	UNIVERSIDAD FRANCISCO DE PAULA SANTANDER OCAÑA			
	<u>Documento</u>	<u>Código</u>	<u>Fecha</u>	<u>Revisión</u>
	FORMATO HOJA DE RESUMEN PARA TRABAJO DE GRADO	F-AC-DBL-007	10-04-2012	A
<u>Dependencia</u>	<u>Aprobado</u>		<u>Pág.</u>	
DIVISIÓN DE BIBLIOTECA	SUBDIRECTOR ACADEMICO		1(119)	

RESUMEN - TESIS DE GRADO

AUTORES	YAN CARLO ANGARITA SANGUINO
FACULTAD	DE INGENIERÍAS
PLAN DE ESTUDIOS	INGENIERÍA DE SISTEMAS
DIRECTOR	CARLOS RENE ANGARITA SANGUINO
TÍTULO DE LA TESIS	DESARROLLO E IMPLEMENTACION DE UN PORTAL WEB PARA LA ADMINISTRACION DE LAS ACTIVIDADES DE FORMACION DE LOS DOCENTES DE LA UNIVERSIDAD SIMON BOLIVAR SECCIONAL CUCUTA

RESUMEN
(70 palabras aproximadamente)

EL SIGUIENTE DOCUMENTO MUESTRA EL DESARROLLO DEL PORTAL WEB PARA LA ADMINISTRACIÓN DE LAS ACTIVIDADES DE FORMACIÓN DE LOS DOCENTES DE LA UNIVERSIDAD SIMÓN BOLÍVAR. SE DESARROLLÓ CON LA METODOLOGÍA XP (PROGRAMACIÓN EXTREMA) Y LA ESTRUCTURA MVC (MODELO-VISTA.CONTROLADOR), DONDE EL MODELO Y LA PERSISTENCIA DE DATOS SE TRABAJÓ CON EL MAPEADOR DE OBJETOS RELACIONALES LLAMADO DOCTRINE, LA VISTA SE DESARROLLÓ CON JQUERY Y BOOTSTRAP EL CUAL ES UN FRAMEWORK PARA DESARROLLAR INTERFACES Y DISEÑOS RESPONSIVE, EL CONTROLADOR QUE ES EL ENCARGADO DE UNIR LA VISTA CON EL MODELO SE DESARROLLÓ CON PHP.

CARACTERÍSTICAS

PÁGINAS: 119	PLANOS:	ILUSTRACIONES: 39	CD-ROM: 1
--------------	---------	-------------------	-----------



**DESARROLLO E IMPLEMENTACION DE UN PORTAL WEB PARA LA
ADMINISTRACION DE LAS ACTIVIDADES DE FORMACION DE LOS
DOCENTES DE LA UNIVERSIDAD SIMON BOLIVAR SECCIONAL CUCUTA**

YAN CARLO ANGARITA SANGUINO

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

**DESARROLLO E IMPLEMENTACION DE UN PORTAL WEB PARA LA
ADMINISTRACION DE LAS ACTIVIDADES DE FORMACION DE LOS
DOCENTES DE LA UNIVERSIDAD SIMON BOLIVAR SECCIONAL CUCUTA**

YAN CARLO ANGARITA SANGUINO

Proyecto de grado presentado para optar el título de Ingeniero de Sistemas

CARLOS RENE ANGARITA SANGUINO
Candidato a Magister
Director del Proyecto

UNIVERSIDAD FRANCISCO DE PAULA SANTANDER OCAÑA
FACULTAD DE INGENIERÍAS
PLAN DE ESTUDIOS DE INGENIERÍA DE SISTEMAS
OCAÑA
2015

CONTENIDO

	Pág.
1. <u>DESARROLLO E IMPLEMENTACION DE UN PORTAL WEB PARA LA ADMINISTRACION DE LAS ACTIVIDADES DE FORMACION DE LOS DOCENTES DE LA UNIVERSIDAD SIMON BOLIVAR SECCIONAL CUCUTA.</u>	11
1.1 <u>PLANTEAMIENTO DEL PROBLEMA</u>	11
1.2 <u>FORMULACION DE LA INVESTIGACION</u>	11
1.3 <u>OBJETIVOS DE LA INVESTIGACION</u>	11
1.3.1 General	11
1.3.2 Específicos	11
1.4 <u>JUSTIFICACION</u>	12
1.5 <u>DELIMITACIONES</u>	13
1.5.1 Geográficas.	13
1.5.2 Temporales.	13
1.5.3 Conceptuales.	13
1.5.4 Operativos.	13
2. <u>MARCO DE REFERENCIA</u>	14
2.1 <u>ANTECEDENTES</u>	14
2.1.1 A Nivel Internacional.	14
2.1.2 A Nivel Nacional.	14
2.1.3 A Nivel Local.	15
2.2 <u>MARCO CONCEPTUAL</u>	15
2.2.1 Arquitectura Modelo Vista Controlador.	15
2.2.2 Base de Datos DB2 de IBM.	16
2.2.3 Lenguaje PHP.	18
2.2.4 Diseño Responsable.	19
2.2.5 Metodologías del Desarrollo de Software.	20
2.2.6 Portal Web.	22
2.2.7 Java Script.	22
2.2.8 Software Libre.	22
2.2.9 HTML 5.	23
2.2.10 CCS3.	23
2.2.11 JQUERY.	24
2.3 <u>MARCO TEORICO</u>	25
2.4 <u>MARCO LEGAL</u>	26
2.4.1 Ley 11723.	26
2.4.2 Proyecto de ley sobre Software Libre.	27
2.5 <u>MARCO CONTEXTUAL</u>	27
3. <u>DISEÑO METODOLOGICO</u>	28
3.1 <u>METODOLOGIA XP</u>	28
3.2 <u>TIPO DE INVESTIGACION</u>	28

3.3	POBLACION	28
3.4	MUESTRA	28
3.5	RECOLECCIÓN DE INFORMACIÓN	29
3.6	ANÁLISIS DE LA INFORMACIÓN	29
4.	ANALISIS Y DISEÑO DEL PORTAL WEB	30
4.1	ROLES Y RESPONSABILIDADES	30
4.2	FASE DE EXPLORACIÓN	30
4.2.1	Actores del Sistema	30
4.2.2	Historias de Usuario	33
4.2.3	Descripción de las historias de usuario.	34
4.3	PLANEACION	39
4.3.1	Priorización y Estimación de las Historias de Usuario.	39
4.3.2	Cronograma de Entregas.	40
4.3.3	Plan de Iteraciones.	41
4.3.4	Planificación de Tareas.	43
4.4	DISEÑO	51
4.4.1	Metáfora	51
4.4.2	Tipo de arquitectura.	51
4.5	DESARROLLO DEL PROTAL WEB	53
4.5.1	Modelo de Datos	53
4.5.2	Diagrama de Clases.	54
4.5.3	Pruebas de aceptación.	64
4.6.	IMPLEMENTAR Y PONER EN MARCHA EL PORTAL WEB.	72
5.	CONCLUSIONES	79
6.	RECOMENDACIONES	80
	BIBLIOGRAFIA	81
	REFERENCIAS DOCUMENTALES ELECTRÓNICAS	82
	ANEXOS	83

LISTAS DE FIGURAS

	Pág.
Figura 1. Funcionamiento del Patrón Modelo-Vista-Controlador. (Bahit, 2012)	16
Figura 2. Actores del Sistema	31
Figura 3. Modelo De Datos	53
Figura 4. Gestionar Información Basic del Docente	55
Figura 5. Gestionar Información Académica del Docente	56
Figura 6. Gestionar Producción del Docente	57
Figura 7. Gestionar Formación Complementaria del Docente	58
Figura 8. Gestionar Actividades Adicionales del Docente	59
Figura 9. Gestionar Reportes del Docente	60
Figura 10. Validación Información del Docente	61
Figura 11. Administrar pertenencia de los Docentes	62
Figura 12. Gestionar Cursos de Formación Académica	63
Figura 13. Autenticaciones del Sistema	64
Figura 14. Prueba de Información Básica del Docente	65
Figura 15. Interfaz de Información Básica del Docente	66
Figura 16. Prueba de Formación Académica del Docente	66
Figura 17. Interfaz Formación Académica del Docente	67
Figura 18. Prueba de Producción de los Docentes	67
Figura 19. Interfaz de Producción de los Docentes	68
Figura 20. Prueba de Formación Continua de los Docentes	68
Figura 21. Interfaz de Formación Continua de los Docentes	69
Figura 22. Actividades Adicionales del Docente	69
Figura 23. Prueba de Reportes	70
Figura 24. Interfaz de Reportes	70
Figura 25. Prueba de Validación de la Información de los Docentes	71
Figura 26. Prueba de Asignación de Programa Principal del Docente	71
Figura 27. Prueba de Autenticación del Sistema	72
Figura 28. Interfaz de Autenticación del Sistema	72
Figura 29. Interfaz Base de Datos DB2	73
Figura 30. Vinculación de Tablas Access	73
Figura 31. Registro de Docentes	74
Figura 32. Registro de Estudios	74
Figura 33. Registro de Formación Continua	75
Figura 34. Usuarios Identificados al Sistema	75
Figura 35. Interfaz Ingreso al Sistema	76
Figura 36. Reporte Control Hoja de Vida	76
Figura 37. Reporte Hoja de Vida	77
Figura 38. Reporte Relación Docente Código	77
Figura 39. Reporte Formación Académica	78

LISTAS DE CUADROS

	Pág.
Cuadro 1. Descripción Actor Docente	32
Cuadro 2. Descripción Actor Recursos Humanos	32
Cuadro 3. Descripción Actor Administración	33
Cuadro 4. Descripción Historia Usuario 1	34
Cuadro 5. Descripción Historia Usuario 2	35
Cuadro 6. Descripción Historia Usuario 3	35
Cuadro 7. Descripción Historia Usuario 4	36
Cuadro 8. Descripción Historia Usuario 5	36
Cuadro 9. Descripción Historia Usuario 6	37
Cuadro 10. Descripción Historia Usuario 7	37
Cuadro 11. Descripción Historia Usuario 8	38
Cuadro 12. Descripción Historia Usuario 9	38
Cuadro 13. Descripción Historia Usuario 10	39
Cuadro 14. Priorización y Estimación de las Historias de Usuario	39
Cuadro 15. Cronograma de Actividades	40
Cuadro 16. Descripción de las Iteraciones	41
Cuadro 17. Primera Iteración	41
Cuadro 18. Segunda Iteración	42
Cuadro 19. Tercera Iteración	42
Cuadro 20. Cuarta Iteración	42
Cuadro 21. Quinta Iteración	43
Cuadro 22. Tareas Abarcadas para M1	43
Cuadro 23. Tareas Abarcadas para M2	44
Cuadro 24. Tareas Abarcadas para M3	45
Cuadro 25. Tareas Abarcadas para M4	46
Cuadro 26. Tareas Abarcadas para M5	46
Cuadro 27. Tareas Abarcadas para M6	47
Cuadro 28. Tareas Abarcadas para M7	48
Cuadro 29. Tareas Abarcadas para M8	49
Cuadro 30. Tareas Abarcadas para M9	49
Cuadro 31. Tareas Abarcadas para M10	50

LISTA DE ANEXOS

	Pág.
Anexo A. Formato de encuesta	85
Anexo B. Resultados de las Encuestas	86
Anexo C. Manual de usuario	89
Anexo E. DOCTRINE 2.0	110
Anexo F. Desarrollo del módulo docente	116

1. DESARROLLO E IMPLEMENTACION DE UN PORTAL WEB PARA LA ADMINISTRACION DE LAS ACTIVIDADES DE FORMACION DE LOS DOCENTES DE LA UNIVERSIDAD SIMON BOLIVAR SECCIONAL CUCUTA.

1.1 PLANTEAMIENTO DEL PROBLEMA

Actualmente en muchas organizaciones se realiza el registro de la formación profesional del talento humano; lo cual es necesario para realizar análisis de currículos, rotación de personal, actualización y acceso rápido de la información y estudio de niveles de formación para redireccionar a nuevos procesos de formación.

La universidad Simón Bolívar cuenta con personal docente altamente calificado, y sus datos no se encuentran almacenados de forma organizada ni estructurada, así que no se puede acceder de forma rápida y confiable. Además es tedioso el proceso de actualización de currículos y el registro de nuevas actividades de formación desarrolladas por los docentes. Es por esto que se ve la necesidad de crear un sistema que respalde todos estos procesos y mantenga toda la información segura y de fácil acceso. Que los docentes puedan entrar al sistema y realizar los procesos que se necesite o realizar las actualizaciones cuando sean necesarias.

También se ve la necesidad de contar con un registro de los cursos de formación que provee la Universidad Simón Bolívar a nivel interno; permitiendo guardar registro de asistencia y el estado final del proceso de formación.

1.2 FORMULACION DE LA INVESTIGACION

¿Un sistema de información para la administración de la formación de los docentes y actualización de hojas de vida en la Universidad Simón Bolívar Seccional Cúcuta permitiría una mejor gestión y actualización del proceso de formación profesoral y el apoyo a la toma de decisiones que orienten los cursos de actualización docente?

1.3 OBJETIVOS DE LA INVESTIGACION

1.3.1 General. Desarrollar e implementar un portal web para la administración de las actividades de formación de los docentes de la universidad Simón Bolívar seccional Cúcuta.

1.3.2 Específicos. Analizar y Diseñar los requerimientos del portal web.

Desarrollar los módulos del portal web.

Implementar y poner en marcha el portal web.

Crear la documentación y soporte del portal web.

1.4 JUSTIFICACION

Con el desarrollo del portal web para la administración de la información de los docentes de la Universidad Simón Bolívar Seccional Cúcuta se conseguirá que la institución de un paso más en avance tecnológico y que se proyecte a los nuevos procesos de mejoramiento en busca de la calidad de los mismos.

Obteniendo una herramienta que permitirá el fácil desarrollo de los procesos de formación de los docentes, la información en tiempo real, bases de datos estructuradas y organizadas; permitiendo el fácil acceso a la información profesional y académica de los docentes y a las nuevas actividades de formación que la Universidad provea.

El portal web permitirá el registro de la información del docente por parte del mismo con la respectiva validación realizada por el Departamento de recursos Humanos; a lo cual se le adicionara las políticas necesarias que permitan asegurar la información registrada.

La herramienta se ajustará a las características institucionales manteniendo la identidad corporativa de la institución.

Una ventaja de aplicar el sistema es la de tener toda la información segura y centralizada y así cuando se necesite realizar algún proceso por parte de la institución o de algún docente. Con el desarrollo e implementación de este sistema se maneja un modelo de actualización de información por parte de los docentes, los cuales conocen sus datos y son los responsables de actualizar la misma, así como lo realizan en Colciencias en aplicaciones como el CVLac. De esta forma se logra la reducción de la carga en el área de recursos humanos, ya que no debe ingresar información, sino realizar la validación de la misma.

De igual forma contar con este sistema permitirá al proceso de pedagogía, la planeación de los cursos institucionales ofertados a los docentes, con base en una estadística real de necesidades que están determinadas por la información almacenada en el sistema, que mientras se mantenga actualizada mantendrá un margen de error mínimo en la asignación de cursos.

La información podrá ser manejada de forma más ágil y en mayor variedad de reportes, para todo lo referente a registro calificado, planta académica, etc., así como podrá ser accedida desde cualquier lugar por los altos directivos en caso de ser necesario.

Poder acceder a la información de forma rápida y en cualquier momento se ha vuelto una necesidad en los sistemas de información, el uso de tecnología Web en este aplicativo logra obtener ese acceso remoto a los datos, y de esta forma poder obtener la información de manera oportuna y significativa, ya que el manejar grandes volúmenes de información en físico se vuelve una opción inmanejable y el replicarla atenta contra el medio ambiente.

1.5 DELIMITACIONES

1.5.1 Geográficas. El proyecto se desarrollara en la Universidad Simón Bolívar Seccional Cúcuta.

1.5.2 Temporales. El tiempo que se tiene establecido para el desarrollo del presente proyecto es de Cuatro (4) meses y sus actividades se encuentran establecidas en el cronograma de actividades.

1.5.3 Conceptuales. Los temas que se desarrollaran en el proyecto se basan en los conceptos de Tecnología Web Libre, Arquitectura Modelo-Vista-Controlador, Lenguaje PHP, Base de Datos DB2 de IBM, HTML, CCS3, ReponsiveDesign, JavaScript, Jquery, Portal Web, Metodologías.

1.5.4 Operativos. Que no se pueda llevar a cabo el desarrollo en el tiempo establecido o no se pueda cumplir con el presupuesto establecido.

2. MARCO DE REFERENCIA

2.1 ANTECEDENTES

2.1.1. A Nivel Internacional. En los estudios realizados de investigación relacionada con la implementación de un sistema de información para llevar el control de una manera más efectiva dentro de una institución educativa. Al respecto;

En el trabajo de investigación titulado “Sistema De Información Para El Control De Acceso Y Registro Del Personal “destacaron que dicho sistema permitirá agilizar el proceso y obtener información confiable y exacta sobre el acceso del personal a las empresas .la fundamentación teórica del proyecto consistió en una amplia documentación referente a la teoría de sistema de información y sistema de control. La investigación se caracterizó por ser de tipo descriptiva y de tipo tecnológica .la metodología tradicional. El análisis de los resultados obtenidos permitió concluir que estas investigaciones representa una herramienta importante para la optimización del proceso de control de acceso y registro del personal.

Lo expuesto por estos autores está muy entrelazado con la investigación realizada debido a que se busca agilizar un registro y control del personal que labora en la unidad Educativa “El Molinete” a través de un sistema de información para así obtener una información confiable y así utilizando las tecnologías de la actualidad.

Por otra parte ;se realizó un proyecto de investigación denominado” Elaborar Un Sistema De Información Donde Registre El Control Del Personal Administrativo De La Unidad Médica Madre María “ teniendo como finalidad un sistema de información donde registre el control del personal administrativo dela Unidad Médica Madre María. Esta investigación según su propósito de tipo proyectivo y descriptivo ya que se orienta a la recolección de datos, se utiliza la técnica como la observación directa y la entrevista no estructurada la metodología seleccionada y aplicada fue la de James Seann.

Otra de las investigaciones que revelan interés sobre esta temática se refiere a la realizada por el Ingeniero Pacheco el cual elaboro un trabajo de investigación titulado “Desarrollo De Un Sistema De Información Para El Control De Acceso Del Personal De URBE” el objetivo de esta investigación consistió en el desarrollo de un sistema de información para el control de acceso del personal tomando en cuenta los requerimientos de los departamentos de nómina. La investigación fue de tipo aplicada.

La citada investigación tiene relación con el presente estudio pues comparte la misma idea de implementar un sistema de información para así llevar el control del personal que labora en dicha casa de estudio. (Molinete, 2012)

2.1.2 A Nivel Nacional. En Colombia se está empezando a implementar el registro del personal en muchas empresas e instituciones para tener un mejor control de la información; de igual forma se ha visto un incremento en usuarios que dejan sus datos, registrando su hoja de vida para que las organizaciones tengan su información disponible y de forma inmediata.

Sitios web como el servicio público de empleo Sena, el empleo.com y otros sitios más ofrecen servicio de búsqueda de empleo con el registro de la hoja de vida del usuario; y lo más importante es que esta información personal se podrá ir actualizando a medida que se requiera y se obtengan más títulos o experiencia de cualquier persona registrada.

Esto es un claro ejemplo de lo importante que es tener el registro de la información en formato digital para tener fácil acceso a ella y poder divulgarla eficazmente.

2.1.3 A Nivel Local. Es casi nula la posibilidad de encontrar a nivel local la utilización de estos sistemas de registro de formación profesoral y hojas de vida, por el contrario si se distinguen los sistemas de registro y control de estudiantes en las instituciones, siendo esto un claro ejemplo de los incontables beneficios que el sistema de registro de formación profesoral traerá al estar implementado.

2.2 MARCO CONCEPTUAL

2.2.1 Arquitectura Modelo Vista Controlador. El patrón MVC es un patrón de arquitectura de software encargado de separar la lógica de negocio de la interfaz del usuario y es el más utilizado en aplicaciones Web, ya que facilita la funcionalidad, mantenibilidad y escalabilidad del sistema, de forma simple y sencilla, a la vez que permite “no mezclar lenguajes de programación en el mismo código”.

Modelo. Aquí se programan todo aquello relacionado con las bases de datos, es decir, las entradas y salidas de datos y se devuelven como se necesiten desde el programa principal. Siendo imaginativos sería como acceder a los datos de algo a través de un API y el API sería el modelo.

Vista. Aquí se programa la parte visual del software, o lo que es lo mismo, la parte que usa el usuario. En el caso de un sitio web la parte de HTML, CSS y JavaScript normalmente.

Controlador. Es quien controla las interacciones del usuario solicitando los datos al modelo y entregándolos a la vista para que ésta, lo presente al usuario, de forma “humanamente legible”.

Cómo Funciona el MVC. El funcionamiento básico del patrón MVC, puede resumirse en:

El usuario realiza una petición.

El controlador captura el evento.

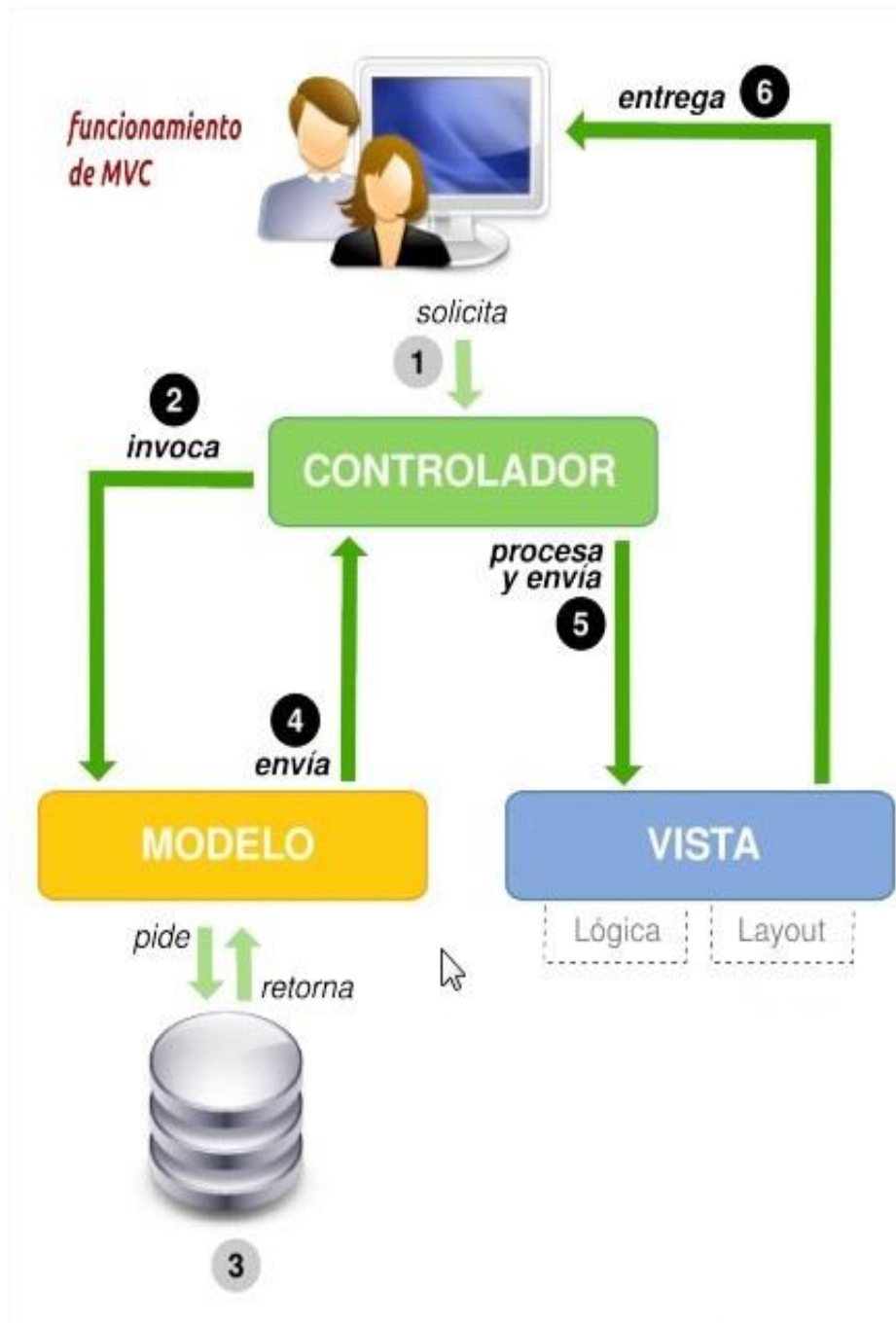
Hace la llamada al modelo/modelos correspondientes efectuando las modificaciones pertinentes sobre el modelo.

El modelo será el encargado de interactuar con la base de datos, ya sea en forma directa, con una capa de abstracción para ello, un Web Service, etc.

El controlador recibe la información y la envía a la vista.

La vista, procesa esta información pudiendo hacerlo desde el enfoque que veremos en este libro, creando una capa de abstracción para la lógica (quien se encargará de procesar los datos) y otra para el diseño de la interfaz gráfica o GUI. (Bahit, 2012)

Figura 1. Funcionamiento del Patrón Modelo-Vista-Controlador. (Bahit, 2012)



Fuente. Autor del proyecto

2.2.2 Base de Datos DB2 de IBM. Una base de datos DB2 en realidad está formada por un conjunto de objetos. Desde la perspectiva del usuario, una base de datos es una colección de tablas que normalmente se relaciona de alguna forma.

Desde la perspectiva de un administrador de base de datos (DBA, ese es usted), es un poco más complicado que eso. La base de datos real contiene muchos de los siguientes objetos físicos y lógicos:

Tablas, vistas, índices y esquemas.

Bloqueos, accionadores, procedimientos almacenados y paquetes.

Agrupaciones de almacenamiento intermedio, archivos de registro y espacios de tabla.

Algunos de estos objetos, como las tablas y las vistas, ayudan a determinar cómo están organizados los datos. Otros objetos, como los espacios de tabla, se refieren a la implementación física de la base de datos. Finalmente, algunos objetos como las agrupaciones de almacenamiento intermedio y otros objetos de memoria, solo se relacionan con cómo se administra el desempeño de la base de datos.

El DBA primero debe concentrarse en la implementación física de la base de datos, más que en considerar todas las combinaciones posibles de parámetros y objetos. ¿Cómo crea usted una base de datos y cómo asigna el espacio en disco requerido para eso? Para responder adecuadamente a esa pregunta usted necesita conocer sobre los objetos básicos de la base de datos y cómo se correlacionan con el almacenamiento físico en disco.

El modelo de almacenamiento DB2. El DB2 tiene tanto un modelo lógico como uno físico para manejar los datos. Los datos efectivos con los que tratan los usuarios se encuentran en tablas. Aunque las tablas pueden estar conformadas por columnas y filas, el usuario no conoce la representación física de los datos. A este hecho se le llama algunas veces la independencia física de los datos.

Por su parte, las tablas se ubican en espacios de tabla. Un espacio de tabla es usado como una capa entre la base de datos y los objetos de contenedor que contienen los datos de la tabla efectiva. Un espacio de tabla puede contener más de una tabla.

Un contenedor es un dispositivo físico de almacenamiento. Este puede identificarse por un nombre de directorio, un nombre de dispositivo, o un nombre de archivo. Un contenedor es asignado a un espacio de tabla. Un espacio de tabla puede abarcar muchos contenedores, lo cual significa que usted puede superar limitaciones de sistema operativo que puedan limitar la cantidad de datos que puede contener un contenedor.

Tablas índices, campos extensos y espacios de tabla. Las tablas, los índices y los espacios extensos (llamados algunas veces objetos binarios grandes o BLOBs) son objetos que se crean dentro de una base de datos DB2. Estos objetos se correlacionan con un espacio de tabla que a su vez se correlaciona con el almacenamiento físico en disco.

Una tabla es un conjunto de registros de datos sin orden. Consiste en columnas y filas que generalmente se conocen como registros. Las tablas pueden ser permanentes (base), temporales (declaradas) temporales (derivadas). Desde la perspectiva del DBA, el espacio se asigna para cada uno de estos objetos de tabla, pero en diferentes espacios de tabla.

Un índice es un objeto físico que está asociado con una tabla individual. Los índices se utilizan para imponer el carácter único dentro de una tabla (esto es, para garantizar que no haya valores duplicados) y para mejorar el desempeño cuando se está recuperando información. Usted no necesita índices para ejecutar sus enunciados SQL (Structured Query Language); Sin embargo, ¡sus usuarios apreciarán su previsión al crear algunos de ellos para agilizar el procesamiento de consultas!

Un campo extenso (o BLOB) es un tipo de datos que se encuentra en una tabla. Este tipo de datos normalmente está compuesto por datos no estructurados (una imagen, un documento, un archivo de audio) y normalmente contiene una cantidad significativa de información. Almacenar este tipo de datos dentro de una tabla podría conducir a costos adicionales excesivos al momento de eliminar, insertar y manipular estos objetos. En lugar de almacenarlos directamente en la fila de la tabla, se almacena un apuntador que se enlaza con un punto de un espacio de tabla Grande (conocido anteriormente como un espacio de tabla de Campo Grande). Los DBA necesitan estar alerta a este tipo de datos, para que puedan crear los espacios de tabla apropiados para que contengan estos objetos.

Otro tipo especial de datos es el XML (eXtensible Markup Language). El XML es un tipo de datos que puede ser almacenado en una fila o en un espacio de tabla aparte similar a los objetos BLOB. El tipo de datos XML es único en cuanto a que puede abarcar múltiples páginas de una tabla, mientras que otros tipos de datos deben permanecer en la misma página que la fila. Un DBA necesitará trabajar con los diseñadores de aplicación para determinar si los objetos XML que se están almacenando en la tabla deben mantenerse en páginas regulares (datos) o puestos en sus propios espacios de tabla separados. Si la recuperación de información es un factor crítico, el DBA debe usar un tamaño de página grande y mantener las columnas XML en el mismo espacio de tabla que los datos regulares. (IBM, 2012)

2.2.3 Lenguaje PHP. Es un lenguaje script (no se compila para conseguir códigos máquina si no que existe un intérprete que lee el código y se encarga de ejecutar las instrucciones que contiene éste código), para el desarrollo de páginas web dinámicas del lado del servidor, cuyos fragmentos de código se intercalan fácilmente en páginas HTML, debido a esto, y a que es de Open Source (código abierto), es el más popular y extendido en la web.

PHP es capaz de realizar determinadas acciones de una forma fácil y eficaz sin tener que generar programas programados en un lenguaje distinto al HTML.

Esto se debe a que PHP ofrece un extenso conjunto de funciones para la explotación de bases de datos sin complicaciones. Es por esto, que levanta un mayor interés con respecto a los lenguajes pensados para los CGI.

PHP fue desarrollado originalmente por Rasmus Ledford en 1994 como un CGI escrito en Perl que permitía la interpretación de un número limitado de comandos. El sistema fue denominado Personal Home Page Tools y consiguió relativo éxito gracias a que otras personas pidieron a Rasmus que les permitiese utilizar sus programas en sus propias páginas. Cuando Rasmus tuvo la necesidad de crear páginas dinámicas que trabajasen con formularios, creó una serie de etiquetas a las que denominó “Form Interpreters”, y lo sacó al público con el nombre de PHP/FI en 1995. Luego salió la versión mejorada, llamada PHP/FI 2.0.

Zeev Suraski y Andi Gutmans programaron el analizador sintáctico incluyendo nuevas funcionalidades como el soporte a nuevos protocolos de Internet y el soporte a la gran mayoría de las bases de datos comerciales, como MySQL y Postgre SQL, así como un módulo para Apache. Con estas mejoras surgió PHP3 en 1997. Este analizador define la sintaxis y semántica de la versión PHP3 y la siguiente: PHP4.

PHP3 carecía del uso de sesiones, algo muy común en las páginas web de cierta complejidad. En el año 2000, PHP3 evolucionó a PHP4, que utiliza el motor Zend (desarrollado por Zeev y Andi encargado de interpretar el código fuente de los scripts de PHP), desarrollado para cubrir las necesidades actuales y solucionar algunos inconvenientes de la anterior versión. Algunas mejoras de esta nueva versión son su mayor independencia del servidor web y su rapidez, ya que primero se compila y luego se ejecuta, mientras que antes se ejecutaba a la vez que se interpretaba el código.

