



**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**  
**Facultad de Ingeniería en Electricidad y Computación**

“DESARROLLO DE UNA APLICACIÓN WEB PARA LA GESTIÓN  
DE ACTIVIDADES DE ALUMNOS INVOLUCRADOS EN  
PROYECTOS DE VINCULACIÓN CON LA SOCIEDAD”

**INFORME DE MATERIA INTEGRADORA**

Previo a la obtención del Título de:

**INGENIERO EN CIENCIAS COMPUTACIONALES**  
**ORIENTACIÓN SISTEMAS MULTIMEDIA**

JOSÉ ANDRÉS ROMERO TRIVIÑO

JOSÉ LUIS MONAR JAIME

GUAYAQUIL – ECUADOR

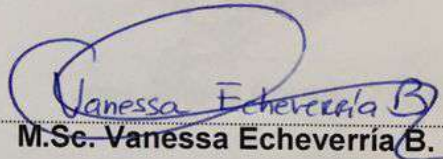
AÑO: 2016

## TRIBUNAL DE EVALUACIÓN



Ph. D. Dennis Romero L.

PROFESOR EVALUADOR

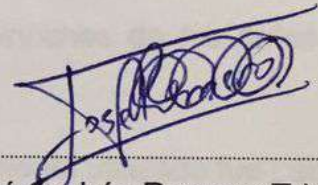


M.Sc. Vanessa Echeverría B.

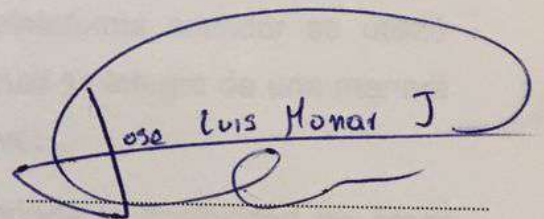
PROFESOR EVALUADOR

## DECLARACIÓN EXPRESA

"La responsabilidad y la autoría del contenido de este Trabajo de Titulación, nos corresponde exclusivamente; y damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"



.....  
José Andrés Romero Triviño



.....  
José Luis Monar Jaime

## RESUMEN

El presente proyecto tiene como principal problema a resolver, la inexistencia de una plataforma que facilite la administración y seguimiento de actividades en proyectos relacionados con vínculos con la sociedad. El ciclo de vida de un proyecto, involucra a diferentes tipos de usuarios, quienes tendrán que llenar diversos formularios a lo largo del proyecto, y ser enviados a revisión.

Para resolver este problema, se optó por la implementación de una aplicación web que permita agilizar este proceso, de tal manera que, cada usuario pueda llenar sus formularios usando el sistema, ayudando así, a disminuir el tiempo dedicado a este proceso, ya que la plataforma genera automáticamente muchos de los datos que tendrían que ser ingresados manualmente.

Para la implementación de este sistema, la principal herramienta utilizada fue Laravel, framework que permite desarrollar aplicaciones web, en lenguaje PHP, de una manera rápida y confiable. Adicionalmente, como plataforma servidor se utilizó Apache, y como motor de base de datos, MySQL, la cual se integra de una manera fácil y eficiente con aplicaciones desarrolladas en Laravel.

Como resultado, se tiene una aplicación web que brinda la posibilidad de crear proyectos de una manera rápida e intuitiva, así como también la administración del equipo del proyecto, director, tutores y estudiantes, para luego continuar con el proceso de asignación de actividades a los estudiantes, y el respectivo seguimiento de dichas actividades. El sistema también ofrece opciones tales como: asignación de roles a usuarios, notificaciones de eventos a lo largo del ciclo de vida del proyecto, generación y revisión de formularios.

## ÍNDICE GENERAL

TRIBUNAL DE EVALUACIÓN .....	ii
DECLARACIÓN EXPRESA .....	iii
RESUMEN.....	iv
ÍNDICE GENERAL .....	v
CAPÍTULO 1.....	1
1. DESCRIPCIÓN DEL PROBLEMA A RESOLVER. ....	1
1.1 Causas .....	2
1.2 Efectos .....	2
1.3 Trabajos Relacionados.....	3
1.3.1 Nuxeo.....	3
1.3.2 Alfresco .....	3
1.3.3 Ecount .....	4
CAPÍTULO 2.....	5
2. DISEÑO E IMPLEMENTACIÓN DE LA APLICACIÓN WEB.....	5
2.1 Objetivos del desarrollo de la aplicación web .....	5
2.1.1 Objetivo general .....	5
2.1.2 Objetivos específicos.....	5
2.2 Tecnologías y Herramientas Utilizadas.....	6
2.2.1 Laravel.....	7
2.2.2 MySQL .....	8
2.2.3 PhpMyAdmin .....	8
2.2.4 Servidor HTTP Apache.....	9
2.3 Diseño de la aplicación web .....	10
2.3.1 Diseño UML.....	10
2.3.2 Diseño de la base de datos .....	17
2.4 Implementación .....	24
2.4.1 Flujo de trabajo.....	24

2.4.2	Servicios Web de ESPOL.....	26
2.4.3	Desarrollo de formularios.....	27
2.4.4	Funcionalidades del sistema.....	33
CAPÍTULO 3.....		42
3.	RESULTADOS.....	42
3.1	Selección de organización beneficiaria.....	42
3.2	Creación de objetivos del proyecto.....	44
3.3	Creación de actividades.....	46
3.4	Asignación de tutores.....	48
3.5	Asignación de estudiantes.....	49
3.6	Flujo del proyecto.....	50
3.7	Pruebas de usabilidad.....	50
3.7.1	Encuesta.....	51
3.7.2	Resultados.....	52
CONCLUSIONES Y RECOMENDACIONES.....		53
BIBLIOGRAFÍA.....		55

## **CAPÍTULO 1**

### **1. DESCRIPCIÓN DEL PROBLEMA A RESOLVER.**

A partir del primer término del año 2016, la Escuela Superior Politécnica del Litoral definió como requisito para graduación, el cumplimiento de una cierta cantidad de horas de pasantías comunitarias, en las cuales, la ESPOL trabajaría en convenio junto a instituciones sin fines de lucro, con el fin de desarrollar proyectos que ayuden a resolver problemas de la comunidad. Los estudiantes por su parte, pueden participar en estos proyectos, para así cumplir con sus horas de pasantías comunitarias.

La creación de proyectos de vínculos con la sociedad, así como también el enrolamiento y seguimiento de las actividades de los estudiantes en los mismos, requiere la revisión de varios formularios, los cuales son escritos por estudiantes, tutores, directores e Instituciones beneficiarias. Dichos formularios, tienen un orden de revisión, por ejemplo, un usuario con el rol de director, debe llenar el formulario FOR-UVS-04, que contiene datos generales de un proyecto, el cual debe ser revisado por el coordinador de la carrera. Una vez que el coordinador lo aprueba, el proyecto está listo para su ejecución y, a continuación, nuevos formularios deberán ser llenados, esta vez, por parte de los estudiantes, quienes deben enviarlos para revisión a sus tutores respectivos.

El proceso de revisión mencionado anteriormente, actualmente es realizado de forma manual, es decir, cada vez que un estudiante necesita la revisión de un formulario, debe planificar una cita con su tutor o con la persona encargada de revisar, firmar y validar dicho formulario. Este proceso puede resultar un desperdicio de tiempo, puesto que, en ocasiones, es difícil coincidir en horarios de disponibilidad entre estudiantes, tutores, e incluso con las Instituciones Beneficiarias. Adicionalmente, varios de los formularios que deben ser llenados, contienen información repetida de otros formularios, lo cual provoca que el estudiante deba solicitar dicha información ya sea al tutor o director del proyecto.

## 1.1 Causas

- El programa para actividades comunitarias, es reciente en ESPOL, motivo por el cual, existe mucho desconocimiento acerca del proceso a seguir por parte de los estudiantes, y en ocasiones de los tutores.
- Los docentes encargados de la dirección de proyectos de vínculos con la sociedad, muchas veces tienen a su cargo más de un proyecto simultáneamente.
- Los docentes que ocupan el rol de tutor en los diversos proyectos, tienen a su cargo varios estudiantes a los cuales debe dar seguimiento, y de igual manera que los directores de proyecto, estos docentes pueden estar enrolados a varios proyectos, lo cual produce que el número de estudiantes a su cargo aumente considerablemente.
- Un docente no necesariamente tiene asignado un solo rol, por el contrario, la gran mayoría tiene más de uno, es decir, un docente podría ser director de un proyecto y simultáneamente ser tutor del mismo.

## 1.2 Efectos

- Se genera confusión en cuanto a la manera en que debe ser llenado un formulario específico, debido a la falta de información y a la falta de experiencia en este tipo de trámites.
- Al no existir un sistema que agilice el seguimiento y validación de formularios para los diversos proyectos, se vuelve prácticamente inevitable para las personas involucradas, realizar trámites mediante documentos en formato físico y organizar reuniones para la revisión de los mismos.
- Existen ocasiones en las cuales se deben volver a planificar reuniones entre estudiantes, tutores y/o instituciones beneficiarias para la revisión de formularios, debido a errores en el llenado de los mismos, lo cual produce un desperdicio de tiempo y recursos.
- La espera para la revisión de formularios suele tomar mucho tiempo, e incluso, retrasar el progreso del proyecto hasta que los formularios sean aprobados.



### 1.3 Trabajos Relacionados

El desarrollo de un sistema que agilice la gestión y aprobación de documentos, es una posible solución para resolver este problema. A continuación, se hablará acerca de sistemas existentes, que se encargan de este tipo de procesos, de los cuales, se obtuvo información útil para plantear una solución.

#### 1.3.1 Nuxeo

Consiste en un sistema de administración de contenidos libre, utilizado como software de gestión para documentos, páginas web, desarrollo colaborativo, entre otros [1]. Tal como lo menciona el sitio oficial de Nuxeo, el sistema está basado en un modelo de código abierto donde ofrecen un conjunto de herramientas para ayudar a los desarrolladores que crean aplicaciones de gestión de contenidos [2], entre las principales características de Nuxeo se encuentran las siguientes:

- Permite gestionar cómodamente documentos.
- Facilita el trabajo colaborativo, flujos de trabajo y búsquedas eficientes.
- Incluye opción para visualizar una vista preliminar de documentos.
- Importación de documentos basados en formularios.
- Es flexible, robusto y seguro, gracias a la implementación de estándares como SSL, Single Sign On(SSO).

#### 1.3.2 Alfresco

Consiste en una plataforma de código fuente libre, que ayuda a llevar el control de contenido crítico de los negocios, optimiza procesos y hace que el trabajo colaborativo sea fácil [3]. Alfresco incluye, repositorio de contenidos, un framework para administrar y usar contenido estándar en portales y también constituye un gestor documental y un completo CMS corporativo. Entre las principales características de Alfresco tenemos las siguientes:

- Posibilita la creación y gestión de contenidos empresariales desde una gran cantidad de paquetes informáticos como Office, OpenOffice.

- Ofrece una variedad de herramientas colaborativas.
- Facilidad de uso.
- Incluye soporte para gestión de Contenido Web.
- Es de Fácil despliegue y administración.

### **1.3.3 Ecount**

ECOUNT ERP es un software disponible en la nube para medianas empresas [4], incluye diversos módulos que ayudan a mejorar la administración y planificación de recursos empresariales. Entre las distintas funcionalidades que Ecount ofrece, llama la atención el módulo que permite la aprobación de documentos online debido a que es una funcionalidad necesaria para la solución propuesta, la cual será mencionada en el próximo capítulo.

Entre las principales características que incluye el software de aprobación online tenemos las siguientes [5]:

- Incluye proceso de aprobación integrado.
- Permite la creación de plantillas de aprobación para distintos tipos de reportes.
- Facilita el rastreo del progreso de una solicitud mediante un status de aprobación.
- Posibilita el adjuntar documentos al crear borradores.

## CAPÍTULO 2

### 2. DISEÑO E IMPLEMENTACIÓN DE LA APLICACIÓN WEB.

La solución propuesta consiste en la implementación de una aplicación web que permita controlar y/o gestionar las actividades de los estudiantes enrolados en los diversos proyectos de vínculos con la sociedad. Adicionalmente, el sistema desarrollado tiene la capacidad de automatizar el flujo de envío y aprobación de los diversos formularios que forman parte del ciclo de vida de un proyecto.

De esta manera, se busca agilizar la aprobación de formularios de manera digital, los actores involucrados en un proyecto, ya sean estos docentes o estudiantes, se verán beneficiados en gran medida por varios motivos: uno de ellos es porque que podrán llenar los respectivos formularios de manera digital, con la capacidad de imprimirlos o descargarlos en cualquier momento. Por otro lado, la(s) persona(s) encargada(s) de aprobar un determinado formulario, será notificada una vez que dicho formulario sea llenado y enviado, así podrá revisarlo digitalmente para asegurarse de que cumpla con todos los requisitos y así proceder con su aprobación.

#### 2.1 Objetivos del desarrollo de la aplicación web

##### 2.1.1 Objetivo general

Desarrollar una aplicación Web que agilice el control de las actividades de los estudiantes en proyectos relacionados con vínculos con la sociedad.

