alcances. No se les exige a los usuarios finales que tengan que conocer desde un principio todos sus requerimientos. Sin embargo, sí se pide que no se cambien en los ciclos iterativos (sprints) que se definen para el desarrollo del producto.<sup>86</sup> A diferencia de los métodos tradicionales, como el modelo en cascada que exige la aceptación de alcances previamente definidos a través de documentos como “Casos de Uso”, en el que se hacen referencias técnicas a partir de los requerimientos del usuario, esto quiere decir que los métodos ágiles son más flexibles en cuanto especificación de requerimientos, se acercan mucho más a lo que el cliente pide y lo integran al grupo de desarrollo, mientras que los métodos tradicionales que solo involucran a los analistas y no dan prioridad a cada punto que tenga el cliente.<sup>87</sup>

En el cuadro 28 se pueden ver las diferencias que hay entre las historias de usuario y los casos de uso:<sup>88</sup>

---

<sup>84</sup> Pressman, R. (2002). Ingeniería del software: Un enfoque práctico. México: McGraw-Hill.

<sup>85</sup> Figuerola, N. (2012). Definiendo requerimientos: Tradicional vs. casos de uso vs. historias de usuario. 2012, de Buenos Aires Sitio web: <https://articulosit.files.wordpress.com/2012/04/requerimientos.pdf>

<sup>86</sup> Velasquez, J. Desarrollo en Cascada (Waterfall) VS Desarrollo Agile-SCRUM. Disponible en: <http://www.northware.mx/wp-content/uploads/2013/04/Desarrollo-cascada-vs-Desarrollo-Agile.pdf>

<sup>87</sup> Kendall, K; Kendall, J (2011). Análisis y diseño de sistemas. México: Pearson educación.

<sup>88</sup> Historias de usuario vs casos de uso disponible en internet en : <http://www.genbetadev.com/metodologias-de-programacion/historias-de-usuario-una-forma-natural-de-analisis-funcional>

**Cuadro 28.** Historias de usuario Vs casos de uso

<b>HISTORIAS DE USUARIO VS CASOS DE USO</b>		
<b>Concepto</b>	<b>Casos de uso</b>	<b>Historia de usuario</b>
<b>Objetivo</b>	Establecer la interacción entre un actor y el sistema.	Escribir una descripción breve del sistema tal y como la percibe el cliente.
<b>Estructura</b>	Usa una plantilla específica predefinida donde se deben definir una serie de conceptos técnicos. (Objetivo, resumen, actor, evento, disparador, extensiones, etc.)	La estructura es breve, se realiza en una o dos fases y es escrita en el lenguaje del usuario.
<b>Planificación</b>	No se utilizan para planificar	Se utilizan para planificar.
<b>Agilidad</b>	Estos requieren mayor tiempo para primero hacer un análisis y luego redactar las plantillas predefinidas.	Se pueden escribir en pocos minutos.
<b>Comprensión</b>	Suelen ser de difícil comprensión, incluso para personal técnico.	Fáciles de leer y comprender.
<b>Mantenimiento</b>	Los documentos poseen cientos de páginas lo cual hace que sean difíciles de mantener.	Muy fáciles de mantener.
<b>Comunicación</b>	Todo debe ir escrito y plasmado en diagramas.	Basada a la comunicación verbal y orientada a la comunicación.
<b>Soporte</b>	Se escriben en documentos para tener referencia su objetivo es ser archivados.	Escritas en tarjetas (teóricas o reales) con el objetivo de que sean usadas directamente.
<b>Duración</b>	Pueden ser implementados en varias iteraciones.	Se implementa y se prueba en una sola iteración.
<b>Autores</b>	Definidos por “interpretes” (analistas, consultores, etc)	Posibilidad de ser definidas por usuarios y clientes.
<b>Pruebas</b>	Cada prueba debe ser redactada en documentos separados.	Contienen pruebas de aceptación en el reverso de las tarjetas.

Cuadro 28. (Continuación)

<b>Contexto</b>	Suministra una visión general de todo el sistema y la integración en el.	La visión es menos obvia por eso se hacen pruebas y hay retroalimentación de los clientes, es por ello que estos son tan importantes en los métodos ágiles.
<b>Metodología</b>	Asociado con RUP	Asociado con programación extrema.

Fuente. Historias de usuario vs casos de uso disponible en internet en: <http://www.genbetadev.com/metodologias-de-programacion/historias-de-usuario-una-forma-natural-de-analisis-funcional>

Con el cuadro 28 se puede observar porque las historias de usuario son mejor elección en cuanto al desarrollo ágil, se especifican claramente las diferencias existentes entre ellos.

**Sprint Backlog:** es el conjunto de elementos del product backlog seleccionados para el desarrollo de cada sprint, más un plan para entregar el Incremento de producto y conseguir el Objetivo del Sprint. El sprint backlog describe una lista detallada de tareas para el desarrollo de cada requerimiento, debe llevar el ID que identifica la historia de usuario que contiene el requerimiento al que se le van a asignar la lista de tareas para completar el sprint, un estado para cada tarea que varía entre (No iniciada/ en progreso/ completada) y el sprint o la iteración a la cual pertenece el desarrollo de esas actividades.

El cuadro 29 muestra la plantilla que se necesita para el desarrollo de este artefacto:

**Cuadro 29.** Plantilla sprint backlog

ID-his	Tareas	Estado	# Sprint o iteración
1			
2			
3			

**Fuente.** Autor del proyecto

Al evaluar en el capítulo anterior las similitudes entre cada método y aquello que tenían en común se pudo notar que en cuanto artefactos existe un gran vacío en XP y SCRUM, ya que características importantes del producto como la arquitectura no son evaluadas detalladamente en sus artefactos, OPENUP es un poco más completo con respecto a esto y por eso que se decide usar los siguientes artefactos que son responsables del manejo de la arquitectura y el diseño, llevan algunas modificaciones para poder ser adoptado por MAF.

**Libro de arquitectura y diseño:**<sup>89</sup> Este artefacto describe la arquitectura y el diseño del sistema. Proporciona una lista donde se observan los problemas que pueda presentar la

<sup>89</sup> Eclipse. (Febrero de 2015). OpenUp. Obtenido de <http://epf.eclipse.org/wikis/openup/>

arquitectura y las decisiones que se han tomado con respecto a diseños, modelos y códigos. Este artefacto debe informar a los miembros del equipo como se organizan para que ellos sepan cómo adaptarse a las necesidades del sistema. Este artefacto describe la realización de la funcionalidad que el sistema requiere y sirve como una abstracción del código fuente. Este producto describe los elementos que compondrán el sistema al implementarse. Mientras que la arquitectura se centra en las interfaces, los patrones y las decisiones clave, el diseño da cuerpo a los detalles técnicos en preparación para la ejecución, o como parte de la implementación. Para realizar este artefacto los miembros del equipo deben tener en cuenta la siguiente plantilla (cuadro 30), es detallada y hará que el equipo mantenga un orden y una documentación específica por artefacto.

**Cuadro 30.** Plantilla libro de arquitectura

<b>Libro de arquitectura</b>	
<b>Nombre del proyecto:</b>	
<b>Objeto:</b>	<b>Objetivos de la arquitectura:</b>
Supuestos y dependencias:	
Requisitos de mayor valor arquitectónico:	
Decisiones, limitaciones y justificaciones:	
Mecanismos de arquitectura:	
Abstracciones clave:	
Capas o marco arquitectónico:	
Vistas de arquitectura:	

**Fuente.** Autor del proyecto

A continuación en el cuadro 31 se describe detalladamente la plantilla anterior que es el artefacto libro de arquitectura y diseño:

**Cuadro 31.** Libro de arquitectura y diseño<sup>90</sup>

<Nombre del proyecto>	
<b>Objeto</b>	Este documento describe la filosofía, las decisiones, las limitaciones, las justificaciones, elementos importantes, y cualquier otro aspecto general del sistema que dan forma al diseño e implementación.
<b>Objetivos de la arquitectura</b>	Describe la filosofía de la arquitectura, se deben identificar los problemas que impulsaran esta filosofía, tal y como son: ¿El sistema será impulsado por preocupaciones de despliegue complejos,

<sup>90</sup> Eclipse. (Febrero de 2015). OpenUp. Obtenido de <http://epf.eclipse.org/wikis/openup/>

Cuadro 31. (Continuación)

	<p>adaptándose a los sistemas de legado, o problemas de rendimiento? ¿Tiene que ser robusto para el mantenimiento a largo plazo? Se debe formular un conjunto de objetivos que la arquitectura tiene que cumplir en su estructura y comportamiento.</p> <p>Identificar los temas críticos que deben ser abordados por la arquitectura, por ejemplo: ¿Existen dependencias de hardware que deben ser aislados del resto del sistema? ¿Es necesario que el sistema funcione de manera eficiente bajo condiciones inusuales?</p>
<b>Supuestos y dependencias</b>	<p>Se debe enumerar los supuestos y dependencias que impulsan las decisiones arquitectónicas. Esto podría incluir zonas sensibles o críticos, dependencias en las interfaces heredadas, la habilidad y la experiencia del equipo, la disponibilidad de recursos importantes, etcétera</p>
<b>Requisitos de mayor valor arquitectónico</b>	<p>Se referencian todos los requisitos que se deben implementar para realizar la arquitectura.</p>
<b>Decisiones, limitaciones y justificaciones.</b>	<p>Se debe enumerar las decisiones que se han tomado con respecto a los enfoques arquitectónicos y las restricciones que tienen los desarrolladores en cuanto a construir el sistema. Esto servirá como directrices para la definición de las piezas de gran importancia arquitectónica del sistema. Justificar cada decisión o restricción para que los desarrolladores puedan entender la importancia de la construcción del sistema de acuerdo con el contexto creado por esas decisiones y limitaciones.</p>
<b>Mecanismos de arquitectura</b>	<p>Se enumeran los mecanismos arquitectónicos y se describe el estado actual de cada uno. Inicialmente, cada mecanismo puede ser sólo el nombre y una breve descripción. Ellos evolucionan hasta que el mecanismo es una colaboración o patrón que se puede aplicar directamente a algún aspecto del diseño.</p>
<b>Abstracciones clave</b>	<p>Se hace una lista y se describe brevemente las abstracciones clave del sistema. Esto debería ser una lista relativamente corta de los</p>

Cuadro 31. (Continuación)

	<p>conceptos fundamentales que definen el sistema. Las abstracciones clave suelen traducir a las clases de análisis iniciales y patrones importantes</p>
<b>Capas o marco arquitectónico</b>	<p>Se describe el patrón de arquitectura que va a utilizar o cómo la arquitectura será consistente y uniforme. Esto podría ser una simple referencia a una ya existente o bien conocido patrón arquitectónico, como el marco de capa, una referencia a un modelo de alto nivel de la estructura, o una descripción de cómo los principales componentes del sistema deben ser puestos juntos</p>
<b>Vistas de arquitectura</b>	<p>Describe las vistas arquitectónicas que va a utilizar para describir la arquitectura de software. Esto sirve para ilustrar las diferentes perspectivas en las que se va a revisar y documentar las decisiones arquitectónicas.</p> <p>Vistas recomendadas</p> <ul style="list-style-type: none"> <li>• Lógicas: describe la estructura y el comportamiento de las partes arquitectónicamente significativos del sistema. Esto podría incluir la estructura de paquetes, interfaces críticas, clases y subsistemas importantes, y las relaciones entre estos elementos. También incluye vistas físicas y lógicas de datos persistentes, si la persistencia se construirá en el sistema. Este es un subconjunto documentado del diseño.</li> <li>• Operativas: Describe los nodos físicos del sistema y de los procesos, subprocesos y componentes que se ejecutan en los nodos físicos. Este punto de vista no es necesario si el sistema se ejecuta en un proceso único.</li> <li>• Product backlog: Una lista que contienen arquitectónicamente los requerimientos más significativos, acompañado de las historias de usuario.</li> </ul>

**Fuente.** Artefacto libro de arquitectura. Disponible en: <http://epf.eclipse.org/wikis/openup/>

Para los escenarios mencionados en los mecanismos de arquitectura y diseño se sugiere al grupo de desarrollo usar alguno de los siguientes diagramas ya que permiten ver con



facilidad las tareas que se están desarrollando además de representar los mensajes en función del tiempo.<sup>91</sup>

**Diagrama de secuencia:** son aquellos que pueden ilustrar una sucesión de interacciones entre clases o instancias de objetos en un periodo determinado. Ponen énfasis en la clasificación de los mensajes según el tiempo.

**Diagrama de clases:** muestran las características estáticas del sistema y no representan ningún procesamiento en particular. Un diagrama de clases también muestra la naturaleza de relaciones entre las clases.

**Diagrama de despliegue:** Un diagrama de despliegue es la forma de mostrar la configuración de nodos de procesamientos en tiempo de ejecución y los componentes que en ellos residen. Estos nodos forman la topología de hardware sobre el que se ejecuta el sistema. Este diagrama se preocupa principalmente de la distribución, entrega e instalación de las partes que constituye el sistema físico.

Los artefactos anteriores se utilizan para establecer los requisitos del sistema, fijar prioridades y aclarar lo que se desea hacer durante el proyecto, así como establecer la arquitectura y diseño del sistema.

A continuación se definirán los artefactos necesarios para implementar las pruebas al sistema y que conste que en cada iteración hecha se ha cumplido con lo que pidió el cliente.

**Prueba de desarrollo:** esta prueba será ejecutada en cada sprint o iteración y al final del proyecto con el fin de saber si los requisitos propuestos han sido cumplidos y de que no existe ningún error en ellos.

Para la ejecución de este artefacto se usa la plantilla correspondiente al cuadro 32, la cual debe llevar el número que identifique esta prueba, la iteración que se está evaluando y su descripción, el resultado que se obtuvo y por último el estado que puede ser aprobada o no aprobada.

**Cuadro 32.** Plantilla prueba de desarrollo

<b>Prueba de desarrollo</b>
Numero de prueba:
ID-his
Iteración o #sprint:
Descripción:
Resultado:
Estado (aprobada, no aprobada):

**Fuente.** Autor del proyecto

---

<sup>91</sup> Kendall & Kendall. (2005). Análisis y diseño de sistemas. México: Pearson educación.

Por ultimo en cuanto a artefactos se debe documentar y dar al cliente información sobre el producto y de cómo se utiliza para esto se usaran los siguientes artefactos.

**Documentación del producto.** Este artefacto proporciona documentación sobre el producto para el propietario. Se crea la documentación del producto para el beneficio de la división de marketing de una organización, el director del programa, y aquellas personas que deben evaluar el valor comercial de un sistema en particular. Este es un aspecto a menudo pasado por alto de ejecución equipo de desarrollo. El equipo debe tener en cuenta que los clientes de un lanzamiento en particular por lo general no son técnicos propios, pero sí requieren un conocimiento bastante detallado de cómo funciona su producto y cómo cumple declararon metas y necesidades del negocio. Este documento debe contener como mínimo los siguientes ítems:

- Información detallada del producto
- Requerimientos del sistema
- Tipo de instalación
- Lenguaje de programación en que fue realizado
- Herramienta de desarrollo web
- Motor de base de datos
- Sistema operativo que se uso
- Servidores de aplicaciones
- Sistemas soportados
- Tipo de soporte

**Documentación del usuario:** Son los documentos que pueden ser utilizados por los usuarios finales de un sistema o producto en particular. Este tipo de documentación normalmente está escrita de una manera que permite a los usuarios del sistema encontrar fácilmente la información que necesitan para utilizar el producto. Como lo son manuales de usuario, Tutoriales, Ayuda en línea e Instrucciones de instalación. Estos documentos deben llevar información tal y como:

- Introducción
- Objetivos del sistema
- Guía de uso
- Sección de solución de problema
- E-mail o teléfonos para soporte técnico

La figura 14 hace referencia a la organización de los artefactos dividido en tres partes, requerimientos, pruebas y documentación final.

**Figura 14.** Artefactos MAF



**Fuente.** Autor del proyecto

A continuación el cuadro 33 especifica que rol desarrolla cada artefacto esto con el fin de orientar a los miembros del equipo, para que sepan que actividades les corresponde desarrollar. A la izquierda se define el artefacto que se desarrollara y a la derecha el rol que será responsable de ejecutarlo.

**Cuadro 33.** Mapeo Artefactos-rol

ARTEFACTO	ROL
Product backlog	Cliente
Historia de usuario	Cliente Delegado
Sprint backlog	Cliente Delegado Desarrollador
Libro de arquitectura	Desarrollador
Diseño	Desarrollador
Prueba de desarrollo	Tester
Documentación del producto	Desarrollador Tester Facilitador
Documentación del usuario	Desarrollador Tester Facilitador

**Fuente.** Autor del proyecto

**Ciclo de vida MAF.** Todo proyecto de desarrollo ágil de software debe tener un ciclo de vida que especifique los pasos y fases que se deben seguir para mantener un orden, XP, SCRUM y OPENUP usan fases que tienen mucho en común como se pudo ver en el capítulo anterior, cada una a su manera se divide en diferentes partes pero al final realizan las mismas actividades, a partir de eso MAF utilizara las siguientes pasos para su desarrollo:

**Fase de planificación.** esta es la fase inicial de todo el proceso de desarrollo en ella se establecen los requisitos del usuario, el cliente expone todo lo que quiere en su producto y se crea el product backlog, el cliente es el encargado de dar prioridad a las historias de usuario, al mismo tiempo se estiman costos. De igual manera el equipo de desarrollo debe familiarizarse con las herramientas y las tecnologías que serán usadas en el proyecto. El grupo de trabajo hace pruebas de la tecnología y al mismo tiempo se examinan las posibilidades de la arquitectura del sistema realizando un prototipo. Se le asigna a cada miembro del equipo la tarea que debe realizar en la realización de cada historia de usuario. Al final de esta fase se debe tener claro y preciso los requisitos, los actores y la arquitectura del sistema.

**Fase de desarrollo.** En esta fase en la primera entrega se incluyen varias iteraciones, la primera iteración crea la arquitectura del sistema, para esto se deben buscar las historias de usuario que cumplan con esos requisitos, el encargado de esto será el cliente. Se pueden presentar cambios y se debe decidir si serán incluidos o no, los cambios en las iteraciones deben ser ejecutados de forma rápida sin que se pierda la calidad. Se deben realizar, integrar y probar todos los componentes y funcionalidades que estén pendientes por probar antes de hacer alguna entrega al cliente. En esta fase se hacen encuentros diarios para analizar el avance del proyecto.

**Fase de culminación.** Esta es la fase de liberación del producto y realización de documentación. Ya no hay más historias de usuario para incluir por lo cual se deben mirar otros aspectos como los son el rendimiento y la confiabilidad del sistema. Consta de sub fases de pruebas y capacitación a los usuarios finales y de los encargados del mantenimiento del sistema. Se genera documentación y no se realizan más cambios en la arquitectura.

La figura 15 nos resume el ciclo de vida de MAF, en ella se ve cada fase y sus respectivas actividades:

**Figura 15.** Fases de MAF



**Fuente.** Autor del proyecto

Después de definir el ciclo de vida y las actividades que se realizan en cada una de sus fases y con el fin de orientar a todos aquellos que deseen implementar MAF, se hará un mapeo entre artefactos y ciclo de vida, el cuadro 34 a su izquierda tiene cada fase del ciclo de vida MAF y a la derecha define que artefacto será utilizado en cada una de ellas.

**Cuadro 34.** Mapeo Fase-Artefacto

FASE DEL CICLO DE VIDA	ARTEFACTO
Planificación	Product backlog Historias de usuario
Desarrollo	Sprint backlog Libro de arquitectura diseño Prueba de desarrollo
Culminación	Documentación del producto Documentación del usuario

**Fuente.** Autor del proyecto

**Administración.** MAF es un método iterativo e incremental por lo cual se deben ir desarrollando iteraciones o sprints y tareas de manera rápida, el cliente al hacer parte del equipo de trabajo y en conjunto con el delegado deben establecer los requerimientos del sistema para completar el product backlog el cliente debe asignarle una prioridad a cada uno de esos requerimientos, esta lista también contara con una fecha de entrega que se establece dependiendo de la prioridad que tenga, estos requerimientos son exportados a las historias de usuario donde ese hace una descripción detallada del requerimiento expuesto, después de esto se crea el sprint backlog en donde a partir de la descripción hecha en la historia de usuario se especifican las tareas necesarias para completar las iteraciones cada iteración llevara incrementos y mejoras solo se pasara a la siguiente iteración cuando el estado de la prueba de desarrollo sea aprobado.

En el transcurso del desarrollo del sprint backlog se deben hacer reuniones diarias entre el equipo para saber si se han cumplido con las tareas establecidas para la iteración que se está desarrollando, dar aportes y resolver dudas que surjan, debe haber una retroalimentación constante para así lograr obtener lo que se quiere en el tiempo establecido, por esto el equipo trabajar 40 horas semanales de esa manera lograra cumplir los objetivos.

Cuando el resultado de la prueba de desarrollo sea Aprobado se podrá seguir con la siguiente iteración hasta tener un producto listo para la entrega.

Los grupos de trabajo no deben superar las 10 personas ya que el método está desarrollado para proyectos pequeños y medianos.

#### 4.4 EVALUACIÓN DEL MÉTODO ÁGIL UTILIZANDO LA HERRAMIENTA DE COMPARACIÓN Y ANÁLISIS 4-DAT

Este capítulo presenta una descripción de lo que es la herramienta de análisis y comparación 4-DAT, a continuación de esto se realizara un análisis detallado del método MAF con la herramienta Dimensional Analítica (4-DAT), basado en cuatro perspectivas de caracterización: el alcance, la agilidad, los valores ágiles y el proceso de software. Un informe generado con la ayuda de 4-DAT ayudará a comprobar si cumple con los criterios que tienen los métodos ágiles.

**Herramienta analítica 4-DAT.** 4-DAT fue desarrollado por investigadores y profesionales para fines de análisis y comparación de los métodos ágiles. 4-DAT tiene cuatro dimensiones que permiten los criterios de evaluación para la evaluación detallada de los métodos ágiles de desarrollo de software desde diferentes perspectivas. La herramienta 4-DAT es una herramienta analítica que sirve para medir la agilidad y adaptabilidad de los métodos ágiles de desarrollo de software, es utilizada por las organizaciones para tomar decisiones acerca de que método debe usas o adoptar.<sup>92</sup>

4-DAT facilita el análisis y comparación de métodos ágiles desde cuatro perspectivas:

- Caracterización del alcance
- Caracterización de la agilidad
- Caracterización de procesos de software
- Caracterización de los valores ágiles

**Caracterización del alcance (primera dimensión)<sup>93</sup>.** El cuadro 35 hace referencia a la primera dimensión en esta se ofrece un conjunto de elementos clave con los que los métodos se pueden comparar en un alto nivel. En ella los métodos deben trabajar en factores como tamaños en proyectos y equipos, estilos de desarrollo y la codificación, tecnología y entornos físicos.

**Cuadro 35.** Primera dimensión 4-DAT

<b>Alcance</b>	<b>Descripción</b>
Tamaño del proyecto	¿El método especifica el apoyo a proyectos pequeños, medianos o grandes (comerciales o de otro tipo)?
Tamaño del equipo	¿Es compatible el método para pequeños o grandes equipos (equipos individuales o múltiples)?
Estilo de desarrollo	¿Qué estilo de desarrollo (iterativo, rápido) cubre el método?
Estilo del código	¿El método especifica estilo de código (simple o

<sup>92</sup> Leusink B. Agile software development process improvement in large organizations. (2012)

<sup>93</sup> Qumer A, Hendsen-Sellers B. Measuring agility and adptability of agile methods: A 4-Dimensional Analytical tool. IADIS international conference Applied Computing (2006).

Cuadro 35. (Continuación)

	complejo)?
Tecnología del medio ambiente	¿Qué ambiente de tecnología (herramientas, compiladores) especifica el método?
Medio físico	¿Qué ambiente físico (co-localizados o distribuido) especifica el método?
Cultura empresarial	¿Qué tipo de cultura empresarial (colaborativa, cooperativa o de no colaboración) ha especificado el método?
Mecanismo de abstracción	¿El método especifica mecanismo de abstracción (orientado a agente u orientado a objetos)?

**Fuente.** Qumer A, Hendeson-Sellers B. Measuring agility and adptability of agile methods: A 4-Dimensional Analytical tool. IADIS international conference Applied Computing (2006).

**Caracterización de la agilidad (segunda dimensión)<sup>94</sup>.** El cuadro 36 hace referencia a la segunda dimensión en la cual se proporcionan una serie de características que miden la agilidad. Puede ser utilizada para comprobar la agilidad tanto a a nivel de proceso como a nivel de práctica del método. Esta es la única dimensión en 4-DAT que es cuantitativa.

**Cuadro 36.** Segunda dimensión 4-DAT

Características	Descripción
Flexibilidad	¿El método es capaz de adaptarse a los cambios esperados o inesperados?
Velocidad	¿El método produce resultados rápidamente?
Delgadez	¿El método hace seguimiento en un corto lapso de tiempo, usa instrumentos económicos, sencillos y de calidad para la producción?
Aprendizaje	¿El método utiliza el conocimiento previo y actualizado para aprender?
Capacidad de respuesta	¿El método tiene sensibilidad de exposición?

**Fuente.** Qumer A, Hendeson-Sellers B. Measuring agility and adptability of agile methods: A 4-Dimensional Analytical tool. IADIS international conference Applied Computing (2006).

