



Universidad Central de Venezuela  
Facultad de Ciencias  
Escuela de Computación

**Simbiosis: Aplicación Web para la  
Gestión del Proceso de Inscripción de Seminarios y  
Trabajo Especial de Grado de la Escuela de Biología  
de la Universidad Central de Venezuela**

Trabajo Especial de Grado  
Presentado ante la ilustre  
Universidad Central de Venezuela  
Por el licenciado:  
Francisco J. Vergara S.

Tutores:  
Profa. Yosly Hernández  
Prof. Jesús Romero

Caracas, mayo de 2015

## Acta

Quienes suscriben, miembros del Jurado designado por el Consejo de Escuela de Computación de la Facultad de Ciencias de la Universidad Central de Venezuela, para examinar el Trabajo Especial de Grado presentado por el bachiller Francisco Javier Vergara Sandoval, portador de la Cédula de Identidad V-17.389.551, con el título: “**Simbiosis: Aplicación Web para la Gestión del Proceso de Inscripción de Seminarios y Trabajo Especial de Grado de la Escuela de Biología de la Universidad Central de Venezuela**”, a los fines de cumplir con el requisito legal para optar al título de Licenciado en Computación, dejan constancia de lo siguiente:

Leído el trabajo por cada uno de los miembros del jurado, se fijó el día dieciocho (18) a las 04:00pm, para que sus autores lo defiendan en forma pública, lo que hicieron en la Sala I de la Escuela de Computación, Facultad de Ciencias de la Universidad Central de Venezuela, lo cual se realizó mediante una presentación oral de su contenido, luego de lo cual respondieron satisfactoriamente las preguntas formuladas por el Jurado, todo ello conforme a lo dispuesto en la Ley de Universidades y demás normativas vigentes de la Universidad Central de Venezuela. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobarlo con la nota de \_\_\_\_\_ puntos.

En fe de lo cual se levanta la presente Acta, en Caracas a los dieciocho (18) días del mes de mayo del año dos mil quince (2015).

\_\_\_\_\_  
**Profa. Yosly Hernández Bieliukas**

**Tutora Firmante**

\_\_\_\_\_  
**Prof. Jesús Romero**

**Tutora**

\_\_\_\_\_  
**Profa. Alexander Laurentin**

**Jurado Principal**

\_\_\_\_\_  
**Profa. Mercy Ospina**

**Jurado Principal**

## **Dedicatoria**

A mis padres y familia quienes con mucho esfuerzo, trabajo y sacrificio forjaron éste logro.

Los amo.

## **Agradecimientos**

A la Universidad Central de Venezuela, como la casa que vence las sombras. A nuestra escuela de Computación por inspirarnos y enseñarnos valores, principios, y ética profesional.

A mis hermanos, por brindarme su apoyo moral y ayuda en todo momento.

A mi tutor, Jesús Romero, por ser ejemplo de compromiso, dedicación, entrega, y sabiduría, por su paciencia y apoyo al atender puntualmente mis dudas; mi respeto y admiración para usted.

A mi tutora, Yosly Hernández, por su paciencia a lo largo del desarrollo de este trabajo, siempre estaré agradecido contigo.

A todos los que colaboraron de alguna u otra manera en mi formación profesional y en la realización de la tesis.

A Stackoverflow, por brindarme su apoyo absoluto y ayuda ilimitada, por atender mis dudas sin importar lo tontas que fuesen.

## Resumen

En la Escuela de Biología intervienen un conjunto de procesos los cuales destacan por su gran flujo de información y complejidad. Uno de estos procesos es el relacionado con la formalización de la inscripción de las asignaturas: Seminario (I y II) y Trabajo Especial de Grado, para los alumnos de pregrado. Dicho proceso requiere la automatización y optimización con el uso de las Tecnologías de Información y Comunicación (TIC), con el fin de gestionar de una mejor manera la información de los involucrados en las actividades, beneficiando a la comunidad estudiantil, docente y el personal involucrado. Con el propósito de apoyar el trabajo en la Escuela de Biología, se plantea como objetivo del presente Trabajo Especial de Grado el desarrollo e implementación de una aplicación web que gestione el conjunto de procesos relacionados con las actividades que se realizan referente a la inscripción de Seminarios (I y II) y TEG; con la finalidad de tener a la disposición una herramienta actualizada que brinde soporte al personal involucrado, para mejorar los tiempos de respuesta, brindando mayores beneficios tanto para la organización como para los usuarios y aumentar así su satisfacción, y al mismo tiempo optimizar los costos y los tiempos de respuesta. Como producto final, se logró desarrollar e implementar una aplicación web que logra optimizar dichos procesos, automatizando sus actividades. En el siguiente trabajo, para cada uno de los módulos desarrollados, se explica en detalle su proceso de desarrollo. Además se explica la metodología empleada la cual es una adaptación de la metodología ágil denominada AgilUs, incluyendo el uso de tecnologías de desarrollo como Html, PHP, CSS, JavaScript, Bootstrap y MySQL.

**Palabras Claves:** Escuela de Biología, Aplicación Web, Solicitudes Estudiantiles, Automatización, estudiantes, docentes, Simbiosis.

# Índice General

Resumen.....	iii
Índice de Figuras.....	v
Índice de Tablas.....	xi
Introducción.....	1
<b>Capítulo1: Problema de Investigación</b>	
1.1 Contexto de la Investigación.....	3
1.2 Proceso de Formalización.....	6
1.3 Planteamiento del Problema .....	10
1.4 Justificación.....	11
1.5 Objetivo General .....	13
1.6 Objetivos Específicos.....	13
1.7 Alcance.....	13
1.8 Metodología de Desarrollo.....	13
1.9 Antecedentes.....	16
<b>Capítulo 2: Marco Conceptual</b>	
2.1 Aplicaciones Web.....	17
2.2 Características de una aplicación Web.....	18
2.3 Arquitectura Cliente-Servidor.....	20
2.3.1 Componentes de la Arquitectura Cliente-Servidor.....	21
2.3.2 Ventajas de la Arquitectura Cliente-Servidor.....	21
2.4 Patrón de Diseño Modelo Vista Controlador.....	23
2.4.1 Flujo de Control en el Modelo Vista Controlador.....	23
2.5 Sistema Manejador de Base de Datos.....	24
2.5.1 MySQL.....	25
2.5.2 Características de MySQL.....	25
2.5.3 Limitaciones de MySQL.....	26
2.6 Tecnologías Web.....	26
2.6.1 Lenguaje de Marcado de Hipertexto (HTML) .....	26

2.6.2	Hojas de Estilo en Cascada (CSS) .....	23
2.6.3	Twitter Bootstrap.....	27
2.6.4	JavaScript (JS) .....	28
2.6.5	JQuery.....	28
2.6.6	JQuery UI.....	29
2.6.7	AngularJS.....	32
2.6.8	Servidor Web Apache.....	33
2.6.9	Lenguaje Preprocesador de Hipertexto.....	33
2.6.10	CoffeeScript.....	35
2.7	Software de Control de Versiones Git.....	35
2.8	BitBucket.....	36
2.9	Licenciamiento Creative Commons.....	36

### **Capítulo 3: Marco Aplicativo**

3.1	Fase I: Requisitos.....	38
3.1.1	Entrevista.....	38
3.1.2	Tormenta de ideas.....	39
3.1.3	Sondeo.....	40
3.1.4	Perfiles de Usuario.....	38
3.1.5	Requerimientos Funcionales y No funcionales.....	48
3.2	Fase II: Análisis.....	52
3.2.1	Prototipos de Interfaz.....	52
3.2.2	Guía de Estilo.....	54
3.2.3	Modelado de los Casos de Uso.....	57
3.2.4	Modelo de Objetos del Dominio.....	60
3.2.5	Diseño Lógico y Físico de la Base de Datos.....	61
3.2.6	Plataforma de Desarrollo.....	62
3.3	Fase III: Prototipaje.....	63
3.3.1	Desarrollo.....	63
3.3.2	Lista de Comprobación.....	67
3.4	Fase IV: Entrega.....	69
3.4.1	Pruebas de Usabilidad.....	69

3.4.2 Pruebas Cross Browser.....	72
3.4.3 Pruebas de Instalación.....	75
3.4.4 Flujo de Trabajo del Proceso Automatizado.....	75

**Capítulo 4: Resultados**

4.1 Página Principal.....	77
4.2 Menú de Funcionalidades.....	79
4.3 Gestión de Usuarios.....	80
4.4 Módulo de Estudiantes.....	84
4.5 Módulo de Docentes.....	90
Conclusiones.....	101
Recomendaciones.....	103
Referencias Bibliográficas.....	104
ANEXO A. Casos de Uso.....	107
ANEXO B Cuestionario de Usabilidad.....	122
ANEXO C Resultados Pruebas de Usabilidad.....	123
ANEXO D Manual de Usuario.....	124
ANEXO E Manual de Usuario.....	132



## Índice de Figuras

Figura 1.1: Organigrama Estructural de la Escuela de Biología.....	5
Figura 1.2: Proceso Manual de Inscripción de Seminarios y TEG.....	10
Figura 1.3: AgilUs: Etapas, Actividades y Artefactos.....	15
Figura 2.1: Interfaz del Menú Principal de Estudiantes en CONEST.....	18
Figura 2.2: Sistema para la inscripción de proyectos de pasantías.....	18
Figura 2.3: Esquema general de la arquitectura Cliente-Servidor.....	20
Figura 2.4: Patrón de Diseño Modelo Vista Controlador MVC.....	23
Figura 2.5: El esquema de un sistema basado en el patrón software MVC.....	24
Figura 2.6: Ejemplo de un menú con efecto de acordeón.....	30
Figura 2.7. Ejemplo de un cuadro de Diálogo.....	31
Figura 2.8. Ejemplo de un calendario implementando con DatePicker.....	31
Figura 2.9: Modelo MVC con AngularJS.....	32
Figura 2.10: Tipos de Licencias de Creative Commons Venezuela.....	37
Figura 3.1 Prototipo de Interfaz planteada en la tormenta de ideas.....	40
Figura 3.2: Rol que cumplen los encuestados en la Facultad de Ciencias.....	41
Figura 3.3: Evaluación de inscripción de materias mediante procesos manuales.....	42
Figura 3.4: Características deseables de la aplicación web.....	43
Figura 3.5: Preferencias de visualización del menú de opciones.....	43
Figura 3.6: Gama de colores considerada adecuada para la aplicación.....	44
Figura 3.7: Opción de entregas digitales en la aplicación web.....	45
Figura 3.8: Redes sociales consideradas adecuadas para la aplicación.....	45
Figura 3.9: Redes sociales consideradas adecuadas para la aplicación.....	46
Figura 3.10: Tipos de usuarios y roles en la aplicación web.....	47

Figura 3.11: Vista general de inicio de sesión en la aplicación.....	53
Figura 3.12: Vista general del Menú de Usuario.....	54
Figura 3.13: Caso de uso de la aplicación Simbiosis. Nivel 0.....	54
Figura 3.14: Caso de uso de la aplicación Simbiosis. Nivel 1.....	54
Figura 3.15: Caso de uso de la aplicación Simbiosis. Actor Estudiante.....	55
Figura 3.16: Modelo de Objetos del dominio.....	60
Figura 3.17: Diseño Lógico y Físico de la Base de Datos.....	61
Figura 3.18: Tecnologías empleadas.....	63
Figura 3.19: Conexión a la BD de Simbiosis.....	64
Figura 3.20: Creación de los roles en la aplicación.....	64
Figura 3.21: Código para la Creación de usuarios de forma manual.....	65
Figura 3.22: Fragmento de código para el inicio de sesión.....	65
Figura 3.23: Código para el template base de la aplicación.....	66
Figura 3.24: Estructura de controladores de la aplicación.....	67
Figura 3.25: Vista de la aplicación Simbiosis en Google Chrome.....	74
Figura 3.26: Vista de la aplicación Simbiosis en Safari.....	74
Figura 3.27: Flujo de Trabajo - Proceso de Inscripción Automatizado.....	76
Figura 4.1: Página principal de la aplicación Simbiosis.....	77
Figura 4.2: Formulario para el inicio de sesión.....	78
Figura 4.3: Formulario para cambiar de contraseña.....	78
Figura 4.4: Funcionalidad para recuperar contraseña.....	78
Figura 4.5: Correo de recuperación de contraseña.....	79
Figura 4.6: Opciones para el usuario Estudiante.....	79
Figura 4.7: Opciones para el usuario Dirección.....	80

Figura 48: Opciones para el usuario Profesor.....	80
Figura 4.8: Opción de Agregar Estudiante.....	81
Figura 4.9: Formulario para Agregar Estudiante.....	81
Figura 4.10: Opción para Cargar Listados.....	81
Figura 4.11: Formulario para Cargar Listado de Estudiante.....	82
Figura 4.12: Opción de Listado de Docentes.....	82
Figura 4.13 Botones para agregas, editar y eliminar usuario.....	83
Figura 4.14: Ventana emergente para eliminar usuario.....	83
Figura 4.15: Ventana emergente para crear usuario.....	83
Figura 4.16: Ventana emergente para editar usuario.....	84
Figura 4.17: Ventana emergente para Actualizar datos.....	84
Figura 4.18: Opción de formalizar inscripción.....	85
Figura 4.19: Opción de Cargar Documento.....	85
Figura 4.20: Opción de imprimir planilla.....	86
Figura 4.21: Planilla de formalización.....	87
Figura 4.22: Lista de Observaciones.....	88
Figura 4.23: Detalle de Observación.....	88
Figura 4.24: Formulario para cambiar contraseña.....	89
Figura 4.25: Opción para ver las notificaciones de la aplicación.....	89
Figura 4.26: Listado de las notificaciones de la aplicación.....	89
Figura 4.27: Cuadro de diálogo para finalizar sesión.....	90
Figura 4.28: Opción para listar las solicitudes de tutoría.....	90
Figura 4.29: Cuadro de diálogo para confirmar rechazo de solicitud.....	91
Figura 4.30: Opción de tutoría en el perfil docente.....	91

Figura 4.31: Ventana de diálogo para confirmar el documento cargado.....	92
Figura 4.32: Ventana de diálogo para rechazar el documento cargado.....	92
Figura 4.33: Opción para Cargar Notas en la aplicación.....	93
Figura 4.34: Ventana para calificar al estudiante.....	93
Figura 4.35: Ventana para listar planillas del departamento.....	93
Figura 4.36: Ventana para listar las notas cargadas.....	94
Figura 4.37: Funcionalidad para enviar comunicados.....	94
Figura 4.38: Menú de configuración de la aplicación.....	95
Figura 4.39: Funcionalidad de configuración de la aplicación.....	95
Figura 4.40: Ventana para escoger docentes.....	96
Figura 4.41: Funcionalidad de configuración de Representantes de la UDPP.....	96
Figura 4.42: Funcionalidad para iniciar el proceso de formalización.....	97
Figura 4.43: Funcionalidad para detener o finalizar el proceso de formalización.....	97
Figura 4.44: Funcionalidad para generar reportes y estadísticas.....	98
Figura 4.45: Reporte generado mediante la aplicación.....	98
Figura 4.46: Opciones para generar reportes.....	99
Figura 4.47: Funcionalidad para generar reportes por departamentos.....	99
Figura 4.48: Funcionalidad para generar reportes individuales.....	99
Figura 4.49: Reporte generado individualmente.....	100

## Índice de Tablas

Tabla 3.1.: Guía de estilo de la aplicación. Parte A.....	54
Tabla 3.1.: Guía de estilo de la aplicación. Parte B.....	55
Tabla 3.3: Especificación del caso de uso: Iniciar Sesión.....	59
Tabla 3.4: Especificación del caso de uso: Recuperar Contraseña.....	59
Tabla 3.5: Lista de comprobación para la aplicación. Parte A.....	78
Tabla 3.6: Lista de comprobación para la aplicación. Parte B.....	79
Tabla 3.7: Descripción de criterios de usabilidad. Parte A.....	79
Tabla 3.8: Descripción de criterios de usabilidad. Parte B.....	80
Tabla 3.9 Resultados de las Pruebas de Usabilidad. Parte A.....	81
Tabla 3.10 Resultados de las Pruebas de Usabilidad. Parte A.....	82
Tabla 3.7: Resultados de las Pruebas de Usabilidad. Parte B.....	71
Tabla 3.11. Descripción de pruebas Cross Browser para Simbiosis.....	83

## Introducción

Debido al auge tecnológico de la última década, son muchas las organizaciones que aprovechan el uso de las tecnologías web para difundir su información institucional, dar a conocer sus actividades a un grupo de usuarios y automatizar sus procesos internos mediante el uso de aplicaciones que faciliten y ayuden en sus actividades cotidianas.

Dichas aplicaciones son programas que se ejecutan en servidores y utilizan páginas web como interfaz del lado del cliente. Este tipo de software es sencillo, económico, confiable, seguro y ofrecen la posibilidad de ser accedido desde cualquier lugar sólo con un dispositivo que cuente con un navegador y acceso a Internet.

En la Dirección de la Escuela de Biología de la Facultad de Ciencias de la UCV, el proceso de Inscripción de Seminarios (I y II) y Trabajo Especial de Grado (TEG), se realiza de forma manual por parte del personal involucrado en el mismo (secretarías, profesores, estudiantes, entre otros), los cuales invierten mucho tiempo y esfuerzo en realizarlos, y eventualmente presentan inconvenientes y errores, debido a la forma y complejidad con que se están llevando a cabo las actividades.

Debido a esto, surge la necesidad de trabajar en conjunto con la Escuela de Biología para desarrollar e implementar una aplicación web que permita automatizar y gestionar dicho proceso de inscripción, para mejorar la calidad y eficiencia del mismo, brindando mayores beneficios tanto para la organización como para los usuarios.

La aplicación debe estar al alcance del personal que participa en las distintas fases del proceso de inscripción (personal administrativo, alumnos, profesores, tutores, jurados y representantes de distintas comisiones inherentes al mismo).

El siguiente trabajo de investigación está estructurado en los siguientes capítulos:

- Capítulo 1: Problema de investigación. Se presenta una reseña histórica de la Escuela de Biología de la Facultad de Ciencias de la UCV, haciendo énfasis en su organización y en el proceso de inscripción de seminarios y TEG. Adicionalmente, se desarrolla el contexto del problema, se plantea la solución a la problemática planteada en el problema de investigación, se presentan los objetivos generales y específicos, la justificación del proceso involucrado en el problema. Finalmente se presenta el alcance

de la aplicación desarrollada en el TEG, que expone las funcionalidades que la aplicación estaría en la capacidad de ofrecer.

- Capítulo 2: Marco Conceptual. En este capítulo se describen conceptos básicos acerca de las tecnologías web de vanguardia que serán utilizadas para el desarrollo del TEG planteado en esta investigación. Se agrupan en tecnologías del lado del cliente, conformadas por HTML, CSS, JavaScript, jQuery y AngularJS; y tecnologías del lado del servidor como PHP y CoffeeScript. Cada una de ellas facilita el desarrollo de aplicaciones web.
- Capítulo 3: Marco Aplicativo. Se describen los aspectos más relevantes relacionados a la metodología de desarrollo de software AgilUs y se especifican las bases y las distintas etapas que la componen. Asimismo, se describen y explican las fases de requisitos y análisis del método aplicado en el TEG, detallando las actividades realizadas, los artefactos generados, las pruebas realizadas y las entregas efectuadas.
- Capítulo 4: Resultados. Se exponen los resultados de este Trabajo Especial de Grado (TEG), mostrando cada una de las funcionalidades e interfaces de la aplicación desarrollada y se demuestra así que los objetivos propuestos fueron logrados.

Finalmente, se presentan las conclusiones y recomendaciones del presente trabajo, así como también las referencias bibliográficas utilizadas durante el desarrollo de este trabajo.

# Capítulo 1

## Problema de Investigación

En el presente capítulo se proporciona una explicación detallada del proceso de inscripción de Seminarios (I y II) y Trabajo Especial de Grado (TEG) en la escuela de biología de la facultad de ciencias de la UCV, de igual forma se presenta el objetivo general y los objetivos específicos propuestos del trabajo especial de grado y se describe la justificación, alcance e importancia del mismo.

### 1.1 Contexto de la Investigación

La Escuela de biología surge en el año 1946, con la creación del departamento de ciencias naturales, un proyecto que fue presentado por el Dr. Tobias Lasser a petición del Dr. Luis Manuel Peñalver, con la misión de formar recursos humanos para prestar asesoría técnica, desempeñarse en industrias, docencia, investigación básica y aplicada, utilizando los resultados obtenidos en la solución de problemas biomédicos, agroecológico, alimentarios, zoológicos, ambientales, de producción marina y otros relacionados con los seres vivos.

En el organigrama de la escuela de Biología se representa al consejo de escuela como la máxima autoridad, a continuación le sigue la dirección de la escuela de biología la cual está conformada por: la Unidad Docente de Prácticas Profesionales y el Personal Administrativo. Los cinco (5) departamentos quienes manejan la docencia; y otras dependencias adscritas cuyo desempeño principal es el área de investigación y el área de servicio, ellas son: el Centro de Microscopía Electrónica, el Laboratorio de Fotografía, el Laboratorio de Ecología y Plantas Epífitas, el Laboratorio de Invertebrados.

El consejo de escuela está constituido por el director quien lo preside y ejecuta las decisiones emanadas del consejo. Los cinco (5) jefes de departamento (uno (1) por departamento), cinco (5) representantes de los profesores y dos (2) representantes estudiantiles. Actualmente la escuela de biología está conformada por cinco (5) departamentos los cuales son: Biología Celular, Botánica, Ecología, Tecnología de Alimentos y Zoología, los cuales se detallan a continuación:



El departamento de Biología Celular de la Escuela de Biología está conformado por las unidades docentes de: Biología Celular, Bioquímica y Genética.

El departamento de Botánica mantiene relaciones de docencia e investigación con otras dependencias de la UCV como la Facultad de Agronomía, la Fundación Instituto Botánico de Venezuela, Instituto Venezolano de Investigaciones Científicas, la Universidad de los Andes y la Universidad Simón Bolívar. Consta de dos (2) Unidades Docentes: Biología Vegetal y Fisiología Vegetal.

El Departamento de Ecología está conformado por cuatro (4) Unidades docentes: Unidad Docente Ecología Aplicada, Unidad Docente Ecología Terrestre, Unidad Docente Ecología Acuática y la Unidad Docente Bioestadística y Comportamiento Animal.

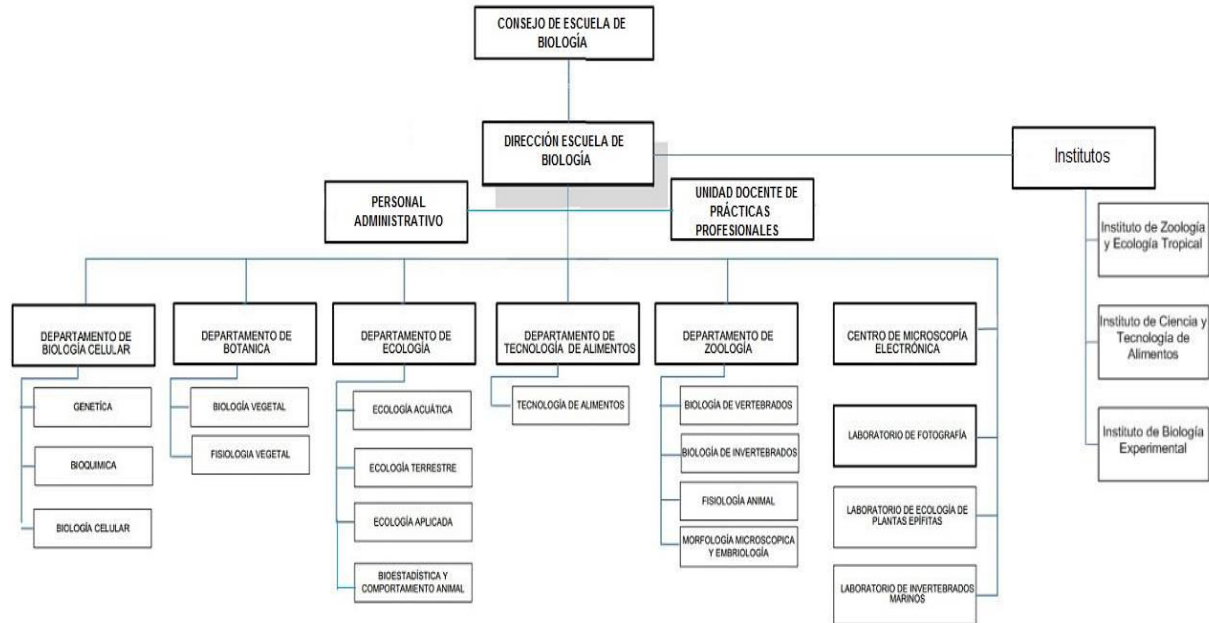
El Departamento de Tecnología de Alimentos es pionero de los estudios de Ciencia y Tecnología de Alimentos en el país, y funciona con sede propia desde marzo de 1961. Está constituido únicamente por la Unidad Docente de Tecnología de Alimentos

El Departamento de Zoología diestra recursos humanos capacitados para desarrollar actividades de docencia en el cuarto nivel, así como investigación en diferentes áreas de la Biomedicina y la Zoología. Cuenta con cuatro (4) Unidades Docentes: Biología de Invertebrados, Biología de Vertebrados, Fisiología Animal, Morfología Microscópica y Embriológica.

A continuación en la Figura 1.1 se muestra el organigrama de la Escuela de Biología, el cual muestra la manera cómo está constituida la escuela. Se puede apreciar como máxima autoridad al Consejo de Escuela, el cual está constituido por el director quien lo preside y ejecuta las decisiones emanadas en el mismo, cinco (5) jefes de departamentos, cinco (5) representantes de los profesores y dos (2) representantes estudiantiles. Seguidamente está la Dirección de la Escuela, la cual realiza la mayoría de las actividades administrativas y de control de actividades de la licenciatura y está integrada por La dirección integrada por la Unidad Docente de Pasantías y Prácticas Profesionales (UDPP) y el personal administrativo.

Posee otras dependencias adscritas cuyo desempeño principal es en el área de investigación y servicio, ellas son el Centro de Microscopía Electrónica, el Laboratorio de Fotografía, el Laboratorio de Ecología y Plantas Epífitas, el Laboratorio de Invertebrados.

Además Cuenta con los institutos de Zoología y Ecología Tropical, Ciencia y Tecnología de Alimentos y Biología Experimental



**Figura 1.1: Organigrama Estructural de la Escuela de Biología.**  
**Fuente: tomado de la página web de la Escuela de Biología.**

## 1.2 Proceso de Formalización

El estudiante de biología debe inscribir, cursar y aprobar Seminario I en el octavo (8°) semestre de la carrera, Seminario II en el noveno (9°) y Trabajo Especial de Grado (TEG) en el décimo (10°) para optar por el título de licenciado en Biología. Sin embargo, existe la posibilidad de que el estudiante pueda inscribir de manera simultánea el Seminario II y TEG en el noveno (9°) semestre de la carrera, presentando y cumpliendo con ambos en el mismo semestre; siempre y cuando haya cumplido con todas las materias obligatorias y electivas que le exige el pensum vigente de la escuela.

La escuela de biología define las reglas que deben cumplirse desde el momento que se realiza la inscripción hasta la presentación de cada uno de éstos.

El proceso comienza con la inscripción de Seminario I a través de la aplicación de Gestión Académica CONEST, la información de los inscritos no puede ser accedida ni

manipulada por la dirección de la escuela de biología. Días después de haberse efectuado la inscripción, CONEST envía una lista que contiene a las personas que inscribieron Seminario I a cada uno de los departamentos de la escuela.

Como la información es accedida por cada uno de los departamentos de la escuela pero no por la dirección en sí, los estudiantes no han completado su proceso de inscripción, éstos deben formalizar su inscripción en la fecha pautada una vez comenzado el semestre.

El proceso de formalización de inscripción es llevado a cabo de la siguiente manera:

1. En la semana tres (3) del semestre en curso cada uno de los estudiantes debe entregar en la Dirección de la escuela una planilla que lleva por nombre Formalización de Inscripción la cual contiene la siguiente información:
  - Datos personales del estudiante, tutor, jurados principales, los cuales son dos (2) y suplentes que son tres (3).
  - Información del proyecto (título y objetivo general).
  - Firma de tutor (Con la que el mismo acepta la responsabilidad de ser tutor del estudiante durante seminarios y TEG).
2. Una vez recibidas todas las planillas, la dirección de la escuela se encarga de clasificarlas por departamentos y enviarlas a cada uno de ellos.
3. El representante de cada departamento, al recibir las planillas de los inscritos correspondientes debe verificar manualmente la coincidencia de los estudiantes que formalizaron inscripción con los que se inscribieron mediante CONEST. Pueden ocurrir varias incidencias:
  - El estudiante haya realizado la inscripción en CONEST y no haya cumplido con el paso de formalización de inscripción: se debe notificar a la dirección de la escuela para que la misma le participe al estudiante.
  - Que el estudiante no se haya inscrito en CONEST y haya entregado la planilla de formalización: el estudiante debe realizar su inscripción de manera extemporánea.

- Que el estudiante se haya inscrito y a su vez haya entregado la planilla de formalización: se notifica al estudiante que ha cumplido satisfactoriamente con la formalización de inscripción.
4. Una vez realizada la verificación, el representante de cada departamento debe convocar a una reunión a los estudiantes que hayan llevado a cabo todo el proceso de inscripción, para explicar cómo se llevará a cabo el desarrollo del Seminario I hasta que el mismo sea entregado.
  5. Durante la semana diez (10) del semestre en curso, el estudiante debe hacer entrega del Seminario I en la dirección de la escuela, tanto en formato digital (en un DVD) como en físico.
  6. La Dirección después de recibir el Seminario I de cada estudiante debe clasificarlos y entregarlos al representante de cada departamento. El estudiante se encarga de entregarlo a tutor y jurados.
  7. El representante de cada departamento debe revisar cada uno de los Seminarios I y luego enviarlos nuevamente a la dirección con un informe (por cada uno) con las respectivas correcciones.
  8. Cuando el documento llega a dirección, el Consejo de Escuela se encarga de aprobar o no el mismo.
    - Si las correcciones manifestadas en el informe no son importantes, el Seminario I es aprobado, con datos del estudiante, tutor, jurados y el título del proyecto.
    - Cuando el tutor no pertenece a la escuela de biología, el Consejo de Escuela debe asignar un asesor académico (propuesto por el departamento), el cual es un profesor que si pertenece a la misma.
    - Si las correcciones son graves, la dirección notifica al estudiante para que las realice y se repite el ciclo de revisión para verificar si se aprueba o no el proyecto.
  9. Una vez aprobado el Seminario I, la dirección genera cartas informando la aprobación del proyecto y notificando el rol asignado al personal correspondiente.

10. Cada persona a la cual fue dirigida la carta, debe aceptar el rol asignado, notificándolo al tutor correspondiente y a la dirección de la escuela.
11. El tutor convoca de manera telefónica a los jurados para que asistan a la presentación del Seminario I.
12. Semanas antes de la culminación del semestre, el tutor o asesor académico según sea el caso debe generar la planilla de evaluación de Seminario I a través de CONEST e ir a retirarla físicamente en la división de control de estudios.
13. En la fecha pautada antes de culminar el semestre, el estudiante realiza la presentación del Seminario I. El tutor y jurados principales aprueban o reprueban y finalmente el tutor es el encargado de cargar las notas en CONEST.

Una vez finalizado el octavo semestre del estudiante (8°), con la aprobación del Seminario I, es momento de inscribir Seminario II. Puede pasar lo siguiente:

Cuando el estudiante decide inscribir únicamente el Seminario II y dejar el TEG para el décimo semestre (10°) de su carrera, el proceso se lleva a cabo de la siguiente manera:

1. Estudiante inscribe Seminario II a través del portal web CONEST.
2. Estudiante contacta a tutor para correcciones y revisiones del Seminario II.
3. Durante el semestre ocurren las revisiones y correcciones necesarias del Seminario II. Semanas antes de la fecha de presentación del mismo, Tutor se encarga de convocar telefónicamente a Jurados (Principales y Suplentes).
4. Estudiante se encarga de hacer llegar físicamente el documento a los Jurados.
5. Días antes de la fecha pautada para la presentación del Seminario II, el Tutor o Asesor Académico según sea el caso debe generar la planilla de evaluación de Seminario II a través de CONEST e ir a retirarla físicamente en Control de Estudios.
6. En la fecha pautada antes de culminar el semestre, el estudiante realiza la presentación del Seminario II, Tutor y Jurados Principales aprueban o reprueban y finalmente el Tutor se encarga de cargar las notas en CONEST.

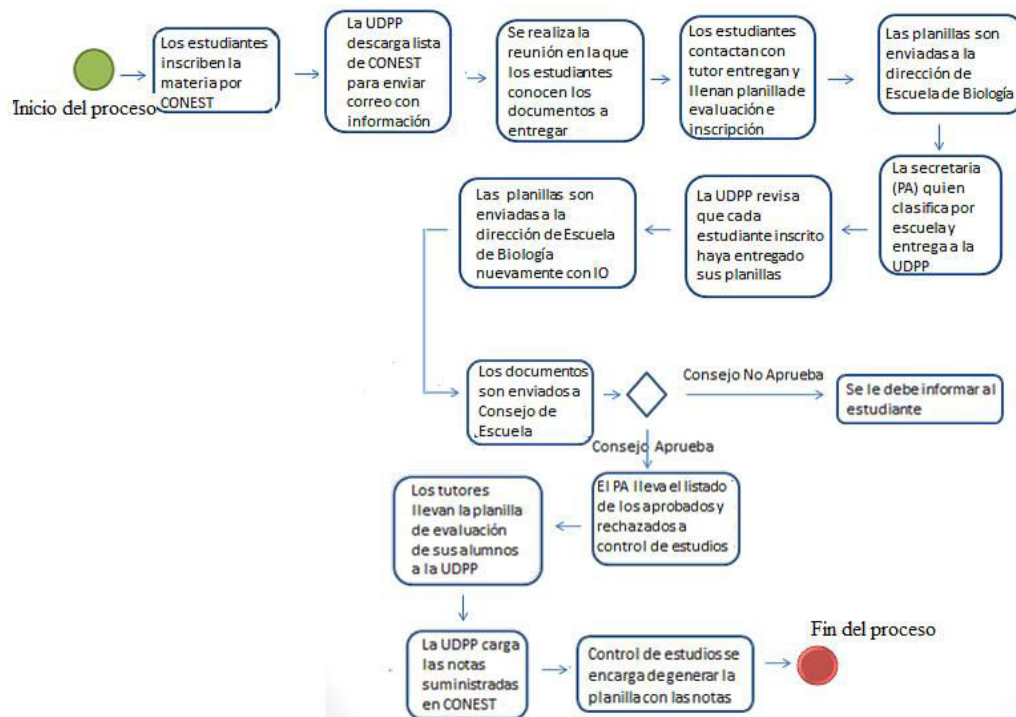
Si el estudiante decide inscribir Seminario II y TEG en el noveno (9°) semestre de su carrera. El proceso es el siguiente:

1. Estudiante inscribe Seminario II y TEG a través del portal web CONEST.
2. Estudiante contacta a tutor para correcciones y revisiones del TEG (telefónicamente, personalmente, a través del correo o de la forma que prefiera)
3. Durante el semestre ocurren las revisiones y correcciones necesarias del TEG. Días antes de la fecha de presentación del mismo, Tutor se encarga de convocar telefónicamente a Jurados (Principales y Suplentes).
4. Estudiante se encarga de hacer llegar físicamente el Seminario II a Jurados
5. Días antes de la fecha pautada para la presentación del TEG, el Tutor e el Asesor Académico dependiendo del caso, debe generar las planillas de evaluación de Seminario II y TEG a través de CONEST e ir a retirarla físicamente en Control de Estudios. (Se deben generar ambas planillas ya que a pesar de que se hará una presentación se debe cumplir formalmente con el reglamento)
6. En la fecha pautada antes de culminar el semestre, el Estudiante realiza la presentación del TEG, Tutor y Jurados Principales aprueban o reprueban y finalmente el Tutor carga las notas en CONEST, tanto del Seminario II como del TEG.

La Dirección de la Escuela de Biología no maneja información de los Estudiantes que aprobaron o reprobaron el Seminario I, Seminario II o TEG.

Es importante destacar que el Seminario I y II llevan por nota reprobado o reprobado, mientras que el TEG lleva por nota una calificación entre uno (01) y veinte (20) puntos. Donde se aprueba obteniendo una calificación entre diez (10) y veinte (20) puntos y reprueba con uno (01) o nueve (09).

En las Figura 1.2 se muestra el proceso manual de inscripción de las materias Seminario I, Seminario II y TEG.



**Figura 1.2: Proceso Manual de Inscripción de Seminarios y TEG. Elaboración Propia**

### 1.3 Planteamiento del Problema

Con base a la situación identificada al realizar el proceso de inscripción de Seminarios y TEG de la Escuela de Biología de la UCV descrita en el apartado anterior, se puede observar que todas las actividades son realizadas de manera manual, por ello, surgen un conjunto de problemas:

- Exceso de consumo de recursos (papel, tinta, entre otros) para el intercambio de información entre todo el personal que participa (alumnos, profesores, secretarías, tutores, director, entre otros).
- El extravío o pérdida de información, debido a la enorme cantidad de documentos que deben ser manipulados y son generados en papel.
- No es posible llevar un control administrativo y en los departamentos de la escuela sobre los alumnos que inscribieron Seminarios o TEG, tampoco, en la entrega que realizan los estudiantes de estos documentos al personal correspondiente debido a que la información no se encuentra centralizada.

- Suele ocurrir que el estudiante no entregue las cartas al personal correspondiente, por lo tanto se observa que es imposible llevar control de esta situación.
- El estudiante al entregar el Seminario I en la dirección y luego encargarse de hacerlo llegar a tutor y jurados, podría hacer cambios en el documento y no entregar el mismo que entrego inicialmente en la dirección, con esto se percibe que en la actualidad no es posible llevar un control de lo que el estudiante entrega en dirección, tutor y jurados.
- No es posible llevar un control de la aceptación de roles por parte de tutor y jurados en la dirección de la escuela de biología, pues muchos de ellos no notifican la aceptación del rol.

Como se pudo observar, la Escuela de Biología lleva a cabo el proceso para la formalización de inscripción de Seminarios y TEG de forma manual en su totalidad, con actividades que pudieran ser eliminadas mediante la automatización del mismo y así convertirlo en un proceso que brinde mayores beneficios a la comunidad universitaria.

Por todo lo antes mencionado, se plantea como pregunta de investigación: ¿de qué forma se puede mejorar el proceso de gestión de inscripción de Seminarios (I y II) y TEG de la escuela de biología de la facultad de ciencias de la UCV?

#### **1.4 Justificación**

Por la problemática antes expuesta, surge la iniciativa de crear e incorporar en la escuela de biología una aplicación web que permita gestionar los procesos llevados a cabo desde la etapa inicial con la inscripción de Seminarios o TEG hasta la final con la presentación de los mismos, mejorando así la ejecución de las tareas administrativas y departamentales; y además logrando que los estudiantes puedan realizar la inscripción del Seminario I, II o TEG, de una manera sencilla y eficiente, haciendo uso de la tecnología.