La última versión es PHP5, que utiliza el motor Zend-2 y presenta mejoras significativas y un entorno de programación orientado a objetos mucho más completo, que permite que el PHP proporcione un alto rendimiento a las aplicaciones Web empresariales a nivel de las plataformas J2EE y .NET.

Otro lenguaje de scripting para la generación dinámica de contenidos en el servidor es ASP. Aunque se parece a PHP en cuanto a potencia y dificultad, su sintaxis llega a diferir notablemente.

Una diferencia sensible es que PHP ha sido desarrollado inicialmente para entornos UNIX y es en este sistema operativo donde se aprovechan mejor sus prestaciones y consigue un mayor rendimiento. ASP, que es una tecnología Microsoft, está orientado a sistemas Windows, especialmente NT. (Angel Cobo, Patricia Gomez, Rocio Royal, 2005)

2.2.4 Diseño Responsable. El Diseño Web Responsable, también conocido como Diseño Web Adaptativo o Adaptable, y en inglés como Responsive Web Design consiste en una serie de técnicas de diseño y desarrollo web que permite que las páginas y portales web creados con este formato, se adapten a las pantallas de cualquier dispositivo con el que un usuario los visita.

Tales dispositivos podrían ser ordenadores con diferentes tamaños y formatos de pantalla, tabletas como el ipad, teléfonos smartpone, etc.

Cuando empecé a trabajar en el mundo del diseño web, hacía el año 2000, tenía muy en cuenta que debía mantener un estándar en cuanto a tamaño de páginas creadas para poderse ver correctamente en las resoluciones de pantalla más difundidas en ese momento. Me refiero a los clásicos formatos, de 800×600 y 1024×768 píxeles.

Hoy en día, el diseño web debe ser lo más “flexible” posible, tal cual como esta palabra indica. Al ser 100% flexible, una web podrá verse correctamente en múltiples dispositivos y ordenadores conectados a internet, haciendo la navegación por las secciones mucho más sencilla y usable.

Beneficios del Diseño Web Responsable. Los beneficios que encontramos en diseñar y desarrollar nuestras páginas con estas técnicas son innumerables y cada vez son más importantes, pero mencionaremos sólo algunas de ellas:

La reducción notable en costes de producción y desarrollo: Hasta hoy día se puede encontrar con webs que usan versiones diferentes para ordenadores y dispositivos móviles. Esto se traduce en el desarrollo de 2 o más versiones para cada plataforma, lo que conlleva a invertir más tiempo y dinero en el desarrollo, tanto así para cuando se quiera hacer una modificación dentro del contenido o estructura del sitio (Ya que se deberían hacer 2 o más veces las modificaciones en las diferentes versiones del sitio).

Lo más importante, es que utilizar diseño web responsable, adaptable o adaptativo, conlleva a mejorar enormemente la experiencia del usuario que visita un sitio web, desde el dispositivo con el que cuenta a mano en ese momento. De lo contrario, la navegabilidad le supondría muchas más dificultades de lo normal para encontrar la información que está buscando. (Sarmiento, 2013)

2.2.5 Metodologías del Desarrollo de Software. Se describe como el conjunto de herramientas, técnicas, procedimientos y soporte documental para el diseño de Sistemas de información.

En Ingeniería de software cuando se habla de desarrollo de software se habla de desarrollo de programas y por lo tanto se considera como una tarea de ingeniería, en el cuál se debe ejecutar una serie de fases, etapas para obtener un programa que funcione de acuerdo con métodos ya establecidos en otras disciplinas de ingeniería. Las actividades que los ingenieros de software realizan se encuentran asociadas a un proceso de software donde intervienen diferentes elementos (fases, actividades, producto, roles, agentes) que permiten la definición del software a producir (producto), el desarrollo o el diseño del software, la validación del software tanto lo interno (requerimientos específicos) como lo externo (expectativas del cliente), y la evolución del software donde se modifica para adaptarlo a los cambios.

Además se define que “un método de ingeniería de software es un enfoque estructurado para el desarrollo de software cuyo propósito es facilitar la producción de software de alta calidad de una forma costeable”, cabe destacar que para usar este enfoque se debe manejar conceptos fundamentales tales como; procesos, métodos, tareas, procedimientos, técnicas, herramientas, productos, entre otros.

Adicionalmente una metodología debería definir con precisión los artefactos, roles y actividades, junto con prácticas, técnicas recomendadas y guías de adaptación de la metodología al proyecto. Sin embargo, la complejidad del proceso de creación de software es netamente dependiente de la naturaleza del proyecto mismo, por lo que el escogimiento de la metodología estará acorde al nivel de aporte del proyecto, ya sea pequeño, mediano o de gran nivel. (Martinez, 2005)

A continuación se revisan brevemente cada una de estas categorías de metodologías:

Metodología Estructurada. Los métodos estructurados comenzaron a desarrollarse a fines de los 70's con la Programación Estructurada, luego a mediados de los 70's aparecieron técnicas para el Diseño (por ejemplo: el diagrama de Estructura) primero y posteriormente para el Análisis (por ejemplo: Diagramas de Flujo de Datos). Estas metodologías son particularmente apropiadas en proyectos que utilizan para la implementación lenguajes de 3ra y 4ta generación.

Ejemplos de metodologías estructuradas de ámbito gubernamental: MERISE (Francia), MÉTRICA (España), SSADM (Reino Unido). Ejemplos de propuestas de métodos estructurados en el ámbito académico: Gane & Sarson, Ward & Mellor, Yourdon & DeMarco e Information Engineering.

Metodología Orientada a Objetos. Su historia va unida a la evolución de los lenguajes de programación orientada a objeto, los más representativos: a fines de los 60's SIMULA, a fines de los 70's Smalltalk-80, la primera versión de C++ por Bjarne Stroustrup en 1981 y actualmente Java o C# de Microsoft. A fines de los 80's comenzaron a consolidarse algunos métodos Orientadas a Objeto.

En 1995 Booch y Rumbaugh proponen el Método Unificado con la ambiciosa idea de conseguir una unificación de sus métodos y notaciones, que posteriormente se reorienta a un objetivo más modesto, para dar lugar al Unified Modeling Language (UML), la notación Orientada a Objetos más popular en la actualidad.

Algunas metodologías orientadas a objetos que utilizan la notación UML son:

Rational Unified Process (RUP),
OPEN,
MÉTRICA (que también soporta la notación estructurada).

Metodologías Tradicionales. Las metodologías no ágiles son aquellas que están guiadas por una fuerte planificación durante todo el proceso de desarrollo; llamadas también metodologías tradicionales o clásicas, donde se realiza una intensa etapa de análisis y diseño antes de la construcción del sistema.

Todas las propuestas metodológicas antes indicadas pueden considerarse como metodologías tradicionales. Aunque en el caso particular de RUP, por el especial énfasis que presenta en

cuanto a su adaptación a las condiciones del proyecto (mediante su configuración previa a aplicarse), realizando una configuración adecuada, podría considerarse Ágil.

Metodologías Ágiles. Un proceso es ágil cuando el desarrollo de software es incremental (entregas pequeñas de software, con ciclos rápidos), cooperativo (cliente y desarrolladores trabajan juntos constantemente con una cercana comunicación), sencillo (el método en sí mismo es fácil de aprender y modificar, bien documentado), y adaptable (permite realizar cambios de último momento).

Entre las metodologías ágiles identificadas son:

Extreme Programming

Scrum

Familia de Metodologías Crystal

FeatureDrivenDevelopment

Proceso Unificado Rational, una configuración ágil

Dynamic Systems Development Method

Adaptive Software Development

Open Source Software Development. (Software, 2010)

2.2.6 Portal Web. El portal web es una herramienta en internet que mejora la imagen y la comunicación de una empresa de la noche a la mañana. Además el portal web le ahorra tiempo, dinero y esfuerzo. Usted se preguntará cómo.

En Innovacion hemos desarrollado el Innovacion Portal, un portal web que permite crear y modificar la información a cualquier hora y desde cualquier lugar. Esto quiere decir, que mediante una serie de módulos previamente programados, usted podrá modificar los contenidos sin la necesidad de conocimientos informáticos. Además, diseñamos la estética acorde a la imagen corporativa que deseé proyectar. (Group, 2012)

2.2.7 Java Script. Netscape creó el lenguaje JavaScript en 1996 y lo incluyó en su Netscape Navigator (NN) 2,0 a través de un intérprete que lee y ejecuta el código JavaScript añadido en páginas Html. El lenguaje ha crecido en popularidad de forma constante desde entonces, y ahora está apoyado por los navegadores más populares. (Heilman, 2006)

Javascript es un lenguaje de programación interpretado (un lenguaje de tipo script). A pesar de que existen intérpretes no dependientes de ningún navegador, es un lenguaje de script que suele encontrarse vinculado a páginas web. Javascript y Java son dos lenguajes de programación distintos con filosofías muy diferentes. El único punto en común es la sintaxis, ya que cuando Netscape diseñó Javascript, se inspiró en la sintaxis de Java. (Mateu, 2006)

2.2.8 Software Libre. Es aquel software, producto o desarrollo a medida, que se distribuye bajo una licencia, según la cual el autor cede una serie de libertades básicas al usuario en el marco de un acuerdo de concesión. Se trata de cuatro libertades de los usuarios del software recogidas en la filosofía de la Fundación para el Software Libre (Free Software Foundation),

en particular: la libertad de usar el programa con cualquier propósito; la libertad de estudiar cómo funciona el programa y adaptarlo a sus necesidades; la libertad de distribuir copias; y la libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie. (Martinez, 2007)

2.2.9 HTML 5. Es la quinta revisión importante del lenguaje básico de la World Wide Web, HTML. HTML5 especifica dos variantes de sintaxis para HTML: un «clásico» HTML (text/html), la variante conocida como HTML5 y una variante XHTML conocida como sintaxis XHTML5 que deberá ser servida como XML (XHTML) (application/xhtml+xml). Esta es la primera vez que HTML y XHTML se han desarrollado en paralelo.

Todavía se encuentra en modo experimental, lo cual indica la misma W3C; aunque ya es usado por múltiples desarrolladores web por sus avances, mejoras y ventajas. Al no ser reconocido en viejas versiones de navegadores por sus nuevas etiquetas, se le recomienda al usuario común actualizar a la versión más nueva, para poder disfrutar de todo el potencial que provee HTML5. El desarrollo de este lenguaje de marcado es regulado por el Consorcio W3C.

Nuevos Elementos. HTML5 establece una serie de nuevos elementos y atributos que reflejan el uso típico de los sitios web modernos. Algunos de ellos son técnicamente similares a las etiquetas `<div>` y ``, pero tienen un significado semántico, como por ejemplo `<nav>` (bloque de navegación del **sitios web**) y `<footer>`. Otros elementos proporcionan nuevas funcionalidades a través de una interfaz estandarizada, como los elementos `<audio>` y `<video>`.

Mejoras en el elemento `<canvas>`, capaz de renderizar en los navegadores más importantes (Mozilla, Chrome, Opera, Safari e IE) elementos 3D.

Algunos elementos de HTML 4.01 han quedado obsoletos, incluyendo elementos puramente de presentación, como `` y `<center>`, cuyos efectos son manejados por el CSS. También hay un renovado énfasis en la importancia del scripting DOM para el comportamiento de la web. 2.0

El lenguaje HTML (hypertext markup language) se utiliza para crear documentos que muestren una estructura de hipertexto. Un documento de hipertexto es aquel que contiene información cruzada con otros documentos, lo cual nos permite pasar de un documento al referenciado desde la misma aplicación. (Torrado, 2012)

2.2.10 CCS3. A diferencia de CSS2, que fue una gran especificación que definía varias funcionalidades, CSS3 está dividida en varios documentos separados, llamados "módulos". Cada módulo añade nuevas funcionalidades a las definidas en CSS2, de manera que se preservan las anteriores para mantener la compatibilidad.

Los trabajos en el CSS3, comenzaron a la vez que se publicó la recomendación oficial de CSS2, y los primeros borradores de CSS3 fueron liberados en junio de 1999.

Debido a la modularización del CSS3, diferentes módulos pueden encontrarse en diferentes estadios de su desarrollo, de forma que a fechas de noviembre de 2011, hay alrededor de cincuenta módulos publicados, tres de ellos se convirtieron en recomendaciones oficiales de la W3C en 2011: "Selectores", "Espacios de nombres" y "Color".

Algunos módulos, como "Fondos y colores", "Consultas de medios" o "Diseños multi-columna" están en fase de "candidatos", y considerados como razonablemente estables, a finales de 2011, y sus implementaciones en los diferentes navegadores son señaladas con los prefijos del motor del mismo.

Limitaciones. Los selectores no pueden usarse en orden ascendente según la jerarquía del DOM (hacia padres u otros ancestros) como se hace mediante XPath.

La razón que se ha usado para justificar esta carencia por parte de la W3C, es para proteger el rendimiento del navegador, que de otra manera, podría verse comprometido. XSLT soporta en la actualidad un mayor número de sistemas operativos. Así mismo, también es mejor para trabajar con la mayoría de buscadores de Internet.

Dificultad para el alineamiento vertical; así como el centrado horizontal se hace de manera evidente en CSS2.1, el centrado vertical requiere de diferentes reglas en combinaciones no evidentes, o no estándares.

Ventajas. Control centralizado de la presentación de un sitio web completo con lo que se agiliza de forma considerable la actualización del mismo. Separación del contenido de la presentación, lo que facilita al creador, diseñador, usuario o dispositivo electrónico que muestre la página, la modificación de la visualización del documento sin alterar el contenido del mismo, sólo modificando algunos parámetros del CSS.

Optimización del ancho de banda de la conexión, pues pueden definirse los mismos estilos para muchos elementos con un sólo selector; o porque un mismo archivo CSS puede servir para una multitud de documentos.

Mejora en la accesibilidad del documento, pues con el uso del CSS se evitan antiguas prácticas necesarias para el control del diseño (como las tablas), y que iban en perjuicio de ciertos usos de los documentos, por parte de navegadores orientados a personas con algunas limitaciones sensoriales. (Torrado, 2012)

2.2.11 JQUERY. Es una biblioteca de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones (FLV) y agregar interacción con la técnica AJAX a páginas web. Fue presentada el 14 de enero de 2006 en el BarCamp NYC. JQuery es la biblioteca de JavaScript más utilizada.

JQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privativos. JQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

Las empresas Microsoft y Nokia anunciaron que incluirán la biblioteca en sus plataformas. Microsoft la añadirá en su IDE Visual Studio y la usará junto con los Frameworks ASP.NET AJAX y ASP.NET MVC, mientras que Nokia los integrará con su plataforma Web Run-Time.

Características.

Selección de elementos DOM.

Interactividad y modificaciones del árbol DOM, incluyendo soporte para CSS 1-3 y un plugin básico de XPath.

Eventos.

Manipulación de la hoja de estilos CSS.

Efectos y animaciones.

Animaciones personalizadas.

AJAX.

Soporta extensiones.

Utilidades varias como obtener información del navegador, operar con objetos y vectores, funciones para rutinas comunes, etc.

Compatible con los navegadores Mozilla Firefox 2.0+, Internet Explorer 6+, Safari 3+, Opera 10.6+ y Google Chrome 8+.

2.3 MARCO TEORICO

En la historia del desarrollo de software aparecen metodologías para realizar el proceso las cuales gracias al trabajo de muchas personas se han ido modificando para cada vez más crear metodologías fáciles para trabajar y ayuden en el proceso del desarrollo.

Este es el caso de **RUP** el cual es definido como:

Un proceso de software desarrollado por la empresa Rational Software, actualmente propiedad de IBM. Junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, diseño, implementación y documentación de sistemas orientados a objetos.

El RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización.

También se conoce por este nombre al software desarrollado por Rational, que incluye información entrelazada de diversos artefactos y descripciones de las diversas actividades. Está incluido en el Rational Method Composer (RMC), que permite la personalización de acuerdo con las necesidades. (Jaramillo, 2010)

Con esta definición ya se pueda entender la gran ayuda que otorga RUP para el desarrollo de software y un método el cual es un ciclo de vida en espiral y donde se encuentran fases e iteraciones, que son número de vueltas que se le realiza al espiral.

Las primeras iteraciones se enfocan hacia la comprensión del problema y la tecnología, la delimitación del ámbito del proyecto, la eliminación de los riesgos críticos, y al establecimiento de una Línea Base de la arquitectura. Durante la fase de inicio las iteraciones hacen mayor énfasis en actividades de modelado del negocio y de requisitos.

En la fase de elaboración, las iteraciones se orientan al desarrollo de la línea base de la arquitectura, abarcan más los flujos de trabajo de requisitos, modelo de negocios, análisis, diseño y una parte de implementación orientado a la línea base de la arquitectura.

En la fase de construcción, se lleva a cabo la construcción del producto por medio de una serie de iteraciones.

Para cada iteración se seleccionan algunos Casos de Uso, se refinan su análisis y diseño y se procede a su implementación y pruebas. Se realiza una pequeña cascada para cada ciclo. Se realizan iteraciones hasta que se termine la implementación de la nueva versión del producto. En la fase de transición se pretende garantizar que se tiene un producto preparado para su entrega a la comunidad de usuarios.

2.4 MARCO LEGAL

2.4.1 Ley 11723. Es una ley compuesta por 89 artículos, sancionada en 1933 (y todavía vigente), conocida como "Ley de Propiedad Intelectual" o también como "Ley de Propiedad Científica, Literaria y Artística". Esta ley regula todo lo referente a derecho de propiedad de una obra artística, científica o literaria, derechos de coautor, enajenación o cesión de una obra, licencias, etc. Además, establece sanciones tanto pecuniarias (multa) como privativas de la libertad (prisión) a quienes violen sus normas.

Su última reforma data de Noviembre de 1998, cuando por Ley 25036 se le introdujeron modificaciones referidas al software, para darle fin a las discusiones doctrinarias y jurisprudenciales sobre la cuestión de si el software estaba o no bajo el amparo de esta ley. Ahora establece expresamente en su Art. 1 que "... las obras científicas, literarias y artísticas comprenden los escritos de toda naturaleza y extensión, entre ellos los programas de computación fuente y objeto; las compilaciones de datos o de otros materiales,..." y en su art. 55 bis que "La explotación de la propiedad intelectual sobre los programas de computación incluirá entre otras formas los contratos de licencia para su uso o reproducción".

2.4.2 Proyecto de ley sobre Software Libre. Es un proyecto presentado en Marzo de 2001 por Marcelo Luis Dragan, Diputado Nacional por la provincia de Tierra del Fuego, del Partido Acción por la República. Originalmente lleva el nombre de "Utilización de Software Libre por el Estado Nacional", y establece la obligación de usar prioritariamente Software Libre en todas las dependencias de la Administración Pública Nacional, salvo excepciones.

Entre los aspectos que motivaron el proyecto, se destacan el económico (por el costo de las licencias y por la libertad de copiar que otorga el Software Libre), el moral (es conocido que en todos los ámbitos de la Administración Pública se utiliza Software Ilegal, ya sea por cuestiones de costos, negligencia, etc., lo cual coloca al Estado como uno de los principales infractores a la Ley 11723), el cultural, el educativo, el de seguridad nacional, etc. Actualmente este proyecto se encuentra en estudio en la comisión de Legislación General y Comunicación. (Libre)

2.5 MARCO CONTEXTUAL

El proyecto se desarrollara en la Universidad Simón Bolívar seccional Cúcuta, Norte de Santander, Colombia.

3. DISEÑO METODOLOGICO

3.1 METODOLOGIA XP

La universidad Simón Bolívar seccional Cúcuta es un entorno en el cual no se tiene bien definidos los requerimientos iniciales por que no se conocen del todo los procesos que se llevan. Además una vez iniciado el proceso los requerimientos que desde un principio no eran confiables van cambiando a medida que se va desarrollando determinado proyecto. En consecuencia a todo lo mencionado se tomó como referencia para el proyecto la metodología XP la cual permite un desarrollo ágil, soporta cambios de los requerimientos iniciales y trabaja con prototipos que es algo esencial para el desarrollo dentro de la universidad.

Con la implementación de la metodología XP lo primero que se debe llevar a cabo es la identificación de los roles y responsabilidades, quien será el cliente, los programadores o testers, el consultor y el manager en caso de haber. Una vez definidos esto se pasa a identificar los actores del sistema que en este caso son el administrador del sistema, recursos humanos y el docente.

Se sigue con establecer las historias de usuario que son una manera simple de describir una tarea concisa que aporta valor al usuario o al negocio; estableciéndoles una priorización dentro del proyecto que puede ser alto, medio o bajo; un cronograma y un plan de iteraciones el cual describe en cada iteración que historias de usuario participaran.

3.2 TIPO DE INVESTIGACION

El tipo de investigación utilizado para el desarrollo del proyecto es de carácter descriptivo, ya que este se caracteriza por describir, analizar e interpretar los datos obtenidos de una situación determinada, mediante la aplicación de técnicas para la recolección de información, observaciones objetivas y exactas, definición y formulación de hipótesis.

3.3 POBLACION

La población seleccionada para aplicar las diferentes herramientas de recolección de información está conformada por ciento cincuenta docentes de la Universidad Simón Bolívar Seccional Cúcuta, y por ser tan grande el número de ellos, se tomara una muestra como se ve más adelante.

Fuente de Información: Universidad Simón Bolívar Seccional Cúcuta.

3.4 MUESTRA

Para calcular el tamaño de la muestra de los docentes se aplicó la siguiente fórmula:

$$n = \frac{4 * P * Q * N}{4 * Q * P + (N - 1) * E^2}$$

Dónde:

N es el tamaño de la población.

4 es el estadístico de prueba del 4% de confianza.

E² máximo error permisible (15%)

P probabilidad de éxito (0.5)

Q probabilidad de fracaso (0.5)

$$n = \frac{4 * 0.5 * 0.5 * 150}{4 * 0.5 * 0.5 + (150 - 1) * 0.15^2}$$

n = 34

La encuesta solo se le aplicara a 34 docentes seleccionados aleatoriamente.

3.5 RECOLECCIÓN DE INFORMACIÓN

Inicialmente se obtuvo información necesaria para el desarrollo del proyecto, a través de la revisión documental de libros y bibliográfica encontrados en la Universidad Simón Bolívar Seccional Cúcuta, consultas a diferentes páginas web, tesis e investigaciones de otras universidades.

Además se realizó una encuesta interna en la institución que correspondía a una serie de preguntas claras y precisas referente al problema con múltiples opciones de respuesta por cada pregunta de tal forma que se centralizara la información para realizar más fácil el análisis buscando en verdad lo que se necesita.

3.6 ANÁLISIS DE LA INFORMACIÓN

Se realizó una encuesta que consta de 7 preguntas claras y precisas para centralizar la información obtenida con la cual se obtuvo que tanto conocimiento hay del tema por parte de los docentes y la viabilidad del proyecto.

Luego de realizada las encuestas se procedió a realizar la tabulación respectiva de cada pregunta y con ayuda de gráficos se podrá hacer una mejor interpretación confiable y segura. (Ver Anexo A y Anexo B).

4. ANALISIS Y DISEÑO DEL PORTAL WEB

4.1 ROLES Y RESPONSABILIDADES

Ciente. Es quien establece que es lo que debe realizar el sistema, tomando la decisión de cuando se debe implementar determinada funcionalidad, así como también en el orden a ser implementadas. En este proyecto el rol Cliente está asignado a recursos humanos que entrara al sistema para realizar los procedimientos requeridos.

El Ingeniero Carlos René Angarita es a quién se le presentarán las respectivas iteraciones y quién realizará las pruebas de aceptación para determinar el éxito o fracaso de cada una, además aportará ideas y sugerencias al proyecto. El equipo de trabajo es quien evaluará cada historia, le asignará una prioridad y un costo medido en semanas de desarrollo, decidirá cómo agrupar las historias y la fecha de la siguiente entrega.

Programadores y Tester. El responsable de escribir los testing del sistema y de mantener el código lo más simple posible es el estudiante de Ingeniería de Sistemas de la Universidad Francisco de Paula Santander Ocaña:

Yan Carlo Angarita Sanguino

Consultor. Es una persona externa al equipo que posee el conocimiento técnico necesario para poder ayudar al equipo con determinados problemas. La persona que actuará en este rol es:

Miguel Roperro Silva, Ingeniero de Sistemas Egresado de la Universidad Francisco de Paula Santander.

Manager. Toma las decisiones más importantes del proyecto. Este rol lo desempeña el Ingeniero Carlos René Angarita Sanguino, quien es el encargado de comunicarse con el equipo para determinar cuál es la situación actual del proyecto y distinguir cualquier dificultad o deficiencia en el proceso y poder solucionarlo.

4.2 FASE DE EXPLORACIÓN

4.2.1 Actores del Sistema

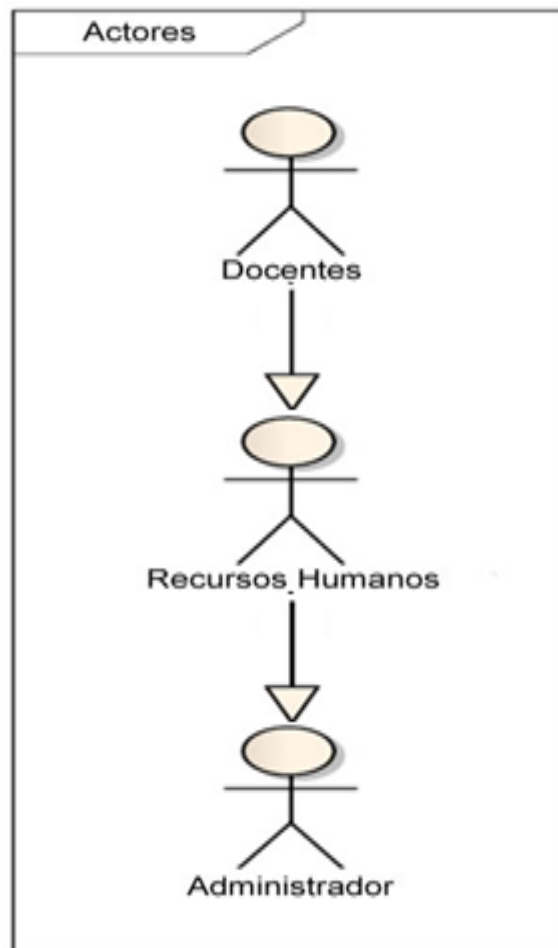
El uso del sistema estará restringido por el rol del usuario. El rol de usuario a su vez dependerá de un inicio de sesión previo a través de credenciales de acceso (nombre de usuario y contraseña). Los usuarios que hayan iniciado sesión dependiendo de su rol pueden ser usuario Administrador, Recursos Humanos o Docente. Los usuarios que acceden al sistema sin un inicio de sesión previo no disponen de los contenidos y la funcionalidad del sistema. Los tipos de usuario se describen a continuación:

Administrador del Sistema. Es el encargado de gestionar y supervisar el sistema. Un usuario con este rol obligatoriamente necesita una autenticación en el sistema por medio de usuario y una contraseña. Puede realizar todas las tareas y acceder a toda la información del sistema.

Recursos Humanos. Es una persona registrada en el sistema y accede con un nombre de usuario y una contraseña. Este usuario podrá acceder a toda la información del sistema referente a los docentes y administrarla. Además tendrá el privilegio de activar o desactivar la información de los docentes.

Docente. Es una persona registrada en el sistema y accede con un nombre de usuario y una contraseña. Este usuario podrá ver su información básica, estudios realizados, experiencias y más; y al mismo tiempo tendrá la opción de generar su hoja de vida entre otras funciones.


Figura 2. Actores del Sistema



Fuente. Autor del proyecto


Descripción de Actores del Sistema

Cuadro 1. Descripción del Actor Docente

Nombre: Docente	
Actor Padre: Ninguno	
Actores Hijos: Recursos Humanos	
<p>Descripción: Actor del sistema que puede y administrar su información básica como los estudios, manejo de idiomas, experiencia laboral, investigaciones y demás.</p> <p>Sus principales interacciones con el sistema son:</p> <ul style="list-style-type: none"> - Administrar información básica que no esté activada por el sistema. - Realizar procesos de cambio de clave o contactar con el administrador del sistema. - Generar Hoja de Vida con los datos que se escojan en el sistema. 	

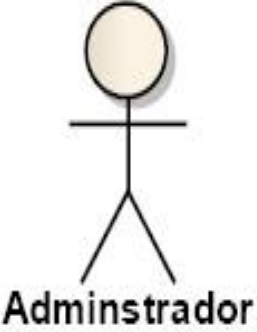
Fuente. Autor del proyecto

Cuadro 2. Descripción del Actor Recursos Humanos

Nombre: Recursos Humanos	
Actor Padre: Usuario Docente	
Actores Hijos: Usuario Administrador	
<p>Descripción: Actor que requiere de un nombre de usuario y una contraseña válidos para acceder al sistema.</p> <p>Sus principales actuaciones en el sistema son:</p> <ul style="list-style-type: none"> - Administrar la información de los docentes. - Validar la información de los docentes. - Administrar los docentes. - Registrar actividades complementarias a los docentes. - Generar Hoja de vida del docente. - Asignar Programa Principal a los docentes. 	

Fuente. Autor del proyecto

Cuadro 3. Descripción del Actor Administrador

Nombre: Usuario Administrador	
Actor Padre: Usuario Docente	
Actores Hijos: Ninguno	
Descripción: Actor encargado de la administración general del sistema. Algunas de sus funciones principales son: <ul style="list-style-type: none">- Administración de usuarios del sistema.- Administrar Perfiles y Privilegios.	

Fuente. Autor del proyecto

4.2.2 Historias de Usuario. Las Historias de Usuario son la técnica utilizada en XP para especificar los requerimientos del software, en ellas se describen brevemente las características que el sistema a implementar debe ofrecer es de la visión del cliente.

A continuación se listarán las Historias de Usuario planteadas por el cliente y que forman los requisitos del proyecto.

H1.Gestionar información básica del docente: Permite registrar, actualizar, consultar y eliminar la información básica de un docente.

H2.Gestionar formación académica de los docentes: Permite registrar, actualizar, consultar y eliminar la formación académica de un docente como lo son los estudios, el manejo de idiomas y demás.

H3.Gestionar la producción de los docentes: Permite registrar, actualizar, consultar y eliminar la producción de un docente como lo son las publicaciones, las investigaciones y demás.

H4. Gestionar la formación complementaria de los docentes: Permite registrar, actualizar, consultar y eliminar la formación complementaria de un docente.

H5.Gestionar actividades adicionales de los docentes: Permite registrar, actualizar, consultar y eliminar las actividades complementarias de un docente.

H6.Gestionar reportes: Permite generar reportes del sistema.

H7. Validar información del docente: Permite activar o desactivar la información de los docentes después de la respectiva validación.

H8.Administrar pertenencia de los docentes al programa: Permite asignar el programa principal al docente.

H9.Gestionar cursos de formación académica: Permite registrar, actualizar, consultar y eliminar los cursos de formación académica de los docentes.

H10. Autenticación del sistema: Un usuario podrá realizar el proceso de autenticación para entrar al sistema.

4.2.3 Descripción de las historias de usuario. Después de terminada la planificación de las entregas se obtiene cada una de las Historias de Usuario con sus respectiva prioridad y estimaciones en tiempo.

Cada una de las Historias de Usuario ha sido evaluada a lo largo del Desarrollo del proyecto, siendo ajustadas a las especificaciones necesarias para el Cumplimiento de los objetivos. El resultado Final de cada Historia de Usuario se muestra a continuación.

Cuadro 4. Descripción Historia de Usuario 1

HISTORIAS DE USUARIO	
NUMERO: 1	USUARIO: Recursos Humanos y Docentes
NOMBRE DE LA HISTORIA: Gestionar Información Básica del Docente	
PRIORIDAD EN NEGOCIO: Alta	RIESGO EN DESARROLLO:
PUNTOS ESTIMADOS: 3	ITERACION ASIGNADA: 1
PROGRAMADOR RESPONSABLE: Yan Carlo Angarita	
DESCRIPCION: Permite realizar la administración de toda la información básica del docente, registrar, modificar, eliminar y consultar todos los datos referentes y tenerlas dispuesta para cualquier situación requerida	
OBSERVACIONES: La información básica la podrá ingresar y editar Recursos Humanos, el usuario docente solo podrá modificar los datos como teléfono, dirección y demás información relevante que no afecte tanto al registro.	