##### 2.1.2 Objetivos específicos

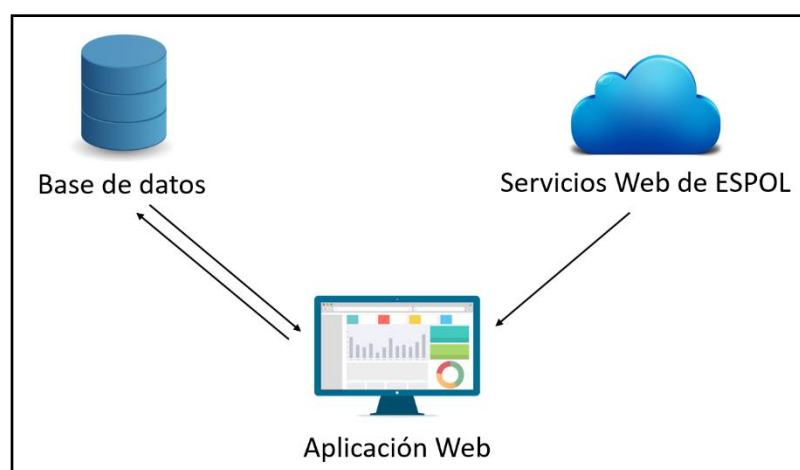
- Diseñar un esquema de base de datos apropiado para el almacenamiento de los datos referentes a los proyectos y formularios, así como también de los usuarios del sistema.
- Diseñar la estructura de la aplicación web mediante diagramas UML, que permitan representar los puntos más importantes del sistema.
- Implementar el proceso de envío y aprobación de formularios correspondientes a la creación de un proyecto, así como también del

seguimiento de las actividades de los estudiantes que forman parte del mismo.

- Permitir que un usuario privilegiado del sistema (administrador) asigne roles a los respectivos docentes que formarán parte de los diversos proyectos de vínculos con la sociedad.
- Exportar los formularios llenados en el sistema en formato PDF para su posterior impresión y firma.
- Administrar los usuarios que forman parte de un proyecto, ya sean estos docentes o estudiantes, con opción de agregarlos o removerlos en cualquier momento del equipo del proyecto.
- Implementar un mecanismo a través del cual los usuarios encargados de aprobar un determinado formulario, sean notificados cada vez que reciben un formulario para revisar.

## 2.2 Tecnologías y Herramientas Utilizadas

En la figura 2.1 se muestra un esquema básico del sistema, y para la implementación del mismo, se emplearon diferentes tecnologías y herramientas lo suficientemente robustas que permitieron el desarrollo de los diversos módulos de la aplicación, entre ellas tenemos, php laravel, mysql, phpMyAdmin y apache. A continuación, se detallará cada una de estas tecnologías.



**Figura 2.1: Esquema básico del sistema.**

### 2.2.1 Laravel

Laravel consiste en un framework de código abierto muy poderoso, que permite el desarrollo de sistemas y servicios web, con php5. Desarrollar aplicaciones con Laravel resulta extremadamente fácil, pues propone una metodología para desarrollar aplicaciones web de una forma ágil y sencilla, programar con este framework es sencillo, así se puede apreciar en el extracto de código mostrado en la figura 2.2. Laravel está hecho para arquitectura MVC, facilita el manejo de eventos, autenticación de usuarios, además es modular y extensible. Entre las principales características de Laravel tenemos las siguientes [6]:

- Cuenta con un sistema de enrutamiento rápido y eficiente.
- Permite analizar y filtrar llamadas HTTP en el servidor, muy útil cuando se trata de autorización de usuarios a determinadas vistas de la web.
- Cuenta con un ORM llamado Eloquent, que sirve para manejar la capa de persistencia de datos de una manera sencilla, esto es, mapear los objetos de PHP a las tablas en la base de datos.
- Existe una extensa y concisa documentación sobre este framework, muy sencilla de leer y entender [7].
- Incluye soporte para caché, con el fin de que las aplicaciones carguen mucho más rápido.
- Provee lo necesario para usar seguridad OpenSSL y cifrado de mensajes.
- Laravel incluye un motor de plantillas denominado Blade, el cual permite la creación de vistas parciales, elementos o zonas en los diversos Layouts de la aplicación.

```

class AuthenticationController extends Controller
{
    public function authenticate(Request $request)
    {
        //Obtengo el usuario y contraseña ingresado por el usuario
        $username = $request->input('username');
        $password = $request->input('password');

        //Se verifica primero con el web service de ESPOL la autenticacion del usuario
        //Se autenticado = EspoLWS::autenticacion($username,$password);
        //TEST
        $autenticado = true;

        if ($autenticado == true) {
            //Autenticacion con ESPOL exitosa
            Log::info('Usuario '.$username.' autenticado por ESPOL');

            //Luego autenticamos el usuario con nuestra base de datos, si no se autentica es porque el usuario
            //no existe, asi que lo creamos
            if (Auth::attempt(['username' => $username])) {
                //Usuario autenticado
                return $this->manejarAutenticacionExitosa();
            } else {
                //El usuario no existe en nuestra base de datos, asi que lo creamos
                $infoUsuario = EspoLWS::infoUsuario($username);
                //TEST
                $infoUsuario = $this->infoUsuarioStubWS();
                $this->registerEstudiante($username, $infoUsuario);
            }
        }
    }
}

```

**Figura 2.2: Extracto de código usando Laravel.**

### 2.2.2 MySQL

MySQL es un sistema de gestión de bases de datos relacional, multihilo, multiusuario y multiplataforma. Considerada como la base de datos código abierto más popular del mundo, muy utilizada por grandes organizaciones del mundo con el fin de ahorrar tiempo y dinero alimentando sus sitios Web de alto volumen [8].

MySQL es conocida por realizar búsqueda de datos e información a alta velocidad, además, cuenta con varias librerías y herramientas que permiten su uso a través de diversos lenguajes de programación, es de fácil uso e instalación.

### 2.2.3 PhpMyAdmin

Consiste en una herramienta escrita en PHP, con la intención de manejar la administración de MySQL a través de la web, adicionalmente, phpMyAdmin es compatible con una gama de operaciones de MySQL (operaciones de gestión de bases de datos, tablas, columnas, relaciones,

índices, usuarios, permisos, entre otros) operaciones las cuales pueden ser realizadas a través de la interfaz de usuario proporcionada, tal y como se muestra en la figura 2.3, además, tiene la capacidad para ejecutar cualquier sentencia SQL [9].

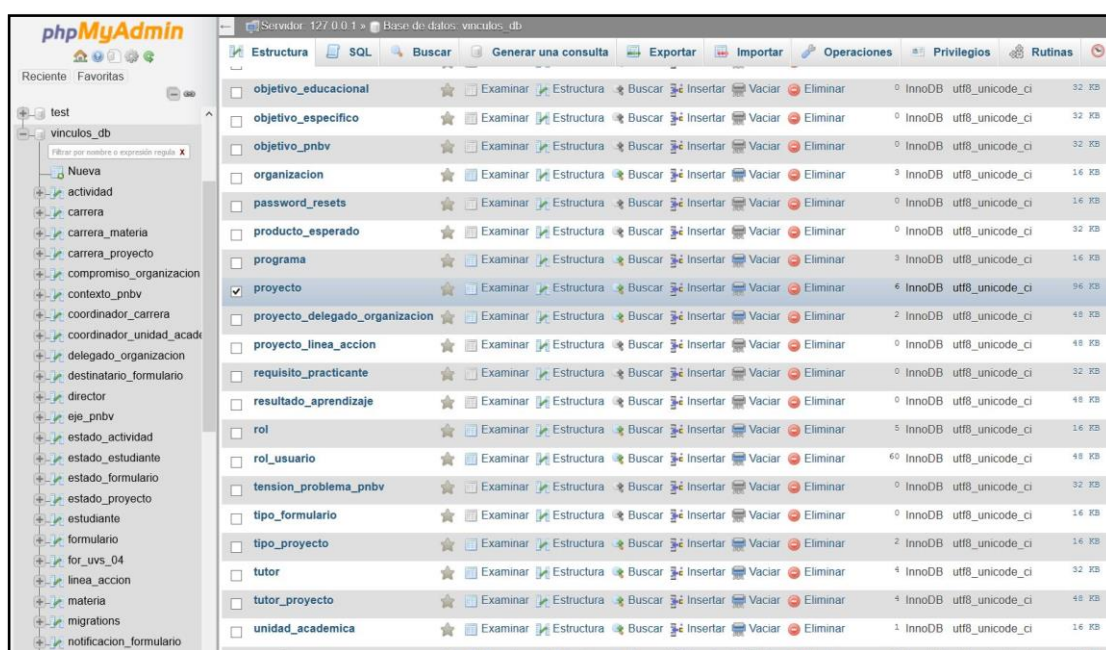


Figura 2.3: Interfaz de usuario de phpMyAdmin.

## 2.2.4 Servidor HTTP Apache

Consiste en un servidor web HTTP de código abierto y multiplataforma, es decir, puede trabajar con diferentes sistemas operativos y mantener un excelente rendimiento. Apache es utilizado principalmente, para realizar servicio a páginas web, ya sean estáticas o dinámicas [10], muy robusto y que destaca por su seguridad y rendimiento.

Entre las principales funcionalidades que permite Apache tenemos las siguientes [11]:

- Atender solicitudes de manera eficiente, debido al gran número de peticiones HTTP que puede recibir.
- Restricciones de acceso a los ficheros que no se quieran 'exponer'.

- Manejar los errores por páginas no encontradas, informando al visitante y/o redirigiendo a páginas predeterminadas.
- Gestión de la información a transmitir en función de su formato e informar adecuadamente al navegador que está solicitando dicho recurso.
- Gestión de logs, es decir, almacenar las peticiones recibidas, errores que se han producido y en general toda aquella información que puede ser registrada y analizada posteriormente para obtener las estadísticas de acceso al sitio web.

## **2.3 Diseño de la aplicación web**

### **2.3.1 Diseño UML**

Para el diseño UML de nuestra aplicación, optamos por utilizar los siguientes tipos de diagramas:

- Diagrama de casos de uso
- Diagramas de estados
- Diagramas de secuencia

Existen varios motivos por los cuales se escogieron estos tipos de diagramas para el diseño de la aplicación, y se detallan en este documento a medida que se presentan cada uno de ellos.

#### **Diagrama de casos de uso**

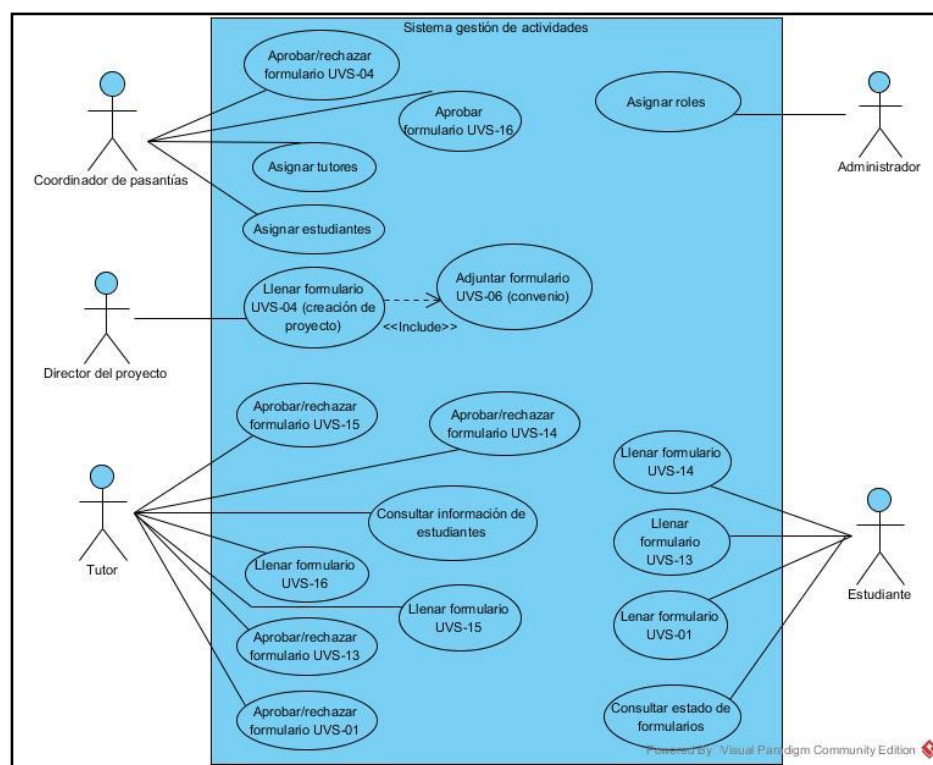
Se decidió utilizar el diagrama de casos de uso, debido a que se requería desmenuzar cada una de las funcionalidades del sistema, y sobre todo, determinar cuáles son los tipos de usuarios, y las tareas que cada uno de ellos pueden realizar en la plataforma.

