



Universidad Central de Venezuela

Facultad de Ciencias

Escuela de Computación

**Trabajo Especial de Grado**

**Desarrollo de una Aplicación Web para la automatización  
de la Unidad de Servicio Comunitario de la Facultad de Ciencias**

Presentado ante la Ilustre  
Universidad Central de Venezuela

Por los Bachilleres

Br. Mariana Simón C.I: 17.124.125

Br. Jesús Mata C.I: 15.487.459

Para optar al título de  
Licenciado en Computación

Tutores:

Profa. Jossie Zambrano

Prof. Sergio Rivas

Caracas, Octubre 2010

---

## Acta

Quienes suscriben, miembros del Jurado designado por el Consejo de Escuela de Computación, para examinar el Trabajo Especial de Grado presentado por los Bachilleres Mariana Simón Luna C.I: 17.124.125 y Jesús Miguel Mata Mora C.I: 16.033.952, titulado: **"Desarrollo de una Aplicación Web para la automatización de la Unidad de Servicio Comunitario de la Facultad de Ciencias"**, a los fines de optar por el título de Licenciado en Computación, dejen constancia de lo siguiente:

Leído como fue, dicho trabajo por cada uno de los miembros del jurado, se fijó el día 25 de Octubre de 2010 a las 03:30 pm en la Sala PBIII de la Escuela de Computación, para que sus autores lo defendieran en forma pública, mediante una presentación oral de su contenido, luego de lo cual respondieron las preguntas formuladas. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobarlo con una nota de \_\_\_\_\_ puntos.

En fé de lo cual se levanta la presente Acta, en Caracas los 25 días del mes de Octubre del año dos mil diez.

---

Profa. Jossie Zambrano (Tutora)

---

Prof. Sergio Rivas (Tutor)

---

Profa. Alecia Eleonora Acosta (Jurado)

---

Prof. Eugenio Scalise (Jurado)

## **Agradecimientos y Dedicatorias**

A Dios por la oportunidad de vivir esta experiencia, dándome las fuerzas necesarias.

A mis padres y hermano por apoyarme incondicionalmente en todo momento.

A mis tíos Clara y Johnny por darme momentos inolvidables de tranquilidad, y en especial a mi primo Johnnatan.

A Charlie por estar presente siempre ofreciéndome su ayuda y sus lindas palabras.

A mi compañero Jesús por ayudarme a que este trabajo fuese posible.

A nuestro tutor Sergio por ser nuestro guía brindándonos su apoyo, su paciencia y su ayuda. A Jossie por su colaboración y soporte.

A la Unidad de Servicio Comunitario y a la División de Control de Estudios, por su participación.

Finalmente quiero dedicarle este trabajo a mi familia, a Charlie y a todas aquellas personas involucradas que lo hicieron posible.

Con mucho amor,  
Mariana Simón

A Dios por permitirme alcanzar esta meta de mi vida con salud, por fortalecer mi espíritu en los momentos difíciles y haber puesto en mi camino a aquellas personas que de una forma u otra contribuyeron a este logro.

A mi familia por todo el apoyo que me brindaron durante mis estudios, por encontrar en ellos siempre una palabra de aliento para ayudarme a continuar y que de forma incondicional entendieron mis ausencias y malos momentos. A mi madre, por ser siempre la fuente de mi inspiración, por enseñarme todo lo que se necesita para triunfar en la vida y por el sacrificio que hoy día se convierten en nuestro triunfo.

A nuestros tutores Jossie y Sergio por brindarnos la oportunidad de compartir este trabajo y por mostrarnos el camino correcto para seguir adelante.

A mis amigos y compañeros que durante este proceso de formación contribuyeron con sus palabras y acciones para brindar apoyo en los momentos indicados.

A Mariana, por ser una buena amiga durante gran parte de la carrera. Por tener la paciencia y estar atenta a cada uno de los detalles durante la elaboración de este trabajo, dándole siempre un aporte especial.

A todas aquellas personas que sin saberlo, fueron y seguirán siendo parte esencial en mi vida por haber compartido este largo camino conmigo, y que hoy, forman parte de este Trabajo Especial de Grado.

Con cariño,  
Jesús Mata

## Resumen

La Unidad de Servicio Comunitario requiere la organización de sus procesos con el uso de las Tecnologías de Información y Comunicación (TIC), con el fin de gestionar de mejor forma la información de los involucrados en las actividades del Servicio Comunitario, beneficiando a la comunidad estudiantil, docente y el personal que la conforma.

Con el propósito de apoyar a la Unidad de Servicio Comunitario se plantea como objetivo de este Trabajo Especial de Grado el desarrollo de una aplicación Web que automatiza los procesos relacionados con los cursos de inducción (inscripción y control de asistencia) y proyectos (inscripción, cronograma de actividades y calificación).

En éste trabajo se utiliza el método de desarrollo de software ágil Programación Extrema, que permite la interacción continua con el cliente para cubrir sus necesidades referentes al software en un tiempo oportuno. Se adoptan tecnologías software libre como: MySQL, Ruby on Rails, JavaScript, HTML, CSS, entre otros.

### **Palabras Clave**

Unidad de Servicio Comunitario, Servicio Comunitario, Prestador Comunitario, Tutor Académico, Tutor Comunitario, Aplicación Web, Programación Extrema, Ruby on Rails.

---

## Índice

Acta .....	II
Agradecimientos y Dedicatorias .....	III
Resumen .....	V
Índice .....	VI
Índice de Figuras .....	IX
Índice de Tablas .....	XI
Introducción .....	12
CAPÍTULO I - Planteamiento del problema y Servicio Comunitario.....	14
1.1. Servicio Comunitario.....	14
1.2. Objetivos del Servicio Comunitario .....	14
1.3. Actores involucrados en el Servicio Comunitario de la Facultad Ciencias ..	15
1.3.1. Comisión de Servicio Comunitario .....	15
1.3.2. Unidad de Servicio Comunitario .....	16
1.3.3. Tutores académicos .....	17
1.3.4. Tutores comunitarios.....	17
1.3.5. Prestadores de Servicio Comunitario .....	18
1.4. Aplicación anterior del Servicio Comunitario .....	19
1.5. Requisitos para la prestación de Servicio Comunitario en la Facultad de Ciencias .....	19
1.5.1. Requisitos para los estudiantes .....	19
1.5.2. Requisitos para ser tutor académico .....	20
1.6. Procesos .....	20
1.6.1. Inscribir el curso de inducción .....	20
1.6.2. Controlar asistencia al curso de inducción .....	21
1.6.3. Aprobar el curso de inducción .....	21

---

1.6.4. Seleccionar la Comunidad .....	21
1.6.5. Inscribir el Proyecto de Servicio Comunitario.....	22
1.6.6. Cambiar de comunidad .....	22
1.6.7. Cambiar de tutor académico .....	23
1.6.8. Controlar el Proyecto de Servicio Comunitario .....	23
1.6.9. Entregar documentos finales del Proyecto de Servicio Comunitario ....	23
1.6.10. Presentar el proyecto de Servicio Comunitario .....	24
1.6.11. Calificar el Servicio Comunitario .....	25
CAPÍTULO II - Tecnologías.....	27
2.1. Aplicaciones Web .....	27
2.2. Cliente Servidor .....	27
2.3. Tecnologías del lado del cliente.....	29
2.3.1. Hojas de Estilo en Cascada (CSS) .....	29
2.3.2. JavaScript.....	30
2.4. Tecnologías del lado del servidor .....	30
2.5. Sistemas de Gestión de Bases de Datos .....	31
2.6. Sistema de Gestión de Base de datos MySQL.....	31
2.7. Ruby On Rails .....	32
2.7.1. Ruby .....	32
2.7.2. Rails .....	33
2.8. Patrón Modelo Vista Controlador en Rails.....	33
2.8.1. Modelo.....	33
2.8.2. Vista .....	34
2.8.3. Controlador.....	34
2.9. Filosofía de Ruby On Rails .....	34
2.9.1. Convención sobre configuración .....	35
2.9.2. No te repitas .....	35
2.10. Plataforma.....	35
CAPÍTULO III - Marco Aplicativo .....	37

---

---

3.1. Metodología de Desarrollo de Software.....	37
3.2. Metodologías Ágiles.....	37
3.3. Programación Extrema .....	38
3.4. Actividades de la Programación Extrema .....	39
3.4.1. Planeación.....	40
3.4.2. Diseño .....	41
3.4.3. Codificación.....	42
3.4.4. Pruebas .....	42
3.5 Aspectos positivos de la Programación Extrema.....	43
3.6. Adaptación de la Programación Extrema .....	44
3.7. Iteración 0 .....	45
3.8. Iteración 1 .....	49
3.9. Iteración 2 .....	56
3.10. Iteración 3 .....	64
3.11. Iteración 4 .....	71
3.12. Iteración 5 .....	76
3.13. Iteración 6 .....	80
3.14. Iteración 7 .....	86
3.15. Iteración 8 .....	90
Conclusiones .....	94
Referencias Bibliográficas .....	96



---

## Índice de Figuras

Figura 1- Flujo a seguir por un estudiante para realizar el Servicio Comunitario...	26
Figura 2 - Cliente Servidor.....	28
Figura 3 - Patrón Modelo Vista Controlador. ....	34
Figura 4 - Plataforma de la aplicación de Servicio Comunitario.....	36
Figura 5 - Proceso de la Programación Extrema. ....	39
Figura 6 - Propuesta del Prototipo de Interfaz de Usuario. ....	48
Figura 7 - Parte modelo de Base de Dato Inicial.....	51
Figura 8 - Script SQL de la tabla (curso_induccion).....	53
Figura 9 - Patrón MVC de la aplicación servicio_comunitario. ....	53
Figura 11 - Autenticación de usuarios. ....	55
Figura 12 - Comprobante de Inscripción en el Curso de Inducción. ....	58
Figura 13 - Modelo del proceso de curso de inducción.....	59
Figura 14 - Almacenamiento de la creación de un Curso de Inducción. ....	61
Figura 15 - Formulario para la creación de un curso de inducción. ....	61
Figura 16 - Interfaz de Inscripción de un Estudiante en el Curso de Inducción. ....	62
Figura 17 - Almacenar Estudiante inscrito en un Curso de Inducción.....	62
Figura 18 - Código de barra que identifica al participante.....	65
Figura 19 - Diseño de interfaz de usuario para la evaluación en línea. ....	66
Figura 20 - Diseño del certificado del Curso de Inducción de Docentes.....	67
Figura 21 - Código para almacenar la asistencia del estudiante.....	68
Figura 22 - Código de creación de la Prueba en Línea.....	69

---

Figura 23 - Código para generar el comprobante de aprobado. ....	70
Figura 24 - Modelo de base de datos Proyecto de Servicio Comunitario. ....	73
Figura 25 - Código para almacenar la creación de Proyectos de Servicio Comunitario. ....	74
Figura 26 - Código guardar la inscripción de un estudiante en el Proyecto de Servicio Comunitario. ....	75
Figura 27 - Confirmar inscripción de proyecto de Servicio Comunitario de un estudiante. ....	77
Figura 28 - Diagrama de Flujo para la asignación de tutores académicos. ....	78
Figura 29 - Código para editar inscripción de estudiante en proyecto. ....	79
Figura 30 - Asignación de carga estudiantil para cada docente.....	81
Figura 31 - Menú de un docente que es tutor académico con su listado de estudiantes. ....	82
Figura 32 - Consulta a los docentes según escuela.....	83
Figura 33 - Parte del código de la asignación de tutores académicos. ....	84
Figura 34 - Asignación de Tutores Académicos primera pasada.....	85
Figura 35 - Asignación de Tutores Académicos segunda pasada .....	85
Figura 36 - Asignación Final de Tutores Académicos. ....	85
Figura 37 - Elaboración de Informe Final de Servicio Comunitario. ....	88
Figura 38 - Opciones para eliminar o editar informe de Servicio Comunitario. ....	88
Figura 39 - Descarga de presentación. ....	89
Figura 40 - Evaluación del Informe Final.....	91
Figura 41 - Calcular el resultado de la evaluación del estudiante. ....	92

---

---

## Índice de Tablas

Tabla 1 - Formato historias de Usuario. ....	45
Tabla 2 - Historia de Usuario para la iteración cero. ....	46
Tabla 3 - Historias de Usuario (Versión Inicial). ....	47
Tabla 4 - Historia de Usuario para la iteración uno. ....	50
Tabla 5 - Casos de Prueba Autenticación. ....	55
Tabla 6 - Historia de Usuario para la iteración dos. ....	57
Tabla 7 - Historia de Usuario para la iteración tres. ....	64
Tabla 8 - Historia de Usuario para la iteración cuatro. ....	72
Tabla 9 - Historia de Usuario para la iteración cinco. ....	76
Tabla 10 - Historia de Usuario para la iteración seis. ....	81
Tabla 11 - Historia de Usuario para la iteración siete. ....	87
Tabla 12 - Historia de Usuario para la iteración ocho. ....	90

## Introducción

La Unidad de Servicio Comunitario de la Facultad de Ciencias de la Universidad Central de Venezuela se crea en el año 2007, con la función de cumplir y hacer cumplir la ley de Servicio Comunitario del estudiante de educación superior gestionando diferentes procesos de capacitación, funcionamiento, convenios y atribuciones que deben realizar los estudiantes para obtener el título de pregrado.

En la actualidad, las actividades se realizan manualmente o en pequeñas aplicaciones, lo que genera desventajas para la organización, tales como, la inconsistencia de datos, desmejoras en la interfaz y la falta de automatización de las actividades de algunos procesos; aumentando la cantidad de tiempo en la que se llevan a cabo y dificultando el acceso a la información de los usuarios involucrados.

Además, hoy en día, disponer de acceso a un computador conectado a internet se ha convertido en algo habitual, por lo que muchas organizaciones han migrado sus servicios, información y aplicaciones a un entorno Web que le permita a sus usuarios acceder desde cualquier lugar del mundo, beneficiándolos en la comodidad, actualización, organización, rapidez, tiempo, entre otros.

Debido a que existe la necesidad de mejorar la gestión de los procesos, el objetivo de este Trabajo Especial de Grado es desarrollar una aplicación Web, que automatice los procesos relacionados a los cursos de inducción y proyecto de Servicio Comunitario, que organiza la Unidad de Servicio Comunitario. Ofreciendo a los usuarios las ventajas de las aplicaciones Web, reduciendo los costes de tiempo y recursos humanos, disponibilidad de operación y la comodidad del navegador como cliente, así como la facilidad para actualizar y mantener sin distribuir e instalar software.

---

Partiendo de la automatización de los procesos relacionados a los cursos de inducción y del proyecto de Servicio Comunitario, se deben elaborar ciertas actividades como gestionar la inscripción, el control de asistencia y calificación de los cursos de inducción, creación de comunidades, proyectos y tipos de proyectos, inscripción de proyecto de Servicio Comunitario, asignación de tutor académico y jurados, control de asistencia de los prestadores, entregas de documentos finales y presentaciones, calificación del Servicio Comunitario y consultas de algunos reportes.

Es importante la automatización de los procesos, ya que mejora la gestión de la información, realiza una colaboración significativa para la eficiencia operacional y la productividad de recursos humanos, ofrece un intercambio importante de información entre los usuarios relacionados al Servicio Comunitario y mejora la evaluación de los estudiantes, entre otros. La estructura del presente documento es la siguiente:

El Capítulo I se describe el planteamiento del problema relacionado al Servicio Comunitario. Cómo se llevan a cabo los procesos actualmente con respecto a los cursos de inducción y proyecto de Servicio Comunitario.

Las tecnologías utilizadas durante el desarrollo de la aplicación del Servicio Comunitario se detallan en el Capítulo II.

En el Capítulo III se presenta el Marco Aplicativo en donde se describe la implementación del método de desarrollo de software Programación Extrema y las iteraciones efectuadas durante el tiempo de desarrollo, especificando la planeación, diseño, codificación y pruebas de cada una de ellas.

Para culminar se expone la conclusión y las referencias bibliográficas utilizadas.

## **CAPÍTULO I - Planteamiento del problema y Servicio Comunitario**

A continuación se define el contexto del problema abarcado en este Trabajo Especial de Grado, para la automatización de las actividades llevadas a cabo durante los procesos relacionados al curso de inducción y de los proyectos de Servicio Comunitario.

### **1.1. Servicio Comunitario**

Según el artículo 4 de la ley de Servicio Comunitario del estudiante de educación superior, el Servicio Comunitario se define como “la actividad que deben desarrollar en las comunidades los estudiantes de educación superior que cursen estudios de formación profesional, aplicando los conocimientos científicos, técnicos, culturales, deportivos y humanísticos adquiridos durante su formación académica, en beneficio de la comunidad, para cooperar con su participación al cumplimiento de los fines del bienestar social, de acuerdo con lo establecido en la Constitución de la República Bolivariana de Venezuela y en esta Ley” (Ley de Servicio Comunitario del estudiante de educación superior, 2005).

### **1.2. Objetivos del Servicio Comunitario**

La ley de Servicio Comunitario de educación superior establece en su artículo 7 los objetivos de la práctica comunitaria, los cuales se mencionan a continuación: (Ley de Servicio Comunitario del Estudiante de Educación Superior, 2005)

- Fomentar en el estudiante, la solidaridad y el compromiso con la comunidad como norma ética y ciudadana.
- Hacer un acto de reciprocidad con la sociedad.

- Enriquecer la actividad de educación superior, a través del aprendizaje servicio, con la aplicación de los conocimientos adquiridos durante la formación académica, artística, cultural y deportiva.
- Integrar las instituciones de educación superior con la comunidad, para contribuir al desarrollo de la sociedad venezolana.
- Formar, a través del aprendizaje servicio, el capital social en el país.

### **1.3. Actores involucrados en el Servicio Comunitario de la Facultad Ciencias**

El Servicio Comunitario de la Facultad de Ciencias está conformado por distintos usuarios los cuales se encargan de realizar diferentes tareas, haciendo que funcione dicha unidad. A continuación se describen cada uno de estos usuarios (Acosta, Eleonora 2009).

#### **1.3.1. Comisión de Servicio Comunitario**

La Comisión de Servicio Comunitario está conformada por distintos docentes que representan a cada una de las licenciaturas (Acosta, Eleonora 2009).

Seguidamente se describen las funciones de las que se encarga la Comisión de Servicio Comunitario.

- Cumplir y hacer cumplir la Ley de Servicio Comunitario de Educación Superior.
- Recibir, evaluar y avalar los proyectos de Servicio Comunitario.
- Analizar y evaluar el impacto en las comunidades, con la prestación del Servicio Comunitario.

- Estudiar y dar respuestas a las solicitudes estudiantiles relacionadas con el Servicio Comunitario.
- Propiciar espacios de encuentro entre la facultad y organismos públicos y privados, comunidades organizadas, gremios y personas naturales o jurídicas interesadas en proyectos de Servicio Comunitario.

### **1.3.2. Unidad de Servicio Comunitario**

La Unidad de Servicio Comunitario cumple diferentes funciones, las cuales se describen a continuación (Acosta, Eleonora 2009).

- Planificar, coordinar y ejecutar actividades dirigidas a estudiantes y profesores de la Facultad de Ciencias de la Universidad Central de Venezuela.
- Coordinar la inscripción en el Servicio Comunitario y la expedición de constancias de aprobación.
- Planificar las acciones para lograr la adecuada ejecución de los proyectos de Servicio Comunitario.
- Planificar inicio, duración, lugar y condiciones para la ejecución de los proyectos de Servicio Comunitario.
- Coordinar el registro, seguimiento y evaluación académica del Servicio Comunitario.
- Mantener actualizada la información de los proyectos de Servicio Comunitario: número, tipo, profesores, alumnos, estado, resultados obtenidos.
- Preparar el anteproyecto de presupuesto anual.
- Gestionar la firma de convenios.