El desarrollo de este Trabajo Especial de Grado permitirá la obtención de los siguientes beneficios:

- Se contribuye al ahorro de tiempo, dedicación y esfuerzo por parte de la administración, los profesores e incluso el estudiante en la ejecución de las actividades que corresponden



a cada uno de ellos durante el proceso de inscripción, por lo que se agiliza la realización de las mismas.

- Se minimizan los posibles errores que puedan existir durante la realización de las actividades elaboradas manualmente.
- Se reduce significativamente la pérdida de información y el gasto de materiales como papel, tinta, entre otros.
- Se permite llevar un control de los alumnos que inscribieron Seminarios o TEG y de la entrega de los mismos.
- El estudiante no tendrá que asistir personalmente a formalizar su inscripción, podrá hacerlo desde un computador con conexión a Internet.
- La información estará disponible en todo momento para las entidades involucradas que requieran de ella, bien sea: dirección, administración, departamentos o estudiantes, lo que permite que cada una disponga de la información que necesite en cualquier momento, agilizando de esta manera el desarrollo de las actividades.
- Centralización de la información y distribución a cada una de las entidades involucradas, por lo tanto se permitirá llevar un control centralizado del desarrollo de cada una de las actividades que deben llevarse a cabo en el proceso de inscripción (formalización de inscripción, entrega de seminarios, aceptación de roles, entre otras).

Cada uno de los beneficios ya mencionados, permitirán que el proceso de inscripción de Seminarios y TEG en la escuela de biología se lleve a cabo de una forma mucho más simple, rápida y eficiente para todas aquellas personas involucradas en el mismo. Asimismo, mediante este trabajo se presenta un conjunto de soluciones a estas dificultades, con el objeto de aumentar la satisfacción en los estudiantes, docentes y personal administrativo, y minimizar el porcentaje de error que pudiese existir a la hora de llevar a cabo estas tareas y actividades, apoyándonos en las bondades que ofrecen las aplicaciones Web.

## **1.5 Objetivo General**

Desarrollar una aplicación web para la gestión del conjunto de procesos relacionados a las diversas solicitudes estudiantiles y actividades que se realizan durante el proceso de formalización de inscripción de Seminarios y TEG.

## **1.6 Objetivos Específicos**

- Determinar las funcionalidades que tendrá la aplicación.
- Diseñar la interfaz gráfica del usuario para la aplicación.
- Definir el modelo lógico y físico de la Base de Datos de la aplicación.
- Implementar la aplicación en un servidor de la Facultad de Ciencias de la UCV.
- Aplicar pruebas de aceptación, funcionalidad y usabilidad para verificar el correcto funcionamiento de la aplicación.

## **1.7 Alcance**

La aplicación web propuesta permitirá gestionar el proceso de formalización de la inscripción de Seminarios y TEG. Estará dirigida a los estudiantes, profesores (director, tutor, representante de la UDPP y jefe de departamento) y a la secretaria de la mencionada Escuela.

Además, se espera asegurar un entorno de trabajo que contribuya al constante mejoramiento de las actividades administrativas de la Escuela de Biología.

Para cada usuario se desarrolló un módulo a través del cual el mismo (iniciando sesión) podrá realizar cada una de las actividades que le competen dentro del proceso.

## **1.8 Metodología de Desarrollo**

Para el desarrollo del presente TEG se utilizó una adaptación de la metodología de desarrollo AgilUs, la cual hereda las características de una metodología ágil y se centra en la usabilidad e interacción con los usuarios. Esta metodología también se caracteriza por la poca documentación requerida y por la búsqueda de acortar el tiempo de desarrollo. Esta adaptación contempla todos los aspectos necesarios que aseguren la productividad de la aplicación y la

comunicación entre los entes involucrados, es decir, se adapta eficaz y eficientemente al desarrollo de este proyecto.

De acuerdo a Acosta (2011), AgilUs formalmente se define como un método de desarrollo ágil que toma en cuenta la usabilidad a tempranas etapas del desarrollo de software. Su enfoque principal es la usabilidad, por lo tanto, la construcción y desarrollo de las interfaces de usuario debe ser tomada en cuenta desde el inicio para así alcanzar el objetivo primordial: dar usabilidad al usuario final.

Los cambios y adaptaciones de la aplicación se determinan en función de los requerimientos, las funcionalidades de la aplicación se van añadiendo progresivamente, por esta razón que el usuario no se tiene que adaptar al sistema, sino la aplicación se va evolucionando conforme a las necesidades del usuario.

Con el método de desarrollo AgilUs se persigue abarcar aspectos de usabilidad incorporando una visión que complementa el área de Interacción Humano Computador con la Ingeniería de Software, dando lugar a un conjunto de buenas prácticas y evaluaciones aplicadas al software.

El ciclo de vida de la metodología se basa en un desarrollo iterativo e incremental de prototipos hasta alcanzar el producto final que será entregado al cliente. Se divide en 4 etapas que son: requisitos, análisis, prototipaje y entrega. Además involucra evaluaciones de usabilidad en cada etapa del proceso.

En la figura 3.1 se muestra un diagrama de la interacción entre cada una de las etapas del ciclo de vida de AgilUs, con las actividades que se realizan y artefactos que se generan en cada etapa. A continuación se describen brevemente las etapas de este método:

- **Requisitos:** En esta etapa se realiza el análisis general del problema a abordar y se lleva a cabo el levantamiento de requerimientos. Se realizan actividades como: tormenta de ideas, entrevistas, encuestas y cuestionarios. Se estudian productos existentes que sean similares, se crea un perfil de usuario y se listan los requerimientos funcionales y no funcionales a desarrollar.
- **Análisis:** Una vez estipulados los requisitos que persigue la aplicación, se lleva a cabo un análisis de los mismos para su posterior desarrollo. En esta etapa se toman como

apoyo los diferentes diagramas realizados bajo la notación estándar del Lenguaje de Modelado Unificado (UML, por sus siglas en inglés Unified Modeling Language), especialmente: casos de uso y modelo de objetos del dominio para definir las funcionalidades de la aplicación.

- Prototipaje: En esta etapa se desarrolla un prototipo rápido de la interfaz de usuario a partir de los patrones de interacción, al cual se le van incorporando funcionalidades hasta llegar al producto final. El usuario lleva a cabo la evaluación de los prototipos a fin de conocer si los mismos cumplen con los principios de usabilidad.
- Entrega: en esta etapa se tiene el software final al que se le realizan pruebas que certifiquen que es un software usable y sin errores, para ello se utilizan las pruebas de aceptación. Una vez aplicadas las pruebas y sabiendo que el software es usable y no tiene errores, es puesto en producción el mismo.



**Figura 1.3: AgilUs: Etapas, Actividades y Artefactos. Acosta (2011).**

AgilUs hace uso de artefactos entregables que dan soporte en la creación de interfaces y usabilidad del desarrollo, los artefactos que contempla la metodología son:

- Tormenta de ideas
- Entrevista
- Sondeos
- Perfil de usuario
- Lista de requerimientos funcionales y no funcionales

- Modelado de Casos de uso
- Modelo de Objetos del Dominio
- Guía de Estilos

Desde un principio se consideró este método ya que la usabilidad es un aspecto importante en el éxito de un software interactivo dirigido a una audiencia amplia como es el caso de este Trabajo Especial de Grado. Adicionalmente, esta metodología está orientada a aplicaciones con un alto grado de interacción con usuarios, ya que es ahí donde el diseño de una interfaz usable se ha vuelto un aspecto fundamental. AgilUs plantea el desarrollo de un sistema, partiendo de un prototipo de interfaz de usuario y de la evaluación de usabilidad desde el inicio de la aplicación.

## **1.9 Antecedentes**

La Escuela de Biología de la UCV desde el 2014 cuenta con una aplicación web denominada Biotec, la cual fue desarrollada y presentada en el Trabajo Especial de Grado titulado: “Aplicación Web para la gestión del proceso de inscripción de Métodos de Investigación y de Laboratorio de la Escuela de Biología de la Universidad Central de Venezuela”. Dicha aplicación se encarga de la gestión del proceso de inscripción de las materias Métodos de Investigación y Métodos de Laboratorio de la Escuela de Biología.

Biotec establece un proceso similar a la aplicación que se desea desarrollar, ya que el proceso para solicitar la inscripción de las asignaturas Métodos de Investigación y Laboratorio, cumple un flujo de trabajo similar al realizado durante la inscripción de Seminarios y TEG.

## Capítulo 2

### Marco Conceptual

El siguiente capítulo tiene como finalidad presentar bases teóricas relacionadas a un ambiente de desarrollo web, que sirven de fundamento para el desarrollo del TEG. El mismo se divide en dos grandes tópicos, los cuales se describen a continuación:

En la primera sección, se hace referencia a lo que es una aplicación web de forma general, así como también su estructura y sus principales características y ventajas. Se explica además la arquitectura Cliente-Servidor, sus características y componentes para luego enumerar sus principales ventajas. Finalmente se dará una noción del patrón de diseño MVC (Modelo Vista Controlador), que se encarga de separar las funciones del cliente, el servidor Web y el servidor de Base de Datos.

En la segunda sección, se muestran los recursos tecnológicos necesarios para el desarrollo de la aplicación web, que a su vez constituyen la columna tecnológica de la aplicación a desarrollar. Las tecnologías del lado del servidor son Apache y PHP, las del servidor de bases de datos MySQL, y las del lado del cliente, HTML, CSS y JavaScript, JQuery, AngularJS, entre otras.

#### 2.1 Aplicaciones Web

De acuerdo a Pereda (2007) y Eñubes (2013), una aplicación web se puede definir como un conjunto de herramientas de software o un sistema informático, que permite al usuario procesar una tarea en particular accediendo a un servidor web a través de Internet mediante un navegador web (Internet Explorer, Mozilla Firefox, Google Chrome, entre otros), o de una Intranet (red entre computadores que sirve para compartir recursos y tiene uso exclusivo dentro del sitio donde fue creada).

En las Figuras 2.1 y 2.2 se pueden apreciar dos ejemplos de aplicaciones web de la Facultad de Ciencias de la UCV, como son la aplicación de Gestión Académica de la División de Control de Estudios (CONEST) y la aplicación para la Inscripción de Proyectos de Pasantías (SIGEPAS).



**Figura 2.1: Interfaz del Menú Principal de Estudiantes en CONEST.**  
**Fuente: tomado de la página web de CONEST.**



**Figura 2.2: Sistema para la inscripción de proyectos de pasantías.**  
**Fuente: tomado de la página web de CONEST.**

## 2.2 Características de una aplicación

De acuerdo a Good (2015), algunas de las principales características de las Aplicaciones Web son:

- **Compatibilidad multiplataforma:** las Aplicaciones Web tienen un camino mucho más sencillo para la portabilidad multiplataforma que las aplicaciones de escritorio. Varias tecnologías incluyendo Java, Flash, ASP y Ajax permiten un desarrollo efectivo de programas soportando todos los sistemas operativos principales (Windows, Linux, MacOS).
- **Actualización:** las aplicaciones basadas en web están siempre actualizadas con la última versión sin requerir que el usuario tome acciones pro-activas, y sin necesitar llamar la atención del usuario o interferir con sus hábitos de trabajo con la esperanza de que va a iniciar nuevas descargas y procedimientos de instalación.
- **Acceso Inmediato:** la mayoría de las aplicaciones web no necesitan ser descargadas, instaladas y configuradas. Para su acceso, simplemente se requiere hacer uso de un navegador web y conexión a Internet como se hizo mención anteriormente.
- **Facilidad de prueba:** no existen obstáculos para permitir pruebas sencillas y efectivas de herramientas y aplicaciones antes de realizar una operación que afecta la realidad como por ejemplo cargar la tarjeta de crédito, es decir poder simular este tipo de operaciones.
- **Menor requerimiento de memoria:** las aplicaciones web tienen muchas más razonables demandas de memoria RAM de parte del usuario final que los programas instalados localmente. Al residir y correr en los servidores del proveedor, esas aplicaciones basadas en web usan en muchos casos la memoria del servidor, dejando más espacio para correr múltiples aplicaciones al mismo tiempo sin afectar el rendimiento.
- **Datos online:** con el desplazamiento de las aplicaciones locales a aquellas basadas en web también los datos que se crean y acceden necesitan experimentar cambios porque van a estar en constante movimiento dentro de la red al alcance de todo el mundo.
- **Múltiples usuarios concurrentes:** las aplicaciones web pueden ser accedidas por miles de usuarios al mismo tiempo. No hay necesidad de compartir recursos cuando múltiples usuarios pueden ver e incluso editar el mismo documento de manera conjunta. En el caso de que llegase a actualizarse la aplicación, todos los usuarios estarán al tanto de esta actualización de manera inmediata.
- **Datos seguros:** A medida que las compañías se hagan cargo del almacenamiento de los datos del usuario, el almacenamiento de datos redundantes, altamente fiables, serán la



norma más que la excepción, y los usuarios van a tener mucho menos riesgo de perder sus datos debido a una falla del equipo.

- **Fácil Desarrollo:** Debido a que las aplicaciones han sido separadas de computadoras locales y sistemas operativos específicos, estas pueden también ser escritas en prácticamente cualquier lenguaje de programación. Es escrita en lenguajes soportados por navegadores web, pues son ellos quienes la ejecutan independientemente del sistema operativo en el que se encuentre.

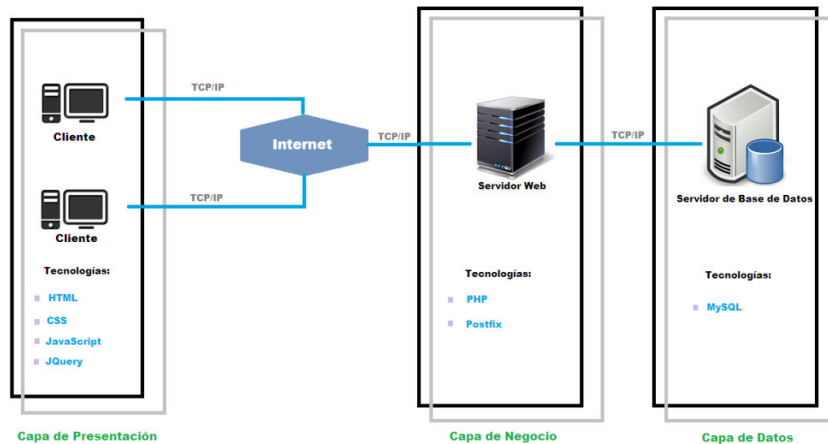
### **2.3 Arquitectura Cliente-Servidor**

La Universidad de Sevilla (2009) plantea que la arquitectura Cliente/Servidor es un modelo para el desarrollo de sistemas de información, en el que las transacciones se dividen en elementos independientes que cooperan entre sí para intercambiar información, servicios o recursos.

Dicha arquitectura es la más utilizada en el contexto de aplicaciones web, debido a que la computadora de cada uno de los usuarios, llamada cliente, inicia un proceso de diálogo: produce una demanda de información o solicita recursos. La computadora que responde a la demanda del cliente, se conoce como servidor. Bajo este modelo cada usuario tiene la libertad de obtener la información que requiera en un momento dado proveniente de una o varias fuentes locales o distantes y de procesarla como según le convenga. (Dptoccai, 2001)

En esta arquitectura se puede encontrar el modelo tradicional de dos (2) capas que involucra: el Cliente y el Servidor y el de tres (3): El Cliente, El Servidor de Aplicaciones y El Servidor de Datos.

Una de las variantes de la arquitectura Cliente/Servidor más utilizadas en la actualidad es la de tres Capas, la cual se muestra en la Figura N° 1.3, donde se muestran cada tipo de tecnología involucrada en la generación e interacción de documentos Web. (Vegas, 2002).



**Figura 2.3: Esquema general de la arquitectura Cliente-Servidor. Elaboración Propia**

La arquitectura Cliente/ Servidor de tres capas, se constituye de un código de presentación, de procesamiento de datos y de almacenamiento de datos (Antelo, 2004). Al momento de hacer solicitud de una aplicación web bajo dicha arquitectura ocurre básicamente lo siguiente:

1. Se envía la dirección del Localizador de Recursos Uniforme (Uniform Resource Locater, por sus siglas en inglés URL) que es una secuencia de caracteres de acuerdo a un formato estándar que se usa para nombrar recursos en Internet para su localización e identificación, a través del cliente web (Navegador web).
2. Se inicia la conexión con el servidor, es importante destacar que siempre se inicia una por cada petición que realiza el cliente.
3. Se solicita la página o recurso.
4. El recurso o página es enviada por parte del servidor, si no se encuentra en el servidor se envía un código de error.
5. Se interpreta el documento HTML y se hace petición de otros objetos a los que se haga referencia
6. Se cierra la conexión una vez que hayan llegado todos los recursos.

### 2.3.1 Componentes de la Arquitectura Cliente-Servidor

Según la Universidad de Sevilla (2009), Universidad de Alicante (s.f.) y Newcomlab (2013):

El modelo de tres (3) capas está conformado por: Capa de Presentación, Capa de Negocio y Capa de Datos, las cuales se explican a continuación.

- Capa de Presentación: Esta capa está conformada por los Clientes (navegadores) y es donde se genera la presentación visual o interfaz de usuario, aquí se realizan peticiones y se reciben respuestas de la Capa de Negocio. El Cliente solicita datos al servidor Web, a través de Internet, y una vez recibida la respuesta de éste, realiza la interpretación y ejecución de las aplicaciones web escritas fundamentalmente en Lenguaje de Marcado de Hipertexto (HTML). Las tecnologías manejadas en esta capa son: HTML, CSS, JavaScript, JQuery entre otras.
- Capa de Negocio: Aquí se mantiene la sincronización entre la Capa de Presentación y la Capa de Datos. En ésta se encuentran alojados módulos que ejecutan e implementan la lógica de la aplicación. Es aquí donde se realiza todo el procesamiento de los datos de forma transparente para el usuario desde que éste realiza una solicitud hasta que recibe una respuesta. Posee comunicación con la capa de presentación para recibir solicitudes del usuario y enviar respuestas; y con la de datos para solicitar al o los Sistemas Manejadores de Base de Datos (SMBD) el almacenamiento, modificación eliminación o recuperación de datos requeridos antes de responder una solicitud al usuario.
- Capa de Datos: En esta capa se encuentran alojados los datos y se encarga de la gestión de los mismos (modificación, almacenamiento, eliminación o recuperación) a través de un Sistema Manejador de Base de Datos (SMBD), como por ejemplo MySQL. Se comunica con la de negocio para atender solicitudes de la misma sobre datos que se requieran recuperar, modificar, eliminar o almacenar y enviar una respuesta según sea el caso.

### **2.3.2 Ventajas de la Arquitectura Cliente-Servidor**

De acuerdo a Dimagin (2007), las principales ventajas de las aplicaciones web bajo la arquitectura Cliente-Servidor se pueden mencionar las siguientes:

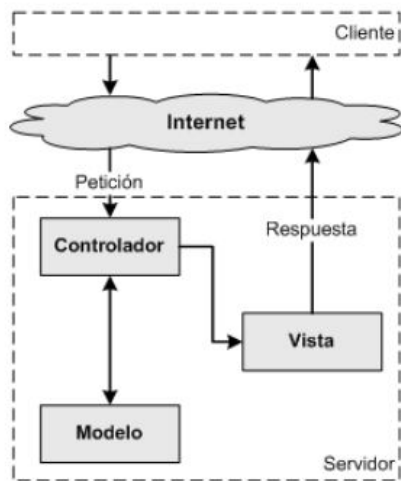
1. Facilita la integración entre sistemas heterogéneos y comparte información permitiendo que las máquinas existentes puedan ser utilizadas con interfaces usables e intuitivas al usuario, y así integrar los computadores con sistemas medianos y grandes, sin necesidad de que todos tengan que utilizar el mismo sistema operacional.
2. La existencia de plataformas de hardware y software de varios fabricantes cada vez más económicas, contribuye a la reducción de costos y favorece la flexibilidad en la implantación de soluciones.
3. Es de fácil mantenimiento ya que se puede modificar un cliente sin afectar a los otros, o inclusive, reparar, actualizar o mover un servidor sin afectar a los clientes.
4. Facilita a las diferentes áreas dentro de una organización soluciones locales, permitiendo la integración y centralización de la información.
5. Permite mantener y desarrollar aplicaciones de manera rápida, pues se pueden emplear herramientas existentes, por ejemplo, SMBD o herramientas de más bajo nivel como las llamadas al sistema.
6. Se puede aumentar el número de clientes y/o servidor(es) por separado, dependiendo de las necesidades.

### **2.4 Patrón de Diseño Modelo Vista Controlador**

Un patrón de diseño en ingeniería del software es un modelo formal que es aplicable a diferentes dominios. Estos problemas aunque de diferentes ámbitos son semejantes desde el punto de vista de la estructura lógica de la solución.

El Modelo Vista Controlador (MCV) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres (3) componentes distintos (Ver Figura 2.4). El patrón MVC es frecuentemente utilizado en aplicaciones web, donde la vista es la interfaz de usuario que muestra información sobre el modelo y que representa el dispositivo de entrada que se usa para modificarlo, el controlador

es la aplicación de Gestión de Base de Datos y el modelo está compuesto por el estado y los datos que la aplicación representa. (Modelo Vista Controlador, 2013).



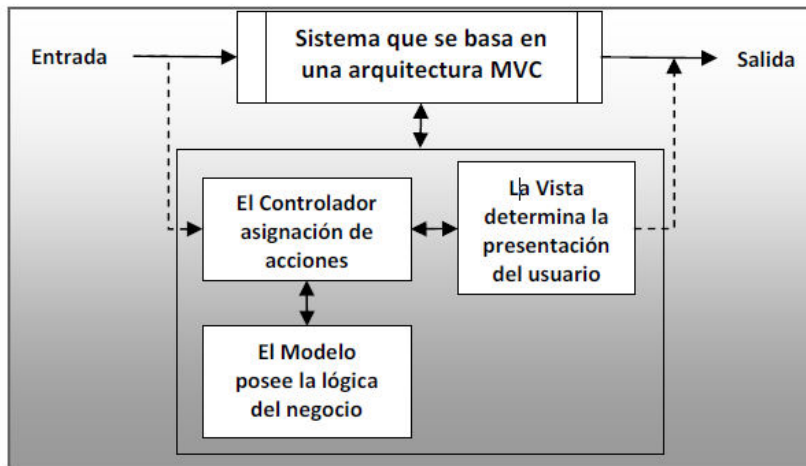
**Figura 2.4: Patrón de Diseño Modelo Vista Controlador MVC. (Modelo Vista Controlador, 2013).**

### 2.4.1 Flujo de Control en el Modelo Vista Controlador

A continuación se presenta una secuencia de tareas del MCV las cuales se combinan para ofrecer una respuesta satisfactoria y coherente con respecto a la acción realizada por el usuario de la aplicación:

1. El usuario interactúa con la interfaz introduce un evento cualquiera (por ejemplo, el usuario pulsa un botón o hace clic en un enlace).
2. El controlador recibe (por parte de los objetos de la interfaz-vista) la notificación del evento introducido por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos (handler) o callback.
3. El controlador se comunica con el modelo (decide quién lleva a cabo la petición en la capa del modelo), actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario (por ejemplo, el controlador actualiza el carro de la compra del usuario). Los controladores complejos están a menudo estructurados usando un patrón de comando que encapsula las acciones y simplifica su extensión.
4. Una vez que el modelo termina las operaciones pertinentes, devuelve el control de ejecución al controlador.

5. Después de recibir la respuesta del modelo el controlador la entrega a la vista para que sea mostrada al usuario. En algunos casos el modelo se comunica con la vista para entregarle los resultados de la solicitud.
6. La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.



**Figura 2.5:** El esquema de un sistema basado en el patrón software MVC (Gutiérrez, 2005).

## 2.5 Sistema Manejador de Base de Datos

Debido al auge tecnológico en los últimos años, existen muchas organizaciones y sitios web que necesitan mantener y manejar de forma eficiente un gran volumen de datos e información. Muchos de ellos optan por soluciones comerciales, aunque muchas otras confían en el software libre optando por una solución como MySQL. A continuación se explicarán brevemente la aplicación Manejador de Base de Datos relacional MySQL. El SMBD para este trabajo fue MySQL debido a su licencia de tipo código abierto, su velocidad en sistemas de alta concurrencia, su alta portabilidad y su facilidad de uso. A continuación se describe el SMBD MySQL.

## 2.5.1 MySQL

De acuerdo a MySQL(2013), se define como un sistema de administración de bases de datos relacional (SMBDR), multihilo y multiusuario, que almacena y distribuye una gran cantidad de datos, típicos de una aplicación.

Actualmente MySQL es la aplicación de manejador de bases de datos más popular con más de 14 millones de instalaciones alrededor del mundo, con aproximadamente 75 mil descargas del software cada día (Schumacher y Lentz, 2011), esto debido a su motor de código abierto, diseñado para tratar con gran cantidad de datos y trabajar en entornos Cliente-Servidor.

## 2.5.2 Características de MySQL

Algunas características del SMBDR MySQL son las siguientes:

- Velocidad de acceso: está basado en una arquitectura multihilos mediante hilos del kernel, por lo que diversos clientes pueden acceder simultáneamente de forma concurrente (multiusuario).
- Velocidad de respuestas: MySQL implementa un almacenamiento parcial de los resultados de las consultas recurrentes, por lo que una vez que se ejecuta una consulta, su resultado es almacenado de forma temporal, por si se requieren esos datos nuevamente, aumentando la velocidad de respuesta.
- Fácil uso: provee una interfaz de línea de comandos. Además cuenta con muchas herramientas gráficas para su manejo.
- Puede ejecutarse en diversos sistemas operativos sin ningún problema.
- Internacionalización: brinda soporte para varios idiomas y diversos juegos de caracteres.
- Sentencias y Funciones: maneja un amplio subconjunto del Lenguaje de Consultas Estructurado o SQL. Algunas extensiones son incluidas igualmente. Disponibilidad en gran cantidad de plataformas y sistemas.
- Código abierto: cualquier desarrollador tiene acceso al código libremente.
- Seguridad: posee un buen enfoque en cuanto a seguridad, ya que ofrece un sistema de contraseñas y privilegios seguro mediante verificación basada en el host y el tráfico de

contraseñas está cifrado al conectarse a un servidor. Soporta gran cantidad de datos. MySQL Server tiene BD de hasta cincuenta (50) millones de registros.

### **2.5.3 Limitaciones de MySQL**

Según (Schumacher y Lentz, 2011) entre las limitaciones de MySQL se pueden mencionar:

- Para algunas transacciones, según el tipo de tablas que se esté utilizando, al momento de consultar o modificar la información de la BD, toda la tabla donde se encuentra la información es bloqueada.
- No puede llevar a cabo copias de seguridad para todas las tablas que maneja. Las copias de seguridad “en caliente” son copias de la información que se realizan durante una operación sin bloquear tablas.
- Muchos SDBD permiten definir tipos de datos por el usuario. MySQL no lo permite.

## **2.6 Tecnologías Web**

En base a Morales (2010), las tecnologías web son aquellas que proporcionan al desarrollador un conjunto de herramientas que facilitan la elaboración de aplicaciones web. Se dividen en tecnologías del lado: del cliente (lenguajes procesados por el navegador web sin ningún tratamiento previo) y del servidor (lenguajes reconocidos, ejecutados e interpretados del lado del servidor y enviados al cliente en un formato comprensible para él).

Es importante mencionar que tanto los lenguajes del cliente como los del servidor son independientes, lo cual permite que la página pueda ser alojada en cualquier sitio. Y a su vez, un lenguaje de lado servidor es independiente del navegador utilizado, por lo que no necesitará algún plugin para visualizar correctamente cualquier página.

Las tecnologías Web se dividen en tecnologías del lado del cliente y del lado del servidor, aquellas del lado del cliente están embebidas en la página HTML y son interpretadas y ejecutadas por el navegador Web, las tecnologías que se mencionaran a continuación fueron las utilizadas para el desarrollo de este TEG, debido a al potencial que ofrecen y la facilidad que brindan en el desarrollo de aplicaciones Web.



### **2.6.1 Lenguaje de Marcado de Hipertexto (HTML)**

Según Álvarez (2010) y W3C (2013) el Lenguaje de Marcado de Hipertexto o HTML, por sus siglas en inglés Hyper Text Markup Language: es un lenguaje de marcado utilizado del lado del cliente que se usa para el desarrollo de páginas web. Consta de etiquetas que indican al navegador web la forma y estructura que tendrá el texto, las imágenes y los videos colocados en la página web.

Los diseñadores de páginas web lo utilizan para crear sus páginas, a través de programas que generan código HTML y los navegadores las muestran a los usuarios después de leer su contenido. Las páginas web creadas por éste contienen enlaces o hipervínculos que conducen a otros documentos o fuentes de información relacionados.

De acuerdo al identificador que tenga la etiqueta cumplirá una función en específico como por ejemplo colocar el texto en negritas, en cursiva, centrar el texto, entre otros. Su última versión es HTML5, la cual posee mejores características para CSS, DOM y JavaScript, reduce la necesidad de utilizar plugins externos como flash y es un lenguaje independiente del dispositivo utilizado. En este Trabajo Especial de Grado fue utilizado HTML5 ya que tiene nuevas etiquetas que permiten expandir las funcionalidades del lenguaje de hipertexto.

### **2.6.2 Hojas de Estilo en Cascada**

De acuerdo al aporte de Egiluz (2012) y W3C (2013), se define las Hojas de Estilo en Cascada (CSS), por sus siglas en inglés Cascading Style Sheet, como un lenguaje de hojas de estilo utilizado del lado del cliente que permite controlar la estructura y forma (presentación) que van a tener los documentos escritos en HTML o XHTML (páginas web) de una manera más fácil, limpia y sencilla. Por lo tanto, permite al desarrollador crear la presentación de sus páginas de una forma mucho más eficiente y sin ensuciar el código HTML.

Existen 3 opciones para aplicar CSS en un documento HTML (Eguiluz, 2007. a):

- Incluir CSS en el mismo documento HTML.
- Definir CSS en un archivo externo.
- Incluir CSS en los elementos HTML.

### **2.6.3 Twitter Bootstrap**

Bootstrap es un framework de código abierto desarrollado por twitter para la creación de sitios y aplicaciones web (Bootstrap, 2014). Contiene plantillas de diseño basadas en HTML, JavaScript y CSS con tipografías, formularios, botones, gráficos, barras de navegación y demás componentes de interfaz. Bootstrap tiene un soporte para HTML5 y CSS y es compatible con la mayoría de los navegadores web.

Bootstrap es modular y brinda la capacidad de construir rápidamente interfaces elegantes y ricas, su funcionalidad consiste esencialmente en una serie de hoja de estilo que implementan la variedad de componentes de la herramienta. En adición a los elementos regulares de HTML, Bootstrap contiene otra interfaz de elementos comúnmente usados, la cual incluye botones con características avanzadas (Por ejemplo grupo de botones o botones con opción de menú desplegable, listas de navegación, etiquetas horizontales y verticales, ruta de navegación, paginación, entre otros), etiquetas, capacidades avanzadas de miniaturas tipográficas, formatos para mensajes de alerta y barras de progreso, muchos de estos usados en el desarrollo de este sistema.

Para usar Bootstrap en una página HTML, el desarrollador solo debe descargar la hoja de estilo Bootstrap CSS y enlazarla en el archivo HTML.

### **2.6.4 JavaScript**

De acuerdo a Eguíluz (2008) se define JavaScript como un lenguaje de programación multiparadigma utilizado principalmente del lado del cliente que permite el desarrollo de páginas web dinámicas que incorporen efectos como: texto que aparece y desaparece, animaciones, acciones que se activan cuando se pulsan botones y ventanas con notificaciones al usuario. Tiene una sintaxis similar al lenguaje C y al lenguaje Java.

Técnicamente, JavaScript es un lenguaje de programación interpretado, debido a que no se necesita compilar el programa para ejecutarlo, por lo que se puede probar en cualquier navegador web sin necesidad de procesos intermedios (en la actualidad, todos los navegadores web interpretan código JavaScript integrado en las páginas web). Es orientado a objetos ya que permite definir clases con sus atributos y métodos.

## 2.6.5 jQuery

Antes de definir jQuery, es conveniente conocer el significado de Modelo de Objetos del Documento o DOM, por sus siglas en inglés Document Object Model, según W3C (2005): “Es una Interfaz de Programación de Aplicaciones o API, por sus siglas en inglés Application Programming Interface, de lenguaje neutro, que permite a los programas y scripts acceder y actualizar dinámicamente el contenido, estructura y estilo de los documentos web” (párr. 1).

jQuery (2013) es una librería rápida y concisa desarrollada en el lenguaje JavaScript, que simplifica la manera de interactuar con los documentos HTML y facilitar la programación de aplicaciones complejas, así como para simplificar el manejo de eventos, manipular el árbol DOM y desarrollar animaciones a páginas web, entre otros.

Esta librería realiza un gran aporte al colocar código no intrusivo JavaScript en las páginas. Su diseño permite especificar el comportamiento e interacción de los elementos del HTML, separado de las etiquetas valiéndose del DOM.

El uso de esta librería, mejora la usabilidad de la aplicación al reducir la carga cognitiva, prevenir errores, proveerle seguridad y confianza al usuario, además de reducir el tiempo de respuesta de la aplicación.

Además, abstrae al programador del problema recurrente entre la compatibilidad de scripts y los navegadores al realizar aplicaciones web en JavaScript, ya que ahorra el trabajo de realizar código que permita detectar el navegador utilizado por el usuario para realizar distintas acciones dependiendo del mismo.

La librería jQuery tiene la particularidad de funcionar de igual forma en cualquier navegador, esto gracias a la incorporación de un conjunto de clases que permiten al desarrollador abstraerse de tomar en cuenta el navegador en el cual se desplegará la aplicación, además el uso de esta librería es no intrusivo ya que está separado de la estructura HTML de la aplicación.

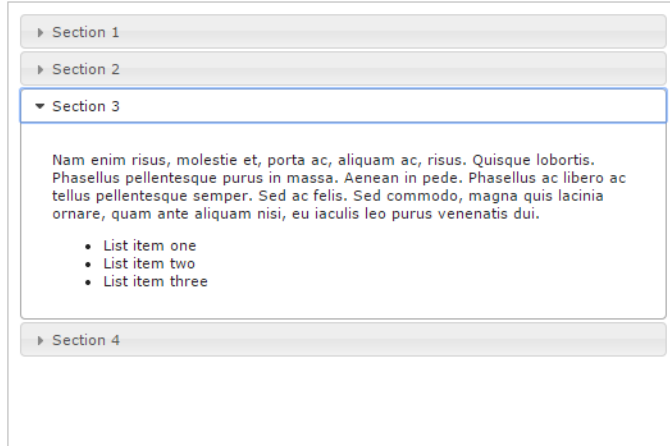
La versión más reciente liberada de ésta librería, ampliamente usada y documentada, es la 2.1.3.

## 2.6.6 jQuery User Interface (jQuery UI)

jQuery UI es una librería de componentes para jQuery que añade un conjunto de funcionalidades, complementos y widgets para la creación de Aplicaciones Web interactivas (jQuery UI, 2014). Esta librería proporciona abstracciones de bajo nivel de interacción y animación, efectos avanzados y de alto nivel, además de un conjunto completo de controles de interfaz de usuario conocido como widgets.

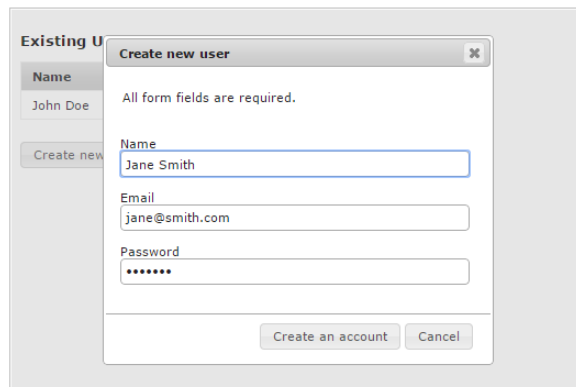
jQuery UI está dividida en cuatro (4) módulos:

- Núcleo: contiene funciones básicas para el resto de los módulos, por ejemplo, las interacciones y los widgets.
- Interacciones: consisten en efectos y comportamientos complejos, pero que brindan una mejor experiencia de usuario, tales como draggable (hacer un elemento arrastrable), resizable (permite redimensionar un elemento), sortable (permite ordenar una lista de elementos), entre otros.
- Efectos: permiten añadir transiciones animadas y facilidades para interacciones, tales como mostrar, ocultar, color, animación, entre otros.
- Widgets: son un conjunto completo de controles de interfaz. Cada control tiene un conjunto de opciones configurables y se les puede aplicar estilos CSS. Dentro de los widgets más utilizados se encuentran los siguientes:
  1. Menú con efecto acordeón (Accordeon): permite mostrar diversos contenidos clasificados en secciones. Como se observa en la Figura 2.6, cada sección posee un título que se utiliza para abrir el contenedor y mostrar su contenido. Al presionar el encabezado de una de las secciones, su contenido se visualiza justo debajo del mismo. Se puede visualizar el contenido de una sección determinada hasta que se cierra automáticamente luego de abrir otra sección. La ventaja de este componente de jQuery es que permite ahorrar espacio en la página web al permitir visualizar una sola sección a la vez.



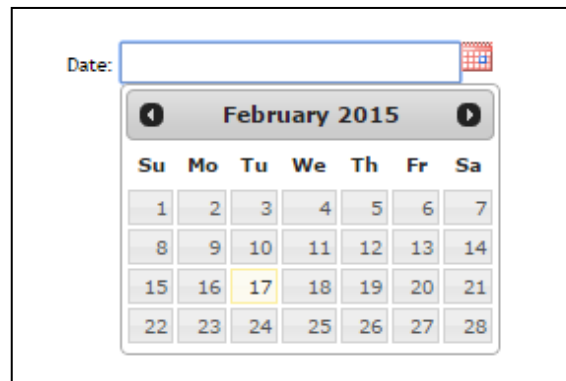
**Figura 2.6: Ejemplo de un menú con efecto de acordeón.**  
**Fuente: tomada de la página oficial de jQuery.**

2. Diálogo (Dialog): es una ventana emergente que contiene un título y un área de contenido. Una ventana de diálogo puede ser desplazada, cambiada de tamaño y cerrada. Permite mostrar un mensaje específico, incluir imágenes o inclusive, contenido interactivo. Una de sus variaciones, muy utilizada, es la ventana modal, que consiste en superponer esta ventana sobre el resto del contenido de la página, inhabilitando el uso del resto de la página con propiedades CSS, obligando al usuario a realizar alguna acción tales como seleccionar un valor de algún menú, rellenar un formulario, entre otros. En la imagen 2.7 se puede observar el efecto de diálogo en forma de ventana modal.



**Figura 2.7. Ejemplo de un cuadro de Diálogo.**  
**Fuente: tomada de la página oficial de jQuery.**

3. Calendario gráfico (Datepicker): es un plugin configurable que permite incorporar la funcionalidad de un calendario a una página web. Se puede personalizar el formato e idioma de la fecha, restringir los rangos de fechas seleccionables y añadir botones y otras opciones para navegar fácilmente entre las fechas (ver Figura 2.7).