Como se aprecia en el diagrama de casos de usos en la figura 2.4, existen varios tipos de usuarios o roles que podrán interactuar con el sistema, entre los cuales tenemos:

- Coordinador de pasantías de la carrera
- Director de proyecto



- Tutor
- Estudiante
- Administrador



**Figura 2.4: Diagrama de caso de uso de la aplicación web de gestión de actividades de vínculos con la sociedad.**

Cada uno de estos roles tiene relación con uno o varios formularios, ya sea para llenarlos o revisarlos. A continuación, se detalla cada uno de los roles del sistema:

**Coordinador de la Carrera.** – Es el coordinador de vínculos con la sociedad de la carrera, o el coordinador de pasantías pre-profesionales, y se encarga de la revisión del formulario FOR-UVS-04, el cual corresponde a la creación de un proyecto, es decir, este rol se encarga de aprobar y rechazar proyectos correspondientes a cuya carrera principal sea la carrera a la que el coordinador en cuestión pertenece. Adicionalmente, este tipo de usuario tiene como funcionalidades disponibles: la asignación

de tutores a un proyecto, la asignación de estudiantes a un proyecto, y la asignación de estudiantes a un determinado tutor. Un punto que vale la pena resaltar es que, cuando el coordinador hace uso de alguna de estas funcionalidades de asignación, se envían notificaciones a los usuarios involucrados.

**Director de proyecto.** – Es el encargado de crear proyectos en el sistema llenando el formulario FOR-UVS-04, en el cual, también puede asignar estudiantes para participar en los mismos. Este rol también tiene la posibilidad de visualizar información acerca de los formularios llenados por los estudiantes asignados en los proyectos, para así llevar un constante monitoreo del avance que lleva cada uno de ellos.

**Tutor.** – Este tipo de usuario puede ser asignado a uno o varios proyectos, y en cada uno de estos proyectos tiene bajo su tutoría varios estudiantes. Este rol tiene como tareas el llenado de varios formularios, tales como el formulario FOR-UVS-13 y el FOR-UVS-16. Adicional a esto, tiene como principal tarea la revisión de formularios llenados por los estudiantes que tiene asignado.

**Estudiante.** – Forma parte de un proyecto en específico, ya sea este de prácticas pre-profesionales o pasantías comunitarias. Cada uno de los estudiantes de un proyecto, tiene un tutor asignado, que como se mencionó previamente, es el encargado de aprobar sus formularios.

**Administrador.** – El rol administrador se encarga de asignar roles a los distintos docentes, cabe recalcar que un docente puede tener más de un rol, es decir, un docente podría ser director de un proyecto y a la vez ser tutor en otro proyecto. Otra de las tareas de este rol, es asignar los roles de coordinador a los diversos coordinadores de las carreras y unidades académicas.

Inicialmente se había incluido un rol para el delegado de la organización que está involucrada en un determinado proyecto, ya que este delegado debe firmar o aprobar uno de los formularios (FOR-UVS-15), el cual corresponde a un informe final por parte de la organización, en donde se

especifica que cada estudiante ha cumplido con las actividades que se comprometió a realizar. Finalmente, se decidió quitar este rol ya que no se justificaba tener un rol solamente para la aprobación de un solo formulario, por lo cual, quedó definido que el tutor sea el encargado de la generación de este formulario para luego hacérselo llegar físicamente al delegado de la organización y lo pueda firmar.

### **Diagramas de estado**

Una vez explicado en detalle el diagrama de casos de uso, otro tipo de diagrama UML importante para un mejor entendimiento del sistema, es el diagrama de estados, ya que en la aplicación existen entidades que manejan varios estados, como es el caso de un proyecto, un formulario y estudiante, por lo tanto, estas serán las entidades que se analizarán a continuación haciendo uso de los siguientes diagramas de estados:

#### **Proyecto**

El proyecto es una de las entidades más importantes en el sistema, ya que cada uno de los formularios que sean llenados por cada uno de los roles, estarán ligados a un proyecto en específico. En otras palabras, sin proyectos no habrá formularios que llenar.

Como se puede apreciar en la figura 2.5, el punto de partida es la creación de un proyecto, quedando dicho proyecto en el estado "Pendiente". Este estado podría ser también un sinónimo de creado, o de pendiente revisión.

En el caso de que el proyecto es rechazado por el rol encargado de su revisión (Coordinador), el proyecto pasaría al estado "Rechazado", estado del cual podrá salir solamente reenviando el formulario FOR-UVS-04 correspondiente a ese proyecto, con las respectivas correcciones que ayuden a que el proyecto sea aprobado. Al igual que la primera vez que se creó el proyecto, este pasara al estado "Pendiente", esperando a que sea revisado.

Una vez que el proyecto ha sido aprobado, mediante la aprobación del formulario FOR-UVS-04, el proyecto cambia su estado a "Aprobado", a

partir de este momento, estudiantes y tutores pueden ser asignados al proyecto. Luego de la aprobación del proyecto y de la asignación de los miembros del mismo, el director del proyecto será el encargado de arrancar oficialmente las actividades del proyecto, pasando el proyecto entonces al estado "Iniciado".

Finalmente, cuando todos los formularios correspondientes a todos los estudiantes que forman parte de ese proyecto, hayan sido entregados y aprobados, el proyecto será declarado oficialmente terminado, es decir, cuando todos los estudiantes hayan cumplido sus respectivas actividades en el proyecto. De esta manera, el proyecto pasara a su estado final "Terminado".

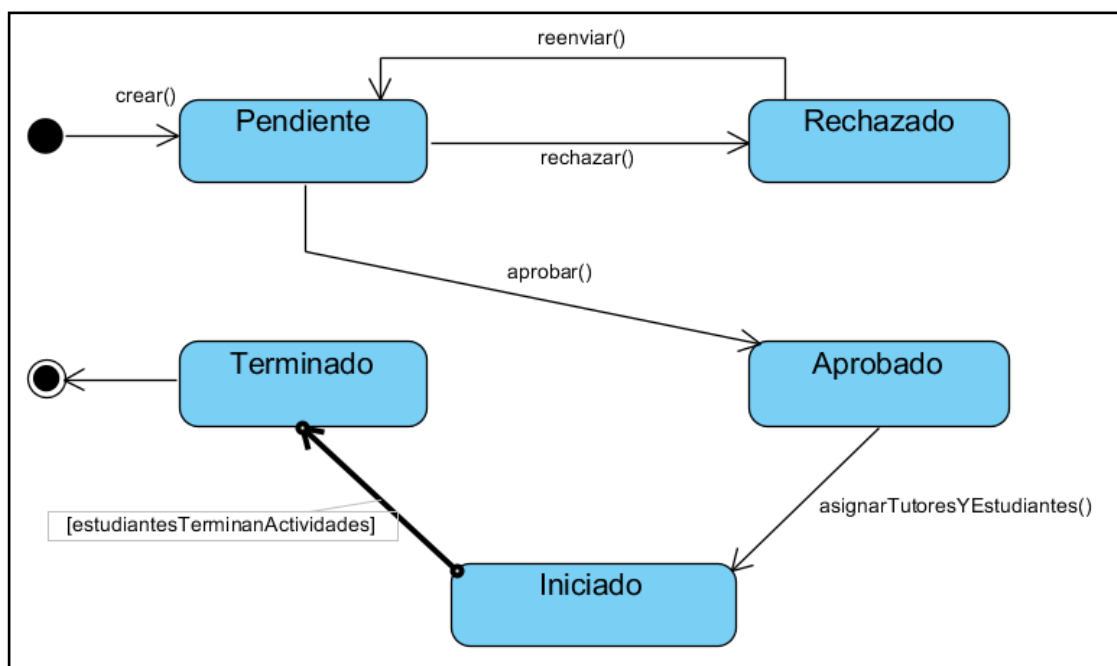


Figura 2.5: Diagrama de estados del proyecto.

## Formulario

El diagrama de estados de la entidad formulario, mostrado en la figura 2.6, tiene cierta similitud con el diagrama de estados de la entidad proyecto, pero con varias diferencias en cuando al contexto.

Todo formulario comienza con el estado "No enviado", del cual podrá salir una vez que haya sido enviado el formulario, pasando así al estado "Enviado", que equivaldría al estado "Pendiente" de un proyecto.

La manera en la cual se manejan los dos posibles estados siguientes ("Aprobado" y "Rechazado") es muy similar a como se manejan en la entidad proyecto, con la diferencia que, para proyecto solo nos referimos al formulario FOR-UVS-04, mientras que en formularios, nos referimos a todos los formularios que forman parte del sistema en general. Entonces al igual que en la entidad proyecto, un formulario pasaría al estado "Rechazado" de haber sido rechazado dicho formulario, y regresaría al estado "Pendiente" luego del reenvío del formulario.

De igual manera, un formulario podría pasar al estado "Aprobado" de ser aprobado dicho formulario, siendo este su estado final dentro del proyecto.

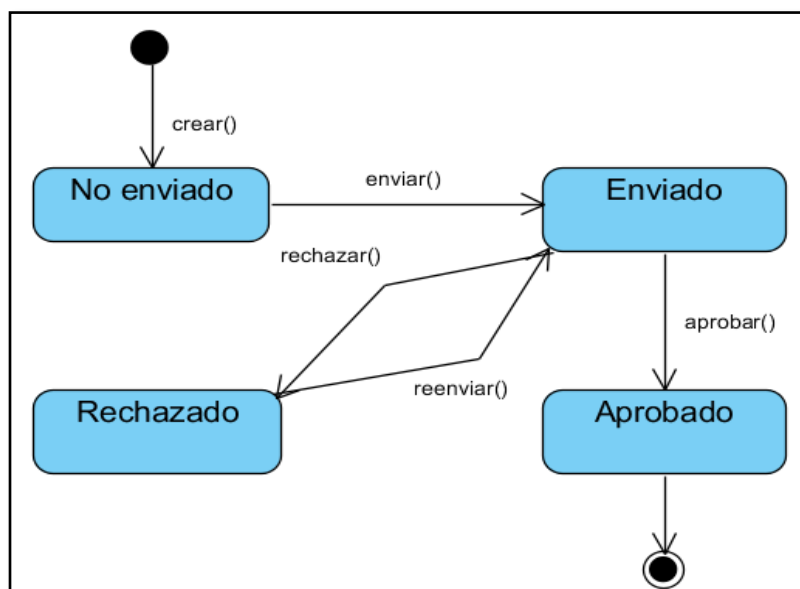


Figura 2.6: Diagrama de estados del formulario.

## **Estudiante**

Otra de las entidades importantes del sistema, es el estudiante, ya que cada uno de ellos tiene asignadas distintas actividades de un proyecto.

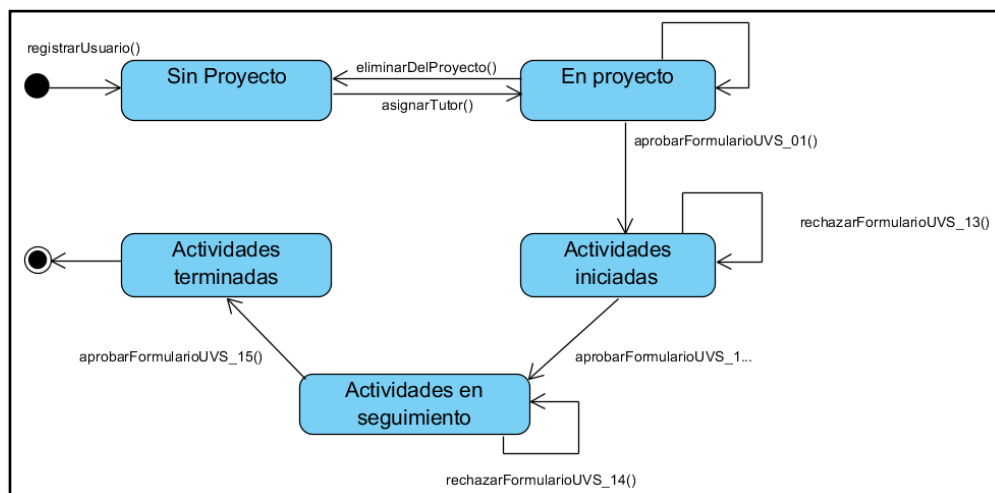
El estudiante podría formar parte de un proyecto de pasantías empresariales o de pasantías comunitarias, por ende, para cada uno de los estados que se muestran en el diagrama de la figura 2.7, se toma en cuenta al estudiante en un proyecto en específico, es decir, el estudiante podrá tener un estado para un proyecto de pasantías empresariales, y un estado para un proyecto de pasantías comunitarias.

Cuando un estudiante es asignado a un proyecto, comenzará con el estado "Sin tutor", hasta que se le asigna un tutor, y al suceder esto, el estudiante pasará al estado "Tutor asignado". De ser desligado el tutor del estudiante, este último regresaría al estado "Sin tutor", hasta que un nuevo tutor se le sea asignado.

Una vez que el estudiante tenga tutor asignado, podrá llenar el formulario FOR-UVS-01, el mismo que al ser aprobado dará paso al estudiante al estado "Actividades iniciadas", de ser rechazado este formulario, el estudiante se mantendrá en el estado "Tutor asignado".

