



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

“DESARROLLO DE UN SISTEMA WEB PARA LA GESTIÓN DE
FORMULARIOS DE PROYECTOS DE VÍNCULOS CON LA
SOCIEDAD”

INFORME DE MATERIA INTEGRADORA

Previo a la obtención del Título de:

INGENIERO/A EN COMPUTACIÓN

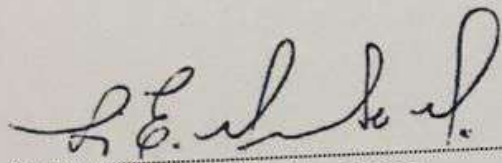
JANNA ELIZABETH BARRETO CASTRO

EDDIE STEVEN HURTADO HERRERA

GUAYAQUIL – ECUADOR

AÑO: 2017

TRIBUNAL DE EVALUACIÓN



Ph.D. Luis Mendoza

PROFESOR EVALUADOR

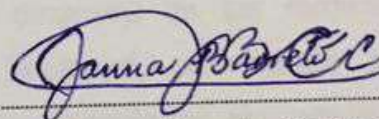


Ph.D. Dennis Romero

PROFESOR EVALUADOR

DECLARACIÓN EXPRESA

"La responsabilidad y la autoría del contenido de este Trabajo de Titulación, me(nos) corresponde exclusivamente; y doy(damos) mi(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"



Janna Barreto Castro



Eddie Hurtado Herrera

RESUMEN

La Unidad de Vínculos con la Sociedad (UVS) de ESPOL, es la encargada de manejar los convenios entre la universidad y las organizaciones sin fines de lucro, de los cuales pueden nacer proyectos para realizar prácticas pre-profesionales.

Previamente se realizó un sistema web capaz de agilizar el proceso de llenado de formularios pertenecientes a proyectos de prácticas comunitarias, dando facilidades a los docentes y estudiantes participantes; pero lamentablemente no pudo ser usado debido a que los formularios de UVS están en constante actualización, causando que los formularios del sistema queden desactualizados.

La presente propuesta considera la actualización dinámica de los formularios relacionados con proyectos de Vínculos con la Sociedad, permitiendo a los coordinadores de la UVS actualizar el formato de los formularios sin requerir de personal técnico para tal efecto. Para ello se aplica la estrategia de Tablas Clave-Valor en la base de datos, que consiste en colocar todos los campos nuevos de un formulario como registros en una tabla (tabla Campos), mientras que los valores al llenar dichos campos, serán colocados como registros en otra tabla ligada a la instancia del formulario (tabla Valores); de esta manera se puede tener una cantidad ilimitada de campos en un formulario.

Como resultado se obtuvo un sistema web que ofrece mayor facilidad para realizar las actividades a cargo de los docentes y estudiantes involucrados en el proceso de prácticas comunitarias. El sistema web incluye ahora la gestión de programas comunitarios y el manejo de versiones de formularios de prácticas comunitarias, reduciendo el costo de mantenimiento del mismo.

ÍNDICE GENERAL

TRIBUNAL DE EVALUACIÓN	ii
DECLARACIÓN EXPRESA	iii
RESUMEN	iv
ÍNDICE GENERAL.....	v
CAPÍTULO 1	7
1. ACTUALIZACIONES CONTINUAS EN EL CONTENIDO DE LOS FORMULARIOS DE PRÁCTICAS PRE-PROFESIONALES.	7
1.1 Antecedentes.....	7
1.2 Propuesta Anterior.....	10
1.3 Problemática.....	12
1.4 Objetivos.....	13
CAPÍTULO 2.....	14
2. ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE FORMULARIOS CON CONTENIDO DINÁMICO.	14
2.1 Solución Propuesta	14
2.2 Soluciones Similares	15
2.2.1 Google forms	15
2.2.2 FormBakery	15
2.2.3 WordPress.....	15
2.3 Ambiente de Programación Web.....	16
2.3.1 PHP	16
2.3.2 Laravel.....	16
2.3.3 MySQL.....	17
2.3.4 Apache	17
2.4 Diseño	18
2.4.1 Diagramas UML.....	18
2.4.2 Diseño de la base de datos	20