**Figura 2.8. Ejemplo de un calendario implementando con Datepicker.**

**Fuente: tomada de la página oficial de jQuery.**

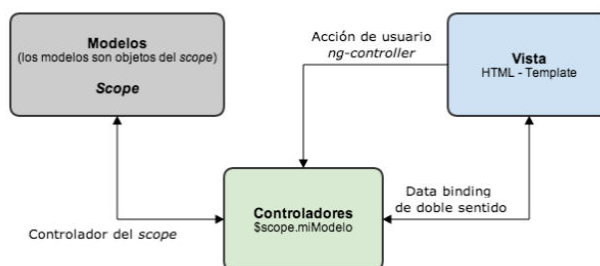
## 2.6.7 AngularJS

AngularJS es un framework JavaScript de código abierto mantenido por Google, que ayuda con la gestión de lo que se conoce como aplicaciones de una sola página, cuyo objetivo es proporcionar una experiencia de usuario más fluida. Su objetivo es facilitar la programación y las pruebas de aplicaciones del lado del cliente a través de la introducción de una estructura modelo, vista y controlador (MVC).

AngularJS proporciona el patrón MVW (Modelo, Vista, Lo que sea), en el caso del desarrollo de la aplicación Simbiosis se decidió usar MVC (Modelo, Vista, Controlador). Divide una aplicación en tres capas interconectadas que permiten separar el cómo el usuario ve los datos a cómo estos se gestionan e interpretan, y cómo se almacenan. Este patrón incrementa la posibilidad de reutilización del proyecto y la flexibilidad del mismo.

El modelo representa la información con la que la aplicación trabaja. La vista presenta el modelo con el que el usuario interactúa. Los controladores gestionan los eventos que se dan en la vista, normalmente generados por interacciones del usuario. En la figura 2.9 se puede ver

el flujo de control del patrón MVC con AngularJS, por lo que es un poco diferente de lo habitual. El usuario realiza una acción en la vista, luego el controlador trata la acción de entrada y modifica los datos del modelo como sea necesario. Al lograr establecer conexiones entre la interfaz de usuario y la lógica de una aplicación (Data Binding) en ambos sentidos, los cambios dentro del modelo de datos se reflejan automáticamente en la interfaz y viceversa.



**Figura 2.9: Modelo MVC con AngularJS. Elaboración propia.**

Para complementar las tecnologías que fueron utilizadas en el desarrollo de este TEG, se debe mencionar las del lado del servidor, el cual posee la habilidad de correr programas que interactúan con los clientes o las páginas de los sitios web. Estos programas residen en los servidores web y permiten tener interactividad en los sitios web. Por ejemplo, si en un sitio web se requiere que todos los usuarios ingresen mediante autenticación por contraseña, la base de datos que contiene los nombres de usuario y contraseñas para dicho sitio, y el programa que ofrece la autenticación de acceso se basará en el servidor y no en la página web como tal.

### 2.6.8 Servidor Web Apache

De acuerdo a Apache (2014), se define como un software servidor de páginas Web de código abierto diseñado para ser un servidor Web potente y flexible que pueda funcionar en la más amplia variedad de plataformas y entornos.

Apache presenta entre sus principales características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración. La arquitectura del servidor Apache es modular, lo cual implica que puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con la Interfaz de Programación de Aplicaciones necesaria. El servidor consta de una sección núcleo y diferentes

módulos que brindan muchas funcionalidades que podrían considerarse básica para un servidor Web.

Entre las ventajas de utilizar apache se encuentran que es multiplataforma, extensible, fácil para conseguir ayuda y/o documentación y es gratuito.

### **2.6.8 Lenguaje Preprocesador de Hipertexto**

De acuerdo a PHP (2013) se define el Lenguaje Preprocesador de Hipertexto (PHP por sus siglas en inglés), como un lenguaje de programación de código abierto, interpretado, de alto nivel, utilizado del lado del servidor, lo que significa que el código es ejecutado en el servidor, generando código HTML y enviándolo al cliente.

Está diseñado originalmente para la creación de páginas web de contenido dinámico de una manera rápida y sencilla, aunque ofrece otras funcionalidades como procesar información de formularios, acceder a bases de datos, entre otros. Se caracteriza por ser simple para el principiante y a su vez ofrece características avanzadas a los desarrolladores profesionales.

Soporta la mayoría de servidores web usados en la actualidad, el de uso común hoy día es Apache; y puede ser utilizado en cualquiera de los principales sistemas operativos como UNIX (Linux, Solaris, Debian, Ubuntu.) y Microsoft Windows, entre otros, sin costo alguno.

Cuando un cliente hace una petición al servidor para que le envíe una página web, sucede lo siguiente del lado del servidor:

1. El servidor web ejecuta el intérprete PHP el cual procesa el script solicitado que generara el contenido de manera dinámica (obtener la información de una BD, por ejemplo).
2. El resultado es enviado desde el intérprete al servidor.
3. El servidor es quien lo envía al cliente en un formato comprensible por él.

PHP además permite la conexión a diferentes tipos de servidores de BD como: MySQL, PostgreSQL, Oracle, entre otros.

Este lenguaje de programación orientado a objetos, a procedimientos o incluso una mezcla de ambos, puede estar incluido en el código “.html” o puede ser un archivo externo que debe ser guardado con extensión “.php”.



PHP proporciona al programador las herramientas necesarias para realizar el trabajo de una manera rápida y muy eficiente. Entre sus principales características destacan las siguientes:

- Familiaridad: la sintaxis del lenguaje es similar a la de otros tales como C y Perl, es por ello que es de fácil adaptación y rapidez de uso para cualquier programador.
- Simplicidad: la sencillez del lenguaje permite que los programadores puedan generar código en menor tiempo.
- Eficiencia: PHP incorpora su propio sistema de administración de memoria ideal para ambientes multiusuario, además de mecanismos para el manejo de sesiones y eliminación de asignaciones de memoria innecesarias, con el fin de conformar un sistema robusto, eficiente y estable.
- Seguridad: proporciona a los desarrolladores y administradores un conjunto eficiente de garantías de seguridad.
- Velocidad: PHP está escrito en lenguaje C, por lo que se ejecuta rápidamente utilizando poca memoria. Además es posible utilizarlo como un módulo de Apache e integrarlo con la tecnología Hip Hop para PHP, el cual toma el código PHP y lo transforma en código compilable C++, acelerando la ejecución del código lo cual disminuye el uso de los recursos del servidor en un 50%.

### **2.6.10 CoffeeScript**

Según Burnham (2011), CoffeeScript es un lenguaje de programación que se compila en JavaScript. La sintaxis de este lenguaje está basada en lenguajes como Ruby o Python y trata de tomar lo mejor de JavaScript para llevarlo a una forma simple de escribir. No solo elimina el uso de llaves ({}), debido a que es indentado, sino que incluye características de comprensión de arreglos y reconocimiento de patrones.

CoffeeScript tiene una correspondencia uno a uno con JavaScript, esto quiere decir que no hay degradación del rendimiento. En algunos casos, muchos de los códigos escritos en JavaScript terminan siendo más eficientes luego de ser migrados a CoffeeScript, debido a ciertas optimizaciones que el compilador puede realizar.

En la mayoría de los casos, mediante el uso de CoffeeScript se escribe aproximadamente un tercio del código, en cuanto a líneas se refiere, en comparación con las líneas de código escritas usando JavaScript. Esto permite obtener mejores resultados durante el desarrollo de una aplicación web.

## **2.7 Software de Control de Versiones Git**

De acuerdo a Git Book (2013), es un software libre y de código abierto que se encarga del control distribuido de versiones y la administración de código fuente. Cada directorio de trabajo en Git es un repositorio completo con toda la historia sobre cada uno de sus archivos y posee plena capacidad de hacer seguimiento de versiones, esto permite recuperar versiones antiguas de los datos, o examinar el historial de cambios de los mismos. No depende de acceso a la red ni de un servidor centralizado.

Git puede acceder al repositorio a través de redes, lo que le permite ser usado por personas que se encuentran en distintos computadores, fomentando la colaboración. Entre sus principales características tenemos:

- Implementa un versionado de directorios.
- Posee un historial de versiones.
- Realiza envíos atómicos.
- Crea un versionado de metadatos.
- Efectúa una manipulación consistente de datos.

## **2.7 Bitbucket**

Bitbucket es un servicio web colaborativo desarrollado por Atlassian que permite el alojamiento, administración y la compartición de repositorios Git en la nube. Bitbucket ofrece planes comerciales y cuentas gratis. Se encuentra desarrollado bajo el lenguaje de programación Python (Atlassian, 2014).

## **2.9 Licenciamiento Creative Commons**
















Según Creative Commons (2011), son licencias inspiradas en la licencia GPL (General Public License) que permiten copiar y distribuir las obras intelectuales siempre y cuando se

respeten las condiciones escogidas por los autores. La licencia Creative Commons consta en la actualidad con seis licencias básicas, las cuales son:

- **Atribución CC BY:** esta licencia permite a otros distribuir, mezclar, ajustar y construir sobre su trabajo, incluso comercialmente siempre y cuando se de crédito a la creación original. Este es el más servicial de las licencias ofrecidas. Recomendado para la máxima difusión y el uso de materiales sujetos a licencia.
- **Atribución – Compartir Igual CC BY-SA:** esta licencia permite a otros mezclar, modificar, y construir sobre su trabajo, incluso para fines comerciales, siempre y cuando se de crédito y la licencia de sus nuevas creaciones en los términos idénticos. Esta licencia es a menudo comparada con copyleft licencias de software libre y de código abierto. Todas las nuevas obras basadas en el original llevan la misma licencia, por lo que cualquier derivado también permite el uso comercial. Esta es la licencia usada por Wikipedia.
- **Atribución – Sin Obras Derivadas CC BY-ND:** esta licencia permite la redistribución, comercial y no comercial, siempre y cuando se pase sin cambios y en su totalidad, con el crédito.
- **Atribución – No Comercial CC BY-NC:** esta licencia permite a otros, modificar, y aprovechar su trabajo para fines no comerciales, y aunque sus nuevas obras también se debe reconocer y no comercial, no tienen la licencia de sus productos derivados en los mismos términos.
- **Atribución – No Comercial – Compartir Igual CC BY-NC-SA:** esta licencia permite a otros mezclar, modificar, y aprovechar su trabajo para fines no comerciales, siempre y cuando se de crédito y la licencia de sus nuevas creaciones en los términos idénticos.
- **Atribución – No Comercial – Sin Obras Derivadas CC BY-NC-N:** esta licencia es la más restrictiva de las seis licencias principales, que sólo permite que otros puedan descargar sus obras y compartirlas con los demás, siempre y cuando se de crédito, pero no se pueden cambiar en cualquier forma o utilizarlos con fines comerciales.

Por ser un trabajo desarrollado para la UCV, se debe implementar el licenciamiento Creative Commons, usando un licenciamiento no comercial bajo la misma licencia, el cual permite que la aplicación pueda ser reutilizada, modificada y usada para fines no comerciales

dentro de la UCV. En la Figura 3.10 se puede observar los tipos de licencias ofrecidos por Creative Commons Venezuela.

Licencias de Creative Commons			¿Usarías de forma comercial?	¿Modificarías?
Atribución	BY			
Compartir bajo la Misma Licencia	BY-SA			Solo bajo una licencia similar
Sin Obras Derivadas	BY-ND			
No comercial	BY-NC			Solo con propósitos no comerciales
No comercial bajo la Misma Licencia	BY-NC-SA			Solo con propósitos no comerciales
No comercial Sin Obras Derivadas	BY-NC-ND			

En todas es necesario darle el crédito al autor

norfipc.com

**Figura 2.10: Tipos de Licencias de Creative Commons Venezuela. Creative Commons Venezuela (2013).**

Una vez descritas las tecnologías utilizadas, tanto en el lado del servidor como en el lado del cliente, en el siguiente capítulo se explicará la aplicación de AgilUs como metodología de desarrollo de software en la aplicación Simbiosis.

## **Capítulo 3:**

### **Marco Aplicativo**

En este capítulo se aplica la metodología escogida para el desarrollo de la aplicación, la cual fue una adaptación de AgilUs, metodología ágil en donde la usabilidad, usuarios e interacciones son tan importantes como los procesos y herramientas, y que el funcionamiento del software sea más importante que una documentación exhaustiva. En este sentido se describe el contexto de desarrollo, y cada una de las fases de la metodología para la gestión de los procesos relacionados con las solicitudes estudiantiles durante el proceso de formalización de inscripción de Seminarios y TEG en la Escuela de Biología.

#### **3.1 Fase I: Requisitos**

En esta fase se aplicaron diversas técnicas de indagación para obtener la información necesaria para un posterior análisis de requerimientos planteados para la aplicación que manejará el proceso de formalización de inscripción de Seminarios y TEG. Las técnicas empleadas fueron: tormenta de ideas, sondeo, perfiles de usuario y listado de requerimientos funcionales y no funcionales los cuales serán descritos a continuación.

##### **3.1.2 Entrevista**

Esta actividad se llevó a cabo en entrevistas no estructuradas entre el grupo de desarrollo, el director de la Escuela de Biología, el profesor Jesús Romero con participación de la secretaría, donde el objetivo fue conocer la situación actual del proceso de inscripción de Seminarios y TEG en la escuela, se recolectaron conocimientos acerca de la problemática existente y se lograron identificar todas y cada una de las necesidades presentes. Al ser una entrevista no estructurada las preguntas fueron surgiendo de acuerdo a las respuestas aportadas por los entrevistados. Entre las preguntas más importantes que se plantearon en conjunto con el director de la Escuela de Biología se encontraron:

- ¿Cómo se lleva a cabo el proceso de formalización de inscripción de Seminarios y TEG en la Escuela de Biología?
- Describa brevemente el flujo del proceso.
- ¿Quiénes participan en el proceso de inscripción?

- ¿Cuáles son las actividades de cada participante dentro del proceso?

De la entrevista realizada se logró obtener la forma como es llevado a cabo el proceso de inscripción y quiénes son los actores del mismo, así como algunas ideas acerca de funcionalidades y opciones que debe poseer la aplicación, entre las cuales se destacan las siguientes:

- Colocar la opción de un calendario para que el personal administrativo maneje las fechas fácilmente para enviar las notificaciones.
- Enviar notificaciones a los estudiantes, mediante correos electrónicos y/o redes sociales cuando se haya registrado un evento referente al proceso.
- Agregar una opción de consulta de datos de estudiantes en caso de que el docente.
- Permitir seleccionar la lista de correos de un grupo (estudiantes, docentes o ambos) para enviar notificaciones y no tener que introducir las direcciones de correo cada vez que se desee.
- Crear una vista que muestre sólo a los estudiantes que han formalizado que tienen clasificándolos por tipo de inscripción y departamento.
- Integrar la aplicación a desarrollar con CONEST y con otros pertenecientes a la escuela (Biotec, por ejemplo) a través de servicios web, de manera que en un futuro formen parte de un sistema único.
- Permitir la visualización de información confidencial (como las calificaciones, créditos y datos personales) sólo para aquellos estudiantes que se encuentren cursando alguna de las asignaturas del proceso (Seminario I o II y TEG).
- Crear un mecanismo que permita a al personal administrativos ingresar la cédula de un estudiante y eliminar la formalización o editarla.
- Estudiar la posibilidad de realizar entregas a través del sitio web para facilitar a tutores y jurados la corrección de los documentos entregables.
- Se desea contar con una funcionalidad para crear, modificar y eliminar una formalización para un estudiante específico.

### 3.1.3 Tormenta de ideas

La idea principal de realizar esta actividad fue exponer al personal administrativo la problemática actual en referencia al proceso manual de formalización de inscripción de Seminarios y TEG en la Escuela de Biología, para que luego estos aportasen todas las ideas, referentes al tema, sin importar la calidad de estas y así crear nuevas ideas a partir de las mencionadas hasta el momento.

Una vez identificado el problema presente gracias a la entrevista no estructurada, se logró intercambiar ideas entre el equipo de desarrollo y el personal involucrado, para dar solución al problema. De aquí se obtuvieron opiniones sobre la problemática existente al realizar todo el proceso de forma totalmente manual y se plantearon ideas para dicha problemática, se definieron las ideas principales acerca de las funcionalidades que deben conformar el módulo de formalización de la aplicación, los roles y perfiles de usuario necesarios. Luego se realizó una segunda reunión en la cual se mostró un boceto en papel de la solución planteada en la primera tormenta de ideas al director de la escuela quien compartió su opinión e intercambio ideas hasta llegar a un acuerdo de las principales funcionalidades que contendría la aplicación. En la figura 3.1 se muestra el prototipo en que se mostró en la segunda reunión basados en las ideas de la primera reunión.

Formalizar Inscripción

Paso 4 - Datos del Jurado

Tipo  Principal  Suplente

Profesor:  Departamento:

Nombres:  Apellidos:

Tlf Celular:  Correo electrónico:

Jurado Principal 1	Nombre y Apellido
Jurado Principal 2	Nombre y Apellido
Jurado Suplente 1	Nombre y Apellido
Jurado Suplente 2	Nombre y Apellido
Jurado Suplente 3	Nombre y Apellido

Al hacer click en finalizar debe aparecer una notificación que diga que la formalización ha sido realizada con éxito, que debe imprimir su planilla y llevarla a la dirección de la escuela en la fecha pautada. (de alguna manera debería aparecer la opción de visualizar planilla, imprimir planilla y guardar planilla)

[Finalizar](#)

Figura 3.1 Prototipo de Interfaz planteada en la tormenta de ideas. Elaboración propia.

Adicionalmente, se ideó el nombre de “Simbiosis” para la aplicación, ya que es un proyecto que nace de la idea del trabajo en conjunto entre la Escuela de Biología y la Escuela de Computación de la UCV, para desarrollar una aplicación que gestione el proceso de formalización y seguimiento de la inscripción de Seminarios y TEG, incentivando la participación de forma activa de estudiantes, docentes y el personal de las distintas comisiones involucradas en el proceso.

### 3.1.4 Sondeo

El siguiente, es un sondeo que se realizó mediante la página web: <http://www.e-encuesta.com>, fue sobre una población de treinta (30) personas pertenecientes a la Escuela de Biología, los cuales fueron estudiantes, docentes y personal administrativo. Este estudio tiene la finalidad de obtener opiniones referentes a una aplicación web donde se puedan gestionar procesos de forma automática, lo cual facilitó la toma de decisiones acerca de los requerimientos a desarrollar. Se realizaron un total de ocho (8) preguntas, diseñadas por expertos en Interacción Humano Computador de la Escuela de Computación.

Para la primera pregunta se deseó verificar el rol que cumplen las personas que respondieron el sondeo, ya que se desea que tanto estudiantes, docentes y el personal de las distintas comisiones involucradas en el proceso colaboren en las diferentes etapas del proyecto. En la figura 3.2 se visualiza el porcentaje de estudiantes con respecto a profesores y personal administrativo, importante recalcar que el sesenta por ciento (60%) de los encuestados son estudiantes, los cuales serán los principales usuarios de la aplicación web.

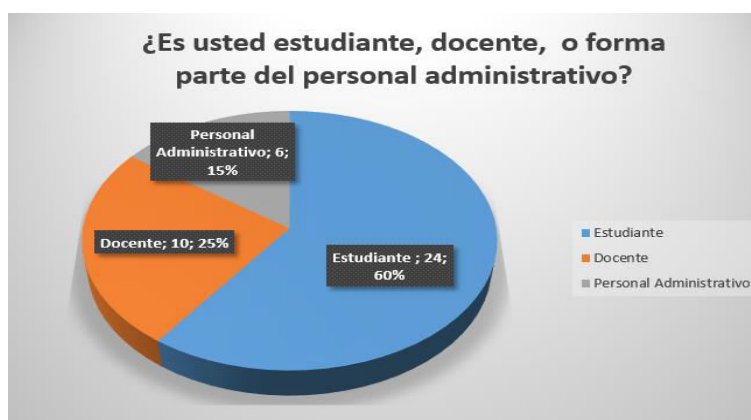
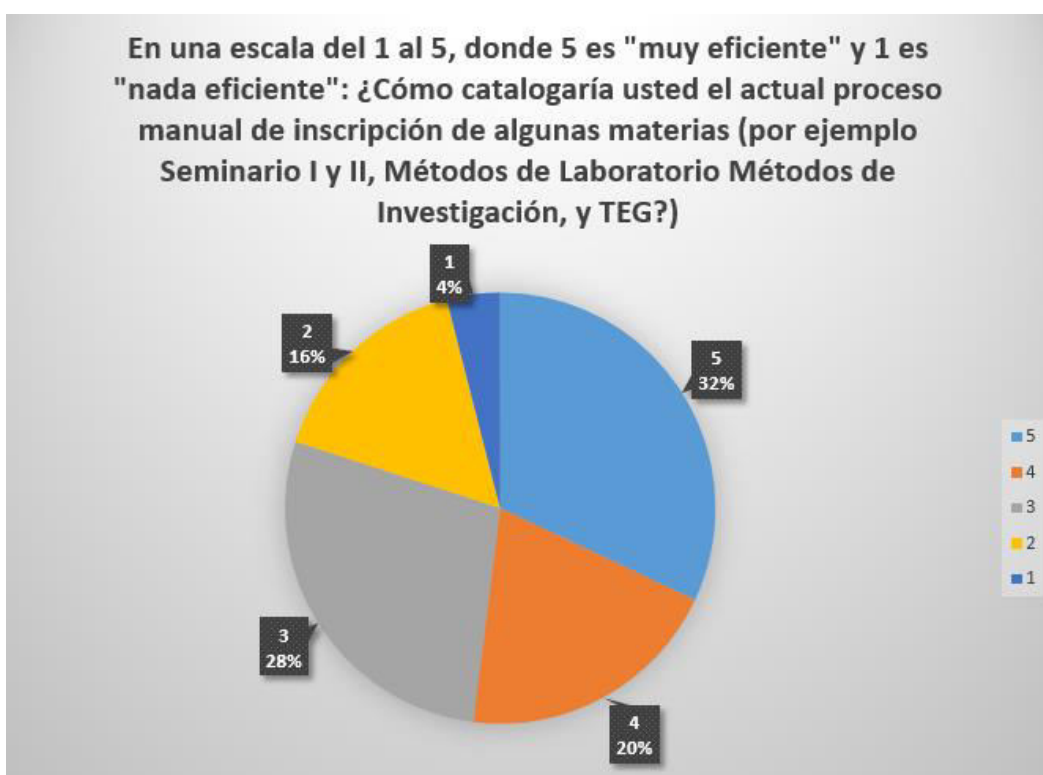


Figura 3.2: Rol que cumplen los encuestados en la Facultad de Ciencias. Elaboración propia.



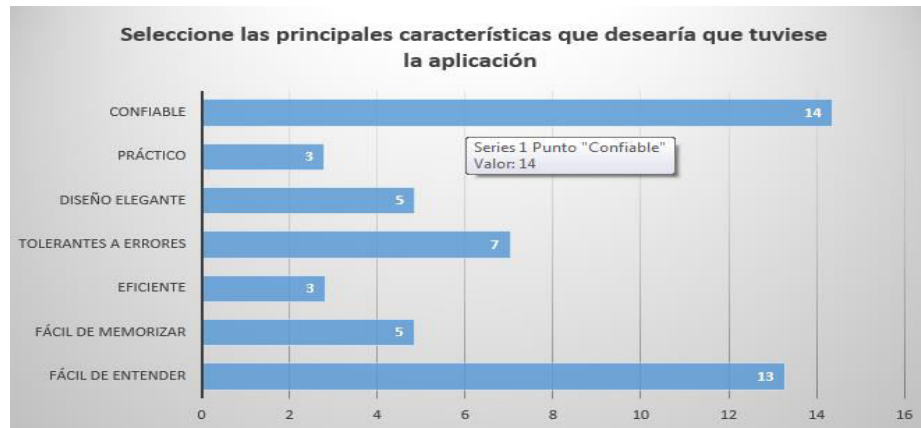
En la segunda pregunta se quiso validar la percepción que tienen los usuarios de los procesos de inscripción de algunas materias en la Escuela de Biología: “¿Cómo catalogaría usted el actual proceso manual de inscripción de algunas materias (por ejemplo Seminario I y II, Métodos de Laboratorio y Métodos de Investigación, y TEG?)”. En la figura 3.3 se puede ver la manera en la cual los encuestados ven los procesos de inscripción manual de algunas materias en la Escuela de Biología, el treinta y dos por ciento (32%) de los encuestados considera que los procesos de inscripción que se llevan de forma manual son muy eficientes, lo cual nos da la idea de que hay que adaptar fielmente el proceso a la versión web automatizada.



**Figura 3.3: Evaluación de inscripción de materias mediante procesos manuales. Elaboración propia.**

Seguidamente, se preguntó acerca de las características principales que debería tener la aplicación: “Seleccione las principales características que desearía que tuviese la aplicación”. Los cuestionados pudieron responder, con más de una opción, las distintas características que consideran pertinentes debe tener la aplicación.

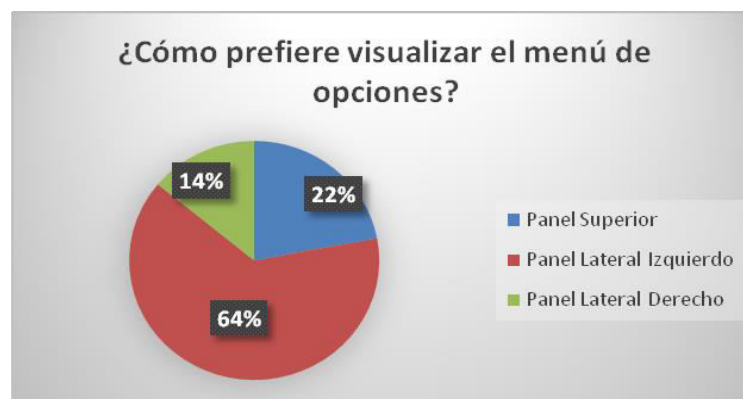
La figura 3.4 muestra que las opciones: confiable, fácil de entender y tolerante a errores fueron las opciones más relevantes escogidas por los encuestados para diseñar la aplicación web, por lo serían las características principales a tomarse en cuenta para desarrollar la aplicación a desarrollar y así minimizar las funcionalidades que no se consideren tan necesarias para el usuario.



**Figura 3.4: Características deseables de la aplicación web. Elaboración propia.**

En la siguiente pregunta se deseó conocer la opinión de los participantes con respecto al posicionamiento del menú de opciones de la aplicación web, mediante la siguiente pregunta: “¿Cómo prefiere visualizar el menú de opciones?”.

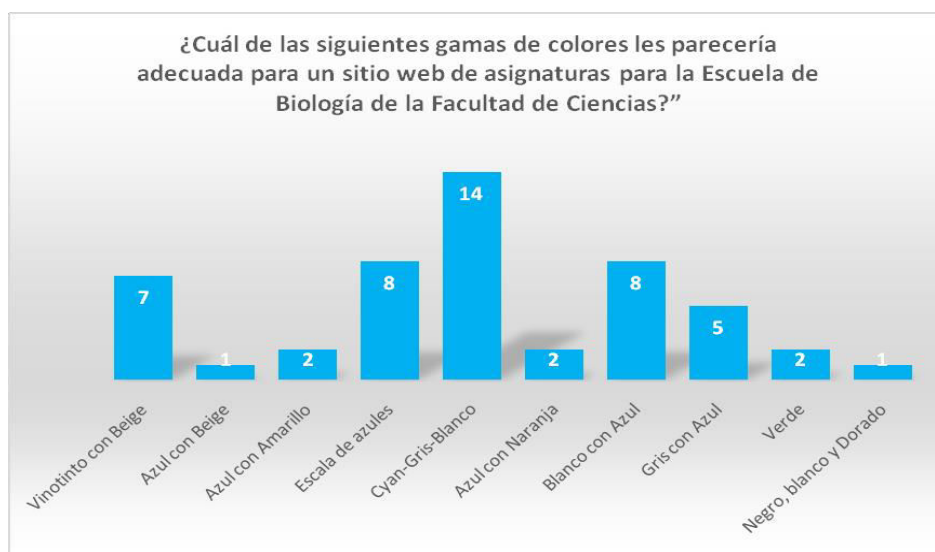
Se puede apreciar en la figura 3.5 que la mayoría de los participantes prefiere que el menú de opciones se encuentre posicionado en un panel lateral izquierdo o en un panel superior, por lo que queda de parte de los desarrolladores decidir dónde colocar este menú.



**Figura 3.5: Preferencias de visualización del menú de opciones. Elaboración propia.**

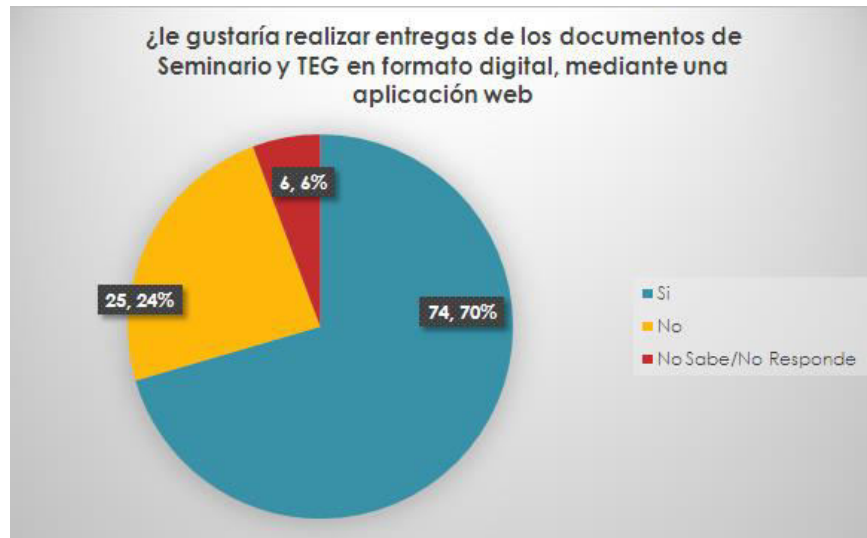
Es de suma importancia conocer la gama de colores a utilizar para la aplicación web que se desarrollará, ya que el mismo debe ser atractivo a los usuarios, y a su vez se desea que los colores representen a la Escuela de Biología de la Facultad de Ciencias de la Universidad Central de Venezuela, así que siguiendo el patrón de preguntas acerca de la apariencia del sitio web: “¿Cuál de las siguientes gamas de colores les parecería adecuada para un sitio web de asignaturas para la Escuela de Biología de la Facultad de Ciencias?”.

Con respecto a la gama de colores a utilizar, se les propuso una serie de combinaciones permitiéndoles agregar más opciones. En la figura 3.6, se refleja una clara diferencia entre la combinación cian-gris-blanco con catorce (14) participantes y las demás opciones, la cual parece ser la combinación más adecuada.



**Figura 3.6: Gama de colores considerada adecuada para la aplicación.**  
**Elaboración propia.**

Para la pregunta siguiente: “¿Le gustaría realizar entregas de los documentos de Seminario y TEG en formato digital, mediante una aplicación web?”, se realizó para evaluar la posibilidad de que se realicen entregas de documentos en formato digital mediante una aplicación web. La figura 3.7 muestra que un setenta y cuatro por ciento (74%), se mostraron de acuerdo con que se realicen entregas digitales de los documentos, lo cual coincide con la idea planteada por el director de la escuela durante las reuniones.

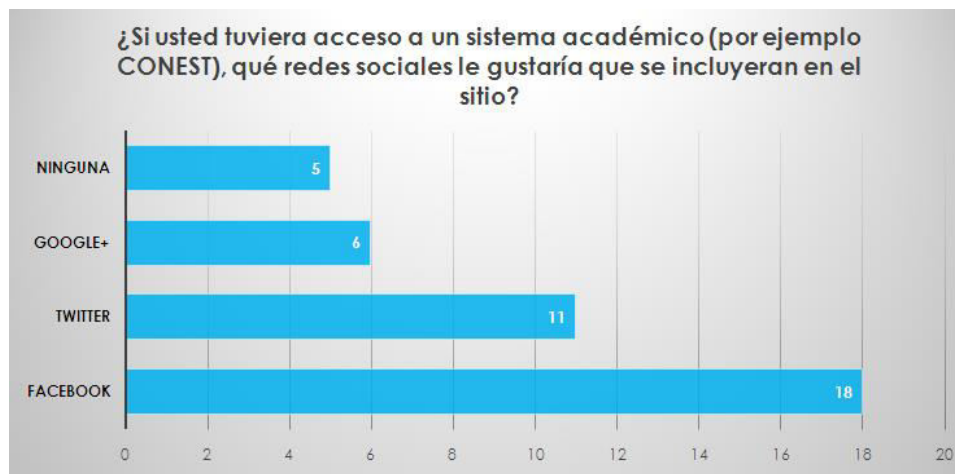


**Figura 3.7: Opción de entregas digitales en la aplicación web.**

**Elaboración propia.**

Seguidamente, se indagó en las redes sociales que los participantes consideran que se deberían agregar al nuevo sistema, para lo cual se utilizó la siguiente pregunta: “¿Si usted tuviera acceso a un sistema académico (por ejemplo CONEST), qué redes sociales le gustaría que se incluyeran en el sitio?”

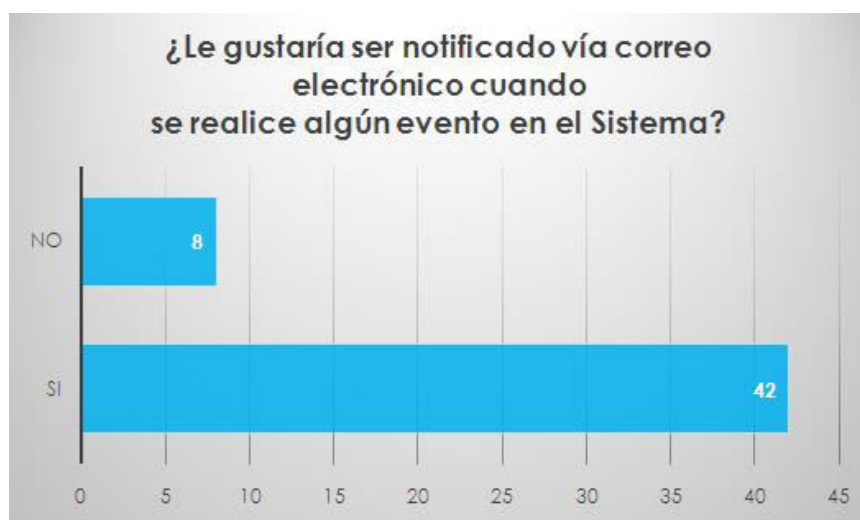
En la figura 3.8 se aprecia que las redes sociales Facebook y Twitter son las que los encuestados consideran más adecuadas para compartir contenido en la aplicación.



**Figura 3.8: Redes sociales consideradas adecuadas para la aplicación.**

**Elaboración propia.**

Finalmente, se utilizó la pregunta: “¿Le gustaría ser notificado vía correo electrónico cuando se realice algún evento en la aplicación?”, para conocer si los participantes están de acuerdo con que se notifique cualquier evento de la aplicación web mediante el correo electrónico. Se puede apreciar que cuarenta y dos (42) de los encuestados se encuentran de acuerdo con que se realice una notificación de cualquier evento dentro del proceso de formalización de inscripción mediante el correo electrónico (figura 3.9).



**Figura 3.9: Redes sociales consideradas adecuadas para la aplicación.**

**Elaboración propia.**

Una vez analizadas cada una de las preguntas del sondeo, se pudo tener una idea más clara de lo que los usuarios desean para la aplicación a desarrollar. Fue importante tomaren cuenta la opinión de cada uno de los participantes para lograr tomar decisiones que correspondan en su mayoría con lo que estos manifestaron.

Una vez analizada la tormenta de ideas y el sondeo, se la hizo descripción y catalogación de los distintos perfiles de usuarios de la aplicación a desarrollar.

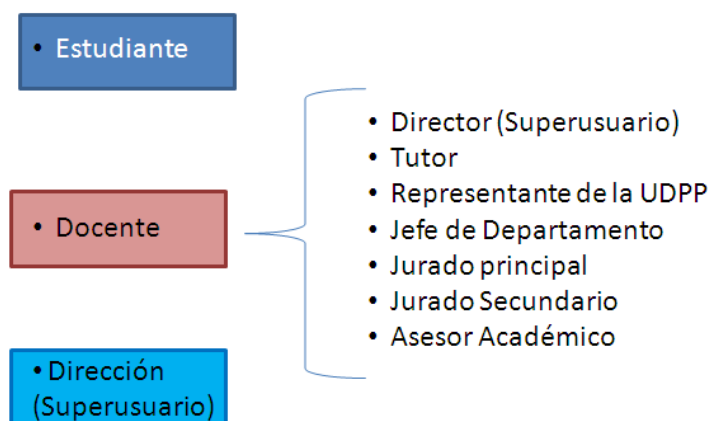
### **3.1.5 Perfiles de Usuarios**

Los Perfiles de usuarios son fundamentales en cualquier proyecto centrado en el usuario; ya que describen a los usuarios previstos de la aplicación y detallan las características relevantes de los mismos.

Esta definición de perfiles de usuarios coincide con la propuesta en el método de desarrollo de software que se aplicará en este proyecto; en tal sentido, se utilizarán estos perfiles en este TEG, ya que se trata de una técnica básica en cualquier proyecto enfocado en la usabilidad como el que se está desarrollando.

Dentro de la iteración de gestión de usuarios existen diversos tipos, los cuales asumen un rol dependiendo de las responsabilidades que desempeñan dentro del proceso de formalización de inscripción de Seminarios y TEG. Los involucrados en el proceso son los empleados administrativos de la Escuela de Biología, los estudiantes y docentes.

La aplicación web cuenta con tres (3) tipos de usuarios principales: dirección, docente y estudiante. A su vez el perfil de profesor, puede tomar varios roles dependiendo de sus actividades durante el proceso, los cuales son: director, tutor, representante de la UDPP, jefe de departamento, jurado principal, jurado secundario y asesor académico (ver figura 3.10).



**Figura 3.10: Tipos de usuarios y roles en la aplicación web. Elaboración propia.**

A continuación se describirán cada uno de estos perfiles:

- Estudiante: en este perfil se encuentran la mayoría de los usuarios de la aplicación Simbiosis, los cuales son estudiantes de pregrado de licenciatura en Biología. Los estudiantes solo necesitan tener conocimientos básicos acerca de navegación web para acceder al sitio web. Este tipo de usuario podrá conseguir información referente a la asignatura que cursa y realizar distintas acciones a fin de formalizar su inscripción. Se desea que este tipo de usuario ingrese eventualmente al sistema para verificar el

estatus de su formalización. Es importante su apoyo a lo largo de este desarrollo ya que la aplicación está dirigida en su mayoría a este perfil.

- **Dirección:** es el encargado de ingresar los datos de los jefes de departamentos, representantes de la UDPP, docentes y estudiantes de la Escuela de Biología de la Facultad de Ciencias, ya sea por un archivo o manualmente. Este usuario debe ingresar al sistema al inicio de cada semestre para agregar, modificar o eliminar la lista de estudiantes, docentes y docentes o entre semestre para modificar los datos de algún registro en específico.