### **1.3.3. Tutores académicos**

Los tutores académicos son profesores de la Universidad Central de Venezuela y, personas que se encuentren preparadas para tal fin (Acosta, Eleonora 2009). Los tutores académicos tienen diferentes deberes y atribuciones, los cuales son:

- Prestar asesoría adecuada y oportuna a sus prestadores de Servicio Comunitario.
- Establecer, con el responsable del proyecto y la comunidad, los detalles del plan de trabajo a ejecutar.
- Remitir informe a la unidad de Servicio Comunitario sobre el desempeño de los prestadores.
- Revisar, evaluar y firmar el informe final presentado por el prestador de Servicio Comunitario.

### **1.3.4. Tutores comunitarios**

Los tutores comunitarios son aquellos encargados de supervisar a los prestadores en la comunidad asignada entre sus deberes están:

- Brindar asesorías a los prestadores de Servicio Comunitario, referente a las funciones que debe cumplir en la comunidad.
- Verificar que los prestadores de Servicio Comunitario cumplan el horario establecido adecuadamente.
- Revisar, firmar y sellar el informe final con sus horas anexas.

### **1.3.5. Prestadores de Servicio Comunitario**

Los prestadores de Servicio Comunitario “son aquellos estudiantes de educación superior que cursan estudios de formación profesional, y deben realizar el Servicio Comunitario aplicando conocimientos científicos, técnicos, culturales, deportivos y humanísticos, adquiridos durante su formación académica en beneficio de la comunidad, para cooperar con su participación al cumplimiento de los fines de bienestar social” (Ley de Servicio Comunitario del estudiante de educación superior, 2005).

Son obligaciones de los prestadores del Servicio Comunitario, según el artículo 18 de la Ley de Servicio Comunitario de la Educación Superior:

- Realizar el Servicio Comunitario como requisito para la obtención del título de educación superior. Dicha labor no sustituirá las prácticas profesionales incluidas en los planes de estudio de las carreras de educación superior.
- Acatar las disposiciones que se establezcan en los convenios realizados por las instituciones de educación superior.
- Actuar con respeto, honestidad y responsabilidad durante el Servicio Comunitario.
- Acatar las directrices y orientaciones impartidas por la coordinación y el asesor del proyecto para el cumplimiento del Servicio Comunitario.
- Cumplir con el Servicio Comunitario según lo establecido en esta Ley y sus reglamentos.
- Cursar y aprobar previa ejecución del Servicio Comunitario, un curso, taller o seminario sobre la realidad de las comunidades.

#### **1.4. Aplicación anterior del Servicio Comunitario**

En el momento que se realizó la presente investigación se utilizaba en la Unidad de Servicio Comunitario una aplicación que efectúa las siguientes funciones:

- Inscripción por parte del estudiante en línea del Proyecto de Servicio Comunitario.
- Genera planilla con los datos del estudiante, tutor comunitario, tutor académico, tipo de proyecto y comunidad.
- Acceso a los prestadores y a la unidad de Servicio Comunitario de los perfiles de los estudiantes.

#### **1.5. Requisitos para la prestación de Servicio Comunitario en la Facultad de Ciencias**

Para la prestación del Servicio Comunitario, la ley de Servicio Comunitario y la Unidad de Servicio Comunitario de la Facultad de Ciencias, establece diferentes requisitos que deben cumplir los estudiantes y los profesores.

##### **1.5.1. Requisitos para los estudiantes**

Según el artículo 16 de la Ley de Servicio Comunitario del Estudiante de Educación Superior, los prestadores de Servicio Comunitario deben cumplir con los siguientes requisitos: (Ley de Servicio Comunitario del Estudiante de Educación Superior, 2005)

- Los estudiantes deben haber cumplido al menos, con el cincuenta por ciento (50%) del total de la carga académica de la carrera.

- Los estudiantes de educación superior, deberán cursar y aprobar previa ejecución del proyecto, un curso, taller o seminario que plantee la realidad de las comunidades.

### **1.5.2. Requisitos para ser tutor académico**

- Debe ser profesor de la Universidad Central de Venezuela o alguna persona que se encuentre preparada para tal fin.
- El representante de la Comisión de Servicio Comunitario de la Facultad de Ciencias por cada una de las licenciaturas, establecen las medidas de selección para que un profesor sea tutor académico.

## **1.6. Procesos**

Actualmente la información de las diferentes actividades que se llevan a cabo en los procesos de curso de inducción y proyecto no llegan de forma oportuna a la comunidad estudiantil, lo cual afecta a los estudiantes, ya que realizan trámites engorrosos, que traen como consecuencia dirigirse repetidas veces a la Unidad de Servicio Comunitario, ocasionando quejas y falta de motivación a los involucrados.

A continuación se describen las actividades a seguir para la prestación del Servicio Comunitario.

### **1.6.1. Inscribir el curso de inducción**

El estudiante debe dirigirse con sus documentos a la Unidad de Servicio Comunitario, en donde verifican su expediente curricular, y comprueban que cumple con los requisitos en cuanto a los créditos aprobados.

De cumplir con los requisitos establecidos, se procede a la formalización de su inscripción en el curso de inducción.

### **1.6.2. Controlar asistencia al curso de inducción**

El estudiante debe asistir durante los días en que se imparten los cursos de inducción, tomando en cada uno de estos la asistencia, la cual es llevada manualmente en carpetas por licenciaturas al momento de entrar el estudiante al auditorio.

### **1.6.3. Aprobar el curso de inducción**

Al finalizar el curso de inducción, el estudiante presenta un examen manualmente en una hoja óptica, rellenando el círculo correspondiente con la respuesta seleccionada, para luego esperar el resultado en días posteriores, lo que genera una pérdida de materiales innecesario como las hojas y de tiempo para el personal encargado de la corrección.

El material necesario que contiene la información de los curso de inducción se encuentra en la página Web del Servicio Comunitario con las presentaciones realizadas y documentación relacionada para su descarga.

### **1.6.4. Seleccionar la Comunidad**

La Unidad de Servicio Comunitario publica en su cartelera física algunas propuestas de comunidades que necesitan prestadores, o bien si el estudiante desea realizarlo en una nueva comunidad que no esté registrada debe hacer una petición de inclusión de comunidad, que será evaluada por la Comisión de Servicio Comunitario.

El estudiante debe acudir a la comunidad, verificar que se encuentre activa y existan cupos para realizar la prestación de Servicio Comunitario.

#### **1.6.5. Inscribir el Proyecto de Servicio Comunitario**

Para la inscripción del Proyecto de Servicio Comunitario, el estudiante debe aprobar previamente el curso de inducción y luego registrarse en línea a través de la aplicación que se encuentra actualmente. Después debe imprimir una constancia generada al introducir sus datos y que el tutor académico lo firme, para luego entregarlo en la Unidad de Servicio Comunitario, en donde lo sellan y crean una nueva celda en el archivo de Excel en el cual se encuentran todos los prestadores comunitarios con sus datos registrados manualmente.

La transcripción repetida de los mismos datos es propensa a errores, lo que genera en muchos casos inconsistencia de datos y lleva consigo una tabla de Excel difícil de entender, que dificulta y retrasa la consulta de algún dato.

#### **1.6.6. Cambiar de comunidad**

En el caso de que un prestador de Servicio Comunitario quiera cambiar de comunidad, debe completar una planilla de solicitud estudiantil en la Unidad de Servicio Comunitario explicando las razones del cambio que será aprobado por la Jefa de la Unidad de Servicio Comunitario.

Si un estudiante acumuló horas en la comunidad anterior, redacta un oficio explicando los motivos del cambio a la Comisión de Servicio Comunitario, la cual avalara las horas cumplidas y procesará el cambio de la comunidad.

### **1.6.7. Cambiar de tutor académico**

Dependiendo del tutor académico asignado, el responsable de su licenciatura en la Comisión de Servicio Comunitario estudia el caso, que pasa al Consejo de Escuela y aprueba el cambio o no.

### **1.6.8. Controlar el Proyecto de Servicio Comunitario**

El control del Proyecto de Servicio Comunitario en cuanto a las horas que debe cumplir, lo realizan manualmente los prestadores de Servicio Comunitario, en un formato descargado de la página Web del Servicio Comunitario. El tutor comunitario debe firmar continuamente cada una de las casillas del control de asistencia correspondientes a las horas y sellarla, al igual que el tutor académico.

Las actividades descritas dependen del microproyecto en donde se esté realizando la práctica de Servicio Comunitario.

### **1.6.9. Entregar documentos finales del Proyecto de Servicio Comunitario**

Para culminar la práctica comunitaria el prestador debe cumplir con el artículo número 8 de la ley de Servicio Comunitario del estudiante de educación superior que establece “El Servicio Comunitario tendrá una duración mínima de ciento veinte horas académicas, las cuales se deben cumplir en un lapso no menor de tres meses. Las instituciones de educación superior adaptarán la duración del Servicio Comunitario a su régimen académico” (Ley de Servicio Comunitario del Estudiante de Educación Superior, 2005).

Una vez que el estudiante ha cumplido con las horas de prestación de Servicio Comunitario, debe hacer un informe final en donde se expliquen las actividades cumplidas en la comunidad, sus experiencias vividas y los resultados

obtenidos. Adicionalmente tiene que anexar el control de asistencia avalada por el tutor académico y tutor comunitario. Finalmente elaborar una presentación que plasme un resumen del informe final y algún contenido extra que el estudiante desee agregar.

Durante la fecha de entrega de los documentos finales se generan esperas en la Unidad de Servicio Comunitario por parte de los estudiante, ya que se revisa uno por uno los siguientes requisitos:

- El informe final debe tener las firmas de los tutores (académico y comunitario) y del prestador de Servicio Comunitario con el sello de la comunidad.
- Resumen de Actividades y Control de Asistencia de la prestación del Servicio Comunitario con firma de los tutores (académico y comunitario), del prestador de Servicio Comunitario y el sello de la comunidad.
- Un CD (el CD debe contener: el micro proyecto, el informe final, la presentación y cualquier material elaborado por el prestador durante su práctica comunitaria debe estar identificado con: el nombre del estudiante, la licenciatura, el título del informe y comunidad). Cuando el CD es entregado se archiva y van ocupando mucho espacio.

Después de la revisión de todos los documentos se le asigna al estudiante una fecha para la presentación pública.

El día de la presentación los jurados están conformados por el tutor académico del estudiante y dos docentes.

#### **1.6.10. Presentar el proyecto de Servicio Comunitario**

Para la presentación del proyecto de Servicio Comunitario la Unidad de Servicio Comunitario, carga la presentación entregada previamente por el



estudiante, que será proyectada ante la audiencia y evaluada, en caso obtener una evaluación satisfactoria, el prestador habrá aprobado el Servicio Comunitario. La evaluación se lleva manualmente por cada uno de los jurados en un formato entregado por la Unidad de Servicio Comunitario.

#### **1.6.11. Calificar el Servicio Comunitario**

Para calificar la presentación del Servicio Comunitario, que puede ser aprobado (A) o aplazado (AP), la Unidad de Servicio Comunitario envía una lista con los estudiantes y su calificación obtenida a la División de Control de Estudios.

Si de un listado de estudiantes alguno de ellos no tiene su calificación, el envío se retrasa, lo que en algunos casos impide que se haga la colocación de la nota en el expediente curricular de los demás estudiantes.

En la Figura 1 se puede observar gráficamente los procesos a seguir para la realización del Servicio Comunitario.



*Figura 1- Flujo a seguir por un estudiante para realizar el Servicio Comunitario*

## **CAPÍTULO II - Tecnologías**

La finalidad de este capítulo es presentar las bases conceptuales de las tecnologías utilizadas durante el desarrollo de la aplicación del Servicio Comunitario.

### **2.1. Aplicaciones Web**

Para automatizar los procesos relacionados a los cursos de inducción y proyectos de Servicio Comunitario se desarrollo una aplicación Web, que se define según Shklar y Rosen como, “Es una aplicación cliente-servidor que usa un navegador Web como su cliente y presta servicios ab través de Servidores Web distribuidos en Internet o una red privada (Intranet). Además puede presentar contenido dinámico basado en los parámetros de la solicitud. Frecuentemente las aplicaciones Web tienen aspectos de seguridad, interacción con los usuarios y pueden almacenar y actualizar información con controles de acceso y soporte de transacciones en línea” (Shklar & Rosen, 2009).

La ventaja de poder acceder a este tipo de aplicaciones vía Web que se encuentran alojadas en un servidor Web, permite que un gran número de usuarios puedan hacer uso de éstas, sin importar el momento o la ubicación en donde se encuentren.

### **2.2. Cliente Servidor**

La arquitectura que se aplicó para el desarrollo de la aplicación fue la de Cliente Servidor, que se define como una arquitectura distribuida que permite a los usuarios finales obtener acceso a la información en forma transparente aún en entornos multiplataforma. En la arquitectura cliente servidor Figura 2, el cliente

envía una petición solicitando un determinado servicio, y este envía una o varias respuestas (Shklar & Rosen, 2009).



*Figura 2 - Cliente Servidor*

- **Cliente**

“El cliente Web es un programa con el que interacciona el usuario para solicitar a un servidor Web el envío de los recursos que desea obtener mediante el protocolo HTTP” (Lujan, Sergio 2001).

Es decir el cliente es el encargado de iniciar la comunicación a través de la red con el servidor, emitiendo solicitudes de recursos o peticiones de servicios, los cuales pueden ser la consulta de una información en particular, un archivo, entre otros, y están ubicados en diferentes estaciones de trabajo.

- **Servidor**

“Los servidores Web son computadoras que ofrecen sus servicios a los clientes. Suelen tener una presencia estable en la red, lo que se concreta en tener

asignadas direcciones IP permanentes. En ellos es donde están alojadas, por ejemplo, las aplicaciones Web” (Cobo, Ángel 2005).

Los servidores Web se encargan de atender a las solicitudes realizadas por los clientes para responder con el recurso o la información solicitada.

### **2.3. Tecnologías del lado del cliente**

Las tecnologías del lado del cliente son todos aquellos lenguajes o formatos que son ejecutados por el navegador Web y que no necesitan intervención alguna de ningún servidor para cumplir sus funciones. Adicionalmente, cualquier modificación en alguna de estas tecnologías no implica grandes cambios por parte del servidor, ya que éste es independiente de las mismas por ser ejecutadas en el navegador Web del usuario final.

#### **2.3.1. Hojas de Estilo en Cascada (CSS)**

El aspecto visual de las interfaces de usuario de la aplicación se manejan con las hojas de estilo en cascada (Cascading Style Sheets), que es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos.

CSS se utiliza para dar estilo a documentos HTML y XML, separando el contenido de la presentación. Los *Estilos* definen la forma de mostrar los elementos HTML y XML. CSS permite a los desarrolladores Web controlar el estilo y el formato de múltiples páginas Web al mismo tiempo. Cualquier cambio en el

estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento (W3C).

### **2.3.2. JavaScript**

Es un lenguaje de programación interpretado, es decir, que no requiere compilación, muy utilizado en páginas Web, con una sintaxis semejante a la de Java y C.

Se ejecuta en el cliente al mismo tiempo que las sentencias HTML. En la aplicación del Servicio Comunitario algunos de los eventos y comportamientos de los elementos HTML fueron manejados con librerías JavaScript.

## **2.4. Tecnologías del lado del servidor**

Una página Web, es un documento adaptado para la Web y que normalmente forma parte de un sitio Web. Su principal característica son los hiperenlaces a otras páginas, siendo esto el fundamento de la Web. Una página Web está compuesta por información texto o multimedia e hiperenlaces, además puede contener o asociar datos de estilo para especificar cómo debe visualizarse, o aplicaciones embebidas para hacerla interactiva.

El contenido de la página puede ser predeterminado lo que se conoce como "página Web estática" se refiere a la solicitud de un recurso y este es devuelto por el servidor HTTP. El recurso es un documento en el sistema de archivos del servidor, el cual se transforma y despliega en el cliente.

## **2.5. Sistemas de Gestión de Bases de Datos**

Existen diversas formas de almacenar información dentro de una base de datos, pero igualmente, se necesita poder manipular esta información, acceder a ella y poder actualizarla de manera sencilla. Es aquí donde entran los denominados Sistemas de Gestión de Bases de Datos, los cuales son un conjunto de programas que manejan todo el acceso a la BD y ejecutan las transacciones que vayan a realizarse sobre la misma, para generar la información respectiva. Verifica también, que estas transacciones sean ejecutadas de forma correcta y brinda una interfaz entre la BD, el usuario y las aplicaciones que las utilizan (MySQL).

## **2.6. Sistema de Gestión de Base de datos MySQL**

Debido a los beneficios que brinda el Sistema de Gestión de base de datos MySQL, además de ser utilizado en las aplicaciones de la Facultad de Ciencias, se explica de forma general a continuación.

Es el Sistema de Gestión de base de datos relacional de código abierto más utilizado en la actualidad. El software MySQL utiliza una licencia pública general (GPL), para definir lo que se puede y no se puede hacer con el software en diferentes situaciones.

MySQL Server se desarrolló originalmente para tratar grandes bases de datos mucho más rápido que soluciones existentes, y ha sido utilizado con éxito en entornos de producción de alto rendimiento durante varios años. MySQL ofrece hoy en día una gran cantidad de funciones. Su conectividad, velocidad, seguridad y licencia pública hacen de MySQL altamente apropiado para acceder bases de datos en Internet y el SGBD más conveniente para el desarrollo de aplicaciones en las que se buscan soluciones rápidas y de bajos costos.

Por otra parte MySQL se basa en la tecnología cliente/servidor que consiste en un, servidor SQL multihilo que trabaja con diferentes programas, clientes, herramientas administrativas y un amplio abanico de interfaces de programación para permitir a muchos lenguajes acceder a las bases de datos MySQL sin ningún problema, ya que para cada uno de estos existe una API específica (MySQL).

## **2.7. Ruby On Rails**

Para el desarrollo de la aplicación del Servicio Comunitario se tomó Ruby on Rails, que es un framework de aplicaciones Web de código abierto escrito en el lenguaje de programación Ruby, siguiendo el patrón Modelo, Vista, Controlador (MVC). Trata de combinar la simplicidad con la posibilidad de desarrollar aplicaciones del mundo real escribiendo menos código y con un mínimo de configuración.

### **2.7.1. Ruby**

Michael Fitzgerald menciona que, “Ruby es un lenguaje de programación orientado a objetos creado por primera vez en Japón en el año 1995 por Yukihiro Matz Matsumoto. Ruby fue adquirido y aceptado mundialmente como un lenguaje fácil de aprender, poderoso e interpretado” (Fitzgerald, Michael 2007).

“Su creador Yukihiro Matz Matsumoto mezcló partes de sus lenguajes favoritos (Perl, Smalltalk, Eiffel, Ada, y Lisp) para formar un nuevo lenguaje que incorporara tanto la programación funcional como la programación imperativa” (Ruby, Ruby Lang).



Ya que la implementación oficial de Ruby es distribuida bajo una licencia de software libre, éste fue uno de los aspectos que se tomo en cuenta para la selección del lenguaje. Además de ofrecer una curva de aprendizaje fácil y rápida.