**Caracterización de los valores ágiles (tercera dimensión).** El cuadro 37 muestra la tercera dimensión en esta se determina mediante una búsqueda las prácticas que apoyan los valores descrito en el Manifiesto Ágil, así como las prácticas para mantener el proceso ágil y económico.

**Cuadro 37.** Tercera dimensión 4-DAT

Valores ágiles	Descripción
Individuos e interacciones sobre procesos y herramientas	¿Valora las prácticas de individuos e iteraciones sobre procesos y herramientas?
Trabajo de software sobre una amplia documentación.	¿Qué prácticas se valoran más el trabajo de software o la documentación excesiva?
Colaboración con el cliente sobre negociación del contrato.	¿Se valora más el cliente sobre el contrato de negociación?
Respuesta al cambio sobre seguimiento de un plan	¿Se valora más la respuesta al cambio sobre el seguimiento de un plan?
Mantener el proceso ágil	¿Qué prácticas ayudan a mantener el proceso ágil?
Mantener el costo efectivo en el proceso	¿Qué prácticas ayudan a mantener el costo?

**Fuente.** Adaptado de: Qumer A, Hendeson-Sellers B. Measuring agility and adptability of agile methods: A 4-Dimensional Analytical tool. IADIS international conference Applied Computing (2006).

**Caracterización de los procesos de software (cuarta dimensión).** El cuadro 38 hace referencia a la cuarta dimensión la cual examina las prácticas que apoyan los procesos de ingeniería.

**Cuadro 38.** Cuarta dimensión 4-DAT

Proceso	Descripción
Proceso de desarrollo	¿Qué prácticas cubren el ciclo de vida principal y las pruebas?
Proceso de gestión de proyectos	¿Qué prácticas abarcan la gestión global del proyecto?
Control de la Configuración de Software Proceso / Proceso de Apoyo	¿Qué prácticas cubren el proceso que permite la gestión de la configuración?
Gestión de procesos	¿Qué prácticas cubren el proceso que se requiere para gestionar el proceso de sí mismo?

**Fuente.** Adaptado de: Qumer A, Hendeson-Sellers B. Measuring agility and adptability of agile methods: A 4-Dimensional Analytical tool. IADIS international conference Applied Computing (2006).



Una vez que se describe lo que es la herramienta de evaluación y comparación de 4-DAT se procede a hacer la evaluación de MAF, comparada con los métodos utilizados para su creación, además de esto se realizó una encuesta (Anexo C), en donde se preguntó a los líderes desarrolladores de software de la UFPSO sobre el manejo de sus proyectos con el fin de hacer una alineación entre el método creado y los usados en la universidad es una caracterización general por esto solo se hizo a los líderes y no a cada integrante de los grupos.

Dicha encuesta extrajo algunas características de valores planteados para la evaluación con 4-DAT, esto con el fin de saber si MAF y los métodos usados en la universidad, tienen relación, los resultados se encuentran en el apartado Alineación de MAF con la UFPSO (pág. 113). A continuación se verán reflejados cada uno de los cuadros que utiliza 4-DAT en cada una de sus dimensiones con sus valores correspondientes:

#### 4.4.1 Evaluación de MAF con 4-DAT.<sup>95</sup>

**Alcance del método (primera dimensión).** El siguiente cuadro muestra la primera comparación que se refiere a la caracterización del alcance del método

**Cuadro 39.** Evaluación y comparación con la primera dimensión

<b>Criterio</b>	<b>XP</b>	<b>SCRUM</b>	<b>OPENUP</b>	<b>MAF</b>
Tamaño de los proyectos	Pequeños y medianos.	Pequeños, medianos y grandes.	Pequeños.	Pequeños y medianos.
Tamaño de equipo	Menor que 10	Múltiples equipos menores que 10	3 a 6 personas.	Menor que 10
Estilo de desarrollo	Iterativo e incremental	Iterativo e incremental	Iterativo e incremental	Iterativo e incremental
Estilo de código	Limpio, sencillo y en parejas	No especificado	No especificado	No especificado
Entorno tecnológico	Requiere rápida retroalimentación	No especificado	Requiere rápida retroalimentación	Requiere rápida retroalimentación
Entorno físico	Equipos en un mismo lugar y equipos distribuidos	No especificado	No especificado	No especificado
Cultura de negocio	Colaborativo y cooperativo	No especificado	Colaborativo y cooperativo	Colaborativo y cooperativo

<sup>95</sup> Avila, E; Meneses, A. Comparative Evaluation of Delfdroid whit XP and Scrum using the 4-DAT. Rev cuba cienc informat vol.7 no.1 La Habana ene.-mar. 2013

Cuadro 39. (Continuación)

Mecanismo de abstracción	Orientado a objetos	a	Orientado a objetos	Orientado a objetos	a	Orientado a objetos	a
--------------------------	---------------------	---	---------------------	---------------------	---	---------------------	---

**Fuente.** Qumer A, Hendeson-Sellers B. Measuring agility and adptability of agile methods: A 4-Dimensional Analytical tool. IADIS international conference Applied Computing (2006).

**Caracterización de la agilidad (segunda dimensión).** Para lograr entender la manera de que se evalúa MAF con esta dimensión se explica la fórmula que se usa y de igual manera que significan las siglas usadas en cada columna, además de esto se dice que características fueron tomadas de cada método para la evaluación.

**Grado de agilidad (DA)**

Flexibilidad (FY)    Velocidad (SD)    Eficiencia (LS)    Aprendizaje (LG)    Adaptabilidad (RS)

Para calcular la agilidad en esta dimensión Qumer y Henderson-Sellers proponen la siguiente fórmula:  $DA(\text{Object}) = \left(\frac{1}{m}\right) \sum mDA(\text{del inglés, Object, Phase or Practices})^{96}$

**Las características:**

1, 2, 3, 4, 6, 7 y 10  
11  
14, 15 y 16

**Han sido tomadas de:**

XP  
SCRUM  
OPENUP

MAF al ser un híbrido de los métodos ágiles XP, SCRUM y OPENUP posee prácticas que estas últimas también poseen, o al menos una de ellas. Las enumeradas con 1, 2, 3, 4, 6, 7 y 10 han sido tomadas de XP por lo que se han evaluado de igual manera que en Qumer y Henderson (2008a). De forma semejante la enumerada con 11 ha sido evaluada, con la diferencia de que pertenece a SCRUM, mientras que la 14, 15 y 16 hacen parte de OPENUP.

Teniendo en cuenta que al realizar reuniones para controlar el Sprint terminado y planificar el siguiente se realiza un juego de planificación, se puede decir que se enumera con 1 pertenece tanto a XP como a SCRUM. Por otra parte se consideran a las prácticas 6, 9 y 14 se consideran como nativas de MAF.

Dando oportunidad a todo el equipo que participe activamente podrán proponer funcionalidades nuevas al sistema o sencillamente cambios significativos que el delegado no ha sugerido. Esta acción es flexibilidad, veloz, eficaz, el equipo aprende con el aumento de la experiencia y perfectamente posee adaptabilidad por lo que ha evaluado de forma global con la característica 5 (Rápida retroalimentación). Al sacar un producto funcional

<sup>96</sup> Qumer, A., Henderson-Sellers, B.: An evaluation of the degree of agility in six agile methods and its applicability for method engineering. In: Information and Software Technology. (2007)

aunque con un número importante de requisitos por cumplir se poseen criterios de usuarios finales de manera rápida; pudiendo el equipo retroalimentarse a una mayor velocidad.

Por último MAF, considera al cliente parte del equipo y es por esto que es el encargado de establecer los requisitos del sistema y darles prioridad en compañía del delegado.

Es necesario aclarar que 1 hace referencia a que existe y 0 a que no hace parte del método.

A continuación la tabla 3 se detalla la evaluación realizada para la segunda dimensión.

**Tabla 3.** Evaluación y comparación con la segunda dimensión

MAF		Características de la agilidad.					TOTAL
		FY	SD	LS	LG	RS	
I Fases							
1	Planificación	1	1	1	0	1	4
2	Desarrollo	1	1	0	1	1	4
3	Culminación	1	0	1	1	1	4
Total		3	2	2	2	3	12
Grado de agilidad		3/3	2/3	2/3	2/3	3/3	12/(5*3)
II Practicas							
1	Desarrollo iterativo e incremental mediante iteraciones cortas	1	1	1	1	1	5
2	Programación en parejas	0	0	0	0	0	0
3	Pruebas	0	1	1	1	1	4
4	40 horas semanales	1	1	1	1	1	5
5	Rápida retroalimentación	1	1	1	1	1	5
6	Diseño simple	0	1	1	1	1	4
7	Refactorización	0	0	0	0	0	0
8	Participación activa de todos los miembros del producto	1	1	1	1	0	4
9	Reutilización constante	0	0	0	0	0	0
10	Estándar de codificación	0	0	0	0	0	0
11	Reunión para examinar la iteración terminada y planificar la siguiente	1	0	1	1	1	4
12	Reportando progreso	1	1	1	1	1	5
13	Cliente en dentro del equipo	1	0	0	1	0	2
14	Uso de artefactos exclusivos para la arquitectura	0	1	1	1	1	4
15	Uso de artefactos exclusivos para el diseño	1	1	1	1	1	5
16	Documentación específica	1	1	1	1	1	5
Total		9	10	11	12	10	52
Grado de agilidad		9/16	10/16	11/16	12/16	10/16	52/80

**Fuente.** Autor del proyecto

## Valores ágiles (tercera dimensión)

**Cuadro 40.** Evaluación y comparación con la tercera dimensión

VALORES AGILES	XP	SCRUM	OPENUP	MAF
Individuos e iteraciones por encima de procesos y herramientas	*Juego de planificación. *Propiedad colectiva *Cliente en el equipo de desarrollo. *Programación en parejas.	*Scrum teams *Spring planing meeting *Daily Scrum meeting	*Participación activa de todos los miembros del equipo *Reuniones diarias *Administración a tiempo de los riesgos	*Participación activa de todos los miembros del equipo *Cliente en el equipo de desarrollo *Reunión para controlar las iteraciones
Software activo sobre documentación excesiva	*Releases cortos *Pruebas *Integración continua	*Sprint *revisión del sprint	*Prueba de desarrollo *Especificación de casos de uso *Reuniones diarias *Proceso definido del proyecto	*Pruebas frecuentes *Revisión de las iteraciones *Rápida retroalimentación *Reuniones diarias
La colaboración con el cliente más que la negociación de un contrato	*Juego de colaboración *Integración continua	*Product backlog *Sprint planning meeting	*Integración continua *Documentación del usuario	*Product backlog *Participación activa de todos los miembros del equipo
Responder a los cambios más que seguir estrictamente un plan	*Metáforas *Diseño simple *refactorización *estándar de codificación	*Sprint review *Sprint planing meeting	*Comunicación de lanzamiento *Plan de pruebas *Libro de arquitectura	*Diseño simple *Rápida retroalimentación *Reuniones diarias para controlar las iteraciones y los problemas que se presenten.
Mantener procesos ágiles	No especificado	No especificado	No especificado	No especificado

Cuadro 40. (Continuación)

Mantener el costo efectivo del proceso	No especificado	No especificado	No especificado	No especificado
--	-----------------	-----------------	-----------------	-----------------

**Fuente.** Autor del proyecto

**Caracterización del proceso de software (cuarta dimensión).** En esta parte de la evaluación se examinan las prácticas de XP, SCRUM, OPENUP y MAF para el apoyo a los procesos de software, el cuadro 41 muestra esta evaluación. Al igual que en la Dimensión 3, la comparación es puramente cualitativa e informativa (descriptiva y no normativa).

**Cuadro 41.** Evaluación y comparación con la cuarta dimensión

PROCESO DE SOFTWARE	XP	SCRUM	OPENUP	MAF
Proceso de desarrollo	<ul style="list-style-type: none"> <li>*Releases cortos.</li> <li>*Metáforas</li> <li>*Diseño simple.</li> <li>*Pruebas.</li> <li>*Refactorización.</li> <li>*Programación en pares.</li> <li>*Propiedad colectiva.</li> <li>*Integración continua.</li> <li>*Cliente en el equipo de desarrollo.</li> <li>*Estándar de codificación</li> </ul>	<ul style="list-style-type: none"> <li>*Scrum teams</li> <li>*Product backlog</li> <li>*Sprint</li> <li>*Sprint review</li> </ul>	<ul style="list-style-type: none"> <li>*Iteraciones rápidas</li> <li>*Diseño simple</li> <li>*Pruebas de desarrollo</li> <li>*Pruebas de despliegue</li> <li>*Traducción de pruebas a código.</li> </ul>	<ul style="list-style-type: none"> <li>*Product backlog</li> <li>*Historias de usuario</li> <li>*Iteraciones rápidas</li> <li>*Cliente en el equipo de desarrollo</li> </ul>
Proceso de gestión de proyecto	<ul style="list-style-type: none"> <li>*Juego de planificación</li> </ul>	<ul style="list-style-type: none"> <li>*Scrum master</li> <li>*Sprint planning meeting</li> <li>*Daily scrum meeting</li> </ul>	<ul style="list-style-type: none"> <li>*Fase de construcción</li> </ul>	<ul style="list-style-type: none"> <li>*Reunión diaria.</li> <li>*Fase de desarrollo.</li> </ul>
Proceso de la configuración del software	No especificado	No especificado	*Documentación del producto	*Documentación del producto

Cuadro 41. (Continuación)

Proceso de gestión de procesos	No especificado	No especificado	No especificado	No especificado
--------------------------------	-----------------	-----------------	-----------------	-----------------

**Fuente.** Autor del proyecto

Al valorar MAF con 4-DAT se tiene lo siguiente:

La primera dimensión que hace referencia al alcance del método muestra que MAF es iterativo e incremental, maneja grupos pequeños de desarrollo, sigue un estándar de rápida retroalimentación, características esenciales en un método ágil.

La segunda dimensión evalúa la agilidad y lo hace de manera cualitativa al observar la tabla y los resultados que arroja se concluye que MAF cumple con un 80% de los requisitos establecidos.

La tercera dimensión califica los valores ágiles MAF describe una tarea para cada ítem establecido por 4-DAT.

Por último se tiene la dimensión que analiza los procesos de software y al ser comparado con XP, SCRUM y OPENUP se puede ver que MAF cumple y tiene mejores características que los métodos mencionados.

Al realizar la comparación de XP, SCRUM y OPENUP con el método creado MAF y evaluarlo con la herramienta 4-DAT para medir el grado de agilidad de este último, se puede concluir que es un método ágil, que posee características que hacen que logre ser una herramienta ágil en el desarrollo de proyectos de software, posee tres fases y una lista de artefactos que hacen que el método sea flexible, veloz y adaptable para proyectos de gestión de software.

Algunos estudios realizados con esta herramienta muestran que es de gran ayuda para saber si un método posee las características para considerarse ágiles, Delfroid y su comparación evaluativa con XP y Scrum mediante el método 4-DAT<sup>97</sup>, ellos crean un nuevo método ágil a partir de XP y SCRUM llamado Delfroid y lo evalúan con la herramienta de análisis y comparación 4-DAT en su investigación llegan a la siguiente conclusión:

Luego de describir Delfroid se puede concluir que se ha propuesto una metodología con tres fases y cuatro flujos de trabajo principales. Por sus características se considera ágil y adaptable a entornos de desarrollo de aplicaciones para dispositivos móviles. Además agrupa muchas de las llamadas “buenas prácticas de desarrollo de software” empleadas en las más importantes y utilizadas metodologías ágiles encontradas en la literatura.<sup>98</sup>

<sup>97</sup> Avila, E, & Meneses, A. (2013). Delfroid y su comparación evaluativa con XP y Scrum mediante el método 4-DAT. Revista cubana de ciencia, 7, 1.

<sup>98</sup> Avila, E, & Meneses, A. (2013). Delfroid y su comparación evaluativa con XP y Scrum mediante el método 4-DAT. Revista cubana de ciencia, 7, 1.

Otro ejemplo del uso de esta herramienta es, Una evaluación del grado de agilidad en seis métodos ágiles y su aplicabilidad para el método de ingeniería (An evaluation of the degree of agility in six agile methods and its applicability for method engineering)<sup>99</sup>, esta investigación evalúa los siguientes métodos: XP, SCRUM, FDD, ASD, DSDM, CRYSTAL y los ciclos tradicionales: Modelo en cascada y modelo en espiral, aplicándole cada una de las dimensiones por método.

Ellos concluyen lo siguiente: El marco analítico (4-DAT) se ha desarrollado y aplicado a la evaluación detallada del grado de agilidad en seis métodos ágiles seleccionados y dos métodos tradicionales (cascada y espiral). El enfoque de evaluación en este marco de análisis evalúa estos métodos desde cuatro perspectivas, tanto a nivel de procesos como a nivel de práctica. Tal evaluación se puede utilizar para seleccionar racionalmente un método o proceso ágil apropiado para un desarrollo particular, o en la ingeniería, donde un método o proceso de alguna situación específica puede ser construido con un grado apropiado de agilidad.

**4.4.2 Alineación de MAF con la UFPSO.** Se realizaron dos encuestas con el fin de demostrar que MAF sirve para todas aquellas dependencias que deseen o desarrollen software en la UFPSO, ya que es un método que además de mejorar la calidad lograra que su tiempo de ejecución sea más cortó.

Con la realización de la primera encuesta (Anexo B), y cuyo análisis se pudo ver en la parte de análisis de análisis de la información (Pág. 33), se concluyó que en la UFPSO solo la mitad de quienes desarrollaban software usaban métodos ágiles.

La realización de esa encuesta conlleva a la siguiente pregunta ¿Son o no utilizados los métodos ágiles en la industria del desarrollo de software?, como respuesta a esto se obtuvieron algunos resultados:

VersionOne es una empresa encargada de ayudar a desarrollar herramientas ágiles y adoptarlas su misión es simplificar el proceso de entrega del software mediante la implementación de un ciclo de vida ágil, esta empresa anualmente hace una encuesta acerca del uso de métodos ágiles, la novena entrega anual arrojo los siguientes resultados:

El desarrollo ágil basado en equipos está agarrando la atención del negocio.

El estado de la encuesta Agile™ del año 2015 encontró que más empresas están abrazando los métodos ágiles como parte de su visión para ofrecer software más rápido, más fácil y más inteligente. Noventa y cuatro por ciento (94%) de todas las organizaciones encuestadas ahora practican ágil. En 2013, la mayoría de los encuestados tenía menos de 1.000 personas en su organización de software. Pero en 2014, aproximadamente el 35% de los encuestados tenía más de 5.000 personas en su organización, y el 20% trabajaba en organizaciones muy grandes con más de 20.000 personas. Además, el 45% de los encuestados de este año

---

<sup>99</sup> A. Qumer, B. Henderson-Sellers. (2007). An evaluation of the degree of agility in six agile methods and its applicability for method engineering. ScienceDirect, 1, 16.

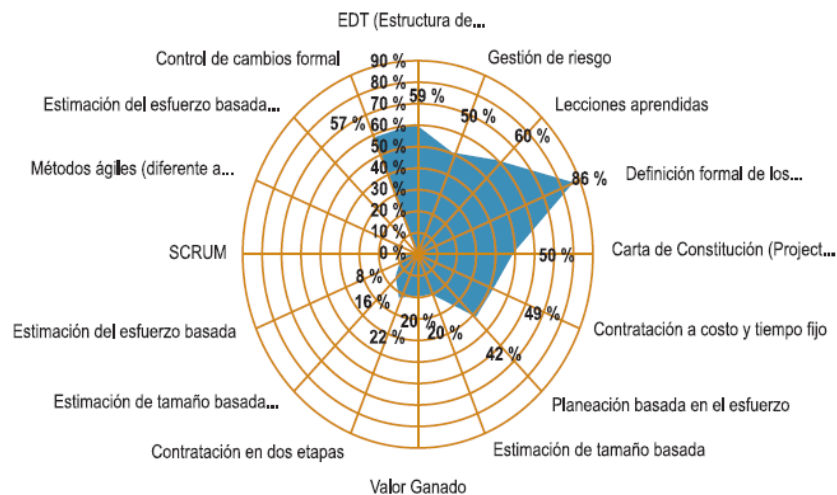
trabajó en organizaciones de desarrollo, donde la mayoría de sus equipos son ágiles. Esto contrasta con el informe de 2009, que encontró que (31%) de los encuestados trabajaban en organizaciones con sólo cero a dos equipos practicando métodos ágiles.

El desarrollo ágil se está extendiendo geográficamente, también. De 2012 a 2014, el porcentaje de encuestados que habían distribuido los equipos que practican ágil saltó del 35% al 80%.<sup>100</sup>

También se puede observar el uso de estos métodos en Colombia, En la XI Jornada de Gerencia de TI se presentó por parte de la Asociación Colombiana de Ingenieros de Sistemas-ACIS, las tendencias actuales en cuanto al desarrollo de proyectos colombianos relacionados con tecnología informática.

Este estudio fue realizado con una participación de 248 proyectos. Para analizar el uso de herramientas y técnicas se muestra la figura 16 donde se resume el porcentaje de utilización de herramientas y técnica para los proyectos en Colombia.<sup>101</sup>

**Figura 16.** Herramientas y técnicas de desarrollo de software.



**Fuente.** Pedroza, P. (2013). Elección de una Metodología de Desarrollo a partir de las Ventajas de una Metodología Ágil y un Modelo Robusto como CMMI-DEV 1.3. INGENIARE, 14, 113-122

Como se puede ver solo el 10% de los proyectos que fueron encuestados practican los métodos ágiles, lo que quiere decir que a diferencia de otros países Colombia no tiene un porcentaje tan alto en el uso de estos métodos.

<sup>100</sup> VersionOne. (Abril de 2015). Agile Project Management. Obtenido de <http://www.versionone.com/>

<sup>101</sup> Pedroza, P. (2013). Elección de una Metodología de Desarrollo a partir de las Ventajas de una Metodología Ágil y un Modelo Robusto como CMMI-DEV 1.3. INGENIARE, 14, 113-122.



La segunda encuesta (Anexo C), se hizo con el fin de lograr una mejor evaluación y comparación MAF, en ella se indaga acerca de los valores presentados y evaluados en 4-DAT, los resultados fueron:

¿Cuántos integrantes hacen parte de su grupo de trabajo?

**Respuesta**

Líder 1                      9 personas  
Líder 2                      personas

¿En el desarrollo de sus proyectos las entregas se hacen en iteraciones cortas o solo se hace una entrega cuando el producto esté finalizado?

**Respuesta**

Líder 1                      Iteraciones cortas  
Líder 2                      Iteraciones cortas

¿Qué lenguaje de programación usa para la realización de sus proyectos?

**Respuesta**

Líder 1                      PHP-Oracle  
Líder 2                      PHP-Java Script

¿Realizan programación en parejas?

**Respuesta**

Líder 1                      Si  
Líder 2                      Si

¿Cuántas horas trabajan semanalmente en la realización de un proyecto?

**Respuesta**

Líder 1                      40 horas  
Líder 2                      40 horas

¿Realizan reuniones diarias o semanales para verificar el progreso del proyecto?

**Respuesta**

Líder 1                      Reuniones diarias  
Líder 2                      Reuniones semanales

¿El cliente hace parte de su equipo de trabajo?

**Respuesta**

Líder 1                      Si  
Líder 2                      Si

¿Qué tipo de artefactos usa para el desarrollo de cada tarea?

**Respuesta**

Líder 1

Casos de uso

Líder 2

Casos de uso, diagramas de clase, modelo de datos.

¿Usa algún artefacto exclusivo para la arquitectura y el diseño?

**Respuesta**

Líder 1

Si

Líder 2

No

¿Realizan documentación del producto?

**Respuesta**

Líder 1

Si

Líder 2

Si

El análisis cualitativo de esta encuesta se hizo antes de realizar la evaluación de MAF con 4-DAT, ya que se usó como instrumento para saber si el método creado estaba cerca de las características que usan los desarrolladores de software de la universidad el ejecutar un proyecto.

**Caso de estudio.** A continuación se hará un ejemplo real sobre el método MAF, el caso de estudio será realizado sobre el DESARROLLO DE UN SISTEMA DE ADMINISTRACIÓN DE CITAS MÉDICAS ORIENTADO A LA WEB PARA LA INSTITUCIÓN PRESTADORA DE SERVICIOS SANAMEDIC S.A.S. el cual fue presentado como proyecto de grado por el estudiante Gerardo Calderón Calderón, bajo la dirección del ingeniero Esp. EDUAR BAYONA IBÁÑEZ. La descripción del proyecto se muestra a continuación:

Hoy, las entidades de salud están sometidas constantemente a presiones competitivas que les exige un mejoramiento continuo en cuanto a calidad de sus servicios, a reducir sus costos operativos y a adoptar las resoluciones, decretos y leyes que son emanadas por el ministerio de la protección social, además, los procesos de gestión de citas médicas y reportes se han vuelto virales para la organización y control en las instituciones médicas, permitiendo que nuevas herramientas tecnológicas se encarguen de mejorar la calidad de los procesos. El propósito del sistema de administración de citas médicas desarrollado para la empresa prestadora de servicios médicos Sanamedic S.A.S es mejorar los procesos que realiza la organización contribuyendo a la calidad de los servicios y eficacia del mismo además de la satisfacción de los usuarios externos e internos.