Una vez transcurrido la mitad del tiempo del proyecto, el formulario FOR-UVS-13 deberá ser llenado por el tutor mediante una reunión entre el tutor y el estudiante, y una vez sea aprobado este formulario, el estudiante pasará al estado "Actividades en seguimiento".

Finalmente, el estudiante podrá pasar a su estado final, "Actividades terminadas", una vez que hayan sido llenado y aprobado todos los formularios correspondientes a los informes finales de actividades, ya sea por parte del tutor o del estudiante.



**Figura 2.7: Diagrama de estados del estudiante.**

### 2.3.2 Diseño de la base de datos

Para el análisis del diagrama de base de datos, debido a la extensión del modelo y al gran número de tablas de la base, se ha dividido el análisis en secciones, en las cuales se presenta en detalle las relaciones más importantes entre las entidades del sistema.

#### Roles y Usuarios

Como ya se ha mencionado previamente en el presente documento, un usuario puede tener asignado varios roles, por lo cual la relación entre la tabla “rol” y la tabla “usuario”, es de muchos a muchos, por ende, se creó la tabla intermedia “rol usuario”.

#### Usuario y Roles específicos

Por rol específico no nos referimos a una tabla que lleva ese nombre, sino más bien, al conjunto de tablas que representan a cada uno de los tipos de usuarios o roles individuales.

El fin principal del diseño, consiste en representar una relación de herencia entre la tabla usuario que simboliza la entidad padre, y las tablas “coordinador carrera comunitarias”, “coordinador carrera empresariales”,

“director”, “tutor”, “estudiante” y “administrador”, que simbolizan las entidades hijas.

El diseño fue hecho de esta manera debido a que existen muchos campos comunes para todos los tipos de usuario, entonces no valía la pena repetir cada uno de estos campos individualmente en las tablas de los tipos de usuario.

Otro de los motivos fue, que hay relaciones en las que participa simplemente un usuario sin importar si este es coordinador, director, estudiante, o cualquier otro rol, entonces se requería un usuario genérico que se pudiera usar en ese tipo de relaciones.

### **Carrera y Coordinadores**

Cada uno de los proyectos están ligados a una carrera, que es la responsable de la creación y aprobación del mismo, pero cabe recalcar también, que un proyecto puede ser multidisciplinario, es decir, pueden haber estudiantes de varias carreras en el mismo.

La carrera responsable del proyecto forma parte de una unidad académica, por ende, tanto la entidad carrera como la entidad unidad académica deben formar parte de la base de datos.

Las entidades mencionadas previamente en esta sección, están relacionadas a la tabla de rol específico, correspondiente a los coordinadores tanto de pasantías comunitarias como de pasantías empresariales de cada una de las carreras.

### **Proyectos y Directores**

Como ya se mencionó previamente en este documento, la entidad proyecto es posiblemente la más importante del sistema, ya que toda la lógica gira en torno a esta entidad. Debido a esto, prácticamente todos los tipos de usuario o roles (a excepción del rol administrador) están relacionados con la entidad proyecto, ya sea de manera directa, o indirecta.



Cada proyecto tiene un director de proyecto, que es el encargado de la creación del proyecto, además del constante seguimiento y monitoreo de las actividades de los estudiantes que forman parte del mismo.

Un proyecto tiene solamente un director de proyecto, mas no obstante, ese director podría ser también director de otro proyecto, por lo tanto, entre las entidades director y proyecto, existe una relación de uno a muchos.

La tabla proyecto es la más extensa de la base de datos, ya que es la entidad en torno a la cual gira toda la lógica del sistema. Dicha tabla contiene toda la información relacionada al proyecto, misma información que estará presente en varios formularios, pero mayoritariamente en el formulario FOR-UVS-04 (creación del proyecto).

### **Proyectos y Tutores**

En cuanto a la relación del rol tutor y proyecto, existe una tabla intermedia “tutor proyecto” entre estas dos entidades, y esto se debe a que existe una relación de muchos a muchos entre ellas, es decir, un usuario podría estar enrolado como tutor a varios proyectos, y a su vez este proyecto puede tener varios tutores enrolados.

### **Proyectos y Estudiantes**

La relación entre las entidades estudiante y proyecto es un tanto particular, ya que existen dos relaciones entre ellas, esto debido a que un estudiante puede estar enrolado a máximo dos proyectos de diferente tipo, es decir, un estudiante puede estar enrolado a un proyecto de pasantías comunitarias y/o a un proyecto de pasantías empresariales, por ende, la entidad estudiante tiene una relación por cada proyecto.

Se decidió separar esta relación en dos debido a que los proyectos son de diferente tipo, siendo manejados de manera distinta e independiente en el sistema.

### **Estudiantes y Tutores**

La relación entre las entidades estudiante y tutor, es muy similar a la que existe entre estudiante y proyecto, ya que como se mencionó previamente, un estudiante puede estar enrolado a máximo un proyecto de pasantías comunitarias, y a un proyecto de pasantías pre-profesionales.

Así mismo, un estudiante puede tener máximo un tutor para su proyecto de pasantías comunitarias, y un tutor para su proyecto de pasantías pre-profesionales, de tal manera que la entidad estudiante tiene dos relaciones con la entidad tutor.

### **Proyectos y Coordinadores**

Por otro lado, los coordinadores ya sean de la unidad académica, de pasantías comunitarias o de pasantías empresariales de la carrera, están relacionados con el proyecto de manera indirecta.

En el caso de los coordinadores de pasantías, están relacionados con proyecto por medio de una carrera, ya que el proyecto tiene una carrera principal que es la que va a definir cuál es el coordinador de pasantías, encargado de aprobar dicho proyecto. De manera similar, el coordinador de la unidad académica está relacionado con un proyecto, por medio de la unidad académica correspondiente a la carrera principal de dicho proyecto.

### **Proyectos, Organizaciones y Delegados**

Un proyecto es realizado en conjunto con una organización, ya sea esta una empresa, o una organización sin fines de lucro, dependiendo del tipo de proyecto. Dicha organización podría estar involucrada en varios proyectos diferentes, por lo tanto, se tiene una relación de uno a muchos entre las entidades “proyecto” y “organización”.

Otra de las entidades que entra en escena es “delegado organización”, la cual representa los diferentes delegados por parte de la organización que son quienes están en contacto con los miembros del equipo del proyecto.

En un proyecto puede haber varios delegados por parte de la organización, así como también, un delegado podría estar involucrado en varios proyectos, por ende, la relación entre estas dos entidades es de muchos a muchos, creando entonces así la tabla intermedia “proyecto delegado organización”.

### **Proyectos y Programas**

Un programa puede contener varios proyectos, por lo que la relación entre estas entidades es de uno a muchos, motivo por el cual, al momento de guardar un proyecto en la base de datos, se deberá tener ya guardado por lo menos un programa para poder asignárselo al proyecto.

### **Proyectos y Formularios**

Otra de las entidades importantes del sistema son los formularios, ya que en torno a ellos gira todo el flujo de trabajo. A lo largo del ciclo de vida de un proyecto, se requiere el llenado de varios formularios, cada uno de los cuales pertenece a un proyecto en específico.

De esta manera, se puede determinar que la relación entre las entidades proyecto y formulario es de uno a muchos.

### **Formularios y Formularios específicos**

Dado que el sistema trabaja con distintos tipos de formularios, manejan información distinta a las demás, por tal motivo, se tuvo la necesidad de crear tablas específicas para cada uno de ellos, los cuales tienen relación de uno a uno con la tabla formulario.

Para el formulario FOR-UVS-01 se creó la tabla “for\_uvs\_01”, un campo importante de esta tabla es la llamada ‘revisado por director’, este campo ayuda a verificar que un formulario ya haya sido revisado por el director del proyecto, ya que mientras el formulario no sea revisado, el estudiante no tendrá disponible la opción para editar el formulario FOR-UVS-01.

Para el formulario FOR-UVS-13 se creó la tabla “for\_uvs\_13”, este formulario es creado para un tutor, pero es específico de un estudiante,

ya que aquí el tutor verifica las actividades realizadas por el estudiante en el proyecto, por tal motivo, la tabla estudiante tiene relación uno a muchos con la tabla estudiante, de esa forma podemos obtener información del estudiante para llenarla automáticamente en el formulario.

Para el formulario FOR-UVS-14 se creó la tabla "for\_uvs\_14", este formulario es creado para un estudiante y aquí se debe llenar información sobre análisis de actividades, conclusiones y recomendaciones, esta tabla tiene creado un campo para llenar por separado toda esta información.

Para el formulario FOR-UVS-15 se creó la tabla "for\_uvs\_15", el formulario es creado para el tutor y es específico para un estudiante tal y como pasa para el formulario FOR-UVS-13, por tal motivo, aquí también se tiene la relación con la tabla estudiante para obtener su información.

Para el formulario FOR-UVS-16, se creó la tabla "for\_uvs\_16", al igual que formulario FOR-UVS-13 y FOR-UVS-15, este formulario se crea para el tutor y es específico sobre un estudiante y también existe la relación con la tabla estudiantes, adicionalmente, esta tabla contiene un campo para guardar el número de horas de prácticas validadas y un campo booleano para establecer si el estudiante aprobó o no la práctica.

### **Notificaciones**

El caso de la entidad notificación es un tanto particular, debido a que está relacionada con varias tablas para poder acceder a información necesaria para el envío de notificaciones de diversos tipos. Las notificaciones sirven para tener un historial de eventos que han ocurrido a lo largo del ciclo de vida del proyecto, uno de los campos importantes de esta entidad es el campo "visto", que indica si dicha notificación ya fue vista por su destinatario.

Existe dos relaciones generales que son utilizadas para todos los tipos de notificaciones: Una de ellas es la relación con proyecto y la otra es la relación con destinatario, que involucra a la entidad usuario, e indica a que usuario va dirigida esta notificación. Las demás relaciones son propias de

un tipo de notificación en específico, tal es el caso de las relaciones “tutor asignado” o “rol asignado”, que son utilizadas por las notificaciones de tutor asignado y de rol asignado específicamente

### **Formularios y Usuarios**

Un formulario generalmente tiene dos o más usuarios involucrados: el usuario que llena el formulario y el que revisa el formulario en cuestión.

Para esto, existen las relaciones “creado por” y “destinatario”, las cuales relacionan las entidades formulario y usuario. De esta manera, se puede especificar al momento de crear un formulario, cual usuario es el llamado a llenar dicho formulario, y cual usuario o usuarios son los llamados a revisarlo.

### **Resultados de Aprendizaje**

En cada uno de los proyectos existen los denominados “Resultados de aprendizaje”, los cuales son utilizados en dos formularios, FOR-UVS-04 y FOR-UVS-16, por lo tanto, fue necesario crear una tabla en la base de datos exclusivamente para almacenar esta información.

Cada uno de los resultados de aprendizaje están ligados a una materia, y una materia pudiera tener varios resultados de aprendizaje. Los resultados de aprendizaje son definidos para el proyecto, mas no para cada estudiante, lo que sí está definido para cada estudiante son las evidencias de los resultados de aprendizaje, por ende, también se creó una entidad para ello, la cual está relacionada con un resultado de aprendizaje, y con un formulario FOR-UVS-16, que a su vez está ligado a un único estudiante.

### **Proyectos y Actividades**

Un proyecto tiene varios objetivos específicos, por tal motivo, existe la relación de uno a muchos entre la tabla proyecto y objetivos específico, es importante esta relación ya que más adelante se crearán actividades.

Todo proyecto tiene su cronograma, el cual será armado por el usuario según los objetivos específicos del proyecto, ya que en este caso se los tratará como componentes, y cada componente requiere de varias actividades o tareas las cuales deben ser completadas para cumplir con el objetivo específico.

Por tal razón, existe la relación de uno a muchos entre la tabla objetivo específico y actividades, y ya que objetivo específico tiene la relación directa con proyecto, se tiene como saber a qué proyecto pertenece una actividad en específico.

### **Actividades y sub-actividades**

Esta relación es necesaria ya que se da el caso en que ciertas actividades pueden tener sub-actividades, las cuales pueden ser agregadas por los estudiantes cuando se trata de un proyecto comunitario, para ello, a la tabla actividad se le agrego dos campos llamados “es actividad micro” y “es actividad macro”, de esta manera se puede diferenciar entre una actividad y una sub-actividad.

Finalmente, para saber que sub-actividades pertenecen a una actividad en concreto, se creó la relación entre actividad consigo misma, de esta manera, si en la tabla actividad se tiene un registro que contiene la clave foránea de otra actividad, entonces se trata de una sub-actividad.

## **2.4 Implementación**

### **2.4.1 Flujo de trabajo**

Previo a describir la implementación del sistema, es importante tener una idea del flujo de trabajo de un proyecto desde el momento en que inicia, hasta que finaliza, a continuación, se muestra en la figura 2.8 una ilustración del flujo de trabajo de un proyecto desde el momento en que inicia, hasta que finaliza, así como también una descripción detallada de dicho flujo.