### **2.7.2. Rails**

Rails según Bruce Tate, Curt Hibbs lo exponen como el “framework más productivo de desarrollo Web de todos los tiempos” (Tate, Bruce 2006). Está basado en el patrón Modelo Vista Controlador.

Fue creado por David Heinemeier Hansson y publicado en Julio del 2004 con una licencia de código abierto.

## **2.8. Patrón Modelo Vista Controlador en Rails**

El patrón Modelo Vista Controlador “es uno de los modelos de infraestructura sugerido para aplicaciones Web el cual permite desacoplar la interfaz de usuario de la funcionalidad y de contenido de información” (Pressman, 2007).

### **2.8.1. Modelo**

El modelo es el responsable de mantener el estado de la aplicación. Algunas veces este estado trasciende y dura sólo un par de interacciones con el usuario. Algunas veces el estado es permanente y será almacenado afuera de la aplicación, algunas veces en una base de datos.

El modelo es más que solo una data, éste aplica todas las reglas de negocios.

### 2.8.2. Vista

La vista contiene todas las funcionalidades específicas de la interfaz y habilita la presentación del contenido y la lógica de procesamiento.

### 2.8.3. Controlador

El controlador dirige la aplicación. El controlador recibe eventos de afuera, es decir entradas normales del usuario, que interactúan con el modelo y luego son mostradas a través de una vista apropiada al usuario.

Se puede ver la arquitectura modelo, vista y controlador en la Figura 3.

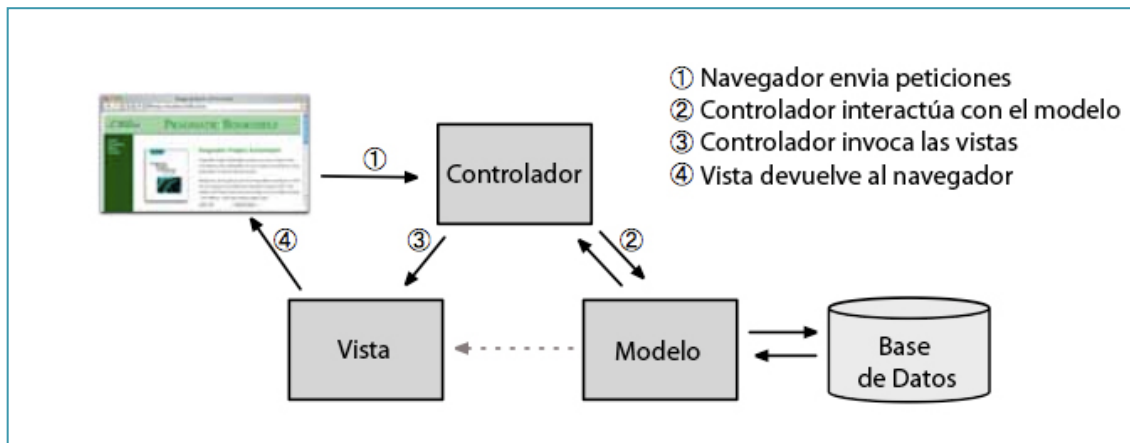


Figura 3 - Patrón Modelo Vista Controlador.

## 2.9. Filosofía de Ruby On Rails

Ruby on Rails está basado en dos principios de desarrollo, los cuales son convención sobre configuración y no te repitas, que se explicarán a continuación.

### **2.9.1. Convención sobre configuración**

La convención sobre la configuración es una programación diseñada que favorece un conjunto de convenciones de programación en vez de configurar un framework de aplicación. Ciertas configuraciones usadas comúnmente (por convención y no por reglas) están preestablecidas y el framework trabaja siguiendo esas convenciones. Por ejemplo en Ruby on Rails, la convención establece que el controlador para el modelo del objeto Book será llamado BookController y todas las vistas relacionadas con el controlador seguirán contenida en una carpeta llamada book.

### **2.9.2. No te repitas**

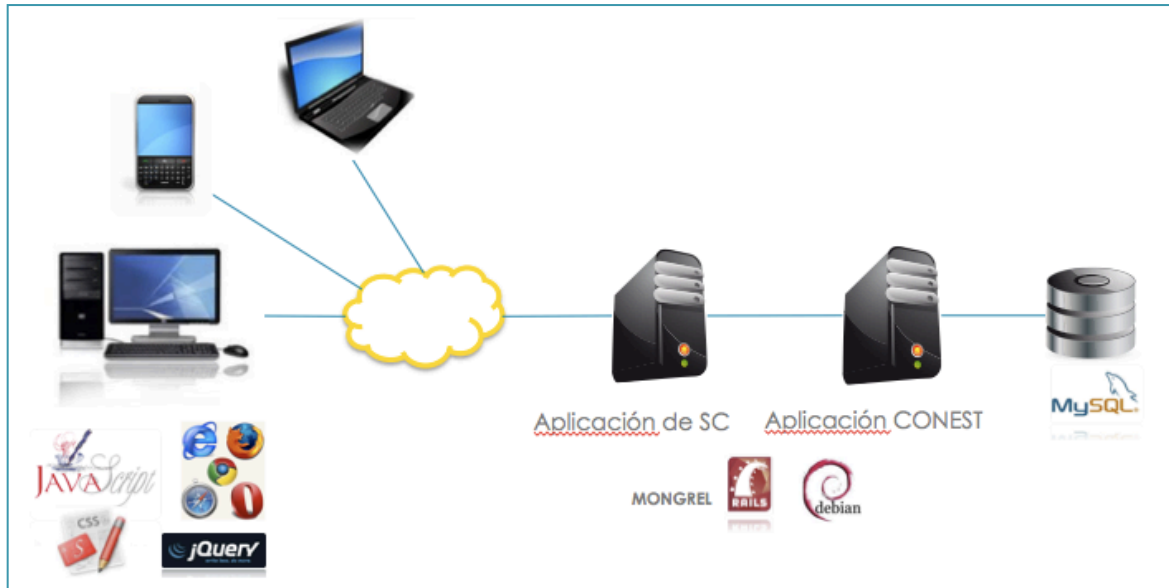
No te repitas es un principio que se enfoca en reducir la duplicidad de la información manteniendo una parte del conocimiento en el sistema solamente en un lugar.

## **2.10. Plataforma**

Debido a que la aplicación de Servicio Comunitario maneja mucha información relacionada a los estudiantes y docentes de la Facultad de Ciencias, fue conveniente hacer una integración con el Sistema de Gestión Académica CONEST, para acceder a su base de datos en la que residía toda la información necesaria.

La integración facilitó agregar las diferentes tablas relacionadas al Servicio Comunitario y mantener la información actualizada de los estudiantes y docente. Otra ventaja fue relacionada a las consultas, ya que como se tenía una conexión directa con la base de datos del Sistema de Gestión Académica CONEST las consultas frecuentes, se realizaban más rápido.

En la Figura 4 se puede observar la integración realizada.



*Figura 4 - Plataforma de la aplicación de Servicio Comunitario.*

Un factor importante para seleccionar las tecnologías explicadas en este capítulo, fue la integración con el Sistema de Gestión Académica CONEST.

## **CAPÍTULO III - Marco Aplicativo**

En este capítulo se describirá el método de desarrollo de software, y la adaptación durante la elaboración de la aplicación del Servicio Comunitario, incluyendo las iteraciones aplicando las actividades de la Programación Extrema.

### **3.1. Metodología de Desarrollo de Software**

Según Kenneth Laudon una metodología de desarrollo es “un conjunto de métodos, uno o más para cada actividad dentro de cada fase de un proyecto de desarrollo de sistemas. La función primaria de una metodología de desarrollo es proporcionar una disciplina para todo el proceso de desarrollo” (Kenneth, Laudon 2004). Ellos mencionan que, una buena metodología de desarrollo puede establecer la disciplina necesaria para el desarrollo de un software sólo si establece estándares para los requerimientos, equipos de trabajo, diseño, programación y pruebas.

También afirman que para lograr un desarrollo de software de alta calidad, las organizaciones deben seleccionar un método apropiado que se adapte al proyecto y luego ponerlo en producción. La metodología se basa en disciplina y una de éstas es llevar el control del proyecto desde sus comienzos, por lo que, es necesario que los documentos de requerimientos y especificaciones de los sistemas estén completamente detallados, exactos y completos siguiendo un protocolo entre las comunidades para una mejor comunicación.

### **3.2. Metodologías Ágiles**

Las metodologías ágiles son adaptables a las situaciones de los proyectos, “pero una adaptación continua sin un progreso logra muy poco. Por lo tanto, un proceso ágil de software debe adaptarse de forma incremental” (Pressman, Roger

2006). Para que la metodología se pueda adaptar debe existir una retroalimentación con el cliente a fin de lograr todos los requerimientos que el mismo desea en el tiempo necesario.

La metodología ágil tiene algunos principios como lo son:

- Los individuos y sus interacciones son más importantes que los procesos y las herramientas.
- El software que funciona es más importante que la documentación exhaustiva.
- La respuesta y soluciones del proyecto antes que seguir un plan específico y cerrado.

A continuación se describe el método ágil Programación Extrema XP.

### **3.3. Programación Extrema**

“El nombre de Programación Extrema fue acuñado por Beck (Beck, 2000) debido a que todos los requerimientos se expresan como escenarios llamados historias de usuario, los cuales se implementan directamente como una serie de tareas. Los programadores trabajan en parejas y desarrollan pruebas para cada tarea antes de escribir el código. Todas las pruebas se debe ejecutar satisfactoriamente cuando el código nuevo se integre al sistema” (Sommerville, Ian 2005).

Según (Kendall, Kenneth 2006), “La administración de proyectos es importante, de tal manera que la Programación Extrema intenta definir rápidamente un plan global del sistema, desarrollar y liberar rápidamente el software y posteriormente revisarlo continuamente para incorporarle características adicionales”.

Es decir, la Programación Extrema da una planificación general en la que se lleva el proyecto al extremo en varios aspectos, pero al final se realizan diferentes pruebas para saber de qué carece el mismo o qué modificaciones necesita y hacerla, para entregar un software completo y de calidad.

### 3.4. Actividades de la Programación Extrema

Según (Pressman, Roger 2006), la Programación Extrema “abarca un conjunto de reglas y prácticas que ocurren en el contexto de cuatro actividades del marco de trabajo: planeación, diseño, codificación y pruebas. En la Figura 5 se ilustra el proceso de la Programación Extrema y se observan algunas de las ideas y tareas claves asociadas con cada actividad del marco de trabajo”

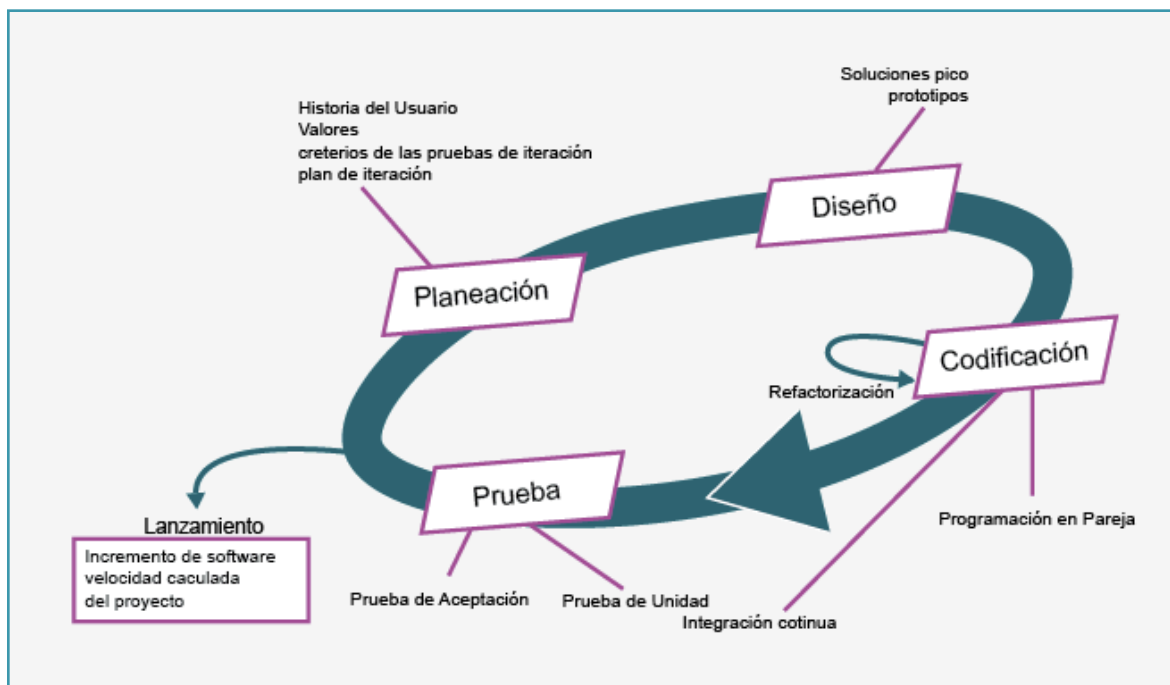


Figura 5 - Proceso de la Programación Extrema.

A continuación se describen las actividades de la Programación Extrema utilizadas durante las diferentes iteraciones del desarrollo de la aplicación del Servicio Comunitario.

### **3.4.1. Planeación**

La actividad de planeación comienza creando una serie de historias de usuarios, las cuales describen las características y las funcionalidades requeridas para el software que se desarrollará. Las historias de usuario son similares a los casos de uso, y son los requisitos que quieren los clientes y la prioridad que tienen cada uno de ellos. Luego, el equipo de trabajo evalúa cada historia y le asigna un costo, el cual se mide en semanas de desarrollo. Si la historia requiere más de tres semanas de desarrollo, se le pide al cliente que la divida en historias menores, y se realiza de nuevo la asignación del valor y el costo. Las historias de usuario nuevas se pueden incorporar cuando se requiera.

Los clientes y el equipo de Programación Extrema trabajan juntos para decidir cómo agrupar las historias para el próximo incremento de software, para que el equipo de trabajo las desarrolle.

Después de que se ha entregado el primer incremento de software, el equipo de trabajo calcula la velocidad del proyecto, de esta forma se pueden estimar fechas de entrega y el programa para lanzamientos subsecuentes y también determinar si se ha hecho un compromiso excesivo en algunas de las historias de todo el proyecto de desarrollo. Si se presenta un compromiso excesivo, el contenido de los lanzamientos se modifica o se cambian las fechas de las entregas finales.

A medida que el avanza el trabajo de desarrollo del software, el cliente puede agregar historias, cambiar el valor de la historia existente, dividir historias o



eliminarlas. Así, el equipo de desarrollo considera los lanzamientos restantes y modifica sus planes de acuerdo con ello.

### **3.4.2. Diseño**

El diseño de la Programación Extrema sigue de manera rigurosa el principio de mantenerlo simple. Siempre se prefiere un diseño simple respecto de una presentación más compleja. Además, el diseño ofrece una guía de implementación para una historia como está escrita.

La Programación Extrema apoya la refactorización, es una técnica de construcción y de diseño. (Fowler, Martin 2000) describe la refactorización de la siguiente forma:

“Refactorización es el proceso de cambiar un sistema de software de tal manera que no altere el comportamiento externo del código y que mejore la estructura interna. Es una manera disciplinada de limpiar el código y modificación o simplificar el diseño interno, lo que minimiza las oportunidades de introducir errores. En esencia, al refactorizar, se mejora el diseño del código después de que se ha escrito”.

El diseño se considera como un artefacto que puede y debe modificarse de manera continua a medida que prosigue la construcción. El propósito de refactorizar es controlar estas modificaciones al sugerir pequeños cambios del diseño. Sin embargo, debe notarse que el esfuerzo requerido para refactorizar puede aumentar en forma drástica a medida que crece el tamaño de la aplicación.

El diseño ocurre antes y después de comenzar con la codificación. Refactorizar significa que el diseño ocurre de manera continua a medida que se construye el sistema.

### **3.4.3. Codificación**

La Programación Extrema recomienda que después de diseñar las historias y realizar el trabajo de diseño preliminar el equipo no debe iniciar la codificación, sino que debe desarrollar una serie de pruebas unitarias que ejerciten cada una de las historias que vayan a incluirse en el lanzamiento actual. Una vez creada la prueba de unidad, el desarrollador es más capaz de centrarse en lo que debe implementarse para pasar la prueba unitaria. Una vez que el código está completo, la unidad puede probarse de inmediato, y así proporcionar una retroalimentación instantánea a los desarrolladores.

Un concepto clave durante la actividad de codificación es la programación en pareja. La Programación Extrema recomienda que dos personas trabajen juntas en una estación de trabajo de computadora para crear el código de una historia. Esto proporciona un mecanismo para la resolución de problemas en tiempo real y el aseguramiento de la calidad en las mismas condiciones. También alienta que los desarrolladores se mantengan centrados en el problema que se tiene a la mano. En la práctica, cada persona tiene un papel sutilmente diferente.

### **3.4.4. Pruebas**

Las pruebas son un conjunto de actividades que se planean con anticipación y se realizan de manera sistemática. La Programación Extrema como se observa en la Figura 5 utiliza las pruebas de unidad y las pruebas de aceptación.

Las pruebas unitarias se concentran en el esfuerzo de verificación de la unidad más pequeña del diseño del software: el componente o módulo de software. Tomando como guía la descripción del diseño al nivel de componentes, se prueban importantes caminos de control para descubrir errores dentro de los límites del módulo. El alcance restringido que se ha determinado para las pruebas

de unidad limita la relativa complejidad de las pruebas y los errores que éstas descubren. Las pruebas unitarias se concentran en la lógica del procesamiento interno y en las estructuras de datos dentro de los límites de un componente. Este tipo de pruebas se puede aplicar en paralelo a varios componentes.

Pruebas de aceptación son aquellas pruebas que especifica el cliente y se enfocan en las características generales y la funcionalidad del sistema, elementos visibles y revisables por el cliente. Las pruebas de aceptación se derivan de las historias de usuario que se han implementado cm parte de un lanzamiento de software.

Un elemento clave para el enfoque de la Programación Extrema es la creación de una prueba de unidad antes de comenzar la codificación. Las pruebas de unidad que se crean deben implementarse con un marco de trabajo que permita automatizarlas.

### **3.5 Aspectos positivos de la Programación Extrema**

Las pruebas unitarias en el código es una práctica generalmente alentada y reconocida como un factor clave para obtener un software de alta calidad, si a eso se le agrega la exigencia de la Programación Extrema de que se hagan constantemente, durante cada etapa de codificación, quizás en el peor de los casos pudiera parecer un exceso. De manera semejante, la integración continua es aceptada y recomendada para evitar catástrofes ocasionadas por defectos no detectados a tiempo (Glass, Robert 2001).

El énfasis en la simplicidad y la refactorización es encontrado como un factor saludable en la práctica de programación, ya que normalmente se asocia el exceso de código con una lógica deficiente, un diseño innecesariamente complejo, problemas para el mantenimiento del sistema y un centro para encontrar defectos

que demeritan la calidad del producto. También se puede ver como positivo el hecho de que, filosóficamente, la Programación Extrema tiene un enfoque extremadamente humano, siendo este un aspecto que el resto del campo del software debería tratar de emular.

Desde el lado del programador, como ejemplos de este enfoque humano, tenemos la premisa de la semana de 40 horas, el alto valor que se le da a la comunicación y el rol protagónico que toma el programador en la etapa de planeación. Por el lado del cliente también se percibe el enfoque humano, ya que tenemos su presencia constante en las instalaciones del desarrollador, el diálogo que se fomenta entre el cliente y el resto del equipo de programación, la declaración de que escuchar al cliente es una de las cuatro actividades esenciales de la programación y el hecho de que se le otorga el mayor peso en la planeación.