Partiendo de un análisis realizado en la IPS SANAMEDIC S.A.S se encontró que aproximadamente 100 personas en un solo día, solicitan por comunicación telefónica

servicios de consulta médica general y odontología, además de las citas por terapia de eventos que las EPS autorizan para que los usuarios acudan directamente a la IPS Sanamedic para que estas sean asignadas como lo son: terapia física, terapia respiratoria, terapia ocupacional, terapia de lenguaje y psicología.

Por lo cual resulta urgente que la IPS SANAMEDIC cuente con un sitio web el cual contenga un módulo de administración de citas que este activo las 24 horas del día, los siete días de la semana, que agilice estos procesos, generando calidad y cambiando por completo el modelo de trabajo buscando ofrecer un mejor servicio.

La necesidad de contar con herramientas eficientes y eficaces en el proceso de gestión médica lleva a las instituciones prestadoras de salud a la adquisición de sistemas que permitan un mejoramiento continuo de la organización y de esta manera dar un buen uso de la información que estas manejan en la toma de decisiones en la alta dirección.

El desarrollo del sistema se trabajará bajo una metodología ágil denominada SCRUM que utiliza un proceso iterativo e incremental el cual se ajusta a las etapas de desarrollo de este proyecto que son la planificación, el análisis, el diseño, la construcción, las pruebas y la implementación<sup>102</sup>

A partir del proyecto descrito, se desarrollaran los artefactos planteados por MAF, para dar validez a que mejoran el desarrollo del software y que es un método ágil, que ayuda a la resolución de problemas y los resuelve de manera rápida y eficaz. En este caso se utilizaran algunos productos del proyecto anterior tales y como son requerimientos y fechas de entrega la plantilla propuesta para el product backlog se le añadieron los campos ID-req y prioridad ya que así está planteado en MAF.

**Product Backlog.** El cuadro 42 muestra el primer artefacto planteado por MAF que es el product backlog, en el encontramos el identificador del requerimiento, el requerimiento como tal su prioridad y su fecha de entrega.

**Cuadro 42.** Product Backlog

ID-req	Requerimientos	Prioridad	Fecha de entrega
01	Crear Login	Alta	24/09/2014
02	Migrar Base de Datos	Alta	29/09/2014
03	Consultar usuarios externos en la base de datos	Media	03/10/2014
04	Asignar Cita	Alta	07/10/2014
05	Modificar Cita	Media	10/10/2014

<sup>102</sup> Gerardo Calderón, desarrollo de un sistema de administración de citas médicas orientado a la web para la institución prestadora de servicios Sanamedic s.a.s. Universidad Francisco de Paula Santander Ocaña.

Cuadro 42. (Continuación)

06	Cancelar Cita	Media	15/10/2014
07	Confirmar Asistencia	Baja	17/10/2014
08	Crear usuarios administradores del sistema	Alta	22/10/2014
09	Actualizar usuarios internos	Media	24/10/2014
010	Consultar usuarios internos	Media	29/10/2014
011	Eliminar usuarios Internos	Media	04/11/2014
012	Recuperar Contraseña	Media	10/11/2014
013	Generar Reportes	Media	18/11/2014
014	Generar Indicadores de Oportunidad	Media	24/11/2014
015	Consultar Citas	Alta	27/11/2014
016	Generar reportes por usuarios externos	Media	02/12/2014
017	Imprimir listado de citas diarias por profesional	Baja	09/12/2014

**Fuente.** Autor del proyecto

**Historias de usuario.** Los siguientes cuadros muestran las 17 historias de usuarios desarrolladas para la descripción de cada requerimiento propuesto en el product backlog.

**Cuadro 43.** Historias de usuario crear login

Historia de Usuario	
ID-his: 1	Nombre de historia de usuario: Crear Login
Modificación (o extensión) de Historia de Usuario (Nro y Nombre):	
Iteración o sprint: 1	
Usuario: Desarrollador	
Prioridad en Negocio: Alta (Alta/ Media/ Baja)	
Descripción: El desarrollador crea el formulario de acceso al sistema para la protección de la información.	
Observaciones: Ninguna	

**Fuente.** Autor del proyecto

**Cuadro 44.** Historias de usuario migrar base de datos

Historia de Usuario	
ID-his: 2	Nombre de historia de usuario: Migrar Base de Datos
Modificación (o extensión) de Historia de Usuario (Nro y Nombre):	
Iteración o sprint: 1	
Usuario: Desarrollador	
Prioridad en Negocio: Alta (Alta/ Media/ Baja)	
Descripción: Permite al administrador de citas migrar las bases de datos provenientes de las EPS vinculadas a la institución para traer la información de los usuarios activos de cada EPS.	
Observaciones: Los usuarios deben estar activos.	

**Fuente.** Autor del proyecto

**Cuadro 45.** Historias de usuario Consultar usuarios externos en las bases de datos

Historia de Usuario	
ID-his:3	Nombre de historia de usuario: Consultar usuarios externos en las bases de datos
Modificación (o extensión) de Historia de Usuario (Nro y Nombre):	
Iteración o sprint: 1	
Usuario: Desarrollador	
Descripción: Permite al administrador consultar los usuarios externos activos en las bases de datos de las EPS vinculadas con la empresa para saber su información, el historial de citas además permite saber si el usuario está activo.	
Observaciones:	

**Fuente.** Autor del proyecto

**Cuadro 46.** Historias de usuario Asignar Cita

Historia de Usuario	
ID-his: 4	Nombre de historia de usuario: Asignar Cita
Modificación (o extensión) de Historia de Usuario (Nro y Nombre):	
Iteración o sprint: 1	
Usuario: Desarrollador	
Prioridad en Negocio: Alta (Alta/ Media/ Baja)	
Descripción: Permite al administrador de citas la asignación de la cita según la agenda disponible de cada profesional médico.	
Observaciones:	

**Fuente.** Autor del proyecto

**Cuadro 47.** Historias de usuario Modificar cita

Historia de Usuario	
ID-his: 5	Nombre de historia de usuario: Modificar cita
Modificación (o extensión) de Historia de Usuario (Nro y Nombre):	
Iteración o sprint: 1	
Usuario: Desarrollador	
Prioridad en Negocio: Media (Alta/ Media/ Baja)	
Descripción: Permite al administrador de citas la modificación de la hora o la fecha de la cita según la disponibilidad de la agenda.	
Observaciones: La fecha modificaba debe estar disponible	

**Fuente.** Autor del proyecto

**Cuadro 48.** Historias de usuario Cancelar cita

Historia de Usuario	
ID-his: 6	Nombre de historia de usuario: Cancelar cita
Modificación (o extensión) de Historia de Usuario (Nro y Nombre):	
Iteración o sprint: 1	
Usuario: Desarrollador	
Prioridad en Negocio: Media (Alta/ Media/ Baja)	
Descripción: Permite al administrador de citas la cancelación de una cita previamente asignada se debe cancelar con dos horas de anticipación pues si no será marcada como inasistente además se debe colocar el motivo de la cancelación.	
Observaciones: debe ser cancelada mínimo con dos horas de anticipación	

**Fuente.** Autor del proyecto

**Cuadro 49.** Historias de usuario Confirmar Asistencia

Historia de Usuario	
ID-his: 7	Nombre de historia de usuario: Confirmar Asistencia
Modificación (o extensión) de Historia de Usuario (Nro y Nombre):	
Iteración o sprint: 1	
Usuario: Desarrollador	
Prioridad en Negocio: Baja (Alta/ Media/ Baja)	
Descripción: Permite al administrador de citas la confirmación de la asistencia de una cita por parte del usuario, teniendo en cuenta que el usuario no supere la hora establecida de la cita médica.	
Observaciones: Debe confirmar antes de la hora de la cita	

**Fuente.** Autor del proyecto

**Cuadro 50.** Historias de usuario Crear usuarios administradores del sistema

Historia de Usuario	
ID-his: 8	Nombre de historia de usuario: Crear usuarios administradores del sistema
Modificación (o extensión) de Historia de Usuario (Nro y Nombre):	
Iteración o sprint: 1	
Usuario: Desarrollador	
Descripción: Permite la Creación de usuarios administradores del sistema ingresando sus datos personales y especificando su rol en el sistema.	
Observaciones: ninguna	

**Fuente.** Autor del proyecto

**Cuadro 51.** Historias de usuario Actualizar usuarios internos

Historia de Usuario	
ID-his: 9	Nombre de historia de usuario: Actualizar usuarios internos
Modificación (o extensión) de Historia de Usuario (Nro y Nombre):	
Iteración o sprint: 2	
Usuario: Desarrollador	
Prioridad en Negocio: Media (Alta/ Media/ Baja)	
Descripción: Permite la actualización y modificación de información de usuarios administradores del sistema.	
Observaciones: ninguna	

**Fuente.** Autor del proyecto

**Cuadro 52.** Historias de usuario Consultar usuarios internos

Historia de Usuario	
ID-his: 10	Nombre de historia de usuario: Consultar usuarios internos
Modificación (o extensión) de Historia de Usuario (Nro y Nombre):	
Iteración o sprint: 2	
Usuario: Desarrollador	
Prioridad en Negocio: Media (Alta/ Media/ Baja)	
Descripción: Permite consultar información cargada en la base de datos sobre los usuarios administradores registrados.	
Observaciones: el usuario debe estar creado.	

**Fuente.** Autor del proyecto

**Cuadro 53.** Historias de usuario Eliminar usuarios Internos

Historia de Usuario	
ID-his: 11	Nombre de historia de usuario: Eliminar usuarios Internos
Modificación (o extensión) de Historia de Usuario (Nro y Nombre):	
Iteración o sprint: 2	
Usuario: Desarrollador	
Prioridad en Negocio: Media (Alta/ Media/ Baja)	
Descripción: Permite eliminar la información cargada sobre usuarios administradores del sistema.	
Observaciones: ninguna	

**Fuente.** Autor del proyecto

**Cuadro 54.** Historias de usuario Recuperar Contraseña

Historia de Usuario	
ID-his: 12	Nombre de historia de usuario: Recuperar Contraseña
Modificación (o extensión) de Historia de Usuario (Nro y Nombre):	
Iteración o sprint: 2	
Usuario:	
Prioridad en Negocio: Media (Alta/ Media/ Baja)	
Descripción: Permite enviar al correo registrado en la base de datos del usuario interno una nueva contraseña generada aleatoriamente y enviarlas al correo del usuario que la solicita.	
Observaciones: ninguna	

**Fuente.** Autor del proyecto

**Cuadro 55.** Historias de usuario: Generar Reportes

Historia de Usuario	
ID-his: 13	Nombre de historia de usuario: Generar Reportes
Modificación (o extensión) de Historia de Usuario (Nro y Nombre):	
Iteración o sprint: 2	
Usuario:	
Prioridad en Negocio: Media (Alta/ Media/ Baja)	
Descripción: Permite generar estadísticas de las consultas atendidas durante un lapso de tiempo determinado de las EPS vinculadas a la institución.	
Observaciones: ninguna	

**Fuente.** Autor del proyecto

**Cuadro 56.** Historias de usuario: Generar Indicadores de Oportunidad

Historia de Usuario	
ID-his: 14	Nombre de historia de usuario: Generar Indicadores de Oportunidad
Modificación (o extensión) de Historia de Usuario (Nro y Nombre):	
Iteración o sprint: 2	
Usuario: Desarrollador	
Prioridad en Negocio: Media (Alta/ Media/ Baja)	
Descripción: Permite a la institución la generación de reportes de la oportunidad en la asignación de la cita según la resolución 1446 de 2006.	
Observaciones: ninguna	

**Fuente.** Autor del proyecto



**Cuadro 57.** Historias de usuario: Consultar Citas

Historia de Usuario	
ID-his: 15	Nombre de historia de usuario: Consultar Citas
Modificación (o extensión) de Historia de Usuario (Nro y Nombre):	
Iteración o sprint: 2	
Usuario: Desarrollador	
Prioridad en Negocio: Alta (Alta/ Media/ Baja)	
Descripción: Permite al administrados de citas consultar las citas registradas.	
Observaciones: ninguna	

**Fuente.** Autor del proyecto**Cuadro 58.** Historias de usuario: Generar reportes por usuarios externos

Historia de Usuario	
ID-his: 16	Nombre de historia de usuario: Generar reportes por usuarios externos
Modificación (o extensión) de Historia de Usuario (Nro y Nombre):	
Iteración o sprint: 2	
Usuario: Desarrollador	
Prioridad en Negocio: Media (Alta/ Media/ Baja)	
Descripción: Permite generar reportes por usuario a los que se les hallan asignado citas médicas, genera reportes de cuál es el rango de tiempo el cual el usuario ha solicitado citas médicas, si el rango está en 20 días, no se podrá asignar una nueva cita siempre y cuando sea consulta médica general, para odontología el rango mínimo de días para acceder nuevamente a una nueva cita es de 8 días.	
Observaciones: ninguna	

**Fuente.** Autor del proyecto**Cuadro 59.** Historias de usuario: Generar reportes por usuarios externos

Historia de Usuario	
ID-his: 17	Nombre de historia de usuario: Imprimir listado de citas diarias por profesional
Modificación (o extensión) de Historia de Usuario (Nro y Nombre):	
Iteración o sprint: 2	
Usuario: Desarrollador	
Prioridad en Negocio: Baja (Alta/ Media/ Baja)	
Descripción: Permite imprimir el listado de las citas diarias atendidas por profesional, genera la lista de las citas médicas atendidas por un profesional específico en un día específico y poder imprimirlas.	
Observaciones: ninguna	

**Fuente.** Autor del proyecto

**SPRINT BACKLOG.** El cuadro 60 hace referencia al sprint backlog, para la realización de este artefacto se modificó la tabla original para acoplarse a la plantilla usada por MAF.

**Cuadro 60.** Sprint Backlog

ID- his	Tareas	Estado	# Sprint o iteración
1	Crear Base de datos	Pendiente	1
	Hacer diseño de la interfaz	Pendiente	1
	Conectar base de datos	Pendiente	1
	Codificar	Pendiente	1
2	Hacer diseño de la interfaz	Pendiente	1
	Codificar	Pendiente	1
	Realizar pruebas	Pendiente	1
3	Hacer diseño de la interfaz	Pendiente	1
	Codificar	Pendiente	1
	Realizar pruebas	Pendiente	1
4	Crear formularios	Pendiente	1
	Crear agenda	Pendiente	1
	Codificar	Pendiente	1
	Realizar pruebas	Pendiente	1
5	Hacer diseño de la interfaz	Pendiente	1
	Codificar	Pendiente	1
	realizar pruebas	Pendiente	1
6	Hacer diseño de la interfaz	Pendiente	1
	Codificar	Pendiente	1
	realizar pruebas	Pendiente	1
7	Codificar	Pendiente	1
8	Crear formularios	Pendiente	1
	Codificar	Pendiente	1
	Ingresar usuarios	Pendiente	1
9	Crear formularios	Pendiente	2
	Codificar	Pendiente	2
	Realizar Pruebas	Pendiente	2

Cuadro 60. (Continuación)

10	Hacer diseño de la interfaz	Pendiente	2
	Codificar	Pendiente	2
	Realizar pruebas	Pendiente	2
11	Hacer diseño de la interfaz	Pendiente	2
	Codificar	Pendiente	2
	Realizar pruebas	Pendiente	2
12	Hacer diseño de la interfaz	Pendiente	2
	Codificar	Pendiente	2
	Realizar pruebas	Pendiente	2
13	Hacer diseño de la interfaz	Pendiente	2
	Crear formularios	Pendiente	2
	Codificar	Pendiente	2
	Realizar pruebas	Pendiente	2
14	Hacer diseño de la interfaz	Pendiente	2
	Crear formularios	Pendiente	2
	Codificar	Pendiente	2
	Realizar pruebas	Pendiente	2
15	Hacer diseño de la interfaz	Pendiente	2
	Codificar	Pendiente	2
	Realizar pruebas	Pendiente	2
16	Hacer diseño de la interfaz	Pendiente	2
	Codificar	Pendiente	2
	Realizar pruebas	Pendiente	2
17	Hacer diseño de la interfaz	Pendiente	2
	Codificar	Pendiente	2
	Realizar pruebas	Pendiente	2

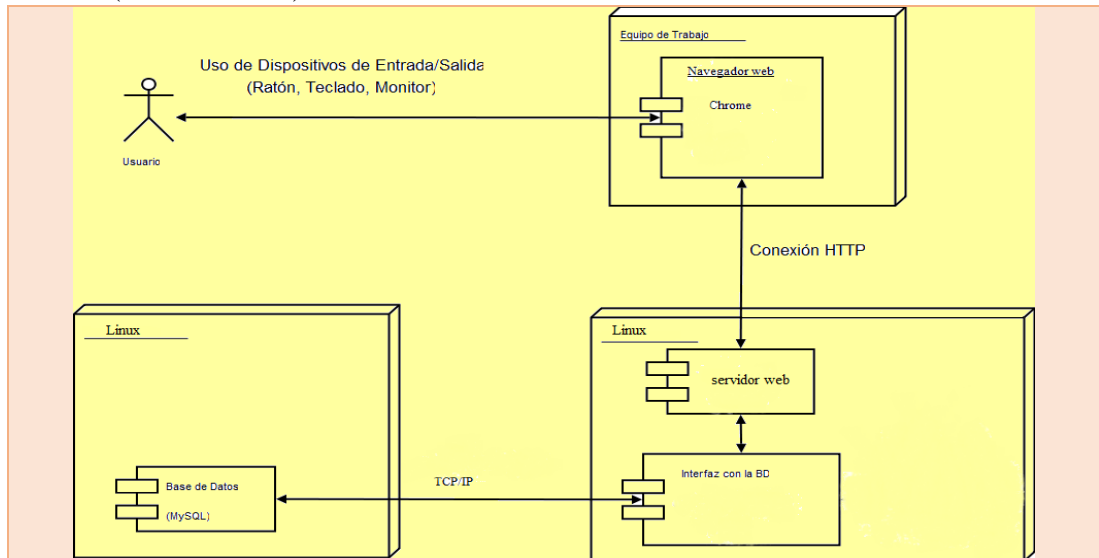
**Fuente.** Adaptado de: Gerardo Calderón, desarrollo de un sistema de administración de citas médicas orientado a la web para la institución prestadora de servicios Sanamedic s.a.s. Universidad Francisco de Paula Santander Ocaña.

**Libro de arquitectura y diseño.** El libro de arquitectura y diseño es un artefacto tomado por MAF, que no hace parte del proyecto escogido para el caso práctico. A continuación el cuadro 61 muestra el libro de arquitectura y diseño para el caso de estudio.

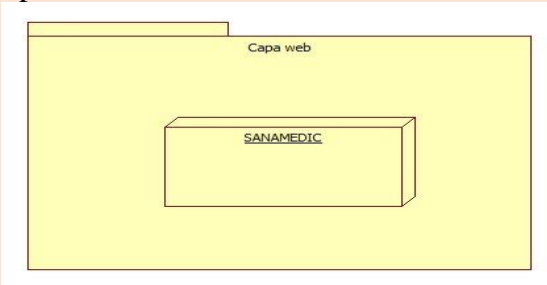
**Cuadro 61.** Libro de arquitectura y diseño

<b>Libro de arquitectura</b> <b>Nombre del proyecto: DESARROLLO DE UN SISTEMA DE ADMINISTRACIÓN DE CITAS MÉDICAS ORIENTADO A LA WEB PARA LA INSTITUCIÓN PRESTADORA DE SERVICIOS SANAMEDIC S.A.S.</b>	
<b>Objeto:</b> se decidió usar la arquitectura cliente-servidor Proporciona mejor aprovechamiento de la potencia de cómputo, reduce el tráfico en la red, opera bajo sistemas abiertos, permite el uso de interfaces gráficas variadas y versátiles.	<b>Objetivos de la arquitectura:</b> mostrar un sistema que permita llevar una organización y un control en la asignación de citas y generación de reportes.
<b>Supuestos y dependencias:</b> Supuestos: Los usuarios que deseen gestionar sus citas médicas vía web deben saber manejar las aplicaciones web. Solo se atenderán personas activas en el sistema. Dependencias: Siempre se debe contar con la base de datos donde se archivan los usuarios activos. La estructura de los archivos debe ser constante. Conexión a internet, ya que sin él no se pueden realizar las consultas.	
<b>Requisitos de mayor valor arquitectónico:</b> El software debe verse de manera intuitiva para que los usuarios puedan interactuar de manera fácil y se familiaricen con el rápidamente. Incorporar mecanismos que faciliten la navegación y el desplazamiento entre las diferentes vistas de una aplicación. Estar disponible Dar respuesta en el menor tiempo posible Validar los usuarios No permitir accesos a usuarios no autorizados	
<b>Decisiones, limitaciones y justificaciones:</b> No existe una topología o estructura de red que satisfaga las necesidades del software. El software debe implementarse solo en el navegador CHROME. Es necesario que los componentes visuales produzcan una interfaz gráfica con apariencia moderna.	
<b>Mecanismos de arquitectura:</b> Capacidad de remplazo de los componentes:	

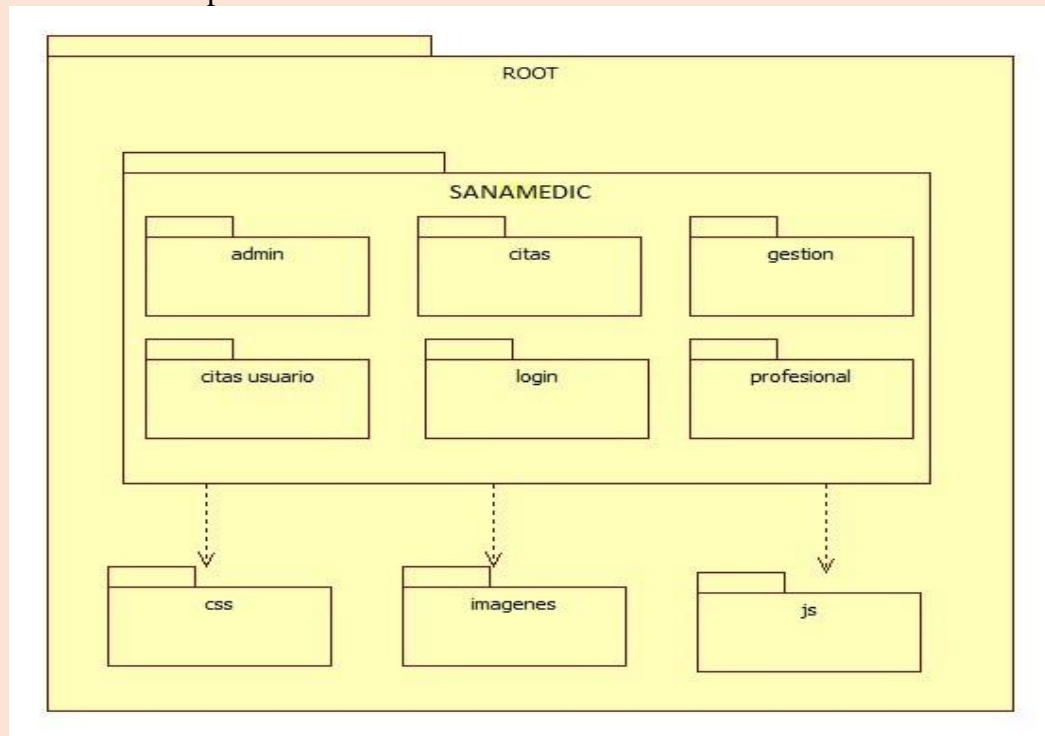
Cuadro 61. (Continuación)



Instanciación de componentes:

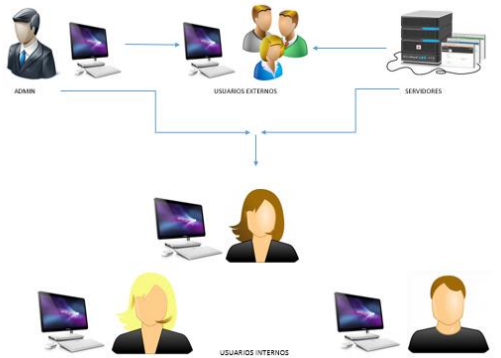


Modelando componentes:



Cuadro 61. (Continuación)

Abstracciones clave:



Capas o marco arquitectónico:

**Patrón de diseño MVC (Modelo-Vista-Controlador):**

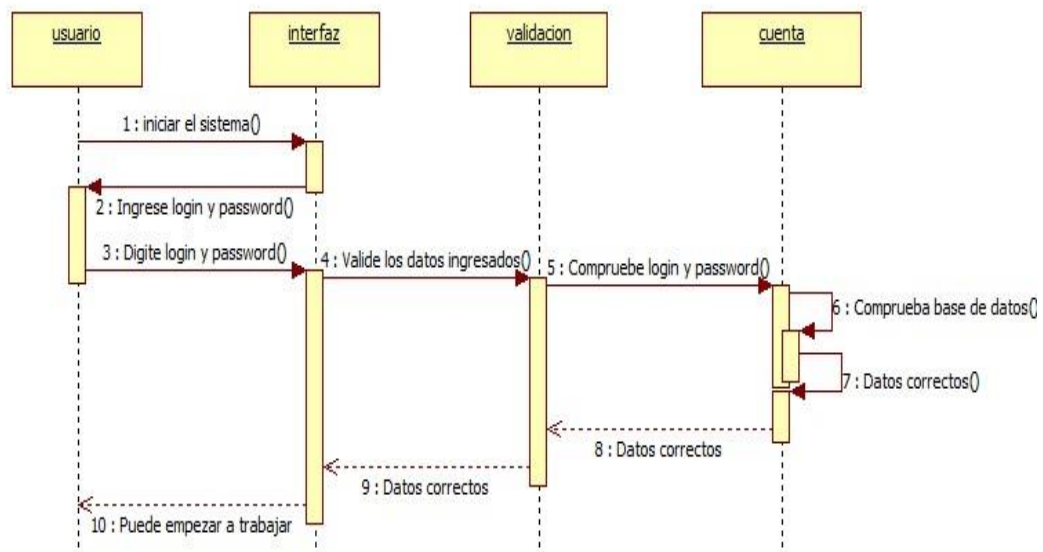
El modelo vista-controlador es aquel que organiza el código dependiendo su función se divide en tres capas las cuales son:<sup>103</sup>

**La capa del modelo** define la lógica de negocio (la base de datos pertenece a esta capa). **La vista** es lo que utilizan los usuarios para interactuar con la aplicación (los gestores de plantillas pertenecen a esta capa).