**Figura 2.8: Ilustración de flujo de trabajo de un proyecto.**

Inicialmente, el proyecto debe ser creado por un director, asignando una carrera principal a la cual pertenece la misma, para tal caso, el formulario FOR-UVS-04 debe ser llenado y posteriormente aprobado por el coordinador de la carrera principal a la que pertenece el proyecto.

Si el proyecto es aprobado, entonces el coordinador, que lo acaba de aprobar, asigna tutores y estudiantes al proyecto. Luego, a cada estudiante, se le debe asignar un tutor, el mismo que será el encargado de dar seguimiento al estudiante, revisar sus formularios y emitir el informe final del estudiante una vez culminada la práctica comunitaria.

Una vez definido el equipo del proyecto, el director del mismo, debe asignar las actividades que cada estudiante realizará, las cuales deben ser seleccionadas del cronograma que fue establecido al momento de crear el proyecto (FOR-UVS-04).

A continuación, todo estudiante debe presentar su respectivo formulario FOR-UVS-01. En este formulario, el estudiante podrá agregar sub-actividades, con respecto a las actividades que le fueron asignadas por el director. El estudiante tendrá que enviar su formulario para ser revisado y aprobado por el tutor.

Cuando ha transcurrido la mitad del tiempo de las prácticas comunitarias, habrá una reunión entre tutores y estudiantes para dar seguimiento a las actividades, para este caso, el tutor debe llenar el formulario FOR-UVS-13 para cada estudiante, formulario el cual debe contener las actividades

que el estudiante ha cumplido hasta la fecha incluyendo sugerencias para cada actividad de ser el caso.

En el momento que finaliza el tiempo de las prácticas comunitarias, habrá una nueva reunión entre estudiantes y tutores, donde los estudiantes presentarán sus respectivos formularios FOR-UVS-14 (Informe final del practicante), formulario donde los estudiantes realizan un análisis de las actividades realizadas, incluyendo conclusiones y recomendaciones. Adicionalmente, deben presentar el informe FOR-UVS-15 donde el responsable o delegado de la organización evalúa las actividades del estudiante

Luego que el tutor aprueba ambos formularios, el tutor debe llenar el formulario FOR-UVS-16, en el cual evalúa las actividades realizadas por el estudiante, este formulario finalmente debe ser aprobado y firmado por el coordinador, dando así por finalizada las practicas del estudiante en el proyecto.

#### **2.4.2 Servicios Web de ESPOL**

Varias de las funcionalidades del sistema, tales como la autenticación o el ingreso de nuevos usuarios, requieren obtener información acerca de los miembros de ESPOL, ya sean estos profesores (directores, tutores), o estudiantes. Para esto se utilizaron los servicios web que proporciona ESPOL por medio de una nueva clase denominada EspoIWS, la cual usa la extensión SoapClient de PHP para consumir los servicios web. Los servicios utilizados por el sistema son 3 y se detallan a continuación:

##### **1. Autenticación**

Recibe como parámetros las credenciales del usuario de ESPOL, usuario y contraseña, retornando un valor booleano “true” en caso que el usuario se autentique satisfactoriamente, caso contrario retorna “false”. Este servicio es utilizado para realizar el inicio de sesión en el sistema.



## **2. DatosProfesionalCedula**

Recibe como parámetro un número de cédula, el cual corresponde a un profesional de la ESPOL, y retorna la información del usuario correspondiente a dicha cédula. Este servicio es utilizado al momento de ingresar nuevos usuarios al sistema tales como coordinadores, directores, y tutores.

## **3. DatosEstudianteMatricula**

Recibe como parámetro el número de matrícula de un estudiante, y retorna la información correspondiente a dicho estudiante. Este servicio es utilizado al momento de asignar un estudiante a un proyecto por medio de su número de matrícula.

### **2.4.3 Desarrollo de formularios**

En esta sección se detallan los detalles acerca de la implementación de 6 de los formularios que forman parte del flujo de trabajo descrito anteriormente.

#### **1. FOR-UVS-04**

El formulario FOR-UVS-04 es uno de los formularios más extensos, y muy probablemente el más importante, ya que por medio de él se efectúa la creación de un proyecto. La mayor parte de los ítems del formulario son campos de texto sencillos, por lo que solamente se explicará la implementación de campos y secciones que involucren cierta lógica. En este formulario se tiene como primera sección los datos generales del proyecto teniendo como primer punto el título del proyecto. Seguidamente se tiene el campo de “Programa al cual pertenece el proyecto”, el cual utiliza una lista desplegable que muestra los programas disponibles en el sistema, donde el usuario debe simplemente seleccionar uno.

Otro de los campos que pertenece a esta primera sección es el de “Carrera(s) que participa(n) en el proyecto”, el cual se decidió dividir en dos campos, ambos con la característica de autocomplete. Se tomó la decisión de dividir en dos campos este ítem debido a que se necesita

diferenciar la carrera principal del proyecto, de las otras carreras adicionales.

Como otro de los campos en esta sección, se tiene la duración del proyecto, en el cual se debe especificar la fecha de inicio y la fecha de finalización del proyecto. Para esto se utilizó el componente “Date Range Picker” [12], el cual sirve para escoger un rango de fechas, por lo tanto, se decidió unir los campos de fechas de inicio y finalización en un solo campo, con la intención de que el “Date Range Picker” valide automáticamente que la fecha de finalización no pueda ser anterior a la fecha de inicio del proyecto.

Otro de los componentes que vale la pena mencionar es el “CKEditor” [13], que consiste en un Rich Text Editor orientado a ser usado en aplicaciones Laravel. Este editor tiene opciones muy útiles para dar el formato deseado a un texto tales como: alineado, tamaño de letra, color, inserción de tablas, entre otras. Este editor es utilizado en varios de los campos de este formulario, como lo son: Área y sub-área del conocimiento y conocimiento específico, Información relacionada al Plan Nacional del Buen Vivir, Justificación, Pertinencia, Metodología, Productos esperados, Compromisos de la organización, y Bibliografía.

En varios de los campos del presente formulario se hace referencia a otros formularios informativos o de ayuda, los cuales contienen información con la cual debe ser llenado el formulario FOR-UVS-04. Estos formularios de ayuda se encuentran almacenados en el servidor para que puedan ser accedidos por el usuario mediante links, y así pueda utilizar la información contenida en los mismos.

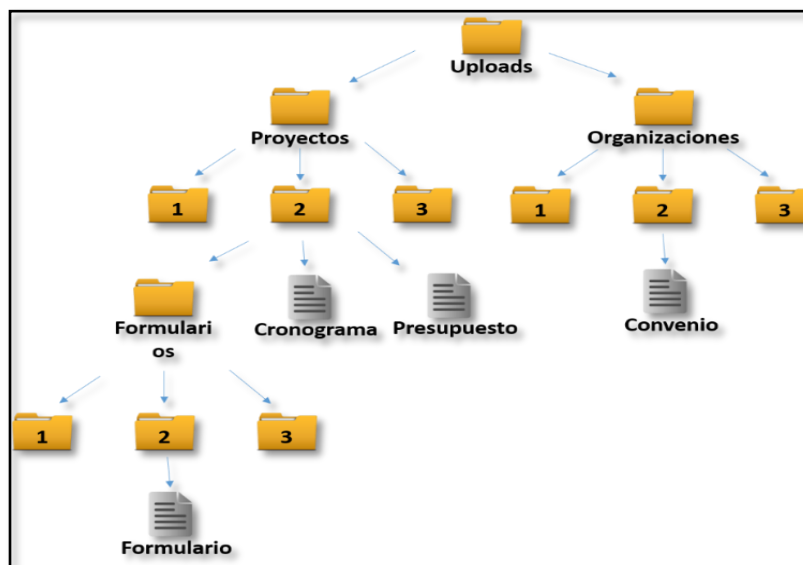
Por otro lado, existen secciones que deben ser llenadas automáticamente, por ejemplo, “Datos Generales del Equipo del Proyecto”, donde se carga automáticamente los datos del director del proyecto. También se deben cargar los datos del coordinador de pasantías de la carrera seleccionada como carrera principal, dependiendo si este es un proyecto de pasantías comunitarias o pre profesionales.

Continuando con el formulario FOR-UVS-04, podemos mencionar también la sección correspondiente a la “Organización beneficiaria del proyecto”, en la cual se presenta al usuario las opciones para “Agregar una nueva organización” o “Elegir existente”.

Seleccionando la primera opción, el usuario tiene la opción de crear una nueva organización en la base de datos mediante un panel modal en el que se ingresarán los datos de la organización contenidos en un elemento ‘form’ de la página, el cual enviará una petición AJAX para guardar la organización. Se decidió usar una petición AJAX en este caso para evitar que la página se recargue y se pierdan los datos del proyecto que hayamos ingresado hasta ese momento.

Por otro lado, al seleccionar la segunda opción, se muestra al usuario un campo de texto con autocomplete, por medio del cual se puede buscar la organización deseada por medio del nombre.

El último punto importante en este formulario, es la subida de archivos adjuntos, tales como: cronograma, presupuesto, y convenio con la organización. Dichos documentos son subidos al servidor haciendo uso de un campo de texto de tipo “archivo”, para posteriormente ser procesado en el controlador y guardado en su directorio correspondiente de acuerdo a la siguiente estructura mostrada en la figura 2.9.



**Figura 2.9: Estructura de directorios para almacenamiento de archivos en el servidor.**

## 2. FOR-UVS-01

El formulario FOR-UVS-01 debe ser llenado por un estudiante, y consiste en el formulario del plan de actividades del mismo, contiene información sobre datos generales del proyecto, identificación del problema a resolver, objetivos, justificación y metodología de trabajo, toda esta información es generada automáticamente obteniendo los datos ingresados al momento de crear el proyecto (FOR-UVS-04).

Adicionalmente, este formulario debe contener el cronograma que el estudiante seguirá durante su práctica, para tal caso, previo a tener disponible el formulario el estudiante para llenarlo, el director debe asignarle las actividades a realizar, las cuales serán seleccionadas del cronograma general del proyecto, esto ocurre en caso de tratarse de un proyecto comunitario, en el caso de tratarse de un proyecto de práctica pre profesional, el director no asigna actividades al estudiante, sino que el mismo estudiante arma su cronograma al momento de llenar el formulario FOR-UVS-01.

Todo formulario FOR-UVS-01 debe ser revisado y aprobado por el tutor que tiene asignado el estudiante, una vez aprobado, se habilita el formulario FOR-UVS-13 específico del estudiante el cual debe ser llenado y revisado por el mismo tutor.

### **3. FOR-UVS-13**

El formulario FOR-UVS-13 debe ser llenado por el tutor, consiste en el formulario de seguimiento de actividades tutoriadas en el cual el tutor verifica las actividades que el estudiante ha cumplido hasta la mitad de la fecha del proyecto.

La información contenida en este formulario consiste en una tabla con la lista de actividades del estudiante, el tutor tendrá la opción de marcar las actividades que el estudiante ha cumplido y agregar un comentario o sugerencia a la actividad de ser necesario. Una vez que el tutor marca como revisado el formulario, a continuación, se habilita el formulario FOR-UVS-14 al estudiante.

### **4. FOR-UVS-14**

El formulario FOR-UVS-14 debe ser llenado por el estudiante, consiste en el informe final del practicante el cual debe ser presentado cuando el estudiante finalice sus prácticas. Este formulario contiene información del practicante, información del tutor e información de la Organización o Empresa. La información del tutor y del estudiante es obtenida mediante los servicios web de ESPOL, mientras que la información de la organización es cargada automáticamente de los datos ingresados al momento de crear el proyecto.

La nueva información que el usuario debe ingresar en este formulario es un análisis de las actividades realizadas, conclusiones y recomendaciones.

Internamente en el sistema, el formulario debe ser aprobado por el tutor del estudiante, sin embargo, se requiere la firma del responsable de la organización en el formulario. Aun así, apenas el tutor aprueba el

formulario en el sistema, se habilitan los formularios FOR-UVS-15 y FOR-UVS-16, específicos de un estudiante, al tutor.

#### **5. FOR-UVS-15**

El formulario FOR-UVS-15 está disponible para el tutor y consiste en el informe final de la Organización o Empresa. Contiene información sobre el practicante y el responsable por parte de la organización beneficiaria, información la cual es generada automáticamente obteniendo los datos ingresados al momento de crear el proyecto.

El formulario también contiene una tabla de calificación, donde la organización evalúa el trabajo realizado por el estudiante. Es importante mencionar que la organización no tendrá acceso al sistema, motivo por el cual, este tipo de formulario se lo trabaja de manera distinta, en este caso, el tutor tiene habilitado el formulario para descargarlo, con toda la información necesaria.

Se hace llegar la versión descargada del formulario a la organización, para que luego una versión escaneada del mismo, sea subida por el tutor en el sistema. Internamente en el sistema, el estado del formulario FOR-UVS-15 cambia su estado a “Aprobado” una vez el formulario sea subido al sistema.

#### **6. FOR-UVS-16**

El formulario FOR-UVS-16 es específico sobre un estudiante y está disponible para ser llenado por el tutor. Consiste en el informe final del docente tutor y contiene información sobre datos generales de la práctica, una tabla de las actividades del estudiante donde el tutor evaluará si las cumplió o no. Este formulario incluye también una sección sobre la evaluación de los resultados de aprendizaje del estudiante y finalmente una sección sobre evaluación general de la práctica, es aquí donde el tutor aprueba la práctica del estudiante y el número de horas de prácticas validadas.