### **3.6. Adaptación de la Programación Extrema**

Para el desarrollo de la aplicación Web que automatiza los procesos de la Unidad de Servicio Comunitario de la Facultad de Ciencias, se implementaron las actividades del método de desarrollo de software Programación Extrema explicados anteriormente.

El desarrollo de software se divide en diferentes partes denominadas iteraciones, las cuales se llevan a cabo en dos o tres semanas dependiendo del caso. De no culminarse alguna iteración en el tiempo correspondiente, se debe continuar en la siguiente hasta finalizarla.

Empleando las actividades propuestas por el método de desarrollo de software a la aplicación del Servicio Comunitario, se elaboraron ocho iteraciones descritas en este capítulo.

Como propuesta de la actividad de planeación, se utilizaron las historias de usuario con el formato que se puede observar en la Tabla 1, el cual tiene la fecha inicial, fecha final, requerimiento y el estado (Finalizado, Pendiente, Activo) del requerimiento para cada una de las iteraciones.

*Tabla 1 - Formato historias de Usuario.*

Nro.	Fecha Inicio	Fecha Final	Requerimientos	Estado

En cuanto al diseño se utilizan distintos modelos y prototipos de interfaz como base del desarrollo de los procesos del software. Por otra parte, para la codificación se explica brevemente una parte del código que se realizó en cada una de las iteraciones.

Para la actividad de pruebas se aplican pruebas funcionales o de aceptación según sea el caso.

A continuación se describen las iteraciones cumplidas con sus diferentes actividades propuestas por el método de desarrollo de software Programación Extrema.

### **3.7. Iteración 0**

La iteración 0 tiene como requerimiento principal presentarle al cliente la interfaz con la que podrán interactuar los diferentes usuarios de la aplicación del Servicio Comunitario de la Facultad de Ciencias.

- **Planeación**

En la Tabla 2 se puede observar en detalle la historia de usuario de la iteración cero.

*Tabla 2 - Historia de Usuario para la iteración cero.*

Nro.	Fecha Inicio	Fecha Final	Requerimientos	Estado
0	18/04/2010	22/04/2010	Realizar una propuesta de interfaz para los usuarios.	Finalizada

- **Diseño**

Para el diseño de la iteración actual se creó un modelo de casos de uso para el levantamiento de información inicial (Tópicos para el desarrollo de una aplicación Web que automatice los procesos de la Unidad de Servicio Comunitario de la Facultad de Ciencias, 2009).

### **Casos de Uso**

A pesar de que los casos de uso no son parte del método de desarrollo de software Programación Extrema, se utilizan para realizar el primer levantamiento de información y elaborar una propuesta del prototipo de interfaz (Ver Figura 6), para luego crear las Historias de Usuario que se observa en la Tabla 3, cabe destacar que a medida que se desarrolló la aplicación las historias de usuario pueden variar.

En los casos de uso muestra los actores que intervienen, tales como prestador de Servicio Comunitario, Unidad de Servicio Comunitario, Comisión de Servicio Comunitario, tutor académico e internautas, y la propuesta de las

funciones principales de la aplicación (Tópicos para el desarrollo de una aplicación Web que automatice los procesos de la Unidad de Servicio Comunitario de la Facultad de Ciencias, 2009).

*Tabla 3 - Historias de Usuario (Versión Inicial).*

Nro.	Requerimientos
0	Realizar una propuesta de interfaz para los usuarios.
1	Diseñar el modelo de base de datos.
2	Crear modelos, vistas y controladores.
3	Autenticar a los usuarios según su rol: estudiante, docente y administrador.
4	Validar los campos de los distintos formularios de la aplicación.
5	Crear interfaces donde se muestre de forma clara la información a los distintos usuarios.
6	Brindar la opción de generar reportes de distintos tipos concerniente al desarrollo de todos los procesos del Servicio Comunitario.
7	Realizar los distintos servicios Web que serán utilizados para efectuar la comunicación con otras aplicaciones.
8	Proveer las facilidades para realizar actualizaciones continuas en cuanto a las distintas informaciones de la aplicación.
9	Hacer uso de distintas herramientas que le permitan al usuario completar de manera más sencilla campos como: fechas, horas, entre otros.
10	Mantener restringida la información de los usuarios y crear perfiles que permitan distinguir a cada uno de ellos.
11	Crear la lógica y estructura necesaria para la implementación de la prueba en línea de los cursos de inducción para los estudiantes.
12	Crear las sesiones de usuario para brindar seguridad a la aplicación.
13	Proveer donde aplique, la comunicación entre distintos usuarios a través de la aplicación.

14	Brindar al prestador comunitario la funcionalidad de cargar sus informes, presentaciones y cualquier tipo de información que sea necesaria para la unidad de Servicio Comunitario, haciendo uso de la aplicación.
15	Brindar a cualquier usuario que no esté registrado en la aplicación la posibilidad de consultar información de su interés como fechas importantes, cursos, procesos y requisitos, entre otros.

### Prototipo de Interfaz

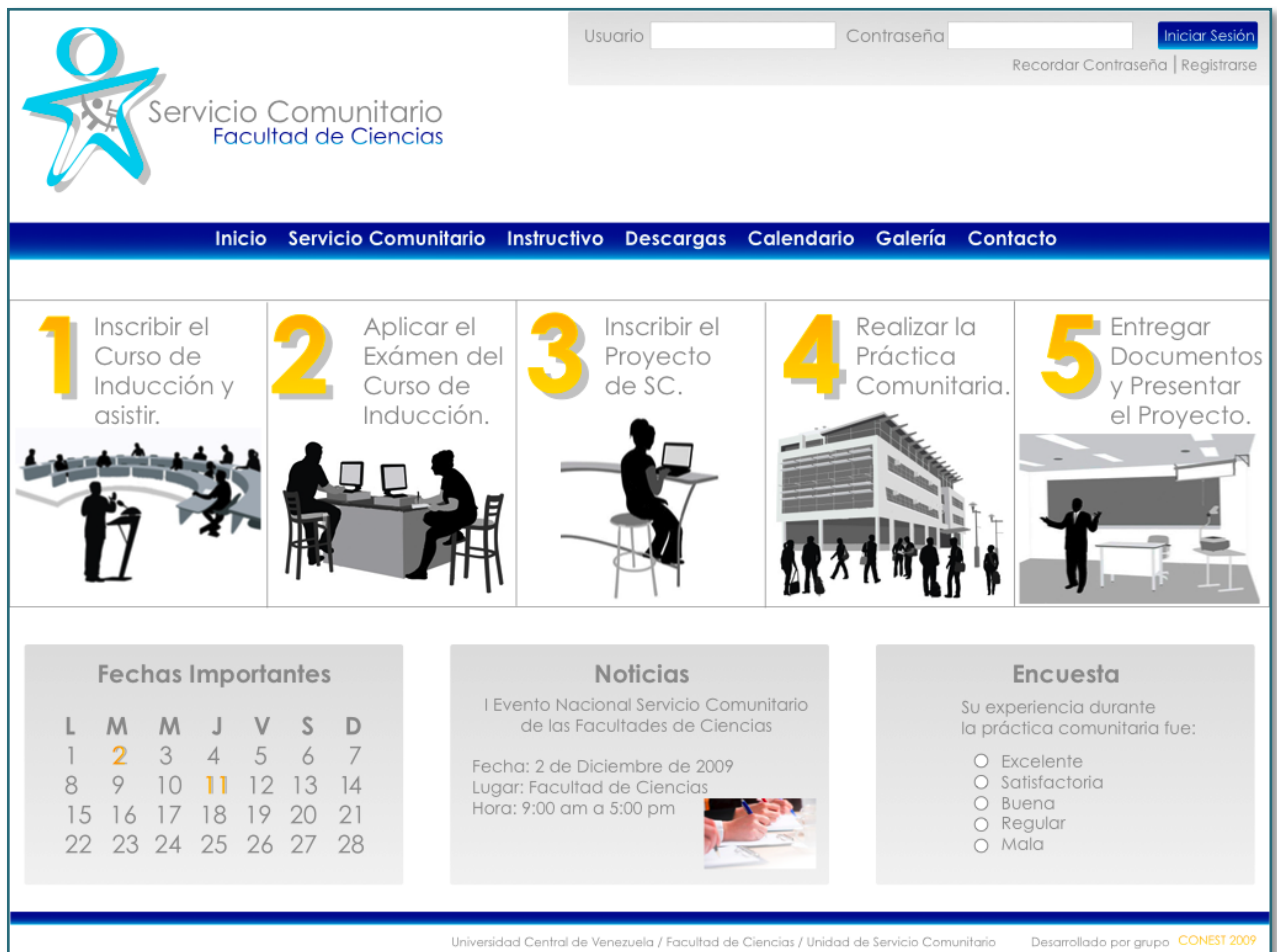


Figura 6 - Propuesta del Prototipo de Interfaz de Usuario.



- **Pruebas**

Se realizaron pruebas de aceptación el día 8 de Diciembre de 2009 durante la presentación del seminario “Tópicos para el Desarrollo de una Aplicación Web para la automatización de la Unidad de Servicio Comunitario de la Facultad de Ciencias”.

El prototipo fue aprobado por la Profa. Inés Tovar, Jefa de la Unidad de Servicio Comunitario y la Lic. Yadira Salcedo trabajadora social de la Unidad de Servicio Comunitario, sugiriendo algunos cambios en la gama colores utilizada.

### **3.8. Iteración 1**

Durante la iteración 1 se realizó el primer modelo y creación de la base de datos de la aplicación, que se incrementará a medida que se realicen nuevas funcionalidades, añadiendo tablas y registros dependiendo del caso que se presente. También, siguiendo el patrón Modelo Vista Controlador que propone Ruby on Rails, se crean los modelos, controladores y vistas según cada usuario de la aplicación.

La iteración 1 tiene otro objetivo crear la autenticación de los siguientes usuarios para brindarle acceso a la aplicación:

- Estudiante de la Facultad de Ciencias.
- Docente de la Facultad de Ciencias.
- Comisión del Servicio Comunitario de la Facultad de Ciencias.
- Administrador de la Unidad de Servicio Comunitario de la Facultad de Ciencias.

Los usuarios podrán acceder a la aplicación con su número de cédula como usuario, y su contraseña es la misma utilizada para ingresar en el sistema de gestión académica CONEST.

En el caso de que un estudiante no recuerde su contraseña debe dirigirse a la División de Control de Estudios para generar una nueva, que será enviada a su correo electrónico.

- **Planeación**

En la Tabla 4 se puede observar en detalle las historias de usuario de la iteración 1.

*Tabla 4 - Historia de Usuario para la iteración uno.*

Nro.	Fecha Inicio	Fecha Final	Requerimientos	Estado
1	25/04/2010	26/04/2010	Diseñar el modelo de base de datos.	Finalizada
2	27/04/2010	28/04/2010	Crear modelos, vistas y controladores.	Finalizada
3	01/05/2010	02/05/2010	Crear la distribución de la interfaz de los diferentes usuarios.	Finalizada
4	03/05/2010	05/05/2010	Autenticar a los usuarios según su rol: comisión, estudiante, docente y administrador.	Finalizada

- **Diseño**

Como parte del diseño de la iteración 1 se realiza el modelo de la base de datos para representar la lógica de los datos y sus relaciones.

## Modelo de base de datos

En la Figura 7 se puede observar parte del modelo de la base de datos inicial de la aplicación, en la que se reflejan las tablas de *curso\_induccion* y *estudiante\_curso\_induccion*. En ellas se almacena la información correspondiente al curso de inducción del estudiante que se inscribe en él.

A fin de poder realizar el modelo de base de datos se aplican los estándares del Sistema de Gestión Académica CONEST, a fin de crear futuras vinculaciones.

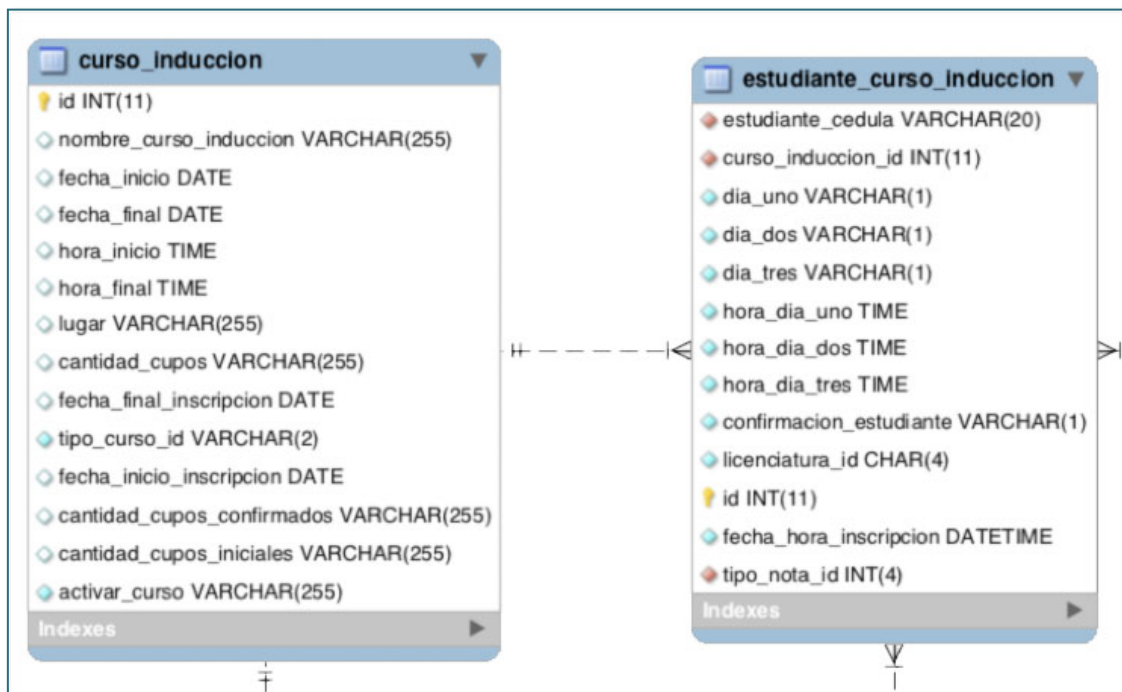


Figura 7 - Parte modelo de Base de Dato Inicial.

- **Codificación**

Llevando a cabo los objetivos propuestos en la iteración 1, se creó una base de datos, que permite el acceso a la información de los diferentes usuarios y

procesos involucrados con los cursos de inducción y proyectos del Servicio Comunitario.

Algunas de las tablas que se crearon en esta iteración fueron la de los cursos de inducción (*curso\_induccion*), en donde se almacena toda la información referente ellos. El curso de inducción lo deben realizar los estudiantes antes de comenzar su prestación de Servicio Comunitario, por lo tanto la tabla (*estudiante\_curso\_induccion*) se encuentra relacionada a la tabla (*curso\_induccion*).

Asimismo, los docentes (*docente\_curso\_induccion*) deben cumplir con el curso de inducción para poder orientar a los estudiantes durante el Servicio Comunitario.

Debido a que existen cursos de inducción para diferentes usuarios se crea una tabla tipo de curso de inducción (*tipo\_curso*).

Cada curso de inducción tiene los siguientes datos: id, nombre, fecha y hora de inicio, fecha y hora final, lugar, cantidad de cupos disponibles, fecha inicio de la inscripción, fecha final de la inscripción, cantidad de cupos iniciales, cantidad de cupos confirmados y la activación del curso, se puede observar en la Figura 8 el script SQL de la tabla.

```
SET NAMES utf8;
SET FOREIGN_KEY_CHECKS = 0;

-----
-- Table structure for `curso_induccion`
-----
DROP TABLE IF EXISTS `curso_induccion`;
CREATE TABLE `curso_induccion` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `nombre_curso_induccion` varchar(255) DEFAULT NULL,
  `fecha_inicio` date DEFAULT NULL,
  `fecha_final` date DEFAULT NULL,
  `hora_inicio` time DEFAULT NULL,
  `hora_final` time DEFAULT NULL,
  `lugar` varchar(255) DEFAULT NULL,
  `cantidad_cupos` varchar(255) DEFAULT NULL,
  `fecha_final_inscripcion` date DEFAULT NULL,
  `tipo_curso_id` varchar(2) NOT NULL,
  `fecha_inicio_inscripcion` date DEFAULT NULL,
  `cantidad_cupos_confirmados` varchar(255) DEFAULT '0',
  `cantidad_cupos_iniciales` varchar(255) DEFAULT NULL,
  `activar_curso` varchar(255) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=20 DEFAULT CHARSET=utf8;
```

Figura 8 - Script SQL de la tabla (*curso\_induccion*).

Durante la iteración 1 se crearon las vistas, modelos y controladores de la aplicación como se puede verificar en la Figura 9.

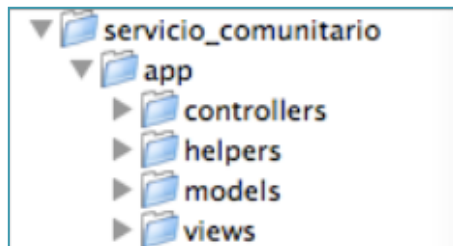


Figura 9 - Patrón MVC de la aplicación *servicio\_comunitario*.

En la autenticación de aplicación Web principalmente se crean todas las distribuciones de interfaces de usuarios, en la Figura 10 se puede observar parte de la distribución del Controlador Principal.

```
<body onload="actualizaReloj()">

  <div id="container">
    <div id="banner">
      <div id="logo"></div>
      <div id="adentro2">
        <div id="cerrar_sesion">
          <%= session[:primer_nombre].capitalize.to_s %><br/>
          <span id="cerrar">Cerrar sesión</span>
          <%= link_to image_tag("cerrar_sesion.gif"),logout_path %>
        </div>
      </div>
      <div id="Fecha_Reloj"></div>
      <%= flash.each do |name, msg| -%>
      <%= content_tag :div, msg, :id => "flash_#{name}" %>
      <%= end -%>
    </div>
  </div>

```

Figura 10 - Distribución de la interfaz de usuario del Controlador Principal.

Para la creación de las distribuciones de interfaces de usuario se realizaron en paralelo los estilos CSS que le dan el formato a cada una de las vistas de la aplicación.

Finalizando la iteración 1, se elabora la autenticación según el rol de cada usuario (estudiante, docente, personal de la Unidad de Servicio Comunitario y miembro de la Comisión del Servicio Comunitario). La autenticación es efectuada introduciendo cédula y contraseña, la cual es cifrada con un algoritmo de una sola vía, a través del sistema criptográfico SHA1 (Secure Hash Algorithm), como ejemplo tenemos:

Una contraseña "123123", después de ser cifrada el resultado es "2b96ba9d340d7c1a481a3237adcb1630a046da52", que se compara con la tabla Estudiante de CONEST, la cual tiene los registros de las cédulas y contraseñas de todos los estudiantes de la Facultad de Ciencias, para el caso de los docentes se compara con la tabla Docente que igualmente tiene los registros necesarios, los demás usuarios se encuentran en la tabla Usuarios en Servicio.

Una vez que el usuario accede a la aplicación se crean todas las variables de sesión y se le despliegan las funcionalidades de acuerdo a su rol. En el caso de

dejar la sesión inactiva durante una hora, la sesión se cierra y de esta forma se protege la información de personas malintencionadas.

En la Figura 11 se puede observar la vista de la autenticación.