**El controlador** es un bloque de código que realiza llamadas al modelo para obtener los datos y se los pasa a la vista para que los muestre al usuario.

Vistas de arquitectura:

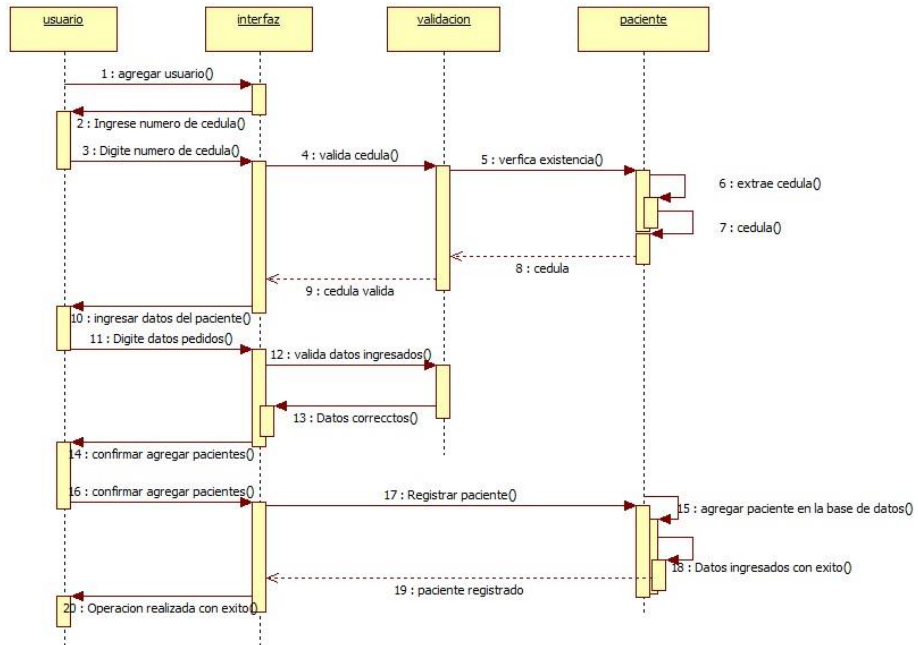
**Login**



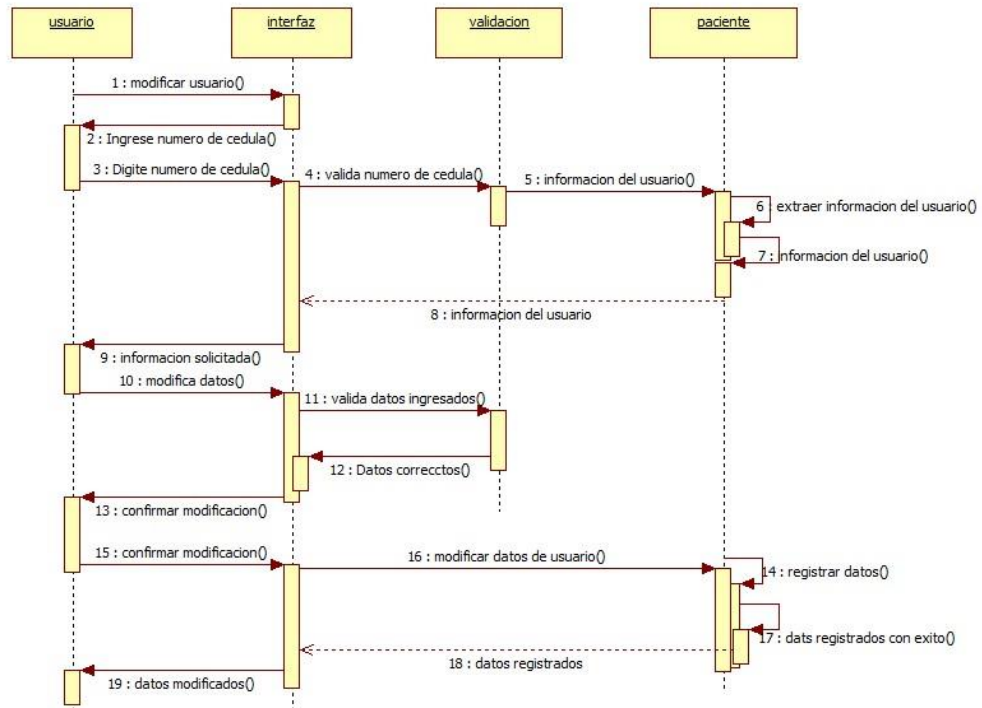
<sup>103</sup> Potencier, F. (2010). El tutorial Jobeet. España: LibrosWeb.

Cuadro 61. (Continuación)

Agregar usuario

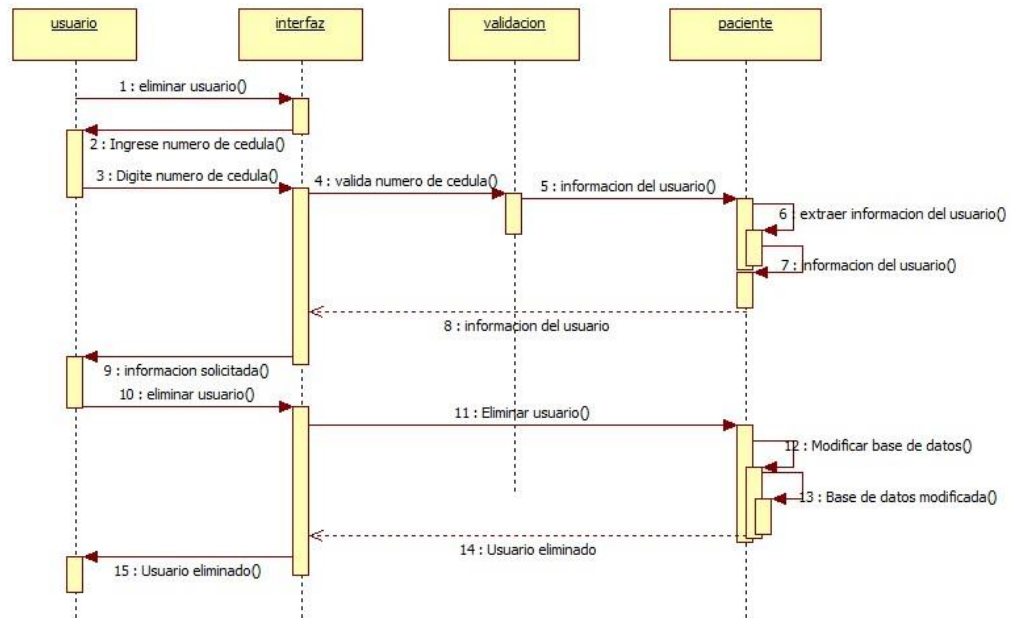


Modificar usuario

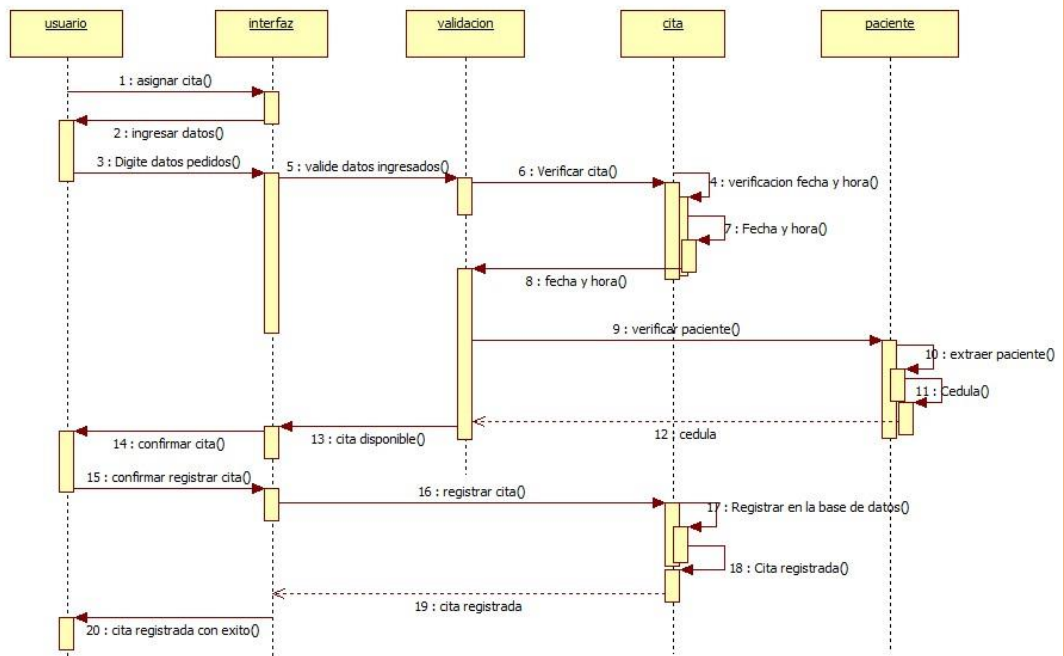


Cuadro 61. (Continuación)

Eliminar usuario



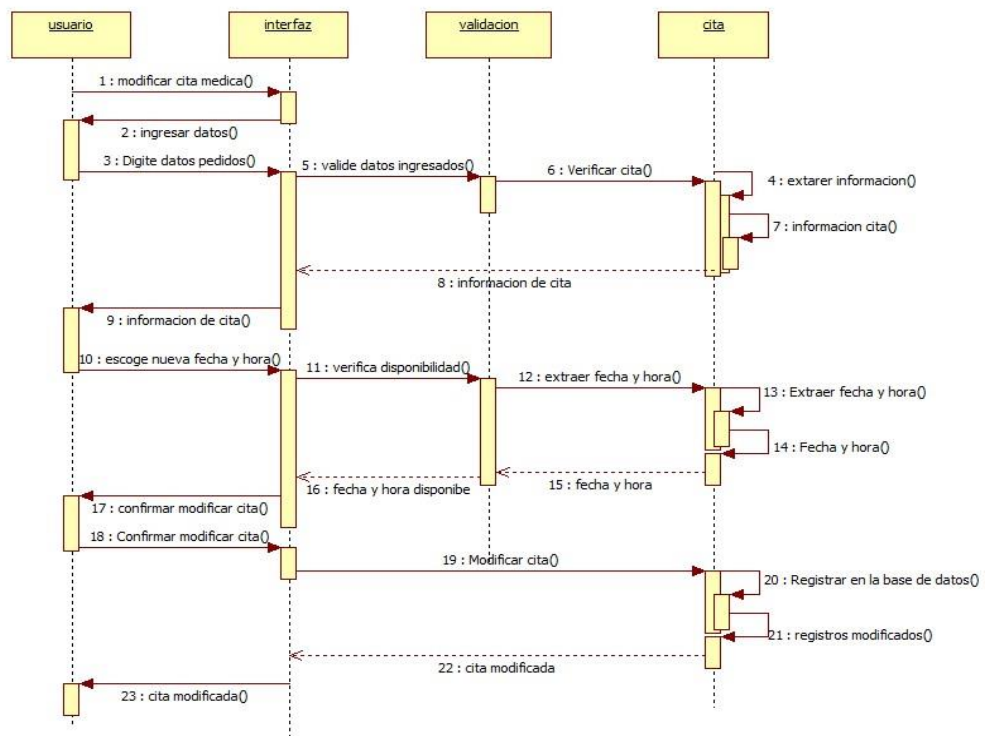
Asignar cita



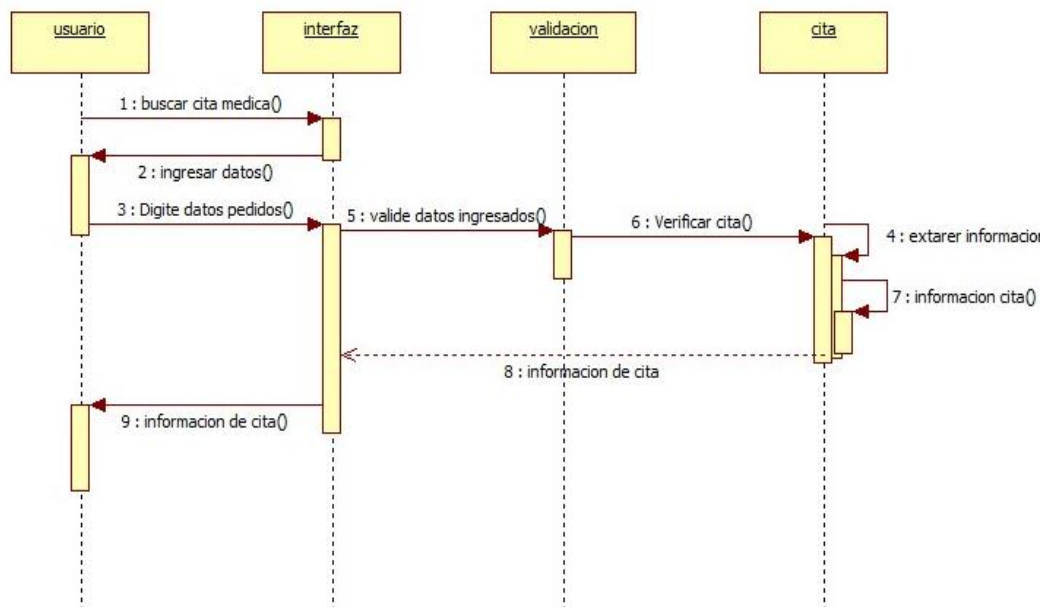


Cuadro 61. (Continuación)

Modificar cita



Buscar cita



Fuente. Autor del proyecto

**Prueba de desarrollo.** La prueba de desarrollo es un artefacto tomado por MAF, que no hace parte del proyecto escogido para el caso práctico pero se usaron las pruebas de caja negra y caja blanca del proyecto seleccionado para la realización de este artefacto. Los siguientes cuadros muestran la prueba de desarrollo hecha a los diferentes requerimientos.

**Cuadro 62.** Prueba de desarrollo 001

<b>Prueba de desarrollo</b>
Numero de prueba: 001
ID-his: 1
Iteración o #sprint: 1
Descripción: crear login para el inicio de sesión de usuarios internos.
Resultado: se logró crear un login para que los usuarios internos tengan acceso al sistema y puedan realizar las actividades de gestión que necesiten.
Estado (aprobada, no aprobada): Aprobada.

**Fuente.** Autor del proyecto

**Cuadro 63.** Prueba de desarrollo 002

<b>Prueba de desarrollo</b>
Numero de prueba: 002
ID-his: 2
Iteración o #sprint: 1
Descripción: Permite migrar las bases de datos de los usuarios externos activos en las EPS Comfaorient y Ecoopsos a la base de datos del sistema. Teniendo en cuenta que hay que borrar primero los datos antiguos de la EPS a la cual se vaya a hacer la migración de datos. Datos de entrada validos: Redireccionamiento a la pantalla donde se eliminara la base de datos actual para proceder con la migración, luego del siguiente paso, aparece el mensaje importación exitosa.” Datos de entrada inválidos: Aparece el mensaje “Archivo invalido” En caso de escoger un código de una EPS que no exista, aparece el mensaje “Error la EPS no es válida.”
Resultado: Se verificó el archivo TXT y la existencia de la EPS en la base de datos.
Estado (aprobada, no aprobada): Aprobada

**Fuente.** Autor del proyecto

**Cuadro 64.** Prueba de desarrollo 003

<b>Prueba de desarrollo</b>
Numero de prueba: 003
ID-his: 3
Iteración o #sprint: 1

Cuadro 64. (Continuación)

Descripción: Consultar la existencia de los usuarios externos activos en las bases de datos de las EPS Comfaorient y Ecoopsos solo con el número de documento.
Datos validos: Re direccionamiento a la pantalla donde mostrara los datos del usuario.
Datos inválidos: Aparece el mensaje “El usuario no está activo, por favor verifique el número de documento”
Resultado: Se verificó la existencia del usuario externo en la base de datos migrada.
Estado (aprobada, no aprobada): Aprobada.

**Fuente.** Autor del proyecto

Cuadro 65. Prueba de desarrollo 004

<b>Prueba de desarrollo</b>
Numero de prueba: 004
ID-his: 4
Iteración o #sprint: 1
Descripción: Permite al administrador de citas la asignación de la cita según la agenda disponible de cada profesional médico.
Datos de entrada validos: Re direccionamiento al formulario para la asignación de la cita médica.
Datos de entrada inválidos: Re direccionamiento al listado de citas de cada médico, No tiene permisos para ejecutar la acción, la fecha u hora de la cita ya está ocupada para este profesional.”
Resultado: Se verificaron los datos de la asignación de la cita médica en la base de datos, prosiguiendo con el proceso según corresponda.
Estado (aprobada, no aprobada): aprobada

**Fuente.** Autor del proyecto

Cuadro 66. Prueba de desarrollo 005

<b>Prueba de desarrollo</b>
Numero de prueba: 005
ID-his: 5
Iteración o #sprint: 1
Descripción: Realizar la modificación en los datos de una cita reservada teniendo en cuenta que esta no debe haberse procesado aun, lo que quiere decir que no debe estar en estado “asistió, ausente, o cancelada”
Datos de entrada validos: Re direccionamiento al formulario para la modificación de la cita médica.
Datos de entrada inválidos: Re direccionamiento al listado de citas del usuario, aparece el mensaje “Error!! No tiene permisos para ejecutar la acción, la cita que intenta modificar ya fue procesada.”
Resultado: Se verificaron los datos de la cita médica en la base de datos, prosiguiendo con el proceso según corresponda
Estado (aprobada, no aprobada): Aprobada

**Fuente.** Autor del proyecto

**Cuadro 67.** Prueba de desarrollo 006

<b>Prueba de desarrollo</b>
Numero de prueba: 006
ID-his:6
Iteración o #sprint: 1
Descripción: Realizar la cancelación de una cita reservada teniendo en cuenta que esta no debe haberse procesado aun, lo que quiere decir que no debe estar en estado “asistió, ausente, o cancelada” Datos de entrada validos: Se procede a la cancelación de la cita médica. Datos de entrada inválidos: Re direccionamiento al listado de citas del usuario, aparece el mensaje “Error!! No tiene permisos para ejecutar la acción, la cita que intenta cancelar ya fue procesada.”
Resultado: Se verificaron los datos de la cita médica en la base de datos, prosiguiendo con el proceso según corresponda.
Estado (aprobada, no aprobada): Aprobada

**Fuente.** Autor del proyecto

**Cuadro 68.** Prueba de desarrollo 007

<b>Prueba de desarrollo</b>
Numero de prueba: 007
ID-his: 7
Iteración o #sprint:1
Descripción: Permite al administrador de citas la confirmación de la asistencia de una cita por parte del usuario, teniendo en cuenta que el usuario no supere la hora establecida de la cita médica. Datos de entrada validos: Se procede a la confirmación de la asistencia a la cita médica. Datos de entrada inválidos: Re direccionamiento al listado de citas del usuario, aparece el mensaje “Error!! No tiene permisos para ejecutar la acción, la cita que intenta confirmar supera la hora establecida.”
Resultado: Se verificaron los datos de la cita médica en la base de datos, prosiguiendo con el proceso según corresponda.
Estado (aprobada, no aprobada): aprobada

**Fuente.** Autor del proyecto

**Cuadro 69.** Prueba de desarrollo 008

<b>Prueba de desarrollo</b>
Numero de prueba: 008
ID-his: 13
Iteración o #sprint: 2
Descripción: Permite la generación de reportes y estadísticas de las citas atendidas

**Cuadro 69. (Continuación)**

durante un rango de fechas establecido por el usuario.
Datos de entrada validos: Re direccionamiento a la página donde se mostrara la tabla generada con los reportes de citas.
Datos de entrada inválidos: En caso de escoger un rango de fecha donde no se hallan asignado citas, aparece el mensaje “error!! No existen citas programadas para las fechas solicitadas, por favor seleccione un rango de fechas valido.”
Resultado: Se verificó la existencia de citas médicas en el rango de fechas seleccionado en la base de datos.
Estado (aprobada, no aprobada): Aprobada.

**Fuente.** Autor del proyecto

**Cuadro 70. Prueba de desarrollo 009**

<b>Prueba de desarrollo</b>
Numero de prueba: 009
ID-his: 14
Iteración o #sprint: 2
Descripción: Permite la generación de indicadores de oportunidad, seleccionando la EPS y el rango de fechas.
Datos de entrada validos: Re direccionamiento a la página donde se mostraran las tablas generada con los indicadores.
Datos de entrada inválidos: En caso de escoger un rango de fecha donde no se hallan asignado citas, aparece el mensaje “error!! No existen citas programadas para las fechas solicitadas, por favor seleccione un rango de fechas valido.”
En caso de escoger un código inválido de las EPS, aparece el mensaje “Error EPS no valida.”
Resultado: Aprobada
Estado (aprobada, no aprobada):

**Fuente.** Autor del proyecto

**Cuadro 71. Prueba de desarrollo 0010**

<b>Prueba de desarrollo</b>
Numero de prueba: 0010
ID-his: 17
Iteración o #sprint: 2
Descripción: Permite imprimir el listado de citas del día de un profesional de la organización escogiendo la fecha en la cual quiere imprimir dicho listado.
Datos de entrada validos: Re direccionamiento a la pantalla donde mostrara el listado de citas del profesional y la opción de imprimir.
Datos de entrada inválidos: Aparece el mensaje “El profesional médico no tiene citas procesadas para la fecha solicitada.”
En caso de escoger un código de un profesional que no exista, aparece el mensaje

Cuadro 71. (Continuación)

“Error, el profesional médico no existe”
Resultado: Se verificó la existencia de citas en la fecha solicitada y la existencia del profesional médico en la base de datos.
Estado (aprobada, no aprobada): Aprobada

**Fuente.** Autor del proyecto

**Documentación del producto.** Para realizar este artefacto se hizo una entrevista a quien realizó el proyecto tomado para el caso de estudio.

Se desarrolló un sistema de administración de citas médicas orientado a la web, el sistema permite asignación, consulta, modificación y cancelación de las citas, permite migrar bases de datos con la información de los usuarios de las EPS vinculadas a la institución, además de la generación de estadísticas y generación de indicadores de oportunidad en la asignación de las citas lo cual es obligatorio para las IPS reportar ante las entidades reguladoras de salud, se tiene un módulo exclusivo para los usuarios que permite que los usuarios puedan de manera autónoma solicitar, modificar y cancelar sus citas vía web, consulta la información de los usuarios y permite ver su historial de citas, en cuanto a los usuarios internos se pueden agregar, visualizar, modificar y eliminar estos usuarios. También se puede imprimir el listado de las citas que se dan durante el día por profesional médico para el control de entrada y salida de historias clínicas.

El sistema está alojado en un servidor, y fue desarrollado en el lenguaje de programación PHP, se realizó en la herramienta de desarrollo web SUBLIME TEXT 2 y como motor de base de datos MySQL, el sistema operativo en que fue realizado fue Windows 7 y el servidor de aplicación utilizado fue XAMP, el navegador en que es soportado es CHROME.

**Documentación del usuario.** Para este artefacto se realiza el manual de usuario correspondiente tomado del proyecto usado para el caso de estudio, el manual de usuario es un documento de comunicación técnica que busca brindar asistencia a los usuarios internos y externos del sistema de administración de citas médicas. Ver anexo D. Manual de usuario del sistema de administración de citas médicas orientado a la web para la institución prestadora de servicios Sanamedic S.A.S.

Al realizar un caso práctico donde se aplicó MAF se puede concluir que este nuevo método creado logra cumplir con todos los aspectos necesarios para considerarse ágil, además de eso sus artefactos hacen que en el desarrollo de cualquier proyecto de gestión de software se tenga más claro lo que se está haciendo, además de tener un orden exacto para que al final no haya ninguna equivocación. En cuanto a los roles al ser este un caso hipotético el director de este trabajo Mg Albeiro Rosado Gómez hizo el papel de facilitador en todo el proceso.

## 5. CONCLUSIONES

Se logró crear MAF, un método ágil de desarrollo de software que posee características esenciales para facilitar y mejorar la calidad de los proyectos desarrollados por los grupos de gestión de software de la UFPSO.

Los Métodos ágiles XP, SCRUM y OPENUP se detallaron, explicando sus fases, roles artefactos y particularidades con lo cual se pudo evaluar sus ventajas y desventajas.

La evaluación de los métodos ágiles permitió crear un método ágil híbrido entre XP, SCRUM y OPENUP un método flexible que se logra adaptar a una gran variedad de proyectos.