El perfil de dirección cuenta con todos los privilegios de la aplicación por lo que cada acción realizada por los otros tipos de usuario, puede ser realizada por éste también. El interés por parte de este perfil de usuario está en proporcionar los recursos a la secretaría para llevar la gestión de ciertos procesos inherentes al proceso, los cuales son realizados de forma manual.

- **Docente:** es el encargado cumplir distintos roles según sea el caso (director, tutor, representante de la UDPP, jefe de departamento, jurado principal, jurado secundario y asesor académico).

Ningún perfil de usuario anteriormente descrito necesita tener experiencia previa con la aplicación, solo necesitará conocimientos básicos de computación para manejar las funciones que desee según la autorización que posea.

Una particularidad que engloba al perfil de director y al perfil docente con el rol de director es que ambos poseen los mismos privilegios (superusuario), y pueden manejar el mismo el contenido de la aplicación web, mientras que los otros integrantes del perfil docente no poseen dichos privilegios.

A continuación, y siguiendo con la fase de requisitos de AgilUs, se presentan los requerimientos funcionales y no funcionales identificados de la aplicación.

### **3.1.6 Requerimientos Funcionales y no Funcionales**

En base a la observación realizada durante las reuniones con el personal administrativo y el encargado de gestionar y coordinar el proceso de inscripción de Seminarios y TEG, se recopilaron diversos requisitos considerados al momento del desarrollo de las funcionalidades de gestión en la aplicación web. A continuación se describirán cada uno de los requerimientos

funcionales que forman parte de la aplicación separados según el perfil de los diferentes usuarios:

Requerimientos para el perfil administrador:

- Iniciar/Cerrar Sesión: el usuario podrá iniciar sesión con su cédula y contraseña creada en un principio en la aplicación, así mismo esta contraseña podrá ser cambiada si así lo desea.
- Iniciar/Detener/Finalizar Proceso de inscripción: este usuario tendrá privilegios para iniciar, detener o finalizar el proceso de formalización de la inscripción del semestre en curso.
- Agregar, modificar o eliminar estudiantes y docentes: esta carga puede ser igualmente mediante un archivo Excel o manual. Para los docentes es necesario ingresar la cédula, el nombre, apellido y el correo electrónico.
- Cargar listados (Estudiantes y Profesores): mediante un archivo en formato Excel, el usuario puede cargar en la aplicación un listado por bloques de estudiantes y profesores.
- Finalizar Solicitudes: durante cualquier momento durante el proceso de formalización este usuario puede dar por finalizada una solicitud.
- Designar roles de Representantes de la UDPP: el usuario puede designar a docentes para que cumplan el rol de Representantes de la UDPP.
- Designar roles de Jefes de Departamento: de igual forma mediante esta funcionalidad, el usuario puede designar a un docente para cumplir el rol Jefe de Departamento.
- Visualizar/Imprimir Proyectos, Planillas y Listados de estudiantes: este usuario puede visualizar o imprimir cualquier documento o planilla relacionada con la inscripción de un estudiante.
- Modificar/Anular formalización: una vez realizada la formalización de inscripción por parte del estudiante, el usuario puede modificar los datos de la formalización o anularla.
- Enviar Comunicados: el usuario puede notificar vía correo electrónico cualquier cambio o evento que se realice durante el proceso de formalización, tanto a usuarios pertenecientes al grupo docente como a estudiantes. Asimismo puede enviar comunicados de interés para todo el personal involucrado en el proceso.



- **Cambiar Contraseña:** durante cualquier momento el usuario puede realizar un cambio de contraseña.
- **Generar Reportes:** el usuario puede ver listados y estadísticas de los estudiantes inscritos, aquellos que han formalizado o no inscripción, así como los listados de docentes por departamento.

#### Requerimientos para el perfil Docente:

- **Iniciar/Cerrar sesión en la aplicación:** al igual que el administrador podrá iniciar y cerrar sesión y podrá cambiar la contraseña. Al intentar iniciar sesión debe resolver un captcha para validar que sea un usuario real quien intenta iniciar sesión.
- **Cambiar Contraseña:** durante cualquier momento el docente puede realizar un cambio de contraseña.
- **Actualizar Datos:** el docente puede actualizar sus datos personales, con excepción de la cédula y nombre completo ya que esta información es extraída de CONEST.
- **Verificar solicitudes:** el docente puede verificar sus solicitudes en referencia a tutorías o su participación como jurado. Además puede verificar la información de la inscripción de los estudiantes bajo su tutoría, así como el documento cargado en la aplicación.
- **Aceptar/Rechazar solicitudes:** el usuario podrá rechazar solicitudes que le sean realizadas para cumplir el rol de tutor o jurado, deberá ingresar observaciones del motivo del rechazo de la propuesta.
- **Cargar Notas:** una vez un estudiante haya culminado la presentación del Seminario I o II o el TEG el profesor puede cargar la nota en la aplicación.
- **Leer Notificaciones:** el usuario puede visualizar y marcar como leídas las observaciones y notificaciones relacionadas sobre su inscripción.
- **Enviar comunicados:** el docente podrá notificar vía correo electrónico cualquier evento que se realice durante el proceso de formalización, tanto a usuarios pertenecientes al grupo docente como a estudiantes.

#### Requerimientos del Módulo de Estudiantes:

- Iniciar/Cerrar sesión en la aplicación: al igual que el administrador podrá iniciar y cerrar sesión y podrá cambiar la contraseña. Al intentar iniciar sesión debe resolver un captcha para validar que sea un usuario real quien intenta iniciar sesión.
- Actualizar Datos: el estudiante podrá actualizar sus datos personales, con excepción de la cédula y nombre completo ya que esta información es extraída de CONEST.
- Formalizar inscripción: el usuario podrá completar los datos relativos a su inscripción.
- Cargar Documento: el estudiante una vez formalizada su inscripción, puede cargar el documento al que hace mención en la formalización.
- Cambiar Contraseña: durante cualquier momento el usuario puede realizar un cambio de contraseña.
- Visualizar/Imprimir Planillas: este usuario puede visualizar o imprimir la planilla de formalización de su inscripción relacionada con la inscripción de un estudiante.
- Leer Notificaciones/Observaciones: el usuario puede visualizar y marcar como leídas las observaciones y notificaciones relacionadas sobre su inscripción.

Los anteriores son los requerimientos funcionales que se realizarán para la aplicación de este Trabajo Especial de Grado, los cuales describen todas las tareas que podrán ejecutar los usuarios tanto en el portal de creación de sitios web de asignaturas como en los diferentes sitios web generados por éste. Una vez establecidos los requerimientos funcionales, se enunciarán los requerimientos no funcionales que forman parte de este sistema.

A continuación se exponen cada uno de los requerimientos No Funcionales de la aplicación.

- Usabilidad: AgilUs busca como objetivo principal incluir al usuario en todas las etapas de su ciclo de vida, con la finalidad de realizar un software que sea fácil de utilizar, es por ello que lo ideal es que el usuario deba comprender y usar con facilidad la aplicación, es decir, debe ser de fácil uso para cualquier usuario que visite la aplicación, ofreciendo funcionalidades y una interfaz de usuario que requieran un bajo conocimiento cognitivo.
- Disponibilidad: se debe permitir usar la aplicación en cualquier momento y desde cualquier ubicación geográfica con conexión a Internet. Al momento de ser puesto en

marcha en el servidor, la aplicación debe ser capaz de estar disponible para todos los usuarios que vayan a utilizarlo sin ninguna restricción de la plataforma donde se vaya a encontrar.

- **Mantenibilidad en el tiempo:** con el objetivo de poder realizar los cambios necesarios en el menor tiempo posible a fin de garantizar la disponibilidad de la aplicación, se desea que la aplicación pueda ser mantenido fácilmente, es decir, que una vez culminado la aplicación no se necesite a una persona verificando constantemente que la aplicación funcione correctamente.
- **Eficiencia:** es tomada en cuenta como la capacidad de que la aplicación cumpla con sus funcionalidades en un tiempo aceptable para el usuario. La aplicación debe ser capaz de responder a las peticiones de muchos usuarios a la vez en un corto tiempo ya que de lo contrario no será satisfactorio y no será utilizado por ellos.
- **Portabilidad.** La aplicación debe poder ser accedida independientemente de la plataforma que se esté utilizando, solo debe requerir de la disposición de un navegador web (browser) sin importar el sistema operativo.
- **Escalabilidad.** Las funcionalidades son implementadas previniendo futuras mejoras sobre las mismas, de manera que pueden ser incorporadas afectando el código existente de la menor manera posible.
- **Confiabilidad.** La aplicación debe comportarse de acuerdo a lo que los usuarios esperan de este.
- **Seguridad:** se debe garantizar que toda la información contenida en la aplicación debe estar resguardada contra accesos no autorizados, mediante mecanismos de autenticación y sesiones que no permitan la fuga de información. La seguridad se tomará en cuenta para validar el inicio de sesión de los usuarios, tanto del grupo docente como de los estudiantes, esto permitirá verificar que personas que no pertenezcan a la comunidad de la Escuela de Biología puedan ingresar a la aplicación.
- **Tolerancia a fallas:** en el exacto momento que ocurra una falla, ya sea intencional o no, la aplicación debe ser capaz de responder al usuario y volver al estado que se encontraba al momento anterior de ocurrir la falla.

Los requerimientos no funcionales mencionados previamente, también se aplicaron para cada módulo de la aplicación, los cuales son una abstracción para los usuarios finales, ya

que no deben realizar ninguna tarea específica para cumplir con ellos. La idea es cumplir con todos estos aspectos para dar una mayor satisfacción a todos los usuarios. Una vez finalizada la fase de requisitos se continúa con la fase de análisis.

## 3.2 Fase II: Análisis

La segunda fase en el ciclo de vida de AgilUs, se refiere al análisis de la solución a desarrollar, en la cual se diseñan distintos prototipos de interfaz (wireframes), se especifican guías de estilo, se realizan modelos de casos de uso y objetos de dominio, para así puntualizar las funcionalidades que tendrá el producto a desarrollar. A continuación se presentarán cada una de estas actividades.

### 3.2.1 Prototipos de Interfaz

Para desarrollar esta actividad se usó una aplicación de código abierto llamada mockingbird, en la figura 3.1 podemos ver la interfaz de inicio de sesión en la aplicación, con un banner donde se encuentran los logos tanto de la universidad Central de Venezuela como el de la Facultad de Ciencias y el logo y nombre. Luego, se encuentra el panel de inicio de sesión con un método captcha de verificación, y dos hipervínculos, uno para refrescar el captcha y el otro para recordarle la contraseña al usuario.

Por último, se encuentra el pie de página con información general de la Facultad de Ciencias y enlaces de interés para los usuarios.

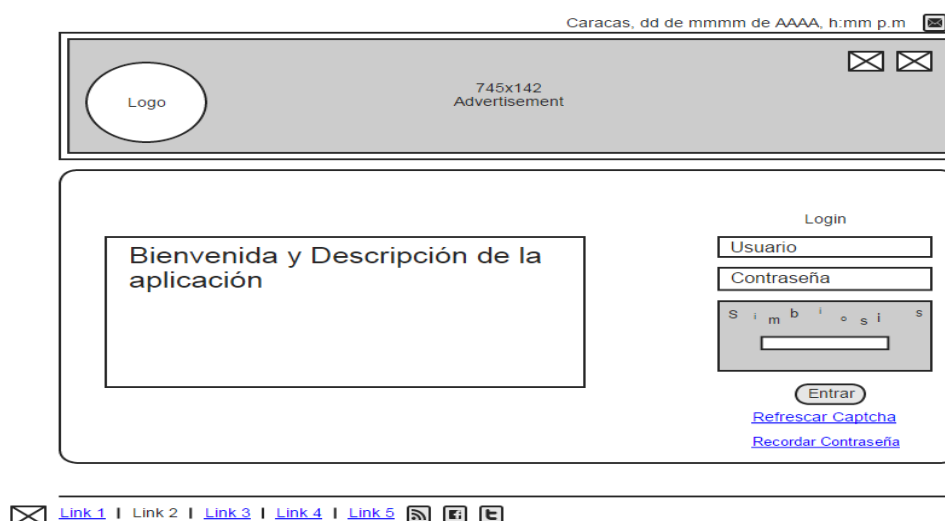
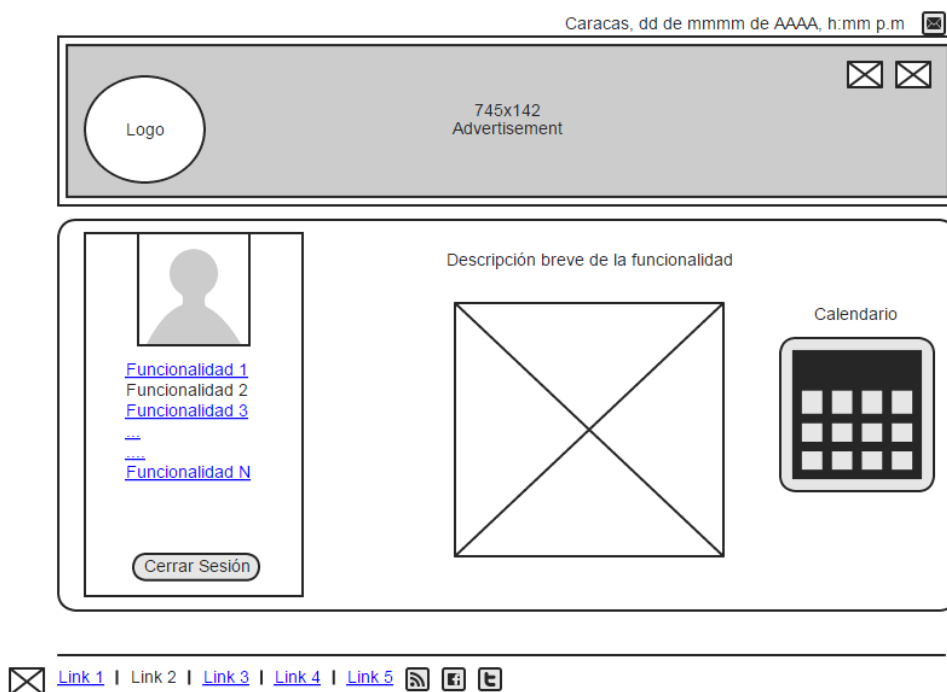


Figura 3.11: Vista general de inicio de sesión para todos los usuarios de la aplicación.

Elaboración propia.

En la figura 3.12 se puede observar el prototipo de interfaz del menú de inicio para todos los usuarios, la cual consta del mismo banner y pie de página que la anterior. La interfaz cuenta con un panel izquierdo y uno central. En el panel izquierdo se encuentran las opciones las distintas funcionalidades dependiendo del perfil. El panel central muestra la opción seleccionada, la descripción de dicha funcionalidad y un calendario.








**Figura 3.12: Vista general del Menú de Usuario. Elaboración propia.**

Una vez mostrado el prototipo de las interfaces, se procede a presentar las guías de estilos por las cuales se registrará la aplicación, para mantener la consistencia a lo largo de todas las interfaces de la aplicación.






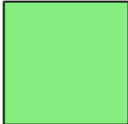




### 3.2.2 Guía de Estilo

Como una actividad contemplada en AgilUs se propone la guía de estilos, que es un documento en el cual se toman las características relacionadas con el aspecto de la interfaz de usuario. El aspecto de una aplicación viene dado por los colores utilizados, la disposición de los elementos en la interfaz, los píxeles utilizados, entre otros, con la finalidad de generar un documento que pueda dar una idea de la apariencia de la aplicación. A continuación se muestra la utilizada en la aplicación web (ver Tabla 3.1)

Tabla 3.1.: Guía de estilo de la aplicación. Parte A

Logos		
		
Logo de la aplicación web Simbiosis	Logo de la Universidad Central de Venezuela	Logo de la Facultad de Ciencias
Navegación		Iconos de la aplicación
		

**Tabla 3.1.: Guía de estilo de la aplicación. Parte B**

<b>Colores Principales</b>				
				
<b>RGB(0,139,222)</b>	<b>RGB(255,229,232)</b>	<b>RGB(240,241,255)</b>	<b>RGB(66,139,202)</b>	<b>RGB(217,83,79)</b>
<b>#0088DE</b>	<b>#FFEFES</b>	<b>#F0F1FF</b>	<b>#428BCA</b>	<b>#D9534F</b>
				
<b>RGB(134,237,128)</b>	<b>RGB(249,249,249)</b>	<b>RGB(227,240,249)</b>	<b>RGB(255,68,68)</b>	<b>RGB(136,136,136)</b>
<b>#86ED80</b>	<b>#F9F9F9</b>	<b>#E3F0F9</b>	<b>#FF4444</b>	<b>#888888</b>
<b>Tipografía</b>	<b>Verdana 13 px - Estilo: Normal</b>			

Las imágenes que serán utilizadas en todas las interfaces son las del logo de la Universidad Central de Venezuela, el de la Facultad de Ciencias de cada lado de la cabecera y el logo de la aplicación Simbiosis.

El diseño del logo de la aplicación Simbiosis, se realizó pensando en el trabajo en conjunto realizado entre la Escuela de Biología y la Escuela de Computación para llevar a cabo este sistema. Consta de una célula vegetal con engranajes adentro, fusionada con un engranaje grande cuyo centro posee el planeta tierra, que a su vez tiene una planta siendo fecundada por una hélice de ADN. Todos estos elementos forman una alegoría al trabajo en sinergia entre ambas Escuelas para lograr satisfacer las necesidades del proceso.

La paleta de los colores principales de la interfaz serán colores verde, rojo y azul, los cuales se mantendrán en toda la aplicación. Se utilizaron combinaciones de gris y azules claros para dar sencillez y elegancia a la página. Los botones son siempre azules para no anular el contraste de azul-gris seleccionado para distintas partes de la aplicación.

La tipografía seleccionada para la aplicación fue Verdana y para los títulos Arial pero en un mayor tamaño. Estos tipos de letras son del tipo sans serif, fueron utilizados ya que son letras básicas sin muchos detalles para no cansar la vista en ningún momento y altamente recomendadas para aplicaciones web. Al navegar a través de las diferentes opciones del portal, la opción seleccionada se sombrea de azul para facilitar la ubicación de las opciones como se muestra en la figura anterior.

Esta guía permite tener una idea más clara sobre los elementos a utilizar, los colores y la disposición de estos en toda la navegación de la aplicación. Se quiso crear una interfaz que fuese lo más simple posible pero que cumpliera con las funcionalidades descritas anteriormente en los requerimientos, y para esto, se evaluó la posición de los elementos y el tamaño que cada uno debía tener, esto con el fin de no sobrecargar la interfaz y que fuera fácil de utilizar.

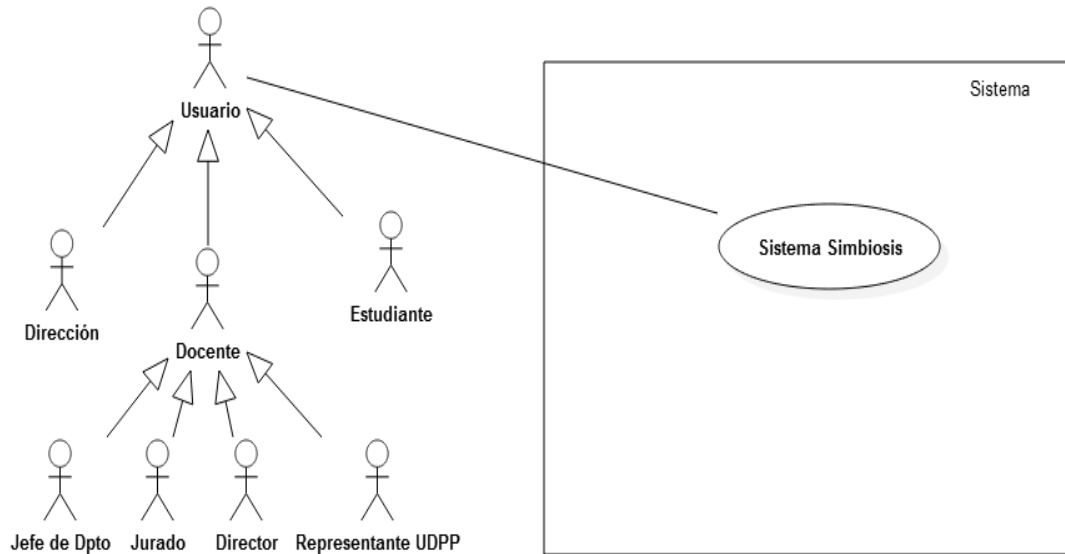
A continuación se describirán los casos de uso que permitirán ahondar más en las funcionalidades de la aplicación.

### **3.2.3 Modelado de los casos de Uso**

A través de los casos de uso se presentan los requerimientos funcionales de la aplicación. A continuación se muestra por cada proceso, una breve descripción de las funcionalidades, y a través de un análisis con diagramas de Casos de Uso, la solución hallada, así como algunas consideraciones a tomar en cuenta, que en muchos casos, fueron aportadas por el personal involucrado en cada uno de los procesos.

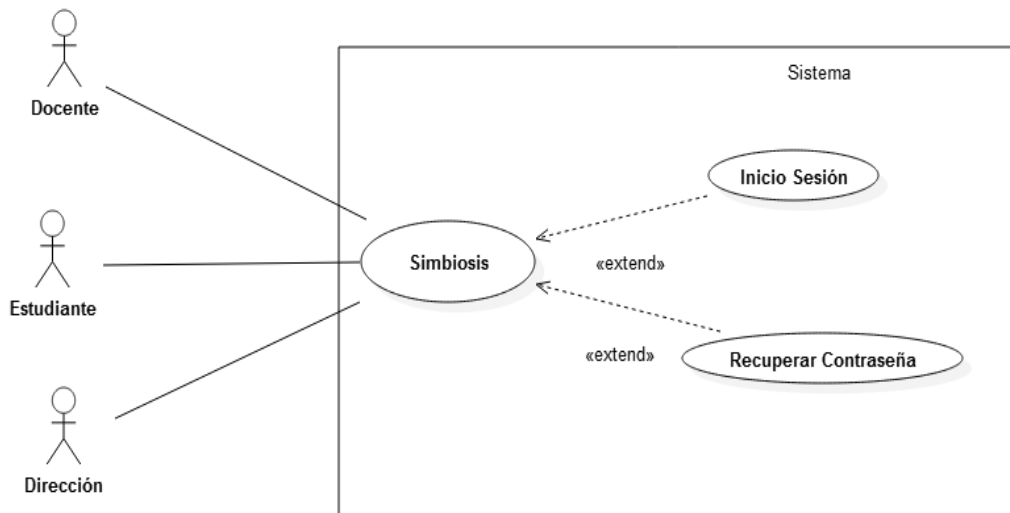
En la figura 3.13, se puede observar el caso de uso correspondiente al nivel 0 de abstracción, en este se modela el sistema a nivel general, con sus respectivos actores. La aplicación hará la gestión de la inscripción de Seminarios y TEG de los estudiantes de la Escuela de Biología.





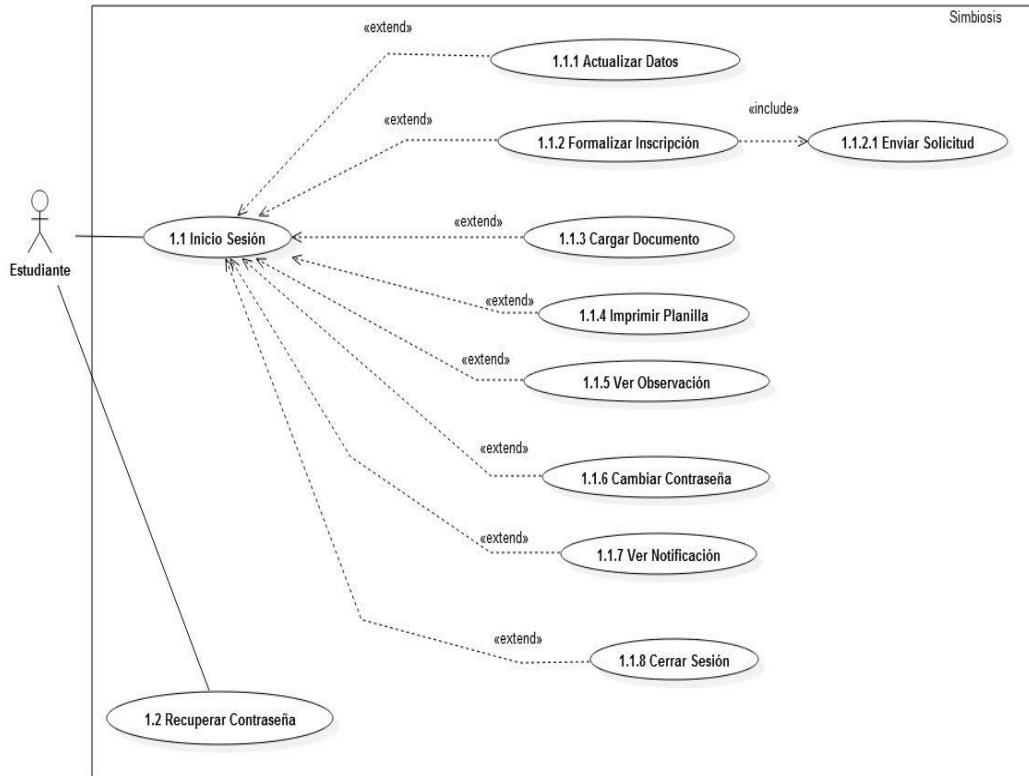
**Figura 3.13: Caso de uso de la aplicación Simbiosis. Nivel 0. Elaboración propia.**

En la figura 3.14 se muestran los casos de uso correspondientes al nivel 1 de abstracción, los cuales presentan las principales funcionalidades de la aplicación y su interacción con los actores.



**Figura 3.14: Caso de uso de la aplicación Simbiosis. Nivel 1. Elaboración propia.**

En la figura 3.15, se muestra el caso de uso para el actor estudiantes y sus principales funcionalidades en la aplicación.



**Figura 3.15: Caso de uso de la aplicación Simbiosis. Actor Estudiante. Elaboración propia.**

A continuación se describen los principales actores de la aplicación:

- Estudiante: realizará la formalización de la inscripción en la aplicación, llevando a cabo los pasos necesarios para completar la misma (completar datos, escoger el tutor, cargar el documento).
- Dirección: se encargara de actualizar las decisiones tomadas por el Consejo de Escuela. Asimismo, llevará a cabo tareas administrativas relacionadas con el proceso de formalización como: cargar listados, iniciar o detener proceso, enviar comunicados.
- Docente: se encargarán de revisar todas las solicitudes de la aplicación y aceptarlas y rechazarlas según sea el caso. Es importante indicar que el actor Docente, puede cumplir varios roles durante el proceso de formalización (Tutor, Jefe de Departamento, Jurado, Director), los cuales serán definidos más adelante mediante otros casos de uso.

Una vez presentado el diagrama de caso de uso del actor estudiante, se presenta su especificación, que se refiere a la descripción de cada una de las partes definidas para lograr la descripción completa de la aplicación. Las descripciones de los casos de uso representan todas

las posibles interacciones de los actores con la aplicación en los eventos enviados o recibidos por estos. En el Anexo A se pueden observar todos los Casos de Uso elaborados que describen las funcionalidades de la aplicación Simbiosis.

**Tabla 3.3: Especificación del caso de uso: Iniciar Sesión.**

Caso de uso: CU.1.1 inicio de sesión.
Actor: todos los actores.
Precondición: el usuario debe existir en la aplicación.
Acción: funcionalidad que permite al usuario acceder a la aplicación, mediante un usuario y una contraseña.
Condición de Éxito: se enviará al usuario a una página con opciones, de acuerdo al tipo de usuario y rol desempeñado.

**Tabla 3.4: Especificación del caso de uso: Recuperar Contraseña.**

Caso de uso: CU. 1.2 Recuperar Contraseña
Actor: todos los actores.
Precondición: el usuario debe existir en la aplicación.
Acción: funcionalidad que permite al usuario obtener una nueva contraseña para ingresar al sistema en caso de que haya sido olvidada.
Condición de Éxito: se enviará al usuario un correo electrónico indicándole sus credenciales de acceso al sistema y la nueva contraseña creada.

Una vez presentados los casos de uso de la aplicación se detalla el diagrama de objetos de dominio, el cual describe los objetos que existen en el contexto de la aplicación. En éste se representan los objetos y relaciones que existen entre estos con el objetivo de lograr una mayor comprensión tanto del contexto como de los objetos que existen en la aplicación. A continuación se presenta el diagrama de objetos de dominio.

### 3.2.4 Diagrama de Objetos del Dominio

El diagrama de objetos del dominio representa los objetos que existen en el contexto de la aplicación y la manera en la que interactúan entre ellos. En dicho diagrama se representan los objetos y relaciones que existen entre ellos, con la finalidad de alcanzar una mejor comprensión tanto del contexto como de los objetos que existen en la aplicación.

En la figura 3.21, se tiene como dominio la aplicación web de la aplicación simbiosis, el cual posee distintos módulos. Existen varios tipos de usuarios que interactúan con la aplicación: Profesor, Dirección y Estudiante. A su vez, el usuario Profesor puede ejercer varios roles durante el proceso (Tutor, Jurado, Representante de la UDPP, Director y Jefe de Departamento).

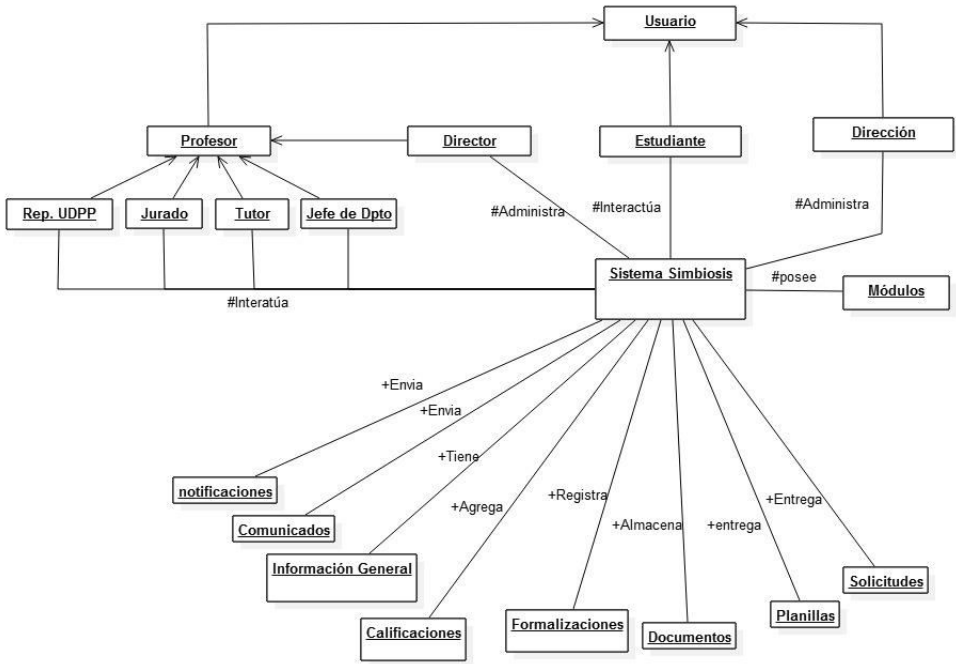


Figura 3.16: Modelo de Objetos del dominio. Elaboración propia.



La Base de Datos contiene 15 tablas, las cuales serán explicadas a continuación:

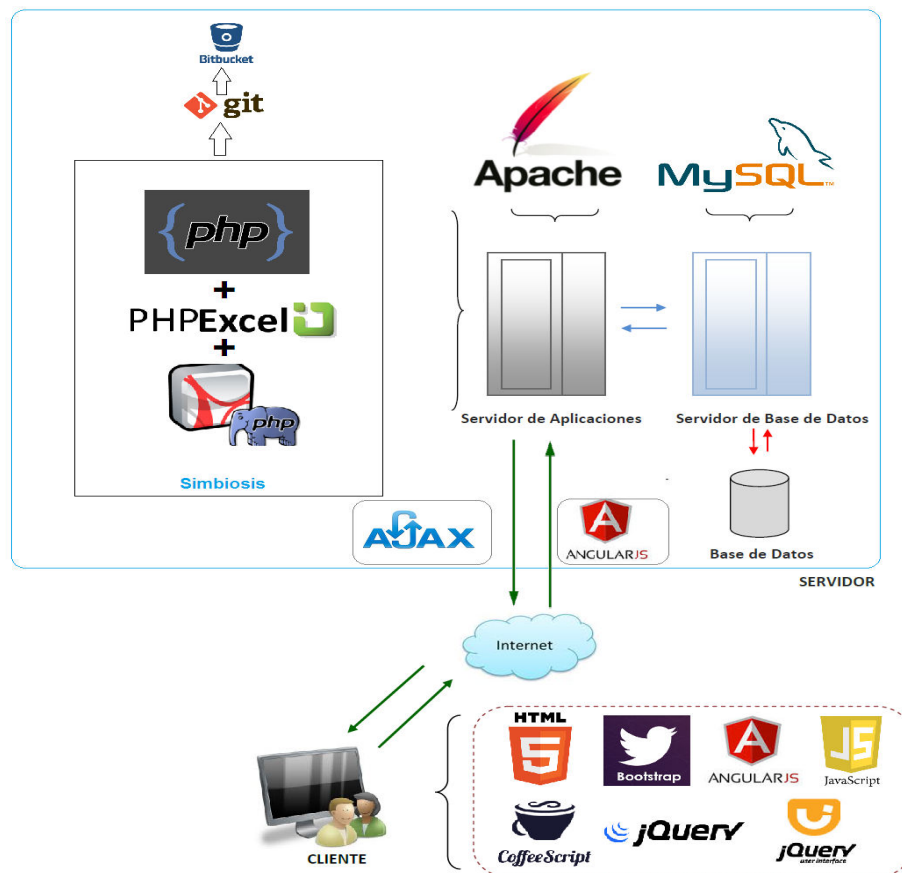
- Usuario: en esta tabla se almacena la información de los usuarios registrados y que tienen acceso al sistema, así como sus características principales.
- Estudiante: almacena todos los estudiantes inscritos en Seminario I y II y Trabajo Especial de Grado de la Escuela de Biología.
- Profesor: contiene información de todos los profesores de la Escuela de Biología (internos y externos).
- Inscribe: esta tabla almacena información acerca de la inscripción de un estudiante, permite establecer la relación entre el estudiante y el documento inscrito.
- Documento\_inscrito: almacenará los datos ingresados por el estudiante relacionados al documento y además la ruta donde se ubicarán en el servidor.
- Tipo\_documento: guarda el tipo de documento inscrito: Seminarios o TEG.
- Departamento: almacena el tipo de departamento, son cinco (5) en total: botánica, zoología, biología celular, tecnología de alimentos, ecología.
- Notificaciones: en esta tabla se almacenan las notificaciones de la aplicación que son enviadas a los usuarios por cada evento que ocurre en el mismo.
- Observaciones: contiene información de cada observación referente al proceso de formalización de un estudiante.
- Tabla\_Roles: tiene los roles que puede ejercer un profesor durante el proceso de formalización en un semestre.
- Proceso: esta tabla almacena información acerca del estado actual del proceso de formalización, el cual puede estar detenido, iniciado o finalizado.
- Histórico de Notas: almacena la calificación obtenida por un estudiante luego de haber sido evaluado durante el semestre en las asignaturas Seminario I y II y TEG.
- Intentos\_login: registra el histórico de intentos fallidos de login, cuando se sobrepasa un valor establecido, automáticamente se prohíbe el acceso al usuario al sistema.

### **3.2.6 Plataforma de Desarrollo**

Las funcionalidades para la aplicación web Simbiosis fueron desarrolladas bajo un enfoque web cliente-servidor, y bajo la plataforma de la aplicación Operativo Linux Ubuntu. El sistema manejador de Bases de Datos usado fue Mysql, y se utilizó la aplicación de control

de versiones Git usando alojamiento del repositorio en Bitbucket. El lenguaje de desarrollo de la aplicación fue PHP.

En cuanto a la programación se utilizó: Html5, Bootstrap, CoffeeScript, JavaScript y AngularJS, además de algunas bibliotecas de desarrollo como jQuery y jQuery UI. Asimismo se hizo uso de la librería fpdf para la generación de documentos en PDF y PHPEXcel para leer y gestionar documentos en formato Excel. En la 3.18 se puede observar el conjunto de herramientas y tecnologías usadas en el proyecto.



**Figura 3.18: Tecnologías empleadas. Elaboración propia.**

Una vez descritas todas las tecnologías utilizadas en la aplicación, se sigue con el método AgilUs, presentando la evaluación heurística. Para la realización de las heurísticas fueron necesarios los artefactos realizados anteriormente.