Fuente. Autor del proyecto

Cuadro 5. Descripción Historia de Usuario 2

HISTORIAS DE USUARIO	
NUMERO: 2	USUARIO: Recursos Humanos y Docentes
NOMBRE DE LA HISTORIA: Gestionar Información Académica de los Docentes	
PRIORIDAD EN NEGOCIO: Alta	RIESGO EN DESARROLLO:
PUNTOS ESTIMADOS: 2	ITERACION ASIGNADA: 2
PROGRAMADOR RESPONSABLE: Yan Carlo Angarita	
DESCRIPCION: Permite realizar la administración de toda la información académica del docente, registrar, modificar, eliminar y consultar estudios como especializaciones, maestrías, universidad y demás	
OBSERVACIONES: La información académica será registrada por el docente, pero una vez activada y validada por Recursos Humanos ya el docente no podrá administrarla. Debe estar desactivada para que el docente la pueda gestionar.	

Fuente. Autor del proyecto

Cuadro 6. Descripción Historia de Usuario 2

HISTORIAS DE USUARIO	
NUMERO: 3	USUARIO: Recursos Humanos y Docentes
NOMBRE DE LA HISTORIA: Gestionar la Producción de los Docentes	
PRIORIDAD EN NEGOCIO: Alta	RIESGO EN DESARROLLO:
PUNTOS ESTIMADOS: 2	ITERACION ASIGNADA: 2
PROGRAMADOR RESPONSABLE: Yan Carlo Angarita	
DESCRIPCION: Permite realizar la administración de toda la información de producción del docente, registrar, modificar, eliminar y consultar las publicaciones, distinciones y demás.	
OBSERVACIONES: La información de las producciones será registrada por el docente, pero una vez activada y validada por Recursos Humanos ya el docente no podrá administrarla. Debe estar desactivada para que el docente la pueda gestionar.	

Fuente. Autor del proyecto

Cuadro 7. Descripción Historia de Usuario 4

HISTORIAS DE USUARIO	
NUMERO: 4	USUARIO: Recursos Humanos y Docentes
NOMBRE DE LA HISTORIA: Gestionar Información Complementaria de los Docentes	
PRIORIDAD EN NEGOCIO: Alta	RIESGO EN DESARROLLO:
PUNTOS ESTIMADOS: 2	ITERACION ASIGNADA: 3
PROGRAMADOR RESPONSABLE: Yan Carlo Angarita	
DESCRIPCION: Permite realizar la administración de toda la información complementaria del docente, registrar, modificar, eliminar y consultar diplomados, cursos, semilleros y demás.	
OBSERVACIONES: La información complementaria será registrada por el docente, pero una vez activada y validada por Recursos Humanos ya el docente no podrá administrarla. Debe estar desactivada para que el docente la pueda gestionar.	

Fuente. Autor del proyecto

Cuadro 8. Descripción Historia de Usuario 5

HISTORIAS DE USUARIO	
NUMERO: 5	USUARIO: Recursos Humanos
NOMBRE DE LA HISTORIA: Gestionar Actividades Adicional de los Docentes	
PRIORIDAD EN NEGOCIO: Media	RIESGO EN DESARROLLO:
PUNTOS ESTIMADOS: 1.5	ITERACION ASIGNADA: 3
PROGRAMADOR RESPONSABLE: Yan Carlo Angarita	
DESCRIPCION: Permite administrar las actividades adicionales de los como tutorías, semilleros, coordinación- apoyo y demás actividades internas de la universidad asignándoles una intensidad de horas.	
OBSERVACIONES: Las actividades adicionales que estén registradas, deben ser actividades para que se reflejen en el sistemas y se cuenten sus horas donde se necesite.	

Fuente. Autor del proyecto

Cuadro 9. Descripción Historia de Usuario 6

HISTORIAS DE USUARIO	
NUMERO: 6	USUARIO: Recursos Humanos y Docentes
NOMBRE DE LA HISTORIA: Gestionar Reportes	
PRIORIDAD EN NEGOCIO: Media	RIESGO EN DESARROLLO:
PUNTOS ESTIMADOS: 2	ITERACION ASIGNADA: 5
PROGRAMADOR RESPONSABLE: Yan Carlo Angarita	
DESCRIPCION: Permite realizar la gestión de reportes que se necesiten que el sistema genere. Hojas de Vida de los docentes y reportes de actividades adicionales	
OBSERVACIONES: En todos los reportes, se tendrán opciones de que información se quiere que se muestre en el informe y solo la información activada se verá reflejada.	

Fuente. Autor del proyecto

Cuadro 10. Descripción Historia de Usuario 7

HISTORIAS DE USUARIO	
NUMERO: 7	USUARIO: Recursos Humanos
NOMBRE DE LA HISTORIA: Validar Información del Docente	
PRIORIDAD EN NEGOCIO: Media	RIESGO EN DESARROLLO:
PUNTOS ESTIMADOS: 2	ITERACION ASIGNADA: 4
PROGRAMADOR RESPONSABLE: Yan Carlo Angarita	
DESCRIPCION: Permite realizar el procedimiento de validación de la información por parte de Recursos Humanos.	
OBSERVACIONES: El procedimiento se hace para comprobar que todo lo que los docentes registran al sistema sea verdadero ya que la universidad dará garantía de lo que el sistema genere.	

Fuente. Autor del proyecto

Cuadro 11. Descripción Historia de Usuario 8

HISTORIAS DE USUARIO	
NUMERO: 8	USUARIO: Recursos Humanos
NOMBRE DE LA HISTORIA: Administrar Pertenencia de los Docentes al Programa.	
PRIORIDAD EN NEGOCIO: Media	RIESGO EN DESARROLLO:
PUNTOS ESTIMADOS: 1.5	ITERACION ASIGNADA: 5
PROGRAMADOR RESPONSABLE: Yan Carlo Angarita	
DESCRIPCION: Permite la asignación de los docentes a un programa principal.	
OBSERVACIONES: Los docentes solo pueden pertenecer a un solo programa como principal.	

Fuente. Autor del proyecto

Cuadro 12. Descripción Historia de Usuario 9

HISTORIAS DE USUARIO	
NUMERO: 9	USUARIO: Recursos Humanos
NOMBRE DE LA HISTORIA: Gestionar Curso de Formación Académica	
PRIORIDAD EN NEGOCIO: Alta	RIESGO EN DESARROLLO:
PUNTOS ESTIMADOS: 3	ITERACION ASIGNADA: 4
PROGRAMADOR RESPONSABLE: Yan Carlo Angarita	
DESCRIPCION: Permite realizar la administración de los cursos internos que realiza la universidad dirigidos a los docentes.	
OBSERVACIONES: Los cursos irán ligados a grupos de los cuales al final se revisara asistencia y calificaciones, dando así aprobado o reprobado.	

Fuente. Autor del proyecto

Cuadro 13. Descripción Historia de Usuario 10

HISTORIAS DE USUARIO	
NUMERO: 10	USUARIO: Recursos Humanos y Docentes
NOMBRE DE LA HISTORIA: Autenticación del Sistema	
PRIORIDAD EN NEGOCIO: Alta	RIESGO EN DESARROLLO:
PUNTOS ESTIMADOS: 1	ITERACION ASIGNADA: 5
PROGRAMADOR RESPONSABLE: Yan Carlo Angarita	
DESCRIPCION: Validará la entrada al sistema, pidiendo un usuario y una contraseña para iniciar su respectiva sesión, así como la verificación de que no se haya iniciado sesión con anterioridad.	
OBSERVACIONES: Los cursos irán ligados a grupos de los cuales al final se revisara asistencia y calificaciones, dando así aprobado o reprobado.	

Fuente. Autor del proyecto

4.3 PLANEACION

4.3.1 Priorización y Estimación de las Historias de Usuario. El usuario establece la prioridad de acuerdo a su importancia para el cumplimiento de los objetivos del proyecto descritas de la siguiente manera:

Alta: La historia de usuario debe implementarse para cumplir con los objetivos del proyecto.

Media: La historia de usuario tiene un impacto medible sobre los objetivos del proyecto.

Baja: La historia de usuario aumentará la satisfacción del usuario, pero no se ha justificado explícitamente.

El equipo desarrollador está a cargo de la estimación del esfuerzo por cada historia de usuario y se determina por semanas. La priorización y estimación de cada historia de usuario está definida en el siguiente cuadro:

Cuadro 14. Priorización y Estimación de las Historias de Usuario.

Historia	Nombre de la Historia de Usuario	Prioridad	Estimación
H1	Gestionar información básica del docente	Alta	3
H2	Gestionar formación académica de los docentes	Alta	2

Cuadro 14. (Continuación)

H3	Gestionar la producción de los docentes	Alta	2
H4	Gestionar la formación complementaria de los docentes	Alta	2
H5	Gestionar actividades adicionales de los docentes	Media	1.5
H6	Gestionar reportes	Media	2
H7	Validar información del docente	Media	2
H8	Administrar pertenencia de los docentes al programa	Media	1.5
H9	Gestionar cursos de formación académica	Alta	3
H10	Autenticación del sistema	Alta	1

Fuente. Autor del proyecto

4.3.2 Cronograma de Entregas. Partiendo de la prioridad y estimación de las Historias de Usuario se establecen las fechas de programación de las entregas de pequeñas versiones de la aplicación.

Nota: Al hablar de versiones se entiende por adición en la funcionalidad de la aplicación, utilidades de la aplicación que no han sido revisadas y probadas por el cliente.

Cuadro 15. Cronograma de actividades.

HISTORIAS	ITERACION 1				ITERACION 2				ITERACION 3				ITERACION 4				ITERACION 5			
	SEMANA 1	SEMANA 2	SEMANA 3	SEMANA 4	SEMANA 1	SEMANA 2	SEMANA 3	SEMANA 4	SEMANA 1	SEMANA 2	SEMANA 3	SEMANA 4	SEMANA 1	SEMANA 2	SEMANA 3	SEMANA 4	SEMANA 1	SEMANA 2	SEMANA 3	SEMANA 4
H1	■	■	■																	
H2				■	■															
H3						■	■													
H4							■	■												
H5									■	■										
H6										■	■	■								
H7											■	■								
H8													■	■						
H9																■	■	■		
H10																				■

Fuente. Autor del proyecto

4.3.3 Plan de Iteraciones. Las iteraciones definidas para el proyecto están basadas en las prioridades y estimaciones descritas anteriormente, cada iteración tendrá una duración máxima de tres semanas. Cada Iteración implementada dará como resultado un prototipo funcional con las respectivas pruebas de unidad y aceptación. A continuación se definen las iteraciones que presentará el proyecto:

Cuadro 16. Descripción de las iteraciones.

Iteración	Historias de Usuario
Iteración 1	H1
Iteración 2	H2, H3
Iteración 3	H4, H5
Iteración 4	H7, H9
Iteración 5	H6, H8, H10

Fuente. Autor del proyecto

A continuación se presenta las historias de usuario abarcadas por cada Iteración y su estimación en puntos.

Primera Iteración. En la primera Iteración se comprobará la adecuación de las herramientas y tecnologías a utilizar, el estudio de las librerías utilizadas para el desarrollo del proyecto y se desarrollará una de las historias de usuario más importantes y generales del proyecto, generando así un prototipo para comprobarla adecuación de las tecnologías utilizadas y se intentará crear la mayor parte de la Arquitectura del Sistema.

Cuadro 17. Primera Iteración.

HISTORIA	NOMBRE	PRIORIDAD	ESTIMACION
H1	Gestionar Información Básica del Docente	Alta	3

Fuente. Autor del proyecto

Iteración. En la segunda iteración se completará la funcionalidad de la información del docente agregándole a los datos básicos, la gestión de la información de producción y estudios de los docentes. Se completará más el prototipo y se seguirá alimentando la arquitectura del sistema.

Cuadro 18. Segunda Iteración.

HISTORIA	NOMBRE	PRIORIDAD	ESTIMACION
H2	Gestionar Información Académica de los Docentes	Alta	2
H3	Gestionar la Producción de los Docentes	Alta	2

Fuente. Autor del proyecto

Tercera Iteración. En la tercera iteración se completara el módulo de los datos de formación profesoral del docente dando por terminado el prototipo referente a este desarrollo. Además se iniciara el segundo proceso de gestión de actividades adicionales. Además se seguirá alimentando la arquitectura del sistema.

Cuadro 19. Tercera Iteración.

HISTORIA	NOMBRE	PRIORIDAD	ESTIMACION
H4	Gestionar Formación Complementaria de los Docentes	Alta	2
H5	Gestionar Actividades Adicionales de los Docentes	Media	1.5

Fuente. Autor del proyecto

Cuarta Iteración. En la cuarta iteración se desarrolla el modulo que permitirá validar la información de los docentes activándola o desactivándola; de igual forma el que permite gestionar los cursos de formación académica dando por terminado la mayor parte del desarrollo.

Cuadro 20. Cuarta Iteración.

HISTORIA	NOMBRE	PRIORIDAD	ESTIMACION
H7	Validar Información del Docente	Media	2
H9	Gestionar Cursos de Formación Académica	Alta	3

Fuente. Autor del proyecto

Quinta Iteración. Al llegar a la quinta iteración ya el sistema estará casi completo, y lo que se hace aquí es desarrollar el módulo de asignación de los docentes a los programas principales y autenticación del sistema.

Se trabajara la generación de reportes y todo lo referente a este proceso.

Cuadro 21. Quinta Iteración.

HISTORIA	NOMBRE	PRIORIDAD	ESTIMACION
H6	Gestionar Reportes	Media	2
H8	Administrar Pertenencia de los Docentes al Programa	Media	1.5
H10	Autenticación del Sistema	Alta	1

Fuente. Autor del proyecto

4.3.4 Planificación de Tareas. Para cada Iteración se tomarán las Historias de Usuario que serán implementadas y se dividen en tareas para llevarlas a cabo de manera que sean de fácil entendimiento y rápido desarrollo, se planificarán asignándoles una estimación de tiempo para su desarrollo.

Cuadro 22. Tareas Abarcadas para H1.

TAREA	
Numero de Tarea: 1	Numero de Historia: H1
Nombre de Tarea: Investigación sobre el uso de las librerías y tecnologías a usar.	
Tipo de Tarea: Documentación	Puntos Estimados: 0,6
Programador Responsable: Yan Carlo Angarita	
Descripción: Se investiga sobre la aplicación de doctrine para el proceso de la persistencia. La utilización de PHP y la librería Bootstrap para el desarrollo de las interfaces. El concepto de MVC (modelo-vista-controlador) y el manejo de base de datos con la tecnología DB2 de IBM.	

TAREA	
Numero de Tarea: 2	Numero de Historia: H1
Nombre de Tarea: Instalación del Software y Hardware	
Tipo de Tarea: Adecuación	Puntos Estimados: 0,2
Programador Responsable: Yan Carlo Angarita	
Descripción: Se debe descargar, instalar y configurar las herramientas necesarias para el desarrollo y funcionamiento de la aplicación, como son librerías, servidor, entornos de desarrollo. Además se debe adecuar el hardware necesario para el funcionamiento y desarrollo de la aplicación. Además se configura el manejador de base de datos y se crea la base de datos con las tablas básicas del sistema.	

Cuadro 22. (Continuación)

TAREA	
Numero de Tarea: 3	Numero de Historia: H1
Nombre de Tarea: Desarrollo Modulo de Gestión de la Información Básica.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1,6
Programador Responsable: Yan Carlo Angarita	
Descripción: Se desarrolla el módulo de gestión de información básica de los docentes, creando las tablas requeridas, desarrollando el código PHP con doctrine para la persistencia y creando las interfaces necesarias para la administración de la información.	

TAREA	
Numero de Tarea: 4	Numero de Historia: H1
Nombre de Tarea: Pruebas de Funcionamiento.	
Tipo de Tarea: Verificación	Puntos Estimados: 0,6
Programador Responsable: Yan Carlo Angarita	
Descripción: Se verifica el funcionamiento del módulo con las pruebas funcionales y el buen acoplamiento de la vista, controlador y el modelo. Se realizan pruebas de administración de la información, navegabilidad y seguridad.	

Fuente. Autor del proyecto

Cuadro 23. Tareas Abarcadas para H2.

TAREA	
Numero de Tarea: 1	Numero de Historia: H2
Nombre de Tarea: Desarrollo de la Persistencia y Proceso de Negocio.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,6
Programador Responsable: Yan Carlo Angarita	
Descripción: Se crean las tablas, las entidades, los modales y desarrolla la persistencia para lograr las funcionalidades del negocio.	

TAREA	
Numero de Tarea: 2	Numero de Historia: H2
Nombre de Tarea: Diseño de Interfaces.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1,2
Programador Responsable: Yan Carlo Angarita	
Descripción: Se desarrollan las interfaces en base a los requerimientos del sistema, la navegación, el menú y la administración de los datos.	

Cuadro 23. (Continuación)

TAREA	
Numero de Tarea: 3	Numero de Historia: H2
Nombre de Tarea: Pruebas de Funcionamiento.	
Tipo de Tarea: Verificación	Puntos Estimados: 0,2
Programador Responsable: Yan Carlo Angarita	
Descripción: Se verifica el funcionamiento del módulo con las pruebas funcionales y el buen acoplamiento de la vista, controlador y el modelo. Se realizan pruebas de administración de la información, navegabilidad y seguridad.	

Fuente. Autor del proyecto

Cuadro 24. Tareas Abarcadas para H3.

TAREA	
Numero de Tarea: 1	Numero de Historia: H3
Nombre de Tarea: Desarrollo de la Persistencia y Proceso de Negocio.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,6
Programador Responsable: Yan Carlo Angarita	
Descripción: Se crean las tablas, las entidades, los modales y desarrolla la persistencia para lograr las funcionalidades del negocio.	

TAREA	
Numero de Tarea: 2	Numero de Historia: H3
Nombre de Tarea: Diseño de Interfaces.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1,2
Programador Responsable: Yan Carlo Angarita	
Descripción: Se desarrollan las interfaces en base a los requerimientos del sistema, la navegación, el menú y la administración de los datos como las publicaciones, distinciones e investigaciones. Se validan los componentes de las vistas y que sean visibles según el usuario.	

TAREA	
Numero de Tarea: 3	Numero de Historia: H3
Nombre de Tarea: Pruebas de Funcionamiento.	
Tipo de Tarea: Verificación	Puntos Estimados: 0,2
Programador Responsable: Yan Carlo Angarita	
Descripción: Se verifica el funcionamiento del módulo con las pruebas funcionales y el buen acoplamiento de la vista, controlador y el modelo. Se realizan pruebas de administración de la información, navegabilidad y seguridad. Se valida que los usuarios que tengan los privilegios sean los únicos que puedan entrar a las interfaces.	

Fuente. Autor del proyecto

Cuadro 25. Tareas Abarcadas para H4.

TAREA	
Numero de Tarea: 1	Numero de Historia: H4
Nombre de Tarea: Desarrollo de la Persistencia y Proceso de Negocio.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,6
Programador Responsable: Yan Carlo Angarita	
Descripción: Se crean las tablas, las entidades, los modales y desarrolla la persistencia para lograr las funcionalidades del negocio. Se crean las tablas necesarias para administrar la información complementaria de los docentes, lugar, titulo, institución y demás.	

TAREA	
Numero de Tarea: 2	Numero de Historia: H4
Nombre de Tarea: Diseño de Interfaces.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1,2
Programador Responsable: Yan Carlo Angarita	
Descripción: Se desarrollan las interfaces en base a los requerimientos del sistema, la navegación, el menú y la administración de los datos como los cursos, diplomados y semilleros. Se validan los componentes de las vistas y que sean visibles según el usuario.	

TAREA	
Numero de Tarea: 3	Numero de Historia: H2
Nombre de Tarea: Pruebas de Funcionamiento.	
Tipo de Tarea: Verificación	Puntos Estimados: 0,2
Programador Responsable: Yan Carlo Angarita	
Descripción: Se verifica el funcionamiento del módulo con las pruebas funcionales y el buen acoplamiento de la vista, controlador y el modelo. Se realizan pruebas de administración de la información, navegabilidad y seguridad. Se valida que los usuarios que tengan los privilegios sean los únicos que puedan entrar a las interfaces.	

Fuente. Autor del proyecto

Cuadro 26. Tareas Abarcadas para H5.

TAREA	
Numero de Tarea: 1	Numero de Historia: H5
Nombre de Tarea: Desarrollo de la Persistencia y Proceso de Negocio.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,4
Programador Responsable: Yan Carlo Angarita	
Descripción: Se crean las tablas, las entidades, los modales y desarrolla la persistencia para lograr las funcionalidades del negocio. Se crean las tablas necesarias para administrar las actividades adicionales.	

Cuadro 26. (Continuación)

TAREA	
Numero de Tarea: 2	Numero de Historia: H5
Nombre de Tarea: Diseño de Interfaces.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,8
Programador Responsable: Yan Carlo Angarita	
Descripción: Se desarrollan las interfaces en base a los requerimientos del sistema, la navegación, el menú y la administración de los datos. Se crea la interfaz de búsqueda de los docentes y de ahí la interfaz de listado de actividades. Se validan los componentes de las vistas y que sean visibles según el usuario.	

TAREA	
Numero de Tarea: 3	Numero de Historia: H5
Nombre de Tarea: Pruebas de Funcionamiento.	
Tipo de Tarea: Verificación	Puntos Estimados: 0,3
Programador Responsable: Yan Carlo Angarita	
Descripción: Se verifica el funcionamiento del módulo con las pruebas funcionales y el buen acoplamiento de la vista, controlador y el modelo. Se realizan pruebas de administración de la información, navegabilidad y seguridad. Se valida que los usuarios que tengan los privilegios sean los únicos que puedan entrar a las interfaces. Se revisa que las búsquedas se realicen correctamente y los datos se ingresen de la misma forma.	

Fuente. Autor del proyecto

Cuadro 27. Tareas Abarcadas para H6.

TAREA	
Numero de Tarea: 1	Numero de Historia: H6
Nombre de Tarea: Desarrollo de la Estructura de Carpetas.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,2
Programador Responsable: Yan Carlo Angarita	
Descripción: Se crean todas las carpetas donde estarán depositados los reportes, y además donde serán guardados los nuevos reportes que se generan.	

TAREA	
Numero de Tarea: 2	Numero de Historia: H6
Nombre de Tarea: Diseño de Interfaces y Reportes.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1,5
Programador Responsable: Yan Carlo Angarita	
Descripción: Se diseñan los reportes a utilizar con el IReport manteniendo el formato institucional. Se crean las interfaces donde se podrán descargar los reportes requeridos por el usuario. De igual forma se valida que los reportes solo sean visibles a usuarios que tengan los privilegios.	

Cuadro 27. (Continuación)

TAREA	
Numero de Tarea: 3	Numero de Historia: H6
Nombre de Tarea: Pruebas de Funcionamiento.	
Tipo de Tarea: Verificación	Puntos Estimados: 0,3
Programador Responsable: Yan Carlo Angarita	
Descripción: Se verifica que los reportes generados por el sistema contengan la información requerida y no hallan fallos en ella. Además que los formatos sean compatibles y no hallan problemas de lecturas en los informes.	

Fuente. Autor del proyecto

Cuadro 28. Tareas Abarcadas para H7.

TAREA	
Numero de Tarea: 1	Numero de Historia: H7
Nombre de Tarea: Verificación del Estado de la Información.	
Tipo de Tarea: Adecuación	Puntos Estimados: 0,5
Programador Responsable: Yan Carlo Angarita	
Descripción: Verificar que toda la información que se registra en el sistema tenga la opción en el campo Estado de desactivado para que el sistema sea después el que verifique los registros y su validez.	

TAREA	
Numero de Tarea: 2	Numero de Historia: H7
Nombre de Tarea: Diseño de Interfaces.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Programador Responsable: Yan Carlo Angarita	
Descripción: Se crean las interfaces para cada módulo de activar y desactivar la información, teniendo el control de que solo la información desactivada se pueda modificar. Se crean los privilegios para cambiar el estado de la información.	

TAREA	
Numero de Tarea: 3	Numero de Historia: H7
Nombre de Tarea: Pruebas de Funcionamiento.	
Tipo de Tarea: Verificación	Puntos Estimados: 0,5
Programador Responsable: Yan Carlo Angarita	
Descripción: Se verifica el funcionamiento del módulo con las pruebas funcionales y el buen acoplamiento de la vista, controlador y el modelo. Se realizan pruebas de administración de la información, navegabilidad y seguridad. Se valida que los usuarios que tengan los privilegios sean los únicos que puedan entrar a las interfaces. Se valida que la información validada solo la pueda modificar Recurso Humanos.	

Fuente. Autor del proyecto

Cuadro 29. Tareas Abarcadas para H8.

TAREA	
Numero de Tarea:1	Numero de Historia:H8
Nombre de Tarea: Desarrollo de la Persistencia y Proceso de Negocio.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,3
Programador Responsable: Yan Carlo Angarita	
Descripción: Se crean las estructuras de datos, nuevas tablas para soportar la asignación de los docentes a los programas, se crean los modales y las entidades para la persistencia. Se desarrollan los controladores para poder conectar la persistencia con la vista.	
TAREA	
Numero de Tarea:2	Numero de Historia:H8
Nombre de Tarea: Diseño de Interfaces.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,8
Programador Responsable: Yan Carlo Angarita	
Descripción: Se crean las interfaces de búsqueda de docentes y de asignación de los docentes a los programas principales. Se hace los respectivos privilegios para garantizar que solo los usuarios autorizados sean los que administren la asignación.	

TAREA	
Numero de Tarea:3	Numero de Historia:H8
Nombre de Tarea: Pruebas de Funcionamiento.	
Tipo de Tarea: Verificación	Puntos Estimados: 0,4
Programador Responsable: Yan Carlo Angarita	
Descripción: Se verifica el funcionamiento del módulo con las pruebas funcionales y el buen acoplamiento de la vista, controlador y el modelo. Se realizan pruebas de administración de la información, navegabilidad y seguridad. Se valida que los usuarios que tengan los privilegios sean los únicos que puedan entrar a las interfaces.	

Cuadro 30. Tareas Abarcadas para H9.

TAREA	
Numero de Tarea:1	Numero de Historia:H9
Nombre de Tarea: Desarrollo de la Persistencia y Proceso de Negocio.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Programador Responsable: Yan Carlo Angarita	
Descripción: Se crean las estructuras de datos, nuevas tablas para soportar la administración, se crean los modales y las entidades para la persistencia. Se desarrollan los controladores para poder conectar la persistencia con la vista. Se desarrolla la persistencia para la administración de los grupos, de los cursos, de los tutores y demás.	

Cuadro 30. (Continuación)

TAREA	
Numero de Tarea: 2	Numero de Historia: H9
Nombre de Tarea: Diseño de Interfaces.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1,5
Programador Responsable: Yan Carlo Angarita	
Descripción: Se crean las interfaces para la administración de los cursos de formación académica, la forma de relacionar los grupos con los cursos. Se habilitan los privilegios y se validan los controladores por privilegios garantizando que los docentes o tutores no puedan modificar los datos; solo lo puedan hacer los usuarios con los privilegios respectivos.	
TAREA	
Numero de Tarea: 3	Numero de Historia: H9
Nombre de Tarea: Pruebas de Funcionamiento.	
Tipo de Tarea: Verificación	Puntos Estimados: 0,5
Programador Responsable: Yan Carlo Angarita	
Descripción: Se verifica el funcionamiento del módulo, que halla correlación con la base de los docentes, que guarde la información y la relación entre los cursos y los grupos estén bien estructurada. Se valida la seguridad, la navegabilidad y la velocidad de las consultas.	

Fuente. Autor del proyecto

Cuadro 31. Tareas Abarcadas para H10.

TAREA	
Numero de Tarea: 1	Numero de Historia: H9
Nombre de Tarea: Desarrollo de la Persistencia y Proceso de Negocio.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,3
Programador Responsable: Yan Carlo Angarita	
Descripción: Se crean la estructura de las nuevas tablas para soportar la autenticación del sistema, las entidades y demás requerimientos para el proceso de la persistencia.	
TAREA	
Numero de Tarea: 2	Numero de Historia: H9
Nombre de Tarea: Diseño de Interfaces.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,4
Programador Responsable: Yan Carlo Angarita	
Descripción: Se crean las interfaces para el ingreso al sistema, se validan los campos de usuario y contraseña, se desarrollan las políticas de seguridad de entrada al sistema y los privilegios necesarios.	

Cuadro 31. (Continuación)

TAREA	
Numero de Tarea: 3	Numero de Historia: H9
Nombre de Tarea: Pruebas de Funcionamiento.	
Tipo de Tarea: Verificación	Puntos Estimados: 0,3
Programador Responsable: Yan Carlo Angarita	
Descripción: Se verifica la autenticación en el sistema, que algún usuario no pueda acceder al sistema sin su respectiva clave, que no se pueda vulnerar la seguridad del sistema por algún método como inyección SQL y demás. Que el sistema verifique el tipo de usuario dependiendo del nombre de usuario que se registre.	

Fuente. Autor del proyecto

4.4 DISEÑO

4.4.1 Metáfora: Aplicación desarrollada en ambiente web que dará soporte al proceso de formación profesoral de los docentes de la universidad Simón Bolívar extensión Cúcuta. En el portal los docentes podrán ingresar y realizar el proceso de actualización de los datos, como lo son datos básicos, estudios, experiencias, formación continua, manejo de idiomas, manejo de software, eventos, distinciones y publicaciones; y posterior a esto realizar el procedimiento de validación de la información presentando los documentos originales que respaldan los datos ingresados al sistema.

Además permite la generación de reportes como la hoja de vida, que docentes han realizado el procedimiento de actualización y que actividades adicionales se asignaron al docente.

4.4.2 Tipo de arquitectura. Para el desarrollo del proyecto se utilizará una arquitectura de tres capas, un diseño que introduce una capa intermedia al proceso. En este tipo de arquitecturas a cada nivel se le confía una misión simple, lo que permite el diseño de arquitecturas escalables es decir, pueden ampliarse con facilidad en caso de que las necesidades aumenten.

Esta arquitectura estará basada en WEB, realmente es una forma modificada de la nueva arquitectura de tres capas, que utiliza un explorador en la estación de trabajo en lugar de la interfaz típica del usuario. Las soluciones basadas en WEB utilizan el protocolo World Wide Web, a través de la Internet o una intranet, para conectar las tres partes de la aplicación.

Las capas de la arquitectura están distribuidas en capa de presentación, capa de negocios y la capa de datos.

Capa presentación. En la capa de presentación se reúnen todos los aspectos del software que tiene que ver con las interfaces y la interacción con los diferentes tipos de usuarios humanos. Típicamente incluyen el manejo y aspecto de las ventanas, el formato de los reportes, menús, gráficos y elementos multimedia en general.

Esta capa es la que ve el usuario, le comunica la información y captura la información del usuario dando un mínimo de proceso. Esta capa se comunica únicamente con la capa de negocio.

En la arquitectura tradicional de tres capas se instala una interfaz de usuario en la computadora del usuario final. La arquitectura basada en WEB transforma la interfaz de búsqueda existente (el explorador de WEB), en la interfaz del usuario final.

En el proyecto actual se utilizó para la capa de presentación **Bootstrap**, el cual es una librería que mezcla todos los lenguajes de presentación como lo son HTML, JavaScript, CSS que ya tiene componentes establecidos como los botones, áreas de texto, menús, barras y demás componentes que solo con cambiar la clase cambia los colores, tamaños y funciones establecidos en la librería. Es una gran ventaja para el desarrollo ágil por que ya tiene la estructura preestablecida.

Capa de negocios. Esta capa reúne todos los aspectos del software que automatizan o apoyan los procesos de negocio que llevan a cabo los usuarios. Es donde residen los programas que se ejecutan, recibiendo las peticiones del usuario y enviando las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) pues es aquí donde se establecen todas las reglas que deben cumplirse. Estos aspectos típicamente incluyen las tareas que forman parte de los procesos, las reglas y restricciones que aplican.

Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar, almacenar o recuperar los datos requeridos por el usuario o los procesos necesarios para cumplir las solicitudes pertenecientes a cada una de ellas.

En el proyecto actual se aplica en la capa de negocios **PHP** el cual es un lenguaje muy factible de utilizar por ser uno de los más conocidos a nivel mundial.

Capa de datos. En esta capa es donde residen los datos, y en este proyecto se trabajara con la base de datos DB2 de IBM y la persistencia con la librería Doctrine; con la posterior creación de las entidades y modales que utiliza esta librería. (Ver Anexo E. Doctrine 2.0)

La base de datos que se utilizó es Db2 de IBM que fue en donde se montó toda la estructura diseñada, las tablas, las relaciones y demás herramientas necesarias.