El uso de la herramienta de análisis y comparación 4-DAT sirvió para comprobar que el método planteado contaba con las características necesarias para considerarse AGIL, se puede concluir que MAF es un método ágil que posee fases, roles y artefactos completos para poder desarrollar un buen proyecto de gestión de software.

## 6. RECOMENDACIONES

MAF posee todo los valores y prácticas que sugiere el manifiesto ágil es por eso que es recomendable su uso para cualquier grupo de desarrollo de software que realizan proyectos de largo o poco alcance.

MAF puede ser utilizado por estudiantes del área de ingeniería de sistemas que estén desarrollando proyectos de gestión de software

MAF recomienda que al usarlo se mantenga el trabajo en equipo y logre que exista una retroalimentación temprana para cumplir a tiempo con los objetivos propuestos.

Llenar todas las plantillas que formula MAF, para el correcto desarrollo del proyecto.

Cumplir con los tiempos establecidos para el desarrollo de los requisitos del sistema.

Los grupos de desarrollo deberán ser muy explícitos en el desarrollo de la documentación del producto ya que sin esto será muy difícil que se logre entender el manejo del sistema.



## BIBLIOGRAFIA

A.Qumer, B. Henderson-Sellers. (2007). An evaluation of the degree of agility in six agile methods and its applicability for method engineering. ScienceDirect, 1, 16.

Alaimo, Diego Martín. Proyectos ágiles con Scrum : flexibilidad, aprendizaje, innovación y colaboración en contextos complejos . - 1ª ed. - Ciudad Autónoma de Buenos Aires : Kleer, 2013. EBook.

AMO Fernando, Martinez Loic & Segovia Francisco. INTRODUCCION A LA INGENIERIA DE SOFTWARE modelos de desarrollo de programas. Delta publicaciones. Madrid (2005).

AVILA, E, & Meneses, A. (2013). Delfdroid y su comparación evaluativa con XP y Scrum mediante el método 4-DAT. Revista cubana de ciencia, 7, 1.

AVILA, E, & Meneses, A. (2013). Delfdroid y su comparación evaluativa con XP y Scrum mediante el método 4-DAT. Revista cubana de ciencia, 7, 1.

BRITO ACUÑA Karennny, (2009), “SELECCIÓN DE METODOLOGÍAS DE DESARROLLO PARA APLICACIONES WEB EN LA FACULTAD DE INFORMÁTICA DE LA UNIVERSIDAD DE CIENFUEGOS”.

CALDERÓN Gerardo, desarrollo de un sistema de administración de citas médicas orientado a la web para la institución prestadora de servicios Sanamedic s.a.s. Universidad Francisco de Paula Santander Ocaña.

FONTELO Carlos, METODOLOGIA DE DESARROLLO DE SOFTWARE Y UML, Alfaomega, México (2005)

HIGHSMITH, J. "Agile Software Development Ecosystems". Addison-Wesley. 2002.

SOMMERVILLE Ian. (2005). INGENIERIA DEL SOFTWARE, séptima edición. Madrid: Pearson educación, S.A.

SCOTT Ambler. “Artifacts for Agile Modeling: The UML and beyond”, 2003

KENDALL & Kendall. (2005). Análisis y diseño de sistemas. México: Pearson educación.

KENT, B... (1999). (Extreme Programming Explained: Embrace Change). EEUU: Addison-Wesley.

PEDROZA, P. (2013). Elección de una Metodología de Desarrollo a partir de las Ventajas de una Metodología Ágil y un Modelo Robusto como CMMI-DEV 1.3. INGENIARE, 14, 113-122.

POTENCIER, F. (2010). El tutorial Jobeet. España: LibrosWeb.

PRESSMAN, R. (1998). Ingeniería del software: Un enfoque práctico. México: McGraw-Hill.

PRESSMAN, R. (2002). Ingeniería del software: Un enfoque práctico. México: McGraw-Hill.

PRESSMAN, R. (2006). Ingeniería del software: Un enfoque práctico. México: McGraw-Hill.

QUMER, A., Henderson-Sellers, B.: An evaluation of the degree of agility in six agile methods and its applicability for method engineering. In: Information and Software Technology. (2007)

ZACHMAN, John. (1999), El modelado de las empresas: la arquitectura Zachman, Zachman Institute for Framework Advancement. Estados Unidos.

## REFERENCIAS DOCUMENTALES ELECTRONICAS

ANAYA, A. (2010). A propósito de programación extrema XP (eXtreme Programming). 2010, de UNIVERSIDAD CARLOS III DE MADRID Sitio web:<https://www.google.com.co/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&cad=rja&uact=8&ved=0CCoQFjAC&url=http%3A%2F%2Fzew.googlecode.com%2Fsvn%2Ftrunk%2Fdoc%2Fformacion%2Fprogramacionextrema.doc&ei=H13eVKGAMsezggS9r4PwAg&usg=AFQjCNEy6WmAiT8knGyo7XpKVI98IPvCTg&sig2=yGly78iJfqi-y8QVN3Enew&bvm=bv.85970519,d.eXY>

Artefactos, disponible en internet en:  
<http://clases3gingsof.wikifoundry.com/page/Artefactos>

Artefactos, disponible en internet en:  
<http://www.abacovirtual.edu.pe/chiclayo/filedocente/140318-CEI2012203MB-000029-19072012-173247.pdf>

BLANCO P, Camarero J, Fumero A, Werleski A, Rodriguez P, (2009); Metodología de desarrollo ágil para sistemas móviles Introducción al desarrollo con Android y el iPhone, recuperado de: [http://www.adamwesterski.com/wp-content/files/docsCursos/Agile\\_doc\\_TemasAnv.pdf](http://www.adamwesterski.com/wp-content/files/docsCursos/Agile_doc_TemasAnv.pdf)

BLANCO P, Camarero J, Fumero A, Werleski A, Rodriguez P, (2009); Metodología de desarrollo ágil para sistemas móviles Introducción al desarrollo con Android y el iPhone, recuperado de: [http://www.adamwesterski.com/wp-content/files/docsCursos/Agile\\_doc\\_TemasAnv.pdf](http://www.adamwesterski.com/wp-content/files/docsCursos/Agile_doc_TemasAnv.pdf)

CONCYTEC Revista de la Facultad de Ingeniería Industrial 13(1): 70-74 (2010) UNMS ISSN: 1560-9146 (Impreso) / ISSN: 1810-9993 (Electrónico) (s.l.) [En línea]. Mar. 2010. [Citado el 25 noviembre de 2014.] Disponible en internet en: [http://revistas.concytec.gob.pe/scielo.php?pid=S1810-99932010000200009&script=sci\\_arttext](http://revistas.concytec.gob.pe/scielo.php?pid=S1810-99932010000200009&script=sci_arttext)

Rivadeneira Molina, Silvia. Metodologías ágiles enfocadas al modelado de requerimientos. En: Revista de informe científicos técnicos. ICT-UNPA-57-2013 ISSN: 1852 - 4516

Egas, Luis. Jativa, Juan. Evolución de las Metodologías de Desarrollo de la Ingeniería de Software en el Proceso la Ingeniería de Sistemas Software y Determinación de una metodología adaptable orientada a una organización pequeña. Repositorio digital ESPE. 2014

Diferencias encontrado en internet en:  
<https://sites.google.com/site/metodololgdelainvestig/modulo-0-2/deferencias-entre>

Disponible en: <http://epf.eclipse.org/wikis/openup/>

Disponible en: [http://librosweb.es/libro/symfony\\_1\\_4/capitulo\\_1/symfony\\_en\\_pocas\\_palabras.html](http://librosweb.es/libro/symfony_1_4/capitulo_1/symfony_en_pocas_palabras.html)

Disponible en: <http://projects.staffs.ac.uk/suniwe/project/rup.html>

Disponible en: <http://www.scrumguides.org/index.html>

Disponible en: <http://www.scrumguides.org/index.html>

Disponible en: <http://www.versionone.com/>

EPF.ECLIPSE. OpenUP (s.l.) [En línea]. Mar. 2010. [Citado el 25 noviembre de 2014.]  
Disponible en internet en: <http://epf.eclipse.org/wikis/openup/>

EPF.ECLIPSE. OpenUP (s.l.) [En línea]. Mar. 2010. [Citado el 25 noviembre de 2014.]  
Disponible en internet en: <http://epf.eclipse.org/wikis/openup/>

FIGUEROLA, N. (2012). Definiendo requerimientos: Tradicional vs. casos de uso vs. historias de usuario. 2012, de Buenos Aires Sitio web: <https://articulosit.files.wordpress.com/2012/04/requerimientos.pdf>

Historias de usuario vs casos de uso disponible en internet en: <http://www.genbetadev.com/metodologias-de-programacion/historias-de-usuario-una-forma-natural-de-analisis-funcional>

<http://agilemanifesto.org/iso/es/manifesto.html>

<http:// analisis1daid.wikispaces.com/Metodolog%C3%ADas+%C3%81giles>

<http://epf.eclipse.org/wikis/openup/>

<http://www.extremeprogramming.org/>

<http://www.proyectosagiles.org/graficos-trabajo-pendiente-burndown-charts>

<http://www.proyectosagiles.org/lista-tareas-iteracion-sprint-backlog>

<http://www.sidar.org/recur/desdi/traduc/es/visitable/tecnicas/Prototyping.htm>

[http://www.ub.edu.ar/catedras/ingenieria/ing\\_software/ubftecwwdfd/cicloproy/ciclo.htm](http://www.ub.edu.ar/catedras/ingenieria/ing_software/ubftecwwdfd/cicloproy/ciclo.htm)  
<https://santimacnet.wordpress.com/page/25/>

<https://sites.google.com/site/jojoaa/analisis-de-sistemas/definicion-de-release-que-es-un-release>

Imagen disponible en sitio web: <http://procesossoftware.wikispaces.com/Modelo+Espiral>

INTELLEGO Desarrollo de software (s.l.) [En línea]. Mar. 2010. [Citado el 25 noviembre de 2014.] Disponible en internet en: <http://www.intellego.com.mx/es/sd-desarrollo-de-software>

LETELIER P, (2006), Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP), Técnica Administrativa, Buenos Aires, 5, recuperado de: [http://www.cyta.com.ar/ta0502/b\\_v5n2a1.htm](http://www.cyta.com.ar/ta0502/b_v5n2a1.htm)

LETELIER P, (2006), Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP), Técnica Administrativa, Buenos Aires, 5, recuperado de: [http://www.cyta.com.ar/ta0502/b\\_v5n2a1.htm](http://www.cyta.com.ar/ta0502/b_v5n2a1.htm)

METODOLOGÍA RUP (RATIONAL UNIFIED PROCESS) Disponible en línea en: <http://procesosdesoftware.wikispaces.com/METODOLOGIA+RUP>

METODOLOGIASAGILES. Carta de Steven Rakin a la revista Computer (s.l.) [En línea] (s.f) [Citado el 25 noviembre de 2014.] Disponible en internet en: <http://metodologiasagiles.wikispaces.com/Cr%C3%ADtica+de+a+las+Metodolog%C3%A4Das+%C3%81gile>

MOUNTAINGOATSOFTWARE. SCRUM (s.l.) [En línea]. Mar. 2010. [Citado el 25 noviembre de 2014.] Disponible en internet en: <http://www.mountaingoatsoftware.com/agile/scrum>

Página oficial OpenUp disponible en línea en: <http://epf.eclipse.org/wikis/openup/index.htm>

PROFA, R. (Febrero 8, 2012). Modelo en cascada o lineal secuencial. Febrero 2012, de isoft3cv2 Sitio web: <https://isoft3cv2.wordpress.com/2012/02/08/modelo-en-cascada-o-lineal-secuencial/>

PROYECTOSAGILES. Iterativo e incremental (s.l.) [En línea]. (s.f). [Citado el 25 noviembre de 2014.] Disponible en internet en: <http://www.proyectosagiles.org/desarrollo-iterativo-incremental>

PROYECTOSAGILES. Iterativo e incremental (s.l.) [En línea]. (s.f). [Citado el 25 noviembre de 2014.] Disponible en internet en: <http://www.proyectosagiles.org/desarrollo-iterativo-incremental>

PROYECTOSAGILES. Iterativo e incremental (s.l.) [En línea]. (s.f). [Citado el 25 noviembre de 2014.] Disponible en internet en: <http://www.proyectosagiles.org/desarrollo-iterativo-incremental>

Revista de la Facultad de Ingeniería Industrial 13(1): 70-74 (2010) UNMS ISSN: 1560-9146 (Impreso) / ISSN: 1810-9993 (Electrónico)  
[http://revistas.concytec.gob.pe/scielo.php?pid=S1810-99932010000200009&script=sci\\_arttext](http://revistas.concytec.gob.pe/scielo.php?pid=S1810-99932010000200009&script=sci_arttext)

RIOS S, Hinojoza C & Delgado R.. (2013). APLICACIÓN DE LA METODOLOGIA OPENUP EN EL DESARROLLO DEL SISTEMA DE DIFUSIÓN DE GESTIÓN DEL CONOCIMIENTO DE LA ESPE. 2013, de La Escuela Politécnica del Ejército Sitio web: <http://repositorio.espe.edu.ec/bitstream/21000/6316/1/AC-SISTEMAS-ESPE-047042.pdf>

VELASQUEZ, J. Desarrollo en Cascada (Waterfall) VS Desarrollo Agile-SCRUM. Disponible en: <http://www.northware.mx/wp-content/uploads/2013/04/Desarrollo-cascada-vs-Desarrollo-Agile.pdf>

VERSION ONE. Metodologías ágiles en el desarrollo en sus proyectos (s.l.) [En línea]. Mar. 2010. [Citado el 25 noviembre de 2014.] Disponible en internet en: <http://www.versionone.com/>

VERSIONONE, Scrum y XP (s.l.) [En línea]. Mar. 2010. [Citado el 25 noviembre de 2014.] Disponible en internet en: <http://www.versionone.com/pdf/2013-state-of-agile-survey.pdf>

WIKIPEDIA. Métodos ágiles (s.l.) [En línea]. Mar. 2010. [Citado el 25 noviembre de 2014.] Disponible en internet en: [http://es.wikipedia.org/wiki/Desarrollo\\_%C3%A1gil\\_de\\_software](http://es.wikipedia.org/wiki/Desarrollo_%C3%A1gil_de_software)

[www.agilealliance.com](http://www.agilealliance.com)

[www.pmoinformatica.com](http://www.pmoinformatica.com)

# **ANEXOS**

**Anexo A. Cronograma de actividades**

OBJETIVOS	ACTIVIDADES	MES 1				MES 2				MES 3				MES 4				MES 5				MES 6			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Proponer un nuevo método ágil a partir de las características propias de xp, scrum y openUp, integrando los puntos en común que existan entre ellas y adicionando características que lo complementen para definir la estructura que tendrá el nuevo método.	Definir y conocer los conceptos de que es una metodología ágil	■	■																						
	Definir y conocer los conceptos, roles, fases y artefactos de XP (extreme programing)			■	■	■																			
	Definir y conocer los conceptos, roles, fases y artefactos de SCRUM					■	■	■																	
	Definir y conocer los conceptos, roles, fases y artefactos de OpenUP						■	■																	
	Establecer las diferencias entre estas tres metodologías									■	■	■													
	Determinar las características de cada metodología para integrarlas en una sola.											■	■	■	■										
	Proponer una nueva metodología ágil que reúna las mejores características de XP, Scrum y OpenUp													■	■	■	■								
Evaluar el método propuesto para comprobar si cumple con	Evaluar el método ágil creado con la herramienta de evaluación y comparación 4-DAT para comprobar que cumple con todos los criterios para ser un método																					■	■	■	



los criterios que tienen los métodos ágiles utilizando la herramienta de comparación y análisis 4-DAT.	ágil																									
	Fin de la investigación																									

## Anexo B. Encuesta 1

La siguiente encuesta será realizada en los grupos de desarrollo de software de la universidad Francisco de Paula Santander, con el fin de hacer un diagnóstico inicial sobre el uso de metodologías ágiles.

1. ¿En qué grupo de desarrollo de software labora?
2. ¿Cuál es su rol dentro del grupo?
3. ¿Cómo calificaría usted su nivel de conocimiento en metodologías ágiles?

Muy Limitado  
Limitado  
Medio  
Extenso  
Muy Extenso

4. ¿Qué metodologías ágiles ha empleado?

Seleccione todas las que apliquen...

Scrum  
XP (eXtreme Programming)  
FDD (Feature-Driven Development)  
AD (Agile Database Techniques)  
AM (Agile Modeling)  
ASD (Adaptive Software Development)  
AUP (Agile Unified Process)  
Crystal  
OpenUp  
DSDM (Dynamic Systems Development Method)  
Lean Software Development  
TDD (Test-Driven Design)  
No lo sé  
No uso metodologías ágiles  
Otro

5. ¿En qué medida su metodología adoptada ha mejorado el rendimiento de sus equipos de trabajo?

Empeorado  
Sin Cambios  
Mejorado  
No lo sé

6. ¿En qué medida su metodología adoptada ha mejorado sus tiempos de entrega?

Empeorado  
Sin Cambio  
Mejorado  
No lo sé

7. ¿En qué medida su metodología adoptada ha mejorado sus costos?

Empeorado  
Sin Cambios  
Mejorado  
No lo sé

8. ¿En qué medida su metodología adoptada ha mejorado la calidad de su producto final?

Empeorado  
Sin Cambios  
Mejorado  
No lo sé

9. ¿Principal problema que tuvo al adoptar las metodologías ágiles?

Falta de experiencia en metodologías ágiles  
Poca administración en metodologías ágiles  
Filosofía y Cultura en la Organización  
Poco entrenamiento en metodologías ágiles  
Seguir utilizando la metodología anterior (Modelo de Cascada)  
Ningún Problema

10. ¿Seguirá empleando las metodologías ágiles en un futuro?

No  
Si  
No lo sé

## Anexo C: Encuesta 2

Con el fin de lograr una mejor evaluación y comparación del método creado se realizó una encuesta a los grupos de gestión de software de la UFPSO, donde se indaga acerca de los valores presentados y evaluados en 4-DAT, los encuestados deben responder preguntas correspondientes al manejo de los proyectos de desarrollo de software que han ejecutado.

6 ¿Cuántos integrantes hacen parte de su grupo de trabajo?

---

---

---

7 ¿En el desarrollo de sus proyectos las entregas se hacen en iteraciones cortas o solo se hace una entrega cuando el producto esté finalizado?

- Iteraciones cortas
- Una sola entrega

8 ¿Qué lenguaje de programación usa para la realización de sus proyectos?

---

---

---

9 ¿Realizan programación en parejas?

- Si
- No

10 ¿Cuántas horas trabajan semanalmente en la realización de un proyecto?

---

---

---

11 ¿Realizan reuniones diarias o semanales para verificar el progreso del proyecto?

- Reuniones diarias.
- Reuniones semanales
- No realizan reuniones

12 ¿El cliente hace parte de su equipo de trabajo?

- Si
- No

13 ¿Qué tipo de artefactos usa para el desarrollo de cada tarea?

---

---

---

14 ¿Usa algún artefacto exclusivo para la arquitectura y el diseño?

- Si
- No

15 ¿Realizan documentación del producto?

- Si
- No

## **Anexo D. Manual de usuario**

### **MANUAL DE USUARIO SISTEMA DE ADMINISTRACIÓN DE CITAS MÉDICAS INTRODUCCIÓN**

El sistema de administración de citas médicas es un sistema desarrollado completamente en entorno web, intuitivo, amigable, fácil de operar por todos los usuarios internos y externos de la organización Sanamedic, se ejecuta desde el navegador Google Chrome con conexión a internet. El sistema permite controlar y administrar los procesos relacionados con los reportes que la organización debe presentar según la resolución 1552 de 2013 del Ministerio de Salud la cual hace referencia a la oportunidad en las citas médicas.

Este manual permite concebir de manera perceptible el entorno gráfico y la operatividad del sistema ya que en él se explica detalladamente los pasos que deben seguir los usuarios externos e internos para el manejo general de las estructuras de las pantallas así como las funciones de los iconos básicos.

Este sistema permite realizar las funciones de:

Contar con un sistema de login para profesionales médicos, administrador de citas gerencia y regente de farmacia.

Cargar las bases de datos provenientes de las EPS Comfaorient y Ecoopsos.

Gestionar Citas

Reportar asistencia, ausencia o cancelación de la cita por parte de los usuarios.

Generar reportes de las citas médicas para una fecha estipulada por el administrador del sistema.

Generar los indicadores de oportunidad por EPS para una fecha estipulada por el administrador del sistema.

Gestionar usuarios administradores del sistema como profesionales médicos, administrador de citas y administrador del sistema.

Consultar usuarios con asignación de citas previas.

Permitir cargar la agenda del día y la agenda para una fecha estipulada para el administrador de citas y el profesional médico.

Consultar usuarios activos de las EPS Comfaorient y Ecoopsos.

Permitir recuperar la contraseña en caso de olvidarla, para este proceso se le enviara un correo al usuario con una nueva contraseña generada automáticamente.

Permitir a los usuarios externos (usuarios activos en las bases de datos de las EPS Comfaorient y Ecoopsos.) consultar, reservar y cancelar citas de medicina general y odontología entrando al sistema solo con su número de documento.

El alcance del sistema de administración de citas en la web, por el cual los usuarios podrán gestionar su cita médica, ofrece los servicios de consulta médica general y odontología pudiendo reservar, cancelar y consultar sus citas médicas.

#### Iconos básicos



Al hacer click sobre este icono el usuario externo o interno podrá modificar la cita médica la cual elija o un usuario según sea el caso.



Al hacer click sobre este icono el usuario externo o interno podrá eliminar la cita médica la cual elija o un usuario según sea el caso.



Al aparecer este icono en la pantalla indica que su solicitud no fue procesada debido a un error inesperado.



Al aparecer este icono en la pantalla indica que su solicitud no fue procesada debido a un error de ingreso de datos.



Al aparecer este icono en la pantalla indica que su solicitud fue procesada con éxito.



Al aparecer este icono en la pantalla indica que la importación de la base de datos de la EPS que haya seleccionado fue procesada con éxito.



Al aparecer este icono en la pantalla indica que la importación de la base de datos no ha sido procesada debido a que el archivo seleccionado no tiene el formato correspondiente o no pertenece a la EPS seleccionada.

El sistema de administración de citas médicas cuenta con cuatro tipos de usuarios internos los cuales son el administrador de citas, gerencia, regente de farmacia y los profesionales médicos además del ingreso por parte de los usuarios activos de las EPS Comfaorient y

Ecoopsos. A continuación se mostrara el ingreso al sistema para cada tipo de usuario además de los procesos que puede llevar a cabo una vez haya iniciado sesión.

### Ingreso al sistema

Para ingresar al sistema de administración de citas médicas debe dar click sobre el icono de Google Chrome y digitar la dirección [www.citas.ipssanamedic.com.co](http://www.citas.ipssanamedic.com.co) y se abrirá la siguiente pantalla.

### Index sistema de administración de citas medicas

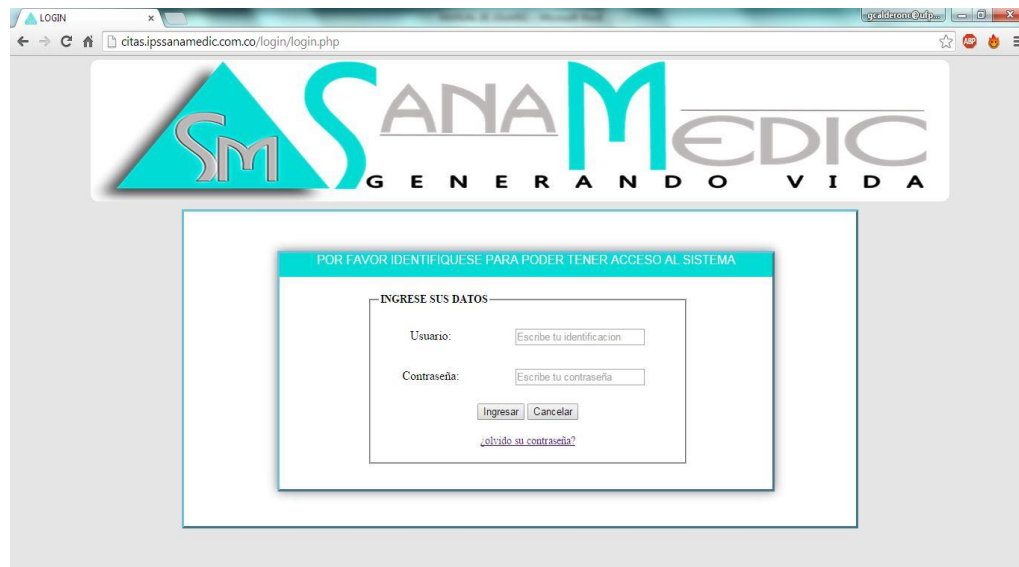


En ella encontrara el link mis citas por el cual ingresan al sistema los usuarios externos pertenecientes a las EPS Comfaorient y Ecoopsos, luego está el link profesional por el cual ingresan los profesionales médicos, usuarios donde ingresan los usuarios internos administrador de citas, gerencia y regente de farmacia.

Todos los links de los usuarios internos del sistema redirigen al login con la siguiente pantalla.