Todo formulario FOR-UVS-16 es enviado al coordinador de carrera, y una vez que se aprueba, se dan por finalizadas las prácticas del estudiante, e internamente en el sistema cambia el estado del estudiante a “Actividades finalizadas”.

#### **2.4.4 Funcionalidades del sistema**

En la siguiente sección se explica a detalle la implementación de cada una de las funcionalidades del sistema, mencionando también ciertos componentes que se utilizaron.

##### **Autenticación**

Para la autenticación de los usuarios, se usa en primera instancia el servicio web de autenticación de ESPOL, para determinar si este usuario debe ser aceptado en el sistema o no. De obtener como respuesta “true” por parte del servicio web, se procede a autenticar el usuario contra nuestra base de datos, utilizando solamente el usuario y no la contraseña, ya que esta fue validada previamente por el servicio web de ESPOL. De ser exitosa la autenticación, se le da acceso al sistema al usuario, caso contrario se redirige al usuario nuevamente la página de inicio de sesión, pero esta vez mostrando un “flash message” indicando que el usuario no fue encontrado en la base de datos. Los “flash messages” son mensajes que duran solamente un requerimiento HTTP, es decir, al hacer un nuevo requerimiento HTTP, este mensaje se perderá.

##### **Administración de roles y usuarios**

El rol de administrador tiene entre sus tareas principales, la asignación de roles, para lo cual se decidió implementar una página con la lista de usuarios, que puede ser filtrada por el nombre del usuario y así poder buscar más rápido un usuario en específico. En esta página también se da la opción de agregar un nuevo usuario por medio del botón “Nuevo Usuario”, que al ser presionado, abrirá un panel modal en donde se ingresará el número de cedula del usuario a ser agregado, cabe recalcar que esta opción está orientada a ser utilizada solamente para usuarios

profesores, debido a que utilizamos el servicio web de ESPOL “DatosProfesionalCedula” que solo aplica para profesionales de la ESPOL. Una vez ingresado el número de cedula y de haber presionado el botón “Buscar”, se realizará el request para comunicarse con el controlador respectivo en donde se efectuará la creación del usuario en la base de datos.

Luego de crearse el usuario, este se verá reflejado en la lista de usuarios de la página principal del administrador, en donde también se da la opción al administrador para que pueda asignar roles a los usuarios mediante un botón con el símbolo “+” que al ser presionado abrirá un panel flotante donde se seleccionará el rol a ser asignado. Si se selecciona el rol tutor, coordinador de pasantías comunitarias, o coordinador de pasantías pre-profesionales, se mostrará adicionalmente debajo del campo de rol, el campo de carrera a la cual pertenece ese tutor o coordinador. Se decidió implementarlo de esta manera debido a que un tutor o un coordinador deben estar ligado a una sola carrera en específico.

Al haber asignado el rol al usuario, este se mostrará en la lista de usuarios a manera de una etiqueta, la cual contiene: el nombre del rol, el botón de editar que sirve para cambiar la carrera en caso de ser este rol tutor o coordinador, y el botón de eliminar que sirve para quitar ese rol al usuario.

### **Administración de programas**

Otra de las opciones que tiene el administrador es la de “Programas”, en donde se presenta la lista de todos los programas guardados en la base de datos, similar a la lista de usuarios, con la diferencia que en este caso se tienen los botones de editar y eliminar en la misma lista y no dentro de una etiqueta de rol. Adicionalmente se tiene la opción de agregar un nuevo programa, al igual que los usuarios.

Un punto que vale la pena recalcar de la implementación de esta página, es la eliminación de programas en la cual no permitimos que el usuario elimine un programa que ya contenga al menos un proyecto, debido a que esto causaría conflictos en cuanto a la integridad de los datos en la base



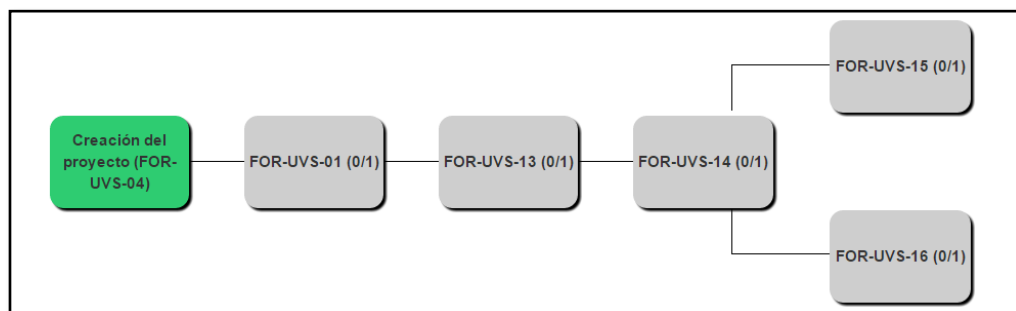
de datos, ya que la tabla proyecto tiene como una de sus claves foráneas el identificador del programa al cual pertenece, entonces, al eliminar ese programa, esa clave foránea quedaría nula.

### **Administración de organizaciones**

Esta es una de las opciones que tiene disponible el director de proyecto, ya sea dentro de la sección de organización beneficiaria del formulario FOR-UVS-04 del proyecto o por medio de la página “Organizaciones” donde se muestra al usuario la lista de organizaciones guardadas en la base de datos, similar a la página de usuarios y la página de programas. Al igual que estas páginas, también se da la opción de agregar un nuevo registro a la base de datos, en este caso una nueva organización, para que esta pueda ser usada posteriormente en otros proyectos. La implementación del listado y agregación de organizaciones, se realizó de la misma manera que en las páginas mencionadas anteriormente.

### **Flujo del proyecto**

El grafico de flujo de proyecto es una manera de dar seguimiento al estado del proyecto, así como también al estado de los formularios de cada estudiante que forma parte del mismo. Este grafico consta de 6 bloques, cada uno representando un tipo de formulario, ubicados de izquierda a derecha en el orden en que deben ir siendo llenados a lo largo del ciclo de vida del proyecto. Cada uno de estos bloques cambian de color de acuerdo al estado del formulario correspondiente a ese bloque, es decir, cuando los formularios correspondientes a un determinado bloque han sido llenados para todos los estudiantes, este bloque será de color verde, si los formularios han sido llenados solamente para algunos estudiantes, el bloque será de color amarillo y, por último, si ningún estudiante ha llenado ese formulario, el bloque se quedará de color gris, que es su color inicial. A continuación, se muestra una ilustración del grafico de flujo del proyecto en la figura 2.10.



**Figura 2.10: Grafico del flujo de un proyecto.**

Para la implementación del cambio de color de estos bloques, se crearon consultas SQL para determinar cuántos de los estudiantes del proyecto han llenado cierto tipo de formularios, de esta manera podemos saber si todos, algunos, o ningún estudiante llenó el formulario, y de acuerdo a esto, se le agrega o no cierta clase CSS al bloque para darle el color correspondiente. De esta manera, también es posible presentar dentro de cada bloque el número exacto de estudiante que llenaron dicho formulario con respecto al número total de estudiantes que participan en el proyecto.

### **Detalles del proyecto**

Esta opción la tendrá disponible el director, coordinador y tutor del proyecto, la idea de esta vista es mostrar información acerca de un proyecto en específico, sus informes principales y el equipo involucrado (tutores y estudiantes).

La sección de informes principales muestra los informes FOR-UVS-04, FOR-UVS-08, FOR-UVS-13 y el cronograma. Los formularios FOR-UVS-04 y FOR-UVS-13 tiene habilitada la opción para vista previa y descarga ya que son formularios creados en el sistema, por su parte, tanto el formulario FOR-UVS-08 como el cronograma, tendrán habilitada únicamente la opción para descargar, esto debido a que ambos son tratados como archivos adjuntos cuando se crea el proyecto.

En cuanto al equipo involucrado, si ha iniciado sesión un director o coordinador, entonces será visible el equipo entero del proyecto, tutores y

estudiantes, por su parte, si inicia sesión un tutor, entonces podrá visualizar únicamente la lista de estudiantes que el tutor tiene asignado.

### **Formularios disponibles**

Esta sección muestra una lista de formularios pertenecientes al proyecto específico y al usuario que inició sesión en el sistema, estos formularios deben ser llenados por el usuario para posteriormente enviarlos a revisión. Estos formularios, según el tipo, estarán disponibles según el avance del flujo de trabajo del proyecto, ya que, cuando un formulario en específico se aprueba, otros se habilitan para ciertos usuarios.

A continuación, se detallan los formularios disponibles que pueden tener los diferentes tipos de usuarios:

- Para un **director**, los formularios que tendrá disponibles para completar serán únicamente los formularios FOR-UVS-01 de los estudiantes, donde seleccionará las actividades que un estudiante realizará.
- Los **tutores**, en algún momento del proyecto tendrán disponible para llenar los formularios FOR-UVS-13, FOR-UVS-15 y FOR-UVS-16.
- Los **estudiantes**, en algún momento del proyecto tendrán disponible para llenar los formularios FOR-UVS-01 y FOR-UVS-14.

### **Revisión de formularios**

En esta opción se muestra una lista de formularios específicos del proyecto que el usuario tiene que revisar o aprobar.

Todo formulario que aparezca en esta lista tendrá únicamente el “Modo ver”, y al visualizar el formulario, al final de la vista del mismo, tendrá los botones donde el usuario puede aprobar o rechazar dicho formulario, en caso de rechazarlo, aparecerá una ventana modal para que el usuario ingrese un comentario de la razón del rechazo del formulario.

A continuación, se detallan los formularios por revisar que pueden tener los diferentes tipos de usuarios:

- Los **tutores**, en algún momento del proyecto tendrán disponibles para revisar los formularios FOR-UVS-01 y FOR-UVS-14 específicos de un estudiante.
- Los **coordinadores**, tendrán disponibles para revisar los formularios FOR-UVS-16, creados por un tutor pero que es específico para un estudiante, y también tendrá disponible para aprobar los formularios FOR-UVS-04 que los directores envían cada vez que crean un proyecto.

### **Asignación de tutores a un proyecto**

Cuando un proyecto es aprobado, el coordinador que lo aprobó tendrá habilitada la opción para asignar tutores. La búsqueda se hace mediante nombre o apellidos del docente. Cabe recalcar que la búsqueda se hará solo entre los docentes que tienen asignado el rol de tutor. Una vez que se hace la búsqueda y selección del docente tutor, éste es agregado a la lista de tutores del proyecto.

### **Asignación de estudiantes a un proyecto**

Al igual que los tutores, una vez que el proyecto es aprobado, el coordinador tendrá habilitada la opción para asignar estudiantes al proyecto. La búsqueda de estudiantes se realiza mediante el número de matrícula. En caso de que se trate de asignar un estudiante que ya forma parte de otro proyecto, el sistema no lo permitirá y mostrará un mensaje de advertencia, caso contrario, el estudiante es agregado a la lista de estudiantes del proyecto.

### **Asignación de estudiantes a un tutor**

Adicionalmente, todo estudiante asignado al proyecto debe tener un tutor, para tal caso, en la lista de estudiantes, hay una columna que contiene un botón donde se puede asignar un tutor al estudiante. Al presionar dicho botón, se abrirá una ventana modal la cual contiene una lista desplegable que contiene la lista de tutores asignados al proyecto, de los cuales el coordinador seleccionará uno para asignarlo al estudiante.

### **Subida de formularios**

La sección de subida de formularios fue implementada con el fin de que los usuarios puedan subir formularios previamente generados, impresos y firmados físicamente, para así poder tener un respaldo de dichos documentos. Para esta sección, se utilizó una implementación igual a la de archivos adjuntos del formulario FOR-UVS-04, es decir, se usaron campos de texto de tipo 'archivo', para posteriormente en el controlador manejar en que directorio del servidor debería ser almacenado dicho documento de acuerdo a la estructura de directorios mencionada en la sección 2.4.3.

### **Revisión de formularios por parte del coordinador**

Para la implementación de la revisión de formularios por parte del coordinador de pasantías de la carrera, se usó la relación de la base de datos entre formularios y usuarios, ya que por medio de ella se obtienen los formularios que el coordinador debe revisar, es decir, los formularios en los cuales él está como destinatario. De esta manera, solamente se debe validar luego que en la vista se muestren únicamente los formularios que tengan el estado de "Enviado".

### **Páginas del estudiante**

Como se detalló en la relación de estudiante con proyecto en la sección 2.3.2, el estudiante puede estar relacionado solamente a un único proyecto de pasantías comunitarias, y a un solo proyecto de pasantías pre-profesionales. Debido a esto, se decidió implementar una página para cada tipo de proyecto para el estudiante, dentro de las cuales se presenta la información del proyecto seguido de la lista de formularios que le corresponden al estudiante, como los son: el formulario FOR-UVS-01 y el formulario FOR-UVS-14.