En el diagrama de la base de datos se observan todas las tablas que se desarrollaron para la creación del portal para la administración de las actividades de formación de los docentes.

Las tablas que soportan el sistema principal, las tablas que soportan el sistema de login y las tablas que soportan la administración de los cursos de formación interna de la universidad, entre las demás que se observan.

En la base de datos se observa que una de las tablas principales es la del docente que es la que guarda la relación de que una persona se convierta en docente y al mismo tiempo la relación del docente con los estudios, las experiencias, investigaciones, publicaciones y demás.

La información del sistema principal es la que está en la parte izquierda de la imagen como lo son las tablas Persona, Lugar, Estado Civil, País, Eventos, Estudios, Experiencia, etc.

En la parte inferior derecha está la estructura de la base de datos que soporta el registro de los usuarios y la relación con los privilegios.

Se ven las tablas Usuario, Rol, Perfil, UsuarioRol, UsuarioPerfil y PerfilRol.

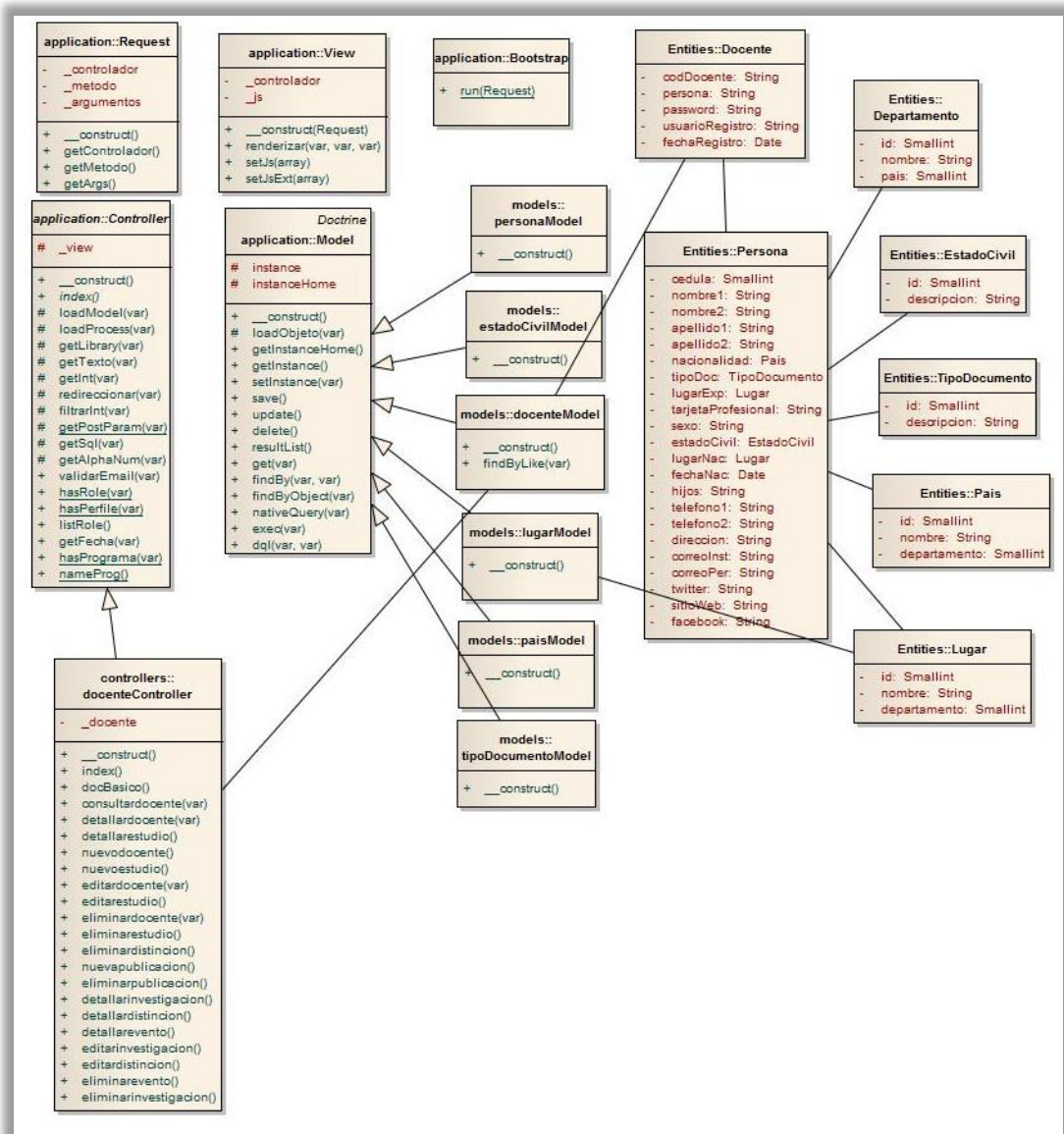
Entre estas tablas se maneja el que cuando un usuario se registra se carguen en sesión los roles que estén en UsuarioRol, los perfiles que estén en UsuarioPerfil y al mismo tiempo los roles que estén en PerfilRol. Al final del proceso lo que se le carga al usuario en sesión es una lista de privilegios o roles.

En la parte superior derecha se observa la estructura de los cursos de formación interna de la universidad donde hay tablas como Grupo, Curso, Tutor y demás las cuales soportan todo el procedimiento de los cursos de formación.

En anexos estará el diccionario de datos de la estructura de la base de datos.

4.5.2 Diagrama de Clases. Diagrama Gestionar Información Básica del Docente.

Figura 4. Gestionar Información Básica del Docente.



Fuente. Autor del proyecto

En el diagrama de clase se observa en la parte superior izquierda la clase Bootstraps, View, Controllers, Model y Config que son las que conforman la base del modelo MVC trabajado.

La clase Bootstraps.php contiene todo el manejo de las URL amigables y como se manejan estas en su distribución de controladores y argumentos. Qué hacer cuando en la URL viene sin parámetros, o viene con algunos parámetros; el saber cómo identificarlos y separarlos.

La clase Config.php contiene toda la información de inicio por defecto. Esta clase es la que se busca apenas empiece el sistema a correr; datos como local host por defecto o dirección IP por defecto entre otros.

Controller.php es la clase principal de todos los controladores la cual le heredara todas sus funciones. En esta clase se suben las variables a sesión y demás acciones para poder acceder a ellos desde cualquier controlador hijo.

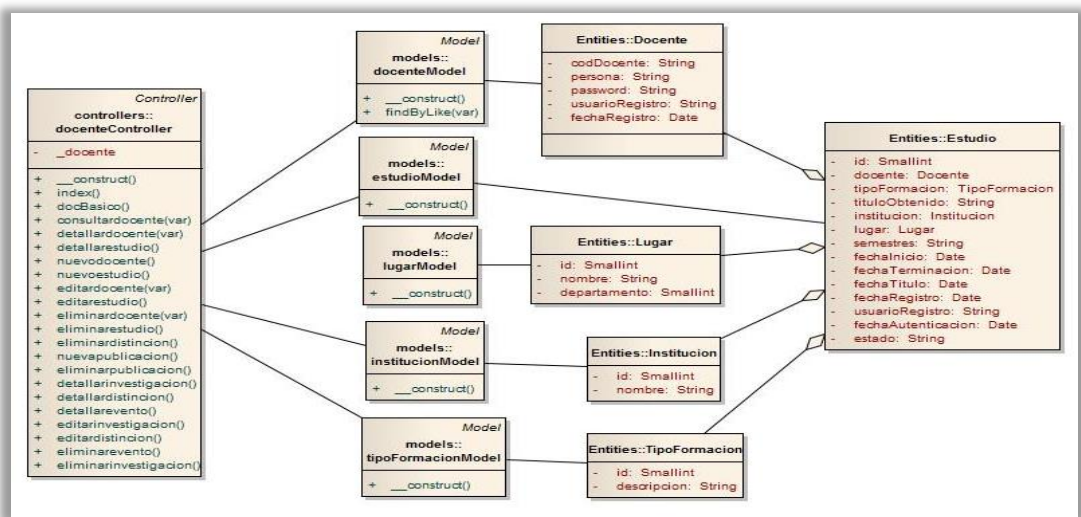
La clase Model.php es la clase principal de todos los modelos. Esta contiene todos los métodos y funciones que se encargan de las conexiones de la base de datos y comunicación a los modales y las entidades.

La clase View.php es la clase principal de todas las vistas. Contiene todo lo referente a la vista del sistema, a las interfaces, al manejo de javascript, Ajax y demás lenguajes usados. Las clases que están a la derecha del diagrama son las Entidades, las cuales son la representación de las tablas. Estas entidades tienen los atributos y métodos.

Se crea la clase docenteController la cual es la que conecta el modelo con la vista, es la columna vertebral del módulo. Aquí se llaman los modales como el personaModal que a su vez carga la entidad Persona. Se realizan las búsquedas necesarias y procedimientos enviando los resultados a la vista docente y ListDocente para ser mostradas al usuario. En la clase docenteController se encuentra los métodos guardar, editar, consultar, eliminar la información del docente los cuales se irán agregando más métodos a medida que el proyecto aumente.

Diagrama Gestionar Formación Académica del Docente.

Figura 5. Gestionar Formación Académica del Docente.



Fuente. Autor del proyecto

Clases que intervienen: docenteController, Docente, TipoFormacion, Lugar, Institución, docenteModel, TipoFormacionModel, LugarModel, institucionModel, estudio, estudioModel.

En este diagrama de clase se ve que la única clase que utilizamos de las ya planteadas es docenteController y docenteModel, las nuevas clases se ven en el diagrama que son las utilizadas en el desarrollo del módulo de Formación Académica de los docentes.

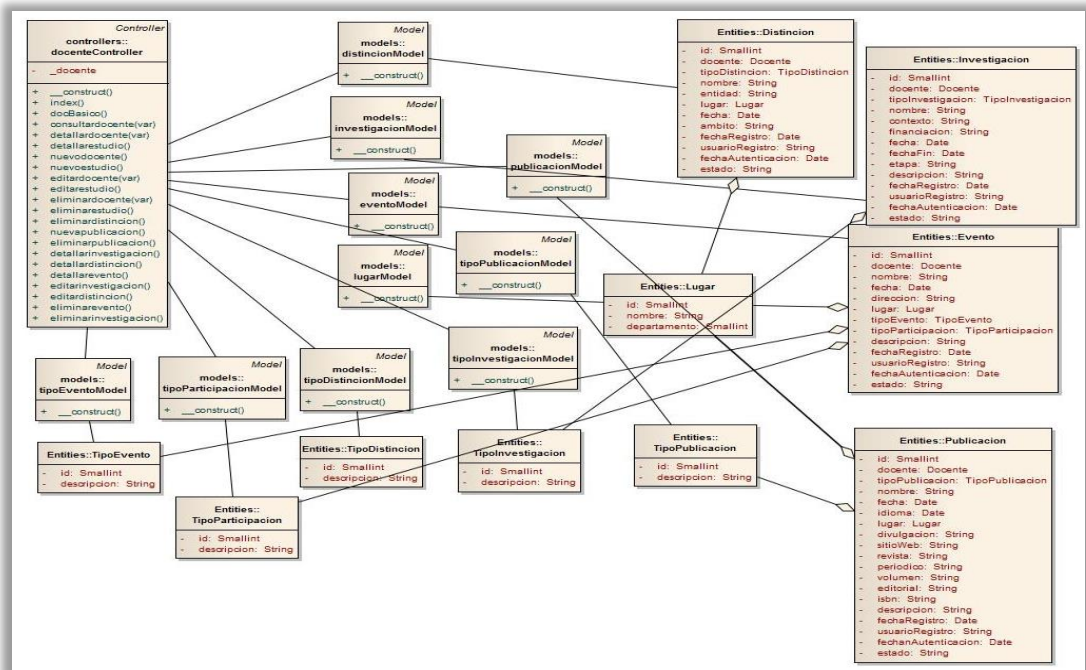
Se agregó a la clase docenteController los métodos de guardar, consultar, modificar y eliminar la formación académica del docente y se desarrolló los métodos para la seguridad por medio de privilegios.

Se carga todos los modales que intervienen y estos a su vez cargan las entidades para realizar las consultas y operaciones. Luego se mandan todos los objetos y parámetros a la vista estudios y listestudios para realizar el procedimiento de administración de estudios validando las fechas, los campos vacíos y la seguridad.

Todo este procedimiento se realiza en la gestión de los estudios validando y guardando la información en la base de datos.

Diagrama Gestionar Producción del Docente.

Figura 6. Gestionar Producción del Docente.



Fuente. Autor del proyecto

Clases que interviene: docenteController, docenteModal, publicacionModal, investigacionModal, distincionModal, eventosModal, tipoEventoModal, tipoDistincionModal, tipoInvestigacion, tipoPublicacionModal, Publicación, Investigación, Distinción, Eventos, TipoEvento, TipoDistincion, Investigacion, TipoPublicacion

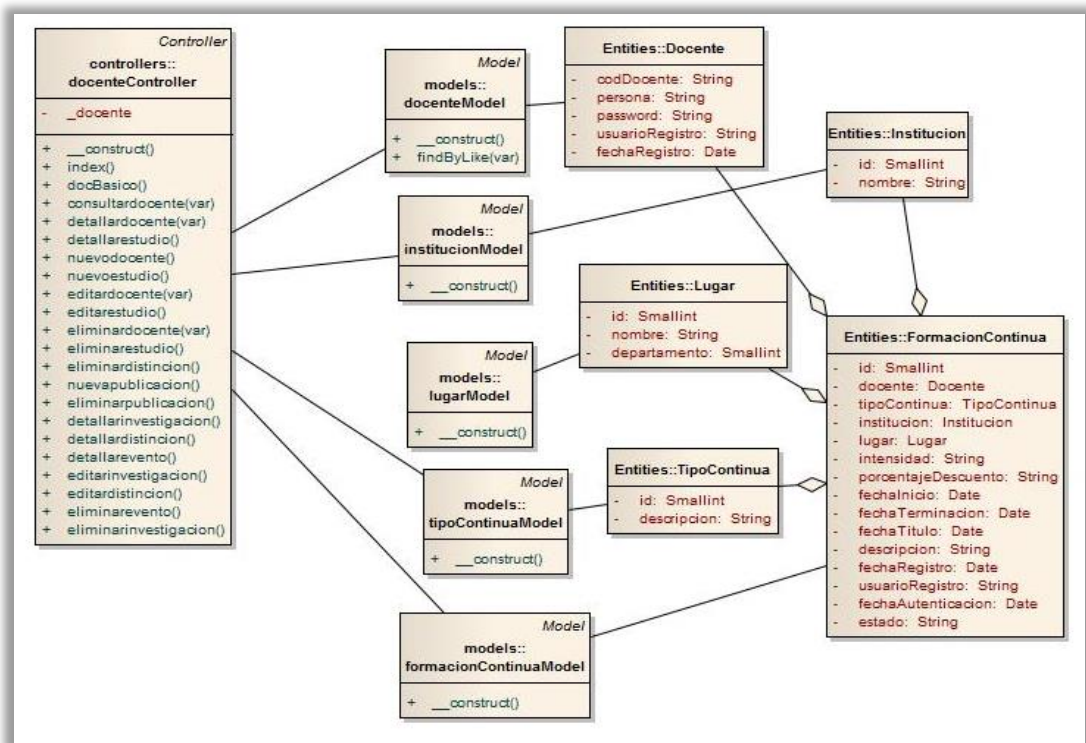
En este diagrama se ve que se sigue trabajando con la clase principal docenteController. Primero se carga el publicacionModel el cual cargara la entidad Publicación; con esto ya se desarrolla la parte de gestión de las publicaciones que son el registro, la consulta, la edición y la eliminación.

Lo siguiente es cargar el investigacionModal que a su vez carga la entidad investigación, y se realizan los métodos de administración de las investigaciones; registra, eliminar, modificar y consultar.

Para los eventos y distinciones se hace el mismo procedimiento de crear los métodos y cargar las clases. Para mandar por ejemplo el listado de instituciones a la vista se carga en el docenteController un objeto con los datos y en la vista se recorre en un foreach.

Diagrama Formación Complementaria del Docente.

Figura 7 Gestionar Información Básica del Docente.



Fuente. Autor del proyecto

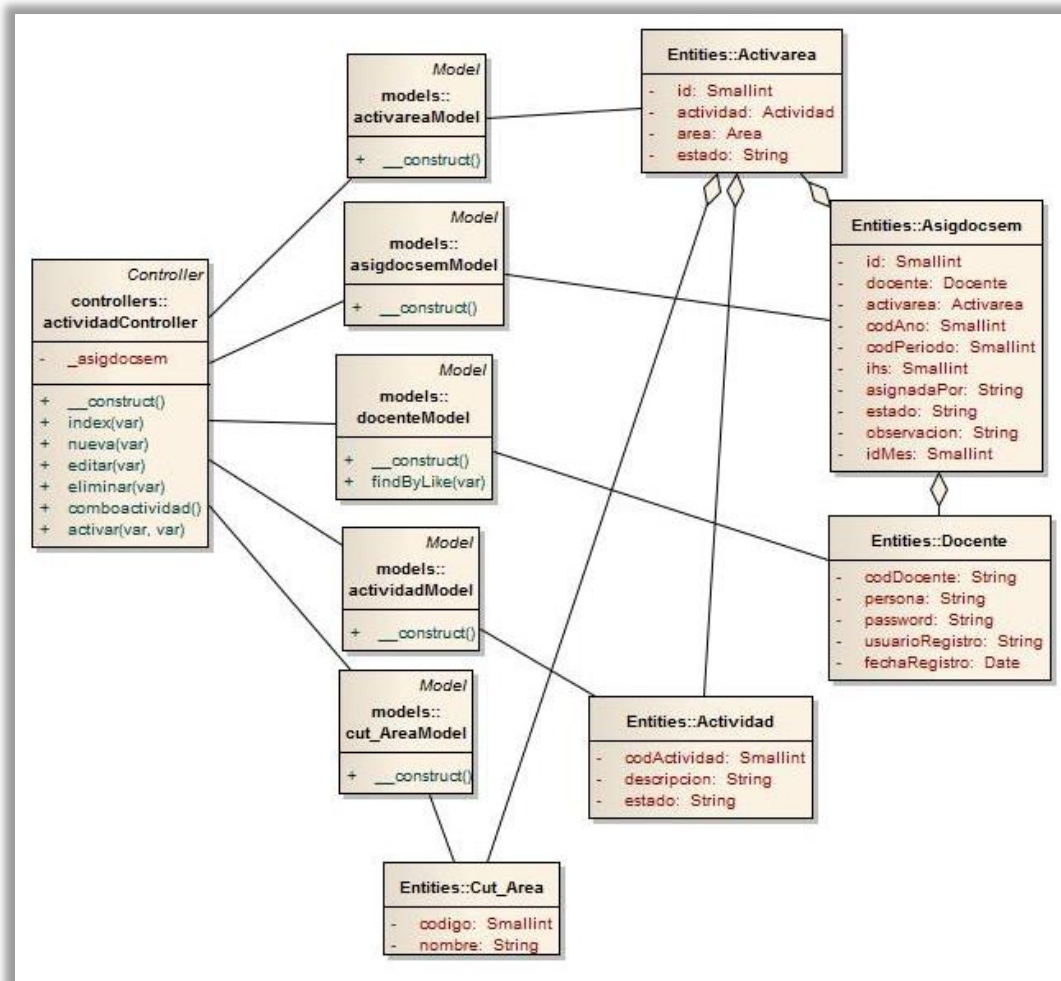
Clases que intervienen: docenteController, docenteModel, Docente, TipoFormacion, tipoFormacionModel, Lugar, lugarModel, Institución, institucionModel

Aquí de igual forma que antes, nuestra clase, controller principal es el del docente, donde estarán cargados los modelstipoformacion, lugar, institución y docente; así se podrán cargar las entidades docente, tipoformacion, institución y lugar. Ya con estos datos cargados se pueden hacer las consultas por los métodos findBy, finByObject, get y demás métodos para luego de esto mandarlos a la vista como objetos.

Con esto después de haber cargado los datos del formulario se guardan o edita la información; y de igual manera se puede eliminar o consultar los datos.

Diagrama Gestionar Actividades Adicionales del Docente.

Figura 8. Gestionar Actividades Adicionales del Docente.



Fuente. Autor del proyecto

Clases que intervienen: actividadController, actividad, área, activarea, asigdocsem, docente, docenteModel, actividadModel, areaModel, activareaModel, asigdocsemModel.

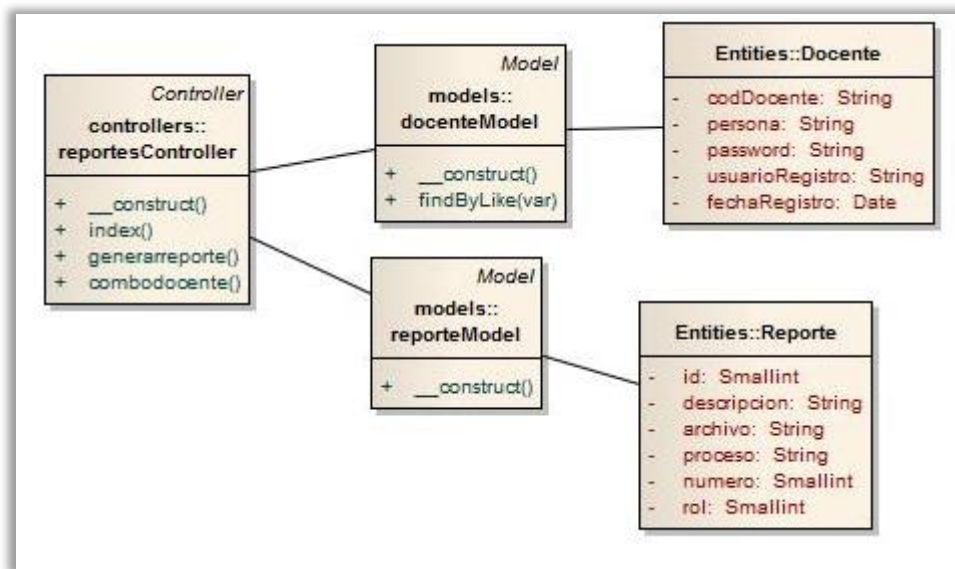
En este diagrama ya deja de ser el docenteController el controlador principal y pasa a ser actividadController y de igual forma cambiamos las vistas a unas nuevas que serán listactividad y actividad. Se comienza de igual forma a cargar los modales docentes para relacionar las actividades con los docentes; se carga el actividadModel, areaModel, activareaModel y asigdocsemModel para lo relacionado a las actividades adicionales.

Se desarrolla en el actividadController los métodos para administrar las actividades, registrar, modificar, consultar y eliminar. Se envían los objetos a la vista y ahí se relacionan las actividades a un docente, y a la vez una actividad es una fusión entre área y actividades.

Asigdocsem hace referencia a la asignación docente semanal.

Diagrama Gestionar Reportes.

Figura 9. Gestionar Reportes.



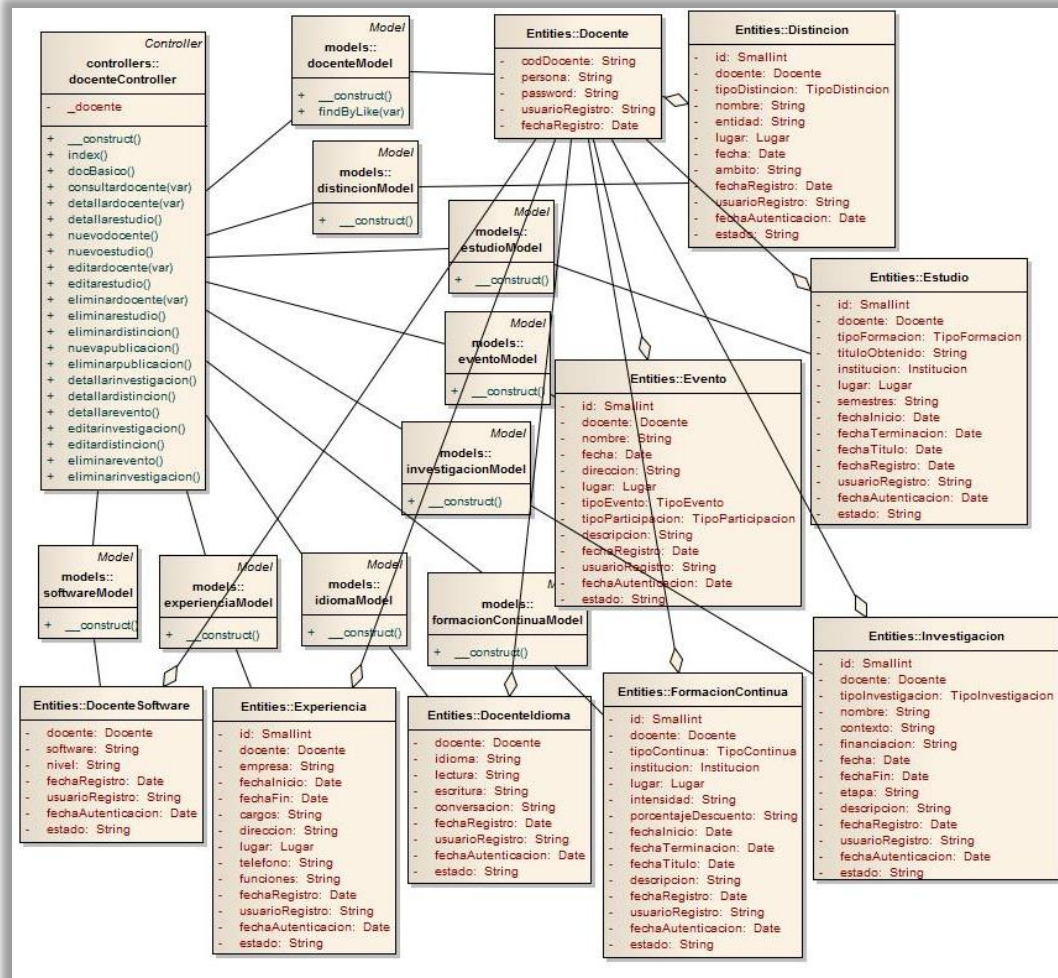
Fuente. Autor del proyecto

Clases que intervienen: reportesController.php, reportesModel.php, reportes.phtml, docenteModel, docente, reportes

Aquí se desarrolla el módulo de los reportes el cual utiliza la clase reportesControllers donde se incluye el reporteModel y carga la entidad reportes. De aquí se captura el valor del reporte desde la vista y todos los parámetros necesarios para generarlo; buscando el reporte hecho en ireport en el directorio raíz y realizando el procedimiento de generación del reporte.

Diagrama Validar Información del Docente.

Figura 10. Validación Información del Docente.



Fuente. Autor del proyecto

Clases que intervienen: `docenteController`, `docenteModel`, `Docente`, `Estudios`, `Experiencia`, `Investigación`, `Publicación`, `Formacioncontinua`, `Idioma`, `Software` `Eventos`.

En este diagrama se trabaja la validación de la información ya ingresada gracias a los diagramas anteriores. Se vuelve a trabajar con el `docenteController` y se cargan todos los modales y entidades principales en el sistema como son el `Docente`, `Estudio`, `Experiencia`, `FormacionContinua`, `Evento`, `publicación`, `investigación`, `distinción`, `idioma` y `software`.

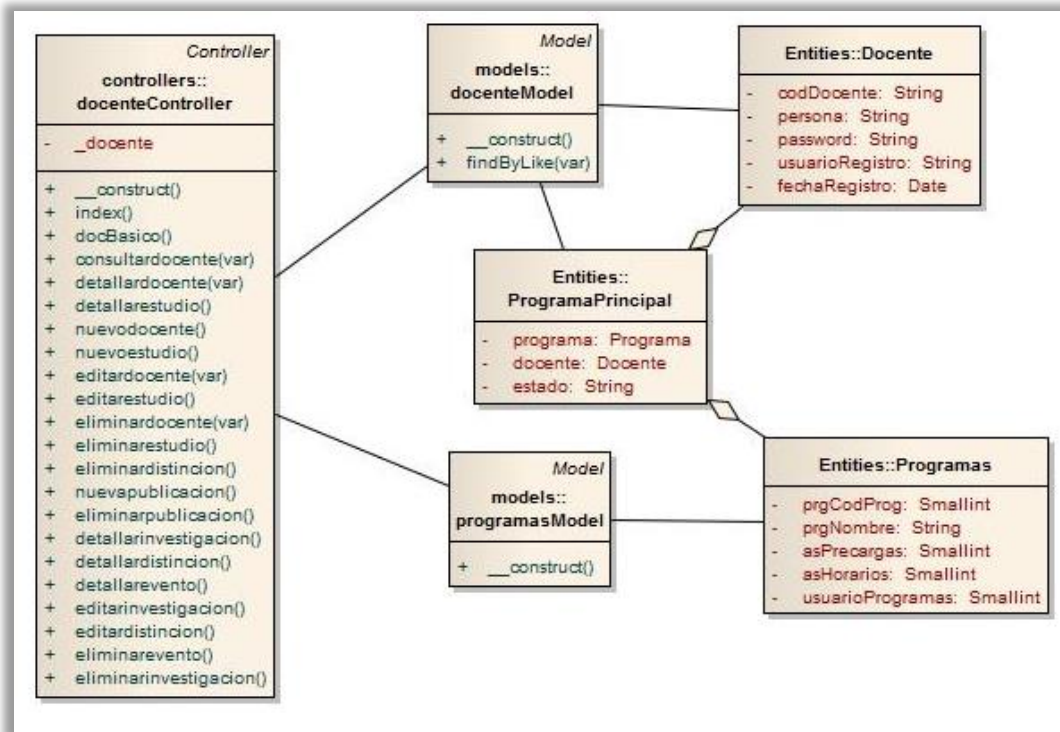
Después se desarrolla el método para activar la información que recibe el docente y el identificador del dato para activar; y de igual forma se valida que solo los usuarios con privilegios puedan realizar este procedimiento y que una vez activada los datos se bloqueen para ser editados por usuarios docentes.

Este procedimiento permite que una vez estando la información activada pueda ser visible para el reporte de la hoja de vida del docente.

El usuario que lleva a cabo la activación es Recursos Humanos que tiene que pedir los documentos físicos reales.

Diagrama Administrar Pertenencia de los Docentes.

Figura 11. Administrar Pertenencia de los Docente.



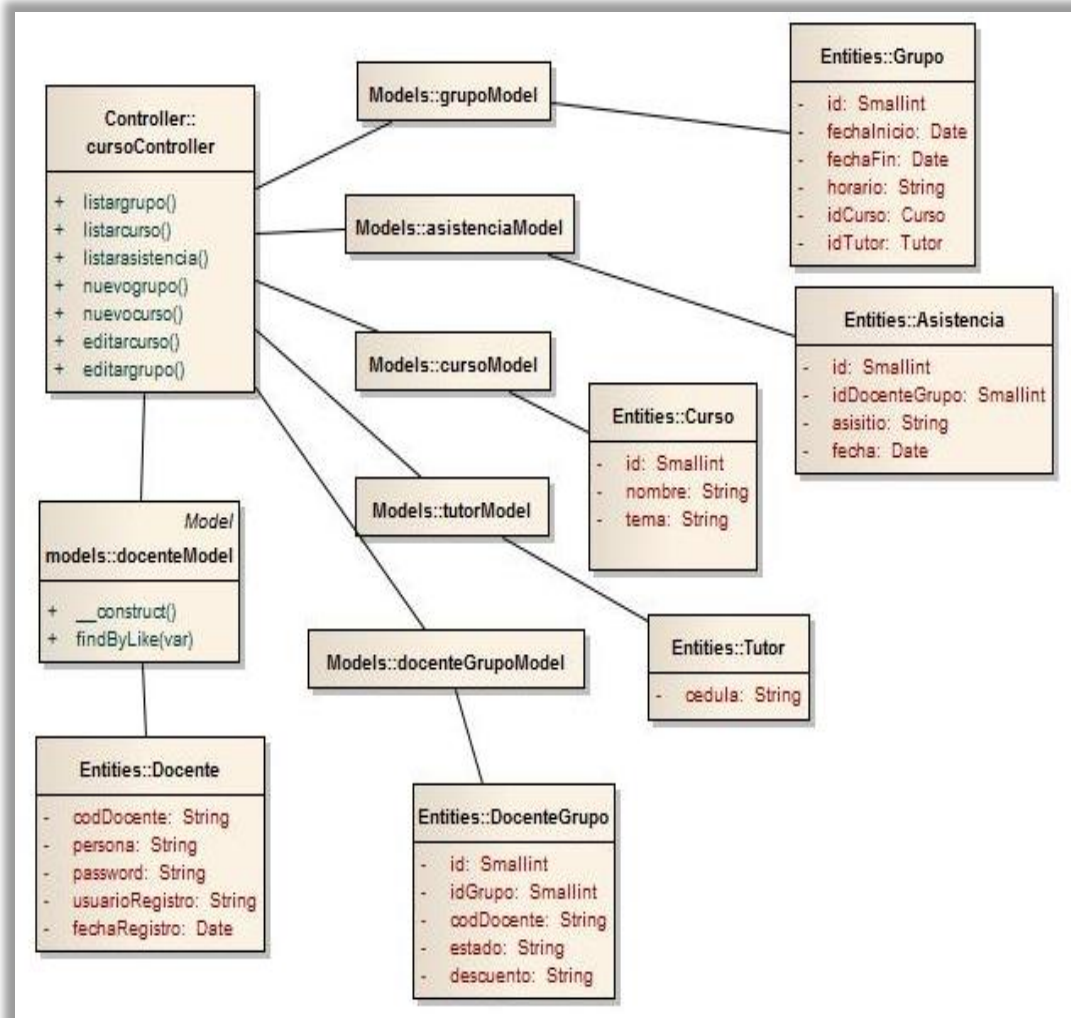
Fuente. Autor del proyecto

Clases que intervienen: Docente, ProgramaPrincipal, docenteController, docenteModel, programaPrincipalModel

En el diagrama actual se observa el desarrollo del módulo de pertenencia de los docentes a los programas. Como base se vuelve a usar el docenteController el cual se encarga se conecta los modales y las entidades para la persistencia; además la vista para capturar y consultar la información.