### 3.3 Fase III: Prototipaje

#### 3.3.1 Desarrollo

Para el desarrollo de la aplicación Simbiosis se usó PHP como lenguaje de procesamiento del lado del servidor. La razón de esta elección se debe a que además de ofrecer muchas funcionalidades ideales para obtener información de Base de datos, posee una sintaxis que sencilla sigue para la creación de objetos y relaciones, el manejo de las rutas y generación de templates es muy intuitivo. En la Figura 3.19 se muestra el script que realiza la conexión a la BD MySQL de la aplicación.

```
1 <?php
2     $db_host = "localhost";
3     $db_user = "root";
4     $db_pass = ".acc,simbiosisucv!";
5     $db_name = "simbiosis";
6     mysqli_report(MYSQLI_REPORT_STRICT);
7     try {
8         $mysqli = new mysqli($db_host, $db_user, $db_pass, $db_name);
9         $mysqli->query("SET NAMES utf8");
10        $mysqli->query("SET CHARACTER SET utf8");
11    } catch (Exception $e) {
12        echo "Service unavailable";
13        echo "message: " . $e->message;
14    }
15    exit;
16    }
17    try {
18        $mysqli2 = new mysqli($db_host, $db_user, $db_pass, $db_name);
19        $mysqli2->query("SET NAMES utf8");
20        $mysqli2->query("SET CHARACTER SET utf8");
21    } catch (Exception $e) {
22        echo "Service unavailable";
23        echo "message: " . $e->message;
24    }
25    exit;
26    }
```

**Figura 3.19: Conexión a la BD de Simbiosis. Elaboración propia.**

La asignación de los roles de los usuarios se realiza mediante un script que vacía la tabla de roles y luego inserta los roles según el tipo de usuario. En la figura 3.20 se muestra el script que se encarga de crear llenar la tabla con los roles existentes en el sistema.



```

1 <?php
2 class RoleTableSeeder extends Seeder {
3
4     public function run()
5     {
6         DB::table('roles')->delete();
7         DB::table('roles')->insert(array(
8             array('name' => 'Secretaria', 'shortname' => 'admin'),
9             array('name' => 'Director', 'shortname' => 'director'),
10            array('name' => 'Profesor', 'shortname' => 'profesor'),
11            array('name' => 'Estudiante', 'shortname' => 'estudiante'),
12        ));
13    }
14
15 }

```

**Figura 3.20: Creación de los roles en la aplicación.**

Los usuarios de la aplicación se pueden crear de forma manual o mediante la carga de un listado en formato Excel con los usuarios y su información. En la figura 3.21 se puede apreciar el código que se encarga de crear los usuarios de forma manual.

```

1 <?php
2 session_start();
3 $uri=parse_url($_SERVER['REQUEST_URI'],PHP_URL_PATH);
4 $uri=urldecode($uri);
5 $path= require __DIR__.'/simbiosis/bootstrap/paths.php';
6 $requested= $paths['public'].$uri;
7
8 if ($uri!=='/' and file_exists($requested)){
9     return false;
10 }
11
12 require_once $paths['public'].'/index.php';
13
14 include ('acceso_db_2.php');
15 //comprobamos que las variables de sesion esten vacias
16 if(empty($_SESSION['usuario_id'])){
17     $salt="";
18     $null = "NULL";
19     $saltedPW=$cedula.$salt;
20     $hashedPW = hash('sha256', $saltedPW);
21     echo '<form action="create_user.php" method=post>
22         <label>Cedula:
23         <input type="text" name="usuario_id" /> <br />
24         <label>Nombre:
25         <input type="text" name="usuario_name" /> <br />
26         <input type="hidden" value="$hashedPW" name="pass" /> <br />
27         <input type="submit" name="enviar" value="Agregar"/> <br />
28         input
29     </form>'
30 }else{
31 ?>

```

**Figura 3.21: Código para la Creación de usuarios de forma manual.**

Una vez creados los roles se hizo el script encargado de iniciar la sesión del usuario en la aplicación y redirigirlo al menú correspondiente dependiendo del rol que tenga asignado. En las figuras 3.22 y 3.23 se muestran el código del script para iniciar sesión en la aplicación.

```

1 <?php
2 session_start();
3 if(empty($_SESSION['captcha'] ) ||
4     strtolower(trim($_POST['captcha'])) != 0 ) {
5     header ("Location: ../../views/error/error.php?login=captcha");
6 }else{ //else captcha
7     if($_SERVER["REQUEST_METHOD"] == "POST"){ //solicitud post
8         if (isset($_POST["password"]) &&
9             isset($_POST["cedula"])){ //si se envió el usuario y la cedula
10            $username=$_POST['cedula'];
11            $password=$_POST['password'];
12            include("../DataBase/acceso_db_2.php");
13            $saltQuery = "SELECT * FROM usuario WHERE usuario= '$username'";
14            $result = $mysqli->query($saltQuery) or die($mysqli->error.__LINE__);
15            $row = $result->fetch_assoc();
16            $salt = $row['salt'];
17            $usuario=$row['usuario'];
18            $privilegios=$row['privilegios'];
19            $saltedPassword = $password.$salt;
20            $hashedPassword = hash('sha256', $saltedPassword);
21
22            $query_s= "SELECT * FROM usuario WHERE (usuario ='. $username .')
23            AND (contrasena ='. $hashedPassword.')" ;
24            $login = $mysqli2->query($query_s) or die($mysqli2->error.__LINE__);
25            $row_cnt=$login->num_rows;
26            if ($row_cnt>0) {
27                session_start();
28                $row= $login->fetch_array(MYSQLI_ASSOC);
29                $cedula= $row['usuario'];
30                $cedula= strtolower($cedula);
31                $clave=$row['contrasena'];
32                $privilegios=$row['privilegios'];
33                $salt=$row['salt'];
34                $saltedCedula = $cedula.$salt;
35                $hashedCedula=hash('sha256', $saltedCedula);
36                if(strcmp($hashedCedula,$clave)==0){
37                    $_SESSION["autenticado"]="SI";
38                    $_SESSION["usuario"] = $cedula;
39                    $_SESSION["perfil"]= $privilegios;
40                    $_SESSION["contraseña"]=$clave;
41                    $_SESSION["salt"]=$salt;
42                    header ("Location: ../../views/$privilegios/index.php");
43                }
44            }
45        }
46    }
47 }
48 ?>

```

**Figura 3.23: Código para el inicio de sesión. Parte B. Elaboración propia**

Para la creación de las vistas que servirán de base para el renderizado final de la página web solicitada, por lo general en formato HTML, se creó un script que permite colocar junto al código HTML código embebido muy parecido a la sintaxis de PHP. De esta manera se logra obtener vistas dinámicas en función de lo solicitado.

Mediante el script de templating se puede lograr herencia entre vistas, hacer includes de porciones de código o vistas definidas en otro archivo, haciendo el código reusable y más fácil de leer, siguiendo la metodología DRY (Don't Repeat Yourself). Para la aplicación, se definió una vista base la cual contiene todo lo que es común en todas las demás vistas (header, menú, footer) y las demás vistas heredan de esta vista base y colocan aquel contenido dinámico que no es igual en las demás. Incluso dado que la aplicación maneja distintos roles, los cuales poseen distintas funcionalidades que eso se traduce en distintos menús en la aplicación, se creó un menú para cada rol siendo este menú también cambiado dinámicamente. En la figura 3.23 se muestra el código del template base para todas las interfaces en la aplicación. Para mantener una estructura y orden, por cada rol en la aplicación, se creó una subcarpeta para cada uno, donde se definen sus vistas según sea el caso.

```

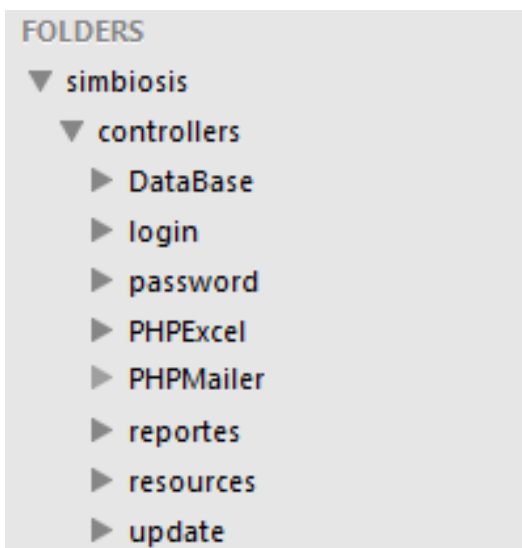
1 <!DOCTYPE html>
2 <html lang="es">
3 <!--[if gt IE 8]><!--> <html class="no-js"> <!--<![endif]-->
4 <head>
5 <meta http-equiv="content-type" content="text/html"; charset="UTF-8">
6 <link rel="shortcut icon" href="../img/favicon.ico" >
7 <title>Simbiosis::Inicio</title>
8 <meta name="description" content="">
9 <meta name="viewport" content="width=device-width, initial-scale=1.0">
10 <meta name="author" content="Simbiosis-UCV">
11 <link href="{{ asset('assets/styles/normalize.css') }}" rel="stylesheet">
12 <link href="{{ asset('assets/styles/style_fancy.css') }}" rel="stylesheet" media="screen">
13 <link href="{{ asset('assets/styles/main.css') }}" rel="stylesheet">
14 <link href="{{ asset('assets/styles/bootstrap.css') }}" rel="stylesheet" media="all">
15 <script src="{{ asset('assets/scripts/jquery.min.js') }}"></script>
16 <script src="{{ asset('assets/scripts/main.js') }}"></script>
17 </head>
18 <body >
19 <div id="tope">
20 <div id="fecha">
21 <script type="text/javascript">document.write(retornarFecha()+", "+retornarHora());</script>
22 </div>
23 <span id="contactenos"><a href="mailto:simbiosis.ucv@gmail.com" title="Enviar Correo">Contáctenos
24 </a></span>
25 </div>
26 <br/>
27 <div class="container">
28 <div class="container" id="head">
29 
31 
32 
33 
35 </div>
36 <br/>
37 </body>
38 </html>

```

**Figura 3.23: Código para el template base de la aplicación. Elaboración Propia**

Debido a que se utilizó el patrón MVC, para separar la lógica de negocio (el modelo) y la presentación (la vista) en la aplicación, se crearon varios directorios por tipo de controlador y se almacenaron en la carpeta simbiosis/controllers. En la figura 3.24 se muestra la estructura de controladores, se tiene un controlador para los accesos a la base de datos, uno que maneja

las sesiones de usuario y se encuentra en la carpeta login, existe un controlador para manejar las contraseñas y cambios que se realicen. Luego, están los controladores para gestionar los correos y archivos Excel para la carga de datos. Finalmente se tienen los controladores para generar reportes, administrar documentos cargados y para actualizaciones de datos en la aplicación.



**Figura 3.24: Estructura de controladores de la aplicación. Elaboración propia.**

Una vez culminada la fase de desarrollo, se procede a realizar las distintas pruebas a la aplicación desarrollada, las cuales se realizan con la finalidad de comprobar que los sistemas desarrollados cumplen con los requerimientos para los cuales han sido creados.













Estas pruebas son basadas en el análisis de datos de entrada/salida, y usabilidad de la aplicación. En este caso, se diseñaron un conjunto de pruebas para cada una de las funcionalidades implementadas para la aplicación.

### **3.3.2 Lista de comprobación**

Se aplicó esta técnica a diez (10) posibles usuarios del sistema, así como también a tres (3) personas que conforman el personal administrativo de la Escuela de Biología involucrados en el proceso académico de formalización de inscripción. Esto permitió identificar fallas en el cumplimiento del control y libertad para el usuario; y en minimizar la carga de la memoria del usuario. A las personas a las que se les aplicó esta técnica, se les facilitó la interfaz de usuario

de la aplicación para la realización de distintas actividades relacionadas con la automatización del proceso. Fue utilizada la siguiente lista de comprobación:

**Tabla 3.5: Lista de comprobación para la aplicación. Parte A**

Lista de Comprobación	Se cumple	No se cumple
Reconocer en lugar de recordar.		
Usar indicadores visuales.		
Flexibilidad en la interfaz de usuario.		
Proveer Feedback a las acciones del usuario.		
Consistencia en apariencia y uso		
Evitar palabras coloquiales y abreviaturas.		
Mostrarle al usuario lo que necesita cuando lo necesite.		
Permitir al usuario recuperarse de los errores		
Mantener la interfaz simple, sencilla y organizada.		
Hacer sentir al usuario que puede tomar.		
Las funciones del sistema deben estar claramente visibles e identificables.		
Permitir al usuario personalizar la IU.		

### 3.4 Fase IV: Entrega

#### 3.4.1 Pruebas de usabilidad

Durante esta prueba, se evaluó la usabilidad de la aplicación web Simbiosis, considerando los criterios creados por expertos en IHC, que se describen en la Tabla 3.6.

**Tabla 3.7: Descripción de criterios de usabilidad. Parte A**

<b>Criterio</b>	<b>Descripción</b>
<b>Accesibilidad</b>	El sitio es visible en cualquier configuración que tenga el cliente.
<b>Navegabilidad</b>	La información debe ser fácil de encontrar, debe haber consistencia en los elementos de navegación.
<b>Lenguaje manejado</b>	Debe ser fácil de entender, de acuerdo al tipo de usuario al que este dirigido la aplicación.
<b>Conceptos utilizados</b>	Deben ser adecuados al usuario al que está dirigida la aplicación.
<b>Consistencia</b>	En los títulos, cabeceras y etiquetas de links. La familiaridad y el reconocimiento deben permitir a los usuarios asociar ciertos elementos con ciertas funcionalidades.
<b>Accesibilidad</b>	El sitio es visible en cualquier configuración que tenga el cliente.
<b>Navegabilidad</b>	La información debe ser fácil de encontrar, debe existir consistencia en los elementos de navegación.
<b>Lenguaje manejado</b>	Debe ser fácil de entender, de acuerdo al tipo de usuario al que este dirigido la aplicación.
<b>Consistencia</b>	En los títulos, cabeceras y etiquetas de links. La familiaridad y el reconocimiento deben permitir a los usuarios asociar ciertos elementos con ciertas funcionalidades.

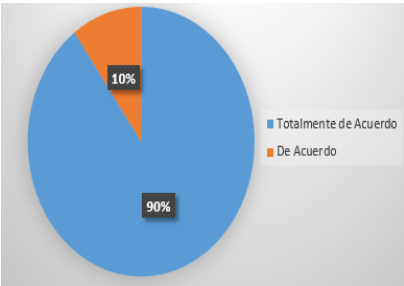
Con la finalidad de poder evaluar los criterios anteriormente detallados, se empleó una encuesta a un grupo de usuarios potenciales de la aplicación, los cuales dieron una puntuación comprendida entre 1 y 5, a cada uno de los criterios de usabilidad. A continuación se muestra la leyenda de números utilizada para identificar cada una de las escalas de valoración:

1. Totalmente de Acuerdo
2. De Acuerdo
3. Ni de acuerdo ni en desacuerdo
4. En desacuerdo
5. Totalmente en desacuerdo

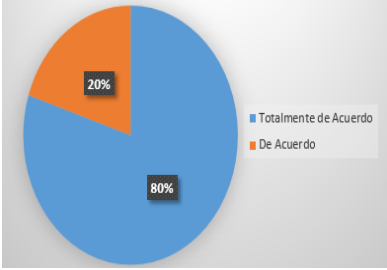
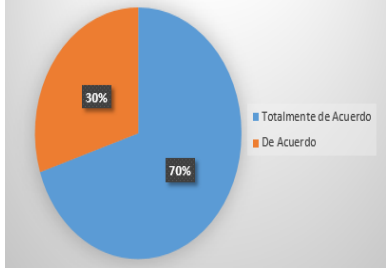
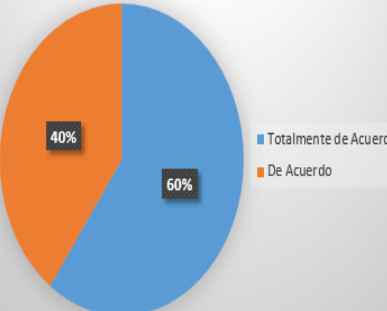
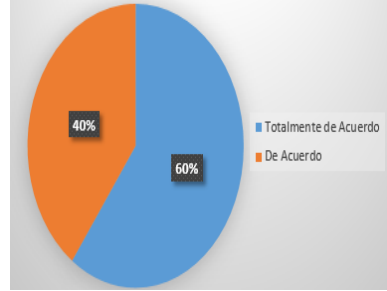
Para el caso de las pruebas de usabilidad se realizaron las pruebas con las personas que serán usuarios potenciales de la aplicación, los cuales fueron dos profesores de la escuela y un grupo de 8 estudiantes. Las pruebas consistieron en observar si a los usuarios se les hizo fácil e intuitivo usar el sistema en cada uno de los roles que deben cumplir.

Seguidamente, los resultados fueron recabados y totalizados, calculando el promedio obtenido por cada uno de los criterios. A continuación se describen los resultados de la encuesta, para la aplicación web, los cuales se muestran con mayor detalle en el anexo B.

**Tabla 3.9: Resultados de las Pruebas de Usabilidad. Parte A**

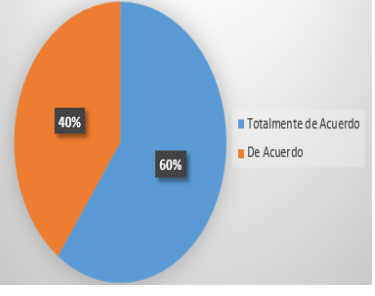
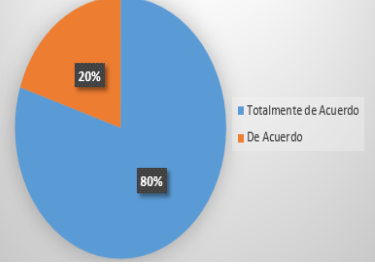
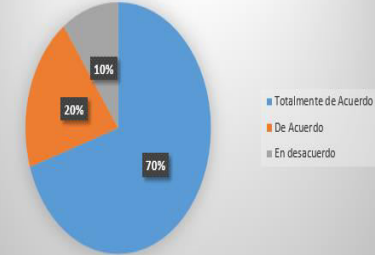
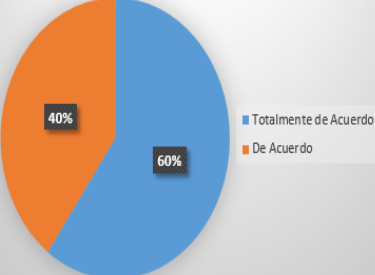
Planteamiento	Resultados Obtenidos	Gráfico
<p><b>Los títulos, menús y textos de la aplicación son legibles.</b></p>	<p>9/10 (90%) de los usuarios estuvo totalmente de acuerdo y 1/10 (10%) estuvo de acuerdo</p>	 <p>A pie chart with two segments. The larger segment is blue and labeled '90%' with a legend entry 'Totalmente de Acuerdo'. The smaller segment is orange and labeled '10%' with a legend entry 'De Acuerdo'.</p>

**Tabla 3.9: Resultados de las Pruebas de Usabilidad. Parte B**

Planteamiento	Resultados Obtenidos	Gráfico
<p><b>Los colores utilizados contribuyen a la percepción adecuada de la aplicación</b></p>	<p>8/10 (80%) de los usuarios estuvo totalmente de acuerdo y 2/10 (20%) estuvo de acuerdo</p>	 <p>A pie chart with a blue segment representing 80% (Totalmente de Acuerdo) and an orange segment representing 20% (De Acuerdo). A legend on the right identifies the segments.</p>
<p><b>Los descriptores de la aplicación son claros y precisos</b></p>	<p>7/10 (70%) de los usuarios estuvo totalmente de acuerdo y 3/10 (30%) estuvo de acuerdo</p>	 <p>A pie chart with a blue segment representing 70% (Totalmente de Acuerdo) and an orange segment representing 30% (De Acuerdo). A legend on the right identifies the segments.</p>
<p><b>Se ofrece una organización que facilita la navegación</b></p>	<p>6/10 (60%) de los usuarios estuvo totalmente de acuerdo y 4/10 (40%) estuvo de acuerdo</p>	 <p>A pie chart with a blue segment representing 60% (Totalmente de Acuerdo) and an orange segment representing 40% (De Acuerdo). A legend on the right identifies the segments.</p>
<p><b>Se ofrece una organización que facilita la navegación</b></p>	<p>6/10 (60%) de los usuarios estuvo totalmente de acuerdo y 4/10 (40%) estuvo de acuerdo</p>	 <p>A pie chart with a blue segment representing 60% (Totalmente de Acuerdo) and an orange segment representing 40% (De Acuerdo). A legend on the right identifies the segments.</p>



**Tabla 3.9: Resultados de las Pruebas de Usabilidad. Parte C**

<p><b>El diseño de la aplicación favorece su uso</b></p>	<p>6/10 (60%) de los usuarios estuvo totalmente de acuerdo y 4/10 (40%) estuvo de acuerdo</p>	 <p>A pie chart with two segments: a blue segment representing 60% labeled 'Totalmente de Acuerdo' and an orange segment representing 40% labeled 'De Acuerdo'.</p>
<p><b>La aplicación se puede ejecutar de manera intuitiva</b></p>	<p>8/10 (80%) de los usuarios estuvo totalmente de acuerdo y 2/10 (20%) estuvo de acuerdo</p>	 <p>A pie chart with two segments: a blue segment representing 80% labeled 'Totalmente de Acuerdo' and an orange segment representing 20% labeled 'De Acuerdo'.</p>
<p><b>Es una aplicación fácil de utilizar</b></p>	<p>7/10 (70%) de los usuarios estuvo totalmente de acuerdo y 2/10 (20%) estuvo de acuerdo y una sola persona (10%) estuvo en desacuerdo</p>	 <p>A pie chart with three segments: a blue segment representing 70% labeled 'Totalmente de Acuerdo', an orange segment representing 20% labeled 'De Acuerdo', and a grey segment representing 10% labeled 'En desacuerdo'.</p>
<p><b>Los elementos de ayuda disponibles en la aplicación son útiles</b></p>	<p>6/10 (60%) de los usuarios estuvo totalmente de acuerdo y 4/10 (40%) estuvo de acuerdo</p>	 <p>A pie chart with two segments: a blue segment representing 60% labeled 'Totalmente de Acuerdo' and an orange segment representing 40% labeled 'De Acuerdo'.</p>

De esta manera, es posible concluir que los resultados de esta evaluación fueron favorables, ya que se estuvo totalmente de acuerdo (1) o de acuerdo (2) con los planteamientos

del cuestionario la mayoría de los criterios tomados en cuenta, lo que indica que las herramientas resultan intuitivas y fáciles de usar para los usuarios potenciales.

### 3.4.2 Pruebas Cross Browser

Estas pruebas se realizaron con el propósito de evaluar el comportamiento de la aplicación web Simbiosis, cuando son ejecutadas en distintos navegadores, esperando obtener la misma funcionalidad y apariencia. En la tabla 3.10 se puede observar los navegadores utilizados durante la evaluación y la descripción del comportamiento que presentó la aplicación web Simbiosis al ser ejecutados sobre ellos.

**Tabla 3.11. Descripción de pruebas Cross Browser para Simbiosis.**

Navegador	Comportamiento
Mozilla Firefox 37.0.1	En este navegador la aplicación funciona correctamente, y la apariencia es adecuada con respecto al diseño realizado.
Internet Explorer 11	En este navegador las aplicaciones funcionan correctamente. Sin embargo, al posicionar el mouse sobre los botones estos no cambian de color para indicar que se está ejecutando alguna acción sobre los mismos. Además algunos íconos cambian de posición.
Google Chrome 42.0	En este navegador las aplicaciones funcionan correctamente, y la apariencia es adecuada de acuerdo al diseño preestablecido.

Los resultados de este conjunto de pruebas fueron favorables, ya que en los navegadores utilizados las diferencias de apariencia fueron mínimas, y el comportamiento de las aplicaciones fue el esperado. De igual manera se realizó la prueba Cross Browser para los navegadores mencionados anteriormente en su versión para dispositivos móviles, obteniendo los mismos resultados.

Finalmente, en la figura 3.25 se observa un ejemplo de la vista de la aplicación Simbiosis, en el navegador Google Chrome.

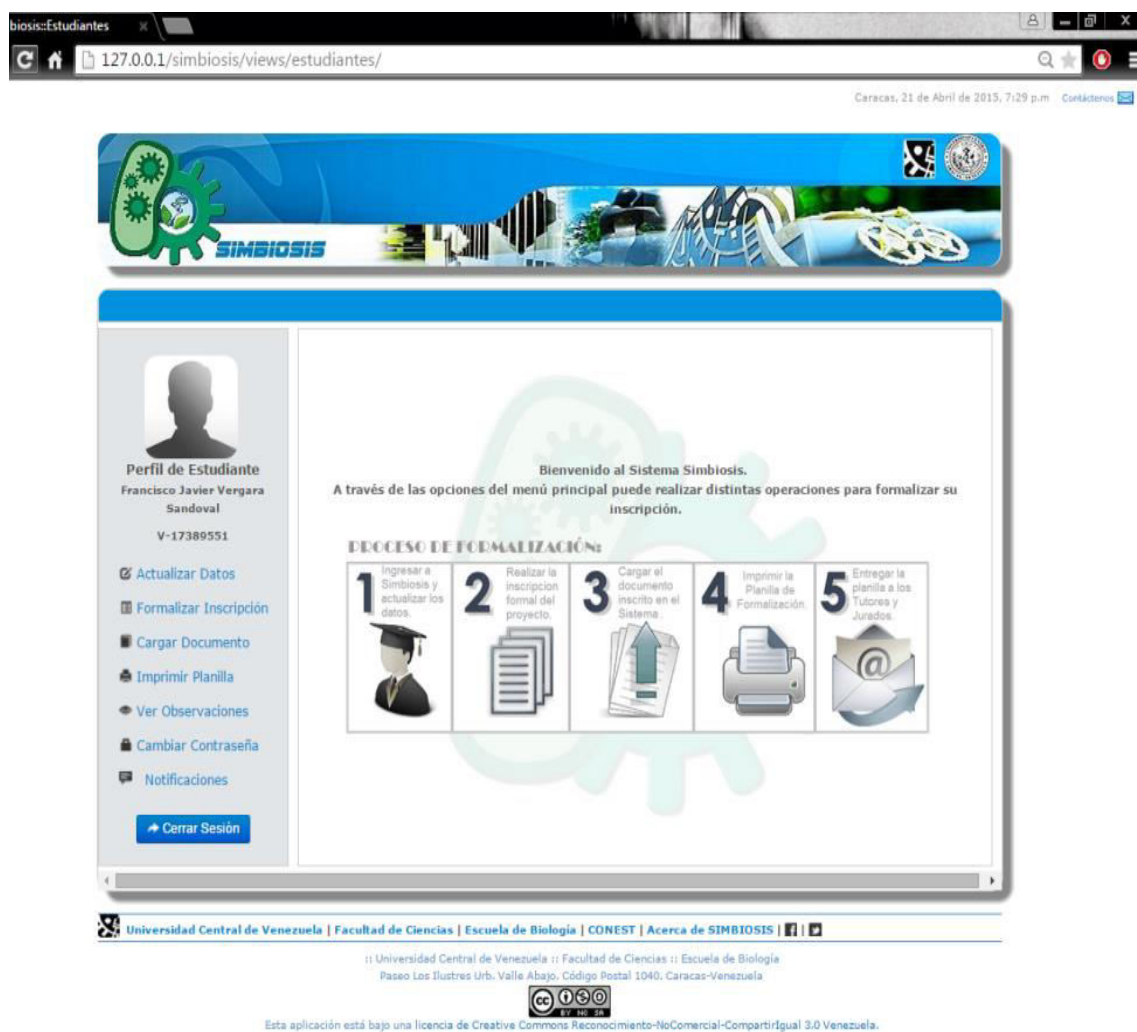


Figura 3.25: Vista de la aplicación Simbiosis en Google Chrome.

### **3.4.3 Pruebas de Instalación**

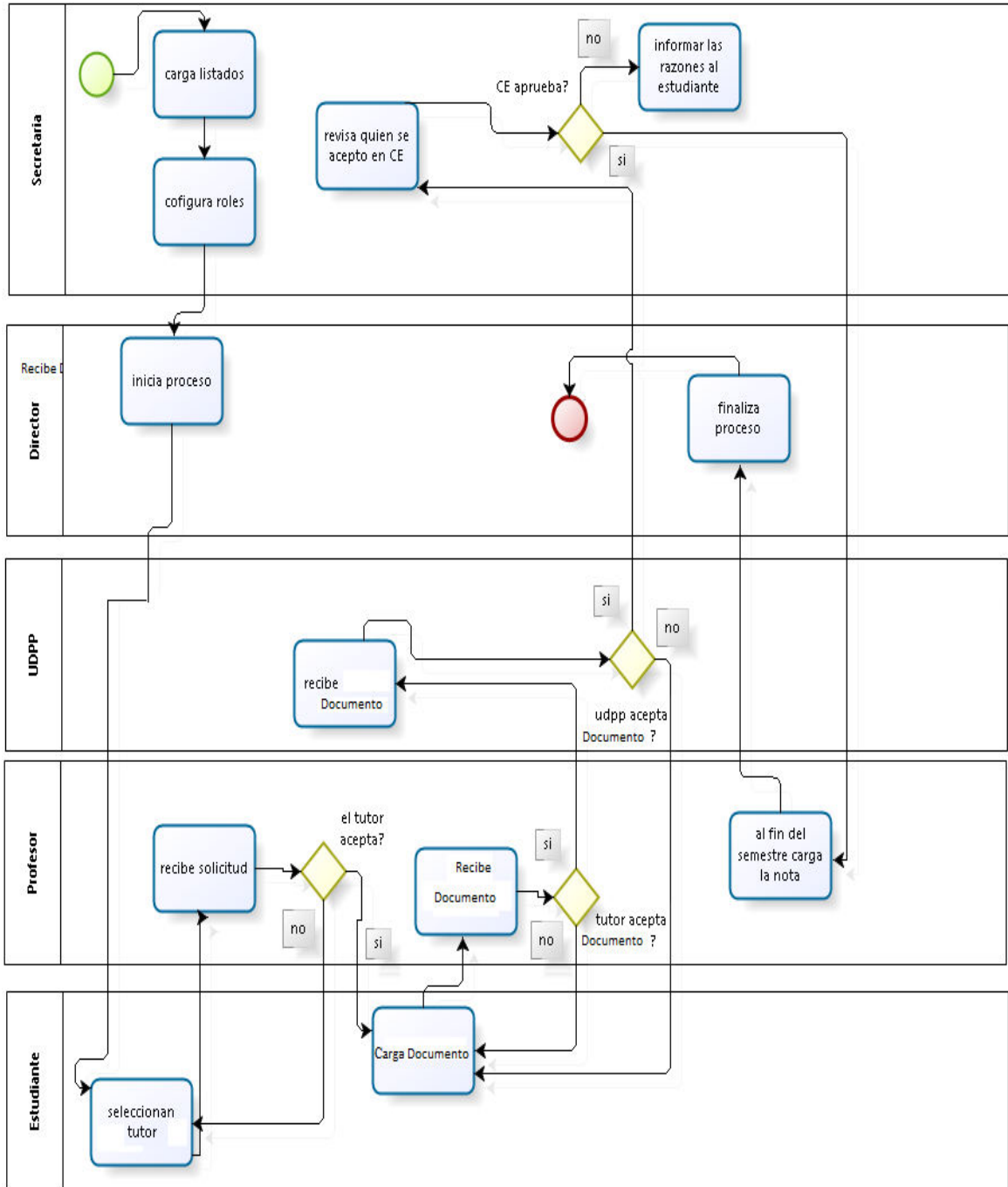
Las pruebas de instalación de la aplicación se realizaron directamente en el ambiente de producción, el cual es un servidor perteneciente a la Facultad de Ciencias de la UCV.

El acceso al servidor se realizó utilizando el protocolo SFTP (File Transfer Protocol) utilizando la aplicación Filezilla versión 3.10.3. Luego de establecida la conexión se procede a almacenar la carpeta que contiene los componentes de la aplicación en la carpeta raíz del servidor de prueba. Adicionalmente para la edición y configuración de algunos archivos en el servidor se usó el protocolo SSH (Secure Shell), el cual permite a un usuario ejecutar comandos por la consola de una máquina sin tener que estar físicamente presente cerca de ella. Para ello se utilizó la aplicación Putty versión 5.6

Para la creación de las tablas necesarias para el modelo de la base de datos, se utilizó la aplicación Phpmyadmin versión 4.2.9, la cual funciona como un cliente de MySQL y permite importar scripts al servidor directamente mediante un navegador web. Finalmente, los resultados fueron exitosos, debido a que la aplicación funcionó correctamente, mientras que los accesos y modificaciones a la base de datos se realizaron de manera satisfactoria.

### **3.4.4 Flujo de Trabajo de Proceso de Automatizado**

Debido a la complejidad del proceso de inscripción automatizado y la cantidad de toma de decisiones que se encuentran dentro del mismo se decidió elaborar un flujo de trabajo a través del cual se explique la forma en que será llevado el mismo, con los roles participantes y los puntos de control respectivos. El mismo se muestra a continuación en la Figura 3.31.



**Figura 3.26: Flujo de Trabajo - Proceso de Inscripción Automatizado. Elaboración Propia**

Una vez mostrado parte del código de la aplicación y la estructura del mismo, así como las pruebas que se realizaron en la metodología AgilUs, en el siguiente capítulo muestran los resultados finales obtenidos en la aplicación.

## Capítulo 4

### Resultados

En el presente capítulo son descritos los resultados en función de la interfaz y las funcionalidades de la aplicación web Simbiosis luego de aplicar la metodología de desarrollo AgilUs. La aplicación dirigida a todo el personal docente, estudiantil y administrativo involucrado en el proceso de inscripción de Seminarios y TEG.

La aplicación consta de siete (7) módulos que son: Estudiante, Dirección y Docente, además el perfil docente puede tener los siguientes roles: Tutor, UDPP, Jefe de Departamento, Director y asesor. Todos los módulos trabajando en conjunto gestionan el proceso de Inscripción de Seminarios y TEG.

#### 4.1 Página Principal

Presenta información referente al proceso a la aplicación web Simbiosis, así como enlaces de interés. Permite el ingreso a la aplicación, tal y como se muestra a continuación en la Figura 4.1.



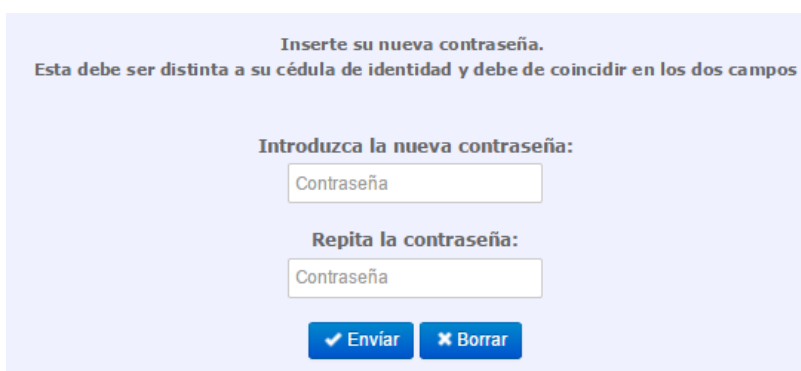
Figura 4.1: Página principal de la aplicación Simbiosis.

La primera funcionalidad que tienen todos los usuarios en general, es iniciar sesión la aplicación, requisito indispensable para utilizar las funcionalidades de la misma (figura 4.2). El usuario debe iniciar sesión ingresando su cédula como usuario y clave, y completar un captcha para poder ingresar, inmediatamente se le solicitará cambiar la contraseña por una distinta a su cédula de identidad (figura 4.3).



Formulario de inicio de sesión con el título "Inicio de Sesión". Incluye un campo de usuario con el valor "2619455", un campo de contraseña con caracteres ocultos por puntos, un campo de captcha con el texto "UIKEAI", un botón "Iniciar Sesión" y un enlace "¿Olvidó su Contraseña? Si olvidó su clave use este enlace.".

**Figura 4.2: Formulario para el inicio de sesión.**



Formulario para cambiar de contraseña con el título "Inserte su nueva contraseña. Esta debe ser distinta a su cédula de identidad y debe de coincidir en los dos campos". Incluye dos campos de contraseña, un botón "Enviar" y un botón "Borrar".

**Figura 4.3: Formulario para cambiar de contraseña.**

Adicionalmente, como se muestra en la figura 4.4, se cuenta con una función de recuperar contraseña, en caso de que el usuario haya olvidado la misma se le enviará un correo electrónico con una nueva contraseña para que ingrese a la aplicación.

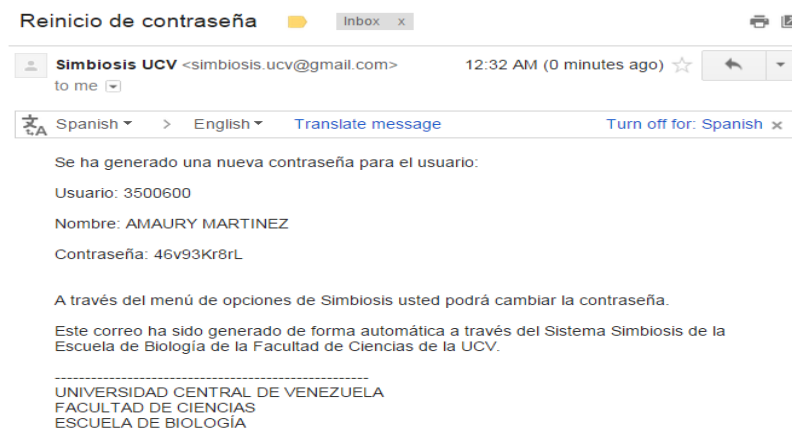
Recuperación de Credenciales

Indique los siguientes datos:

Cédula:

**Figura 4.4: Funcionalidad para recuperar contraseña.**

Al enviar la petición de recuperación de contraseña se envía un correo electrónico, con la información de acceso al sistema, como se muestra en la figura 4.5.



**Figura 4.5: Correo de recuperación de contraseña.**

## 4.2 Menú de Funcionalidades

Luego de iniciar sesión, el usuario tiene un panel de opciones ubicado en la parte lateral izquierda, que posee distintas funcionalidades dependiendo de los privilegios y roles del usuario. En las figuras 4.6, 4.7 y 4.8, se muestra el panel de opciones para Estudiantes, Profesores y para el usuario Dirección.





Figura 4.6: Opciones para el usuario Estudiante.



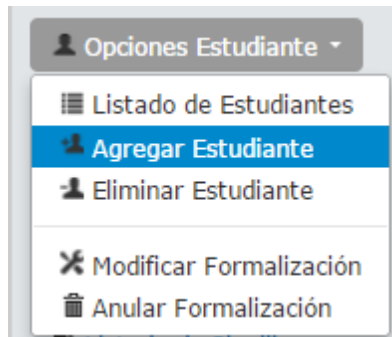
Figura 4.7: Opciones para el usuario Dirección.



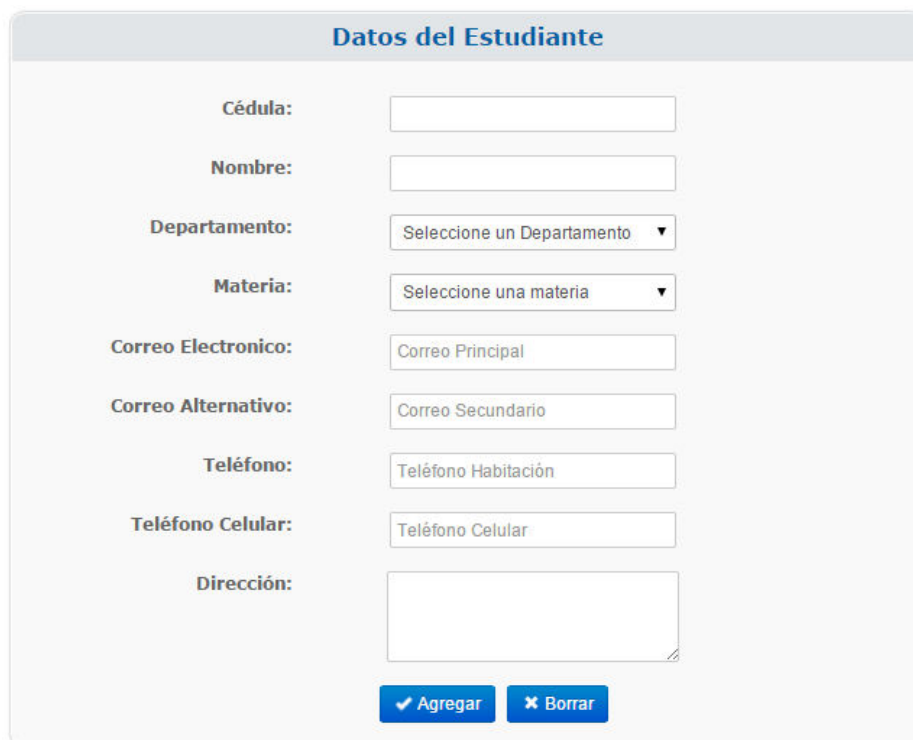
**Figura 4.8: Opciones para el usuario Profesor.**

### 4.3 Gestión de Usuarios

El usuario docente con el rol de Director y el usuario Dirección pueden agregar, eliminar y modificar usuarios en la aplicación. Los mismos, podrán ser cargados mediante un archivo o manualmente. Para agregar usuarios se debe hacer clic en la opción mostrada en la figura 4.8 y seguidamente se despliega un el formulario mostrado en la figura 4.9 para agregar al estudiante.