Figura 11 - Autenticación de usuarios.

- **Pruebas**

Para las pruebas de la autenticación se escogieron diferentes usuarios con distintos roles de una base de dato de desarrollo de CONEST, donde colocamos la cédula de los usuarios y “123123” como clave.

Algunos casos se pueden ver en la Tabla 5.

Tabla 5 - Casos de Prueba Autenticación.

Rol	Cédula	Contraseña	Acceso
Estudiante	17124125	123123	Correcto
Estudiante	16033952	123123	Correcto
Docente	10170468	123123	Correcto
Docente	10201585	123123	Correcto
Docente/Estudiante	13736933	123123	Correcto
Administrador	9483179	123123	Correcto
Estudiante	10002466	123123	Se muestra al estudiante un mensaje notificándole que no se encuentra inscrito en el

			período actual.
Docente Estudiante Administrador	17124125&^*	^&*(#\\$	Se muestra al estudiante un mensaje notificándole que sus datos son inválidos.

Comprobando los casos de prueba mostrados en la Tabla 6, se verifica que el usuario que accedía a la aplicación cumplía con las validaciones de la autenticación.

En el caso de que el usuario tenga varios roles puede escoger con el que desee ingresar, si es docente y miembro de la comisión puede seleccionar el que le sea conveniente en ese momento.

Se realizaron pruebas de aceptación a la Lic. Yadira Salcedo encargada de la Unidad de Servicio Comunitario y el auxiliar docente Br. Juan Carlos Florville, los cuales hicieron diferentes ingresos y verificación de interfaz, aceptando la iteración 1.

Las pruebas funcionales de autenticación por parte de la comunidad estudiantil y docente, se realizarán durante las actividades de inscripción en los cursos de inducción.

### 3.9. Iteración 2

Para efectuar el Servicio Comunitario es necesario asistir previamente al curso de inducción, el cual sirve de guía para los estudiantes y docentes, por ello la iteración 2 abarca algunos de las actividades que se llevan a cabo.



- **Planeación**

En la tabla 6 se pueden apreciar de forma general las historias de usuario de la iteración 2.

*Tabla 6 - Historia de Usuario para la iteración dos.*

Nro.	Fecha Inicio	Fecha Final	Requerimientos	Estado
5	15/05/2010	22/05/2010	Crear, editar y modificar cursos de inducción para estudiantes y/o docentes.	Finalizada
6	23/05/2010	26/05/2010	Realizar inscripción de estudiantes y/o docentes en los cursos de inducción.	Finalizada
7	28/05/2010	04/06/2010	Confirmar inscripción estudiantes y/o docentes.	Finalizada

- **Diseño**

En el diseño de la iteración actual se realizó un modelo que ilustra las actividades del proceso de curso de inducción, y el diseño del PDF que se utilizará para la inscripción de los estudiantes (Ver Figura 12 y 13).

### **Diseño PDF de Inscripción en el Curso de Inducción**

Una vez que el estudiante se inscribe en el curso de inducción, se genera un comprobante. También durante la realización del curso se lleva la asistencia de los participantes, por lo que se propuso un comprobante innovador con código de barra para agilizar los procesos de toma de asistencia a través de lectores ópticos.

Se diseñó la primera propuesta del comprobante de inscripción la cual se puede observar en la Figura 12.








	<b>Universidad Central de Venezuela</b> <b>Facultad de Ciencias</b> <b>Unidad de Servicio Comunitario</b> <b>Comprobante de Inscripción del Curso de Inducción</b>	
Planilla Nro: 1100		
Fecha de Inscripción: 26-08-2010		
Escuela Cédula Apellido Nombre Correo Teléfono Sexo Dirección	Computacion 17124125 Simon Mariana m.simonluna@gmail.com 0212-2395752 0412-6274267 Femenino Final Av. El Rosario, Calle Alegria, Ota Anaelara, Urb Los Chorros	 
Universidad Facultad Curso de Inducción Fecha Inicio Fecha Inscripción	Universidad Central de Venezuela Ciencias Curso II-2008 02-04-2008 26-08-2010	
Unidad de Servicio Comunitario		Estudiante
UCV CIENCIAS, Paseo los Ilustres, Los Chaguaramos, Caracas 1040 Tlf: (0212) 605.1186, Correo Electrónico: serviciocomunitario@ciencias.ucv.ve Dirección Postal: AP 20513, Caracas 1020-A		
Enlido por CONEST Servicio Comunitario		
-----		
Curso II-2008 Nombre: Mariana Apellido: Simon Cédula: 17124125 Escuela: Computacion	 	 Universidad Central de Venezuela Facultad de Ciencias Unidad de Servicio Comunitario 

Figura 12 - Comprobante de Inscripción en el Curso de Inducción.

### Modelo del proceso de curso de inducción

El siguiente modelo (Ver Figura 13) ilustra las actividades que realizan los distintos usuarios durante el proceso del curso de inducción.

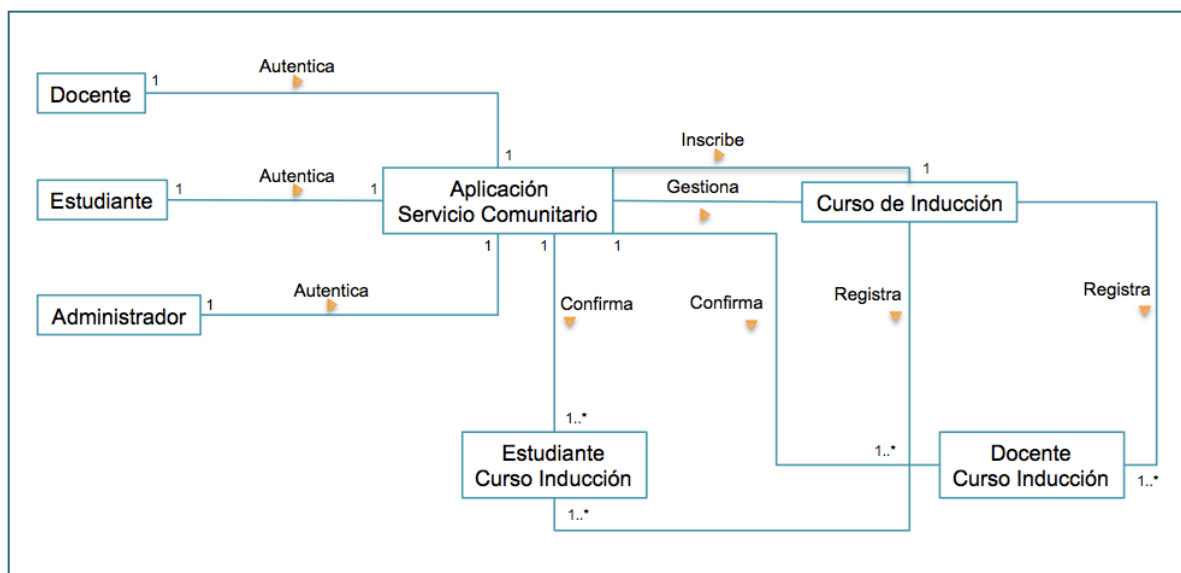


Figura 13 - Modelo del proceso de curso de inducción.

A continuación se muestran los usuarios que intervienen con la aplicación del Servicio Comunitario.

**Estudiante:** Son los estudiantes de la Facultad de Ciencias que interactúan con los servicios que ofrece la aplicación del Servicio Comunitario, en este caso la inscripción en un curso de inducción, autenticándose con su cédula de identidad y su clave.

**Docente:** Son los docentes de la Facultad de Ciencias que interactúan con los servicios que ofrece la aplicación de Servicio Comunitario en este caso la inscripción en un curso de inducción, autenticándose con su cédula de identidad y su clave de CONEST.

**Administrador:** Son los encargados de gestionar la aplicación del Servicio Comunitario autenticándose con su cédula y su clave asignada por el grupo de trabajo de esta aplicación, en esta iteración se encargan de administrar los docentes y estudiantes en los cursos de inducción.

**Aplicación Servicio Comunitario:** Representa gestión de la información de los procesos del Servicio Comunitario, controla los datos de los usuarios y controla diferentes módulos.

**Curso de Inducción:** Son los diferentes cursos de inducción en los que se puede inscribir los estudiantes y docentes. El encargado de crear, editar, eliminar y cambiar el estado de cada curso de inducción es el administrador.

**Estudiante Curso de Inducción:** Representa los estudiantes que se encuentran inscritos en un curso de inducción de tipo estudiante. El administrador confirma a los estudiantes inscritos.

**Docente Curso de Inducción:** Representa los docentes que se encuentran inscritos en un curso de inducción de tipo docente. El administrador confirma a los docentes que asistirán.

- **Codificación**

Para la creación de los cursos de inducción se debe introducir información como el nombre, fecha de inicio y final de inscripción, fecha de inicio y final de curso, lugar, cantidad de cupos disponibles, cantidad de cupos iniciales, cantidad de cupos confirmados, estado del curso y tipo de curso (docente o estudiante).

Se realizó un formulario que el usuario Administrador debe completar como se puede ver en la Figura 14 y el almacenamiento en la Figura 15.

```

def guardar

  @fecha_inicio_inscripcion="#{@ano_inicio_inscripcion}-#{@mes_inicio_inscripcion}-#{@dia_inicio_inscripcion}"
  @variable = CursoInduccion.new(:tipo_curso_id => params[:tipo_curso],:nombre_curso_induccion => params[:nombre_cursoinduccion],
  :fecha_inicio => @fecha_inicio,:fecha_final => @fecha_final,:hora_inicio => params[:hora_inicio],
  :hora_final => params[:hora_final],:lugar => params[:lugar].to_s,:cantidad_cupos => params[:numero_cupo],
  :cantidad_cupos_iniciales => params[:numero_cupo],:fecha_final_inscripcion => @fecha_final_inscripcion,
  :fecha_inicio_inscripcion => @fecha_inicio_inscripcion,:activar_curso => params[:activar_curso] )
  @variable.save

  if params[:tipo_curso]=="D"
    @bitacora = Bitacora.create(:tipo_importancia_evento_id => "A",:tipo_modulo_id => "SC",:tipo_evento_id => "INF",
    :fecha_hora => Time.now,:descripcion => "El Administrador "+session[:Administrador]+" ha creado un nuevo curso
    de induccion para Docentes",:nombre_tabla_afectada=>"curso_induccion",:ip_origen=>request.remote_ip)
  else
    @bitacora = Bitacora.create(:tipo_importancia_evento_id => "A",:tipo_modulo_id => "SC",:tipo_evento_id => "INF",
    :fecha_hora => Time.now,:descripcion => "El Administrador "+session[:Administrador]+" ha creado un nuevo curso
    de induccion para Estudiantes",:nombre_tabla_afectada=>"curso_induccion",:ip_origen=>request.remote_ip)
  end

  flash[:notice] = 'Se Creo Correctamente el Curso de Inducción'
  redirect_to :controller => 'administrador', :action => 'configurar_curso_induccion'
end

```

Figura 14 - Almacenamiento de la creación de un Curso de Inducción.

The form contains the following fields and controls:

- Tipo curso:
- Nombre del curso:
- Fecha Inicio:  /  /
- Fecha Final:  /  /
- Hora Inicio:
- Hora Final:
- Lugar:
- Cantidad Total de Cupos:
- Fecha Inicio Inscripción:  /  /
- Fecha Final Inscripción:  /  /
- Activar Curso:

Buttons:

Figura 15 - Formulario para la creación de un curso de inducción.

Durante las fechas de inscripción, el administrador tiene que cambiar el estado del curso a activo, para que los estudiantes puedan acceder a la aplicación y realizar su inscripción como se observa en la Figura 16, de igual forma al crear un curso de inducción tipo docente.

Se verifica que el estudiante no esté inscrito en otro curso y se almacena en la base de datos en la tabla *estudiante\_curso\_induccion*, ver Figura 17.

Haga Click en el Curso de Inducción


Nombres	Fecha Inicio	Fecha Final	Hora Inicio	Hora Final	Lugar	Cupos	Fecha Final Inscripción	Inscribir
Curso II-2010	29-06-2010	01-07-2010	08:00 am	05:00 pm	Auditorio tobias laser	21	18-06-2010	

Figura 16 - Interfaz de Inscripción de un Estudiante en el Curso de Inducción.

```
#----- Inscribir Curso de Induccion
def inscribir
  @licenciatura = Licenciatura.find(:all,:conditions => ['id = ?',session[:licenciatura_id]])
  @inscrito = EstudianteCursoInduccion.find(:first,:conditions=>["estudiante_cedula = ? ",session[:cedula]])
  if @inscrito==nil
    @variable = EstudianteCursoInduccion.new(:estudiante_cedula => session[:cedula],:curso_induccion_id => params[:radio],
      :licenciatura_id => session[:licenciatura_id],:fecha_hora_inscripcion => Time.now)
    @variable.save
    @cursoestudiante = CursoInduccion.find(params[:radio]) |
    CursoInduccion.update_counters @cursoestudiante, :cantidad_cupos => -1
    @bitacora = Bitacora.create(:estudiante_cedula => session[:cedula],:tipo_importancia_evento_id => "M",
      :tipo_modulo_id => "SC",:tipo_evento_id => "INF",:fecha_hora => Time.now,
      :descripcion => "El Estudiante se ha inscrito en un Curso de Induccion",:nombre_tabla_afectada=>"curso_induccion,
      estudiante_curso_induccion",:ip_origen=>request.remote_ip)
    flash[:notice] = "Estudiante inscrito correctamente"
  end
  redirect_to :controller => 'estudiante', :action => 'index'
end
```

Figura 17 - Almacenar Estudiante inscrito en un Curso de Inducción.

Otra de las opciones que tiene el administrador en la aplicación es la de editar y eliminar los cursos de inducción creados.

Una vez que los estudiantes se inscriban en el curso de inducción, deben imprimir su comprobante, que pueden descargar de la aplicación en formato PDF, para luego dirigirse a la Unidad de Servicio Comunitario y confirmar su asistencia, de esta forma poder ofrecer cupos disponibles a las personas que realmente quieran hacer el curso.

- **Pruebas**

Al culminar esta iteración se colocó la aplicación en producción, ajustándonos al evento del curso de inducción II-2010 en las fechas determinadas en el calendario del Servicio Comunitario.

Para verificar los procesos se hicieron pruebas funcionales. Se crearon dos cursos de inducción, uno de estudiantes y otro para docentes.

En los cursos se inscribieron y confirmaron a través de la aplicación 129 estudiantes y 7 docentes. El proceso de inscripción se llevó a cabo entre las fechas 14 y 18 de junio de 2010, sin ningún inconveniente para luego confirmar a los participantes la semana siguiente en la Unidad de Servicio Comunitario.

Se realizaron pruebas de aceptación durante el desarrollo de la aplicación para el comprobante de inscripción del curso de inducción, en donde el cliente encargado de realizar los comprobantes anteriormente la Lic. Yadira Salcedo, añadió nuevos campos como el número de planilla y sexo del participante.

Los campos fueron añadidos al comprobante y el cliente aceptó la propuesta de la iteración actual.

Con respecto a las pruebas de la autenticación e inscripción del curso de inducción por parte de la comunidad estudiantil y docente, no se presentaron eventualidades, y se realizaron todas las pruebas funcionales y actividades de logísticas de forma normal.

También se realizaron mejoras en la interfaz de confirmación de los estudiantes, se agregó la foto del estudiante, teléfonos de contacto y correo electrónico.

### 3.10. Iteración 3

En la iteración 3 se añaden nuevas funcionalidades para la gestión del proceso de curso de inducción, ya que se debe realizar la toma de asistencia de los participantes y la evaluación del contenido del curso a los estudiantes.

- **Planeación**

En la presente iteración se exponen en la Tabla 7 las historias de usuario.

*Tabla 7 - Historia de Usuario para la iteración tres.*

Nro.	Fecha Inicio	Fecha Final	Requerimientos	Estado
8	08/06/2010	17/06/2010	Controlar asistencia de los estudiantes y/o docentes durante el Curso de Inducción.	Finalizada
9	19/06/2010	20/06/2010	Realizar evaluación en línea del Curso de Inducción de los estudiantes.	Finalizada
10	21/06/2010	23/06/2010	Realizar certificados de los Cursos de Inducción de docentes y constancias de los estudiantes.	Finalizada

- **Diseño**

Se tomó en cuenta el diseño de la iteración 2 con respecto al comprobante de Inscripción de los inscritos en el curso de inducción para el control de la asistencia.



El comprobante de los inscritos contiene un código de barra, en el cual se almacena la cédula del participante y la licenciatura a la que pertenecen. En la Figura 18 se puede ver un ejemplo de código de barra.



*Figura 18 - Código de barra que identifica al participante.*

De esta forma el estudiante llega al lugar donde se realiza el curso de inducción presenta el comprobante, los datos se capturan con un lector óptico y se almacena la asistencia por esa sesión.

Después que el estudiante asiste al curso de inducción debe presentar una evaluación que contiene los tópicos presentados. Se diseñó un prototipo de interfaz de usuario que se observa en la Figura 19. Se muestran las preguntas, respuestas y el tiempo restante para enviarla.

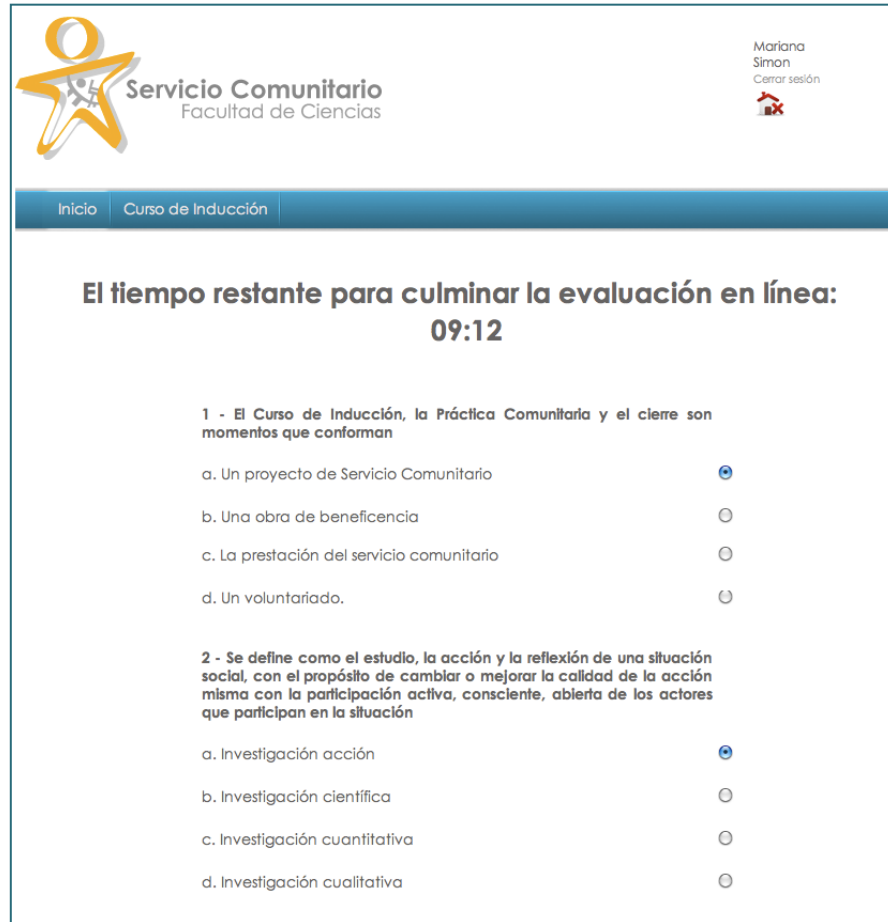


Figura 19 - Diseño de interfaz de usuario para la evaluación en línea.