Diagrama Gestionar Cursos de Formación Académica.

Figura 12. Gestionar Cursos de Formación Académica.



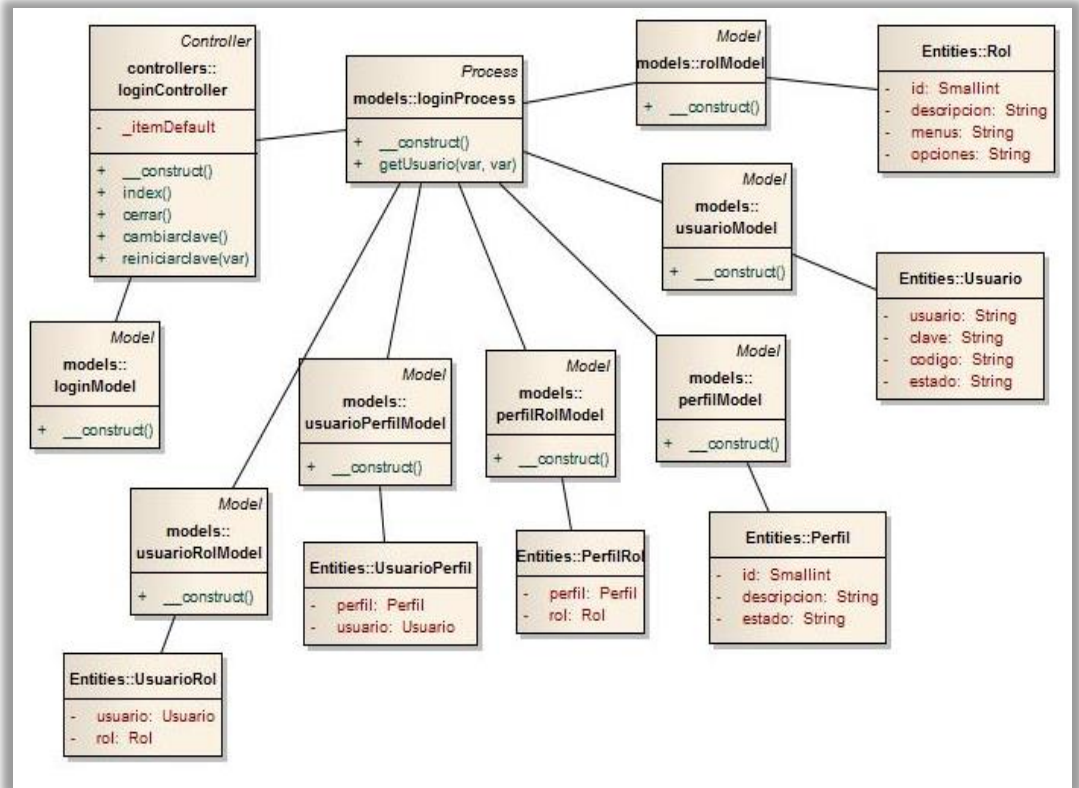
Fuente. Autor del proyecto

Clases que intervienen: **cursoController**, **docenteModel**, **Docente**, **docenteGrupoModel**, **DocenteGrupo**, **tutorModel**, **Tutor**, **cursoModel**, **Curso**, **asistenciaModel**, **Asistencia**, **grupoModel**, **Grupo**.

En este caso todo va alrededor de **cursoController**, el cual es quien carga todos los models, carga las entidades, procesa todos los datos y los envía a la vista para formar la interfaz. Gracias a esto se puede administrar toda la información de los cursos de formación continua internos de la universidad.

Autenticación del Sistema.

Figura 13. Autenticación del Sistema.



Fuente. Autor del proyecto

Clases que intervienen: Usuario, Rol, Perfil, UsuarioRol, UsuarioPerfil, PerfilRol, usuarioModel, rolModel, perfilModel, usuarioRolModel, usuarioPerfilModel, perfilRolModel, loginProces, loginController.

En el último diagrama se desarrolla el módulo de autenticación del sistema. Se utilizan las entidades Usuario que representa los usuarios del sistema; la entidad rol que representa los roles del sistema; la entidad perfil que representa los perfiles del sistema.

Asimismo las entidades usuariorol tiene la relación del usuario y los roles que tiene asignado; la entidad usuarioperfil tiene la relación entre el perfil y los usuarios a los que tiene asignados; la entidad perfilrol son los roles que están asignados a cada perfil.

De igual forma se cargan todos los modales necesarios para el desarrollo del módulo.

4.5.3 Pruebas de aceptación. Para realizar las pruebas de aceptación se utilizaron diferentes tipos de software y herramientas que actualmente se utilizan para garantizar el proceso de las

pruebas; tomando todo aquello que se hacía manualmente y convirtiéndolo en las nuevas herramientas de validación de software.

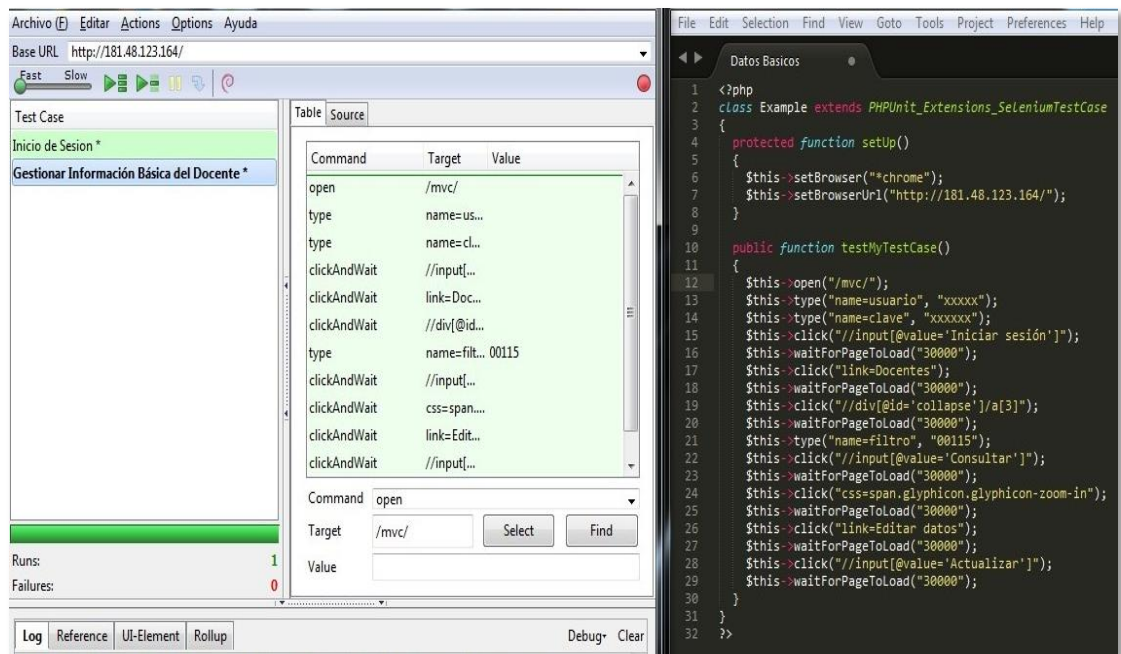
Uno de esas herramientas es Selenium el cual es un plugins del navegador Firefox que ayude al procedimiento de pruebas de navegación y comunicación entre las capas. Lo que se hace es grabar una prueba, la cual consiste en realizar un procedimiento en el navegador con el Selenium en ejecución, por ejemplo el procedimiento de identificarse en el sistema, digita el usuario, digita la clave, da clic en iniciar y demás; Selenium graba el procedimiento para después ejecutarlo y dar una respuesta de que si hubo errores o el procedimiento estuvo correcto.

Después de trabajar con el Selenium se toma y se exporta el procedimiento a código PHPUnit, el cual es otra herramienta para realizar las pruebas. Se toma el código y se crea una clase para realizar pruebas y así se verifica los procedimientos a nivel de código unicidad entre las clases.

Gestionar Información Básica del Docente. Se realizó el procedimiento de actualizar un registro en la base de datos de un docente sobre su información básica y se generó la prueba.

Aquí hubo algunos errores por la compatibilidad del nuevo sistema al que ya estaba en uso con los datos básicos del docente. Se encontraron errores en el momento de registrar los datos los cuales se solucionaron ajustando tipos de datos.

Figura 14. Prueba Información Básica del Docente.



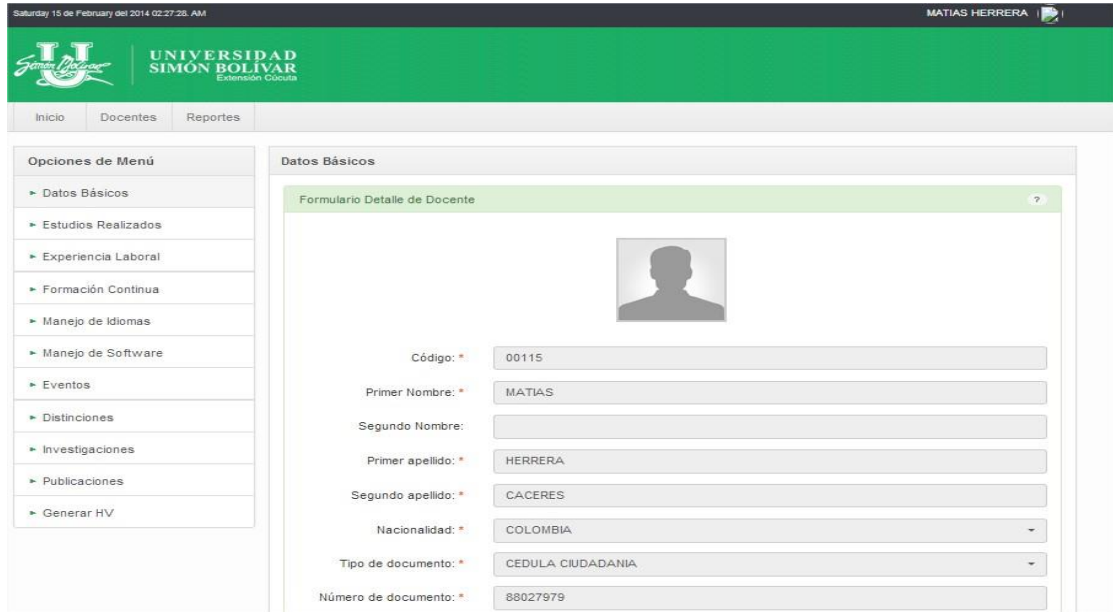
The image shows two side-by-side screenshots. The left screenshot is from Selenium IDE, displaying a test case named 'Gestionar Información Básica del Docente *'. It features a table with columns 'Command', 'Target', and 'Value'. The table contains the following entries:

Command	Target	Value
open	/mvc/	
type	name=us...	
type	name=cl...	
clickAndWait	//input[...	
clickAndWait	link=Doc...	
clickAndWait	//div[id...	
type	name=filt...	00115
clickAndWait	//input[...	
clickAndWait	css=span...	
clickAndWait	link=Edit...	
clickAndWait	//input[...	

Below the table, the 'Command' field is set to 'open', the 'Target' to '/mvc/', and the 'Value' field is empty. The right screenshot shows the corresponding PHPUnit code in a dark-themed editor. The code defines a class 'Example' extending 'PHPUnit_Extensions_SeleniumTestCase' and includes a 'setUp()' method and a 'testMyTestCase()' method. The 'testMyTestCase()' method contains Selenium commands translated into PHP, such as '\$this->open("/mvc/")', '\$this->type("name=usuario", "xxxxx")', and '\$this->click("//input[@value='Iniciar sesión']");'. The code ends with '\$this->waitForPageToLoad("30000");' and a closing brace for the test method.

Fuente. Autor del proyecto

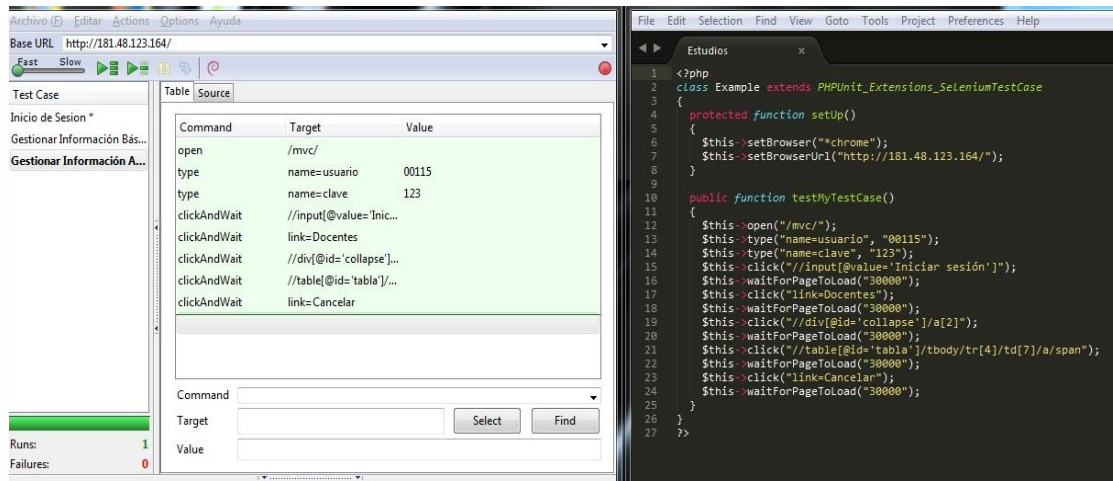
Figura 15. Interfaz Información Básica del Docente.



Fuente. Autor del proyecto

Gestionar Formación Académica de los Docentes. Se realizó el procedimiento de registro de información académica del docente lo cual al principio mal por falta de una llave primaria en la tabla la cual debía ser auto incremental y no lo era porque el manejador de base de datos DB2 versión 6 no lo soporta. La solución fue crear un procedimiento con la ayuda de Doctrine y crear los datos autoincrementales. Con esto se solucionaban el resto de campos autoincrementales para los demás módulos. Aquí está la prueba.

Figura 16. Prueba Formación Académica del Docente.



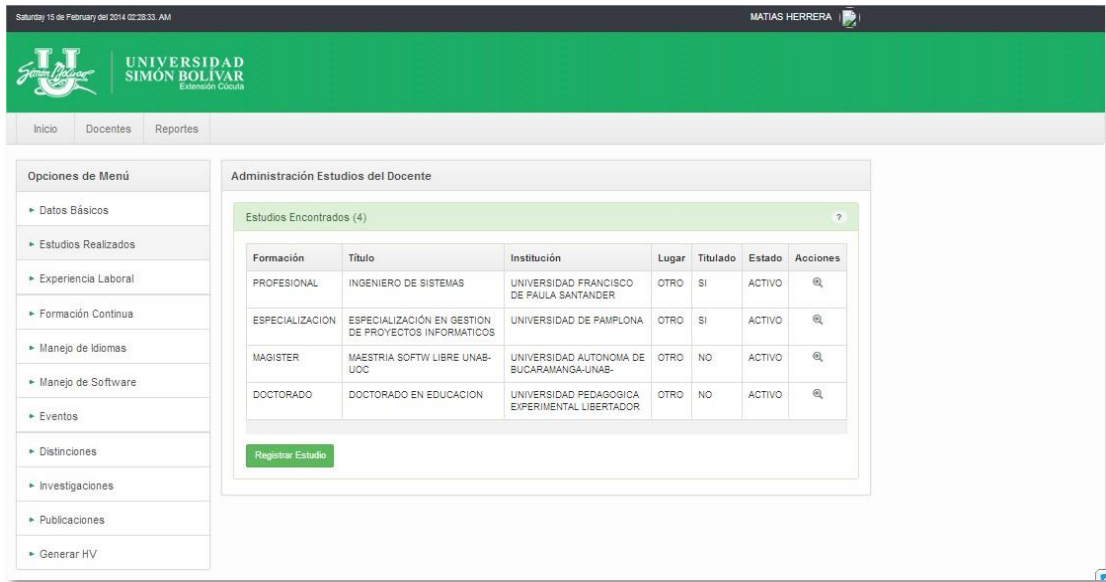
Command	Target	Value
open	/mvc/	
type	name=usuario	00115
type	name=clave	123
clickAndWait	//input[@value='Inic...]	
clickAndWait	link=Docentes	
clickAndWait	//div[@id='collapse']...	
clickAndWait	//table[@id='tabla']/...	
clickAndWait	link=Cancelar	

```

1 <?php
2 class Example extends PHPUnit_Extensions_SeleniumTestCase
3 {
4     protected function setUp()
5     {
6         $this->setBrowser("**chrome");
7         $this->setBrowserUrl("http://181.48.123.164/");
8     }
9
10    public function testMyTestCase()
11    {
12        $this->open("/mvc/");
13        $this->type("name=usuario", "00115");
14        $this->type("name=clave", "123");
15        $this->click("//input[@value='Iniciar sesión']");
16        $this->waitForPageToLoad("30000");
17        $this->click("link=Docentes");
18        $this->waitForPageToLoad("30000");
19        $this->click("//div[@id='collapse']/a[2]");
20        $this->waitForPageToLoad("30000");
21        $this->click("//table[@id='tabla']/tbody/tr[4]/td[7]/a/span");
22        $this->waitForPageToLoad("30000");
23        $this->click("link=Cancelar");
24        $this->waitForPageToLoad("30000");
25    }
26 }
27 ?>
    
```

Fuente. Autor del proyecto

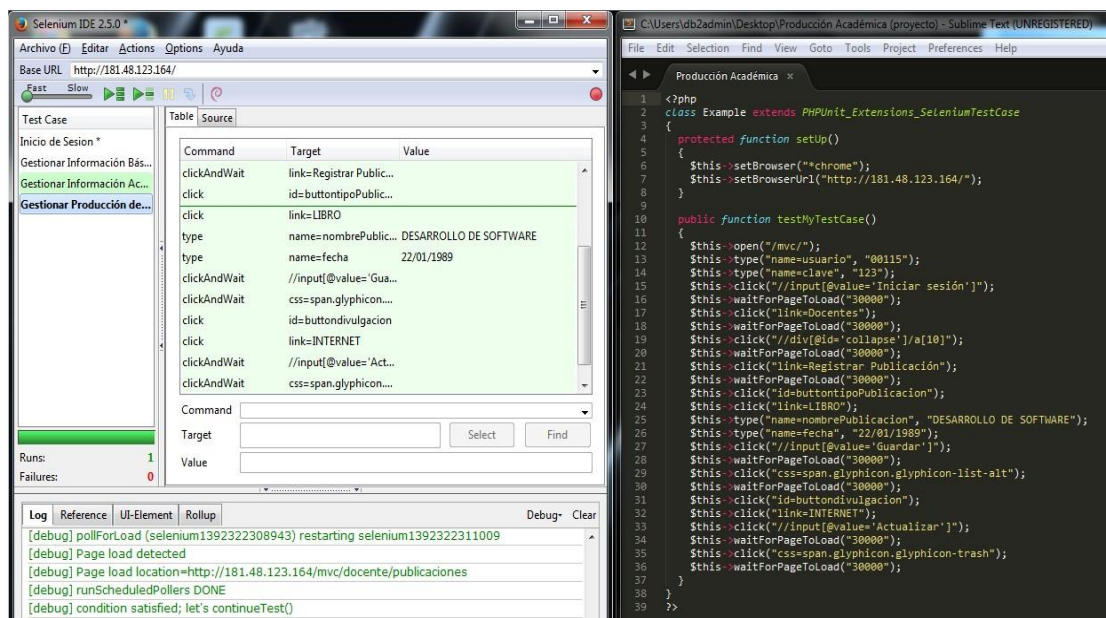
Figura 17. Interfaz Formación Académica del Docente.



Fuente. Autor del proyecto

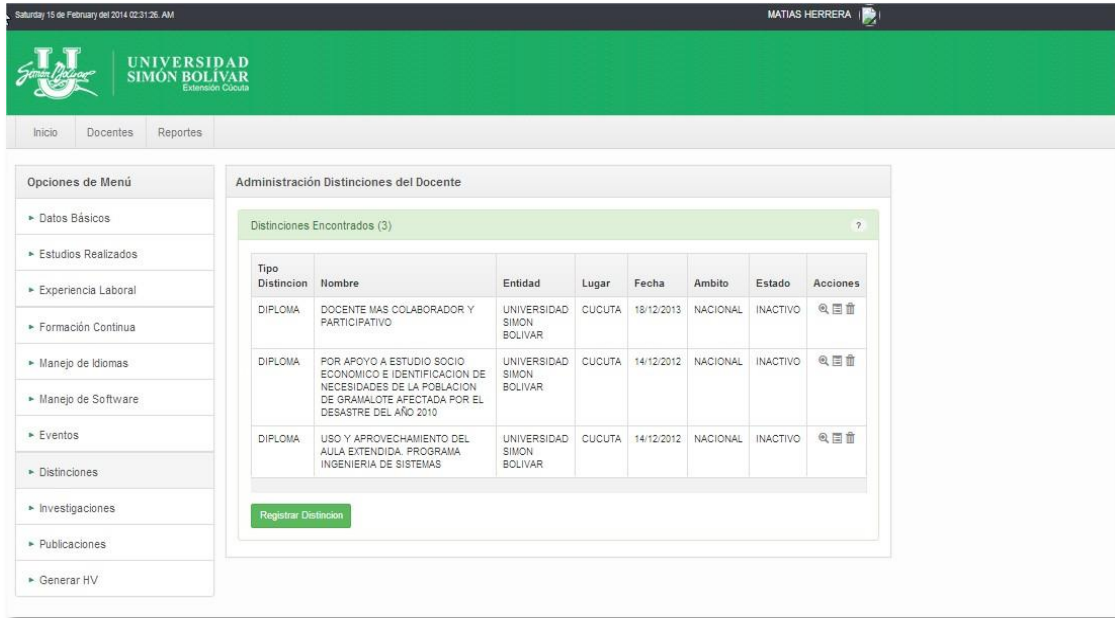
Gestionar Producción de los Docentes. Se realizó el procedimiento de gestionar la producción del docente donde se administraron las publicaciones.

Figura 18. Prueba Producción de los Docentes.



Fuente. Autor del proyecto

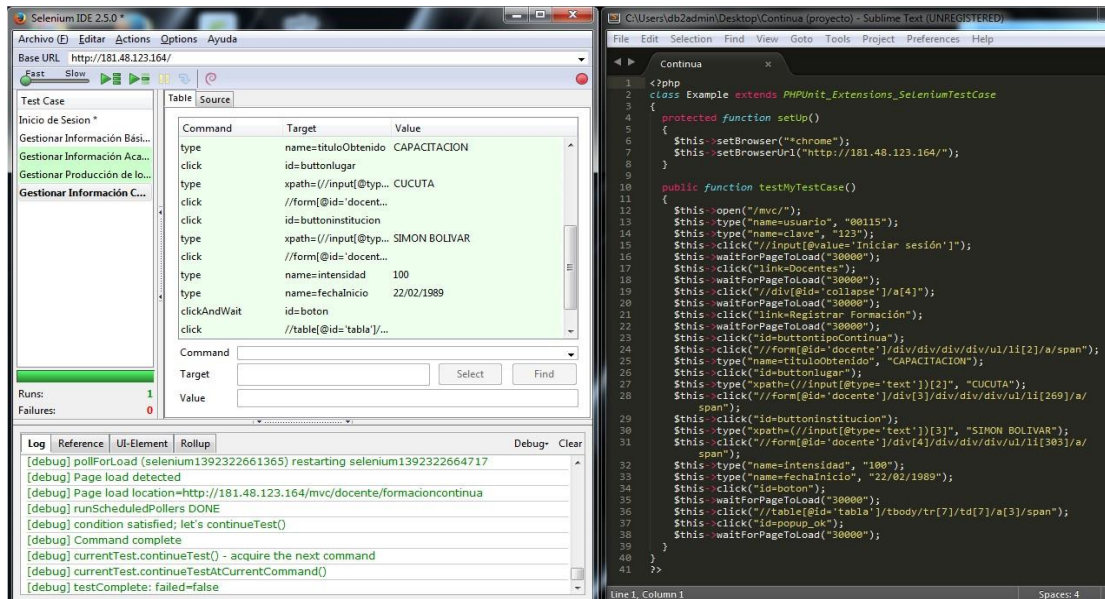
Figura 19. Interfaz Producción de los Docente.



Fuente. Autor del proyecto

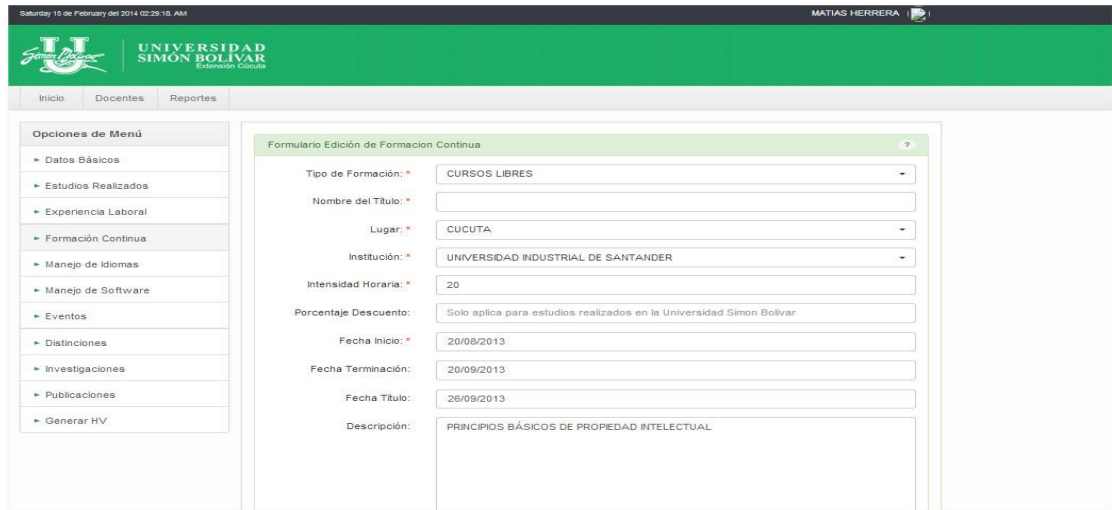
Gestionar Formación Continua de los Docentes. Se realizó la gestión de la formación continua, y después de encontrar algunos faltantes en la historia de usuario de este proceso se pudo actualizar y realizar el correcto funcionamiento.

Figura 20. Prueba Formación Continua de los Docentes.



Fuente. Autor del proyecto

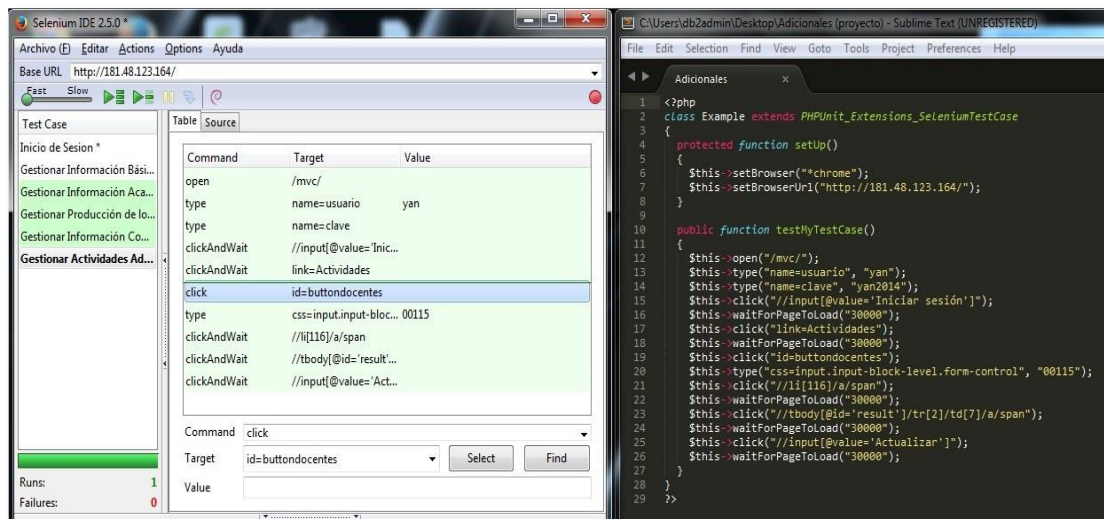
Figura 21. Interfaz Formación Continua de los Docente.



Fuente. Autor del proyecto

Gestionar Actividades Adicionales. Realizando el proceso de asignación de actividades adicionales a los docentes se encontraron problemas en que el usuario debía hacer demasiados clics para entrar y realizar las operaciones necesarias; por lo que toco corregir la navegación y adaptarla de tal forma que sea fácil y rápido el manejo de la administración de las actividades adicionales de los docentes.

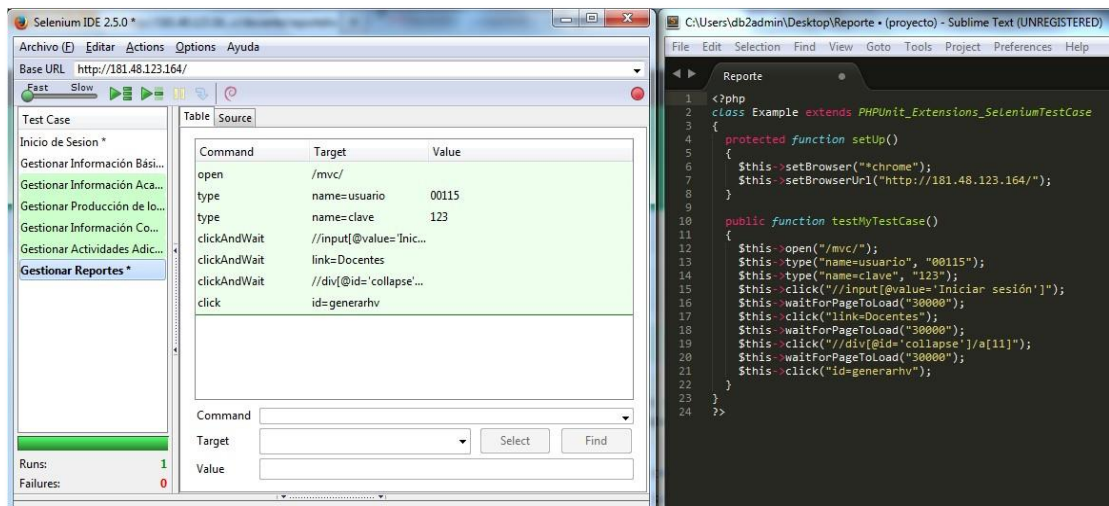
Figura 22. Prueba Actividades Adicionales del Docente.