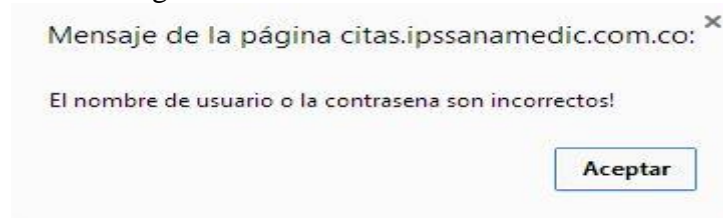
## Login – usuarios internos



En el campo de usuario debe ingresar el número de documento del usuario y luego la contraseña, este campo es sensible a mayúsculas y minúsculas por lo cual el usuario debe tener en cuenta al momento de digitarla. Luego dar click en el botón ingresar o cancelar en caso de no querer ingresar al sistema.

Cuando se ingresa el documento o la contraseña incorrecta aparecerá la siguiente alerta que indica el error.

Alerta datos incorrectos al ingresar al sistema



En caso de que el usuario haya olvidado su contraseña tiene la posibilidad de generar una nueva contraseña que se enviara al correo con el cual esté registrado en el sistema.

Para recuperarla debe dar click en el link [¿olvidó su contraseña?](#) Como se muestra a continuación.

## Login – cambio de contraseña



LOGIN

citassanamedic.com.co/login/login.php

**Sm SANAM MEDIC**  
GENERANDO VIDA

POR FAVOR IDENTIFIQUESE PARA PODER TENER ACCESO AL SISTEMA

INGRESE SUS DATOS


Usuario:

Contraseña:

[¿olvidó su contraseña?](#)

Luego de haber dado click aparecerá la siguiente pantalla

## Cambio de contraseña



RECUPERACION DE CONI

citassanamedic.com.co/login/recuperar\_contraseña.php

**Sm SANAM MEDIC**  
GENERANDO VIDA

POR FAVOR INGRESE SU DOCUMENTO PARA REESTABLECER SU CONTRASEÑA

INGRESE SU DOCUMENTO CON EL CUAL ACCEDE AL SISTEMA

Usuario:

Estimado usuario recuerde que para poder reestablecer su contraseña debe estar registrado como usuario activo en la base de datos. se le enviara a su correo una nueva contraseña.

En esta pantalla el usuario debe digitar su número de documento y dar click en el botón enviar o presionar enter, o dar click en el botón cancelar en caso de no querer generar la nueva contraseña. En caso de que el usuario no este registrado en el sistema aparecerá una alerta de error la cual indica que el usuario no está registrado en la base de datos, en caso contrario aparecerá una alerta donde se mostrara el correo al cual se ha enviado la nueva contraseña.

## USUARIO ADMINISTRADOR DE CITAS

Cuando el usuario administrador de citas ingresa al sistema de forma correcta aparece la siguiente pantalla donde se muestran los links para realizar los respectivos procesos.

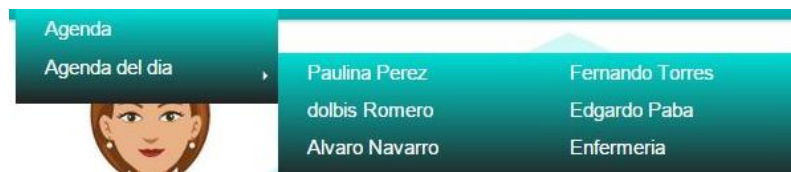
Index – Usuario administrador de citas



### Agendas

Al poner el cursor sobre el link Agenda se despliega el menú de agenda y agenda del día con el listado de médicos registrados actualmente en el sistema.

### Menú Agendas



Al dar click sobre cualquiera de los médicos aparece la agenda del día según sea profesional de medicina general con rango de tiempo de 15 minutos o profesional odontológico con rango de tiempo cada 20 minutos desde las 07:00 AM hasta las 5:30 PM y 05:40 PM respectivamente.

## Agenda del día – Medicina General

AGENDA POR PROFESION: x gcalderonc@ufp...

← → ↻ 🏠 | citas.ipssanamedic.com.co/citas/agendadeldia.php?id=3

**SM SANAM MEDIC**  
G E N E R A N D O V I D A

Inicio Agendas Citas Consultar Usuarios Cerrar Sesión

Agenda del día, profesional: Dolbis Romero Ospino

Hora	Documento	Nombre	Apellido	Telefono	Fecha Consulta	Fecha Asignacion	Doctor(a)	Asistencia	Guardar
07:00 a.m.	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	2015-01-14	2015-01-14	3	pendiente	Guardar
07:15 a.m.	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	2015-01-14	2015-01-14	3	pendiente	Guardar
07:30 a.m.	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	2015-01-14	2015-01-14	3	pendiente	Guardar
07:45 a.m.	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	2015-01-14	2015-01-14	3	pendiente	Guardar
08:00 a.m.	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	2015-01-14	2015-01-14	3	pendiente	Guardar
08:15 a.m.	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	2015-01-14	2015-01-14	3	pendiente	Guardar
08:30 a.m.	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	2015-01-14	2015-01-14	3	pendiente	Guardar

## Agenda del día – Odontología

AGENDA POR PROFESION: x gcalderonc@ufp...

← → ↻ 🏠 | citas.ipssanamedic.com.co/citas/agendadeldia.php?id=1

**SM SANAM MEDIC**  
G E N E R A N D O V I D A

Inicio Agendas Citas Consultar Usuarios Cerrar Sesión

Agenda del día, profesional: Paulina Perez

Hora	Documento	Nombre	Apellido	Telefono	Fecha Consulta	Fecha Asignacion	Doctor(a)	Asistencia	Guardar
07:00 a.m.	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	2015-01-14	2015-01-14	1	pendiente	Guardar
07:20 a.m.	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	2015-01-14	2015-01-14	1	pendiente	Guardar
07:40 a.m.	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	2015-01-14	2015-01-14	1	pendiente	Guardar
08:00 a.m.	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	2015-01-14	2015-01-14	1	pendiente	Guardar
08:20 a.m.	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	2015-01-14	2015-01-14	1	pendiente	Guardar
08:40 a.m.	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	2015-01-14	2015-01-14	1	pendiente	Guardar
09:00 a.m.	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	2015-01-14	2015-01-14	1	pendiente	Guardar

## Asignar citas

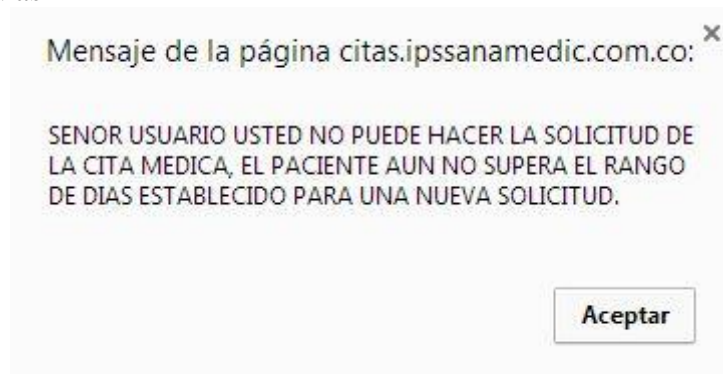
Para la asignación de las citas existen tres opciones, desde el link agenda del día seleccionando el profesional médico. Desde el link agenda o desde el link nueva cita y consultar usuarios que llevan a la misma pantalla ejecutando el mismo procedimiento.

Para asignar una cita desde la agenda del día del profesional solo se debe llenar el campo documento y el campo teléfono y dar click en el botón Guardar, si la información suministrada es correcta y el usuario no cuenta con citas en estado pendiente, ausente o asistido dentro del rango de días los cuales no puede asignar las citas, aparecerá la alerta según sea el caso.

Para medicina general y odontología si un usuario no asiste a la cita médica dicha cita llevara el estado 'ausente' y deberán pasar 8 días para poder signar una nueva cita.

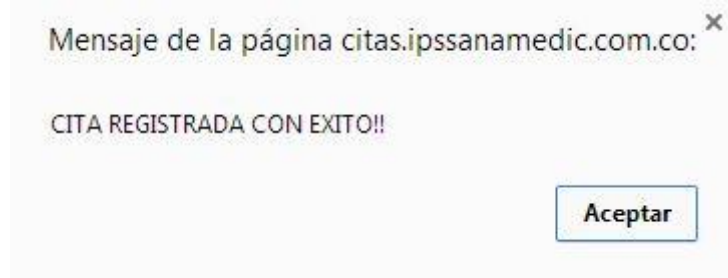
Si el usuario asiste el estado cambiara a 'asistió' y deberán pasar 20 días para una nueva solicitud, en caso de tener una cita en estado pendiente el cual es el primero que se asigna automáticamente al momento de asignar una cita médica igualmente deberán pasar 20 días para poder realizar una nueva solicitud esto en el caso de medicina general, para odontología deberán pasar solo 8 días para una nueva solicitud según sea el caso. Cabe resaltar que el rango de días para una nueva solicitud de citas es establecido de acuerdo a políticas internas de la organización Sanamedic S.A.S. la siguiente imagen muestra la alerta que saldrá en caso de cumplirse cualquiera de los hechos anteriormente dichos.

## Alerta rango de días



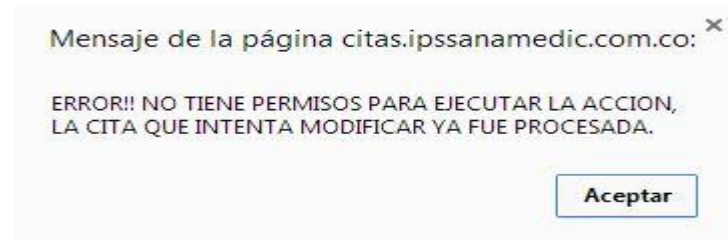
Si el paciente no presenta ningún inconveniente para la asignación de su cita médica se mostrara la siguiente alerta.

Alerta cita guardada exitosamente



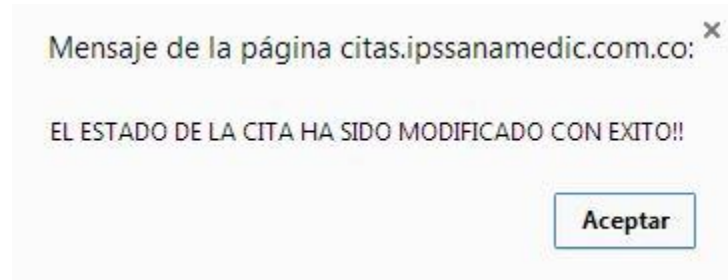
El estado lo cambia el usuario administrador de citas al momento de cumplirse la fecha de asignación y la hora establecida de la cita médica, cabe resaltar que solo podrá modificar el estado si la cita está en estado 'pendiente' de lo contrario aparecerá la siguiente alerta.

Alerta error al modificar



En caso contrario se mostrara la siguiente alerta

Alerta cita modificada con éxito



En el caso de dar click sobre el link agenda se mostrara la siguiente pantalla

## Consultar agenda

CONSULTAR AGENDA DE CITAS MEDICAS

Estimado usuario recuerde que para realizar la respectiva consulta de la agenda de citas medicas debe seleccionar el profesional y la fecha a consultar.

Seleccione el profesional a consultar:

Por favor seleccione la fecha:

SANAMEDIC S.A.S Cra 13 No. 12-66 Centro - Ocaña  
Horario de atención: Lunes a viernes de 07:00 am a 12:00 m. y de 02:00 pm a 06:00 pm  
Telefono 562.2020 Celular 317 668 8218  
E-mail: sanamedic@pssanamedic.com.co  
© 2014

Se debe seleccionar el profesional médico y la fecha de la agenda a consultar y dar click en el botón consultar o presionar enter y se desplegará la siguiente pantalla, en caso de no querer ejecutar la acción se debe dar click en cancelar.

## Agenda profesional en odontología

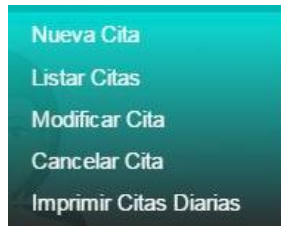
Agenda del día, profesional: Fernando Torres

Hora	Documento	Nombre	Apellido	Telefono	Fecha Consulta	Fecha Asignacion	Doctor(a)	Asistencia	Guardar
07:00 a.m.	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	2015-01-16	2014-12-17	2	pendiente	<input type="button" value="Guardar"/>
07:20 a.m.	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	2015-01-16	2014-12-17	2	pendiente	<input type="button" value="Guardar"/>
07:40 a.m.	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	2015-01-16	2014-12-17	2	pendiente	<input type="button" value="Guardar"/>
08:00 a.m.	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	2015-01-16	2014-12-17	2	pendiente	<input type="button" value="Guardar"/>
08:20 a.m.	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	2015-01-16	2014-12-17	2	pendiente	<input type="button" value="Guardar"/>
08:40 a.m.	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	2015-01-16	2014-12-17	2	pendiente	<input type="button" value="Guardar"/>
09:00 a.m.	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	2015-01-16	2014-12-17	2	pendiente	<input type="button" value="Guardar"/>

El procedimiento para asignar una cita médica es el mismo que se aplica en la agenda del día.

Al pasar el cursor sobre el link citas se desplegará el siguiente menú

Menú citas



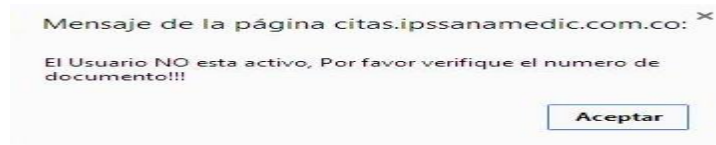
Existe otra manera de asignar una cita médica en medicina general u odontología dando click sobre el link 'nueva cita', en el cual se deberá verificar primero que el paciente esté activo en la base de datos.

Consultar usuarios activos



Se debe ingresar el número de documento del paciente y dar click en consultar o presionar enter. Si el usuario externo no está activo en la base de datos aparecerá la siguiente alerta.

Alerta usuario no activo



En caso contrario se desplegara la siguiente pantalla.



## Datos usuario externo activo

Carnet: 98415463	Identificacion N: 1004862677
Nombres: JESUS AURELIO FUENTES MARTINEZ	
Fecha de nacimiento: 24/12/1990	Sexo: M
Dirección:	Teléfono:
EPS: comfariante	

REGISTRO DE CITAS ASIGNADAS HASTA LA FECHA:  
El usuario no ha solicitado ninguna cita hasta el momento

[Volver](#) [Asignar Cita Medicina General](#) [Asignar Cita Odontologica](#) [Asignar Cita prioritaria](#) [Imprimir](#)

En esta pantalla se puede tener acceso a la información del paciente y al registro de citas que este haya solicitado en la IPS, en caso de no haber solicitado ninguna se mostrara el mensaje que se puede apreciar en la imagen 'el usuario no ha solicitado ninguna cita hasta el momento'. En los botones de abajo se pueden ver 4 opciones, asignar una cita de medicina general u odontología teniendo en cuenta las reglas para asignarla que son los estados y el rango de días establecidos explicados anteriormente. Igualmente esta la opción de imprimir el documento y asignar una cita prioritaria.

Las citas prioritarias solo se pueden asignar en casos especiales donde el paciente haya asistido a su última cita o se encuentre en estado 'ausente' y no hayan pasado los días requeridos para una nueva solicitud y este necesite de una consulta urgente en medicina general u odontología. Al dar click en el botón 'asignar cita medicina general' u 'asignar cita odontología' aparecerá la siguiente pantalla.

## Formulario ingresar cita

INFORMACION REQUERIDA PARA LA ASIGNACION DE LA CITA - MEDICINA GENERAL

POR FAVOR INGRESE TODA LA INFORMACION REQUERIDA.

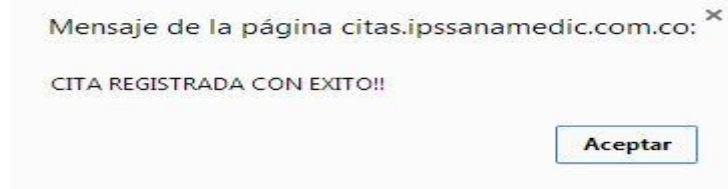
Documento: 1004862677	Fecha de la Consulta: 22/12/13
Nombre: JESUS	Fecha asignacion de la Cita: 24/12/13
Apellido: FUENTES	Médico: Seleccione...
Teléfono:	Hora de la Cita:
Sexo: M	EPS:

[Guardar](#) [Cancelar](#) [Volver](#)

El usuario solo tendrá que llenar los campos de teléfono en caso de no tenerlo, fecha de la consulta, fecha de asignación de la cita, el médico y la hora de la cita y dar click en el botón guardar o presionar enter. El botón cancelar reanuda todos los campos al estado inicial. Si

el proceso tiene éxito y no existen citas anteriormente asignadas con el profesional en la fecha estipulada se mostrara la siguiente alerta.

Alerta cita guardada exitosamente



En caso contrario se mostrara la siguiente alerta

Alerta horario no disponible



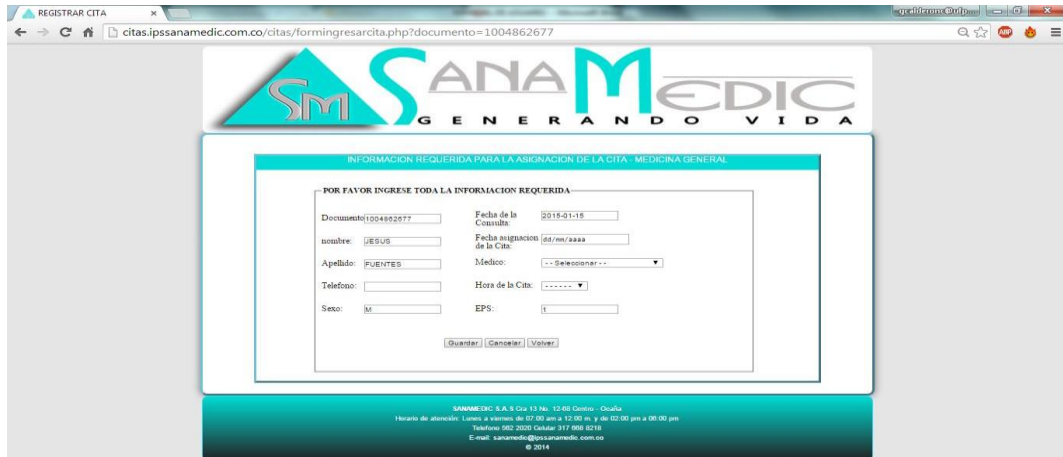
Al dar click sobre el botón 'asignar cita prioritaria' se mostrará la siguiente pantalla

Citas prioritarias



Se debe seleccionar el tipo de cita de la lista y digitar el motivo de la cita prioritaria y dar click en asignar cita y se desplegará la siguiente pantalla.

## Formulario ingreso de citas prioritarias



REGISTRAR CITA

citassanamedic.com.co/citas/formingresarcita.php?documento=1004862677

**SANAMEDIC**  
GENERANDO VIDA

INFORMACION REQUERIDA PARA LA ASIGNACION DE LA CITA - MEDICINA GENERAL

POR FAVOR INGRESE TODA LA INFORMACION REQUERIDA

Documento: 1004862677	Fecha de la Consulta: 2018-01-16
Nombre: JESUS	Fecha asignacion de la Cita: 08/01/2018
Apellido: FUENTES	Medico: -- Seleccionar --
Telefono:	Hora de la Cita: .....
Sexo: M	EPS: S

SANAMEDIC S.A.S. Cra 13 No. 12-60 Ciudad - Cali  
Horario de atención: Lunes a viernes de 07:00 am a 12:00 pm, y de 02:00 pm a 08:00 pm  
Telefono: 862 2020 Ciudad 317 606 8216  
E-mail: sanamedic@citassanamedic.com.co  
© 2014

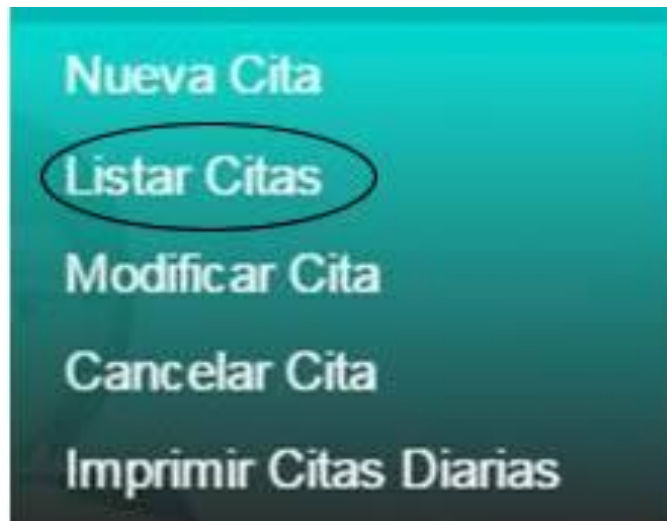
El proceso para asignar la cita es el mismo anteriormente explicado.

## Modificar citas

Existen dos maneras de modificar la cita de un paciente y es mediante el link 'listar citas' o 'modificar cita'.

Al dar click sobre el link listar citas se mostrará la siguiente pantalla

## Listar citas



## Listado de citas

Hora	Documento	Nombre	Apellido	Telefono	Fecha Consulta	Fecha Asignacion	Doctor(a)	Asistencia	Cancelar	Modificar
07:45:00	1004862675	JOHAN	FUENTES	86868	2015-01-15	2015-01-15	Dolbis Romero Ospino	ausente		
08:15:00	1004862693	DIOMEDES	PEREZ	3163269417	2015-01-15	2015-01-15	Dolbis Romero Ospino	asistio		
14:30:00	1004862677	JESUS	FUENTES	3163269417	2015-01-15	2015-01-16	Edgardo Paba Gonzales	pendiente		

En esta pantalla se mostrará todas las citas asignadas con las opciones de cancelación y modificación. Además de un filtro para buscar las citas de un paciente ya sea por su número de documento, nombre o apellido.

Para proceder a modificar la cita se debe dar click sobre el botón el cual mostrara la siguiente alerta en caso de que la cita ya no esté en estado pendiente.

Alerta error al modificar

Mensaje de la página citas.ipssanamedic.com.co: ✕

ERROR!! NO TIENE PERMISOS PARA EJECUTAR LA ACCION.

Aceptar

En caso contrario se desplegara la siguiente pantalla

## Modificar cita – opción 1

Documento	Nombre	Apellido	Telefono	Fecha Consulta	Fecha Asignacion	Doctor(a)	Hora	Asistencia
1004862677	JESUS	FUENTES	3163269417	2015-01-15	16/81/2015	-- Seleccionar --	-----	pendiente

Solo se puede modificar la fecha de la asignación de la cita, se deberá escoger de nuevo un profesional médico y la hora de la cita y dar click en el botón guardar o presionar enter, y de inmediato se mostrara la alerta

Alerta verificación al modificar

Mensaje de la página citas.ipssanamedic.com.co: ✕

¿Estas seguro que desea modificar los datos de esta cita?

Si se da click en aceptar y el horario escogido no tiene citas pendientes se mostrara la alerta de éxito. De lo contrario si se da click en el botón cancelar no se ejecutará ninguna acción.

Alerta cita modificada

Mensaje de la página citas.ipssanamedic.com.co: ✕

LOS DATOS HAN SIDO MODIFICADOS CON EXITO!

Evita que esta página cree cuadros de diálogo adicionales.

Si el nuevo horario escogido no está disponible se mostrará la alerta

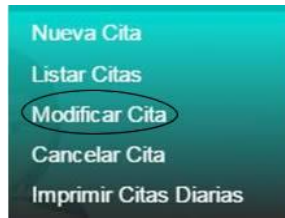
Alerta error al modificar cita

Mensaje de la página citas.ipssanamedic.com.co: ✕

ERROR AL MODIFICAR, Horario NO disponible

Para modificar una cita por la opción 2 se debe dar click en el botón ‘modificar cita’

Link modificar cita – opción 2



Se mostrará la siguiente pantalla

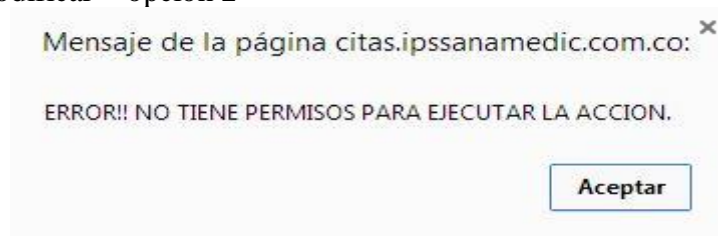
Consultar citas a modificar



Fuente: Pasante

Se debe digitar el número de documento del paciente sin caracteres especiales ni espacios, si el usuario no está activo en la base de datos o presenta citas con estado diferente a pendiente se mostrará la siguiente alerta.

Alerta error al modificar – opción 2



Si no se presentan inconvenientes se mostrará la siguiente pantalla

## Modificar cita – opción 2



El procedimiento para modificar la cita es el mismo anteriormente mencionado en opción 1.

## Cancelar citas

Existen dos opciones para cancelar las citas de un paciente la opción 1 es dando click en el link listar citas y la opción dos es dar click en la opción cancelar cita. A continuación se presentara el proceso a seguir para la opción 1.