**Figura 4.8: Opción de Agregar Estudiante.**



**Datos del Estudiante**

Cédula:

Nombre:

Departamento:

Materia:

Correo Electronico:

Correo Alternativo:

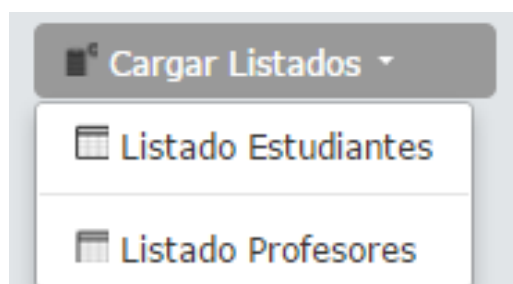
Teléfono:

Teléfono Celular:

Dirección:

**Figura 4.9: Formulario para Agregar Estudiante.**

Si se quiere agregar usuarios de manera masiva, mediante un archivo o lista en formato Excel, se debe hacer clic en la opción mostrada en la figura 4.10 y seguidamente se despliega un formulario para seleccionar el archivo a cargar (figura 4.11).



**Figura 4.10: Opción para Cargar Listados.**

### Cargar Listado de Estudiantes

Departamento:

Materia:

**Seleccione el documento a cargar:**  
(Haga click en la imagen para buscar el listado a cargar)



El nombre del archivo cargado es:

Advertencia: Verifique que el listado sea el correcto antes de Cargarlo

**Figura 4.11: Formulario para Cargar Listado de Estudiante.**

Al agregar los estudiantes o docentes se podrán visualizar en la opción Listado de Docentes o Listados de Estudiantes del panel de opciones, separadas por licenciatura y con opciones para ordenarlas o buscar una en específico mostrado en la figura 4.12. Es importante destacar que en esta misma vista se puede agregar, editar o eliminar un Estudiante o Docente.

Departamentos:

#	Nombre	Cédula	Departamento	Acciones
1	Alejandro J Pieters	5299518	Docente Externo	
2	Alexander Ivan Laurentin Torres	9565764	Biología Celular	
3	Alicia Josefina Caceres Sanchez	4361203	Botánica	
4	Alonso David Ojeda	6364695	Ecología	
5	Amaury Martinez	3500600	Tecnología de Alimentos	
6	Ana Ciarfella	8315404	Docente Externo	
7	Ana Leonor Bonilla Rivero	5073256	Zoología	
8	Ana Mercedes Herrera Benzecri	37713907	Botánica	
9	Ana Rascon De Tovar	2766482	Biología Celular	
10	Andrea Menendez Yuffa	6239224	Botánica	

**Figura 4.12: Opción de Listado de Docentes.**

Al ingresar en la opción de Listar Estudiantes o Listar Profesores, se puede de igual manera agregar, eliminar o modificar los datos del usuario haciendo clic sobre los iconos mostrados en la figura 4.13.



**Figura 4.13: Botones para agregas, editar y eliminar usuario.**

Al hacer clic sobre el botón de eliminar, mostrado en la figura 4.14, se mostrará la ventana de diálogo de la figura 4.14, para confirmar que efectivamente se desea eliminar dicho usuario.

**Eliminar Profesor** xClose

¿Está seguro que desea eliminar el siguiente Profesor?

Nombre:  
Pedro Jose Romero Ruiz

Cédula:  
622057

**Figura 4.14: Ventana emergente para eliminar usuario.**

Si se hace clic sobre el botón de agregar se desplegara una ventana emergente, para completar los datos de registro del nuevo usuario, tal como se muestra en la figura 4.15.

**Crear Nuevo Profesor** x

Complete los siguientes campos:

Cédula:

Nombre:

Departamento:

Tipo de Profesor:

Correo Electrónico:

Correo Alternativo:

Teléfono:

Teléfono Secundario:

**Figura 4.15: Ventana emergente para crear usuario.**

Finalmente, si se presiona el botón de modificar, se desplegará una ventana emergente, como la mostrada en la figura 4.16, que permite editar los datos del usuario ya creado.

The screenshot shows a modal window titled "Editar Profesor" with a close button (X) in the top right corner. Below the title, it says "Complete los Sigüientes Campos:". The form contains several input fields: "Cédula:" with the value "622057"; "Nombres y Apellidos:" with the value "Pedro Jose Romero Ruiz"; "Departamento:" with a dropdown menu showing "Biología Celular"; "Correo Electrónico:" with the value "javiervergara2004@gmail.com"; "Correo Alternativo:" with the value "Correo Electrónico Secundario"; "Teléfono:" with the value "0212-7535897"; and "Teléfono Secundario:" with the value "Teléfono Secundario". At the bottom right, there are two buttons: "Cancelar" and "Aceptar".

**Figura 4.16: Ventana emergente para editar usuario.**

#### 4.4 Perfil de Estudiante

Una vez que el estudiante inició sesión en la aplicación, se muestra el panel de opciones mostrado anteriormente en la figura 4.6. La opción “actualizar datos” nos muestra un formulario, que contiene los datos del estudiante para su actualización, en la figura 4.17 se puede observar.

The screenshot shows a modal window titled "Actualizar Datos". It contains several input fields, each with a blue checkmark icon to its right, indicating they are filled: "Cédula:" with "17389551"; "Nombre:" with "Francisco Javier Vergara Sandoval"; "Correo Electronico:" with "javiervergara2004@gmail.com"; "Correo Alternativo:" with "javiervergara@gmcaail.com"; "Teléfono:" with "0212-4842125"; and "Dirección:" with "Avenida Siempre Viva Springfield". The "Teléfono Celular:" field is empty and has a red border, with a red error message below it: "Este campo no puede quedar vacío". At the bottom, there are two buttons: "Guardar" and "Borrar".

**Figura 4.17: Ventana emergente para Actualizar datos.**

Entre las opciones se encuentra de los Estudiantes se encuentra “formalizar inscripción” y “cargar documento” inscrito, las cuales se pueden observar en las figuras 4.18 y 4.19.



The screenshot shows a web form titled "Datos del Proyecto". It contains four input fields: "Tipo Inscripción" with the value "Seminario I", "Departamento" with the value "Tecnología de Alimentos", "Titulo del Proyecto" with the placeholder text "Escriba el título del Proyecto Inscrito.", and "Objetivo General" with the placeholder text "Escriba el Objetivo General del Proyecto Inscrito.". At the bottom, there is a navigation bar with four buttons: "Datos Personales", "Proyecto", "Tutor", and "Enviar".

**Figura 4.18: Opción de formalizar inscripción.**



The screenshot shows a web form titled "Cargar Documento". It features a dropdown menu for "Tipo de Inscripción" with "Seminario I" selected. Below this, it says "Seleccione el documento a cargar:" followed by a red PDF icon and a "Quitar Archivo" link. Underneath, it displays "El nombre del archivo cargado es:" followed by a text box containing "diagramas de actividades.pdf". A blue "Subir Archivo" button is positioned below the text box. At the bottom, a warning message reads: "Advertencia: recuerde que únicamente puede subir este documento una vez".

**Figura 4.19: Opción de Cargar Documento.**

En la opción de “formalizar inscripción”, se completan los datos referentes a la inscripción de la materia a formalizar (Seminario I, Seminario II o TEG), y los datos correspondientes del proyecto como: título, objetivo general y tutor. Luego en la opción “Cargar Documento”, se muestra un botón para realizar la carga del documento especificado en la formalización de la inscripción.

Otra de las opciones del Módulo de Estudiantes permite visualizar y descargar la planilla de formalización, para ingresar en dicha opción se debe hacer clic en la opción “Imprimir Planilla”, lo cual nos enviará a las opciones mostradas en la figura 4.20. Al hacer clic sobre la opción “Descargar Planilla”, se descargará la misma en formato pdf, si por el contrario se hace clic en “Ver Planilla” se mostrará la planilla en el navegador tal como muestra la figura 4.21.



**Figura 4.20: Opción de imprimir planilla.**





viernes 17 de abril del 2015

PLANILLA PARA LA ELABORACIÓN DEL PROYECTO DE  
TRABAJO ESPECIAL DE GRADO (SEMINARIO I)

SECCIÓN I: DATOS PERSONALES DEL ESTUDIANTE	
Apellido(s) y Nombre(s): FRANCISCO JAVIER VERGARA SANDOVAL	C.I: V-17389551
Dirección: Avenida Siempre Viva Springfield	
Teléfono Habitación: 0212-4842125	Teléfono Celular: 0424-1504652
Correo principal: javiervergara2004@gmail.com	Correo alternativo: javiervergara@gmcaail.com
Departamento: TECNOLOGÍA DE ALIMENTOS	

SECCIÓN II: DATOS DEL TUTOR	
Apellido(s) y Nombre(s): JORMAN SANDOVAL	
Institución a la cual pertenece (señala la ubicación completa): Entidad Externa	
Teléfono Oficina: 0212-4842122	Teléfono Celular: 0424-1504652
Correo Principal: javiervergara2004@gmail.com	Correo Secundario: javiervergara2004@gmail.com

SECCIÓN III: DATOS DEL PROYECTO	
Titulo: COMPORTAMIENTO DE SIETE CEPAS DE (RHIZOBIUM LEGUMINOSARUM BIOVAR, PHASEOLI) Y EFECTO DE	
LA FERTILIZACION FOSFORICA EN EL CRECIMIENTO DE FREJOL (PHASEOLUS VULGARIS L.) EN	
INVERNADERO	
Objetivo General: COMPORTAMIENTO DE SIETE CEPAS DE (RHIZOBIUM LEGUMINOSARUM BIOVAR, PHASEOLI) Y	
EFECTO DE LA FERTILIZACION FOSFORICA EN EL CRECIMIENTO DE FREJOL (PHASEOLUS	
VULGARIS L.) EN INVERNADERO	

**Figura 4.21: Planilla de formalización.**

El estudiante puede ver las observaciones que realizan los profesores sobre su formalización de inscripción, al hacer clic sobre la opción del panel “Ver Observación”, se mostrarán aquellas observaciones realizadas por docentes acerca de la formalización, tal como se ve en la figura 4.22.

::OBSERVACIONES::				
#	Fecha	Docente	Tipo	Observación
1	2015-04-04 00:39:06	Jorman Sandoval	Rechazo de Documento por el Tutor	El documento no posee bien especifi ...
2	2015-04-04 00:39:25	Jorman Sandoval	Rechazo de Documento por el Tutor	el documento no cumple con los esta ...

**Figura 4.22: Lista de Observaciones. Elaboración propia.**

Al presionar el botón “Ver más” en la lista de observaciones, se desplegará una ventana emergente con todos los detalles de la observación (figura 4.23).

**Detalle de la Observación** ✕

Nombre del Docente:  
**Jorman Sandoval**

Fecha:  
**2015-04-04 00:39:06**

Observación:

El documento no posee bien especificado el objetivo general y los objetivos específicos

**Figura 4.23: Detalle de Observación.**

Una opción del panel que tienen todos los usuarios en común, es permitirles cambiar la contraseña por una de su preferencia cuando lo deseen. Para ello, se debe seleccionar la opción del panel “Cambiar Contraseña”, lo cual nos desplegará el formulario mostrado en la figura 4.24, en el que debemos colocar la nueva contraseña.

**Cambio de Contraseña**

Inserte su nueva contraseña.  
Esta debe ser distinta a su usuario y debe de coincidir en los dos campos

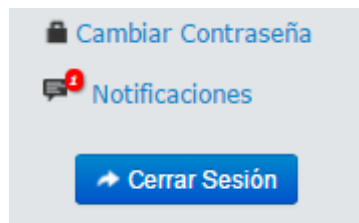
Introduzca la contraseña actual:

Introduzca la nueva contraseña:

Repita la nueva contraseña:

**Figura 4.24: Formulario para cambiar contraseña.**

La siguiente funcionalidad es ver las notificaciones de la aplicación cuando ocurre un evento durante el proceso. En la figura 4.25 se observa la opción llamada “Notificaciones”, la cual indica con un contador en color rojo que se tiene una notificación por leer. Al hacer clic sobre la opción muestra al usuario el listado de notificaciones de la aplicación mostrado en la figura 4.26.

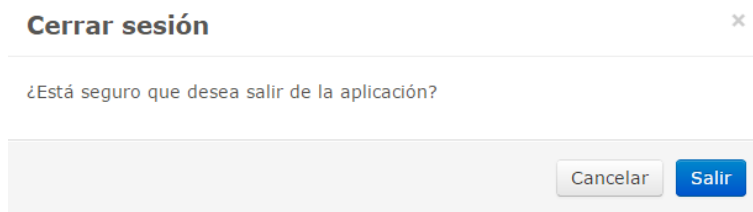


**Figura 4.25: Opción para ver las notificaciones de la aplicación.**

<b>Notificaciones del Sistema Simbiosis:</b>		
#	Fecha	Mensaje
1	2015-04-12 11:05:11	El Docente Jorman Sandoval ha aceptado la solicitud para ser su tutor. Por favor cargue el documento.

**Figura 4.26: Listado de las notificaciones de la aplicación.**

Finalmente el usuario dispone de la opción cerrar sesión, lo cual finalizará la sesión actual y eliminará todos los datos de la sesión, al presionar la opción “Cerrar Sesión”, se muestra un cuadro de diálogo mostrado en la figura 4.27, para confirmar que el usuario quiere salir de la aplicación, luego enviará al usuario al inicio de la aplicación.



**Figura 4.27: Cuadro de diálogo para finalizar sesión.**

#### 4.4 Módulo de Docentes

A continuación se mostrarán las funcionalidades que puede realizar el docente en la aplicación durante el proceso de formalización. La opción “Solicitudes de Tutoría”, permite al docente verificar aquellas solicitudes que le haya realizado un estudiante para dar tutoría en su inscripción, bien sea Seminario I, Seminario II o TEG.

En la figura 4.28 se muestra la opción del listado de solicitudes de tutoría para docentes.



#	Nombre y Apellido	Cedula	Estatus	Acciones
1	Francisco Javier Vergara Sandoval	17389551	pendiente	<input checked="" type="checkbox"/> <input type="checkbox"/>

**Figura 4.28: Opción para listar las solicitudes de tutoría.**

Al listar las solicitudes éstas pueden ser aceptadas o rechazadas por el docente. En el primero de los casos al hacer clic sobre el visto bueno en color verde el docente acepta la solicitud y se envía al estudiante una notificación de la decisión del profesor. En caso de que el tutor desee rechazar dicha solicitud, debe presionar la equis en color rojo y se desplegará un cuadro de diálogo en el que se le pedirá confirmar el rechazo a la solicitud (ver figura 4.29).



**Figura 4.29: Cuadro de diálogo para rechazar solicitud.**

La función “Solicitudes de Jurados”, es análoga a la opción anterior, con la diferencia de que lista aquellas solicitudes realizadas por la dirección de la escuela para ser jurado en un TEG.

A continuación, se encuentra la sección de tutoría en la cual el docente puede ver el documento cargado por el estudiante, aceptar o rechazar dicho documento, y ver la planilla de formalización de aquellos estudiantes cuyo documento ya haya sido aprobado. En la figura 4.30, se muestra la información mostrada al ingresar en la opción de tutoría.

Tutoría en el Sistema Simbiosis:				
Solicitudes con Documento por Aprobar				
#	Nombre y Apellido	C.I.No.	Email	Acciones
1	Francisco Javier Vergara Sandoval	17389551	javiervergara2004@gmail.com	🔍 ✓ ✕

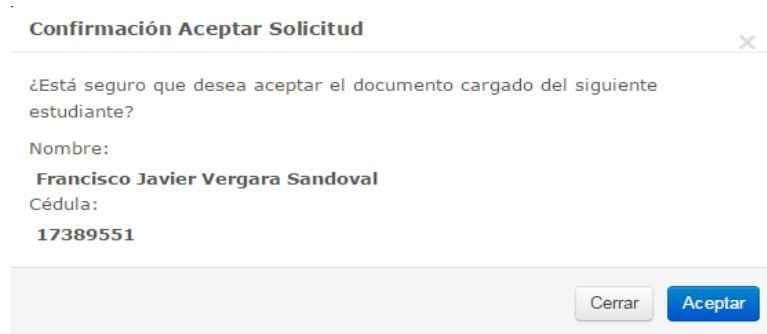
  

Planillas de Solicitudes Aprobadas				
#	Nombre y Apellido	C.I.No.	Email	Acciones

**Figura 4.30: Opción de tutoría en el perfil docente.**

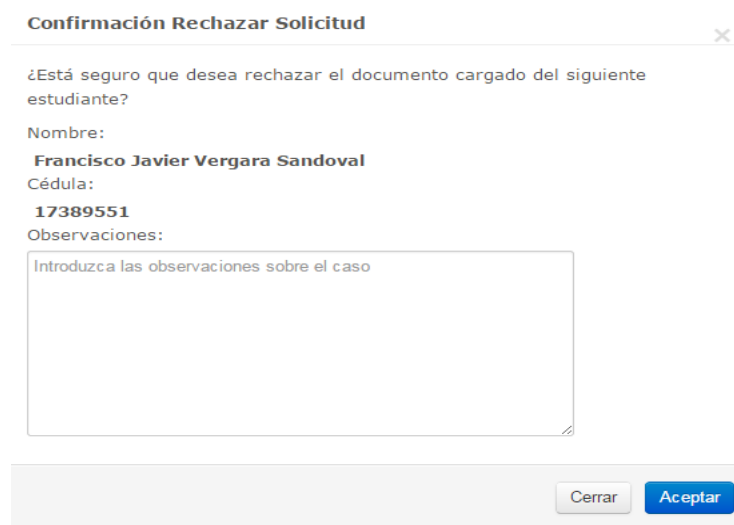
Al hacer clic en la acción marcada con una flecha, se iniciará la descarga del documento cargado por el estudiante en la aplicación.

Si el docente hace clic en la acción marcada con un visto bueno, se pedirá la confirmación de aceptación del documento, como se muestra en la figura 4.31.



**Figura 4.31: Ventana de diálogo para aceptar el documento.**

En cambio, si el docente hace clic en la acción marcada con una equis, se pedirá la confirmación de rechazo del documento, y la observación pertinente para justificar el rechazo, tal como se muestra en la figura 4.32.



**Figura 4.32: Ventana de diálogo para rechazar el documento.**

Otra de las funcionalidades para docentes es el opción de poder evaluar al estudiante una vez haya presentado y cumplido todos los requerimientos para aprobar la asignatura que haya inscrito. Mediante la opción "Cargar Notas", el docente puede asignar una calificación al estudiante para tenerla como histórico en la aplicación. En la Figura 4.33 se muestra la opción de cargar nota, la cual nos enseña el listado de estudiantes bajo tutoría y las opciones realizar las acciones de descargar la planilla de formalización y colocar la calificación obtenida.

Agregar Notas de los Estudiantes que posee bajo tutoría.					
#	Nombre y Apellido	C.I.No.	Tema	Nota	Acciones
1	MARIA GABRIELA MORENO AVILA	18183939	Tema 2	SN	

**Figura 4.33: Opción para Cargar Notas en la aplicación.**

Al hacer clic sobre el ícono en color verde, se desplegará una ventana con los datos del estudiante y un campo donde puede seleccionar la nota para realizar la calificación, tal como se muestra en la figura 4.34.

Calificacando Estudiante x

---

Nombre: MARIA GABRIELA MORENO AVILA

Cédula: 18183939

Nota del Estudiante:

---

**Figura 4.34: Ventana para calificar al estudiante.**





Las siguientes funcionalidades corresponden al perfil docente con el rol de Jefe de Departamento. Básicamente el Jefe de Departamento cuenta con las mismas funciones que un profesor tutor, teniendo opciones adicionales inherentes a su departamento como son: ver listados de estudiantes, listar planillas de alumnos inscritos y observar calificaciones obtenidas por los estudiantes.

En la figura 4.35, se puede apreciar la funcionalidad para listar planillas de alumnos inscritos por el Jefe de Departamento, el cual permite ver y descargar la planilla para su revisión al hacer clic sobre la opción con el ícono de observar.

Lista de Planillas en el Sistema Simbiosis:				
Planillas por Aprobar				
#	Nombre y Apellido	C.I.No.	Email	Acciones
Planillas Aprobadas				
#	Nombre y Apellido	C.I.No.	Email	Acciones
1	Francisco Javier Vergara Sandoval	17389551	javiervergara2004@gmail.com	

**Figura 4.35: Ventana para listar planillas del departamento. .**

El Jefe de Departamento puede además ver las calificaciones obtenidas por los distintos estudiantes pertenecientes a su departamento, en la figura 4.36 se observa dicha funcionalidad, la cual nos permite ver el documento evaluado y hacer modificaciones sobre la calificación cargada en la aplicación.

Listados de Notas de Estudiantes del Departamento					
#	Nombre y Apellido	C.I.No.	Tema	Nota	Acciones
1	MARIA GABRIELA MORENO AVILA	18183939	Tema 2	20	 
2	FRANCISCO JAVIER VERGARA	17389551	Tema 5	18	 

**Figura 4.36: Ventana para listar las notas cargadas. .**

A continuación se describirán las funcionalidades corresponden al perfil docente con el rol de Representante de la UDPP. El representante de la UDPP, cuenta con las mismas funcionalidades de un profesor tutor y un Jefe de Departamento, teniendo la opción adicional de enviar comunicados de interés a docentes y estudiantes de su departamento, la cual también puede ser ejecutada por el perfil docente con rol de director y el perfil de Dirección.

En la figura 4.36se muestra la funcionalidad de enviar comunicado, la cual nos permite escoger al grupo de destinatarios, el asunto del comunicado a enviar y el mensaje del comunicado.



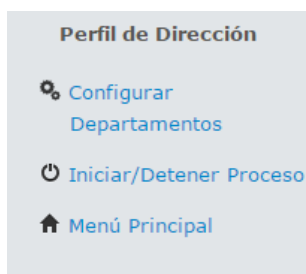
**Figura 4.37: Funcionalidad para enviar comunicados.**

El siguiente perfil a describir es el de docente con rol de director, el cual posee perfil de superusuario o control total sobre la aplicación. A continuación se describen las funcionalidades principales de este perfil.



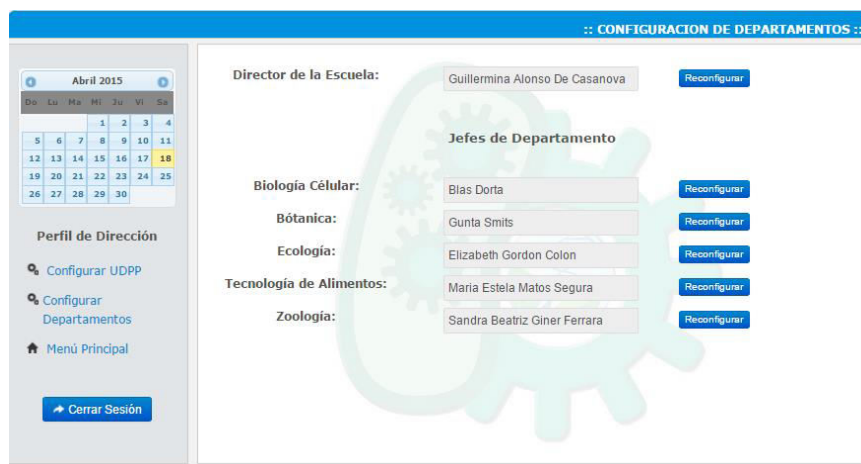
La primera opción a describir para el perfil dirección, es la de configuración de la aplicación, la cual permite al usuario configurar los docentes que serán representantes de la UDPP, Jefes de Departamento y Director, para el semestre en curso; adicionalmente permite iniciar, detener o finalizar el proceso de formalización.

En la figura 4.38 se observan las opciones mencionadas anteriormente, al ingresar en “Configurar Departamentos”, mostrará al usuario el menú mostrado en la figura 4.39, la cual muestra las opciones de configuración de departamentos e iniciar o detener el proceso de formalización.



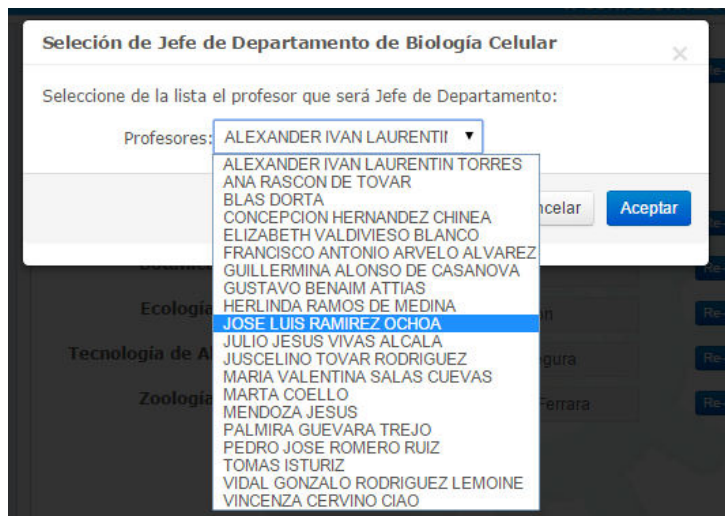
**Figura 4.38: Menú de configuración de la aplicación.**

Al hacer clic en la opción “Configurar Departamentos” se muestra el menú y opciones mostrados en la figura 4.39, en dicha interfaz podemos asignar al docente que ejercerá el cargo como Director de la Escuela de Biología y a los Jefes de Departamento.



**Figura 4.39: Funcionalidad de configuración de la aplicación.**

Al hacer clic en el botón reconfigurar, se despliega la ventana mostrada en la figura 4.40 que nos permite seleccionar un docente de la aplicación para asignarlo en el cargo.



**Figura 4.40: Ventana para escoger docentes.**

Si se desea configurar los Representantes de la UDPP, se debe hacer clic en la opción “Configurar UDPP”, lo cual llevará al usuario a la interfaz mostrada en la figura 4.41.



**Figura 4.41: Funcionalidad de configuración de la UDPP.**

La siguiente funcionalidad del perfil Dirección es poder iniciar, detener, o finalizar el proceso de formalización durante cualquier instante del semestre. Al hacer clic en la opción “Iniciar/Detener Proceso” se llevará al usuario a una interfaz donde puede hacer las acciones anteriormente mencionadas. Si el proceso aún no ha sido iniciado se verá la interfaz mostrada en la figura 4.42, la cual muestra el botón Iniciar Proceso en color verde. Por el contrario, si el proceso ya ha sido iniciado con anterioridad, se mostrará la interfaz de la figura 4.43 en la que se puede detener el proceso o finalizarlo.



**Figura 4.42: Funcionalidad para iniciar el proceso de formalización.**



**Figura 4.43: Funcionalidad para cambiar el status del proceso de formalización.**

La última funcionalidad por describir para el perfil de Dirección, es la generación de reportes y estadísticas de la aplicación, la cual permite generar reportes de los estudiantes inscritos, profesores registrados en la aplicación y estudiantes que han o no formalizado su inscripción. Al hacer clic en la opción “Generar Reportes”, se mostrará un listado de opciones para generar distintos reportes, como se aprecia en la figura 4.44, y mostrará un menú de opciones en el panel lateral izquierdo, en el que podemos realizar reportes por estudiantes, por docentes o por estudiantes que han formalizado su inscripción.

**Figura 4.44: Funcionalidad para generar reportes.**

Al hacer clic en cualquiera de los reportes genéricos de la funcionalidad, se despliega la ventana mostrada en la figura 4.45, que contiene un informe detallado en formato pdf, que lista la opción seleccionada.

UNIVERSIDAD CENTRAL DE VENEZUELA  
 FACULTAD DE CIENCIAS  
 ESCUELA DE BIOLOGIA  
 DIRECCION DE LA ESCUELA  
 SISTEMA SIMBIOSIS  
 LISTADO DE PROFESORES INTERNOS  
 TOTAL PROFESORES: 115

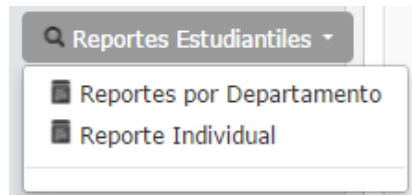


Lunes 20 de abril del 2015

Nro	Cedula	Nombre	Telefono	Telefono	Email	Departamento	Entidad	Estado
1	9565764	ALEXANDER IVAN LAURENTIN TORRES	0212-7751011		javiervergara2004@gmail.com	BIOLOGIA CELULAR	INTERNO	ACTIVO
2	4361203	ALICIA JOSEFINA CACERES SANCHEZ	0212-7510544		javiervergara2004@gmail.com	BOTANICA	INTERNO	ACTIVO
3	6364695	ALONSO DAVID OJEDA	0212-6051090		javiervergara2004@gmail.com	ECOLOGIA	INTERNO	ACTIVO
4	3500600	AMAURY MARTINEZ	0212-7533871		javiervergara2004@gmail.com	TECNOLOGIA DE ALIMENTOS	INTERNO	ACTIVO
5	5073256	ANA LEONOR BONILLA RIVERO	0212-6051309		javiervergara2004@gmail.com	ZOOLOGIA	INTERNO	ACTIVO
6	37713907	ANA MERCEDES HERRERA BENZECRI	0212-7510544		javiervergara2004@gmail.com	BOTANICA	INTERNO	ACTIVO
7	2766482	ANA RASCON DE TOVAR	0212-7510111	4126625515	javiervergara2004@gmail.com	BIOLOGIA CELULAR	INTERNO	ACTIVO
8	6239224	ANDREA MENENDEZ YUFFA	0212-7510111		javiervergara2004@gmail.com	BOTANICA	INTERNO	ACTIVO
9	3400571	ANIBAL JOSE CASTILLO SUAREZ	0212-7535897		javiervergara2004@gmail.com	BOTANICA	INTERNO	ACTIVO
10	3139563	ANTONIO JOSE MACHADO ALLISON	0212-6051534		javiervergara2004@gmail.com	ECOLOGIA	INTERNO	ACTIVO
11	6160230	BLAS DORTA	0212-7535897		javiervergara2004@gmail.com	BIOLOGIA CELULAR	INTERNO	ACTIVO
12	3973369	CARIBAY TERESA URBINA DE NAVARRO	0212-6051036		javiervergara2004@gmail.com	ECOLOGIA	INTERNO	ACTIVO
13	3182096	CARLOS ENRIQUE AYESTA CHAVEZ	0212-6051259		javiervergara2004@gmail.com	ECOLOGIA	INTERNO	ACTIVO
14	4359157	CARMEN FERREIRA MARQUEZ	0212-6051301		javiervergara2004@gmail.com	ZOOLOGIA	INTERNO	ACTIVO
15	5073799	CONCEPCION HERNANDEZ CHINEA	0212-7510111		javiervergara2004@gmail.com	BIOLOGIA CELULAR	INTERNO	ACTIVO
16	3366068	CRESSA CLAUDIA	0212-7535897		javiervergara2004@gmail.com	ECOLOGIA	INTERNO	ACTIVO
17	3408062	DIEGO JOSE RODRIGUEZ MORALES	0212-6051579		javiervergara2004@gmail.com	ECOLOGIA	INTERNO	ACTIVO
18	2619455	EDIE MONTIEL	0212-6051399		javiervergara2004@gmail.com	ECOLOGIA	INTERNO	ACTIVO

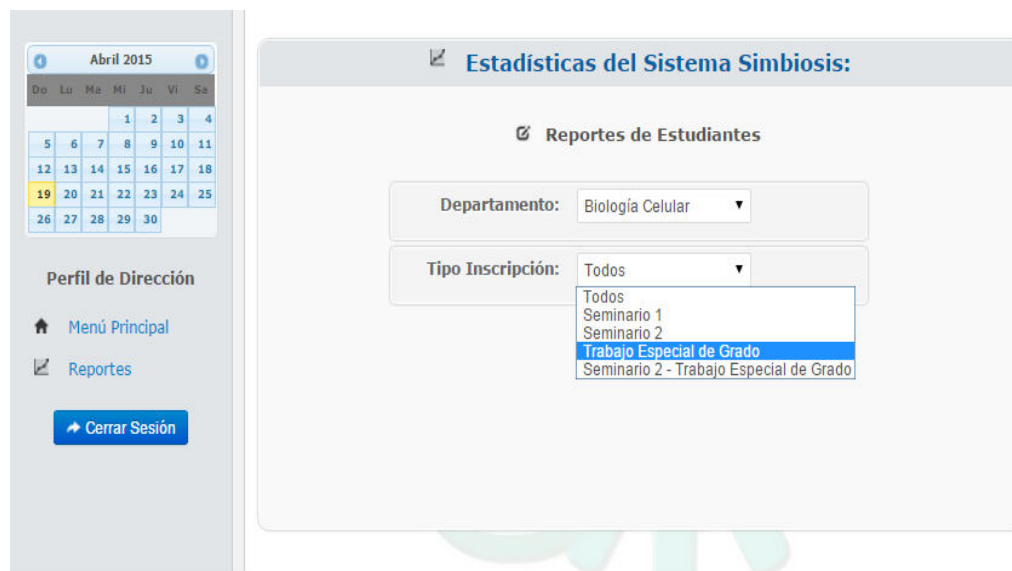
**Figura 4.45: Reporte generado mediante la aplicación.**

Si el usuario hace clic en las opciones del panel lateral, se mostrarán dos opciones (figura 4.46), en la cual puede generar reportes por departamento o reportes individuales.



**Figura 4.46: Opciones para generar reportes.**


Al hacer ingresar en “Reportes por Departamento”, se desplegará la ventana mostrada en la figura 4.47, en la que podemos escoger, el departamento y el tipo de inscripción en el caso de estudiantes, o el tipo de docente en el caso de profesores.





**Figura 4.47: Funcionalidad para generar reportes por departamentos.**

Si se ingresa en la opción “Reporte Individual”, se solicitará al usuario la cédula de identidad, tal como se muestra en la figura 4.48, y al presionar el botón generar, se creará un documento con la información referente a la cédula ingresada como se aprecia en la figura 4.49.

Figura 4.48: Funcionalidad para generar reportes por departamentos.


**Universidad Central de Venezuela**  
 Facultad de Ciencias  
 Escuela de Biología  
 Dirección

viernes 17 de abril del 2015

PLANILLA PARA LA ELABORACIÓN DEL PROYECTO DE TRABAJO ESPECIAL DE GRADO (SEMINARIO I)

SECCIÓN I: DATOS PERSONALES DEL ESTUDIANTE	
Apellido(s) y Nombre(s): FRANCISCO JAVIER VERGARA SANDOVAL	C.I: V-17389551
Dirección: Avenida Siempre Viva Springfield	
Teléfono Habitación: 0212-4842125	Teléfono Celular: 0424-1504652
Correo principal: javiervergara2004@gmail.com	Correo alternativo: javiervergara@gmcaail.com
Departamento: TECNOLOGÍA DE ALIMENTOS	

SECCIÓN II: DATOS DEL TUTOR	
Apellido(s) y Nombre(s): JORMAN SANDOVAL	
Institución a la cual pertenece (señala la ubicación completa): Entidad Externa	
Teléfono Oficina: 0212-4842122	Teléfono Celular: 0424-1504652
Correo Principal: javiervergara2004@gmail.com	Correo Secundario: javiervergara2004@gmail.com

SECCIÓN III: DATOS DEL PROYECTO	
Título: COMPORTAMIENTO DE SIETE CEPAS DE (RHIZOBIUM LEGUMINOSARUM BIOVAR, PHASEOLI) Y EFECTO DE LA FERTILIZACION FOSFORICA EN EL CRECIMIENTO DE FREJOL (PHASEOLUS VULGARIS L.) EN INVERNADERO	
Objetivo General: COMPORTAMIENTO DE SIETE CEPAS DE (RHIZOBIUM LEGUMINOSARUM BIOVAR, PHASEOLI) Y EFECTO DE LA FERTILIZACION FOSFORICA EN EL CRECIMIENTO DE FREJOL (PHASEOLUS VULGARIS L.) EN INVERNADERO	

Universidad Central de Venezuela - Facultad de Ciencias - Escuela de Biología - Sistema Simbiosis - Página 1/2

Figura 4.49: Reporte generado individualmente.

Mediante el prototipo anterior se pudieron visualizar las principales funcionalidades para los distintos usuarios que usarán la aplicación. Es importante resaltar que por cada acción que se realicé durante el proceso, la aplicación enviará un correo para notificar este cambio e incentivar a la comunidad a ingresar y usar la aplicación con mayor frecuencia.



## Conclusiones

La Escuela de Biología de la UCV no contaba con una aplicación para gestionar la inscripción de Seminarios y TEG por parte de los estudiantes, por lo que en este TEG se cumplió el objetivo general, el cual fue el desarrollo de una aplicación web Simbiosis, que gestiona el proceso relacionado a las diversas solicitudes estudiantiles y actividades que se realizan durante el proceso mencionado, la cual es sencilla de utilizar para los docentes y estudiantes.

Se logró la obtención de un producto final que cumple con todos los requerimientos planteados por los usuarios al principio de este trabajo, contribuyendo de esta manera con la automatización de una cantidad de tareas altamente demandadas e indispensables dentro de la dirección de la Escuela de Biología. Simbiosis fue desarrollada utilizando tecnologías de desarrollo web Open Source de vanguardia, permitiendo un desarrollo económico de la misma, facilitando así su construcción.

Se logró tomar en cuenta cada uno de los requerimientos planteados. Los procesos y actividades tomados en cuenta en este TEG abarcan a un grupo de personas (estudiantes, docentes y personal administrativo) que forman parte del personal de la Escuela de Biología y están directamente involucrados con el proceso de formalización de inscripciones.

El uso de PHP como tecnología del lado del servidor agilizó considerablemente el proceso de desarrollo, ya que se aprovecharon las diferentes estructuras y métodos predefinidos que posee, los cuales minimizan el tiempo y esfuerzo de codificación. Asimismo, la aplicación de estándares de programación ayudó a la obtención de un código legible y ordenado, lo que facilita el mantenimiento y evolución de la aplicación a lo largo del tiempo. El paradigma MVC que se implementó, brindó una organización para trabajar en el desarrollo, ya que permite tener una separación lógica y física de los componentes de la aplicación.

El método AgilUs permitió que el diseño y construcción se realizara en iteraciones, facilitó el desarrollo de la aplicación, puesto que dio flexibilidad para cubrir e implementar cada uno de los requerimientos, además de garantizar la constante participación del usuario en todo el proceso de desarrollo, quienes manifestaron su conformidad con el desempeño de la aplicación y con los resultados obtenidos.



Adicionalmente, el método AgilUs permitió implementar un conjunto de técnicas y actividades de forma organizada y rápida. Igualmente, mediante AgilUs fue posible adaptar a las necesidades por parte del equipo de desarrollo y del usuario para la implementación de esta aplicación.

En cuanto a la usabilidad, mediante AgilUs se usaron técnicas que permitieron medir este factor de calidad en la aplicación. Se emplearon pruebas de usabilidad a distintos usuarios con el perfil necesario para usar la aplicación y se encontraron problemas de usabilidad que fueron solventados. Además se facilitó la interfaz de usuario de los distintos módulos a los potenciales usuarios de la aplicación y éstos aceptaron la usabilidad de los mismos.