Fuente. Autor del proyecto

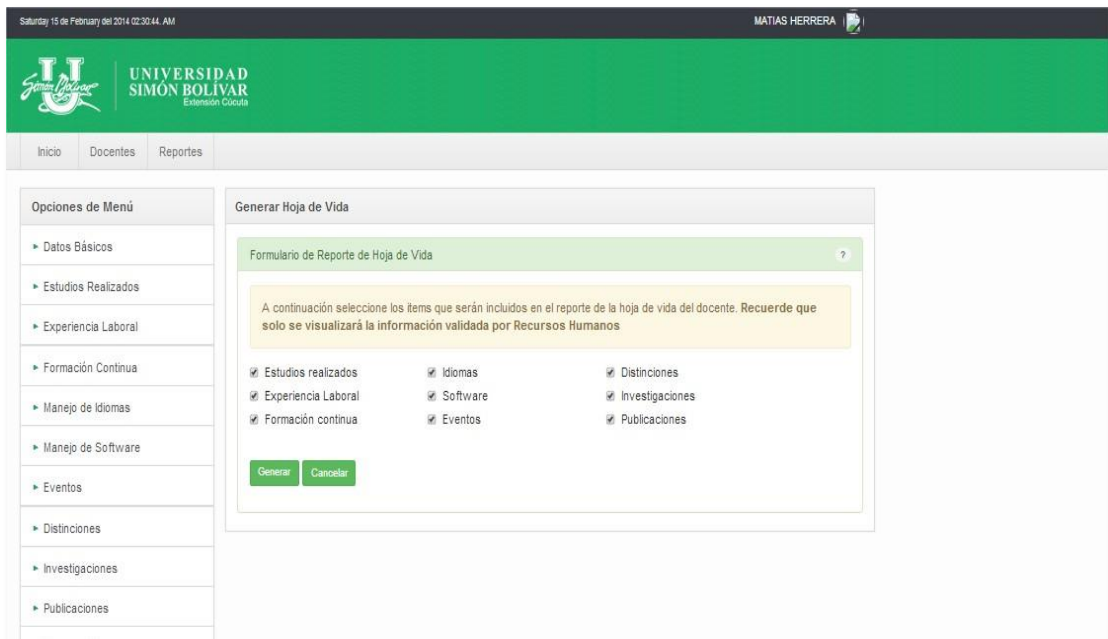
Gestionar Reportes. Los problemas encontrados fueron por el servidor JAVA que utilizaba el iReport para generar los reportes. El servidor había que iniciarlo y configurarlo para que cuando se llamara el reporte desde php se generara correctamente. Al final se solucionó el problema implementando métodos de control y seguridad para la generación de los reportes.

Figura 23. Prueba Reportes.



Fuente. Autor del proyecto

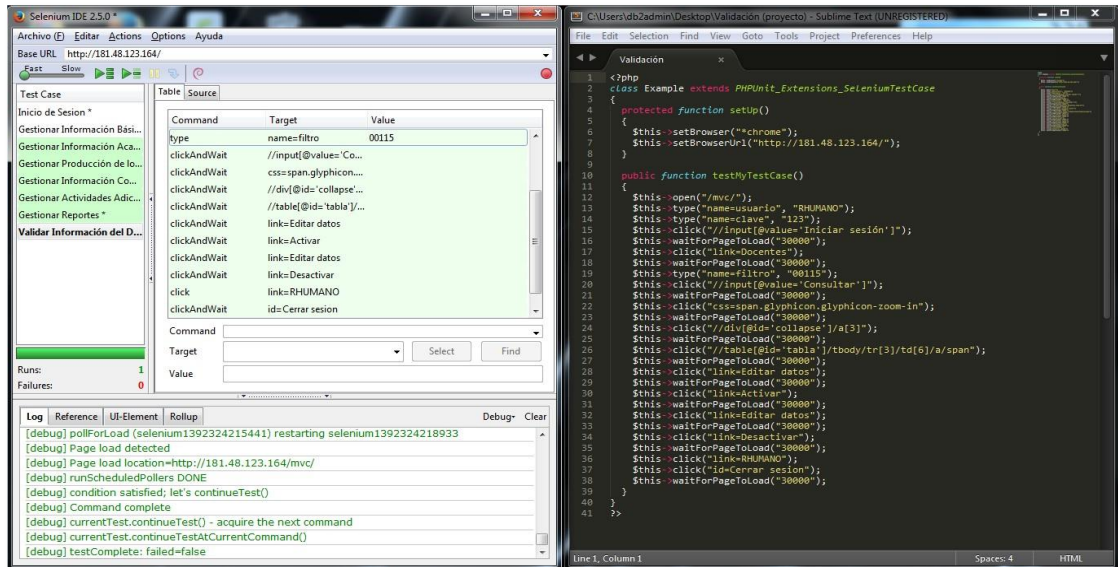
Figura 24. Interfaz Reportes



Fuente. Autor del proyecto

Validación de la Información Docentes. Se realizó el procedimiento de la validación de la información de los docentes.

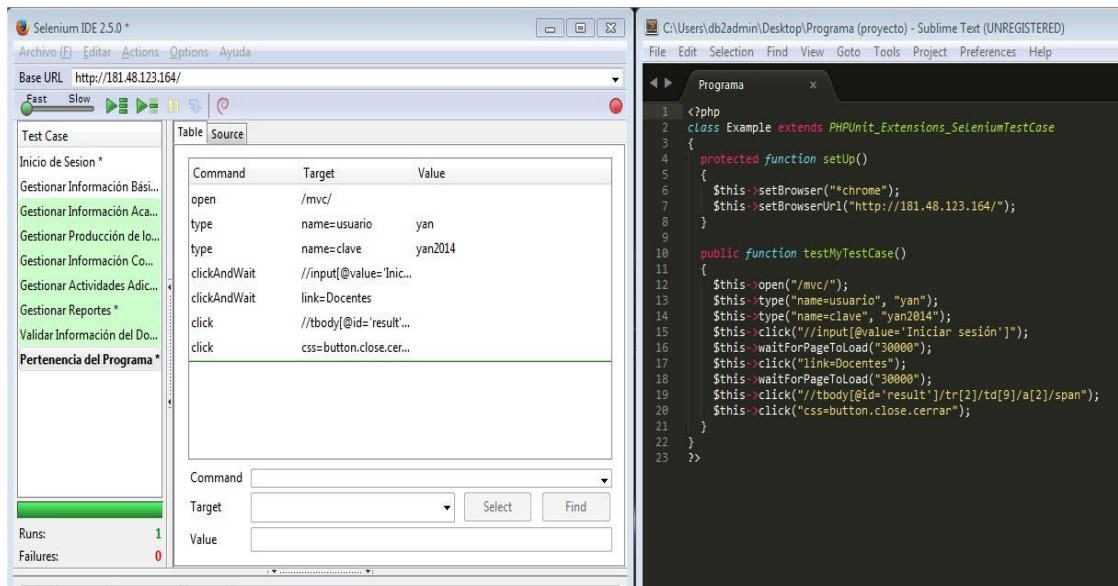
Figura 25. Prueba Validación de la Información de los Docente.



Fuente. Autor del proyecto

Asignación del Programa Principal al Docente. Se realizó la asignación del programa principal a los docentes.

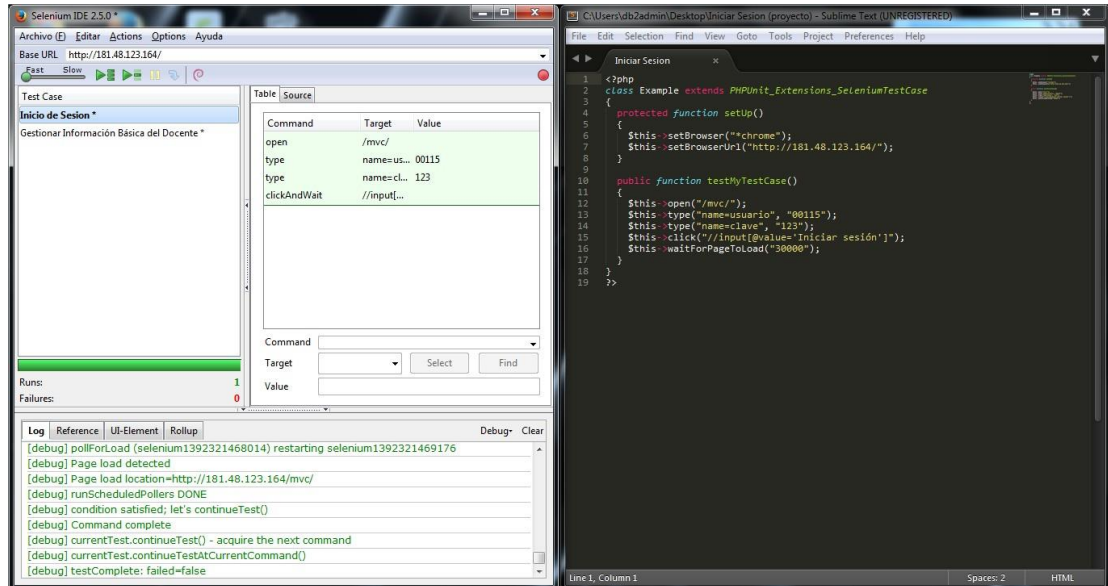
Figura 26. Prueba Asignación del Programa Principal del Docente.



Fuente. Autor del proyecto

Autenticación del Sistema.

Figura 27. Prueba Autenticación del Sistema.



Fuente. Autor del proyecto

Figura 28. Interfaz Autenticación del Sistema.

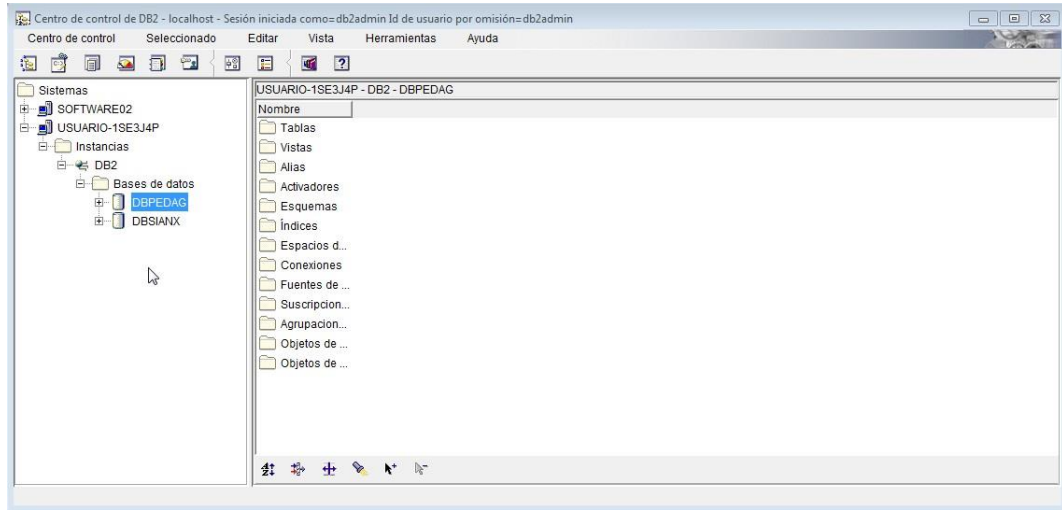
The image shows a web form for authentication. It has a title 'Iniciar sesión' at the top. Below the title are two input fields: 'Nombre de usuario' and 'Contraseña'. At the bottom of the form is a green button labeled 'Iniciar sesión'.

Fuente. Autor del proyecto

4.6. IMPLEMENTAR Y PONER EN MARCHA EL PORTAL WEB.

Para la implementación lo primero que se hizo fue crear la estructura de la base de datos en el sistema principal DBSIANX, con todas las relaciones respectivas.

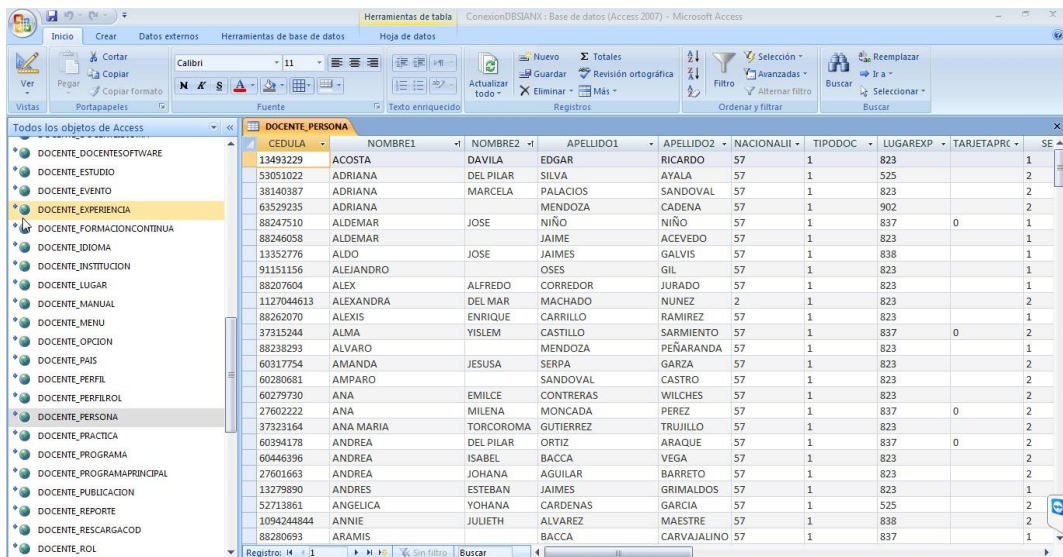
Figura 29. Interfaz Base de Datos DB2.



Fuente. Autor del proyecto

Se pasa del sistema DBPEDAG que es el sistema de pruebas al principal. Para administrar la información de la base de datos se puede hacer en Access vinculando las tablas de DB2 al programa.

Figura 30. Vinculación de tablas en Access.



Fuente. Autor del proyecto

Después ya pudiendo administrar las tablas en el sistema principal se suben los archivos del sistema desarrollado a la carpeta raíz del servidor y ya se tiene acceso vía web. A la fecha se tiene registrado 323 docentes de la universidad.

Figura 31. Registro de Docentes.

CODDOCEN	CEDULA	ESTADO	PASSWORD
00001	5415297	A	53362d5ea52a28e1a960323ea1!
00002	88155275	A	7598095b3084bcc39aae21ea54e
00003	109424082	A	53362d5ea52a28e1a960323ea1!
00004	88248998	A	53362d5ea52a28e1a960323ea1!
00005	27601663	A	53362d5ea52a28e1a960323ea1!
00006	37246693	A	53362d5ea52a28e1a960323ea1!
00007	1013578122	A	53362d5ea52a28e1a960323ea1!
00008	60330688	A	53362d5ea52a28e1a960323ea1!
00009	88222873	A	53362d5ea52a28e1a960323ea1!
00010	79481534	A	367b839f94d4e4e9cf8456e6a9
00011	19411813	A	9b5f94318ff086fe7b0e1c4afb
00012	91533603	A	53362d5ea52a28e1a960323ea1!
00013	1094244844	A	53362d5ea52a28e1a960323ea1!
00014	88198580	A	53362d5ea52a28e1a960323ea1!
00015	1094241595	A	53362d5ea52a28e1a960323ea1!
00016	52706927	A	53362d5ea52a28e1a960323ea1!
00017	88213579	A	53362d5ea52a28e1a960323ea1!
00018	27592816	A	53362d5ea52a28e1a960323ea1!
00019	88279563	A	53362d5ea52a28e1a960323ea1!
00020	60346905	A	53362d5ea52a28e1a960323ea1!
00021	13470050	A	53362d5ea52a28e1a960323ea1!
00022	88280693	A	53362d5ea52a28e1a960323ea1!
00023	60446396	A	53362d5ea52a28e1a960323ea1!
00024	60344496	A	53362d5ea52a28e1a960323ea1!
00025	88213245	A	53362d5ea52a28e1a960323ea1!

Fuente. Autor del proyecto

Se han registrado más de 500 estudios.

Figura 32. Registro de Estudios

ID	CODDOCEN	TIPOFORMA	TITULO OBTENIDO	INSTITUCION	LUGAR	SEMESTRES	FECHA INICIO	FECHA TITULO
1	00162	5	LICENCIADO EN BIOLOGIA	284	1103		01/01/1900	14/03/1989
2	00162	6	HISTORIADOR	284	1103		01/01/1900	19/02/2002
3	00162	7	ESPECIALISTA EN FILOSOFIA	242	1103		01/01/1900	02/12/1994
4	00162	9	MAGISTER EN HISTORIA	295	1103		01/01/1900	30/09/2005
5	00162	91	DOCTORADO EN HUMANIDADES	311	1103	1	01/01/1900	
6	00102	6	ABOGADO	286	1103		01/01/1900	16/06/2006
7	00102	6	CONTADOR PUBLICO	281	1103		01/01/1900	14/06/2002
8	00102	9	MAGISTER EN DERECHO ADMINISTRATIVO	286	1103		01/01/1900	08/06/2011
9	00256	9	MAGISTER EN DERECHO ADMINISTRATIVO	280	1103	4	01/01/1900	
10	00256	7	ESPECIALISTA EN DERECHO PUBLICO	280	1103		01/01/1900	01/11/2011
11	00256	6	ABOGADO	286	1103		01/01/1900	09/07/2010
12	00109	7	ESPECIALISTA EN DERECHO LABORAL Y REL	280	1103		01/01/1900	06/11/2008
13	00109	6	ABOGADO	286	1103		01/01/1900	18/04/1997
14	00121	6	ABOGADO	281	1103		01/01/1900	14/12/1994
15	00121	5	LICENCIADO EN EDUCACION ESPECIALIDAD	281	1103		01/01/1900	21/02/1997
16	00121	7	ESPECIALISTA EN DERECHO ADMINISTRATI	300	1103		01/01/1900	01/04/1997
17	00121	7	ESPECIALISTA EN DOCENCIA UNIVERSITARI	300	1103		01/01/1900	25/02/2000
18	00121	7	ESPECIALISTA EN DERECHO ADUANERO	280	1103		01/01/1900	04/06/2009
19	00244	7	ESPECIALISTA EN DOCENCIA UNIVERSITARI	286	1103		01/01/1900	19/12/1991
20	00244	6	ABOGADO	286	1103		01/01/1900	12/12/1997
21	00244	6	CONTADOR PUBLICO	281	1103		01/01/1900	29/06/1984
22	00002	5	LICENCIADO EN FILOSOFIA E HISTORIA	300	1103		01/01/1900	16/04/1993
23	00002	7	ESPECIALISTA EN EDUCACION PARA LA DEF	262	1103		01/01/1900	07/05/1998
24	00002	7	ESPECIALISTA EN GESTION DE PROYECTOS	262	1103		01/01/1900	26/09/2008
25	00004	6	ABOGADO	284	1103		01/01/1900	03/10/2006

Fuente. Autor del proyecto

Además se han registrado muchas formaciones continuas.

Figura 33. Registro de Formaciones Continuas.

ID	CODDOCEN	TIPOCONTIP	INSTITUCIOI	LUGAR	INTENSIDAE	FECHAINICI	FECHATITUL
1	00102	3	223	1103	140	01/01/1900	01/02/2009
2	00004	3	223	1103	140	01/01/1900	20/07/2010
3	00004	3	223	1103	112	01/01/1900	01/09/2010
4	00215	3	281	1103	100	01/01/1900	07/09/2010
5	00165	3	286	1103	144	01/01/1900	15/08/2009
6	00165	2	83	1103	700	01/01/1900	17/08/2010
7	00165	2	223	1103	120	01/01/1900	21/09/2009
8	#N/A	3	255	1103	80	01/01/1900	01/08/2011
9	00099	1	3	1103	90	01/01/1900	08/03/2011
10	00133	2	242	1103	90	01/01/1900	01/12/2011
11	00140	2	223	1103	370	01/01/1900	10/08/2009
12	00248	1	286	1103	130	01/01/1900	12/12/2009
13	00248	3	286	1103	144	01/01/1900	06/02/2010
14	00248	3	286	1103	144	01/01/1900	15/08/2009
15	00067	3	281	1103	100	01/01/1900	15/12/2009
16	00067	1	242	1103	90	01/01/1900	01/12/2011
17	00070	3	281	1103	120	01/01/1900	23/06/2009
18	00076	1	286	1103	130	01/01/1900	18/07/2009
19	00076	3	2	1103	318	01/01/1900	21/10/2011
20	00244	3	281	1103	100	01/01/1900	15/12/2009
21	00037	3	281	1103	270	01/01/1900	02/12/2010
22	00181	1	223	1103	140	01/01/1900	27/07/2009
23	00204	3	242	1103	90	01/01/1900	01/12/2011
24	00222	3	262	1103	100	01/01/1900	07/10/2009
25	00197	1	223	1103	140	01/01/1900	02/12/2010

Fuente. Autor del proyecto

Y los usuarios identificados en el sistema hasta la fecha son 1339.

Figura 34. Usuarios identificados en el sistema.

ID	USUARIO	PROCESO	IP	FECHA
0	00115	OK		03/02/2014 02:59:34 p.m.
1	ADMIN	OK		03/02/2014 03:00:48 p.m.
2	ADMIN	OK	192.168.3.1	03/02/2014 03:01:33 p.m.
3	ADMIN	OK		03/02/2014 03:02:17 p.m.
4	ADMIN	OK	192.168.3.1	03/02/2014 03:02:57 p.m.
5	KMORA	OK	192.168.3.1	03/02/2014 03:03:19 p.m.
6	INGRIDH	OK	192.168.3.1	03/02/2014 03:10:27 p.m.
7	00197	OK	192.168.3.1	03/02/2014 03:15:50 p.m.
8	NGARCIA	OK	192.168.3.1	03/02/2014 03:30:19 p.m.
9	00025	OK	186.144.64.19	03/02/2014 03:33:21 p.m.
10	KMORA	OK	192.168.3.1	03/02/2014 03:51:45 p.m.
11	KMORA	OK	192.168.3.1	03/02/2014 04:02:21 p.m.
12	SJAIMES	OK	192.168.3.1	03/02/2014 04:10:57 p.m.
13	OROZO	OK	192.168.3.1	03/02/2014 04:14:50 p.m.
14	RPALMA	OK	192.168.3.1	03/02/2014 04:20:31 p.m.
15	LGARCIA	OK	192.168.3.1	03/02/2014 04:49:22 p.m.
16	YPEINADO	OK	192.168.3.1	03/02/2014 04:49:38 p.m.
17	DIANAM	OK	192.168.3.1	03/02/2014 04:52:40 p.m.
18	LSANTANA	OK	192.168.3.1	03/02/2014 05:09:57 p.m.
19	KMORA	OK	192.168.3.1	03/02/2014 05:26:52 p.m.
20	SVARGAS	OK	192.168.3.1	03/02/2014 05:36:47 p.m.
21	KMORA	OK	192.168.3.1	03/02/2014 05:40:26 p.m.
22	NGARCIA	OK	192.168.3.1	03/02/2014 05:44:38 p.m.
23	sjmora	ERROR	192.168.3.1	03/02/2014 05:47:11 p.m.
24	SJAIMES	OK	192.168.3.1	03/02/2014 05:47:28 p.m.

Fuente. Autor del proyecto

El link para acceder al sistema está disponible en la página web de la universidad.

Figura 35. Interfaz Ingreso al Sistema.



Fuente. Autor del proyecto

Figura 36. Reporte Control Hoja de Vida

181.48.123.164/mvc/

17 febrero 2014 16:38:20 PM

PROGRAMA: INGENIERIA DE SISTEMAS Versión 1.0

UNIVERSIDAD SIMÓN BOLÍVAR
Extensión Cúcuta

CONTROL DE ACTUALIZACIÓN HOJA DE VIDA

DOCENTES CON ASIGNACION DE HORAS

CÓDIGO	NOMBRE DEL DOCENTE	ESTUD	EXPER	CONTI	EVENT	DISTIN	PUBLI	OTROS	FECHA
00032	BAUTISTA VEGA NAHID ANTUAN	-	-	-	-	-	-	-	SIN REGISTRAR
00090	GALLARDO PEREZ OSCAR ALBERTO	-	✓	✓	✓	✓	✓	✓	31/01/2014
00115	HERRERA CACERES MATIAS	✓	✓	✓	✓	✓	-	-	13/02/2014
00128	JAIMES PALACIOS DANIEL	-	-	-	-	-	-	-	SIN REGISTRAR
00130	LAGUADO AGUILAR OSWALDO ENRIQUE	-	✓	-	-	-	-	-	29/01/2014
00181	PEREZ VILLAMIZAR MIGUEL ANGEL	-	✓	✓	✓	✓	✓	-	30/01/2014
00197	RESTREPO CELIS OSCAR DANIEL	-	✓	✓	-	-	-	-	17/02/2014
00203	RODRIGUEZ FERNANDEZ JOHEL ENRIQUE	-	✓	-	-	✓	-	-	31/01/2014
00204	RODRIGUEZ IBAÑEZ RAUL EDUARDO	-	-	-	-	-	-	-	SIN REGISTRAR
00213	SAENZ PEÑA FRANK HERNANDO	-	✓	-	-	-	-	-	31/01/2014
00276	VARGAS ANGULO SANDRA	-	-	-	-	-	-	-	SIN REGISTRAR
00285	VILLAMIZAR NUÑEZ CESAR ANTONIO	-	-	-	-	-	-	-	SIN REGISTRAR
00299	CABRALES PINEDA JOVER ALONSO	-	-	-	-	-	-	-	SIN REGISTRAR
00314	CADENA CHAPARRO FLOR DINORA	-	-	-	-	-	-	-	SIN REGISTRAR

Fuente. Autor del proyecto

Figura 37. Reporte Hoja de Vida



14 febrero 2014 21:32:10 PM
Versión 1.0

FORMATO DE HOJA DE VIDA

MATIAS HERRERA CACERES
 Nacionalidad: COLOMBIA
 cedula ciudadanía: 88027979 de CUCUTA
 Edad: 31 Años
 Sexo: MASCULINO
 Estado civil: CASADO(A)
 Dirección: AV. 0 No.13-36
 Teléfono: 5944961
 e-mail: mahc83@gmail.com
 Fecha Nacimiento: 03 noviembre 1983

ESTUDIOS REALIZADOS	
Actualmente OTRO	MAESTRIA SOFTW LIBRE UNAB-UOC UNIVERSIDAD AUTONOMA DE BUCARAMANGA-UNAB-
2011 OTRO	ESPECIALIZACIÓN EN GESTION DE PROYECTOS INFORMATICOS UNIVERSIDAD DE PAMPLONA
2007 OTRO	INGENIERO DE SISTEMAS UNIVERSIDAD FRANCISCO DE PAULA SANTANDER
Actualmente OTRO	DOCTORADO EN EDUCACION UNIVERSIDAD PEDAGOGICA EXPERIMENTAL LIBERTADOR

Fuente. Autor del proyecto

Figura 38. Reporte Relación Docente Código



17 febrero 2014 16:45:39 PM
Versión 1.0

RELACIÓN DE CÓDIGOS DE DOCENTES

CÓDIGO	CEDULA	1er APELLIDO	2do APELLIDO	1er NOMBRE	2do NOMBRE	E-MAIL
00032	88215027	BAUTISTA	VEGA	NAHID	ANTUAN	nahid@colombia.com
00299	88196961	CABRALES	PINEDA	JOVER	ALONSO	JCABRALE@HOTMAIL.COM
00314	60362363	CADENA	CHAPARRO	FLOR	DINORA	dinorac22@gmail.com
00090	19453653	GALLARDO	PEREZ	OSCAR	ALBERTO	oagallardo@gmail.com
00115	88027979	HERRERA	CACERES	MATIAS		mahc83@gmail.com
00128	13494061	JAIMES	PALACIOS	DANIEL		djaimesp@yahoo.es
00130	88214446	LAGUADO	AGUILAR	OSWALDO	ENRIQUE	oswaldolag@hotmail.com
00181	88233663	PEREZ	VILLAMIZAR	MIGUEL	ANGEL	mapv24n@hotmail.com
00197	88212997	RESTREPO	CELIS	OSCAR	DANIEL	osdali@hotmail.com
00203	88206787	RODRIGUEZ	FERNANDEZ	JOHEL	ENRIQUE	joheler@hotmail.com
00204	88199787	RODRIGUEZ	IBAÑEZ	RAUL	EDUARDO	rauldj77@hotmail.com
00213	13497960	SAENZ	PEÑA	FRANK	HERNANDO	frasa_p@hotmail.com
00276	60393890	VARGAS	ANGULO	SANDRA		
00285	13275297	VILLAMIZAR	NUÑEZ	CESAR	ANTONIO	cvillamn@hotmail.com

Fuente. Autor del proyecto

Figura 39. Reporte Formación Académica.



17 febrero 2014 16:42:00 PM
Versión 1.0

DOCENTE CON FORMACIÓN ACADÉMICA

PROGRAMA: INGENIERIA DE SISTEMAS

DOCENTES CON ASIGNACION DE HORAS

CÓDIGO	NOMBRE DEL DOCENTE	NIVEL DE ESTUDIO
00032	BAUTISTA VEGA NAHID ANTUAN	
00090	GALLARDO PEREZ OSCAR ALBERTO	MAGISTER
00115	HERRERA CACERES MATIAS	DOCTORADO
00128	JAIMES PALACIOS DANIEL	ESPECIALIZACION
00130	LAGUADO AGUILAR OSWALDO ENRIQUE	MAGISTER
00181	PEREZ VILLAMIZAR MIGUEL ANGEL	MAGISTER
00197	RESTREPO CELIS OSCAR DANIEL	ESPECIALIZACION
00203	RODRIGUEZ FERNANDEZ JOHEL ENRIQUE	MAGISTER
00204	RODRIGUEZ IBAÑEZ RAUL EDUARDO	ESPECIALIZACION
00213	SAENZ PEÑA FRANK HERNANDO	ESPECIALIZACION
00276	VARGAS ANGULO SANDRA	MAGISTER
00285	VILLAMIZAR NUÑEZ CESAR ANTONIO	
00299	CABRALES PINEDA JOVER ALONSO	
00314	CADENA CHAPARRO FLOR DINORA	

Fuente. Autor del proyecto

5. CONCLUSIONES

Del presente desarrollo se desprende lo importante que es para la Universidad Simón Bolívar Seccional Cúcuta que sus docentes conozcan el término de la formación profesoral y que al mismo tiempo mantengan esta información actualizada gracias al nuevo sistema implementado, manteniendo una integridad de la información de los docentes para ser utilizada en los momentos requeridos por la Universidad.

También se obtuvo la ganancia de que el análisis de la información de los docentes será más rápido, fácil de calcular y así poder analizar los resultados los cuales son por medio de reportes que hace mejor el proceso de considerar los datos.

Además se descubre lo mucho que mejora la Universidad con la implementación del sistema ya que va desarrollando la parte tecnológica dentro de la institución aplicando soluciones novedosas a las necesidades presentadas.

Por último en la parte del desarrollo del sistema se demuestra lo factible que se hace trabajar con la implementación del MVC (Modelo-Vista-Controlador). Utilizar para la parte del Modelo Doctrine, para la parte del Controlador se utilizó PHP y para la parte de Vista se utilizó Bootstrap. Con esta combinación de tecnologías se logró que el desarrollo fuera más rápido dejando al mismo tiempo una base estructurada para desarrollos futuros.

6. RECOMENDACIONES

A usuarios del sistema, considerando que la aplicación es on-line se debe contar con acceso a Internet con una velocidad considerable para lograr un óptimo desempeño del sistema.

Los usuarios deben conocer el manejo del sistema antes de usarlo, debido a esto se diseñó un manual de usuario el cual permitirá una mejor interacción con la aplicación y el uso será más sencillo si se logra seguir los pasos del manual.

Para la Universidad Francisco de Paula Santander Ocaña se recomienda seguir este tipo de proyectos de grado vinculados a otras universidades. Además se recomienda que se implemente el proyecto dentro de la universidad UFPSO para obtener el beneficio de la formación profesoral de los docentes.

Para la Universidad Simón Bolívar Cúcuta se recomienda que sus docentes lleven el proceso de formación profesoral para ver las bondades del sistema. Además se recomienda que el departamento de sistemas de la Universidad comience a utilizar la estructura de desarrollo realizada para que los procesos sean más ágiles.

BIBLIOGRAFIA

Angel Cobo, Patricia Gomez, Rocio Royal. 2005. *PHP y MYSQL, Tecnologias para el Desarrollo de aplicaciones web*. Diaz de Santos : s.n., 2005.