Se debe dar click en la opción listar citas


## Link cancelar citas – opción 1



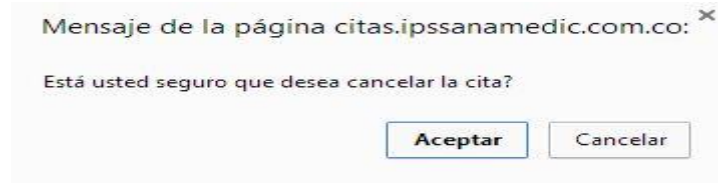
Luego se mostrará la siguiente pantalla

## Listado de citas



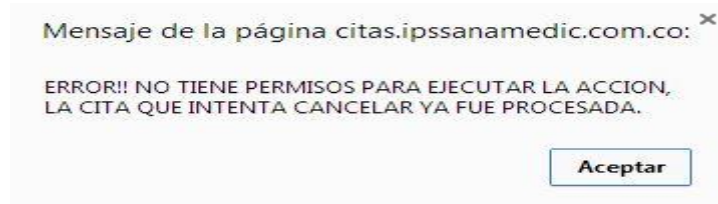
Para proceder con la cancelación hacer click sobre el icono  y se mostrar la siguiente alerta

#### Verificación al cancelar cita



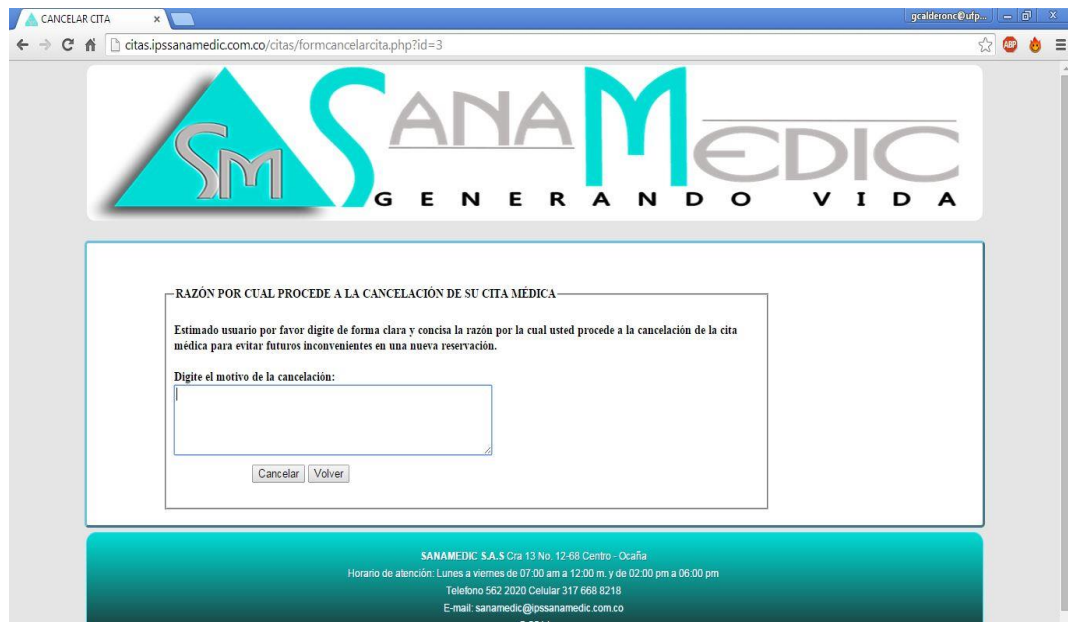
Si se da click en el botón aceptar se prosigue con la cancelación de lo contrario si se da click en el botón cancelar no se ejecutará ninguna acción, Si la cita no está en estado pendiente, sino que ya fue procesada se mostrara el siguiente pantallazo.

#### Alerta error al cancelar cita



En caso contrario si la cita está en estado pendiente procederá a la cancelación de la misma, en la siguiente pantalla.

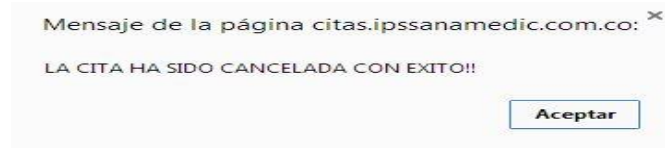
#### Cancelar cita – opción 1





Se debe digitar el motivo de la cancelación de la cita y dar click en el botón cancelar o dar enter y debe aparecer la siguiente alerta.

Cita cancelada con éxito



Para cancelar una cita con la opción dos se debe dar click en el link cancelar cita

Link cancelar cita – opción 2



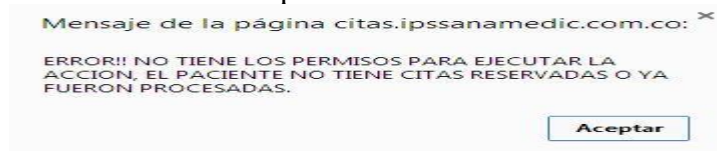
Posteriormente se desplegara la siguiente pantalla

Consultar citas a eliminar – opción 2



Se debe digitar el número de documento del paciente para verificar que existan citas y que estas estén solo en estado pendiente de lo contrario aparecerá la siguiente alerta.

**magen N83.** Error al cancelar citas – opción 2



Si el paciente tiene citas en estado pendiente aparecerá la siguiente pantalla

## Formulario cancelar citas – opción 2



Para proceder a la cancelación de la cita médica se ejecutan los pasos anteriormente descritos.

Imprimir listado de citas diarias por profesional

Para proceder a imprimir el listado de citas diarias por profesional se debe dar click en el link imprimir citas diarias.

Link imprimir citas del día por profesional



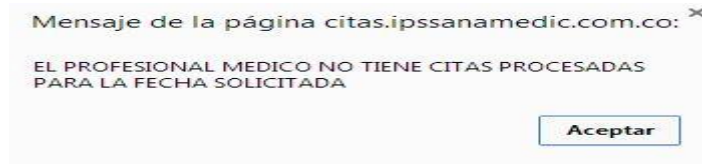
Posteriormente se desplegara la siguiente pantalla

Formulario imprimir citas del día por profesional



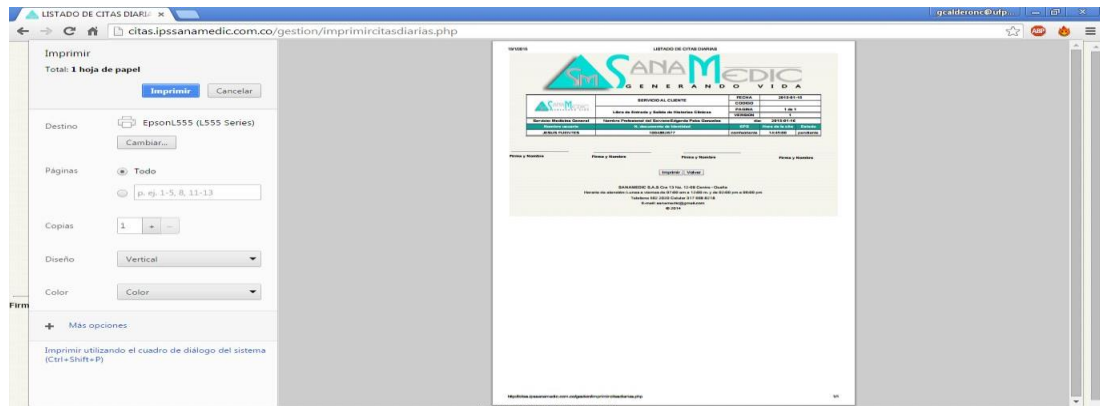
Se debe seleccionar un médico de la lista y el la fecha la cual se desea consultar y dar click en el botón consultar o presionar enter. Si el medico no tiene citas para la fecha solicitada se mostrara la siguiente alerta

### Alerta profesional sin citas



En caso contrario se cargara la página automáticamente para imprimir como se muestra a continuación.

### Imprimir citas del día por profesional



### Migrar bases de datos

El sistema de administración de citas médicas necesita alimentarse cada mes con una base de datos nueva la cual proviene de las dos EPS actualmente en convenio con la organización que son Comfaorient y Ecoopsos. Para proceder con la migración de datos se sigue el siguiente procedimiento.

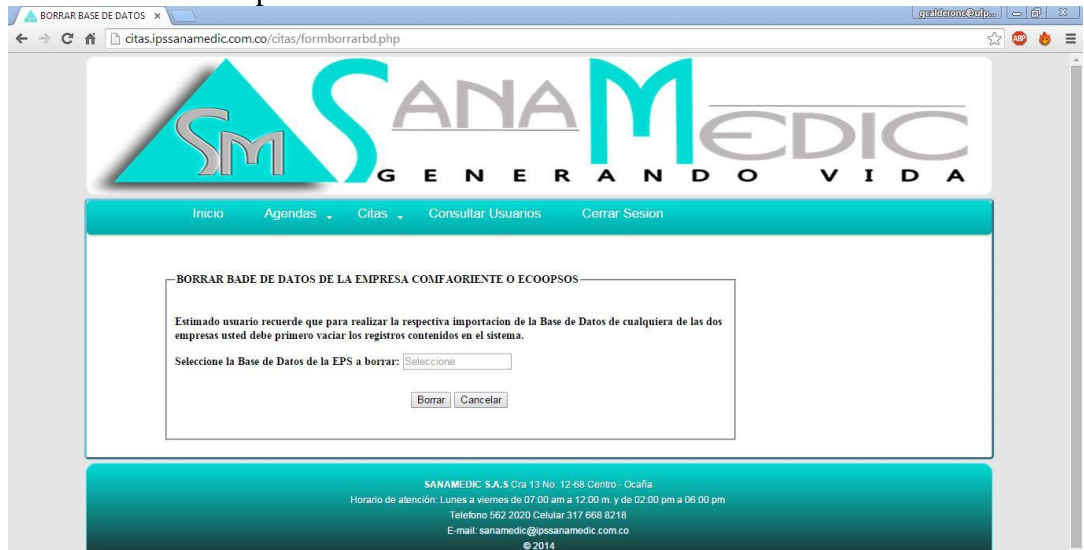
Dar click en el link 'cargar BD'

Link cargar base de datos



Luego de haber dado click en este link aparecerá la siguiente pantalla

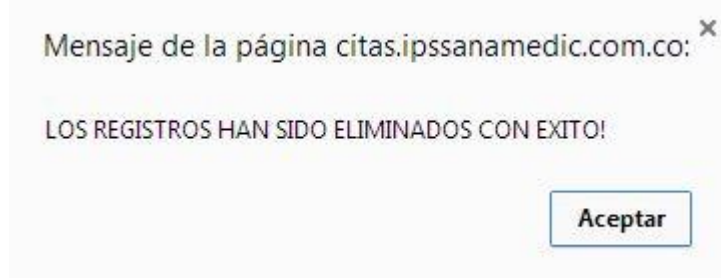
Eliminar base de datos por EPS



En esta pantalla se debe seleccionar primero la base de datos que va a eliminar hay que tener en cuenta que debe ser la misma que se va a proceder a migrar. Y dar click en el botón borrar o dar enter.

Si todo ocurre de la manera esperada aparecerá la siguiente alerta.

Alerta registros eliminados con éxito



Luego se dar click en el botón aceptar o haber presionado enter se desplegara la siguiente pantalla.

Cargar base de datos por EPS



Para proceder a la migración de datos se debe seleccionar la misma EPS a la cual elimino los registros anteriormente y seleccionar el archivo a subir. Cabe resaltar que el archivo debe ser en formato TXT y debe corresponder a la base de datos que seleccione de la lista.

Luego se debe dar click en el botón importar o presionar enter y aparecerá el siguiente mensaje si los datos fueron importados correctamente.

## Importación exitosa



Si se ha seleccionado un archivo con un formato diferente a TXT aparecerá el siguiente mensaje y se deberá hacer nuevamente el proceso y seleccionar un archivo correcto.

## Importación fallida



## PROFESIONALES MÉDICOS

Los profesionales médicos hacen parte de los usuarios internos del sistema, sus funciones son consultar agenda del día, agenda de cualquier día, y modificar sus datos personales.

Consultar agendas

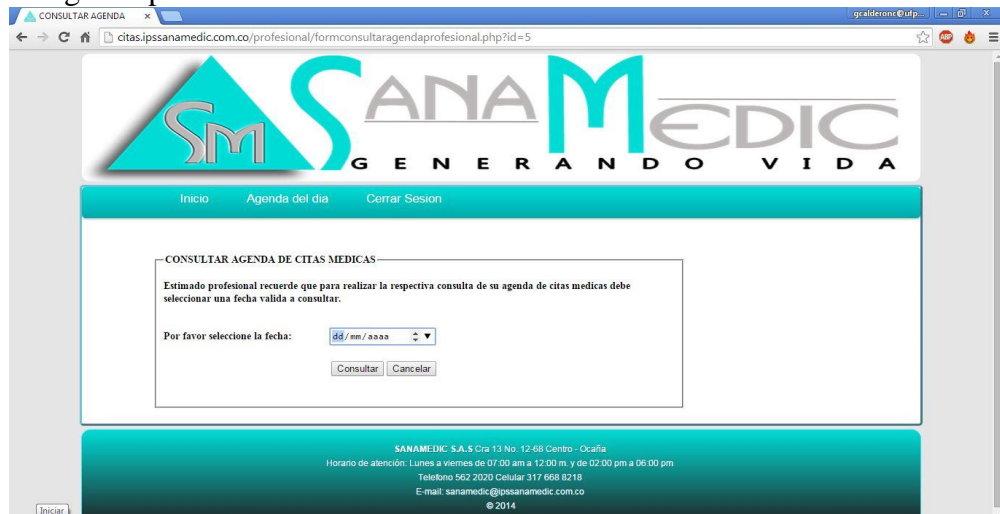
Para consultar la agenda de cualquier día solo deben dar click en el link agenda.

Index – Profesional medico



Al dar click se desplegará la siguiente pantalla

Consultar agenda profesional medico



Se debe seleccionar una fecha a la cual se hará la consulta de la agenda y se desplegara la siguiente pantalla.

## Imagen N97. Agenda profesional medico

The screenshot shows a web browser window with the URL `citasi.pssanamedic.com.co/profesional/agendaprofesional.php`. The page features the SANAMEDIC logo and a navigation menu with options: Inicio, Agenda, Agenda del día, Datos Personales, and Cerrar Sesión. The main heading reads "Agenda para la fecha 2015-01-21, profesional: Alvaro Navarro Bonett". Below this, there is a "Listar" dropdown menu set to "10" and a "Buscar:" search box. A table displays the appointment schedule for the day.

Hora	Documento	Nombre	Apellido	Telefono	Fecha Consulta	Fecha Asignacion	Doctor(a)	Estado
08:00 a.m.	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	2015-01-21	Alvaro Navarro Bonett	
08:15 a.m.	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	2015-01-21	Alvaro Navarro Bonett	
08:30 a.m.	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	2015-01-21	Alvaro Navarro Bonett	
08:45 a.m.	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	2015-01-21	Alvaro Navarro Bonett	

Aquí aparecerán todas las citas que el profesional tenga asignadas para la fecha estipulada. Para consultar la agenda del día se debe dar click en el link 'agenda del día'

Link agenda del día profesional medico

The screenshot shows a web browser window with the URL `citasi.pssanamedic.com.co/profesional/index.php`. The page features the SANAMEDIC logo and a navigation menu with options: Agenda, Agenda del día, Datos Personales, and Cerrar Sesión. The main heading reads "Bienvenido (a) Alvaro Navarro Bonett". Below this, there is a welcome message: "Hoy es Jueves, 15 de Enero de 2015". At the bottom, there is contact information for SANAMEDIC S.A. and a copyright notice for 2014.

SANAMEDIC S.A. & Cía 13 No. 12-68 Centro - Ocaña  
Horario de atención: Lunes a viernes de 07:00 am a 12:00 m. y de 02:00 pm a 06:00 pm  
Telefono 512 2020 Celular 317 668 8218  
E-mail: sanamedic@pssanamedic.com.co  
© 2014

Al haber ejecutado la acción se mostrara la siguiente pantalla con todas las citas que el profesional médico tenga asignadas para el día.



## Agenda del día profesional medico

Hora	Documento	Nombre	Apellido	Telefono	Fecha Consulta	Fecha Asignacion	Doctor(a)	Estado
08:00 a.m.						2015-01-15	Alvaro Navarro Bonett	
08:15 a.m.						2015-01-15	Alvaro Navarro Bonett	
08:30 a.m.						2015-01-15	Alvaro Navarro Bonett	
08:45 a.m.						2015-01-15	Alvaro Navarro Bonett	

## GERENCIA

El usuario gerencia cuenta con las opciones de ingresar, modificar y eliminar usuarios, profesionales médicos, consultar usuarios externos, generar reportes e indicadores de oportunidad.

### Ingresar usuarios

Para el registro de usuarios internos de la organización se debe dar click en el link usuarios y luego escoger de la lista 'agregar usuarios'.

### Link agregar usuarios

Usuarios  
Profesionales  
Usuarios Externos  
Informes  
Cerrar Sesión

Agregar Usuarios  
Listar Usuarios

Bienvenido (a)  
Yaneth Mora Castro  
Hoy es Jueves, 15 de Enero de 2015

SANAMEDIC S.A.S. Cra 13 No. 12-68 Centro - Ocaña  
Horario de atención: Lunes a viernes de 07:00 am a 12:00 m. y de 02:00 pm a 06:00 pm

Al haber dado click en el enlace se desplegara el formulario de ingreso de usuarios.

### Formulario agregar usuario

REGISTRAR PROFESIONAL

citass.ipssanamedic.com.co/admin/formagregarusuario.php

GENERANDO VIDA

INFORMACION REQUERIDA PARA EL REGISTRO DE USUARIOS

POR FAVOR INGRESE TODA LA INFORMACION REQUERIDA

Documento:

Nombre:

Apellido:

Sexo:

Tipo Usuario Administrador:

Fecha de Nacimiento:

Lugar de Nacimiento:

Telefono o Celular:

Direccion:

Correo Electronico:

Clave:

Repetir Clave:

SANAMEDIC S.A.S Cra 13 No. 12-68 Centro - Ocaña  
Horario de atención: Lunes a viernes de 07:00 am a 12:00 m. y de 02:00 pm a 06:00 pm  
Telefono 262-2020 Celular 317-688-9218

Se deben llenar todos los campos y dar click en el botón guardar o presionar enter, el botón cancelar limpia todos los campos y los restaura al estado inicial. Si los datos han sido guardados con éxito aparecerá la siguiente alerta

### Alerta usuario registrado exitosamente

Mensaje de la página citass.ipssanamedic.com.co: ×

Usuario registrado CON EXITO

### Modificar usuarios

Para modificar los datos de un usuario se debe dar click en el link listar usuarios

### Link listar usuarios

GESTION DE CITAS

citass.ipssanamedic.com.co/admin/index.php

SANAMEDIC GENERANDO VIDA


Usuarios Profesionales Usuarios Externos Informes Cerrar Sesión

Agregar Usuarios

Listar Usuarios

Bienvenido (a)  
Yaneth Mora Castro  
Hoy es Jueves, 15 de Enero de 2015

SANAMEDIC S.A.S Cra 13 No. 12-68 Centro - Ocaña  
Horario de atención: Lunes a viernes de 07:00 am a 12:00 m. y de 02:00 pm a 06:00 pm

Al hacer click sobre el enlace se desplegará la lista de los usuarios existente y la opción de modificar. Para esto se debe dar click en el icono  del usuario el cual se modificará.

### Listado de usuarios



Documento	Nombre	Fecha de Nacimiento	Lugar de Nacimiento	Edad	Telefono	direccion	correo	Tipo Usuario	Eliminar	Modificar
37331146	Yaneth Mora Castro	1976-11-28	Ocaña	38	3163321373	Cra 9 N 10-03	gerencia@ipssanamedic.com.co	Gerencia		
1091661166	Monica Amaya Romero	1989-05-05	Teorama (N.S)	25	3168109433	Cra 17A N 3-01 B. Modelo	farmacia@ipssanamedic.com.co	Regente de farmacia		
1091663850	Gerardo Calderon	1990-03-19	San Alberto (cesar)	24	3163269417	calle 8 #16-62(El llano)	gerar1790@gmail.com	Administrador de citas		
1091664435	Sandra Milena Perez Velasquez	1990-07-12	Ocaña (N.S)	24	3106730942	Barrio Nueva España	admincitas@ipssanamedic.com.co	Administrador de citas		

Luego aparecerá el formulario con los datos del usuario con la posibilidad de modificarse, para esto se debe dar click en el botón guardar, el botón cancelar restaura cualquier modificación hecha en las cajas de texto.

### Formulario modificar usuario



INFORMACION REQUERIDA PARA LA ACTUALIZACION DE USUARIOS

POR FAVOR INGRESE TODA LA INFORMACION REQUERIDA

Documento:  Lugar de Nacimiento:

Nombre:  Telefono o Celular:

Apellido:  Direccion:

Sexo:

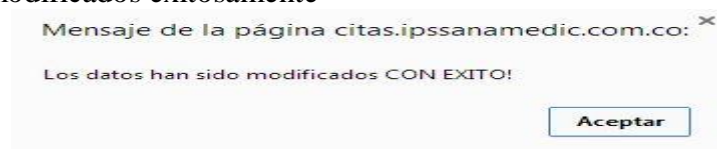
Tipo Usuario:  Correo Electronico:

Fecha de Nacimiento:  Clave:

Repetir Clave:

Al guardar los datos en la base de datos correctamente se mostrara la siguiente alerta


### Alerta usuarios modificados exitosamente



Mensaje de la página citas.ipssanamedic.com.co: x

Los datos han sido modificados CON EXITO!

## Eliminar usuarios

Para eliminar un usuario de la base de datos se debe dar click sobre el icono  del usuario el cual se quiera eliminar.

## Eliminar usuario



Inicio Usuarios Profesionales Usuarios Externos Informes Cerrar Sesión

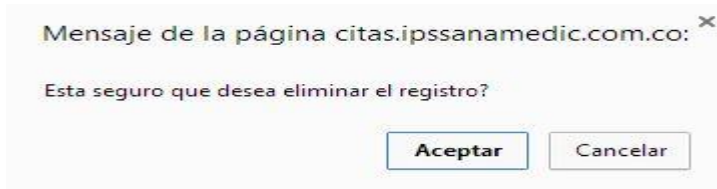
Listar 10 Entradas

Buscar:

Documento	Nombre	Fecha de Nacimiento	Lugar de Nacimiento	Edad	Telefono	direccion	correo	Tipo Usuario	Eliminar	Modificar
37331146	Yaneth Mora Castro	1976-11-28	Ocaña	38	3163321373	Cra 9 N 10-03	gerencia@ipssanamedic.com.co	Gerencia		
1091661166	Monica Amaya Romero	1989-05-05	Teorama (N.S)	25	3168109433	Cra 17A N 3-01 B. Modelo	farmacia@ipssanamedic.com.co	Regente de farmacia		
1091663850	Gerardo Calderon	1990-03-19	San Alberto (cesar)	24	3163269417	calle 8 #16-62(El llano)	gerar1790@gmail.com	Administrador de citas		
1091664435	Sandra Milena Perez Velasquez	1990-07-12	Ocaña (N.S)	24	3106730942	Barrio Nueva España	admin citas@ipssanamedic.com.co	Administrador de citas		

Luego de haber dado click aparecerá la alerta de confirmación

## Confirmación a eliminar el usuario

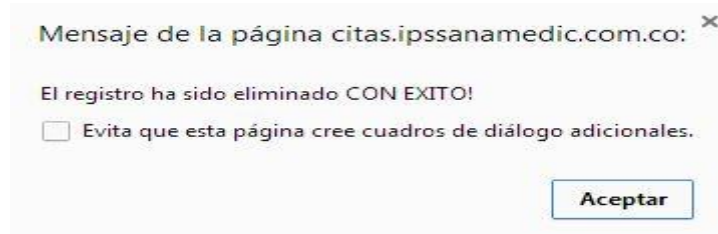


Mensaje de la página citas.ipssanamedic.com.co: ✕

Esta seguro que desea eliminar el registro?

Al hacer click sobre el botón aceptar se procede a la eliminación de la cita y aparece la siguiente alerta.

## Usuario eliminado exitosamente



Mensaje de la página citas.ipssanamedic.com.co: ✕

El registro ha sido eliminado CON EXITO!

Evita que esta página cree cuadros de diálogo adicionales.

Ingresar profesional medico

Para ingresar un nuevo profesional en la base de datos se debe dar clic en el link profesional y escoger de la lista desplegable la opción 'agregar profesional'.

Link agregar profesionales médicos

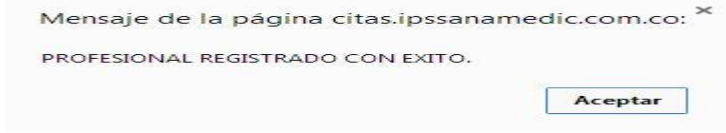


Posteriormente se desplegará la pantalla del formulario con los campos que se deben llenar para un correcto ingreso de datos del profesional médico.

Formulario registro de profesionales médicos

Si se han llenado todos los campos de forma correcta aparecerá la alerta de confirmación.

Alerta usuario registrado exitosamente



**Modificar profesional medico**

Para modificar los datos de un profesional médico se debe dar click en el link ‘listar profesionales’.

**Link listar profesionales médicos**



Fuente: Pasante

En el listado de profesionales se debe dar click en el icono  y se mostrara el formulario con los datos del profesional a modificar.