### **Generación y descarga de archivos PDF**

Al ser llenados los formularios por medio del sistema, es necesario de todas formas que dichos documentos sean firmados físicamente. Por ende, surgió la necesidad de generar el formulario ya llenado y aprobado como documento PDF, para así poder imprimirlo y firmarlo. Para esto se utilizó la herramienta “Snappy PDF” [14], la cual sirve para la generación de archivos PDF a partir de una página HTML, esto se hace gracias a que internamente usa la herramienta “wkhtmltopdf” [15].

Con la intención de cumplir con el objetivo de generar un archivo PDF, se genera una vista HTML en la cual se presenta la información del formulario que se llenó (Modo editar) formateada de tal manera que pueda ser interpretada de manera correcta por Snappy (Modo ver) y el PDF se genere exitosamente.

### **Notificaciones**

La página de notificaciones consiste en un log de eventos que ocurren a lo largo del ciclo de vida del proyecto, o también denominadas notificaciones. Para implementarlo se creó un controlador específico para ello (NotificacionController), el cual contiene una función por cada tipo de notificación que exista en el sistema, de esta manera, cada una de estas funciones pueden ser llamadas desde los demás controladores en el momento indicado en que ocurra un evento para crear una notificación acerca del mismo, y que así quede registrado y pueda ser mostrado a los usuarios involucrados, junto con su fecha y hora de creación.

### **Información de usuario**

Existen formularios en los cuales se requiere cierta información del usuario que no puede ser obtenida por medio de un servicio web, debido a esto, se creó la página “Información de usuario” en la cual se pueden llenar ciertos campos de información necesaria. Con el fin de obligar al usuario a llenar su información faltante, la página de información de usuario es la primera página que se le muestra al usuario cuando este inicia

sesión por primera vez, y no se le permite usar ninguna de las demás opciones hasta que no haya llenado todos los campos requeridos. Para la implementación de esta restricción se creó el campo "información\_actualizada" en la tabla usuario, el cual indica si el usuario ya ha llenado toda su información. Esta propiedad es consultada en el middleware "RedirectIfNotInformacionActualizada", el cual pregunta si ese campo es igual a "true", y si es así lo deja continuar usando la opción que desee, caso contrario, se lo redirige a la misma página de información de usuario, mostrándole un mensaje de error.

## CAPÍTULO 3

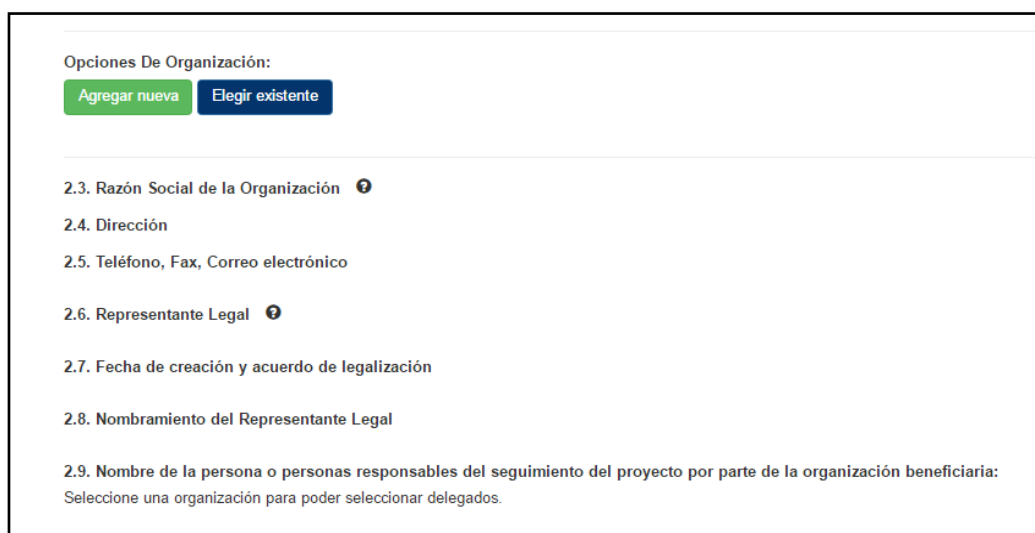
### 3. RESULTADOS.

Como resultados del proyecto se presenta una descripción de varios de los escenarios que se presentan al momento de usar el sistema, acompañada por capturas de pantalla que ilustran dichos escenarios. De esta manera se puede apreciar como luce el producto final, y como este se comporta ante ciertas situaciones.

Debido a que la creación de proyecto es un proceso bastante extenso, no se tomó en cuenta para esta presentación de resultados, todo el escenario completo. En lugar de eso, se presentarán solamente ciertas partes del proceso, es decir, ciertos campos del formulario FOR-UVS-04. Adicionalmente, se presentarán otros escenarios posteriores a la creación del proyecto.

#### 3.1 Selección de organización beneficiaria

Uno de los campos importantes en el formulario de creación de proyecto, es el de información de la organización beneficiaria, cuyos campos están inicialmente vacíos, y donde se presentan dos opciones: Agregar nueva, y Elegir existente, tal y como se muestra en la figura 3.1.



Opciones De Organización:

---

2.3. Razón Social de la Organización ⓘ

2.4. Dirección

2.5. Teléfono, Fax, Correo electrónico

2.6. Representante Legal ⓘ

2.7. Fecha de creación y acuerdo de legalización

2.8. Nombramiento del Representante Legal

2.9. Nombre de la persona o personas responsables del seguimiento del proyecto por parte de la organización beneficiaria:  
Seleccione una organización para poder seleccionar delegados.

**Figura 3.1: Sección de información de la organización beneficiaria.**



Al seleccionar la opción de “Agregar nueva” se muestra el panel flotante mostrado en la figura 3.2, con los campos que el usuario debe llenar para la organización.



Panel flotante de creación de organización con los siguientes campos:

- Razón Social de la Organización:** Corporación Viviendas del Hogar de Cristo
- Dirección:** Av. Casuarina, Coop. Sergio Toral, Mz. 101/Bloque 1/ Solar 1
- Teléfono:** (593) 390 4440 Ext. 101
- Fax:** (593) 390 4441
- Correo electrónico:** hdc@gmail.com
- Representante Legal:** P. Eduardo Vega S.J., Director General de Fundación Hogar de Cristo
- Fecha de creación y acuerdo de legalización:** Octubre 6, 1971
- Nombramiento del Representante Legal:** 2009
- Resumen:** (campo vacío)

Botones: Cancelar, Guardar

**Figura 3.2: Panel de creación de organización.**

Luego de haber llenado la información y haber presionado el botón guardar, se verá reflejada en el formulario la información de la organización que se ha creado, tal y como se puede apreciar en la figura 3.3.

Opciones De Organización:

[Agregar nueva](#) [Elegir existente](#) [Quitar](#)

Convenio O Carta De Compromiso:

[Ver convenio adjunto](#)

[Seleccionar archivo](#) No se eligió archivo

---

2.3. Razón Social de la Organización ⓘ  
Corporación Viviendas del Hogar de Cristo

2.4. Dirección  
Av. Casuarina, Coop. Sergio Toral, Mz. 101/Bloque 1/ Solar 1

2.5. Teléfono, Fax, Correo electrónico  
(593) 390 4440 Ext. 101  
(593) 390 4441  
hdc@gmail.com

2.6. Representante Legal ⓘ  
P. Eduardo Vega S.J., Director General de Fundación Hogar de Cristo

2.7. Fecha de creación y acuerdo de legalización  
Octubre 6, 1971

2.8. Nombramiento del Representante Legal  
2009

2.9. Nombre de la persona o personas responsables del seguimiento del proyecto por parte de la organización beneficiaria:

[Crear nuevo delegado](#)

**Figura 3.3: Información llena de la organización.**

### 3.2 Creación de objetivos del proyecto

Otro campo importante en el formulario de creación de proyecto, corresponde a los objetivos específicos, ya que, al ser agregados en la sección de objetivos específicos por el usuario, también deben ser agregados automáticamente por el sistema a la sección de matriz de marco lógico.

La figura 3.4 muestra la sección donde se deben agregar objetivos específicos, basta con escribir el objetivo en el campo de texto respectivo y presionar “Agregar objetivo específico”, de esta manera, los objetivos se verán reflejados en la tabla que se encuentra debajo del campo de texto junto con las opciones de editar y eliminar, tal y como se puede apreciar en la figura 3.5.

**6. OBJETIVOS DEL PROYECTO (General Y Especifico) ①**

**Objetivo General**

Estructurar un programa de educación digital para que las mujeres del semillero de emprendimientos de Hogar de Cristo generen el marketing online adecuado de sus productos o servicios para que fortalezcan su emprendimiento.

**Lista de Objetivos Especificos**

Crear un manual de marca para el desarrollo una página web Agregar Objetivo específico

Descripción Objetivo	Accion

**Figura 3.4: Sección de objetivos específicos.**

**Lista de Objetivos Especificos**

Ingrese objetivo específico Agregar Objetivo específico

Descripción Objetivo	Accion
Crear un manual de marca para el desarrollo una página web	✎ -
Dictar los talleres de capacitación de las mujeres emprendedores de Hogar de Cristo.	✎ -
Elaborar el material didáctico y grabación de videos para impartir talleres.	✎ -

**Figura 3.5: Tabla de objetivos específicos.**

Como se mencionó previamente, estos objetivos serán agregados automáticamente en la sección de matriz de marco lógico, donde se podrán añadir actividades relacionadas a cada objetivo específico, la figura 3.6 ilustra la matriz de marco lógica resultante luego que se agregaron objetivos específicos.

7. MATRIZ DE MARCO LÓGICO

Objetivos Narrativos	Indicadores Verificables	Medios de verificación	Supuestos
FIN			
PROPÓSITO Estructurar un programa de educación digital para que las mujeres del semillero de emprendimientos de Hogar de Cristo generen el marketing online adecuado de sus productos o servicios para que fortalezcan su emprendimiento.			
Crear un manual de marca para el desarrollo una página web			
Actividades: +			
Dictar los talleres de capacitación de las mujeres emprendedores de Hogar de Cristo.			
Actividades: +			
Elaborar el material didáctico y grabación de videos para impartir talleres.			
Actividades: +			

**Figura 3.6: Matriz de marco lógico con los objetivos agregados automáticamente.**

### 3.3 Creación de actividades

Como se mencionó en la sección anterior, la matriz de marco lógico permite ingresar las actividades del proyecto asociadas a cada objetivo específico, para realizar esto basta con dar click en el botón con el símbolo “+” dentro de la sección correspondiente al objetivo específico deseado, mostrándose así un campo de texto en el que se debe escribir el nombre de la actividad, la figura 3.7 muestra el resultado de agregar nuevas actividades en la matriz de marco lógica.

**Figura 3.7: Matriz de marco lógico con actividades agregadas.**

Similar a la sección de objetivos específicos, al agregar las actividades en la matriz de marco lógico, estas se agregarán automáticamente a la sección de cronograma mostrada en la figura 3.8, donde el usuario podrá ingresar las fechas de inicio y fin de cada actividad, así como también su duración en horas.

**9. CRONOGRAMA**

Seleccionar archivo No se eligió archivo

Crear un manual de marca para el desarrollo una página web

Actividades	Horas	Fecha de Inicio - Fin
1.1 Entender los antecedentes de la empresa o entidad corporativa.		03/09/2016 - 03/09/2016
1.2 Establecer tipografías, gráficos o demás anexos.		03/09/2016 - 03/09/2016
1.3 Establecer los patrones del color.		03/09/2016 - 03/09/2016

**Figura 3.8: Sección de cronograma con actividades agregadas automáticamente.**

### 3.4 Asignación de tutores

Al ser aprobado el formulario FOR-UVS-04 de un proyecto, el coordinador de la carrera tiene la opción de agregar tutores a dicho proyecto entrando a la opción “Asignar tutores” dentro de las opciones de ese proyecto, la figura 3.9 muestra más a detalle el contenido de esta opción. Para añadir un tutor, simplemente se lo debe buscar por medio del nombre o apellido y seleccionar el tutor deseado para finalmente dar click en “Agregar”.

The screenshot shows a web interface for assigning tutors. The page title is 'Estructuración de un programa de educación digital para el "Semillero de Emprendimientos" de Hogar de Cristo.' and the status is 'Estado: Aprobado'. On the left, there is a sidebar with options: 'Flujo del proyecto', 'Detalles del Proyecto', 'Asignar Tutores' (selected), and 'Asignar Estudiantes'. The main area is titled 'Lista de Tutores' and contains a search input field with 'Sergio' entered, an 'Agregar' button, and a dropdown menu showing 'Sergio Roberto Wong Diaz' as the selected option. Below the dropdown is a table with columns for 'Cédula', 'Carrera', and 'Correo'.

**Figura 3.9: Búsqueda de tutor por nombre.**

Al ser agregado correctamente, el tutor aparecerá en la tabla de tutores junto con el botón de eliminar, con el cual se podrá quitar tutores del proyecto, la figura 3.10 muestra un ejemplo de un tutor ya asignado con la opción de poder eliminarlo.