Bahit, Eugenia. 2012. *POO y MVC en PHP*. España : s.n., 2012.

Group, Innovation. 2012. Innovation Group. [En línea] 2012. <http://innovationgroup.com/PORTALWEB.aspx>.

Heilman. 2006. *Beginning JavaScript with DOM Scripting ans Ajax From Novice to Professional*. New York : s.n., 2006.

IBM. 2012. IBM Developer Works. [En línea] 06 de 08 de 2012. <http://www.ibm.com/developerworks/ssa/data/tutorials/db2-cert7303/section2.html>.

Jaramillo, Eduardo. 2010. Direccional de Innovaciones Academicas. [En línea] 2010. <http://www.virtual.unal.edu.co/cursos/sedes/manizales/4060024/Lecciones/Capitulo%20I/p roblemas.htm>.

Libre. Ley Software. [En línea] <http://glosario.over-blog.es/article-leyes-sobre-software-libre-en-colombia-59255157.html>.

Martinez. 2007. *Nuevas Tecnologias para Nuevas Bibliotecas*. España : AlfaGram, 2007.

Martinez, Alfonso. 2005. *Introduccion a la Ingenieria de Software*. s.l. : Delta, 2005.

Mateu. 2006. *Desarrollo de Aplicaciones Web*. Barcelona : Eureka Media, 2006.

Molinete. 2012. Buenas Tareas. [En línea] 12 de 07 de 2012. <http://www.buenastareas.com/ensayos/Sistema-De-Informacion-Para-Registro-y/4703061.html>.

Sarmiento, Carlos. 2013. *Adrenalina Posicionamiento Web*. 02 : 22, 2013.

Software, Procesos de. 2010. WikiSpaces. [En línea] 12 de 05 de 2010. <http://procesosdesoftware.wikispaces.com/METODOLOGIAS+PARA+DESARROLLO+D E+SOFTWARE>.

Torrado, Alonso. 2012. Wikipedia. [En línea] 2012. http://es.wikipedia.org/wiki/Hojas_de_estilo_en_cascada#CSS3.

Villamarin, Antonio. Antonio SW. [En línea] <http://ant.onio.org/2013/04/10/modelo-vista-controlador-adaptado-a-la-web.html>.

REFERENCIAS DOCUMENTALES ELECTRÓNICAS

IBM. (06 de 08 de 2012). IBM. Obtenido de Developer Works: <http://www.ibm.com/developerworks/ssa/data/tutorials/db2-cert7303/section2.html>

Group. (2012). Obtenido de <http://innovactiongroup.com/PORTALWEB.aspx>

Jaramillo, E. (2010). Direccional de Innovaciones Academicas. Obtenido de <http://www.virtual.unal.edu.co/cursos/sedes/manizales/4060024/Lecciones/Capitulo%20I/p/roblemas.htm>

Molinete. (12 de 07 de 2012). Buenas Tareas. Obtenido de <http://www.buenastareas.com/ensayos/Sistema-De-Informacion-Para-Registro-y/4703061.html>

Procesos de Software. (12 de 05 de 2010). Obtenido de WikiSpaces: <http://procesosdesoftware.wikispaces.com/METODOLOGIAS+PARA+DESARROLLO+D+E+SOFTWARE>

Torrado, A. (2012). Wikipedia. Obtenido de http://es.wikipedia.org/wiki/Hojas_de_estilo_en_cascada#CSS3

Villamarin, A. (10 de 04 de 2013). Antonio. Obtenido de <http://ant.onio.org/2013/04/10/modelo-vista-controlador-adaptado-a-la-web.html>

ANEXOS

Anexo A. Formato de encuesta

UNIVERSIDAD SIMON BOLIVAR SECCIONAL CUCUTA NORTE DE SANTANDER 2013

Encuesta dirigida a docentes de la Universidad Simón Bolívar Seccional Cúcuta.

OBJETIVO: Recolectar y obtener información válida y confiable que permita conocer si los docentes conocen el estado actual de su información profesoral y los lugares donde esta se deposita.

1. ¿Tiene usted conocimiento de lo que significa la formación profesoral de un docente?

_____ SI _____ NO _____ NO RESPONDE

Porqué? _____

2. ¿Debería hacerse copias de seguridad a la información de la institución?

_____ SI _____ NO _____ NO RESPONDE

Porqué? _____

3. ¿Hay disponibilidad por parte de la sección administrativa cuando usted solicita _____ información referente a su puesto y formación?

_____ SI _____ NO _____ NO RESPONDE

Porqué? _____

4. ¿Aceptaría que su información profesoral la pueda consultar cualquier persona del _____ exterior de la institución?

_____ SI _____ NO _____ NO RESPONDE

Porqué ? _____

5. ¿Cree usted recomendable que su información profesoral este en un portal web donde _____ usted diligenciaría su información y la actualizaría cada vez que fuera necesario?

_____ SI _____ NO _____ NO RESPONDE

Porqué? _____

6. ¿Usted realizaría el procedimiento de actualizar sus datos cada vez que sea necesario en _____ el nuevo portal web?

_____ SI _____ NO _____ NO RESPONDE

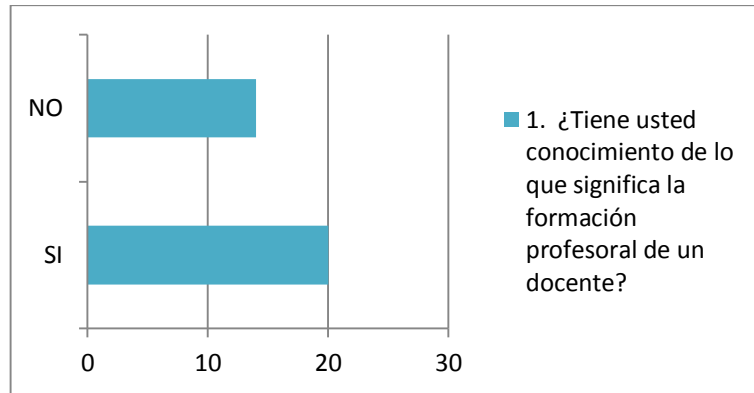
Porqué? _____

7. ¿Cree usted que su información profesoral estaría segura en un portal web?

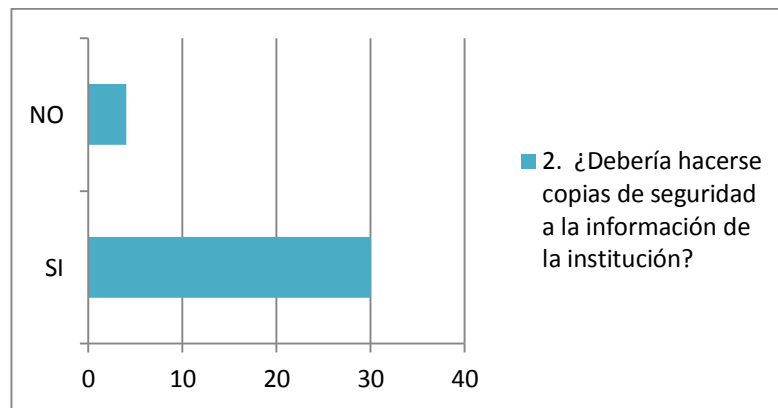
_____ SI _____ NO _____ NO RESPONDE

Porqué? _____

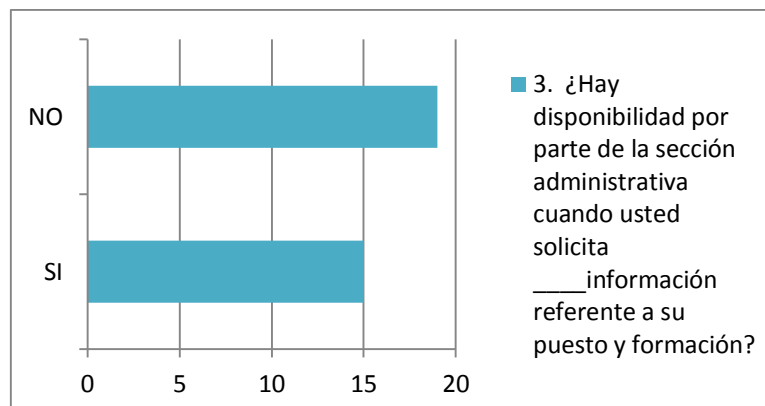
Anexo B. Resultados de las Encuestas



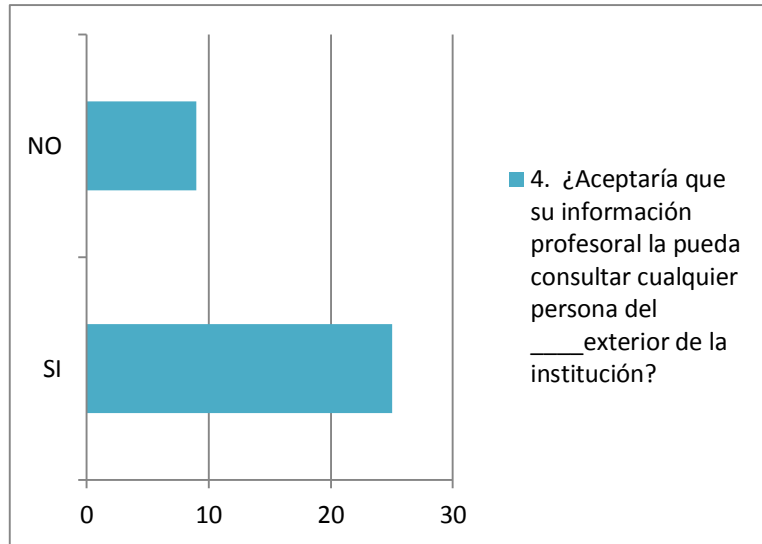
Se observa que aunque algunos saben de qué se habla con la formación profesoral otros docentes no tienen idea y esto es lo que toca corregir.



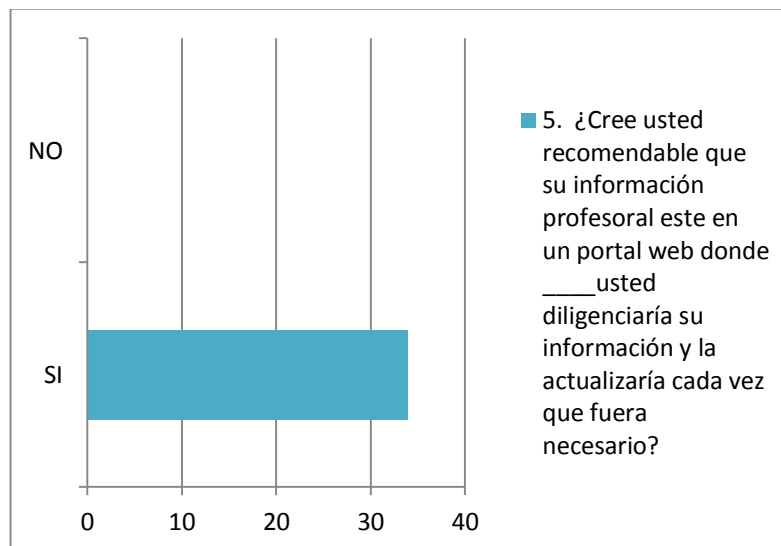
Todos los docentes están atentos de lo importante que es la información y que si debe ser cuidarse y realizar copias de seguridad.



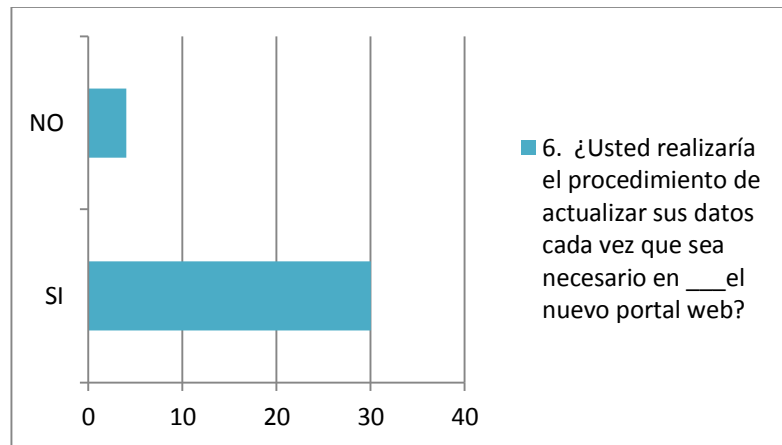
Se ve que la parte administrativa de la universidad tiene la información pero no la tiene en un buen formato o no la tiene disponible en el momento que se necesita. Esto se corregirá ya que se va a tener la información disponible en cualquier momento.



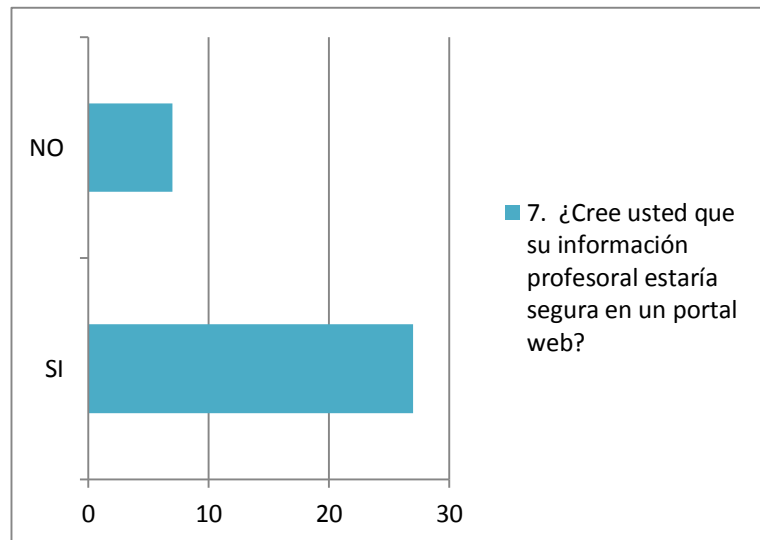
Se observa que los docentes, por lo menos la mayoría están de acuerdo en que su información esté disponible en la web, para que los alumnos que los quieran conocer y saber más de sus estudios y experiencias puedan consultarlo.



Definitivamente los docentes están de acuerdo en poder administrar su información profesoral, experiencias, eventos, publicaciones y demás vía web.



Los docentes cumplirán con diligenciar los procedimientos para gestionar la información que estará en el sistema disponible.

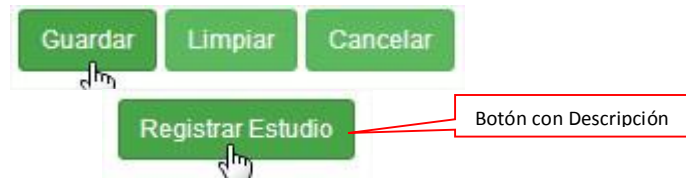


La seguridad es muy importante en cualquier sistema vía web, garantizar la integridad de los datos y que nadie que no esté autorizado pueda modificarlos. Por esta razón el sistema futuro será con una seguridad calificada y validada por las norma.

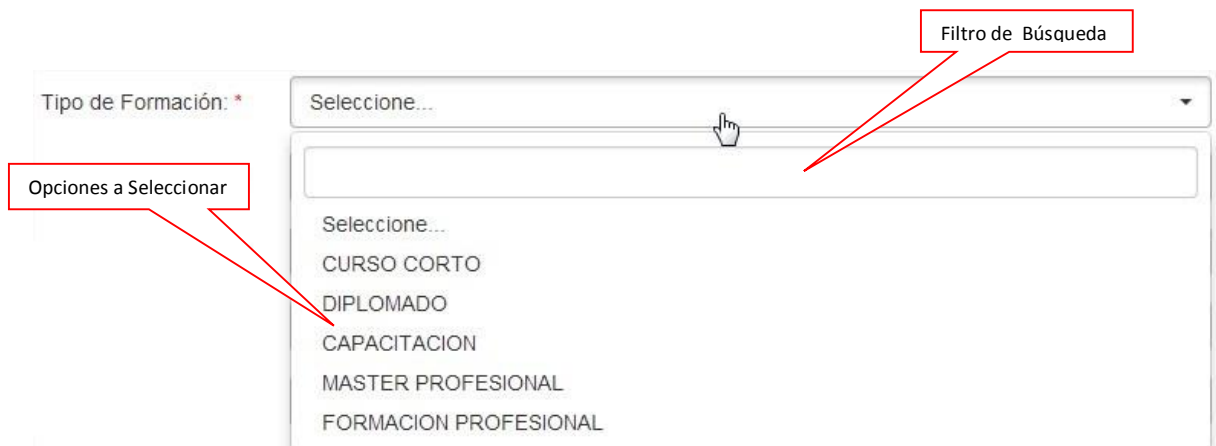
Anexo C. Manual de usuario

Descripción de los Componentes a Utilizar.

Botones: Los Botones del Portal permiten la administración de la información, guardar, consultar, editar y eliminar entre otras funciones. Los botones son de color verde y en su centro tiene la descripción de lo que hace cada uno.



Combos Desplegables: Los combos son utilizados para mostrar lista de datos; por lo que en casos de que la listas sean largas tiene un filtro de texto donde con poner las letras referentes a lo buscado se mostrara en el combo.



Tablas: Las tablas son utilizadas para mostrar la información de un registro ya sea un estudio, una experiencia, una distinciones y demás. Se muestran solo algunos campos del registro, ya que la información completa estará en el detalle de ese registro. Además se verá el estado del registro y las acciones que se pueden hacer con ese registro.

Formación	Título	Institución	Lugar	Titulado	Estado	Acciones
PRIMARIO	PRIMARIA	CENTRO DE EDUCACION MILITAR - CEMIL	AMAGÁ	SI	ACTIVO	
TECNICO NIVEL MEDIO	BACHILLER TECNICO	CENTRO EDUCACIONAL DE COMPUTOS Y SISTEMAS-CEDESISTEMAS	OCAÑA	SI	INACTIVO	
UNIVERSITARIO/PREGRADO	INGENIERO DE SISTEMAS	COLEGIO DE ESTUDIOS SUPERIORES DE ADMINISTRACION-CESA	OCAÑA	SI	INACTIVO	

Registro

Estado

Acciones

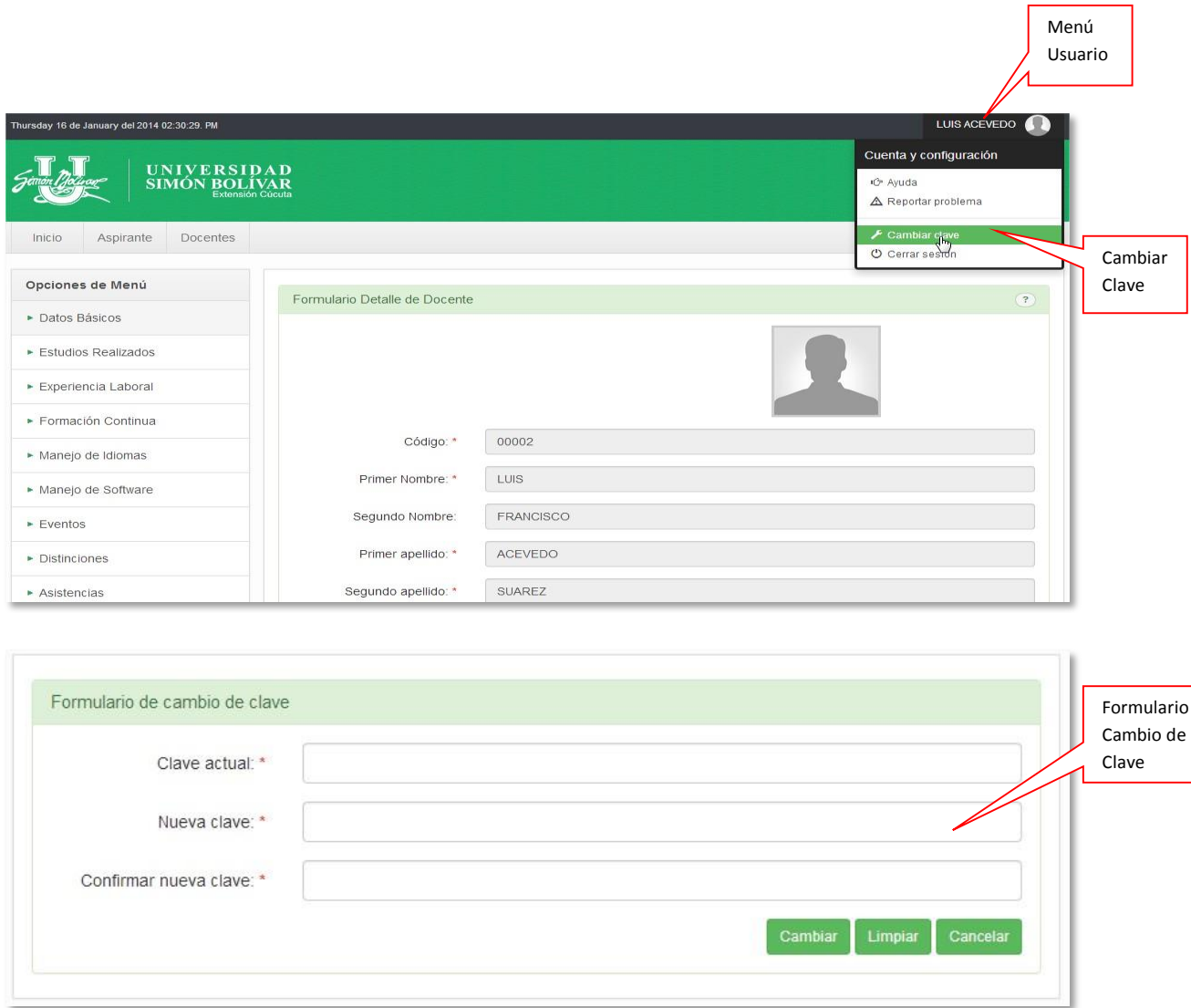
detallarest

Ingreso al Sistema.

En el **Portal de Administración de Formación Académica** lo primero que toca identificar es la dirección **IP** de ingreso.

La dirección para entra al portal es **181.48.123.164/mvc** cargándose una página de inicio como la siguiente:

Una de las recomendaciones es realizar el procedimiento de cambio de clave el cual se realiza como se explica a continuación.



Aquí se digitara la clave actual y la nueva clave con su confirmación. De este modo el docente habrá cambiado su contraseña y su sesión estará segura.

A continuación se mostrara la interfaz principal del sistema una vez este autenticado. Se mostraran los componentes que se encuentran en la pantalla como la barra horizontal principal, la barra secundaria lateral y la zona dinámica que es la que va a cambiar cada vez que demos clic en alguna opción.

Barra de Navegación Principal

Barra de Navegación Lateral

Tuesday 21 de January del 2014 05:10:04. PM

YAN ANGARITA 



UNIVERSIDAD
SIMÓN BOLÍVAR
Extensión Cúcuta

Inicio Aspirante Docentes

Zona Dinámica

Opciones de Menú

- ▶ Datos Básicos
- ▶ Estudios Realizados
- ▶ Experiencia Laboral
- ▶ Formación Continua
- ▶ Manejo de Idiomas
- ▶ Manejo de Software
- ▶ Eventos
- ▶ Asistencias
- ▶ Distinciones
- ▶ Investigaciones
- ▶ Publicaciones
- ▶ Generar HV

Datos Básicos

Formulario Detalle de Docente



Información Básica del Docente

Código: *	0500
Primer Nombre: *	YAN
Segundo Nombre:	CARLO
Primer apellido: *	ANGARITA
Segundo apellido: *	SANGUINO
Nacionalidad: *	COLOMBIA
Tipo de documento: *	CEDULA CIUDADANIA
Número de documento: *	12345
Lugar de expedición: *	CUCUTA
Tarjeta profesional:	5412587412
Sexo: *	MASCULINO
Estado civil:	CASADO
Número de hijos:	1
Datos de nacimiento	
Pais: *	COLOMBIA
Ciudad: *	OCAÑA
Fecha de nacimiento: *	22/04/1989
Datos de contacto	
Teléfono 1: *	3166760041
Teléfono 2:	5690156
Dirección de residencia: *	KRA 10 N° 4-32 TEJARITO
Correo institucional:	ycangaritas@ufpso.edu.co
Correo personal: *	yancario1206@gmail.com
Twitter:	@yancarlos
Facebook:	ejemplo: /micuenta
Página web:	http://www.google.com
Información de la cuenta	
Estado:	Activado

Información Básica del Docente

Editar datos Cancelar

En la parte inferior del formulario están los datos de la información básica que se puede modificar por parte del docente y para guardarlos se le da la opción **Actualizar**.

The image shows a web form with two main sections: "Datos de contacto" and "Información de la cuenta".

Datos de contacto

- Teléfono 1: * 5871351
- Teléfono 2: 3138328900
- Dirección de residencia: * CLL 19N N° 2-154 PRADOS NORTE
- Correo institucional:
- Correo personal: * LUISFRA55@HOTMAIL.COM
- Twitter: ejemplo: @micuenta
- Facebook: ejemplo: /micuenta
- Página web: ejemplo: http://www.miweb.com

Información de la cuenta

- Estado: Activado

At the bottom right, there are three buttons: "Actualizar", "Limpiar", and "Cancelar". A red callout box points to the "Actualizar" button with the text "Actualizar Cambios". Another red callout box on the left points to the "Correo personal" field with the text "Solo los datos que están en la parte inferior son los que se pueden editar".



Esta barra de menú le permite al docente navegar a través de todas las opciones y como cada una lo indica con su nombre, **Estudios Realizados** es para administrar los estudios realizados del docente, **Experiencia Laboral** es para administrar las experiencias y así es con cada una de las opciones.

En cada una de las opciones el docente podrá registrar, editar, eliminar y consultar los datos los cuales deben pasar primero por recursos humanos por un proceso de validación con las pruebas de que si es veras la información que se digita y queden activados al sistema.

Administrar Formación Académica.

Cuando se da clic en la opción **Estudios Realizados** aparece la pantalla con el listado de los estudios registrados (Si no hay estudios no saldrá la lista). Para registrar nuevos estudios se da clic en **Registrar Estudio**.



Se mostrara el siguiente formulario.

Formulario Registro de Estudios

Tipo Formacion: * Seleccione...

Titulo Obtenido: *

Institucion: * Seleccione...

Lugar: * Seleccione...

Semestres: Aplica solo para estudios superiores

Fecha Inicio: * dd/mm/aaaa

Fecha Terminacion: dd/mm/aaaa

Fecha Titulo: dd/mm/aaaa

Estado del Registro: Desactivado

Guardar Limpiar Cancelar

Botón Guardar Estudio

- El campo **Tipo Formación***da la opción Estudio como tipo de estudio entre las cuales aparecen primaria, secundaria, superiores, especialización, maestrías y demás.
- El campo **Título Obtenido***es para el título que se obtuvo en el tipo de formación seleccionado.
- El campo **Institución***se busca la institución donde se realizó el estudio, si no esta se selecciona la opción OTRO.
- El campo **Lugar***se busca el lugar donde se realizó el estudio, si no esta se selecciona la opción OTRO.
- El campo **Semestre** es para el numero de semestres que duro el estudio, solo aplica para estudios superiores.
- El campo **Fecha Inicio***es la fecha de inicio del estudio.
- El campo **Fecha Terminación** es la fecha de terminación del estudio, si es un estudio que aún sigue desarrollando no se asigna fecha de terminación.
- El campo **Fecha Titulo** es en caso de haber terminado el estudio asignar la fecha de título.
- El campo **Estado del Registro** siempre comienza desactivado hasta que Recursos Humanos lo active.

Este formulario lo deberá diligenciar el número de veces referente a los estudios que haya realizado y así el sistema le mostrara el listado de los que ya haya registrado y su estado.

Estudios Encontrados (3) ?						
Formación	Título	Institución	Lugar	Titulado	Estado	Acciones
PRIMARIO	PRIMARIA	CENTRO DE EDUCACION MILITAR - CEMIL	AMAGÁ	SI	ACTIVO	
TECNICO NIVEL MEDIO	BACHILLER TECNICO	CENTRO EDUCACIONAL DE COMPUTOS Y SISTEMAS-CEDESISTEMAS	OCAÑA	SI	ACTIVO	
UNIVERSITARIO/PREGRADO	INGENIERO DE SISTEMAS	COLEGIO DE ESTUDIOS SUPERIORES DE ADMINISTRACION-CESA	OCAÑA	SI	ACTIVO	

[Registrar Estudio](#)

En donde si le da clic al icono de la derecha de una lupa le abrirá el detalle del estudio escogido donde se podrá detallar la información completa y editarla en caso de que aun este desactivado.

Administrar Experiencia Laboral.

En la opción del menú lateral **Experiencia Laboral** es lo referente de la experiencia del docente en trabajos pasados en donde podrá registrar, editar, consultar y eliminar los datos referentes. El formulario que se debe diligenciar para registrar una nueva experiencia es el siguiente:

Formulario de Registro de Experiencia Laboral

Formulario Registro de Experiencia Laboral ?	
Nombre de Empresa: *	<input type="text"/>
Cargo Desempeñado: *	Seleccione... ▼
Telefono: *	<input type="text"/>
Direccion: *	<input type="text"/>
Lugar: *	Seleccione... ▼
Fecha Inicio: *	dd/mm/aaaa
Fecha Fin: *	dd/mm/aaaa
Funciones: *	<input style="height: 80px;" type="text"/>
Estado del Registro:	Inactiva ▼
<input type="button" value="Guardar"/> <input type="button" value="Limpiar"/> <input type="button" value="Cancelar"/>	

- El campo **Nombre de Empresa*** es para el nombre de la empresa donde se llevó a cabo el trabajo a referenciar.
- El campo **Cargo Desempeñado*** se selecciona el cargo el cual ejerció en el trabajo, si el cargo no se encuentra se selecciona la opción OTRO.
- El campo **Teléfono*** es el teléfono de la empresa.
- El campo **Dirección*** es la dirección de la empresa.
- El campo **Lugar*** es el lugar donde se ubica la empresa, si no encuentra el lugar selecciona la opción OTRO
- El campo **Fecha Inicio*** es la fecha de inicio de la experiencia laboral.
- El campo **Fecha Fin** es la fecha de terminación de la experiencia, si es un trabajo que aún sigue desarrollando no se asigna fecha de fin.
- El campo **Funciones*** es para registrar las funciones que desarrollo en docente en el trabajo.
- El campo **Estado del Registro** siempre comienza desactivado hasta que Recursos Humanos lo active.

Administrar formación Continua.

En la opción del menú lateral **Formación Continua** es lo referente a la formación continua que ha desarrollado el docente y aquí se podrá administrar esa información; cursos, capacitaciones, diplomados y demás. El formulario que se debe diligenciar para registrar una nueva experiencia es el siguiente:

Formulario Registro de Formación Continua

Tipo de Formación: * Seleccione...

Lugar: * Seleccione...

Institución: * Seleccione...

Intensidad Horaria: * Horas totales del proceso de formacion

Porcentaje Descuento: Solo aplica para estudios realizados en la Universidad Simon Bolivar

Fecha Inicio: * dd/mm/aaaa

Fecha Terminación: dd/mm/aaaa

Fecha Título: dd/mm/aaaa

Descripción:

Estado del Registro: Desactivado

Formulario de Registro de Formación Continua

- El campo **Tipo de Formación***se selecciona el tipo de formación que se desarrolló, las cuales pueden ser cursos cortos, diplomados, capacitaciones y demás.
- El campo **Lugar*** es el lugar donde se desarrolló la formación, si no encuentra el lugar selecciona la opción OTRO.
- El campo **Institución***se busca la institución donde se realizó la formación, si no esta se selecciona la opción OTRO.
- El campo **Intensidad Horaria*** es el número de horas que dura la formación.
- El campo **Porcentaje de Descuento** es el porcentaje que se le descuenta en caso de ser formaciones internas a la Universidad Simón Bolívar.
- El campo **Fecha Inicio***es la fecha de inicio de la formación.
- El campo **Fecha Terminación** es la fecha de terminación de la formación, si aún se sigue desarrollando no se asigna fecha de terminación.

- El campo **Fecha Titulo** es en caso de haber terminado la formación se asigna la fecha de título.
- El campo **Descripción** es para la descripción de la formación y observaciones.
- El campo **Estado del Registro** siempre comienza desactivado hasta que Recursos Humanos lo active.

Administrar Manejo de Idiomas.