La evaluación en línea del curso de inducción de los estudiantes se realiza accediendo a la aplicación y cumpliendo los requisitos previos, tales como haberse inscrito en un curso de inducción, confirmado su asistencia y haber estado presente al menos dos días en el curso de inducción.

Para el último requisito de esta iteración se elaboraron los certificados y constancias para los docentes y estudiantes respectivamente, que aprobaron el curso de inducción. El diseño del certificado de un docente se ve en la Figura 20.



Figura 20 - Diseño del certificado del Curso de Inducción de Docentes.

- **Codificación**

Cumpliendo con el requerimiento de llevar el control de la asistencia que puede ser de forma manual o a través de los lectores ópticos, se codificaron diferentes métodos. La Figura 21 muestra cómo se almacena la asistencia de los estudiantes cuando no son utilizados los lectores ópticos.

```
def guardar_asistencia_estudiante

  @cont=0
  @contador=0

  @conta=params[:contador]
  @id_curso=params[:id_curso]

  $i = @contador.to_i;
  $num = @conta.to_i;

  while $i < $num do
    @aux=$i

    @paramcedula="cedula_#{@aux}"

    # CHECKS DIAS UNO
    @param_check_dia_uno="hecdiauno_#{@aux}"
    @checks_dia_suno=params[@param_check_dia_uno]

    if (@checks_dias_uno)
      @checks_dias_uno = "1"
    else
      @checks_dias_uno = "0"
    end
  end
end
```

Figura 21 - Código para almacenar la asistencia del estudiante.

La asistencia se maneja a través de la interfaz de usuario, donde se muestra un listado de los inscritos y los días en los que se impartirá el curso de inducción.

Dependiendo del día se selecciona por participante que asistió y su hora de llegada. En el caso de que haya asistido se le asigna el valor de 1 a la variable definida para el día que se seleccionó, en el caso contrario esta variable tendrá el valor 0, he indicará que el participante no asistió el día elegido.

Si se llevó la asistencia a través de los lectores ópticos se pasa un parámetro indicando el día y la cédula del participante, con esto se toma su información y se almacena la presencia en el curso de inducción.

La evaluación en línea la debe gestionar el personal de la Unidad de Servicio Comunitario, quienes crean las preguntas con sus respuestas correspondientes.

En el caso de que se desee, tienen las opciones de eliminar o editar cualquiera de las preguntas o respuestas.

Cuando activen en el manejador de eventos la evaluación en línea, al estudiante se le despliega en su menú principal la opción de presentarla, que puede seleccionarla en el período que se establezca. Una vez que el estudiante selecciona la opción de la evaluación en línea se mostrarán las preguntas y respuestas que se encuentran en la base de datos de forma aleatoria, con el fin de que la prueba sea lo menos parecida entre los estudiantes.

En la Figura 22 se puede observar los requerimientos que debe tener un estudiante para presentar la prueba como lo son: no haber presentado antes, y haber asistido al mínimo de días exigidos por la Unidad de Servicio Comunitario.

```

def shuffle
  self.sort_by{rand}
end
def prueba_en_linea
  @estudianteprueba=ServiciocomunitarioResultadoPrueba.first(:first, :conditions => ['estudiante_cedula= ?', session[:cedula]])
  @asistioacurso=EstudianteCursoInduccion.first(:first, :conditions => ['estudiante_cedula= ?', session[:cedula]])
  @cantidadpreguntas=ServiciocomunitarioPruebaPregunta.first
  if(@cantidadpreguntas!=nil)
    if((@asistioacurso.dia_uno=="1" && @asistioacurso.dia_dos=="1" && @asistioacurso.dia_tres=="0") ||
      (@asistioacurso.dia_uno=="0" && @asistioacurso.dia_dos=="1" && @asistioacurso.dia_tres=="1") ||
      (@asistioacurso.dia_uno=="1" && @asistioacurso.dia_dos=="0" && @asistioacurso.dia_tres=="1") ||
      (@asistioacurso.dia_uno=="1" && @asistioacurso.dia_dos=="1" && @asistioacurso.dia_tres=="1") )
      if (@estudianteprueba==nil)
        pruebapreguntatemporal=ServiciocomunitarioPruebaPregunta.first
        pruebapreguntatemporal=ServiciocomunitarioPruebaPregunta.first
        resultadopruebatemporal = ServiciocomunitarioResultadoPrueba.new(:estudiante_cedula => session[:cedula],
          :serviciocomunitario_prueba_pregunta_id => pruebapreguntatemporal.id,
          :serviciocomunitario_prueba_respuesta_id => pruebapreguntatemporal.id,:tipo_nota_id =>1)
        resultadopruebatemporal.save
        @preguntas=ServiciocomunitarioPruebaPregunta.all
        @preguntas=@preguntas.shuffle
        @respuestas=ServiciocomunitarioPruebaRespuesta.all
        @respuestas=@respuestas.shuffle
      else
        flash[:error] = "Usted ya realizó la Prueba en Línea del Curso de Inducción"
        redirect_to :controller => 'estudiante', :action => 'index'
      end
    else
      flash[:error] = "Usted no puede presentar la Prueba en Línea porque no cumplió con la asistencia mínima del Curso de Inducción"
      redirect_to :controller => 'estudiante', :action => 'index'
    end
  else
    flash[:error] = "En este momento no existe una prueba disponible"
    redirect_to :controller => 'estudiante', :action => 'index'
  end
end
end

```

Figura 22 - Código de creación de la Prueba en Línea.

Una vez que el estudiante haya presentado la evaluación, los datos son almacenados en la tabla *serviciocomunitario\_resultado\_prueba* y la calificación en la tabla *estudiante\_curso\_induccion*. La calificación del estudiante es enviada automáticamente a su correo electrónico indicándole si aprobó o no.

Finalmente, para generar las constancias de los estudiantes que aprobaron el curso de inducción, se genera un documento en formato PDF con los datos necesarios como se exponen en la Figura 23.

```
#----- Comprobante de que realizo y aprobo el curso de induccion
def generar_pdf_aprobados
  @tipo=params[:tipo]
  @curso_induccion_id=params[:curso_induccion_id]
  @cedula=params[:cedula]
  @nombre=params[:nombre]
  @apellido=params[:apellido]
  @licenciatura=params[:licenciatura]
  @nombre_curso_induccion=CursoInduccion.find(:first,:conditions=>["id=?",@curso_induccion_id]) # Nombre del curso de induccion
  @tiempo=Time.now

  if @tipo=="estudiante"
    pdf = PDF::Writer.new(:paper=>'LETTER')
    i2 = pdf.add_image_from_file("public/images/fondo_pdf_dos.jpg", 0, 0, width = 612 , height = 792)
    @coordinador=ParametroGeneral.find(:first,:conditions=>["nombre=?", "COORDINADOR_UPSA"]) # Nombre del coordinador de SC
    @tiempo=Time.now

    pdf.text texto("Universidad Central de Venezuela
    Facultad de Ciencias
    Coordinación Académica - Coordinación de Extensión
    Unidad de Servicio Comunitario
    "), :font_size => 12, :justification => :center,:spacing=>1.5,:right=>50, :left=>50
    pdf.text texto("
    CONSTANCIA
    "), :font_size => 14, :justification => :center,:spacing=>1.5,:right=>50, :left=>50

    pdf.text texto("
    Quien suscribe Profa. Inés Tovar de Rodríguez, Coordinadora de la Unidad de Servicio Comunitario de la
    Facultad de Ciencias,Universidad Central de Venezuela, por medio de la presente hace constar que el (la)
    Br. #{@apellido} #{@nombre}, portador (a) de la C.I N° #{@cedula}, aprobó el
    #{@nombre_curso_induccion.nombre_curso_induccion}
    en el semestre #{@nombre_curso_induccion.nombre_curso_induccion.split("Curso")},
    con base en lo estipulado en la Ley de Servicio Comunitario del Estudiante de Educación Superior,
    Gaceta N° 38.272 del 14 de septiembre de 2005.
```

Figura 23 - Código para generar el comprobante de aprobado.

- **Pruebas**

Mientras se realizó el Curso II 2010 se hicieron pruebas funcionales, llevando la asistencia de los cursos con los lectores ópticos.

Asistieron 129 estudiantes distribuidos de la siguiente forma: el día uno 111, el día dos 122 y el día tres 125. El proceso se llevó a cabo sin ningún inconveniente y los usuarios de la Unidad de Servicio Comunitario aceptaron satisfechos la forma en se ejecutó el proceso de control de la asistencia de los cursos de inducción.

Por parte de los prestadores se vieron complacidos en cuanto el tiempo de ingreso al Auditorio donde se impartió el curso de inducción, mencionando que las actividades se realizaron de forma rápida y sencilla.

La Unidad de Servicio Comunitario validó el funcionamiento correcto de la aplicación durante esta iteración.

### **3.11. Iteración 4**

El objetivo de esta iteración es realizar las funcionalidades para la gestión de proyecto de Servicio Comunitario.

Posteriormente de haber aprobado el curso de inducción el estudiante debe inscribir el proyecto de Servicio Comunitario, por lo que se necesitan funcionalidades en la aplicación que permitan crear proyectos, tipos de proyectos, comunidades, inscripción del proyecto, entre otros.

- **Planeación**

En la Tabla 8 se muestran las historias de usuario de la iteración 4.

*Tabla 8 - Historia de Usuario para la iteración cuatro.*

<b>Nro.</b>	<b>Fecha Inicio</b>	<b>Fecha Final</b>	<b>Requerimientos</b>	<b>Estado</b>
11	24/06/2010	25/06/2010	Gestionar proyectos.	Finalizada
12	26/06/2010	27/06/2010	Gestionar tipo de proyectos.	Finalizada
13	29/06/2010	30/06/2010	Gestionar comunidades	Finalizada
14	02/07/2010	10/07/2010	Gestionar inscripción de los estudiantes.	Finalizada
15	11/07/2010	00/00/0000	Confirmar y editar a los estudiantes inscritos en el Proyecto de Servicio Comunitario.	Pendiente

El último requerimiento que se encuentra en estado pendiente por factores de tiempo, se agregó en la historia de usuarios de la siguiente iteración, en la que se finalizó.

- **Diseño**

Para el diseño de la presente iteración se realizó un modelo de base de datos correspondiente a los proyectos de Servicio Comunitario. (Ver Figura 24)



## Modelo de base de datos

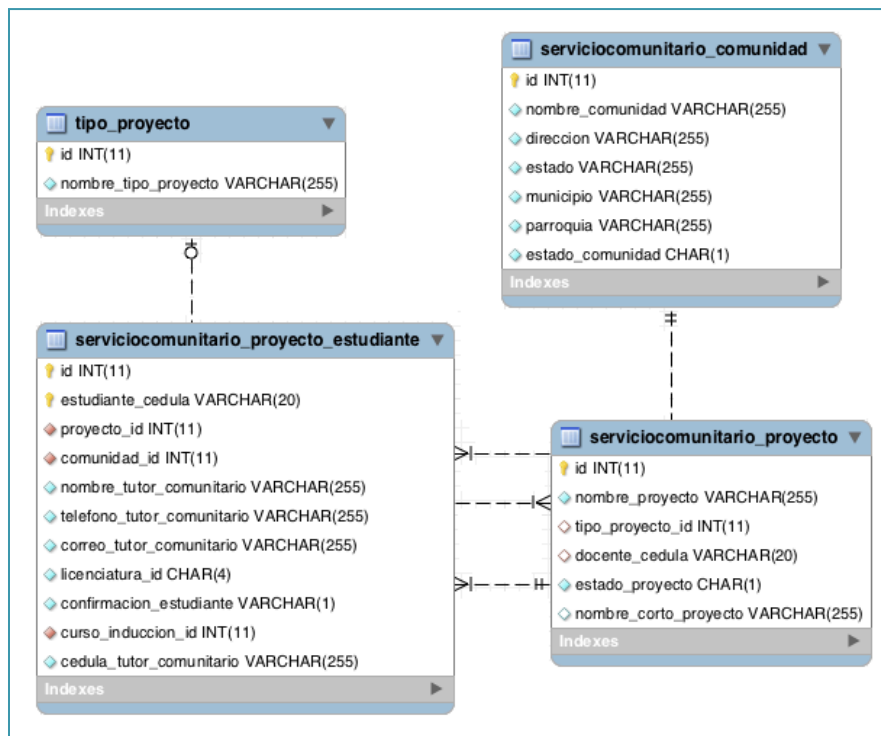


Figura 24 - Modelo de base de datos Proyecto de Servicio Comunitario.

Cuando un estudiante inscribe el proyecto de Servicio Comunitario se almacenan los datos en la tabla *serviciocomunitario\_proyecto\_estudiante* que se encuentra relacionada con las tablas de *serviciocomunitario\_proyecto*, *tipo\_proyecto* y *serviciocomunitario\_comunidad* para realizar la práctica de Servicio Comunitario.

- **Codificación**

Cumpliendo con los requerimientos de esta iteración se realizó la gestión de los proyectos de Servicio Comunitario disponibles para los estudiantes. Se elaboró un formulario en donde se debe completar nombre, nombre corto, tipo de proyecto, estado y el responsable del proyecto.

Después de completar el formulario y almacenada la información, estará disponible para la actividad de inscripción de los estudiantes en el proyecto de Servicio Comunitario. En la Figura 25 se muestra parte del código para almacenar los proyectos.

```
#----- Crear Proyecto
def crear_proyecto
  @tipo_proyecto = TipoProyecto.all
end

def guardar_crear_proyecto
  @responsable=params[:cedula_responsable_proyecto]
  @split=@responsable.split(' ')
  @cedula=@split[0]
  @nombres=@split[1]
  @confirmaciondocente = Docente.first(:first, :conditions => ['cedula= ?',@cedula])
  if(@confirmaciondocente==nil)
    flash[:error] = 'Debe introducir un RESPONSABLE correcto'
    redirect_to :controller => 'administrador', :action => 'crear_proyecto'
  else
    @responsable=params[:cedula_responsable_proyecto]
    @split=@responsable.split(' ')
    @cedula=@split[0]
    @nombres=@split[1]
    @guardar_crear_proyecto = ServiciocomunitarioProyecto.new(:nombre_proyecto => params[:nombre_proyecto],
      :tipo_proyecto_id => params[:tipo_proyecto],:docente_cedula =>@cedula,
      :estado_proyecto => params[:activar_curso],:nombre_corto_proyecto=>params[:nombre_proyecto_corto])
    @guardar_crear_proyecto.save
    @bitacora = Bitacora.create(:tipo_importancia_evento_id => "A",:tipo_modulo_id => "SC",
      :tipo_evento_id => "INF",:fecha_hora => Time.now,
      :descripcion => "El Administrador "+session[:Administrador]+"Creo
      correctamente un Proyecto de Servicio Comunitario",
      :nombre_tabla_afectada=>"serviciocomunitario_proyecto",:ip_origen=>request.remote_ip)
    flash[:notice] = 'Se Creo Correctamente el Proyecto de Servicio Comunitario'
    redirect_to :controller => 'administrador', :action => 'crear_proyecto'
  end
end
```

Figura 25 - Código para almacenar la creación de Proyectos de Servicio Comunitario.

Una vez que el administrador habilite las comunidades disponibles y la inscripción de proyectos en el manejador de eventos. Los estudiantes proceden a realizar su inscripción en el proyecto de Servicio Comunitario.

Al estudiante se le despliega la opción de inscripción de proyecto en su menú principal, donde se ofrecen los proyectos y comunidades disponibles para la prestación del Servicio Comunitario. Seleccionado el proyecto, la comunidad, y

habiendo completado la información del tutor comunitario, él quedará inscrito. La Figura 26 corresponde al proceso de inscripción del estudiante.

```
def guardar_inscripcion_proyecto
  @curso_induccion=EstudianteCursoInduccion.first(:first, :conditions => ['estudiante_cedula= ?', session[:cedula]])
  @estudiante_inscrito=ServiciocomunitarioProyectoEstudiante.first(:first,
  :conditions => ['estudiante_cedula= ?', session[:cedula]])
  if(@estudiante_inscrito==nil)
    @crearproyecto = ServiciocomunitarioProyectoEstudiante.new(:estudiante_cedula => session[:cedula],
    :proyecto_id => params[:nombre_proyecto], :comunidad_id => params[:comunidad],
    :nombre_tutor_comunitario => params[:nombre_tutor_comunitario],
    :nombre_tutor_comunitario => params[:nombre_tutor_comunitario],
    :telefono_tutor_comunitario => params[:telefono_tutor_comunitario],
    :correo_tutor_comunitario => params[:correo_tutor_comunitario],
    :licenciatura_id => session[:licenciatura_id],
    :curso_induccion_id=>@curso_induccion.curso_induccion_id,
    :cedula_tutor_comunitario=>params[:cedula_tutor_comunitario])
    @crearproyecto.save
    flash[:notice] = 'Usted se inscribio correctamente en un Proyecto de Servicio Comunitario'
    redirect_to :controller => 'estudiante', :action => 'index'
  else
    flash[:error] = 'Usted ya se encuentra inscrito en un Proyecto de Servicio Comunitario'
    redirect_to :controller => 'estudiante', :action => 'index'
  end
end
```

Figura 26 - Código guardar la inscripción de un estudiante en el Proyecto de Servicio Comunitario.

Se verifica que el estudiante no se encuentre inscrito en ningún proyecto, ya que una vez inscrito sólo puede ser cambiado por el administrador de la Unidad de Servicio Comunitario. Si cumple con los requisitos entonces se almacena en la tabla *serviciocomunitario\_proyecto\_estudiante*. De esta forma puede acceder a su cuenta y observar todos sus datos del Servicio Comunitario y generar el comprobante de inscripción.

- **Prueba**

Se realizaron pruebas de aceptación al cliente de la Unidad de Servicio Comunitario que aprobó y creó proyectos, microproyectos y comunidades cambiando su estado y observando el flujo de trabajo. Entre los cambios que surgieron se encuentra el formato del PDF que se genera al inscribirse un estudiante. El PDF generado contiene un carnet que presentan los estudiantes al

momento de ingresar a la comunidad, al cual se le realizaron cambios de tamaño y de información.

Se realizaron pruebas funcionales entre los días 4 de octubre y 8 de octubre de 2010 a los estudiantes que debían inscribir proyecto de Servicio Comunitario para poder empezar la práctica en la comunidad que seleccionó.

Un total de 113 estudiantes se inscribieron en el proyecto de Servicio Comunitario durante el período II-2010 a través de la aplicación, haciendo con esto las pruebas funcionales.

### 3.12. Iteración 5

En esta iteración se continúa uno de los requerimientos de la iteración 3 que consiste en confirmar y editar a los estudiantes que han inscrito el proyecto de Servicio Comunitario. También se gestiona la asignación de tutores académicos.

- **Planeación**

Cabe destacar que la iteración 4 se empezó a desarrollar después de inscribir a los estudiantes en Cursos de Inducción y llevar su asistencia al grupo curso de inducción II -2010, por ello las fechas son las que se señalan en la tabla.