**Listado de profesionales médicos**

Documento	Nombre	Sexo	Especialidad	Consultorio	Registro Profesional	Fecha Nacimiento	Edad	Telefono	direccion	Correo	Modificar
49773872	Dolbis Romero Ospino	F	Medicina General	2	81725	1974-11-09	40	3176688223	Calle 8B N 29-186 B. altos de la colina	dromero@ipssanamedic.com.co	
51833689	Paulina Perez	F	Odontologia	4	0	1964-11-23	50	3186054707	Calle 9 N 11-115	pperez@ipssanamedic.com.co	
73190548	Fernando Torres	M	Odontologia	4	33	1982-03-06	32	3184879476	-----	ftorres@ipssanamedic.com.co	
88144645	Edgardo Paba Gonzales	M	Medicina General	1	541715	1970-04-10	44	3174014028	Cra 11 N 9B-34	epaba@ipssanamedic.com.co	
91204604	Alvaro Navarro Bonett	M	Medicina General	1	324	1960-04-07	54	3105585911	Calle 8B N 29-186 B. altos de la colina	anavarro@ipssanamedic.com.co	
1065617021	Monica Guevara	F	Medicina	3	54	1990-05-04	24	3167549750	Calle 4 N 27A-95KDX	enfermeria@ipssanamedic.com.co	

**Formulario modificar profesional medico**

MODIFICAR PROFESIONAL

citassanamedic.com.co/profesional/formmodificarprofesional.php?documento=49773872

INFORMACION REQUERIDA PARA ACTUALIZAR PROFESIONALES MEDICOS

POR FAVOR INGRESE TODA LA INFORMACION REQUERIDA

Documento: 49773872      Horario de atención:

Nombre: Dolbis      Fecha de Nacimiento: 09/11/1974

Apellido: Romero Ospino      Telefono o Celular: 317668223

Sexo: F      Direccion: Calle 8B N 29-186 B. altos di

Especialidad del Profesional: 2      Correo Electronico: dromero@ipssanamedic.com

Registro Profesional: 81725      Clave: .....

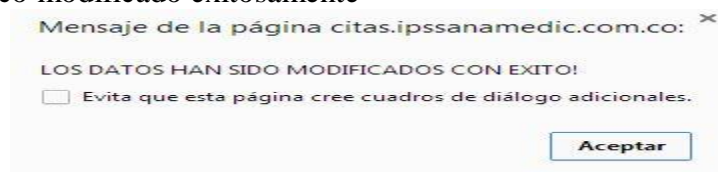
Consultorio: 2      Repetir Clave: .....

Guardar    Cancelar    Volver

SANAMEDIC S.A.S Cra 13 No. 12-48 Centro - Ocaña  
 Horario de atención: Lunes a viernes de 07:00 am a 12:00 m. y de 02:00 pm a 06:00 pm  
 Telefono 562 2020 Celular 317 668 8218  
 E-mail: sanamedic@gmail.com  
 © 2014

Luego de haber modificado los datos se debe dar click en el botón guardar o presionar enter, en caso de deshacer las acciones realizadas se debe dar click en el botón cancelar. Si los datos fueron modificados aparecerá la alerta de confirmación.

### Profesional médico modificado exitosamente



### Generar reportes

La generación de reportes es uno de los procesos más importantes porque es la ayuda para la toma de decisiones en la organización de acuerdo al proceso de citas médicas. Para realizar el proceso se debe dar click en el link informes y escoger de la lista desplegable la opción 'reportes'.

### Link generar reportes



Aparecerá la siguiente imagen donde se debe escoger un Arango de fechas para generar los reportes.

## Formulario generar reportes

**GENERAR REPORTES**

Recuerde que para poder generar las estadísticas de las citas medicas, debe seleccionar un rango de fechas valido para poder ejecutar este proceso.

Por favor seleccione el rango de fechas a generar los indicadores:

dd/mm/aaaa y dd/mm/aaaa

SANAMEDIC S.A. Cra 13 No. 12-68 Centro - Ocaña  
Horario de atención: Lunes a viernes de 07:00 am a 12:00 m. y de 02:00 pm a 06:00 pm

Si en el rango de fechas seleccionado no existen citas asignadas se mostrara la siguiente alerta.

Alerta error generar reportes

Mensaje de la página citas.ipssanamedic.com.co: ✕

**ERROR!! NO EXISTEN CITAS PROGRAMADAS PARA LAS FECHAS SOLICITADAS, POR FAVOR SELECCIONE UN RANGO DE FECHAS VALIDO.**

En caso contrario se mostrara la siguiente pantalla.

Reportes de citas medicas

**ESTADÍSTICAS DE LAS CITAS MEDICAS ATENDIDAS EN LAS FECHAS DE 2015-01-01 A 2015-01-29**

**TABLA DE ESTADÍSTICAS**

	Ecoopos	Porcentaje %	Compañoriente	Porcentaje %	Total Global
Citas Registradas	0	0 %	2	100 %	2
Citas Canceladas	0	0 %	0	0 %	0
Citas Asesibles	0	0 %	1	100 %	1
Citas Asistidas	0	0 %	1	100 %	1
Poblacion Mujeres	0	0 %	0	0 %	0
Poblacion Hombres	0	0 %	2	100 %	2

**CITAS PRIORITARIAS REGISTRADAS**

Nombre Completo	Documento	Sexo	Fecha de asignación	Edad	Caso
No existen citas programadas por casos especiales hasta la fecha					

SANAMEDIC S.A. Cra 13 No. 12-68 Centro - Ocaña  
Horario de atención: Lunes a viernes de 07:00 am a 12:00 m. y de 02:00 pm a 06:00 pm

En la anterior pantalla se muestra información acerca de las citas y las citas prioritarias asignadas en el rango de fechas seleccionado. En la parte de abajo está el botón imprimir en caso de querer ejecutar la acción se debe dar click y proceder a imprimir el documento.



## Generar indicadores de oportunidad

Las entidades prestadoras de salud están obligadas a reportar los indicadores de oportunidad de acuerdo a la resolución 1552 de 2013 del Ministerio de Salud la cual hace referencia a la oportunidad en las citas médicas.

En el sistema de administración de citas médicas es muy sencillo generar estos reportes solo se debe dirigir al link informes y seleccionar 'indicadores de oportunidad'.

## Link generar indicadores de oportunidad



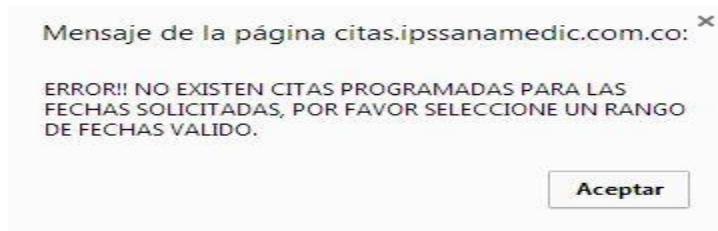
Aparecerá la siguiente pantalla en donde se deben llenar todos los campos y posteriormente dar click en el botón generar.

## Formulario generar indicadores de oportunidad

A screenshot of the 'GENERAR INDICADORES DE OPORTUNIDAD' form in the SANAM MEDIC system. The form includes a title bar, a header with the company logo, and a main content area with instructions: 'Recuerde que para poder generar los indicadores de oportunidad usted debe tener en cuenta la ultima fecha en la cual ejecuto este proceso, y seleccionar el día despues a la ultima fecha generada.' Below the instructions, there is a 'Selecciona la EPS' dropdown menu, a 'Por favor seleccione el rango de fechas a generar los indicadores:' section with two date pickers (dd/mm/aaaa), and 'Generar' and 'Volver' buttons. The footer of the page is identical to the previous screenshot.

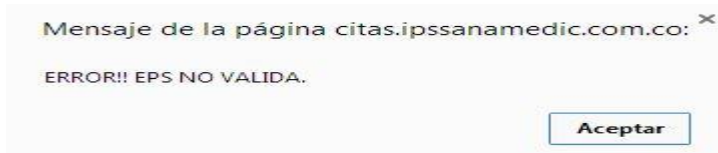
Si el usuario selecciona un rango de fechas donde no se han asignado citas médicas aparecerá la siguiente alerta.

Alerta error al generar indicadores de oportunidad



El usuario debe tener en cuenta que se debe seleccionar una de las EPS que aparecen en el listado de lo contrario aparecerá la siguiente alerta.

#### Alerta error de EPS



Si se introducen los datos correctamente (una EPS válida y un rango de fechas donde se hallan asignado citas médicas) aparecerá la siguiente pantalla.

#### Indicadores de oportunidad

CONSOLIDADO PARA LAS FECHAS DE 2015-01-01 A 2015-01-31 de la empresa Ecoopsos

Odontología				
Total Citas Registradas	Total Citas Canceladas	Total Citas Ausentes	Total Dias Transcurridos	Oportunidad en la asignacion de la cita
65	12	3	63	1

Medicina General				
Total Citas Registradas	Total Citas Canceladas	Total Citas Ausentes	Total Dias Transcurridos	Oportunidad en la asignacion de la cita
211	5	11	283	1.3

Volver Imprimir

### REGENTE DE FARMACIA

El usuario regente de farmacia es la persona encargada de dar los medicamentos a los usuarios pertenecientes de las EPS Comfaoriente y Ecoopsos, por lo cual antes de entregar dichos medicamentos necesita verificar que el usuario esté activo en la base de datos. Cuando este usuario ingresa al módulo encontrará la siguiente pantalla.

Index – regente de farmacia



## Consultar usuarios

Para proceder con la consulta de usuarios debe dar click en el link 'Consultar Usuarios' y se desplegará la siguiente pantalla.

## Listado de usuarios externos activos

Carnet	Documento	Nombre	Apellido	Fecha de Nacimiento	Edad	Sexo	Dirección	Telefono	EPS
1301006532	1007930237	DIANA	JAIMES	04/10/2000	14	F	CRA 34#7-05 LA PRIMAVERA	3134525818	ecoopsos
1301010549	1091661566	TAIRA	OSORIO	10/09/1988	26	F	CRA 25A#3B-02 CAMILO TORRES	3175336584	ecoopsos
1302002882	88265121	ALBEIRO	SUAREZ	08/05/1983	31	M	CRA 23-9-52 SIMON BOLIVAR	3175279846	ecoopsos
1302007149	1091658763	JOAQUIN	NAVARRO	09/07/1988	26	M	CLL 6 # 1-40 EL LLANO	5622778	ecoopsos
1302008126	1004808722	EMERSON	BARRANCO	19/08/2001	45	M	BARRIO LOS CRISTALES	3142562952	ecoopsos
1302112043	1090427221	NICOLL	LOBO	01/06/2008	7	F	BARRIO CRISTO REY	3118298036	ecoopsos
1302113256	13197885	HERMES	MALDONADO	08/12/1972	42	M	VDA SANTA LUCIA LIMONIA	3133834550	ecoopsos
1302126016	60443366	LEDIS	TRILLOS	30/08/1981	45	F	KDX 422-380 VILLA PARAISO	3212327419	ecoopsos
1302136291	1092529520	YENI	VERJEL	12/06/2006	8	F	CLL 19B#12A-02 SANTA LUCIA	3133983154	ecoopsos
1302141342	13371873	LIRO	LAZARO	20/07/1952	45	M	COLINAS DE LA PROVINCIA	3202385807	ecoopsos

En el listado anterior encontrara todos los pacientes activos de las bases de datos de las EPS Comfaorient y Ecoopsos los cuales podrá filtrar por número de carnet, documento, nombre, apellido o EPS según lo desee, solo digitando en la caja de texto buscar como se muestra a continuación.

## Filtro de usuarios externos

Carnet	Documento	Nombre	Apellido	Fecha de Nacimiento	Edad	Sexo	Dirección	Telefono	EPS
1302007149	1091658763	JOAQUIN	NAVARRO	09/07/1988	26	M	CLL 6 # 1-40 EL LLANO	5622778	ecoopsos

## USUARIOS EXTERNOS

Los usuarios externos son aquellos usuarios que están activos en las bases de datos provenientes de las EPS Comfaorienté y Ecoopsos, el módulo de administración de citas médicas está diseñado para que dichos usuarios se asignen, modifiquen y cancelen sus citas médicas de manera autónoma, cabe resaltar que solo se pueden asignar citas de lunes a jueves y con un solo día de diferencia, en horarios establecidos. Para ingresar al módulo se debe dar click en el link ‘Mis citas’ como se muestra a continuación.

### Link ingresar al módulo – usuarios externos



Inmediatamente se mostrara la siguiente pantalla donde se debe digitar el número de documento.

### Login – usuarios externos



Luego de haber digitado el número de documento, pero este no está activo en la base de datos o está mal digitado se mostrara la siguiente alerta.

#### Error de ingreso – usuarios externos



Luego de unos segundos se vuelve a mostrar la pantalla donde debe volver a digitar su número de documento, si este corresponde con los registros de la base de datos se mostrara la siguiente pantalla.

#### Index – usuario externo



## Reservar cita médica

Para proceder con la reservación de una cita médica se debe dar click en el link 'reservar cita' y escoger de la lista si se desea reservar una cita por medicina general u odontología.

## Link reservar citas



Luego de escoger un tipo de especialidad se desplegará la siguiente pantalla donde se podrá hacer la reservación de la cita médica.

## Formulario ingresar cita – usuarios externos

INFORMACION REQUERIDA PARA LA ASIGNACION DE LA CITA - MEDICINA GENERAL

POR FAVOR INGRESE TODA LA INFORMACION REQUERIDA

Documento: 1092181625      Fecha de la Consulta: 2015-01-18

nombre: ZAIRETH      Fecha asignacion de la Cita: dd/mm/aaaa

Apellido: ALVERNIA      Medico: -- Seleccionar --

Telefono:      Hora de la Cita: -----

Sexo: F      EPS: 1

Guardar    Cancelar    Volver

SANAMEDIC S.A.S. Cra. 13 No. 12-66 Centro - Ciénaga  
 Horario de atención: Lunes a viernes de 07:00 am a 12:00 m. y de 02:00 pm a 06:00 pm  
 Telefono 562 2020 Celular 317 668 8218  
 E-mail: sanamedic@ipssanamedic.com.co

El usuario debe llenar los campos de teléfono en caso de no aparecer, la fecha de la asignación, el médico y la hora de la cita médica, el campo hora aparece inmediatamente después que haya elegido un médico, cabe recordar que los horarios establecidos para la reservación de citas médicas por los usuarios externos son solo de 5 citas cada día por profesional, y solo se podrán hacer reservaciones de citas de un (1) día de diferencia lo que quiere decir que solo se puede hacer una reservación de un día para otro de lunes a jueves. Si no se cumple con lo anterior y se solicita una cita para dos o más días después, o en caso contrario para el mismo día aparecerá la siguiente alerta.

#### Alerta error diferencia de días inválido

Mensaje de la página citas.ipssanamedic.com.co: ✕

ERROR!! Estimado usuario su solicitud solo es valida con citas programadas para un dia despues, de Lunes a Jueves.

Aceptar

Si se solicita una cita y el horario no esta disponible aparecera la siguiente alerta.

#### Alerta error horario no disponible

Mensaje de la página citas.ipssanamedic.com.co: ✕

Horario NO disponible

Aceptar

En caso contrario si todos los datos ingresados son correctos y el horario solicitado está disponible se mostrara la siguiente alerta de éxito.

#### Alerta cita registrada exitosamente

Mensaje de la página citas.ipssanamedic.com.co: ✕

CITA REGISTRADA CON EXITO!!

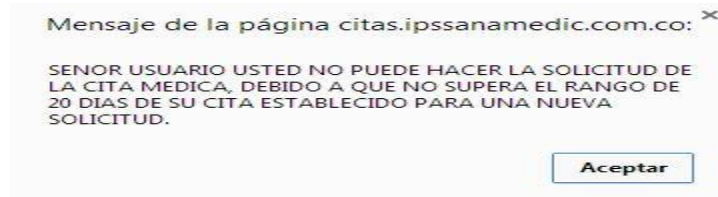
Aceptar

**NOTA:** en caso de que el usuario ya haya hecho una reservación en Medicina General anteriormente y no hayan pasado 20 días, este no podrá hacer una nueva reservación hasta

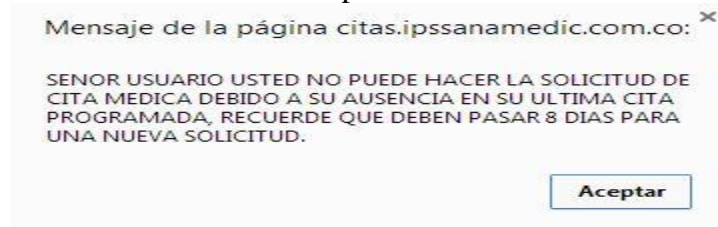
pasado dicho tiempo, si el usuario se ausento a su última cita y no han pasado 8 días tampoco podrá hacer una nueva reservación. En caso de la citas por Odontología el intervalo de tiempo para poder hacer una nueva reservación es de 8 días desde su última cita también aplica para caso de ausencia.

Si el usuario aplica para cualquier caso anteriormente descrito, estas son las alertas que se mostraran en dichos casos.

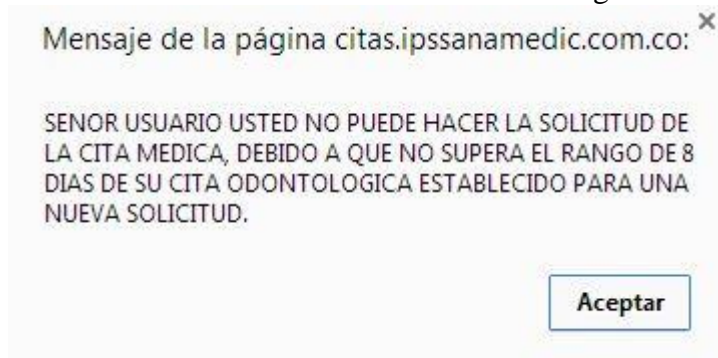
#### Alerta error rango de días desde última cita invalido



#### Alerta error rango de dias desde ultima cita por ausencia invalido



#### Alerta error rango de días desde última cita invalido – odontología



#### Consultar citas

Para proceder a consultar las citas que se han reservado se debe dar click en el link 'Consultar Cita' como se muestra en la siguiente imagen.

. Link consultar citas – usuario externo





Inmediatamente después de haber dado click se desplegara la siguiente pantalla


#### Listado de citas – usuario externo



En la pantalla de la imagen anterior aparecerán todas las citas que el usuario haya solicitado con las opciones de modificarlas y cancelarlas.

**NOTA:** una cita médica solo puede ser modificada o cancelada siempre y cuando esta no haya sido procesada, lo que quiere decir que ya no esté en estado ‘pendiente’.

#### Modificar cita

Para proceder a la modificación de una cita médica se debe dar click en el icono  y se desplegará la siguiente pantalla.

### Modificar cita – usuario externo

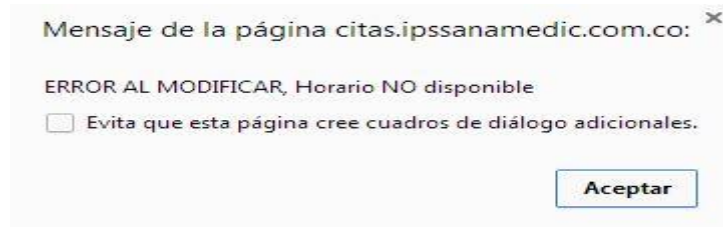


Documento	Nombre	Apellido	Telefono	Fecha Consulta	Fecha Asignacion	Doctor(a)	Hora	Asistencia
1007283607	ANGIE	TORRADO	3134768142	2015-01-16	19 / 01 / 2015	-- Seleccionar --	.....	pendiente

SANAMEDIC S.A.S Cra 13 No. 12-68 Centro - Ocaña  
Horario de atención: Lunes a viernes de 07:00 am a 12:00 m. y de 02:00 pm a 06:00 pm  
Telefono 562 2020 Celular 317 668 8218  
E-mail: sanamedic@ipssanamedic.com.co  
© 2014

**NOTA:** Solo se puede modificar la fecha de asignación, se debe elegir nuevamente el médico y la hora de la cita. Luego de modificar los campos requeridos se debe dar click en el botón ‘Guardar’ si el nuevo horario no está disponible, aparecerá la siguiente alerta.

### Alerta error al modificar – usuario externo



Mensaje de la página citas.ipssanamedic.com.co: ✕

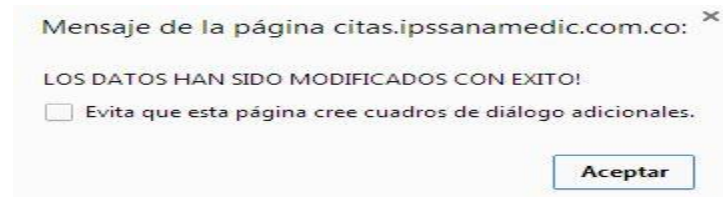
ERROR AL MODIFICAR, Horario NO disponible

Evita que esta página cree cuadros de diálogo adicionales.

Aceptar

De lo contrario si se modifica con éxito, aparecerá la siguiente alerta.

### Cita modificada exitosamente – usuario externo



Mensaje de la página citas.ipssanamedic.com.co: ✕

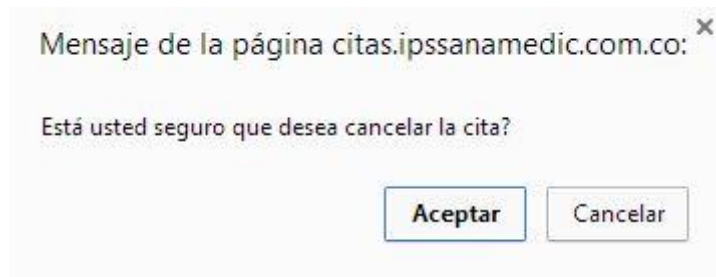
LOS DATOS HAN SIDO MODIFICADOS CON EXITO!

Evita que esta página cree cuadros de diálogo adicionales.

Aceptar

**Cancelar citas.** Para proceder con la cancelación hacer click sobre el icono  y se mostrará la siguiente alerta

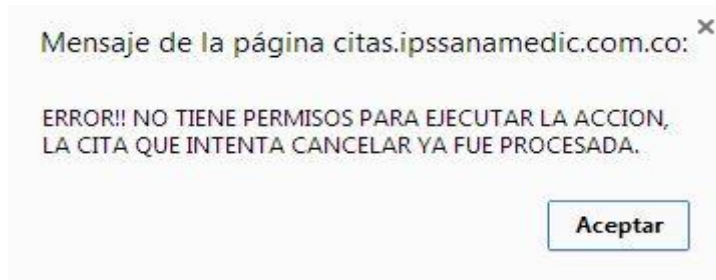
### Alerta verificación al cancelar cita – usuario externo



Si se da click en el botón aceptar se prosigue con la cancelación de lo contrario si se da click en el botón cancelar no se ejecutará ninguna acción.

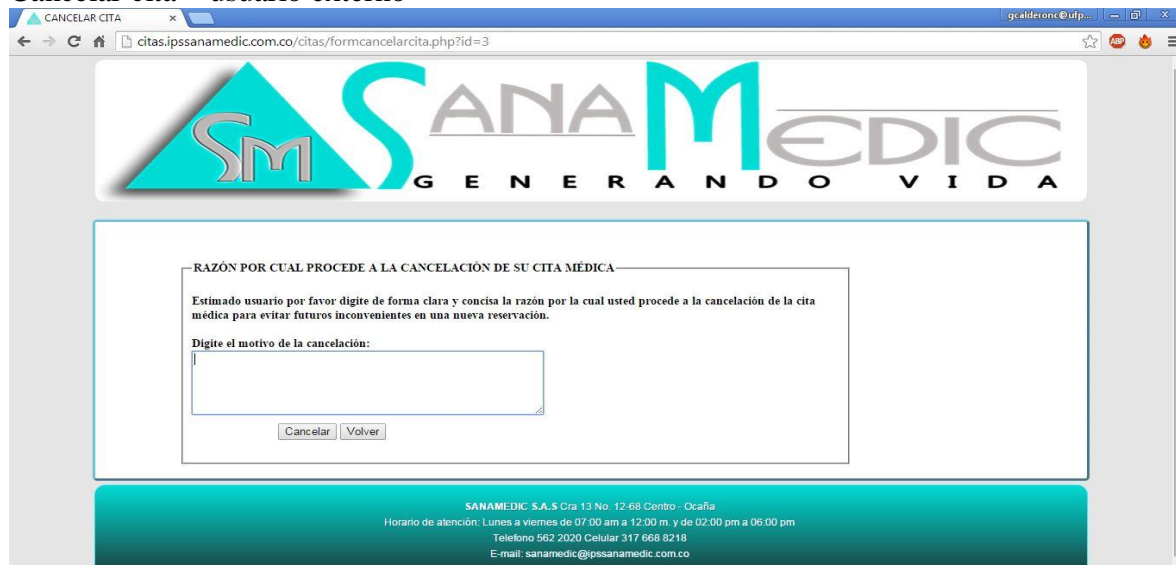
Si la cita no está en estado pendiente, sino que ya fue procesada se mostrara el siguiente pantallazo.

Alerta error al cancelar cita – usuario externo



En caso contrario si la cita está en estado pendiente procederá a la cancelación de la misma, en la siguiente pantalla.

Cancelar cita – usuario externo



Se debe digitar el motivo de la cancelación de la cita y dar click en el botón cancelar o dar enter y debe aparecer la siguiente alerta.

Alerta cita cancelada exitosamente