En la opción **Manejo de Idiomas** se referencia el manejo de idiomas del docente, el nivel de lectura, escritura, y conversación. El formulario que se debe diligenciar para registrar un nuevo idioma es el siguiente:

Formulario de detalle de manejo de idiomas del docente

Idioma: * Seleccione...

Nivel de Lectura: * Seleccione...

Nivel de Escritura: * Seleccione...

Nivel de Conversación: * Seleccione...

Estado del Registro: Desactivado

Guardar Limpiar Cancelar

Formulario de Registro de Idiomas

- El campo **Idioma*** se selecciona el idioma que se quiera registrar.
- El campo **Nivel de Lectura*** se selecciona el nivel de lectura.
- El campo **Nivel de Escritura*** se selecciona el nivel de escritura.
- El campo **Nivel de Conversación*** se selecciona el nivel de conversación.
- El campo **Estado del Registro** siempre comienza desactivado hasta que Recursos Humanos lo active.

Administrar Manejo de Software

En la opción **Manejo de Software** se referencia el manejo de software del docente, el nivel de lectura, escritura, y conversación. El formulario que se debe diligenciar para registrar un nuevo software es el siguiente:

- El campo **Software*** se selecciona el software que se quiera registrar.
- El campo **Nivel de Manejo*** se selecciona el nivel de manejo del software.
- El campo **Estado del Registro** siempre comienza desactivado hasta que Recursos Humanos lo active.

Administrar Eventos.

En la opción **Eventos** del menú lateral hace referencia a la administración de los eventos a los cuales ha participado el docente. El formulario que se debe diligenciar para registrar un nuevo evento es el siguiente:

- El campo **Tipo de Evento*** se selecciona el tipo de evento al cual se asistió donde se encuentran congresos, eventos, seminarios entre otros.
- El campo **Nombre del Evento*** es para el nombre del evento al cual se asistió.
- El campo **Fecha Evento*** es para la fecha en la cual se desarrolló el evento.
- El campo **Ubicación del Evento** es para la ubicación del evento al cual asistió el docente.
- El campo **Lugar del Evento*** es el lugar donde se desarrolló el evento, si no encuentra el lugar selecciona la opción OTRO.
- El campo **Tipo de Participación*** es el tipo de participación que tuvo el docente en el evento, entre las cuales esta participante, orador, organizador entre otros.
- El campo **Descripción del Evento** es para la descripción del evento al cual se asistió.
- El campo **Estado del Registro** siempre comienza desactivado hasta que Recursos Humanos lo active.

Administrar Distinciones

En la opción **Distinciones** se hace referencia a la administración de las distinciones del docente, se podrán registrar, consultar, modificar y eliminar los datos de las distinciones otorgadas. El formulario que se debe diligenciar para registrar nuevas distinciones es el siguiente:

Formulario Detalle de Distinciones
?

Tipo de Distinción: *

Nombre de la Distinción: *

Entidad que Otorgó: *

Lugar de la Distinción: *

Fecha: *

Ambito: *

Estado del Registro:

Formulario de Registro de Distinciones

- El campo **Tipo de Distinción*** se selecciona el tipo de distinciones que pudo haber recibido el docente entre los que se encuentra placa de honor, medalla, diploma y demás.
- El campo **Nombre de la Distinción*** es para el nombre de la distinción obtenida.
- El campo **Entidad que Otorgó*** es para el nombre de la entidad que otorgo la distinción.
- El campo **Lugar de la Distinción*** es el lugar donde se entregó la distinción, si no encuentra el lugar selecciona la opción OTRO.
- El campo **Fecha*** es para la fecha en la cual se entregó la distinción.
- El campo **Ámbito*** se selecciona el ámbito de la distinción, entre lo que se encuentra ámbito nacional o internacional.
- El campo **Estado del Registro** siempre comienza desactivado hasta que Recursos Humanos lo active.

Administrar Investigaciones

En la opción **Investigaciones** se hace referencia a la administración de las investigaciones del docente, se podrán registrar, consultar, modificar y eliminar los datos de las investigaciones otorgadas. El formulario que se debe diligenciar para registrar nuevas investigaciones es el siguiente:

Formulario Registro de Investigaciones

Tipo de Investigación: * Seleccione...

Nombre: *

Contexto: * Seleccione...

Financiación: *

Fecha Inicio: * dd/mm/aaaa

Fecha Final: * dd/mm/aaaa

Etapa: * Seleccione...

Descripción: *

Guardar Limpiar Cancelar

- El campo **Tipo de Investigación*** se selecciona el tipo de investigación que realizó el docente entre los que se encuentran investigación básica, avanzada, desarrollo tecnológicos y demás.
- El campo **Nombre*** es para el nombre de la investigación.
- El campo **Contexto*** es para contexto de la investigación, el cual puede ser interno o externa a la universidad.
- El campo **Financiación*** es para describir el tipo de financiación que se obtiene a la investigación.
- El campo **Fecha Inicio*** es la fecha de inicio de la investigación.
- El campo **Fecha Fin** es la fecha de terminación de la investigación, si es una investigación que aún sigue desarrollando no se asigna fecha de fin.
- El campo **Etapas*** se selecciona la etapa en la que se encuentra la investigación la cual puede ser inicial, en proceso, final y demás.
- El campo **Descripción** es para la descripción de la investigación y todos los detalles que se quieran añadir al registro.

Administrar Publicaciones

En la opción del menú lateral **Publicaciones** hace referencia a la administración de las publicaciones como lo es el registro, la consulta, la modificación y eliminación de publicaciones. Lo primero que hay que hacer después de haber entrado a la opción de publicaciones es en el campo **Tipo de Publicación** donde se desplegará una serie de opciones para seleccionar.

Dependiendo del Tipo de Publicación se mostraran más Campos

Formulario de Registro de Publicaciones

Dependiendo de las opciones que seleccione serán los campos que se muestren para diligenciar.

En primer caso tomaremos la opción **Artículo**:



Formulario de Registro de Artículos

Tipo de Publicación: * ARTICULO

Nombre: *

Fecha: * dd/mm/aaaa

Revista:

ISSN:

Medio de Divulgacion: Seleccione...

Idioma: Seleccione...

Sitio Web:

Resumen:

- El campo **Nombre*** es para el nombre del artículo.
- El campo **Fecha*** es para la fecha de publicación del artículo.
- El campo **Revista** es la revista en el que se publicó el artículo (Si Aplica).
- El campo **ISSN** es el número de identificación del artículo.
- El campo **Medio de Divulgación** se selecciona el tipo de divulgación del artículo que puede ser internet, radio, y demás.
- El campo **Idioma** se selecciona el idioma del artículo.
- El campo **Sitio Web** es para la dirección del sitio web donde se ubica el artículo.
- El campo **Resumen** es para describir un resumen del artículo referenciado.

Para la opción de **Libro** es el siguiente caso:

Formulario de Registro
de Libros

Tipo de Publicación: *	LIBRO
Nombre: *	
Fecha: *	dd/mm/aaaa
Editorial:	
ISBN:	
Medio de Divulgación:	Selecione...
Idioma:	Selecione...
Sitio Web:	
Descripción:	

- El campo **Nombre*** es para el nombre del libro.
- El campo **Fecha*** es para la fecha de publicación del libro.
- El campo **Editoriales** la donde se lleva a cabo el proceso de publicación.
- El campo **ISBN** es el número de identificación del libro.
- El campo **Medio de Divulgación** se selecciona el tipo de divulgación del libro que puede ser internet, radio, y demás.
- El campo **Idioma** se selecciona el idioma del libro.
- El campo **Sitio Web** es para la dirección del sitio web donde se ubica el libro.
- El campo **Descripción** es para describir un resumen del libro referenciado.

Para la opción de **Capítulo Libro** es el siguiente caso:

Formulario de Registro
de Capitulo de Libro

Tipo de Publicación: *	CAPITULO LIBRO
Nombre: *	
Nombre del Capitulo: *	
Fecha: *	dd/mm/aaaa
Lugar Publicación:	Seleccione...
Editorial:	
ISBN:	
Medio de Divulgación:	Seleccione...
Idioma:	Seleccione...
Sitio Web:	
Descripción:	

- El campo **Nombre*** es para el nombre del libro.
- El campo **Nombre del Capitulo*** es para el nombre del capítulo del libro.
- El campo **Fecha*** es para la fecha de publicación del capítulo del libro.
- El campo **Lugar de Publicación** se selecciona el lugar donde se desarrolló el capítulo del libro.
- El campo **Editoriales** la donde se lleva a cabo el proceso del capítulo del libro.
- El campo **ISBN** es el número de identificación del libro.
- El campo **Medio de Divulgación** se selecciona el tipo de divulgación del libro que puede ser internet, radio, y demás.
- El campo **Idioma** se selecciona el idioma del libro.
- El campo **Sitio Web** es para la dirección del sitio web donde se ubica el libro.
- El campo **Descripción** es para describir un resumen del capítulo del libro.

Para la opción de **Ensayo** es el siguiente caso:

Tipo de Publicacion: * ENSAYO

Nombre: *

Fecha: * dd/mm/aaaa

Medio de Divulgacion: Seleccione...

Idioma: Seleccione...

Sitio Web:

Descripcion:

- El campo **Nombre*** es para el nombre del ensayo.
- El campo **Fecha*** es para la fecha de publicación del ensayo.
- El campo **Medio de Divulgación** se selecciona el tipo de divulgación del ensayo que puede ser internet, radio, y demás.
- El campo **Idioma** se selecciona el idioma del ensayo.
- El campo **Sitio Web** es para la dirección del sitio web donde se ubica el ensayo.
- El campo **Descripción** es para describir un resumen del capítulo del ensayo.

Para la opción de **Columna** es el siguiente caso:

Tipo de Publicacion: * COLUMNA

Nombre: *

Fecha: * dd/mm/aaaa

Periodico:

Medio de Divulgacion: Seleccione...

Sitio Web:

Descripcion:

- El campo **Nombre*** es para el nombre de la columna.
- El campo **Fecha*** es para la fecha de publicación de la columna.
- El campo **Periódico** es para el nombre del periódico donde se publicó la columna.
- El campo **Medio de Divulgación** se selecciona el tipo de divulgación de la columna que puede ser internet, radio, y demás.
- El campo **Sitio Web** es para la dirección del sitio web donde se ubica la columna.
- El campo **Descripción** es para describir un resumen de la columna.

Para la opción de **Otro** es el siguiente caso:

Formulario de Registro de Otras Publicaciones

Tipo de Publicación: * OTRO

Nombre: *

Fecha: * dd/mm/aaaa

Descripción:

- El campo **Nombre*** es para el nombre de la publicación.
- El campo **Fecha*** es para la fecha de publicación.
- El campo **Descripción** es para una pequeña descripción de la publicación.

Generar Hoja de Vida.

En la opción **Generar HV** se puede generar el reporte de hoja de vida del docente; dando la opción de poder seleccionar lo que se quiera que se muestre en la hoja de vida.

En la Hoja de Vida solo se mostrara la información que este validada y activa por parte de recursos humanos.

Generar Hoja de Vida

Formulario de Reporte de Hoja de Vida

A continuación seleccione los items que serán incluidos en el reporte de la hoja de vida del docente

<input checked="" type="checkbox"/> Estudios realizados	<input checked="" type="checkbox"/> Idiomas	<input checked="" type="checkbox"/> Distinciones
<input checked="" type="checkbox"/> Experiencia Laboral	<input checked="" type="checkbox"/> Software	<input checked="" type="checkbox"/> Investigaciones
<input checked="" type="checkbox"/> Formación continua	<input checked="" type="checkbox"/> Eventos	<input checked="" type="checkbox"/> Publicaciones

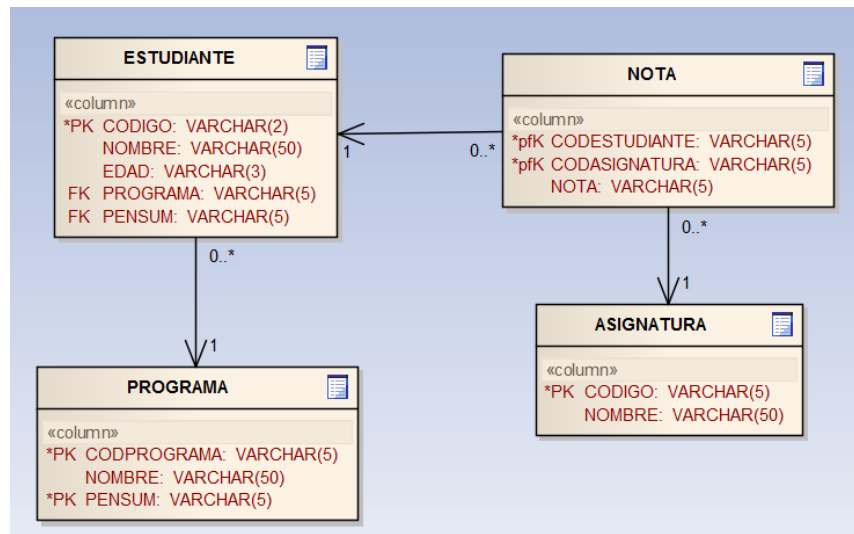
Botón Generar Hoja de Vida

Opciones que aparecerán en la hoja de vida.

Anexo E. DOCTRINE 2.0

Los archivos para la utilización de Doctrine serán ubicados en la carpeta “[libs](#)” del proyecto, donde se utilizara como una librería. Cada tabla de la base de datos DB2 que se quiera utilizar debe ser mapeada como una [Entidad](#) para poder acceder a su información y a las operaciones básicas. Las entidades mapeadas se guardaran en “[models/Entities](#)”, donde por lo general se recomienda que lleven el mismo nombre de la tabla; en tal caso si la tabla se llama Docente su entidad respectiva será Docente.php. Todas las anotaciones que utiliza Doctrine en las entidades utilizan el símbolo @ y se encierran en su totalidad entre los símbolos de comentarios /** y */.

La estructura interna de las entidades será explicada en base a las siguientes tablas con sus respectivos atributos.



Primero se le realiza el mapeo a la tabla asignatura que quedara de la siguiente forma.

```
<? Php
/**@Entity @Table(name="asignatura")*/
class Asignatura
{
    /** @Id @Column (length = 5) */
    Private $codigo;
    /** @Column (length = 50) */
    Private $nombre;
```

Se marca la clase Asignación como una entidad añadiendo la línea donde se ve la palabra [@Entity](#). Se le debe asignar la relación del nombre de la tabla y la entidad con la línea [@Table\(name="mensaje"\)](#). El siguiente paso después de marcar una clase PHP como una

entidad es el mapeo de sus propiedades. Para configurar una propiedad se utiliza `@Column` donde se especifica el *type* con un atributo específico, el cual si no se especifica tomara el valor por defecto *string* y al mismo tiempo se le debe asignar la opción *length* la cual establece la longitud del tipo especificado anteriormente. Vale agregar que algunos tipos como *integer* no necesitan que se les establezca *length* porque ellos no aplican en este caso. Cada entidad debe tener una clave primaria identificadora. Es posible seleccionar el campo que sirve como identificador con la anotación `@Id`. Cuando no se especifica explícitamente un nombre de la columna a través de la opción *name*, Doctrine asume el nombre del campo es también el nombre de la columna de la tabla.

En caso que la llave primaria sea una llave compuesta como en el caso de la tabla nota, se deberá indicar esa llave como se ve en el siguiente código. Además el constructor deberá recibir esos valores de la llave compuesta.

```
<? Php
/**@Entity @Table(name="nota")*/
class Nota
{
    /** @Id @Column (length = 5) */
    Private $codEstudiante;
    /** @Id @Column (length = 5) */
    Private $codAsignatura;
    /** @Column (length = 5) */
    Private $nota;

    public function __construct($codEstudiante, $codAsignatura)
    {
        $this->codEstudiante = $codEstudiante;
        $this->codAsignatura= $codAsignatura;
    }
    METODOS SET Y GET DE TODOS LOS ATRIBUTOS
}
}
```

Para trabajar con las relaciones se deben añadir unas notaciones para identificar el tipo de relación, cuales entidades participan y demás características necesarias. En este caso se realizara la relación entre Estudiante y Nota.

```

<? Php
/**@Entity @Table(name="estudiante")*/
class Estudiante
{
    /** @Id @Column (length = 2) */
    Private $codigo;
    /** @Column (length = 50) */
    Private $nombre;
    /** @Column (length = 3) */
    Private $edad;
    /** @Column (length = 5) */
    Private $programa;
    /** @Column (length = 5) */
    Private $pensum;
    /**
     * @OneToMany (targetEntity="Nota", mappedBy="estudiante")
     */
    Private $nota;

    public function __construct(){ }

    METODOS SET Y GET DE TODOS LOS ATRIBUTOS
}
?>

```

La anotación que se le añade de más es una relación OneToMany donde el valor de **targetEntity** es el nombre de la entidad Many, y el valor de **mappedBy** la entidad One. Esto se ubica antes de declarar el objeto nota con el cual es que se establece la relación. En la entidad Nota también toca añadir líneas de notación como las siguientes.

```

class Nota
{
    /** @Id @ManyToOne(targetEntity="Estudiante", inversedBy="nota")
     * @JoinColumn (name = "estudiante", referencedColumnName = "codigo")
     */
    Private $codEstudiante;
}

```

Se cambia la notación inicial al codEstudiante por una ManyToOne donde **targetEntity** es el nombre de la entidad One, el **inversedBy** es el nuevo objeto declarado en la entidad Estudiante. El **JoinColumn** es el que se usa para referenciar las equivalencias de nombres donde **name** es el nombre nuevo y **referencedColumnName** el nombre de la columna en la tabla. De igual forma se hace para establecer la relación entre las entidades Nota y

Asignatura. Para la relación entre Estudiante y Programa se le agregan a estudiante las siguientes notaciones.

```
/** @ManyToOne(targetEntity="Programa")
 * @JoinColumn({
 * @JoinColumn (name = "programa", referencedColumnName = "codprograma"),
 * @JoinColumn (name = "pensum", referencedColumnName = "pensum")
 * })*/
Private $programa;
```

Y se elimina el objeto pensum de la entidad Estudiante. En Programa quedaría de la siguiente forma:

```
<? Php
/** @Entity @Table(name="programa")*/
class Programa
{
    /** @Id @Column (length = 5) */
    Private $codPrograma;
    /** @Column (length = 50) */
    Private $nombre;
    /** @Id @Column (length = 5) */
    Private $pensum;
    /**
     * @OneToMany (targetEntity="Estudiante", mappedBy="programa")
     */
    Private $estudiante;
    public function __construct(){ }

    METODOS SET Y GET DE TODOS LOS ATRIBUTOS
}
?>
```

Además de crear el mapeo de las entidades, Doctrine maneja la persistencia de una forma rápida y ágil ya que con solo correr las siguientes líneas se realiza el guardado de datos.

```
$user = new User;
$user->setName('Mr.Right');
$em->persist($user);
$em->flush();
```

De igual forma se ejecuta la eliminación.

```
$em->remove($user);  
$em->flush();
```

También cuenta con búsqueda de datos. Por llave primaria.

```
$user = $em->getRepository('MyProject\Domain\User')->find($id);  
$ user = $ em -> getRepository ( 'MyProject \ dominio \ usuario' ) -> findOneBy ( array ( '  
nick' => 'romanb' ) );
```

ARQUITECTURA MODELO VISTA CONTROLADOR

Trata de separar la programación del proyecto en tres partes y del mismo modo ser más factible para realizar cambios a futuro.

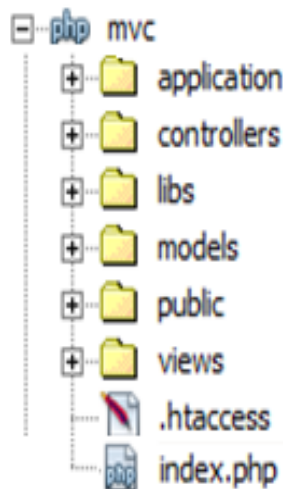
Modelo: Aquí se programan todo aquello relacionado con las bases de datos, es decir, las entradas y salidas de datos y se devuelven como se necesiten desde el programa principal. Siendo imaginativos sería como acceder a los datos de algo a través de un API y el API sería el modelo.

Vista: Aquí se programa la parte visual del software, o lo que es lo mismo, la parte que usa el usuario. En el caso de un sitio web la parte de HTML, CSS y Java Script normalmente.

Controlador: Es la lógica del programa. Aquella que le pide al modelo los datos y los muestra en la vista. Vamos, lo que viene a ser el núcleo.

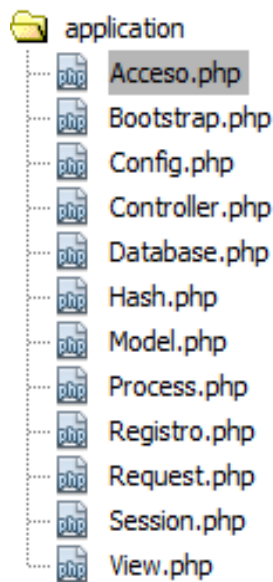
MVC Aplicado en PHP.

En el proyecto se realizara el esquema mvc como se observa en la siguiente imagen.



En esta imagen se observa la división que tendrá el proyecto; la primera es application donde estarán todas las clases principales, sigue controllers en la cual estarán todos los controladores que se encargaran de la comunicación entre la vista y el modelo. La carpeta libs que tendrá todas las librerías necesarias para el proyecto. La carpeta models que contendrá todos los modelos del proyecto; la carpeta views que de la misma forma contendrá todas las vistas del proyecto, y la carpeta public que tendrán todas las clases de orden públicas como el código css, javascript, jquery y demás.

Además se encontraran el index.php y el htaccess.



La clase Acceso.php contiene todo lo referente al control de acceso a los usuarios.

La clase Bootstraps.php contiene todo el manejo de las URL amigables y como se manejan estas en su distribución de controladores y argumentos.

La clase Config.php contiene toda la información de inicio por defecto.

Controller.php es la clase principal de todos los controladores la cual le heredara todas sus funciones.

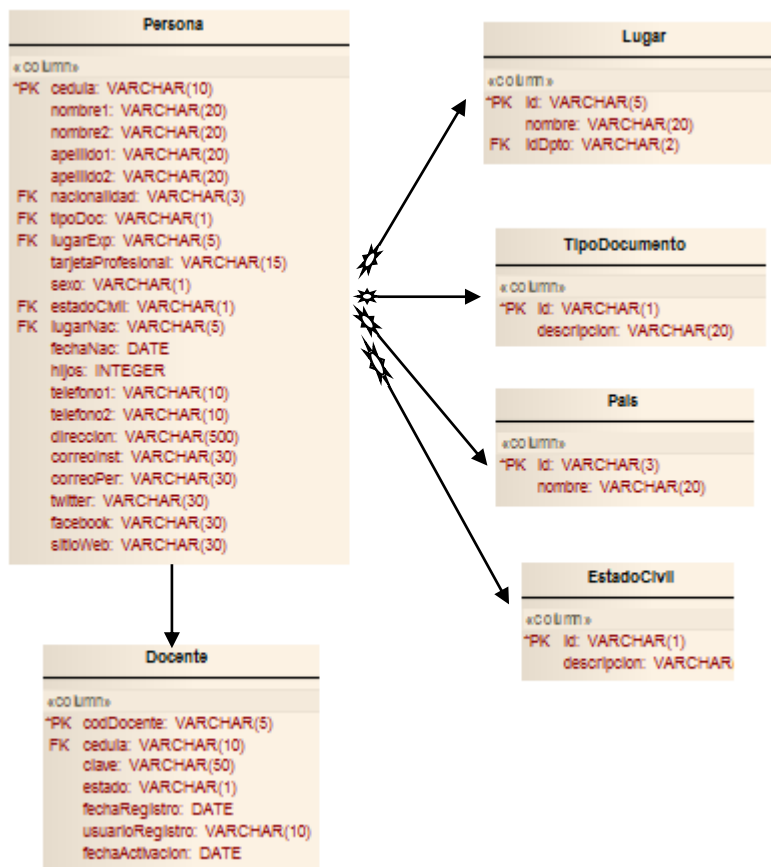
De igual forma, la clase Model.php es la clase principal de todos los modelos. Y la View.php es la clase principal de todas las vistas.

Las demás clases que se observan en la carpeta application son para las configuraciones de encriptación de la información, carga de procesos y manejo de sesiones.

En la carpeta controllers irán contenidos todos los controladores del proyecto, como ejemplo se toma al Docente, el cual su respetivo controllers se llamara docenteControllers que tendrá todo el código necesario para la comunicación entre el modelo que estará en la carpeta models llamado docenteModel y la vista que estará en la carpeta views y se llamara docente.

Anexo F. Desarrollo del módulo docente

Situación: El modulo permite el ingreso, edición, eliminación y selección de la información básica del docente. A continuación se ve la siguiente imagen con la información y las tablas relacionadas como lo son el tipo de documento, el lugar, el país y el estado civil entre otras.



Entidades:

De aquí se pasa a realizar las entidades por cada tabla en las cuales se definen los campos y las relaciones de cada tabla. Cada entidad se debe llamar de la misma forma como esta en la base de datos. Comenzando con la tabla persona a la cual se describirá las notaciones generales para el resto de entidades.

1. <?php
- 2.
3. Namespace Entities;
- 4.
5. /**
6. **@Entity @Table (name = "docente.persona")*

```

7. **/
8. class Persona {
9.
10. /**
11.  *@Id @column (type= "string", length = 10)
12.  */
13. private $cedula;
14.
15. /** @column (type= "string", length = 10) */
16. private $nombre1;
17.
18. /**
19.  *@ManyToOne(targeEntity = "TipoDocumento")
20.  *@JoinColumn(name = "tipoDoc", referencedColumnName = "id")
21.  */
22. private $tipoDoc;
23.
24. function __construct(){
25.
26. }
27.
28. //Se realizan todos los método get y set//
29.
30. ?>

```

La línea 3 se le debe agregar a todas las entidades que se realizan lo cual es para especificar que se encuentra en el espacio Entities.

La línea 6 especifica el nombre de la entidad persona, en este caso particular se le antepone docente.persona debido a que docente es el esquema en la base de datos y persona la tabla.

La línea 11 especifica que ese campo es el ID de la tabla, la propiedad @column declara que es una columna con un tipo y una longitud. El tipo si es string se le asigna una longitud, si es tipo integer o date no se le asigna longitud.

Las demás columnas se definen como en la línea 15 cambiando el tipo y la longitud cuan sea necesaria.

La línea 19 se declara una relación muchos a uno entre la entidad persona y tipo de documento donde la propiedad targeEntity es el nombre de la otra entidad. El JoinColumn sirve para referenciar los nombres de los campos que están en la tabla, name el nombre en la tabla persona y referencedColumnName el nombre en la tabla tipo documento.

Las demás relaciones se realizan como el caso tipo de documento especificando cada propiedad.

Así se realizan las entidades de tipo documento, país, estado civil y lugar con sus cambios específicos para cada una. Cabe aclarar que a estas últimas entidades no se le asignan las relaciones ya que la relacionan es unidireccional.

Modelos:

El siguiente paso es la creación de los modelos los cuales estarán ubicados en la carpeta models y su nombre será por ejemplo personaModel.php o tipoDocumentoModel.php

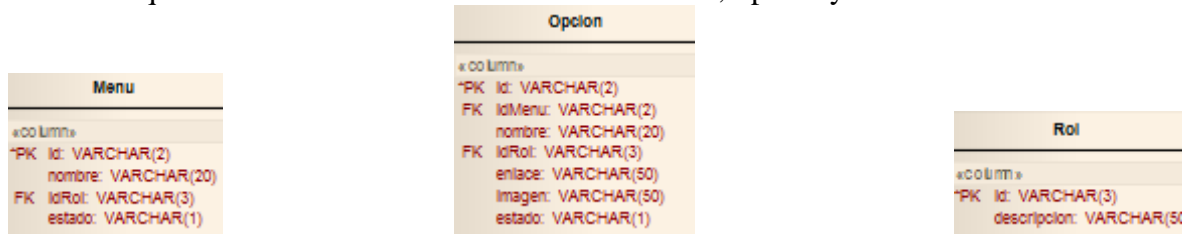
```
1. <?php
2.
3. class personaModel extends Model {
4.
5.     public function __construct() {
6.
7.     }
8.     $this->instance = $this->loadObjeto('Persona')
9. }
10. }
11.
12. ?>
```

Lo único que cambia para cada modelo es el nombre de la clase y el objeto que se carga, que en este caso es el nombre de la entidad.

Menú Dinámico:

El menú se carga de forma dinámica realizando las consultas a la base de datos donde lo que compara es el menú, las opciones, los roles y perfiles.

Las tablas que deben estar en la base de datos son menú, opción y rol.



La tabla menú contiene las opciones que saldrán en la parte superior. Por ejemplo en este caso se quiere que salga la opción Docente en la parte superior, pero además también estará Detallar Docente, pero al estar este en la opción principal como 0 no se mostrara pero sus opciones estarán ahí.

ID	NOMBRE	IDROL	ESTADO	URL	NIVEL	PRINCIPAL
1	Docentes	2	1	docente	1	1
2	Detallar Docer	22	1	detallar	1	0
*						

Inicio Docentes

La tabla opción tiene las opciones que se quiere que aparezcan en el menú docente con el ID = 2. Además se escribe la url de cada opción.

ID	IDMENU	NOMBRE	IDROL	ENLACE	IMAGEN	ESTADO
1	1	Listar docentes	25	docente/		1
2	2	Datos Básicos	22	docente/detallardocente		1
3	2	Estudios Realizados	35	docente/estudios		1
4	2	Experiencia Laboral	45	docente/experiencias		1
5	2	Formación Continua	55	docente/formacioncontinua		1
6	2	Manejo de Idiomas	65	docente/docenteidioma		1
7	2	Manejo de Software	75	docente/docentesoftware		1
8	2	Eventos	85	docente/eventos		1
9	2	Distinciones	95	docente/distinciones		1
91	2	Investigaciones	105	docente/investigaciones		1
92	2	Publicaciones	115	docente/publicaciones		1
98	2	Reiniciar Clave	11	login/reiniciarclave		1
99	2	Cambiar Clave	10	login/cambiarclave		0
*						

Opciones de Menú

- ▶ Datos Básicos
- ▶ Estudios Realizados
- ▶ Experiencia Laboral
- ▶ Formación Continua
- ▶ Manejo de Idiomas
- ▶ Manejo de Software
- ▶ Eventos
- ▶ Distinciones
- ▶ Investigaciones
- ▶ Publicaciones

Y la tabla Rol se especifican todos los roles y su Id, el cual se referencia en la tabla menú y opción.

ID	DESCRIPCION
1	ADMINISTRADOR
2	MENU DOCENTE

Controlador.

Este es el encargado de conectar el modelo con la vista. Todos los controladores van dentro de la carpeta controllers y reciben el nombre de personaController para este caso en específico; otros casos pueden ser cargoController, docenteController, etc.

Primero dentro del controller se carga el modelo el cual a su vez carga el objeto de la entidad requerida.

```
$this->_persona = $this->loadModel('persona');
```

Con esta línea se tiene el objeto en la variable `$this->_persona`, y en cualquier momento aquí puedes acceder a los métodos get y set con la siguiente línea:

```
$this->_persona->getInstance()->getNombre();
```

O

```
$this->_persona->getInstance()->setNombre();
```

Para pasar objetos o variables a la vista se utiliza la línea siguiente:

```
$this->_view->estudio = $this->_estudio->getInstance();
```

O

```
$this->_view->titulo = 'Estudio';
```

Donde en la primera línea se manda un objeto (estudio) a la vista, y la otra línea se manda una variable (titulo) también a la vista. Esta vista más adelante la tenemos que renderizar con la siguiente línea donde el parámetro 'estudio' es el nombre de la vista que se encuentra en la carpeta views.

```
$this->_view->renderizar('estudio', 'Detallar Docente');
```

Y esta línea muestra como redireccionar:

```
$this->redireccionar("docente/experiencias");
```

Y el valor que se le manda al método redireccionar es docente/experiencias; donde docente es el controlador(docente es concatenado con controller que sería igual a docenteController) y experiencias sería el método que está dentro del controller.

Vista

En la vista ya se tiene las variables u objetos enviados desde el controller que para nuestro caso se llama estudio y título. Para acceder a la variable título solo basta con la siguiente línea:

```
$this->titulo;
```

Y para acceder al objeto estudio que mandamos a la vista es con la siguiente línea:

```
$this->estudio->getNombre();
```

O cualquiera que sea el método de ese objeto.