*Tabla 9 - Historia de Usuario para la iteración cinco.*

Nro.	Fecha Inicio	Fecha Final	Requerimientos	Estado
16	03/08/2010	10/08/2010	Confirmar y editar a los estudiantes inscritos en proyecto de Servicio Comunitario.	Finalizada
17	11/08/2010	20/08/2010	Gestionar la asignación de Tutores	Pendiente

			Académicos	
--	--	--	------------	--

- **Diseño**

Pra el requerimiento número 16 se propone la siguiente interfaz de usuario por parte del administrador de la Unidad de Servicio Comunitario, ver Figura 30. En donde el personal de la Unidad de Servicio Comunitario debe seleccionar el icono de confirmar expuesto en la Figura 27. Si se quiere editar la inscripción de proyecto de un estudiante se selecciona el icono de editar, para cambiar la información que se desee.

Estudiantes Inscritos en el Proyecto de Servicio Comunitario										
Nro	Escuela	Nombre	Apellido	Cédula	Contacto	Curso	Foto	Confirmar	Comprobante	Editar
1	Computacion	Jesus Miguel	Mata Mora	16033952	jesus.mata183@gmail.com 5555555 5555555	Curso II-2008				

Figura 27 - Confirmar inscripción de proyecto de Servicio Comunitario de un estudiante.

A continuación se describe un diagrama de flujo para visualizar las actividades de asignación de tutores académicos.

### Diagrama de Flujo

Se propone un diseño de diagrama de flujo para observar cómo se realizará la asignación de los tutores académicos del prestador de Servicio Comunitario. En la Figura 28 se encuentran los pasos a seguir, si un docente es miembro de la Comisión de Servicio Comunitario de su escuela se le desplegará el menú correspondiente.

Si el representante de la Comisión de Servicio Comunitario por licenciatura, quiere asignar los tutores académicos para los estudiantes que hayan aprobado un curso de inducción, deberá seleccionarlo. Seguidamente aparecerá un listado de docentes de la licenciatura del representante de la Comisión de Servicio Comunitario, en donde se seleccionará la cantidad de estudiantes que estos tendrán a cargo como tutor académico. Luego se podrá ver el listado y generarlo en formato PDF.

El listado con la asignación deberá ser presentado ante el Consejo de Escuela para su aprobación. Después el representante de la Comisión de Servicio Comunitario por licenciatura tiene la opción de enviarle el correo electrónico al estudiante indicándole el tutor académico asignado y así mismo al tutor académico comunicarle los datos del estudiante bajo su tutela.

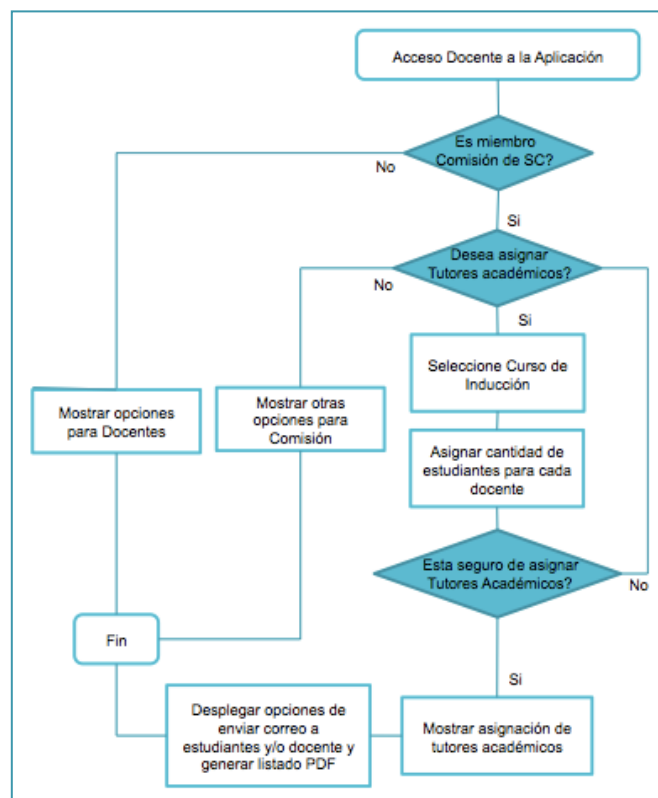


Figura 28 - Diagrama de Flujo para la asignación de tutores académicos.

- **Codificación**

La codificación del requerimiento 16 de la tabla de historias de usuario de esta iteración, referente a la edición de los estudiantes inscritos en los proyectos de Servicio Comunitario, es explicada a continuación.

En la Figura 29 se refleja un fragmento de la codificación realizada para editar a los estudiantes inscritos en proyecto, la cual muestra las consultas a las tablas de *serviciocomunitario\_proyecto\_estudiante*, *serviciocomunitario\_tutor academico\_asignado*, a los proyectos y comunidades que se encuentran relacionados el estudiante.

El personal de la Unidad de Servicio Comunitario podrá seleccionar de un listado de estudiantes inscritos, el estudiante que necesite editar y podrá cambiar la información que desee con respecto al proyecto, comunidad y tutor comunitario, para actualizarlos en la base de datos.

```
def editar_estudiante_en_proyecto
  @cedula=params[:cedula]
  @estudiante_proyecto = ServiociomunitarioProyectoEstudiante.first(:first,
  :conditions => ['estudiante_cedula= ?' ,@cedula])
  @tutor_academico = ServiociomunitarioTutorAcademicoAsignado.find(:first,
  :conditions=>["estudiante_cedula = ?",@cedula])
  @proyecto=ServiociomunitarioProyecto.all
  @comunidad=ServiociomunitarioComunidad.all
end

def editar_estudiante_proyecto_inscrito
  @estudiante_proyecto = ServiociomunitarioProyectoEstudiante.first(:first,
  :conditions => ['estudiante_cedula= ?' ,params[:estudiante_cedula]])
  @estudiante_proyecto.update_attributes(:proyecto_id => params[:tipo_proyecto],
  :comunidad_id => params[:comunidad],:nombre_tutor_comunitario => params[:nombre_tutor_comunitario],
  :telefono_tutor_comunitario => params[:telefono_tutor_comunitario],
  :correo_tutor_comunitario => params[:correo_tutor_comunitario],
  :cedula_tutor_comunitario=>params[:cedula_tutor_comunitario])
  @bitacora = Bitacora.create(:tipo_importancia_evento_id => "A",:tipo_modulo_id => "SC",
  :tipo_evento_id => "INF",:fecha_hora => Time.now,
  :descripcion => "El Administrador "+session[:Administrador]+" edito el Proyecto de un Estudiante",
  :nombre_tabla_afectada=>"serviciocomunitario_comunidad",:ip_origen=>request.remote_ip)
  flash[:notice] = 'Se Edito Correctamente el Proyecto del Estudiante'
  redirect_to :controller => 'administrador', :action => 'listar_inscrito_proyecto_estudiante'
end
```

Figura 29 - Código para editar inscripción de estudiante en proyecto.

Para la siguiente iteración se describirá la codificación de la asignación de tutor académico, ya que en ésta no pudo ser finalizada por factores de tiempo.

- **Prueba**

Durante el proceso se realizaron cambios en el comprobante de inscripción del proyecto de Servicio Comunitario, sustrayendo información que no era necesaria para él.

Se dirigieron a la Unidad de Servicio Comunitario varios estudiantes para realizar modificaciones en cuanto al cambio de tutor comunitario, proyecto y comunidad. En conversaciones con los representantes de la Unidad de Servicio Comunitario, se llegó al acuerdo de permitirle al estudiante editar su inscripción de proyecto de Servicio Comunitario durante el período de inscripciones desde su perfil, de este modo evitar que los estudiante se dirijan a la Unidad.

### **3.13. Iteración 6**

Para la iteración 6 se continúa con el proceso de asignación de tutores académicos y se realiza la asignación de jurado para las presentaciones de los estudiantes sobre su práctica de Servicio Comunitario.

- **Planeación**

En la siguiente Tabla 10 se exponen los requerimientos planeados para la iteración 6.



Tabla 10 - Historia de Usuario para la iteración seis.

Nro.	Fecha Inicio	Fecha Final	Requerimientos	Estado
18	22/08/2010	23/08/2010	Editar inscripción del proyecto de Servicio Comunitario por parte de los estudiantes.	Finalizada
19	23/08/2010	27/08/2010	Gestionar la asignación de Tutores Académicos	Finalizada
20	28/08/2010	30/08/2010	Gestionar la asignación de Jurados	Finalizada

- **Diseño**

El diseño de la iteración 6 se apoya en el diagrama de flujo de la asignación de tutores de la iteración 5. Se presenta en la Figura 30 la interfaz de usuarios que se le muestra al representante de la Comisión de Servicio Comunitario por licenciatura, en la que se despliegan los docentes, su categoría, condición y dedicación, además de la cantidad de estudiante del cual ha sido tutor académico y finalmente la cantidad de estudiante que se le asignará.

Nro	Nombre	Cédula	Categoría	Condición	Dedicación	Alumnos Tutorados	Cantidad de Alumnos
1	Acosta alecia e	6814548	Titular	Sin Condición	Sin Dedicación		0
2	Acosta carlos	7267764	Instructor	Sin Condición	Dedicacion exclusiva	2	0
3	Alfonzo moises	7122690	Titular	Ad honores	Convencional 6 horas		0
4	Alonzo yajaira	50	Titular	Sin Condición	Sin Dedicación		0
5	Alvarez adonahis	9958083	Titular	Sin Condición	Sin Dedicación		0
6	Angelini rolando	16724379	Titular	Sin Condición	Sin Dedicación		0
7	Angulo rafael	13114353	Titular	Sin Condición	Sin Dedicación		0
8	Aponte hugo	11312763	Titular	Sin Condición	Sin Dedicación		0
9	Areyan enrique	17023919	Instructor	Contratado	Convencional 6 horas		0
10	Astudillo reinaldo	15207591	Titular	Sin Condición	Dedicacion exclusiva	1	0

Figura 30 - Asignación de carga estudiantil para cada docente.

En la asignación de tutores académicos, se le despliega al representante de la Comisión de Servicio Comunitario por licenciatura un listado de docentes según su escuela y la cantidad de alumnos que hay por el curso de inducción que seleccionó. Luego debe asignarle la carga al docente con un máximo de 10 estudiantes para cada uno. En la Figura 30 también se puede ver la cantidad de alumnos que han estado bajo la tutela de cada uno de los docentes, para realizar una asignación justa.

Si un docente accede a la aplicación y es tutor académico podrá ver la opción tutor académico en el menú principal y el listado de alumnos bajo su tutela, de esta forma poder obtener la información del estudiante como los son: la comunidad, proyecto, curso de inducción, informe y microproyecto, manteniendo al tutor informado del avance de su alumno (Ver Figura 31).

Castillo jimmy  
Cerrar sesión

Viernes, 22 de Octubre de 2010, a las 15:00:48

Inicio Curso de Inducción Tutor Académico Contacto

Usted es Tutor Académico de los siguientes Estudiantes

Nro	Cédula	Nombres	Apellidos	Teléfonos	Correo
1	18067262	Grener Jesus	Garcia Herrera	5555555 - 5555555	algo@servidor.com
2	15796273	Yadelsy Oddaly	Martin Chirinos	5555555 - 5555555	algo@servidor.com
3	15147042	MairhelÉ De los angeles	Carrillo Rojas	5555555 - 5555555	algo@servidor.com
4	17970558	Rohany Edith	Flores Smither	5555555 - 5555555	algo@servidor.com
5	15638059	Eduis Irardo	Guerra Barrios	5555555 - 5555555	algo@servidor.com
6	17779594	Clara Del valle	Palma González	5555555 - 5555555	algo@servidor.com
7	17218192	Ana Carina	Palma Gonzalez	5555555 - 5555555	algo@servidor.com
8	14451294	Arianny Victoria	Roca González	5555555 - 5555555	algo@servidor.com

Universidad Central de Venezuela/ Facultad de Ciencias/ Unidad de Servicio Comunitario Desarrollado por Grupo CONEST 2010

Figura 31 - Menú de un docente que es tutor académico con su listado de estudiantes.

- **Codificación**

Para la codificación de la asignación de tutor académico, primero se realiza una consulta a los docentes según el miembro de la comisión por licenciatura que haya ingresado en la aplicación, como se puede observar en la Figura 32. Si un usuario accede a la aplicación como miembro de la comisión de computación, se verifica su licenciatura y luego se despliega el listado de docentes activos para ser tutor académico.

```
@docente=Docente.find(:first,:conditions=>['cedula=?',session[:cedula]])
@licenciatura_nombre=@docente.licenciatura.nombre_corto

if(@licenciatura_nombre=="BIOLOGIA")
  @docente_tutor_academico=Docente.find(:all,:conditions => ['licenciatura_id = ?', "B"],:order=>'nombre')
  @licenciatura="B"
end
if(@licenciatura_nombre=="COMPUTACION")
  @docente_tutor_academico=Docente.find(:all,:conditions => ['licenciatura_id = ?', "C"],:order=>'nombre')
  @licenciatura="C"
end
if(@licenciatura_nombre=="FISICA")
  @docente_tutor_academico=Docente.find(:all,:conditions => ['licenciatura_id = ?', "F"],:order=>'nombre')
  @licenciatura="F"
end
if(@licenciatura_nombre=="GEOQUIMICA")
  @docente_tutor_academico=Docente.find(:all,:conditions => ['licenciatura_id = ?', "G"],:order=>'nombre')
  @licenciatura="G"
end
```

*Figura 32 - Consulta a los docentes según escuela.*

Al mostrarse el listado el miembro de la comisión puede cambiar la cantidad de estudiantes por docente y se realizará la asignación (ver Figura 33).

```

if (@cedula=nil)
  @tutor_cantidad = ServiciocomunitarioTutorAcademico.find(:first, :conditions => ['docente_cedula=?',@cedula])
  @tutor_cantidad.update_attributes(:cantidad_tutoriados =>@alumnos_nuevos_nro)
end

i +=1;
end

#----- Docentes que tienen tutoriados mas de 0 estudiantes y la licenciatura correspondiente
@docentes_disponibles=ServiciocomunitarioTutorAcademico.find(:all, :conditions => ['cantidad_tutoriados>?' AND licenciatura_id=?',0,params[:licenciatura]])

#----- Estudiantes que aprobaron el Curso de Induccion
@primera_letra=@docente.licenciatura_id
@estudiantes_aprobados=EstudianteCursoInduccion.find(:all,
:conditions => "tipo_nota_id>11 AND curso_induccion_id=#{params[:curso_induccion_id]} AND licenciatura_id=#{@primera_letra}")

if (@docentes_disponibles.size>0 && @estudiantes_aprobados.size>0)

  @tamano_arreglo_docentes=@docentes_disponibles.size
  @contador_size=0

  @estudiantes_aprobados.each do |estudiantes_aprobados|
    @docentes_cedulas=@docentes_disponibles[@contador_size,1]

    @docentes_cedulas.each do |docentes_cedulas|
      if(docentes_cedulas.cantidad_tutoriados>0)

```

Figura 33 - Parte del código de la asignación de tutores académicos.

A fin de ilustrar la asignación de tutores académicos se elaboraron las Figuras 34, 35 y 36.

Principalmente para la asignación de tutores académicos, se cargaron los estudiantes y los docentes en dos arreglos distintos. En la Figura 34 se puede visualizar el arreglo de docentes, en cada una de sus posiciones la cédula de identidad y la carga determinada Dx (CI,#), y para el estudiante sólo contiene la cédula de identidad Ex(CI).

Luego se recorren ambos arreglos en cada una de sus posiciones y cada vez que se le asigna un estudiante al tutor académico, se resta la cantidad de estudiantes que se le asignó como carga de ese docente, cuando esta cantidad llegue a cero significa que ya se le asignó la carga total a ese docente. En la Figura 34 se puede observar la primera pasada al arreglo de docentes.

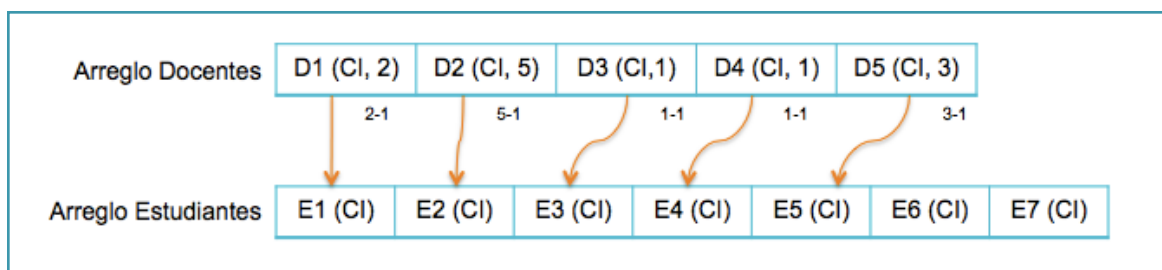


Figura 34 - Asignación de Tutores Académicos primera pasada.

Después que se realiza la primera pasada, se vuelve al D1 si tiene carga distinta de cero y se le asigna el próximo estudiante, igualmente se le resta el estudiante y se continúa con el siguiente docente (Ver Figura 35).

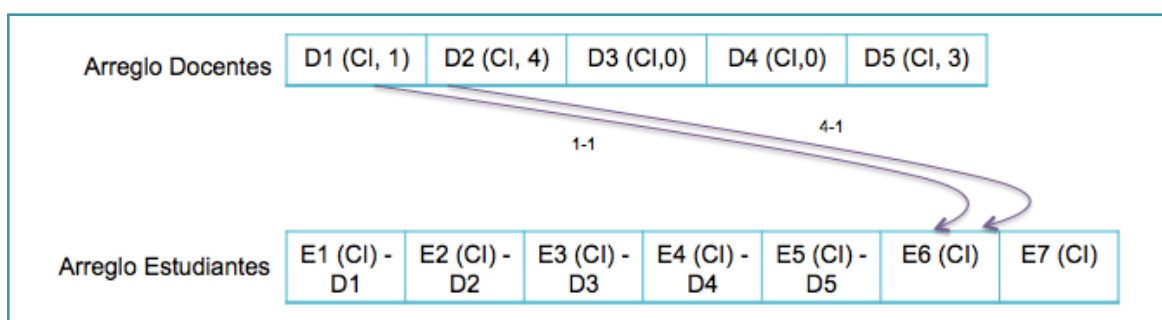


Figura 35 - Asignación de Tutores Académicos segunda pasada

Finalmente, cuando todos los estudiantes tienen asignado tutor académico se finalizan los recorridos y se muestra la asignación (Ver Figura 36).

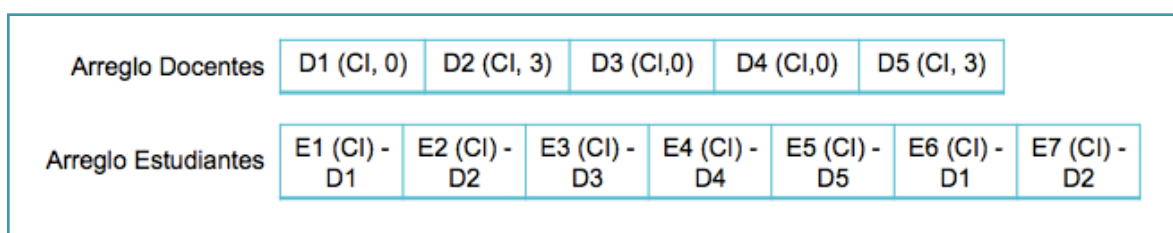


Figura 36 - Asignación Final de Tutores Académicos.

Una vez que se realiza la asignación final el representante de la Comisión de Servicio Comunitario de cada licenciatura tiene la opción de generar el listado de

asignación en formato PDF, además de enviar correos electrónicos a los tutores académicos informándoles sobre el alumno e igualmente a los alumnos sobre su tutor académico.