The screenshot shows the 'Lista de Tutores' interface after a tutor has been added. The page title and status are the same as in Figure 3.9. The search input field now contains 'Nombre profesor' and the 'Agregar' button. Below the search field is a table with the following data:

Nombre	Cédula	Carrera	Correo	Acción
Sergio Roberto Wong Diaz	0977856245	Ingeniería en Telemática	tutor@espol.edu.ec	✖

**Figura 3.10: Tutor agregado correctamente.**

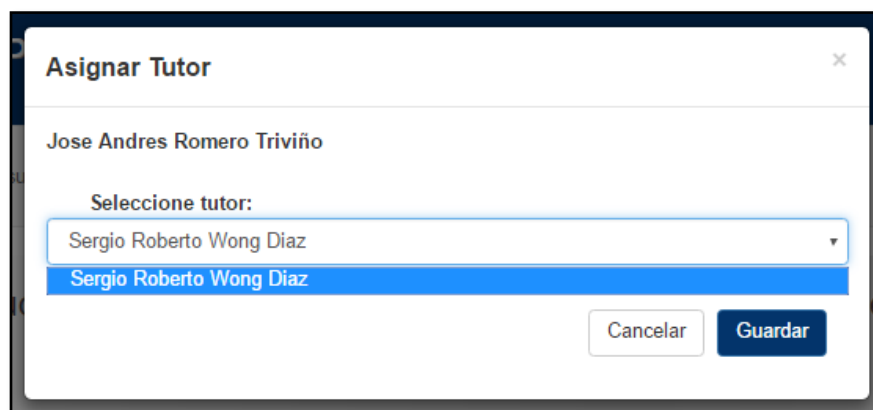
### 3.5 Asignación de estudiantes

Para la asignación de estudiantes se sigue un proceso similar al de la asignación de tutores, con la diferencia de que al estudiante se lo busca por medio de su número de matrícula, y se lo agrega de igual manera con el botón “Agregar”. La figura 3.11 muestra más a detalle la opción de “asignación de estudiantes”.



**Figura 3.11: Estudiante agregado correctamente.**

Luego que un estudiante es agregado, se le debe asignar un tutor, y para esto, se debe dar click en el link “Sin tutor asignado”, con lo cual se mostrará el panel flotante mostrado en la figura 3.12, donde se podrá escoger el tutor que dará seguimiento al estudiante,



**Figura 3.12: Selección de tutor para ser asignado al estudiante.**

### 3.6 Flujo del proyecto

Otro de los resultados que vale la pena presentar es el gráfico de flujo del proyecto mostrado en la figura 3.13, el cual da una vista general del avance y estado del proyecto. Como se explicó en secciones anteriores, el gráfico de flujo del proyecto está compuesto de bloques que representan cada uno de los formularios que forman parte del ciclo de vida del proyecto, los mismos que cambian de color dependiendo del estado de los distintos formularios.

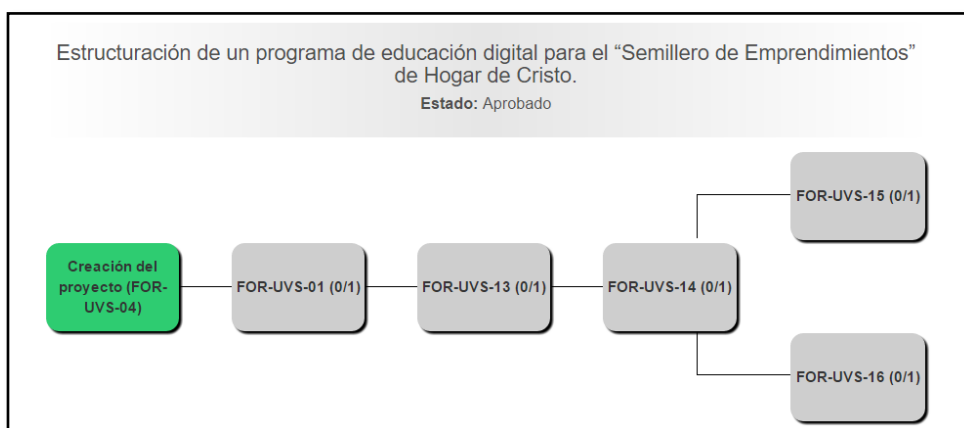


Figura 3.13: Flujo de un proyecto aprobado con un estudiante.

### 3.7 Pruebas de usabilidad

Para probar el sistema, se realizaron pruebas de usabilidad con usuarios de diferentes roles, director, tutor y estudiante, a quienes se les pidió completaran ciertas tareas, las más importantes para cada tipo de usuario, en el sistema. Las tareas consistieron en lo siguiente.

La tarea propuesta al **director**, consistió en crear un nuevo proyecto en el sistema, siendo esta la tarea más importante que el director puede realizar.

Las tareas propuestas al **tutor**, consistieron en el llenado y aprobación de formularios de los estudiantes, estas son las tareas más importantes que el tutor puede realizar, ya que de esta manera, se da el seguimiento al estudiante.

Las tareas propuestas al **estudiante**, consistieron en el llenado de los formularios FOR-UVS-01 y FOR-UVS14, y luego enviarlos a revisión a su tutor respectivo.



Una vez que los usuarios completaron sus tareas, se proporcionó una encuesta de satisfacción sobre el uso del mismo, la encuesta se detalla a continuación en la siguiente sección.

### 3.7.1 Encuesta

La encuesta de satisfacción, consistió en una serie de preguntas donde el usuario calificó que tan fácil le resultó usar y completar sus tareas en el sistema. La tabla 1 muestra a detalle la descripción de las preguntas contenida en la encuesta, donde el usuario podía calificar en la escala del 1 al 10, siendo 1 “Muy en desacuerdo” y 10 “Muy de acuerdo”.

Número pregunta	Descripción
1	Estoy satisfecho con el fácil uso del Sistema.
2	Completé mi trabajo de manera efectiva.
3	Completé mi trabajo rápidamente.
4	Completé de manera eficiente mi trabajo.
5	Me siento cómodo usando el Sistema.
6	Fue fácil aprender a usar el Sistema.
7	El sistema proporciona mensajes de error.
8	Puedo recobrar fácilmente tareas si cometo un error.
9	La información provista del sistema es clara.
10	Es fácil encontrar la información que necesitaba.

**Tabla 1: Descripción de preguntas de la encuesta de satisfacción**

### 3.7.2 Resultados

Finamente, se realizaron pruebas con 2 estudiantes, 2 tutores y 1 director. La figura 3.14 muestra una gráfica de los resultados de la encuesta, el eje horizontal representa las preguntas de la encuesta, mientras que el eje vertical representa la calificación del 1 al 10 de la pregunta específica.

Como resultado, se tienen las curvas de satisfacción, donde cada curva representa la facilidad que tuvo cada tipo de usuario al usar el sistema.

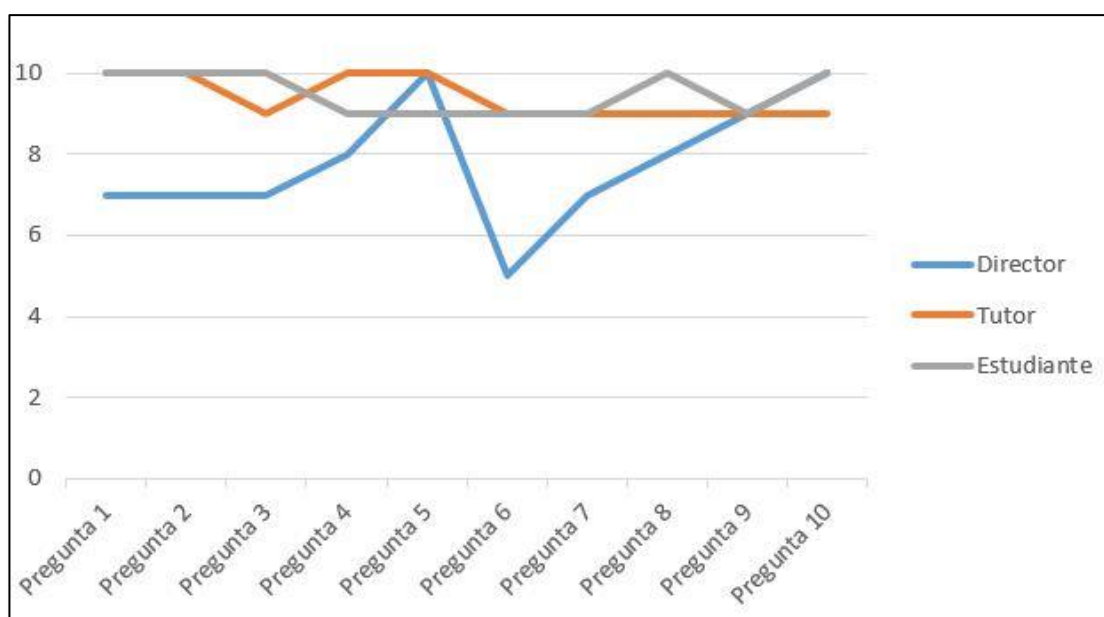


Figura 3.14: Resultado de encuesta de satisfacción

## CONCLUSIONES Y RECOMENDACIONES

Según los resultados obtenidos por la encuesta de satisfacción, se concluye que, el sistema resulta de fácil uso para tutor y estudiante, sin embargo, para un director, el sistema no resulta tan amigable, debido a que le toma demasiado tiempo en completar su principal tarea que es la creación de un proyecto.

El sistema desarrollado, agiliza el proceso de creación de formularios, debido a que la mayor parte del contenido de los mismos, se completa automáticamente, mientras que la información nueva que se requiere para un tipo de formulario en particular, se la maneja dinámicamente.

El mecanismo aplicado para la aprobación de los formularios, es el apropiado para la gestión de los mismos, ya que con un simple click, se aprueba o rechaza, cada vez que se rechaza un formulario, se tiene la habilidad para escribir comentario explicando la razón del rechazo.

El framework utilizado para el desarrollo del sistema, demostró ser el adecuado debido a su nivel de abstracción, flexibilidad y robustez, además, existe mucha información en la web acerca de este framework lo cual facilitó al momento de la implementación.

Para documentos PDF, el paquete “snappy” que se utilizó, resultó ser el mejor al momento de generar documentos a partir de etiquetas HTML, es el paquete que mejor arma el documento, sobre todo dando un mejor resultado visual cuando se tratan de tablas, cosa que otros paquetes o componentes que se trató usar, no eran bien manejados.

El sistema puede ser mejorado para manejar dinámicamente otros tipos de formularios, por ejemplo, el formulario FOR-UVS-08 del presupuesto, en el actual sistema se la trabaja como un documento adjunto, pero para una futura versión del sistema, puede ser manejado dinámicamente por el usuario dentro del sistema.

Algunos campos que deben ser llenados por el usuario al crear formularios, deben ser escritos por él mismo, sin embargo, existen campos los cuales pueden ser precargados ingresando dicha información en la base de datos, tal es el caso de

datos como ciudad, cantón, provincia, entre otros. De esta manera, el usuario puede simplemente seleccionar una opción sin necesidad de escribir nada en dichos campos.

Todo usuario que vaya usar el sistema, es recomendable que tenga una breve explicación sobre el uso del mismo, sin embargo, una futura mejora del sistema, puede evitar esto, desarrollándola de una forma mucho más intuitiva.

## BIBLIOGRAFÍA

- [1] Información General sobre NUXEO. Disponible en: <https://es.wikipedia.org/wiki/Nuxeo>
- [2] Sitio Oficial de NUXEO. Disponible en: <http://www.nuxeo.com/>
- [3] Sitio Oficial de Alfresco. Disponible en: <https://www.alfresco.com/>
- [4] Información general sobre ECOUNT ERP. Disponible en: <http://owwww.ecounterp.com/es/product/web-based/product-saas-erp-overview.aspx?p=T2>
- [5] Características sobre el módulo de aprobación incluido en el software ECOUNT ERP. Disponible en: <http://www.ecounterp.com/es/product/collaborate/product-e-approval.aspx?p=T4>
- [6] Características principales de Laravel. Disponible en: <https://platzi.com/blog/laravel-framework-php>
- [7] Documentación Oficial sobre Laravel. Disponible en: <https://laravel.com/docs/5.2>
- [8] ¿Por qué utilizar MySQL? Disponible en: <https://www.mysql.com/why-mysql>
- [9] Sitio Oficial phpMyAdmin. Disponible en: <https://www.phpmyadmin.net>
- [10] ¿Qué es servidor Apache? Disponible en: <http://culturacion.com/que-es-apache>
- [11] Características de Apache. Disponible en: <http://www.digitallearning.es/blog/apache-servidor-web-configuracion-apache2-conf>
- [12] Date Range Picker for Bootstrap. Disponible en: <http://github.com/dangrossman/bootstrap-daterangepicker>
- [13] CKEditor. Disponible en: <http://ckeditor.com/>
- [14] Snappy PDF/Image Wrapper for Laravel 5 and Lumen 5.1. Disponible en: <https://github.com/barryvdh/laravel-snappy/>
- [15] Wktohtml Home Page. Disponible en: <http://wkhtmltopdf.org/index.html>