Esta aplicación Web permite llevar a cabo procesos de solicitudes estudiantiles y docentes de manera eficaz y eficiente. Al mismo tiempo, se ha contribuido en la reducción de errores cometidos durante la realización de las tareas involucradas en los procesos abarcados en este TEG, ya que los datos utilizados son los extraídos directamente de la aplicación CONEST, alcanzando una mayor satisfacción por parte de los involucrados, reduciendo el tiempo invertido en la realización de listados.

Debido a las pruebas realizadas en cada iteración se verificaron los errores y fueron corregidos inmediatamente, de igual forma, los aportes del personal administrativo, el objetivo general y los objetivos específicos planteados al comienzo del trabajo especial de grado fueron alcanzados.

Para finalizar, la elaboración de este TEG permitió realizar un aporte significativo a la Escuela de Biología de la Facultad de Ciencias de la UCV, optimizando el proceso de inscripción de las Materias Seminario I y II y TEG mediante Simbiosis, aplicación que permitirá una gestión automatizada del mencionado proceso.

## Recomendaciones

A continuación se presenta una lista de recomendaciones y/o sugerencias que están fuera del alcance del trabajo especial de grado y que pudieran ser aplicadas a las bases de la aplicación existente:

- Entre las limitaciones encontradas en la aplicación, se puede indicar que no se permite la personalización del diseño, por lo que puede ser una mejora que se puede realizar al sistema.
- Integrar la aplicación web Simbiosis con una aplicación que actualmente se encuentra culminada llamada Biotec, la cual permite la inscripción de las materias de las Materias Métodos de Laboratorio y Laboratorio.
- La aplicación web fue desarrollada estrictamente para cumplir el proceso de gestión de inscripción establecido en el reglamento de la Escuela de Biología de la UCV, por lo que se deben implementar adaptaciones si se desea trasladar a otras escuelas.
- La aplicación depende de un administrador quien es el encargado de agregar a los docentes al sistema, cargar listados o iniciar el proceso a principio de semestre, por lo que sería recomendable automatizar dichas actividades sin recurrir a la interacción humana.
- Un módulo para la conexión al sistema de gestión académica CONEST, lo cual reduciría el trabajo del administrador para la carga de docentes y estudiantes.
- Un servicio web que permita la carga de las calificaciones de manera automática desde el sitio web generado hacia la aplicación de gestión académica CONEST, lo que optimizaría la carga de notas.
- Una interfaz de programación de aplicaciones (API, por sus siglas en inglés) para teléfonos inteligentes, lo cual facilitaría el acceso al sitio web.
- Funcionalidades de accesibilidad que permitan la inclusión de los usuarios con discapacidades, como aumentar el tamaño de las fuentes, cambiar el contraste de los colores, entre otros.
- Ampliación de las funcionalidades del módulo de docentes que les permita tener un mayor control sobre sus tesis o seminaristas; como por ejemplo el ingreso de sus horarios disponibles, la comunicación vía chat con sus alumnos, etc.

## Referencias Bibliográficas

- Alvarez, R. (2010). Introducción a HTML. Recuperado el 15 de Abril de 2013, de <http://www.desarrolloweb.com/articulos/534.php>
- Apache (2012). Servidor HTTP Apache. Recuperado el 6 de abril de 2014, de <http://projects.apache.org/indexes/releases.html>
- Atlassian Bitbucket (2014). Recuperado el 15 de Agosto 2013, de <https://bitbucket.org/>
- Barzanallana, R. (2012). ¿Qué es una aplicación web?. Recuperado el 15 de Abril de 2013, de <http://www.um.es/docencia/Historia-desarrollo-aplicaciones-web.html>
- Bootstrap (2014). Recuperado el 14 de Agosto de 2014, de <http://getbootstrap.com/2.3.2/getting-started.html>
- Brown, Dan M. (2011). Communicating Design: Developing Web Site Documentation for Design and Planning, Second Edition. New Riders.
- Burnham, T. (2011). CoffeeScript: Accelerated JavaScript Development. Pragmatic Programmers.
- Chesco. (2007). CSS (Cascade Style Sheets). Recuperado el 17 de Abril de 2013, de <http://www.chesco.info/curso/css.htm>
- Cockburn, A. (2000). Agile Software Development. Highsmith Series. Recuperado el 21 de Abril de 2013, de <http://www.itu.dk/~oladjones/semester2/Project2/materials/newmaterials/Agile%20Software%20development.pdf>
- Creative Commons. (2011). Creative Commons. Recuperado el 30 de Agosto de 2012, de <http://creativecommons.org/>
- Diccionario de Informática. (2012). Definición de Aplicación Web. Recuperado el 15 de Abril de 2013, de <http://www.alegsa.com.ar/Dic/20web.php>.
- Dimagin. (2007). Dimagin: Las Aplicaciones Web. Recuperado el 11 de enero de 2015, de <http://www.dimagin.net/es/contenido.php?t id=6>.

Eguíluz, J. (2012). Introducción a CSS. Recuperado el 19 de enero de 2015, de <http://elbloggerperu.blogspot.com/2012/12/introduccion-css-javier-eguiluz-perez.html>.

Eguíluz, J. (2008). Introducción a JavaScript. Recuperado el 19 de Abril de 2013, de <http://es.scribd.com/doc/36947379/Javascript-manual-en-espanol>.

Enubes. (2013). Desarrollo de Aplicaciones Web. Recuperado el 15 de Abril de 2013, de <http://www.enubes.com/desarrollo-de-aplicaciones-web.html>

Escuela de Biología. (2008). Reseña historia y departamentos de la escuela de Biología. Recuperado el 20 de Marzo de 2013, de <http://www.ciens.ucv.ve/biologia/>

Git Book (2013). Recuperado el 20 de Marzo de 2013, de <http://git-scm.com/book/en/>

Good, R. (2005). Ventajas de las aplicaciones web. Recuperado el 16 de febrero de 2015, de <http://www.pixima.net/aplicaciones-web/ventajas-de-las-aplicaciones-web/>

Gutierrez, A. (2005). GSYC: Patrón de Diseño MVC. Recuperado el 3 de Octubre de 2014, de <http://gsysc.es/~agutierr/pfc-tecnicahtml/node43.html>

Hernández, P. El perfil del usuario de información. Recuperado el 20 de enero de 2015, de <http://www.ejournal.unam.mx/ibi/vol07-15/IBI000701502.pdf>

Ingeniero de Software (2008). Prácticas y métodos para mejorar el desarrollo de Proyectos de Software. Consultado el 18 de diciembre de 2014, de [www.ingenierosoftware.com](http://www.ingenierosoftware.com)

jQuery (2013). jQuery: What is JQuery?. Recuperado el 12 de Abril de 2013, de <http://jquery.com/>.

León, O.G. y Montero, I. (1997) Diseño de Investigaciones. Madrid: McGraw Hill.

Morales, J. (2012). Lenguajes del Lado del Cliente. Recuperado el 15 de Abril de 2013, de <http://www.slideshare.net/JeremiasMorales/22-lenguajes-del-lado-cliente-9782560>.

Newcomlab (2013). Modelo de Tres Capas. Recuperado el 15 de Abril de 2013, de [http://www.newcomlab.com/default.aspx?id\\_seccion=936](http://www.newcomlab.com/default.aspx?id_seccion=936).

MySQL (2013). About MySQL. Recuperado el 19 de Abril de 2013, de <http://www.mysql.com/about>.

- Pereda. (2007). ¿Qué es una Aplicación Web?. Recuperado el 15 de Abril de 2013, de <http://jimpereda.wordpress.com/2007/08/24/definiendo-la-plantilla>.
- Pérez, J. (s.f.). Notaciones y Lenguajes de Procesos. Una Visión Global. Recuperado el 05 de mayo de 2013, de <http://www.lsi.us.es/docs/doctorado/memorias/Perez,%20Juan%20D.pdf>.
- PHP (2013). Manual de PHP. Recuperado el 19 de Abril de 2013, de <http://www.php.net/manual/es/index.php>.
- Pressman, R. (2007). Ingeniería del Software: Un Enfoque Práctico. McGraw-Hill.
- R. G. Bias, D. J. Mayhew (2012). Cost-Justifying Usability. Academic Press, 1994.
- Schumacher y Lentz. (2007). Schumacher, R. y Lentz, A., s.f. Dispelling the Myths. Recuperado el 8 de septiembre de 2014, de <http://dev.mysql.com/tech-resources/articles/dispelling-the-myths.html>.
- Sommerville, I. (2005). Ingeniería del Software. Addison Wesley.
- Soto, L (2011). Requerimientos funcionales y no funcionales. Recuperado el 19 de enero de 2013, de [http://www.mitecnologico.com/Req\\_Funcionales](http://www.mitecnologico.com/Req_Funcionales).
- Universidad de Murcia (2011). Introducción a AngularJS. Recuperado el 15 de Agosto 2013, de <http://www.webdesignerdepot.com/2013/04/an-introduction-to-angularjs/>
- Universidad de Sevilla. (2009). Introducción a las Aplicaciones Web. Recuperado el 15 de Abril de 2013, de <http://www.lsi.us.es/docencia/get.php?id=3887>
- Vegas, J. (2012). Módulos: Introducción a las Aplicaciones Web. Recuperado el 11 de julio de 2014, de <http://www.infor.uva.es/~jvegas/cursos/buendia/pordocente/node11.html>.
- W3C (2013). Document Object Model (DOM). Recuperado el 05 de Abril de 2013, de <http://www.w3.org/DOM/>.
- W3C (2013). HTML and CSS. Recuperado el 25 de Mayo de 2013, de <http://www.w3.org/standards/webdesign/htmlcss>.

## ANEXO A

### Casos de Uso

**Tabla 1: Especificación del caso de uso: Actualizar Datos.**

Caso de uso: CU. 1.1.1 Actualizar Datos.
Actor: todos los actores.
Precondición: el usuario debe existir en la aplicación y debe haber iniciado sesión en el mismo.
Acción: funcionalidad que permite al usuario actualizar sus datos personales como correo electrónico y teléfonos de contacto.
Condición de Éxito: se actualizará la información en la BD de la aplicación web.

**Tabla 2: Especificación del caso de uso: Formalizar Inscripción.**

Caso de uso: CU 1.1.2 Formalizar Inscripción.
Actor: estudiante.
Precondición: el usuario debe haber iniciado sesión en la aplicación y debe estar cursando la asignatura correspondiente en el semestre en curso y haberse inscrito por CONEST.
Acción: funcionalidad que permite al usuario completar los datos referentes a su inscripción de Seminario I, Seminario II o TEG.
Condición de Éxito: se actualizará la información en la BD de la aplicación web.

**Tabla 3: Especificación del caso de uso: Cargar Documento.**

Caso de uso: CU 1.1.3 Cargar Documento.
Actor: estudiante.
Precondición: el usuario debe haber iniciado sesión en la aplicación y debe haber completado la información referente a la inscripción de seminario.
Acción: funcionalidad que permite al usuario cargar el documento especificado en la formalización de inscripción.
Condición de Éxito: se cargará el documento en la BD de la aplicación web.

**Tabla 4: Especificación del caso de uso: Imprimir Planilla.**

Caso de uso: CU 1.1.4 Imprimir Planilla.
Actor: estudiante.
Precondición: el usuario debe haber iniciado sesión en la aplicación y debe haber completado la información referente a la inscripción de seminario.
Acción: funcionalidad que permite al usuario imprimir o descargar la planilla de inscripción con la información especificada en la formalización de inscripción.
Condición de Éxito: se imprimirá o descargará la planilla de inscripción.

**Tabla 5: Especificación del caso de uso: Ver Observación. Parte A.**

Caso de uso: CU 1.1.5 Ver Observación.
Actor: estudiante.
Precondición: el usuario debe haber iniciado sesión en la aplicación.

**Tabla 6: Especificación del caso de uso: Ver Observación. Parte B**

Caso de uso: CU 1.1.5 Ver Observación.
Acción: funcionalidad que permite al usuario ver las observaciones realizadas referente a su inscripción.
Condición de Éxito: se desplegará la lista de observaciones realizadas sobre su inscripción en case de que las haya.

**Tabla 7: Especificación del caso de uso: Cambiar Contraseña.**

Caso de uso: CU 1.1.6 Cambiar Contraseña
Actor: todos los actores.
Precondición: el usuario debe haber iniciado sesión en la aplicación.
Acción: funcionalidad que permite al usuario cambiar la contraseña en la aplicación.
Condición de Éxito: se actualizará la contraseña en la aplicación.

**Tabla 8: Especificación del caso de uso: Ver Notificación.**

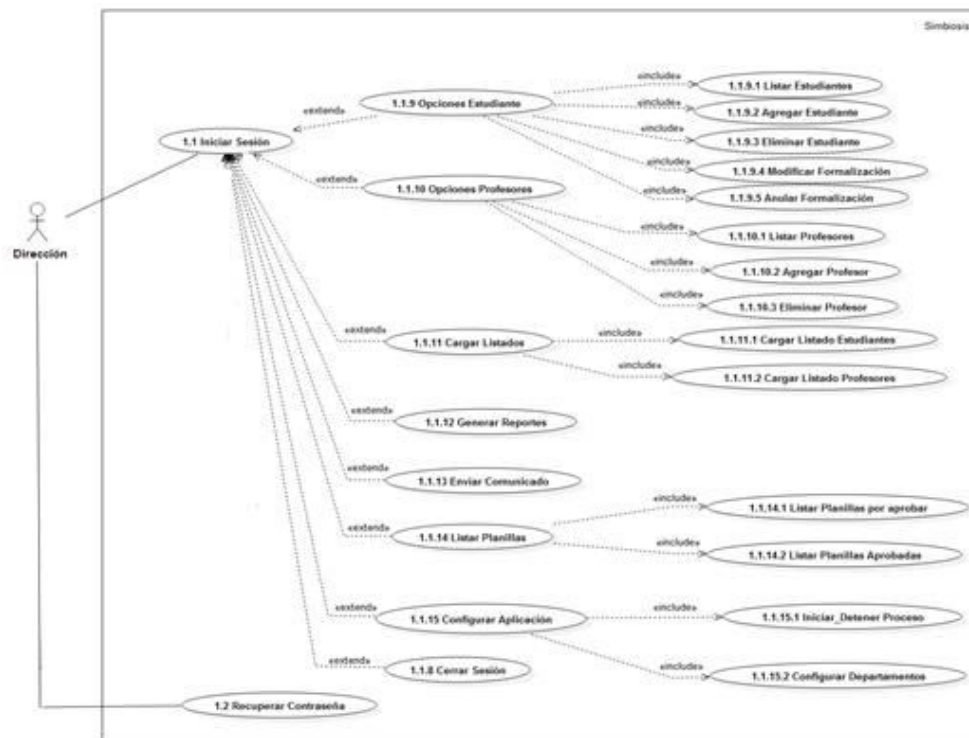
Caso de uso: CU 1.1.7 Ver Notificación
Actor: todos los actores.
Precondición: el usuario debe haber iniciado sesión en la aplicación.
Actor: todos los actores.
Precondición: el usuario debe haber iniciado sesión en la aplicación.
Acción: funcionalidad que permite al usuario ver las notificaciones de los eventos de la aplicación.



**Tabla 9: Especificación del caso de uso: Cerrar Sesión.**

Caso de uso: CU 1.1.8 Cerrar Sesión
Actor: todos los actores.
Precondición: el usuario debe haber iniciado sesión en la aplicación.
Acción: funcionalidad que permite al usuario finalizar la sesión en la aplicación.
Condición de Éxito: se finalizará la sesión del usuario.

En la figura 1 se muestra el caso de uso para el actor Dirección en la aplicación web simbiosis, juntos con sus principales funcionalidades.



**Figura 1: Caso de uso de la aplicación Simbiosis. Actor Dirección.**

**Elaboración propia.**

**Tabla 10: Especificación del caso de uso: Opciones de Estudiante.**

Caso de uso: CU 1.1.9 Opciones de Estudiante
Actor: Director, Dirección.
Precondición: El usuario debe haber iniciado sesión en la aplicación.
Acción: funcionalidad que permite al usuario desplegar el menú de opciones para realizar distintas funcionalidades sobre estudiantes (agregar, eliminar, modificar, anular, listar)
Condición de Éxito: se desplegará el menú de opciones de estudiantes.

**Tabla 11: Especificación del caso de uso: Listar Estudiantes.**

Caso de uso: CU 1.1.9.1 Listar Estudiantes
Actor: Director, Dirección.
Precondición: El usuario debe haber iniciado sesión en la aplicación.
Acción: funcionalidad que permite al usuario listar los estudiantes registrados en la aplicación por departamento o por materia inscrita, pudiendo editar la información personal de los mismos.
Condición de Éxito: se desplegará el listado de estudiantes inscritos.

**Tabla 12: Especificación del caso de uso: Agregar Estudiantes.**

Caso de uso: CU 1.1.9.2 Agregar Estudiantes
Actor: Director, Dirección.
Precondición: El estudiante no debe existir en la BD de la aplicación.

**Tabla 13: Especificación del caso de uso: Eliminar Estudiantes.**

Caso de uso: CU 1.1.9.3 Eliminar Estudiantes
Actor: Director, Dirección.
Precondición: El estudiante debe existir en la BD de la aplicación.
Acción: funcionalidad que permite al usuario eliminar un estudiante en la BD de la aplicación web.
Condición de Éxito: se borrará el estudiante y su información en la BD de la aplicación web.

**Tabla 14: Especificación del caso de uso: Modificar Estudiantes.**

Caso de uso: CU 1.1.9.4 Modificar Formalización
Actor: Director, Dirección.
Precondición: El estudiante debe existir en la BD de la aplicación y debe haber realizado la formalización.
Acción: funcionalidad que permite al usuario modificar los datos de la formalización de la inscripción de un estudiante en la BD de la aplicación web.
Condición de Éxito: se actualizará la formalización del estudiante y su información en la BD de la aplicación web.

**Tabla 15: Especificación del caso de uso: Anular Formalización. Parte A.**

Caso de uso: CU 1.1.9.5 Anular Formalización
Actor: Director, Dirección.
Precondición: El estudiante debe existir en la BD de la aplicación.

**Tabla 16: Especificación del caso de uso: Anular Formalización. Parte B.**

Caso de uso: CU 1.1.9.5 Anular Formalización
Acción: funcionalidad que permite al usuario anular la formalización de la inscripción de un estudiante en la BD de la aplicación web.
Condición de Éxito: se eliminará la formalización hecha por el estudiante y su información en la BD de la aplicación web.

**Tabla 17: Especificación del caso de uso: Opciones de Profesor.**

Caso de uso: CU 1.1.10 Opciones de Profesor
Actor: Director, Dirección.
Precondición: El usuario debe haber iniciado sesión en la aplicación.
Acción: funcionalidad que permite al usuario desplegar el menú de opciones para realizar distintas funcionalidades sobre docentes (agregar, eliminar, listar)
Condición de Éxito: se desplegará el menú de opciones de estudiantes.

**Tabla 18: Especificación del caso de uso: Listar Profesores.**

Caso de uso: CU 1.1.10.1 Listar Profesor
Actor: Director, Dirección.
Precondición: El usuario debe haber iniciado sesión en la aplicación.
Acción: funcionalidad que permite al usuario listar los profesores registrados en la aplicación por tipo (Interno o externo) o por departamento, pudiendo editar la información personal de los mismos.

**Tabla 19: Especificación del caso de uso: Agregar Docente.**

Caso de uso: CU 1.1.10.2 Agregar Profesor
Actor: Director, Dirección.
Precondición: El docente no debe existir en la BD de la aplicación.
Acción: funcionalidad que permite al usuario agregar un docente en la BD de la aplicación web.
Condición de Éxito: se agregará el docente y su información en la BD de la aplicación web.

**Tabla 20: Especificación del caso de uso: Eliminar Docente.**

Caso de uso: CU 1.1.10.3 Eliminar Profesor
Actor: Director, Dirección.
Precondición: El docente debe existir en la BD de la aplicación.
Acción: funcionalidad que permite al usuario eliminar un docente en la BD de la aplicación web.
Condición de Éxito: se borrará el docente y su información en la BD de la aplicación web.

**Tabla 21: Especificación del caso de uso: Cargar Listados. Parte A.**

Caso de uso: CU 1.1.11 Cargar Listados
Actor: Director, Dirección.
Precondición: El usuario debe haber iniciado sesión en la aplicación.

**Tabla 22: Especificación del caso de uso: Cargar Listados. Parte B.**

Caso de uso: CU 1.1.11 Cargar Listados
Actor: Director, Dirección.
Precondición: El usuario debe haber iniciado sesión en la aplicación.
Acción: funcionalidad que permite al usuario desplegar el menú de opciones para realizar la carga de listados de docente o estudiantes.
Condición de Éxito: se desplegará el listado de estudiantes inscritos.

**Tabla 23: Especificación del caso de uso: Cargar Listado Estudiantes.**

Caso de uso: CU 1.1.11.1 Cargar Listado Estudiantes
Actor: Director, Dirección.
Precondición: El usuario debe haber iniciado sesión en la aplicación.
Acción: funcionalidad que permite al usuario cargar el listado de los estudiantes cursantes de las materias Seminario I, Seminario II y TEG.
Condición de Éxito: se cargará el listado en la BD de la aplicación web.

**Tabla 24: Especificación del caso de uso: Cargar Listado Profesores.**

Caso de uso: CU 1.1.11.2 Cargar Listado Profesores
Actor: Director, Dirección.
Precondición: El usuario debe haber iniciado sesión en la aplicación.
Acción: funcionalidad que permite al usuario cargar el listado de docentes internos y externos de la Escuela de Biología

**Tabla 25: Especificación del caso de uso: Generar Reportes.**

Caso de uso: CU 1.1.12 Generar Reportes
Actor: Director, Dirección.
Precondición: El usuario debe haber iniciado sesión en la aplicación.
Acción: funcionalidad que permite generar reportes de estudiantes y docentes registrados en la aplicación. Puede realizar reportes por departamento, materia o tipo de docente.
Condición de Éxito: se mostrará un archivo en formato pdf sobre el reporte generado.

**Tabla 26: Especificación del caso de uso: Enviar Comunicado.**

Caso de uso: CU 1.1.13 Enviar Comunicado
Actor: Director, Dirección, Representante UDPP, Jefe de Departamento.
Precondición: El usuario debe haber iniciado sesión en la aplicación.
Acción: funcionalidad que permite enviar comunicados vía correo electrónico a estudiantes y docentes registrados en la aplicación.
Condición de Éxito: se enviará el comunicado.

**Tabla 27: Especificación del caso de uso: Listar Planillas aprobadas.**

Caso de uso: CU 1.1.14.2 Listar Planillas Aprobadas
Actor: Director, Dirección.
Precondición: El usuario debe haber iniciado sesión en la aplicación.
Acción: funcionalidad que permite al usuario listar las planillas de formalizaciones que aún no han sido aprobadas.
Condición de Éxito: se desplegará el listado de planillas aprobadas.

**Tabla 29: Especificación del caso de uso: Iniciar/Detener Proceso.**

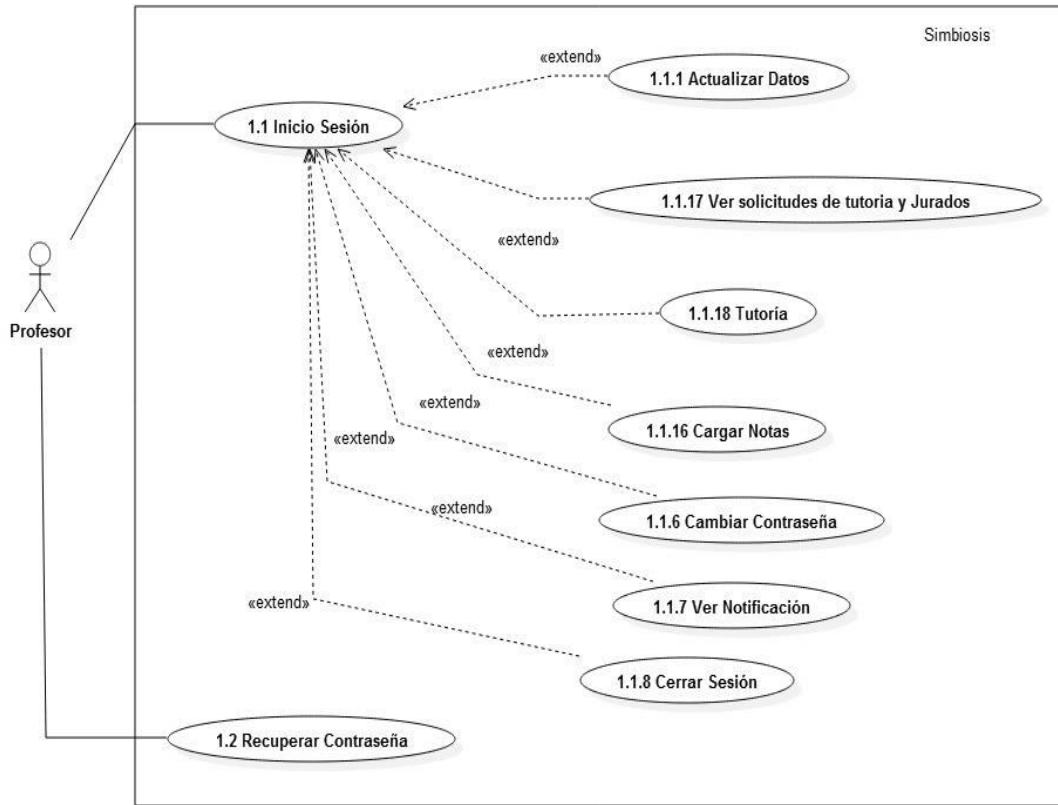
Caso de uso: CU 1.1.15.1 Iniciar/Detener Proceso
Actor: Director, Dirección.
Precondición: El usuario debe haber iniciado sesión en la aplicación.
Acción: funcionalidad que permite al usuario iniciar, pausar o detener el proceso de formalización de inscripciones.
Condición de Éxito: se aplicará la acción realizada en la aplicación (Detener, iniciar o finalizar el proceso).

**Tabla 30: Especificación del caso de uso: Configurar Departamentos.**

Caso de uso: CU 1.1.15.2 Configurar Departamentos
Actor: Director, Dirección.
Precondición: El usuario debe haber iniciado sesión en la aplicación.
Acción: funcionalidad que permite asignar roles a docentes.

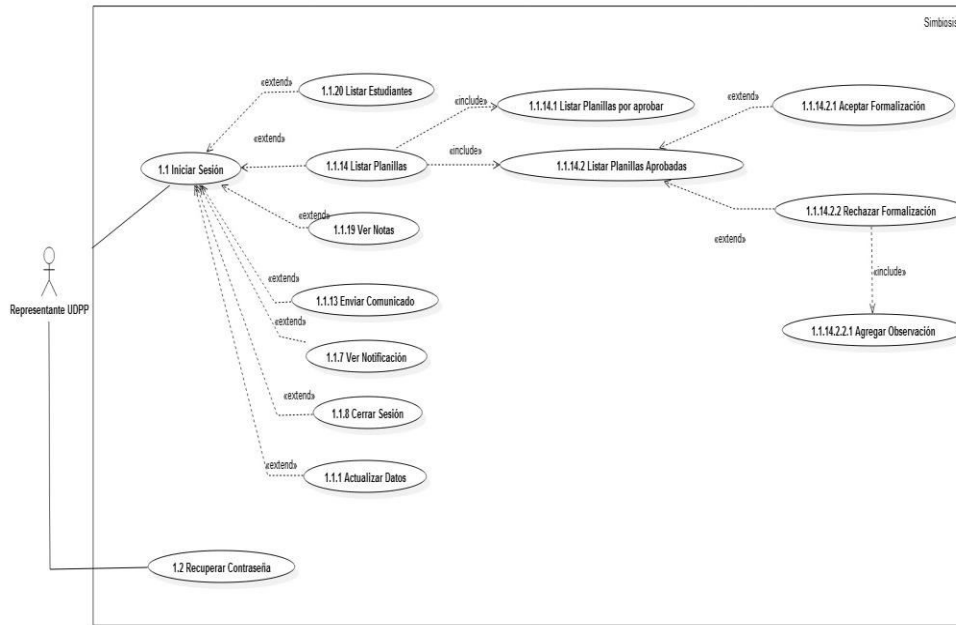
En la figura 2 se muestra el caso de uso del actor Profesor, el cual es desempeñado por cualquier docente perteneciente a la Escuela de Biología de la Facultad de Ciencias de la UCV. Muchas de las funcionalidades especificadas en los casos de uso anteriores son aplicables para los profesores, por lo que sólo se detallan aquellas que no han sido especificadas.



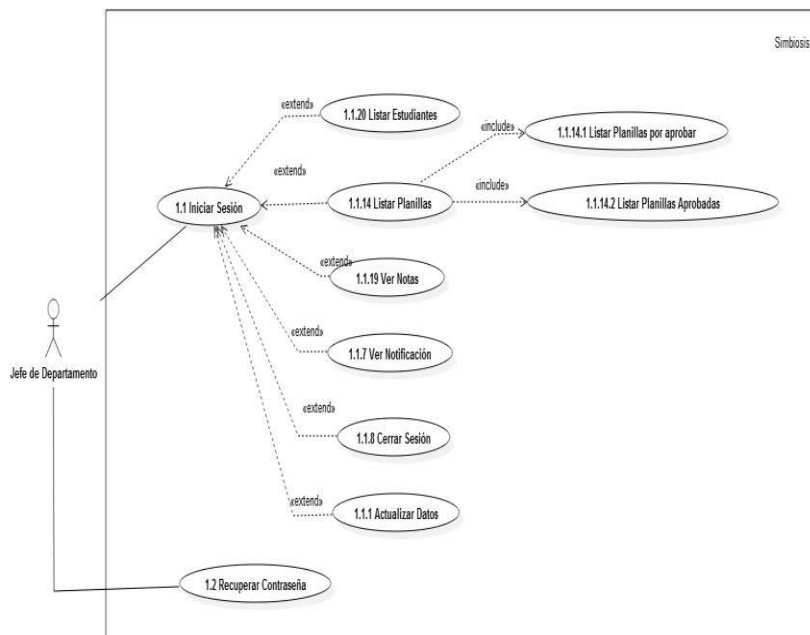


**Figura 2: Caso de uso de la aplicación Simbiosis. Profesor.**  
**Elaboración propia.**

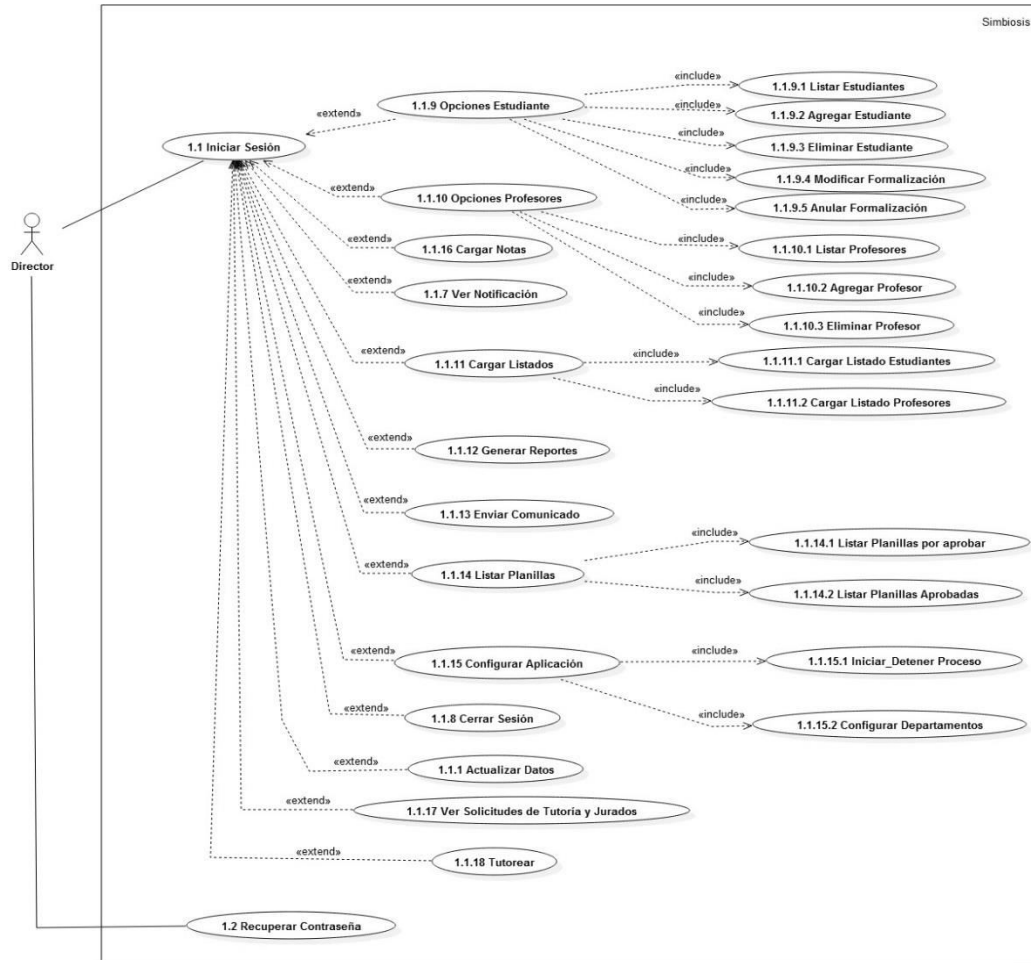
El actor Profesor puede desempeñar varios roles durante el proceso de formalización de inscripción, incluso puede desempeñar varios roles a la vez (tutor, jurado, jefe de departamento, entre otros). En la figura 3, figura 4 y figura 5 se muestran los distintos roles que puede desempeñar un docente, los cuales serán detallado más adelante.



**Figura 3: Caso de uso de la aplicación Simbiosis. Representante UDPP.**  
**Elaboración propia.**



**Figura 4: Caso de uso de la aplicación Simbiosis. Jefe de Departamento.**  
**Elaboración propia.**



**Figura 5: Caso de uso de la aplicación Simbiosis. Director.**  
**Elaboración propia.**

**Tabla 3.34: Especificación del caso de uso: Cargar Notas.**

Caso de uso: CU 1.1.16 Cargar Notas
Actor: Director, Profesor.
Precondición: El usuario debe haber iniciado sesión en la aplicación.
Acción: funcionalidad que permite al usuario cargar las notas obtenidas por un estudiante luego de haber presentado el trabajo inscrito.

**Tabla 3.35: Especificación del caso de uso: Ver solicitudes de Tutoría y Jurados.**

Caso de uso: CU 1.1.17 Ver solicitudes de Tutoría y Jurados
Actor: Profesor.
Precondición: El usuario debe haber iniciado sesión en la aplicación.
Acción: funcionalidad que permite al usuario ver aquellas solicitudes realizadas en la aplicación para ser tutor o jurado de un trabajo.
Condición de Éxito: se mostrará la notificación al docente.