- **Prueba**

Se realizaron pruebas funcionales asignándole los tutores académicos a estudiantes de las diferentes licenciaturas, que hayan aprobado el curso de inducción II - 2010.

Se verificó por parte del administrador la asignación, y se seleccionó un docente para verificar que efectivamente aparecían los datos del estudiante asignado.

También se hicieron pruebas de aceptación el día 8 de octubre de 2010 a los miembros de la comisión de todas las licenciaturas, los cuales aprobaron la asignación de tutores académicos y de jurado.

Algunos estudiantes que quisieron modificar la inscripción de proyecto de Servicio Comunitario pudieron hacerlo dirigiéndose a la Unidad de Servicio Comunitario o a través de su perfil en la aplicación.

### **3.14. Iteración 7**

En la iteración 7, se abarcan los procesos finales de la realización del Servicio Comunitario, como lo son la elaboración del microproyecto, informe final y presentación, en los que se plasma las experiencias de los estudiantes durante el Servicio Comunitario.

- **Planeación**

En la Tabla 11, se pueden observar los requerimientos de la iteración 7.

*Tabla 11 - Historia de Usuario para la iteración siete.*



<b>Nro.</b>	<b>Fecha Inicio</b>	<b>Fecha Final</b>	<b>Requerimientos</b>	<b>Estado</b>
21	01/09/2010	04/09/2010	Crear y editar informe final de Servicio Comunitario.	Finalizada
22	05/09/2010	09/09/2010	Crear y editar microproyecto.	Finalizada
23	10/09/2010	14/09/2010	Cargar presentación de Servicio Comunitario y descargarla para editarla.	Finalizada

- **Diseño**

Para el diseño de la iteración 7 se propone completar dos formularios, uno para la elaboración del informe final y otro para el microproyecto.

Ambos formularios tienen los campos requeridos para ser aprobados por la Unidad de Servicio Comunitario, el tutor académico y el jurado. Una vez que los formularios son completados, el estudiante puede generarlo en formato PDF para su impresión u otra necesidad, igualmente el tutor académico y la Unidad de Servicio Comunitario.


En la Figura 37, se puede ver la interfaz de usuario del estudiante al momento de elaborar el informe final, y en la Figura 38, como se puede generar en formato PDF el documento.



**Servicio Comunitario**  
Facultad de Ciencias
Mariana Simon  
Cerrar sesión  



Viernes, 22 de Octubre de 2010, a las 14:46:47

Inicio
Curso de Inducción
Proyecto Servicio Comunitario
Microproyecto
Informe Final
Presentación

**Informe Final de Servicio Comunitario**

Fecha de Ejecución  /  /  

Fecha de Inicio  /  /  

Fecha de Culminación  /  /  

Nro de Horas de Dedicación

Título

El título del informe debe estar compuesto por el título del microproyecto y el nombre de la comunidad específica donde se realizó la prestación del servicio comunitario. Por ejemplo: Alfabetización Tecnológica en el Megainfocentro "Ramón Ismael Ramos" La Cañada y en la Universidad Bolivariana de Venezuela.

Figura 37 - Elaboración de Informe Final de Servicio Comunitario.


**Servicio Comunitario**  
Facultad de Ciencias
Mariana Simon  
Cerrar sesión  


Viernes, 22 de Octubre de 2010, a las 14:56:54

Se guardo correctamente el Informe final de Servicio Comunitario

Inicio
Curso de Inducción
Proyecto Servicio Comunitario
Microproyecto
Informe Final
Presentación

**Informe Final de Servicio Comunitario**

Informe	Editar	Generar PDF
Servicio comunitario en el infocentro		

Universidad Central de Venezuela/ Facultad de Ciencias/ Unidad de Servicio Comunitario    Desarrollado por Grupo CONEST 2010

Figura 38 - Opciones para eliminar o editar informe de Servicio Comunitario.

- **Codificación**

Uno de los requisitos finales para terminar el Servicio Comunitario es exponer a través de una presentación las experiencias vividas durante la práctica de Servicio Comunitario, por lo que el estudiante en la aplicación actual debe



cargar un archivo con extensión .ppt, .pptx, .odp o .pdf. Si el estudiante quiere editar la presentación puede hacerlo, descargándola para editarla, y luego volver a cargarla a través de la aplicación.

La codificación para este proceso se realiza primero cargando el archivo y luego se almacena en la tabla creada para las presentaciones de los estudiantes *serviciocomunitario\_presentacion*. Luego el estudiante puede descargar su presentación, al igual que el tutor académico, el jurado y la Unidad de Servicio Comunitario para su corrección y evaluar al estudiante.

En la Figura 39 se puede ver cómo se envía la presentación para su descarga con la función *send\_data*.

La función *send\_data* posee algunos parámetros como lo son filename, type, disposition y status, en la Figura 39 se puede observar los parámetros enviados para la presentación.

```
def descargar_presentacion
  if params[:CI]
    @presentacion = ServiociocomunitarioPresentacion.find(:first,:conditions=>["estudiante_cedula = ?",params[:CI]])
  else
    @presentacion = ServiociocomunitarioPresentacion.find(:first,:conditions=>["estudiante_cedula = ?",session[:cedula]])
  end
  @presentacion_sola=@presentacion.presentacion
  @tipo=@presentacion.extension_archivo
  send_data @presentacion_sola, :type => "application/vnd.openxmlformats-officedocument.presentationml.presentation", :disposition => "attachment"
end
```

Figura 39 - Descarga de presentación.

- **Prueba**

Se realizaron pruebas funcionales, se escogió un estudiante de la base de datos de prueba y desde su perfil se realizó un microproyecto, el informe final y finalmente la carga de su presentación.

En los tres casos se pudo editar los dos documentos y la presentación como se describió anteriormente.

Por parte del tutor académico y jurado se pudo descargar los documentos, al igual que desde el perfil de la Unidad de Servicio Comunitario de forma satisfactoria.

### 3.15. Iteración 8

La iteración 8 tiene como objetivo evaluar el informe final y la presentación, para la calificación final del Servicio Comunitario.

- **Planeación**

Uno de los propósitos fundamentales de esta iteración es la calificación de los estudiantes, ya que es necesario para la División de Control de Estudios cargar las notas antes de la culminación de cada semestre, durante una fecha específica (Ver Tabla 10).

*Tabla 12 - Historia de Usuario para la iteración ocho.*

Nro.	Fecha Inicio	Fecha Final	Requerimientos	Estado
24	15/09/2010	20/09/2010	Evaluar informe final y presentación de los estudiantes.	Finalizada
25	21/09/2010	30/09/2010	Enviar calificación final de los estudiantes.	Finalizada

• **Diseño**

Para evaluar a un estudiante se debe contestar un conjunto de preguntas, cada una de ellas tiene una ponderación de excelente, distinguido, suficiente y reprobado.

A un estudiante se le evalúa el informe final y la presentación del informe final, estas evaluaciones deben ser realizadas por el tutor académico del estudiante, el jurado asignado por el representante de la Comisión de Servicio Comunitario por licenciatura y el jurado que puede ser algún docente determinado por el consejo de escuela, miembro de la comisión o de la Unidad de Servicio Comunitario.

Para esta iteración se propone la interfaz de usuario de la Figura 40.

Freites m yeiremi  
Cerrar sesión

Viernes, 22 de Octubre de 2010, a las 15:02:47

Inicio | Curso de Inducción | Jurado Comisión | Contacto

**Por favor evalúe al estudiante:**  
Jesus Miguel Mata Mora  
Cédula: 16033952

**Recuerde que debe seleccionar todos los criterios de evaluación**

Aspectos a Evaluar	Excelente	Distinguido	Suficiente	Reprobado
<b>Datos del Proyecto</b>				
Describe claramente la comunidad en la que participó: ubicación, conformación, aspectos característicos.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Describe claramente el entorno físico en que se prestó el servicio: infraestructura, recursos disponibles.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Describe la situación en la que participó identificando la problemática social en que se inserta.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Identifica y expresa claramente el problema específico a enfrentar en la comunidad.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Identifica las variables que intervienen en el problema y su posible correlación.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Describe claramente los objetivos planteados a fin de alcanzar la solución del problema.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Figura 40 - Evaluación del Informe Final.

El usuario que va a evaluar al estudiante debe seleccionar una ponderación que desee por cada pregunta. Al completar la evaluación se registra la calificación, y la Unidad de Servicio Comunitario puede visualizar la nota de cada estudiante. Cuando concluye la evaluación del estudiante, la nota se envía automáticamente a la División de Control de Estudios.

- **Codificación**

Para calcular y guardar en la base de datos la nota de la evaluación del informe final de un estudiante, primero se verifica que la evaluación se realizó correctamente.

Después se tomaron las ponderaciones determinadas por el tutor académico y los jurados por pregunta evaluada, y se fueron agregando a un arreglo *@resultado*, para contar la cantidad de cada una de estas. Por último la ponderación que más se repita entre ellos es la que determina si fue aprobado o aplazado el Servicio Comunitario de ese estudiante (Ver Figura 41).

```
def guardar_evaluacion_estudiante
  @usuario=params[:usuario]

  if(params[:evaluacion_1]==nil || params[:evaluacion_2]==nil || params[:evaluacion_3]==nil || params[:evaluacion_4]==nil ||
  params[:evaluacion_5]==nil || params[:evaluacion_6]==nil || params[:evaluacion_7]==nil || params[:evaluacion_8]==nil ||
  params[:evaluacion_9]==nil || params[:evaluacion_10]==nil || params[:evaluacion_11]==nil || params[:evaluacion_12]==nil ||
  params[:evaluacion_13]==nil || params[:evaluacion_14]==nil || params[:evaluacion_15]==nil || params[:evaluacion_16]==nil ||
  params[:evaluacion_17]==nil || params[:evaluacion_18]==nil || params[:evaluacion_19]==nil || params[:evaluacion_20]==nil)

    flash[:error] = 'Debe realizar toda la evaluación'
    redirect_to :controller => 'docente', :action => 'evaluar_estudiante',:estudiante_cedula=>params[:estudiante_cedula]
  end

  i = 1;
  @resultado=[]

  while i < 21 do
    @aux=i
    @evaluacion="evaluacion_#{@aux}"

    if(params[@evaluacion]=="e")
      @resultado.concat( ["d"]) # EXCELENTE
    else
      if(params[@evaluacion]=="d")
        @resultado.concat( ["c"]) # DISTINGUIDO
      else
        if(params[@evaluacion]=="s")
          @resultado.concat( ["b"]) # SUFICIENTE
        else
          if(params[@evaluacion]=="r")
            @resultado.concat( ["a"]) # REPROBADO
          end
        end
      end
    end
    i +=1;
  end
end
```

Figura 41 - Calcular el resultado de la evaluación del estudiante.

Para la evaluación de la presentación se realizó el mismo proceso descrito anteriormente.

- **Prueba**

Se realizaron pruebas funcionales, evaluando a un estudiante que haya entregado informe final y cargado la presentación final.

Se evaluó al estudiante desde el perfil de su tutor académico, su jurado y se verificó la calificación desde el administrador de la Unidad de Servicio Comunitario.

## Conclusiones

El objetivo propuesto de este Trabajo Especial de Grado se cumplió satisfactoriamente, ya que se logró la automatización de los procesos relacionados al curso de inducción y el proyecto de Servicio Comunitario, optimizando los tiempos de espera y la organización de la información que se maneja en la Unidad de Servicio Comunitario de la Facultad de Ciencias de la Universidad Central de Venezuela.

Debido a la integración con el Sistema de Gestión Académica CONEST, se dispone de la información actualizada de los estudiantes y docentes. Además, el hecho de seguir los estándares de CONEST permitirá a futuros grupos de trabajo efectuar actividades de mantenimientos correctivos, preventivos y adaptativos a la aplicación de una forma más simple.

Con respecto al método de desarrollo de software ágil Programación Extrema, se puede concluir que se adaptó a las necesidades del grupo de trabajo. El método utilizado permite la interacción continua con el cliente para ajustarse a los requerimientos que planteen, lo cual facilitó determinar la solución más conveniente. Asimismo la Programación Extrema propone la programación en pareja, esto contribuyó al desarrollo de soluciones más apropiadas. La práctica de la integración continua establece integrar funcionalidades de forma frecuente a la aplicación, lo que permitió probar pequeños cambios de desarrollo de manera oportuna.

El uso de las tecnologías de software libre como Ruby on Rails, MySQL, librerías de JavaScript, hojas de estilo CSS, trajo como beneficio el bajo costo del desarrollo de la aplicación, además de otras ventajas como independencia tecnológica que permite el acceso al código fuente para el desarrollo de nuevos productos.

El desarrollo de la aplicación se ajustó al cronograma propuesto por la Unidad de Servicio Comunitario, lo cual contribuyó a realizar las distintas pruebas funcionales y de aceptación al curso de inducción y proyecto del Servicio Comunitario durante los dos períodos lectivos del año 2010.

En conclusión la Unidad de Servicio Comunitario cuenta con una aplicación Web que hace que sus actividades relacionadas a los procesos de curso de inducción y proyecto del Servicio Comunitario sean más organizadas, fáciles y rápidas.

Como recomendaciones para trabajos futuros relacionados con el Servicio Comunitario se propone, agregar módulos de integración con las comunidades, con el fin de mantener comunicación continua con los responsables de las comunidades y sus beneficiarios. Incluir nuevas funcionalidades de solicitud en línea, tales como, inclusión de proyectos, comunidades, cambios de tutor académico, entre otros. Y finalmente crear una red colaborativa para que los estudiantes compartan sus experiencias.

## Referencias Bibliográficas

- (Ley de Servicio Comunitario del Estudiante de Educación Superior, 2005) - (Septiembre 2005) - Ley de Servicio Comunitario del Estudiante de Educación Superior - [www.usb.ve/proyectar/pdf/isceu.pdf](http://www.usb.ve/proyectar/pdf/isceu.pdf)
- (Acosta, Eleonora 2009) - Acosta Eleonora, I.T. (Julio 2009). Servicio Comunitario en la Facultad de Ciencias – UCV – [http://www.ciens.ucv.ve/ciencias/servicio\\_comunitario/index.htm](http://www.ciens.ucv.ve/ciencias/servicio_comunitario/index.htm)
- (Lujan, Sergio 2001) - Sergio Luján Mora (Editorial Club Universitario, 2001). Programación en Internet: cliente web - <http://books.google.co.ve/books?id=2VxcwMst6RYC&dq=lujan+sergio+2001>
- (Cobo, Ángel 2005) - Ángel Cobo, PHP y MySQL- tecnologías para el desarrollo de aplicaciones web (Ediciones Díaz de Santos, 2005) - <http://books.google.co.ve/books?id=zMK3GOMOpQ4C&dq=inauthor:%22Ángel+Cobo%22>
- (Tópicos para el desarrollo de una aplicación Web que automatice los procesos de la Unidad de Servicio Comunitario de la Facultad de Ciencias, 2009) - Mariana Simón, Jesús Mata (Diciembre 2009)
- (Servicios Web, W3C) - W3C.(S.F) W3C Servicios Web – <http://www.w3c.es/divulgacion/guiasbreves/serviciosWeb>
- (Fitzgerald, Michael 2007) - Michael Fitzgerald, (O'Reilly Media, Inc., 2007) - Learning Ruby - [http://books.google.co.ve/books?id=pYS\\_Fm5LqUYC&dq=fitzgerald,+michael+2007](http://books.google.co.ve/books?id=pYS_Fm5LqUYC&dq=fitzgerald,+michael+2007)
- (Tate, Bruce 2006) - Bruce Tate, (O'Reilly Media, Inc., 2006) - Ruby on rails: up and running - <http://books.google.co.ve/books?id=pUPX8joph10C&dq=tate,+bruce+2006>
- (Thomas, David 2006) - David Thomas, (Pragmatic Bookshelf, 2006) - Agile web development with rails -



- [http://books.google.co.ve/books?id=ue5SAAAAMAAJ&q=thomas,+david+2005&dq=thomas,+david+2005&hl=es&ei=zK6yTILcCIL\\_8AaJq-2dCQ&sa=X&oi=book\\_result&ct=result&resnum=1&ved=0CCsQ6AEwAA](http://books.google.co.ve/books?id=ue5SAAAAMAAJ&q=thomas,+david+2005&dq=thomas,+david+2005&hl=es&ei=zK6yTILcCIL_8AaJq-2dCQ&sa=X&oi=book_result&ct=result&resnum=1&ved=0CCsQ6AEwAA)
- (Wirdemann, Ralf 2007) - Ralf Wirdemann, (Hanser Verlag, 2007) - Rapid Web Development mit Ruby on Rails -  
[http://books.google.co.ve/books?id=BL\\_pDUhiX5cC&dq=wirdemann,+ralf+2007](http://books.google.co.ve/books?id=BL_pDUhiX5cC&dq=wirdemann,+ralf+2007)
  - (Kenneth, Laudon 2004) - Kenneth C. Laudon, (Pearson Educación, 2004) - Sistemas de información gerencial: administración de la empresa digital -  
<http://books.google.co.ve/books?id=PmnhMJpfsu8C&dq=kenneth,+laudon+2004>
  - (Pressman, Roger 2006) - Roger S. Pressman, (McGraw-Hill, 2006) - Ingeniería del software: un enfoque práctico -  
[http://books.google.co.ve/books?id=rEoxQQAACAAJ&dq=pressman+roger+2006&hl=es&ei=31CyTP\\_RLoG78gbXwtmhCQ&sa=X&oi=book\\_result&ct=result&resnum=1&ved=0CCoQ6AEwAA](http://books.google.co.ve/books?id=rEoxQQAACAAJ&dq=pressman+roger+2006&hl=es&ei=31CyTP_RLoG78gbXwtmhCQ&sa=X&oi=book_result&ct=result&resnum=1&ved=0CCoQ6AEwAA)
  - (Sommerville, Ian 2005) - Ian Sommerville, (Pearson Educación, 2005) - Ingeniería del software -  
<http://books.google.co.ve/books?id=gQWd49zSut4C&dq=sommerville,+ian+2005>
  - (kendall, Kenneth 2006) - Kenneth E. Kendall, (Pearson/Prentice Hall, 2006) - Systems analysis and design -  
[http://books.google.co.ve/books?id=F3JGAAAYAAJ&q=kendall,+kenneth+2006&dq=kendall,+kenneth+2006&hl=es&ei=1VGyTMHHH8KC8gal5\\_mnCQ&sa=X&oi=book\\_result&ct=result&resnum=10&ved=0CFcQ6AEwCQ](http://books.google.co.ve/books?id=F3JGAAAYAAJ&q=kendall,+kenneth+2006&dq=kendall,+kenneth+2006&hl=es&ei=1VGyTMHHH8KC8gal5_mnCQ&sa=X&oi=book_result&ct=result&resnum=10&ved=0CFcQ6AEwCQ)
  - (Fowler, Martin 2000) - Martin Fowler, (Addison-Wesley, 2000) - UML distilled: a brief guide to the standard object modeling language -  
<http://books.google.co.ve/books?id=nHZsISr1gJAC&dq=fowler,+martin+2000>

- (Glass, Robert 2001) - Robert L. Glass, (Addison-Wesley, 2001) - Facts and fallacies of software engineering - <http://books.google.co.ve/books?id=3Ntz-UJzZN0C&dq=glass,+robert+2001>
- (Shklar & Rosen, 2009) - Web Application Architecture. Wiley.