## ANEXO B

### Cuestionario de Usabilidad

#### Cuestionario de Usabilidad de la Aplicación Web Simbiosis

El presente cuestionario tiene como finalidad conocer su opinión con respecto a la usabilidad de la Aplicación web Simbiosis. La cual se refiere a la efectividad, eficiencia y satisfacción con la que la aplicación puede ser comprendida, utilizada y ser atractiva al usuario. También se relaciona con la organización intencional de la aplicación. Se agradece seleccionar la puntuación que corresponda a su opinión de acuerdo a la siguiente escala:

Totalmente de acuerdo	De acuerdo	Ni de acuerdo ni en desacuerdo	En desacuerdo	Totalmente en desacuerdo
5	4	3	2	1

Nº	Item	5	4	3	2	1
1	Los títulos, menús y textos de la aplicación son legibles					
2	Los colores utilizados contribuyen a la percepción adecuada a la aplicación					
3	Los descriptores de la aplicación son claros y precisos					
4	Se ofrece una organización que facilita la navegación					
5	El diseño de la aplicación favorece su uso					
6	La aplicación se puede ejecutar de manera intuitiva					
7	Es una aplicación fácil de utilizar					
8	Los elementos de ayuda disponibles en la aplicación son útiles					

## ANEXO C

### Resultados de Pruebas de Usabilidad

<b>Marca Temporal</b>	<b>Item 1</b>	<b>Item 2</b>	<b>Item 3</b>	<b>Item 4</b>	<b>Item 5</b>	<b>Item 6</b>	<b>Item 7</b>	<b>Item 8</b>
<b>04/05/2015</b>	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente de acuerdo	De acuerdo	Totalmente de acuerdo	Totalmente de acuerdo	De acuerdo	Totalmente de acuerdo
<b>04/05/2015</b>	Totalmente de acuerdo	De acuerdo	De acuerdo	De acuerdo	Totalmente de acuerdo	De acuerdo	Totalmente de acuerdo	De acuerdo
<b>04/05/2015</b>	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente de acuerdo	De acuerdo	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente de acuerdo
<b>04/05/2015</b>	Totalmente de acuerdo	De acuerdo	De acuerdo	Totalmente de acuerdo	Totalmente de acuerdo	De acuerdo	De acuerdo	De acuerdo
<b>04/05/2015</b>	Totalmente de acuerdo	Totalmente de acuerdo	De acuerdo	Totalmente de acuerdo	De acuerdo	De acuerdo	Totalmente de acuerdo	Totalmente de acuerdo
<b>04/05/2015</b>	De acuerdo	Totalmente de acuerdo	Totalmente de acuerdo	De acuerdo	De acuerdo	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente de acuerdo
<b>04/05/2015</b>	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente de acuerdo	De acuerdo	De acuerdo	Totalmente de acuerdo	Totalmente de acuerdo
<b>04/05/2015</b>	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente de acuerdo
<b>04/05/2015</b>	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente de acuerdo	De acuerdo	De acuerdo	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente de acuerdo
<b>04/05/2015</b>	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente de acuerdo

## ANEXO D

### Manual Técnico de la Aplicación Web Simbiosis para la gestión de Inscripción de Seminarios y TEG.

El presente manual tiene el propósito de informar sobre las secciones y funcionalidades de la aplicación a nivel de código para el personal de soporte tecnológico. A continuación se muestran cada uno de los script que se utilizaron para la creación de la base de datos de la aplicación Simbiosis.

```
CREATE DATABASE IF NOT EXISTS `simbiosis` DEFAULT CHARACTER SET latin1
COLLATE latin1_swedish_ci;
USE `simbiosis`;

-- Estructura de tabla para la tabla `departamento`
--
CREATE TABLE IF NOT EXISTS `departamento` (
  `cod_departamento` int(11) NOT NULL AUTO_INCREMENT,
  `nombre` varchar(45) CHARACTER SET latin1 DEFAULTNULL,
  PRIMARY KEY(`cod_departamento`))
ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=6;

-- Volcado de datos para la tabla `departamento`

INSERT INTO `departamento` (`cod_departamento`, `nombre`) VALUES
  (1, 'BIOLOGIA CELULAR'),
  (2, 'BOTANICA'),
  (3, 'ECOLOGIA'),
  (4, 'TECNOLOGIA DE ALIMENTOS'),
  (5, 'ZOOLOGIA');

CREATE TABLE IF NOT EXISTS `documento` (`cod_documento` varchar(100) NOT
NULL, `departamento` varchar(50) NOT NULL,
  `titulo` varchar(500) CHARACTER SET latin1 DEFAULTNULL,
  `obj_general` varchar(500) CHARACTER SET latin1 DEFAULTNULL, `nombre_doc`
varchar(100) CHARACTER SET latin1 DEFAULTNULL, `ruta_doc` varchar(150)
CHARACTER SET latin1 DEFAULTNULL, `cod_tipodocumento`
enum('seminario1', 'seminario2', 'teg', 'seminario_teg') CHARACTER SET latin1
DEFAULTNULL,
  PRIMARY KEY(`cod_documento`),
  KEY `cod_tipodocumento_idx` (`cod_tipodocumento`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```

-- Estructura de tabla para la tabla `observacion_documento`
--

CREATE TABLE IF NOT EXISTS `observacion_documento` (
  `cod_profesor` int(11) NOT NULL,
  `cod_documento` varchar(100) NOT NULL,
  `observacion` varchar(500) CHARACTER SET latin1 NOT NULL,
  `fecha` date NOT NULL,
  PRIMARY KEY(`cod_profesor`,`cod_documento`,`observacion`,`fecha`),
  KEY `cod_documento_obs` (`cod_documento`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Estructura de tabla para la tabla `profesor`
--

CREATE TABLE IF NOT EXISTS `profesor` (
  `cod_profesor` int(11) NOT NULL AUTO_INCREMENT,
  `ci_profesor` varchar(60) CHARACTER SET latin1 DEFAULTNULL,
  `nombre` varchar(120) CHARACTER SET latin1 DEFAULTNULL,
  `telefono` varchar(60) CHARACTER SET latin1 DEFAULTNULL,
  `telefono2` varchar(60) CHARACTER SET latin1 DEFAULTNULL,
  `correo` varchar(50) CHARACTER SET latin1 DEFAULTNULL,
  `correo2` varchar(50) CHARACTER SET latin1 DEFAULTNULL,
  `cod_usuario` int(11)DEFAULTNULL,
  `estado` enum('ACTIVO','INACTIVO') CHARACTER SET latin1 NOT NULL,
  `cod_departamento` int(11)DEFAULTNULL,
  `cod_tipoprofesor` enum('INTERNO','EXTERNO') CHARACTER SET latin1
  DEFAULTNULL,
  PRIMARY KEY(`cod_profesor`),
  KEY `cod_usuario_idx` (`cod_usuario`),
  KEY `cod_departamento_idx` (`cod_departamento`),
  KEY `cod_tipoprofesor_idx` (`cod_tipoprofesor`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=80852309;
--

-- Estructura de tabla para la tabla `rol_profesores`

CREATE TABLE IF NOT EXISTS `rol_profesores` (
  `cod_profesor` int(11) NOT NULL,
  `rol_profesor`
  enum('tutor','jurado_p','jurado_s','cotutor','director','jefe_departame
  nto','udpp','asesor_academico') NOT NULL,
  `cod_estudiante` int(11) NOT NULL,
  `fecha` varchar(100) CHARACTER SET latin1 NOT NULL,
  PRIMARY KEY(`cod_profesor`,`rol_profesor`,`cod_estudiante`,`fecha`),
  KEY `cod_profesor_idx` (`cod_profesor`),
  KEY `rol_profesor_idx` (`rol_profesor`),
  KEY `cod_estudiante_idx` (`cod_estudiante`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```



```

-- Estructura de tabla para la tabla `tabla_rol`
CREATE TABLE IF NOT EXISTS `tabla_rol` (
  `rol_profesor`
enum('tutor','jurado_p','jurado_s','cotutor','director','jefe_departam
ento','udpp','asesor_academico') CHARACTER SET latin1 NOT NULL,
  PRIMARY KEY(`rol_profesor`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Volcado de datos para la tabla `tabla_rol`
INSERT INTO `tabla_rol` (`rol_profesor`) VALUES
('tutor'),('jurado_p'),('jurado_s'),('cotutor'),
('director'),('jefe_departamento'),('udpp'),
('asesor_academico');

--
-- Estructura de tabla para la tabla `tipo_documento`
CREATE TABLE IF NOT EXISTS `tipo_documento` (
  `cod_tipodocumento`
enum('seminario1','seminario2','teg','seminario_teg') CHARACTER SET
latin1 NOT NULL,
  PRIMARY KEY(`cod_tipodocumento`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Volcado de datos para la tabla `tipo_documento`
INSERT INTO `tipo_documento` (`cod_tipodocumento`) VALUES
('seminario1'),('seminario2'),('teg'),('seminario_teg');

-- Estructura de tabla para la tabla `tipo_profesor`
--
CREATE TABLE IF NOT EXISTS `tipo_profesor` (
  `cod_tipoprofesor` enum('INTERNO','EXTERNO','','') CHARACTER SET
latin1 NOT NULL,
  PRIMARY KEY(`cod_tipoprofesor`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;


-- Estructura de tabla para la tabla `usuario`
CREATE TABLE IF NOT EXISTS `usuario` (
  `cod_usuario` int(11) NOT NULL AUTO_INCREMENT,
  `usuario` varchar(60) CHARACTER SET latin1 DEFAULTNULL,
  `contrasena` char(128) CHARACTER SET latin1 DEFAULTNULL,
  `privilegios` enum('ESTUDIANTE','DIRECCION','PROFESOR') CHARACTER SET
latin1 DEFAULTNULL,
  `salt` char(128) CHARACTER SET latin1 DEFAULTNULL)

```

Para el desarrollo de la aplicación fue necesario el construir un conjunto de tablas mediante MySQL 5.5.1, las cuales conforman la lógica que hace posible el funcionamiento y la persistencia de la aplicación. Dichas tablas serán explicadas a continuación:

- Usuario: en esta tabla se almacena la información de los usuarios registrados y que tienen acceso al sistema, así como sus características principales.
- Estudiante: Almacena todos los estudiantes inscritos en Seminario I y II y Trabajo Especial de Grado de la Escuela de Biología.
- Profesor: Almacena todos los profesores de la Escuela de Biología (internos y externos) así como su información.
- Inscribe: esta tabla almacena información acerca de la inscripción de un estudiante, permite establecer la relación entre el estudiante y el documento inscrito.
- Documento\_inscrito: almacenara los datos ingresados por el estudiante relacionados al documento y además la ruta donde se almacenarán en el servidor.
- Tipo\_documento: almacena el tipo de documento inscrito: Seminario I, Seminario II o Especial de Grado.
- Departamento: almacena el tipo de departamento, son 5 en total: botánica, zoología, biología celular, tecnología de alimentos, ecología.
- Notificaciones: en esta tabla se almacenan las notificaciones del sistema que son enviadas a los usuarios por cada evento que ocurre en el mismo.
- Observaciones: Almacena la información de cada observación referente al proceso de formalización de un estudiante.
- Tabla\_Roles: almacena los roles que puede ejercer un profesor durante el proceso de formalización en un semestre.
- Proceso: Esta tabla almacena información acerca del estado actual del proceso de formalización, el cual puede estar detenido, iniciado o finalizado.
- Histórico de Notas: Almacena la calificación obtenida por un estudiante luego de haber sido evaluado durante el semestre en las asignaturas Seminario I y II y TEG.
- Intentos\_login: Almacena el histórico de intentos fallidos de login, cuando se sobrepasa un valor establecido, automáticamente se prohíbe el acceso al usuario al sistema.

En la siguiente figura se muestra en detalle la versión y configuración de PHP que tiene el servidor para el correcto funcionamiento de la aplicación.

PHP Version 5.4.16	
	
<b>System</b>	Windows NT FRANCISCO-PC 6.1 build 7601 (Windows 7 Business Edition Service Pack 1) i586
<b>Build Date</b>	Jun 5 2013 20:58:05
<b>Compiler</b>	MSVC9 (Visual C++ 2008)
<b>Architecture</b>	x86
<b>Configure Command</b>	cscript /nologo configure.js "--enable-snapshot-build"--disable-isapi"--enable-debug-pack"--without-mssql"--without-pdo-mssql"--without-pi3web"--with-pdo-oci=C:\php-sdk\oracle\instantclient10\sdk,shared"--with-oci8=C:\php-sdk\oracle\instantclient10\sdk,shared"--with-oci8-11g=C:\php-sdk\oracle\instantclient11\sdk,shared"--enable-object-out-dir=../obj"--enable-com-dotnet=shared"--with-mcrypt=static"--disable-static-analyze"--with-pgo"
<b>Server API</b>	Apache 2.0 Handler
<b>Virtual Directory Support</b>	enabled
<b>Configuration File (php.ini) Path</b>	C:\Windows
<b>Loaded Configuration File</b>	C:\wamp\bin\apache\apache2.4.4\bin\php.ini
<b>Scan this dir for additional .ini files</b>	(none)
<b>Additional .ini files parsed</b>	(none)
<b>PHP API</b>	20100412
<b>PHP Extension</b>	20100525
<b>Zend Extension</b>	220100525
<b>Zend Extension Build</b>	API220100525,TS,VC9

## Configuration apache2handler

<b>Apache Version</b>	Apache/2.4.4 (Win32) PHP/5.4.16
<b>Apache API Version</b>	20120211
<b>Server Administrator</b>	admin@example.com
<b>Hostname:Port</b>	localhost:0
<b>Max Requests</b>	Per Child: 0 - Keep Alive: on - Max Per Connection: 100
<b>Timeouts</b>	Connection: 60 - Keep-Alive: 5
<b>Virtual Server</b>	No
<b>Server Root</b>	C:\wamp\bin\apache\Apache2.4.4
<b>Loaded Modules</b>	core mod_win32 mpm_winnt http_core mod_so mod_access_compat mod_actions mod_alias mod_allowmethods mod_asis mod_auth_basic mod_authn_core mod_authn_file mod_authz_core mod_authz_groupfile mod_authz_host mod_authz_user mod_autoindex mod_cgi mod_dir mod_env mod_include mod_isapi mod_log_config mod_mime mod_negotiation mod_rewrite mod_setenvif mod_php5

Directive	Local Value	Master Value
engine	1	1
last_modified	0	0
xbithack	0	0

De la configuración anterior es importante y fundamental tener activo el módulo `mod_rewrite` en el servidor Apache. Ya que es el encargado de permite crear direcciones URL alternativas a las dinámicas generadas por la programación de la aplicación web Simbiosis, de tal modo que sean más legibles y fáciles de recordar.

En caso de no tener privilegios para activar el módulo `mod_rewrite` se puede simular mediante el siguiente script que se encarga de virtualizar dicho módulo.

```
<?php
$uri=parse_url($_SERVER['REQUEST_URI'], PHP_URL_PATH);
$uri=urldecode($uri);
$paths=require __DIR__.'/bootstrap/paths.php';
$requested=$paths['public'].$uri;

// This file allows us to emulate Apache's "mod_rewrite"
// functionality from the
// built-in PHP web server. This provides a convenient way to
// test a
// application without having installed a "real" web server
// software here.
if($uri!== '/' and file_exists($requested))
{
returnfalse;
}

require_once$paths['public'].'/index.php';
```

Sin embargo, este archivo únicamente debe ser utilizado con fines de desarrollo, nunca debe ser usado en un ambiente de producción.

El script más importante es el encargado de permitir que el usuario inicie sesión en el sistema y sea redirigido al perfil correspondiente. A continuación se muestra el código del script `login.php`.

```

<?php /*script para iniciar session*/
@session_start();
if(empty($_SESSION['captcha'])||strcasecmp($_SESSION['captcha'],strto
lower(trim($_POST['captcha'])))!=0){
header("Location: ../../views/error/error.php?login=captcha");
}else{//else captcha
if($_SERVER["REQUEST_METHOD"]=="POST"){//solicitud post
if(isset($_POST["password"])&&isset($_POST["cedula"])){//si se envia
el usuario y la cedula
$username=$_POST['cedula'];
$password=$_POST['password'];
include("../DataBase/acceso_db_2.php");
$saltQuery="SELECT * FROM usuario WHERE usuario= '$username'";
$result=$mysqli->query($saltQuery) or die($mysqli->error.__LINE__);
$row=$result->fetch_assoc();
$salt=$row['salt'];
$usuario=$row['usuario'];
$privilegios=$row['privilegios'];
$saltedPassword=$password.$salt;
$hashedPassword=hash('sha256',$saltedPassword);
$query_s="SELECT * FROM usuario WHERE (usuario ='".$username."' ) AND
(contrasena ='".$hashedPassword."' )";
$login=$mysqli2->query($query_s) or die($mysqli2->error.__LINE__);
$row_cnt=$login->num_rows;
if($row_cnt>0){
@session_start();
$row=$login->fetch_array(MYSQLI_ASSOC);
$cedula=$row['usuario'];
$clave=$row['contrasena'];
$privilegios=$row['privilegios'];
$salt=$row['salt'];
$saltedCedula=$cedula.$salt;
$hashedCedula=hash('sha256',$saltedCedula);
if(strcmp($hashedCedula,$clave)==0){
$_SESSION["autenticado"]="SI";
$_SESSION['usuario']=$cedula;
$_SESSION['perfil']=$privilegios;
$_SESSION['contraseña']=$clave;
$_SESSION['salt']=$salt;
header("Location: ("Location: ../../views/$privilegios /index.php");}
} }
?>

```

## ANEXO E

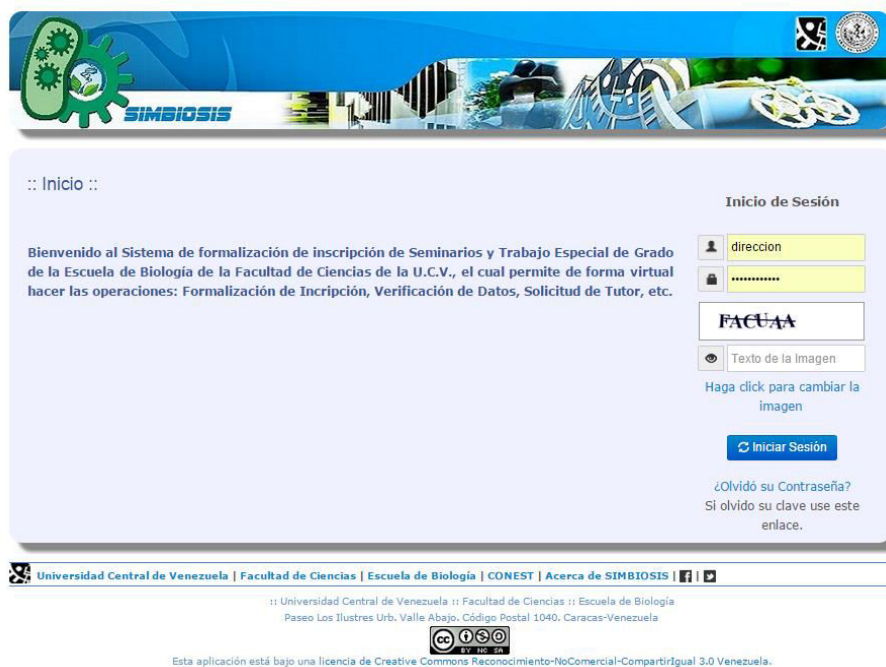
### Manual de Usuario de la Aplicación Web para la gestión del proceso de Inscripción de Seminarios y TEG.

Sin embargo, este archivo únicamente debe ser utilizado con fines de desarrollo, nunca debe ser usado en un ambiente de producción.

El presente manual tiene el propósito de informar sobre las secciones y funcionalidades de la aplicación a nivel administrativo. En este manual encontrará:

1. Guía para configurar la aplicación web Simbiosis.
2. Guía para cargar un listado de estudiantes en formato Excel.
3. Guía para generar reportes.

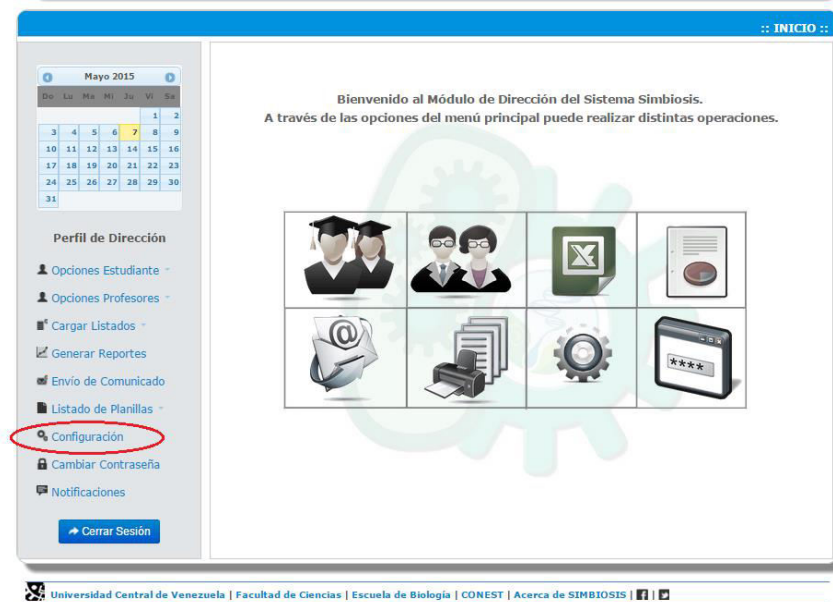
Para ingresar en la aplicación web Simbiosis debe ingresar con mediante un navegador en la siguiente dirección: [www.ciens.ucv.ve/simbiosis/views](http://www.ciens.ucv.ve/simbiosis/views), lo cual mostrará la página principal de la aplicación, donde debe colocar el usuario y contraseña suministrados para el perfil de dirección, tal como se muestra en la figura 1.



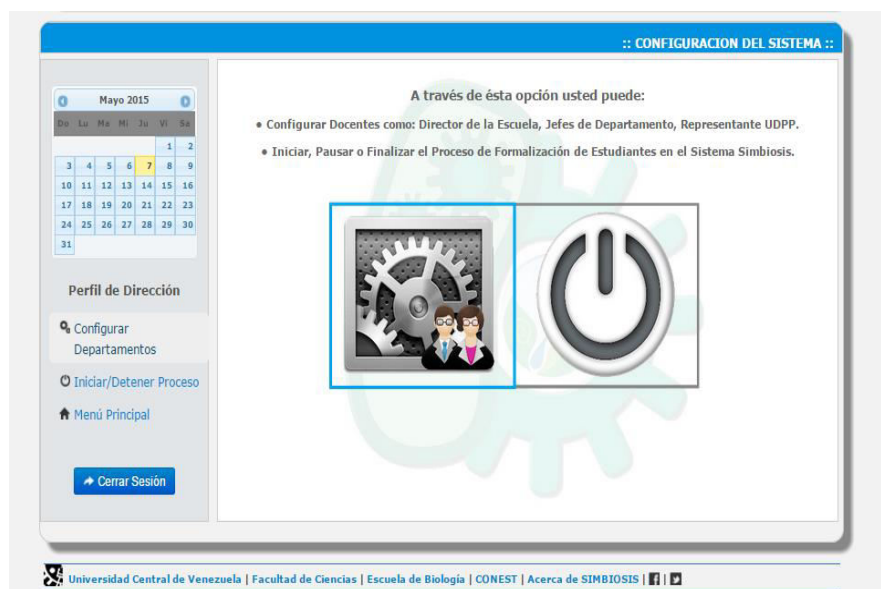
**Figura 1. Página principal de la aplicación Simbiosis.**

▪ **Pasos para configurar la aplicación web Simbiosis.**

Luego de haber ingresado en la aplicación, se mostrará el menú para el perfil administrativo de dirección, haciendo clic en la opción Configuración mostrado en la figura 2, se mostrara una nueva interfaz para configurar la aplicación, tal como se muestra en la figura 4.



**Figura 2. Menú Principal del perfil dirección.**



**Figura 3. Menú de configuración de la aplicación Simbiosis.**

Una vez en el menú de configuración se debe hacer clic en la opción de Iniciar/Detener proceso, lo cual mostrará un botón para iniciar el mismo, tal como se muestra en la figura 4, es importante indicar que si el proceso no ha sido iniciado se mostrará un botón de color verde con la palabra “Iniciar Proceso”, mientras que si ya está iniciado mostrará dos (2) botones con las palabras “detener proceso” y “finalizar proceso”, tal como se muestra en la figura 4 y figura 5.



**Figura 4. Botón para Inicia el proceso de formalización.**



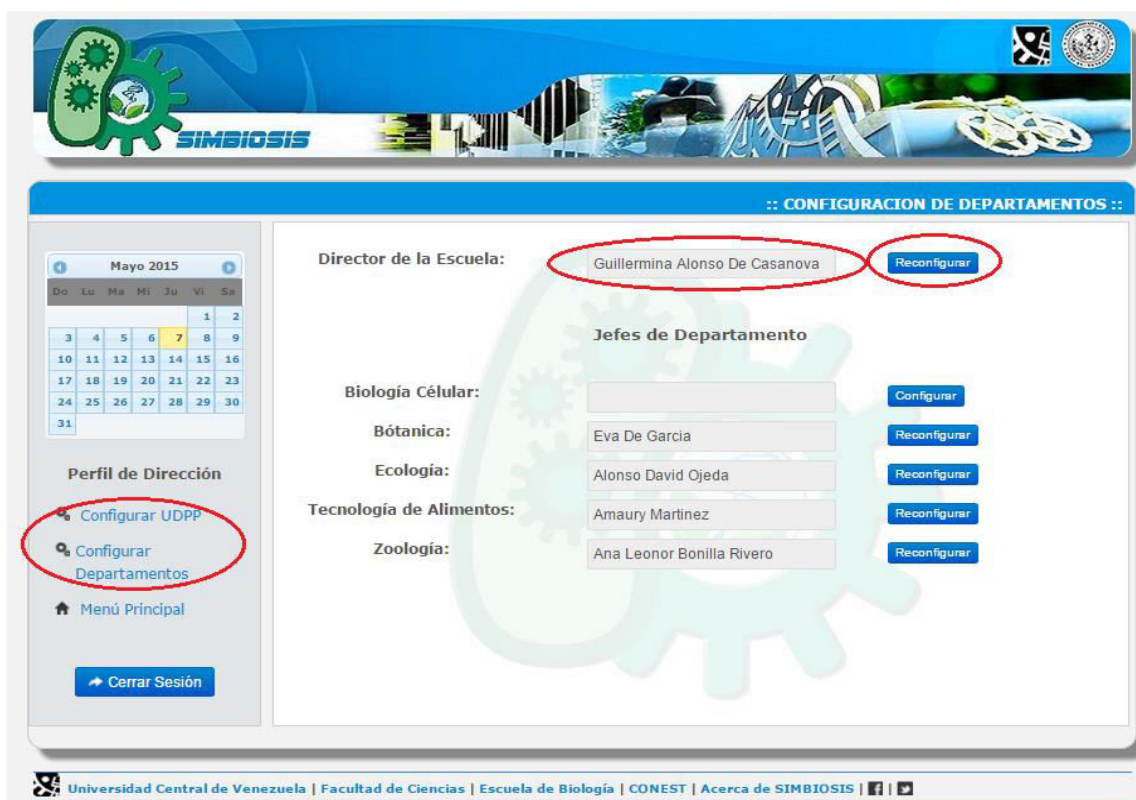
**Figura 5. Botones para detener y finalizar el proceso de formalización.**

Al hacer clic en los botones se mostrará un cuadro de dialogo para confirmar que efectivamente se quiere realizar la acción solicitada. Posteriormente, para asignar los roles a

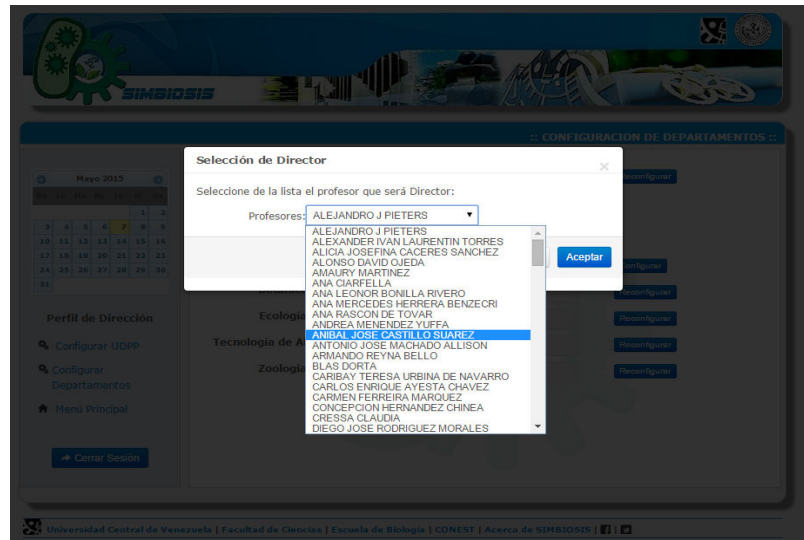


docentes como: Director, Jefes de Departamentos y Representantes de la Unidad Docente de Pasantías y Prácticas Profesionales (UDPP), se debe hacer clic en la opción Configurar Departamentos, lo cual lleva al usuario a la pantalla mostrada en la figura 6. En dicha interfaz aparece un menú para escoger entre si se desea configurar a los jefes de departamento o a los representantes de la UDPP.

Para asignar un rol a un docente, se debe hacer clic en el botón configurar o reconfigurar, y seleccionar de entre una lista desplegable el tutor que ocupara dicho cargo, tal como se muestra en la figura 7.



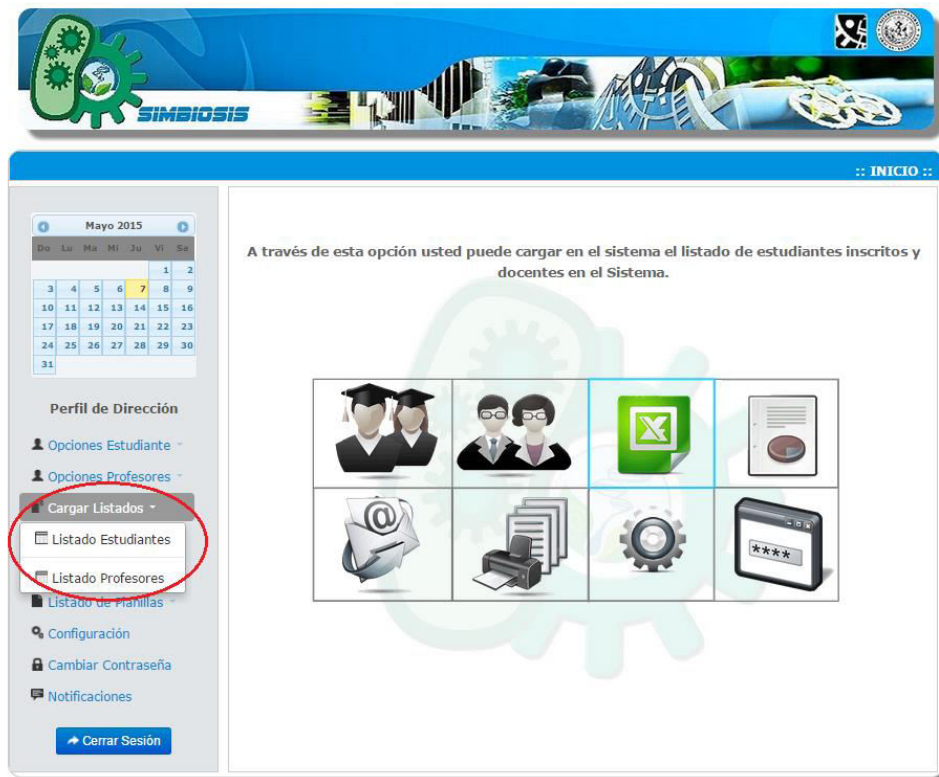
**Figura 7. Interfaz para configurar Director, Jefes de Departamento y Representantes de la UDPP.**



**Figura 7. Lista desplegable para escoger al docente.**

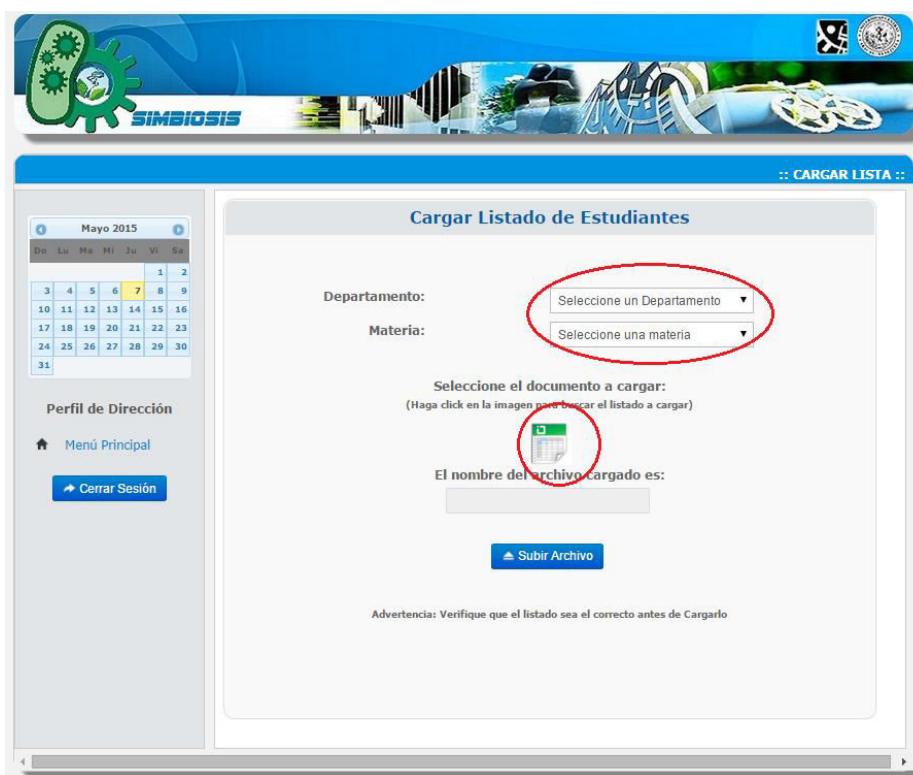
- **Pasos para cargar un listado de estudiantes en formato Excel:**

Para cargar un listado de estudiantes o docentes en la aplicación Simbiosis, se debe hacer clic en la opción Cargar Listados, la cual contiene dos opciones: Listado de Estudiantes y Listado de Profesores (figura 8).



**Figura 8. Opción del menú para cargar listados.**

Para cargar el listado de estudiantes en la aplicación se debe hacer clic en la opción Listado Estudiantes, que muestra la vista de la figura 9, en la cual debemos escoger el departamento y la materia, lo cual debe coincidir con la información del archivo Excel seleccionado. Para seleccionar el archivo se debe hacer clic en el ícono de Excel y se desplegará una ventana donde se podrá ubicar el archivo en el equipo.



**Figura 9. Interfaz para cargar un listado de estudiantes.**

El archivo a ser cargado debe estar en formato xlsx (Excel 2003 en adelante), ya que este es el formato con el que se descarga directamente desde CONEST. Asimismo, el archivo debe cargarse respetando el formato original que se muestra en la figura 10.

	A	B	C	D	E	F
1	FACULTAD DE CIENCIAS					
2	DIVISION DE CONTROL DE ESTUDIOS					
3	PERIODO: 01-2013					
4	MATERIA: [1491] SEMINARIO II (CELULAR)					
5	SECCION: B1					
6						
7	<b>Nro</b>	<b>LICENCIATURA</b>	<b>CEDULA</b>	<b>NOMBRE</b>	<b>CORREO</b>	<b>ESTADO</b>
8	1	BIOLOGIA	19693507	ACEVEDO DA SILVA JAVIER EDUARDO	jeadas@gmail.com	
9	2	BIOLOGIA	20317720	FERNANDEZ SALVATIERRA DAVID JOSE	salvat_mund@hotmail.com	
10	3	BIOLOGIA	18738935	GARIBALDI SOSA LISETH GERALDINE	liseth_th@hotmail.com	
11	4	BIOLOGIA	20190783	GOMEZ DI GIACINTO YONATHAN	yonathangomezdig@hotmail.com	
12	5	BIOLOGIA	17775536	LOPEZ PACHECO ELIANNY MADELINE	eliannymadeline@gmail.com	
13	6	BIOLOGIA	16870303	MARQUEZ REYES KRUKZAYA JOSE MILAGROS	kruzkaya.marquez@gmail.com	
14	7	BIOLOGIA	18112818	RAMIREZ QUINTERO KERLY DEL VALLE	kerly_ucv@hotmail.com	
15	8	BIOLOGIA	19291008	SANCHEZ GONZALEZ ANDREINA GABRIELA	andre_otaku_lo@hotmail.com	
16	9	BIOLOGIA	18467179	TRIVINO PARTIDA JENNIFER JOSEFINA	jennifit@hotmail.com	
17	10	BIOLOGIA	18818121	VIVAS VARELA WOLFGANG EDUARDO DAVID TAD	wolfgang.vivas.v@gmail.com	
18						
19						
20	<b>Horario</b>					
21	<b>DIA</b>	<b>HORA</b>	<b>AULA</b>			
22						
23						

**Figura 10. Formato del archivo Excel descargado de CONEST.**

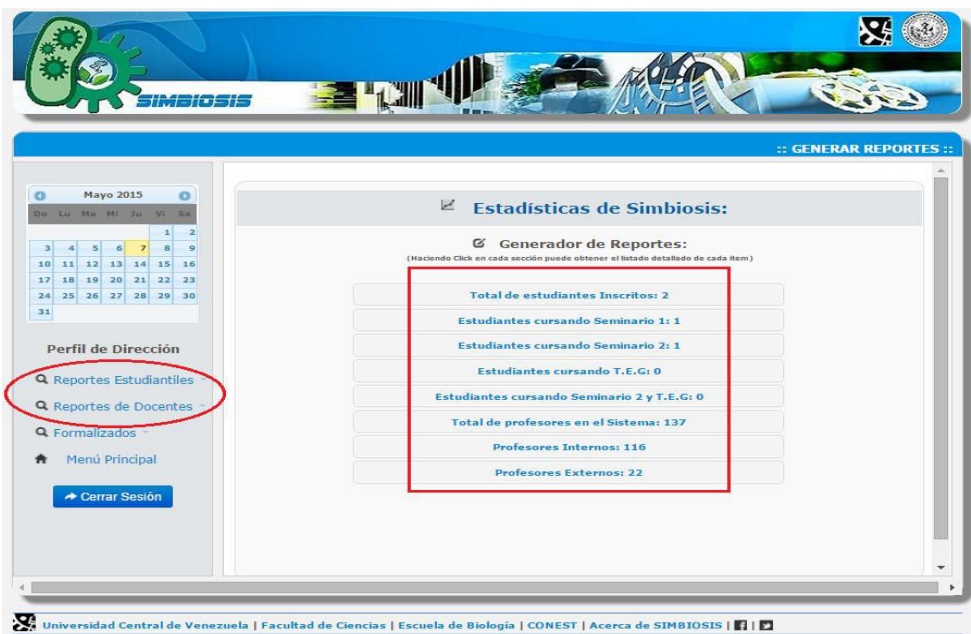
- **Pasos para generar reportes.**

Para generar reportes en formato pdf mediante la aplicación web Simbiosis, se debe ingresar en la opción “Generar Reportes” del Menú Principal (figura 11).



**Figura 11. Opción del menú para generar reportes.**

Al hacer clic se muestra una nueva interfaz (figura 12) en la que se tiene un menú de opciones para generar reportes y reportes predefinidos por la aplicación.



**Figura 12. Interfaz para generar reportes.**

Los reportes generados son listas como la que se muestra a continuación, los cuales pueden ser exportados en formato pdf para almacenarlo en el computador.

UNIVERSIDAD CENTRAL DE VENEZUELA  
 FACULTAD DE CIENCIAS  
 ESCUELA DE BIOLOGIA  
 DIRECCION DE LA ESCUELA  
 SISTEMA SIMBIOSIS  
 LISTADO TOTAL DE PROFESORES  
 TOTAL PROFESORES: 138



Thursday 07 de May del 2015

Nro	Cedula	Nombre	Telefono	Telefono	Email	Departamento	Entidad	Estado
1	5299518	ALEJANDRO J PIETERS			javiervergara2004@gmail.com	Docente Externo	EXTERNO	ACTIVO
2	9565764	ALEXANDER IVAN LAURENTIN TORRES	0212-7751011		javiervergara2004@gmail.com	BIOLOGIA CELULAR	INTERNO	ACTIVO
3	4361203	ALICIA JOSEFINA CACERES SANCHEZ	0212-7510544		javiervergara2004@gmail.com	BOTANICA	INTERNO	ACTIVO
4	6364695	ALONSO DAVID OJEDA	0212-6051090		javiervergara2004@gmail.com	ECOLOGIA	INTERNO	ACTIVO
5	3500600	AMAURY MARTINEZ	0212-7533871		javiervergara2004@gmail.com	TECNOLOGIA DE ALIMENTOS	INTERNO	ACTIVO
6	8315404	ANA CIARFELLA			javiervergara2004@gmail.com	Docente Externo	EXTERNO	ACTIVO
7	5073256	ANA LEONOR BONILLA RIVERO	0212-6051309		javiervergara2004@gmail.com	ZOOLOGIA	INTERNO	ACTIVO
8	37713907	ANA MERCEDES HERRERA BENZECRI	0212-7510544		javiervergara2004@gmail.com	BOTANICA	INTERNO	ACTIVO
9	2766482	ANA RASCON DE TOVAR	0212-7510111	4126625515	javiervergara2004@gmail.com	BIOLOGIA CELULAR	INTERNO	ACTIVO
10	6239224	ANDREA MENENDEZ YUFFA	0212-7510111		javiervergara2004@gmail.com	BOTANICA	INTERNO	ACTIVO
11	3400571	ANIBAL JOSE CASTILLO SUAREZ	0212-7535897		javiervergara2004@gmail.com	BOTANICA	INTERNO	ACTIVO
12	3139563	ANTONIO JOSE MACHADO ALLISON	0212-6051534		javiervergara2004@gmail.com	ECOLOGIA	INTERNO	ACTIVO
13	6563449	ARMANDO REYNA BELLO			javiervergara2004@gmail.com	Docente Externo	EXTERNO	ACTIVO
14	6160230	BLAS DORTA	0212-7535897	0424-1504652	javiervergara2004@gmail.com	BIOLOGIA CELULAR	INTERNO	ACTIVO
15	3973369	CARIBAY TERESA URBINA DE NAVARRO	0212-6051036		javiervergara2004@gmail.com	ECOLOGIA	INTERNO	ACTIVO
16	3182096	CARLOS ENRIQUE AYESTA CHAVEZ	0212-6051259		javiervergara2004@gmail.com	ECOLOGIA	INTERNO	ACTIVO
17	4369157	CARMEN FERREIRA MARQUEZ	0212-6051301		javiervergara2004@gmail.com	ZOOLOGIA	INTERNO	ACTIVO
18	5073799	CONCEPCION HERNANDEZ CHINEA	0212-7510111		javiervergara2004@gmail.com	BIOLOGIA CELULAR	INTERNO	ACTIVO

**Figura 13. Listado generado mediante la funcionalidad de generación de reportes.**