2.5	Implementación	24
2.5.1	Actualización de Formularios.....	24
CAPÍTULO 3.....		32
3.	RESULTADOS OBTENIDOS.....	32
3.1	Esquema del Funcionamiento del Sistema.....	32
3.1.1	Propuesta Anterior vs. Propuesta Actual.....	32
3.1.2	Gestión de Programas.....	33
3.2	Actualización de Formularios.....	34
3.2.1	Menú Principal de los Formularios	34
3.2.2	Nueva Versión y Versiones Anteriores	35
3.2.3	Activación de Nueva Versión.....	36
3.3	Programas y Proyectos	37
3.3.1	Menú Principal de Programas	37
3.3.2	Asignación de Director a Proyectos Comunitarios.....	39
3.3.3	Edición de Proyecto Asignado.....	40
3.3.4	Vinculación de Nuevo Proyecto a Programa Existente	41
3.3.5	Visualización del FOR-UVS-20 entre los detalles del proyecto.....	42
3.4	Sistema Web en Producción	42
3.4.1	Apache	42
3.4.2	Pruebas del Flujo del Proceso.....	43
CONCLUSIONES Y RECOMENDACIONES		48
BIBLIOGRAFÍA.....		49

CAPÍTULO 1

1. ACTUALIZACIONES CONTINUAS EN EL CONTENIDO DE LOS FORMULARIOS DE PRÁCTICAS PRE-PROFESIONALES.

Este capítulo tiene como objetivo mostrar la información acerca de cómo surgió el proyecto, en qué problemas se enfocó el proyecto y el alcance del mismo, definidos más adelante mediante los objetivos.

En la sección 1.1 se detalla información acerca del problema que originó el nuevo proceso de prácticas pre-profesionales, y los malestares causados en las personas involucradas en dicho proceso. Seguido, un resumen correspondiente a una solución propuesta con anterioridad en la sección 1.2. En la sección 1.3 se menciona cuáles son los problemas y por qué se decidió enfocar el proyecto en ellos. Y, por último, la sección 1.4 donde se definen los objetivos de este proyecto.

1.1 Antecedentes

Según el Art. 87 de la Ley Orgánica de Educación Superior (LOES), previo a la obtención del título, los y las estudiantes deberán acreditar servicios a la comunidad mediante prácticas pre-profesionales debidamente monitoreadas [1]. La Escuela Superior Politécnica del Litoral (ESPOL), a partir del año 2016, con la finalidad de cumplir con este nuevo régimen, procedió a redefinir sus requisitos de graduación. De esta manera, 160 horas del total de horas de prácticas pre-profesionales corresponderían a las actividades de servicio comunitario; lo cual consiste en desarrollar proyectos que ayuden a resolver problemas de la comunidad. Todo esto, mediante convenios contraídos entre ESPOL y organizaciones sin fines de lucro [2].

Para poder llevar un adecuado control de las actividades realizadas por los estudiantes, se incorporó un conjunto de formularios que deben ser llenados por los estudiantes, tutores, directores y coordinadores a cargo del proyecto; tanto de prácticas pre-profesionales comunitarias como de prácticas pre-profesionales

empresariales. El orden en el cual deben ser llenados y aprobados estos formularios, dependen del flujo establecido por la Unidad de Vínculos con la Sociedad (UVS), (ver Figura 1.1). Como este procedimiento era totalmente nuevo y no se contaba con un sistema informático que permita la administración de documentos, se generaron algunos malestares en todas las personas involucradas en el proceso:

- Confusión al momento de llenar los formularios, debido a falta de información.
- Desaprobaciones continuas de los formularios, debido a errores de llenado.
- Retrasos en el proyecto, debido a aprobaciones pendientes de los formularios.
- Desperdicio de tiempo.

Algunos de los formularios que maneja UVS varían de acuerdo al tipo de práctica pre-profesional. A continuación, se muestran los formularios usados en las prácticas pre-profesionales comunitarias:

➤ **Formularios para Prácticas Pre-profesionales Comunitarias**

- FOR-UVS-20 (Formulación de Programa)
- FOR-UVS-04 (Formulación de Proyecto)
- FOR-UVS-01 (Plan de Actividades)
- FOR-UVS-13 (Seguimiento de Actividades)
- FOR-UVS-14 (Informe Final Practicante)
- FOR-UVS-15 (Informe Final de Prácticas en Organización)
- FOR-UVS-16 (Informe Final del Tutor de Prácticas)

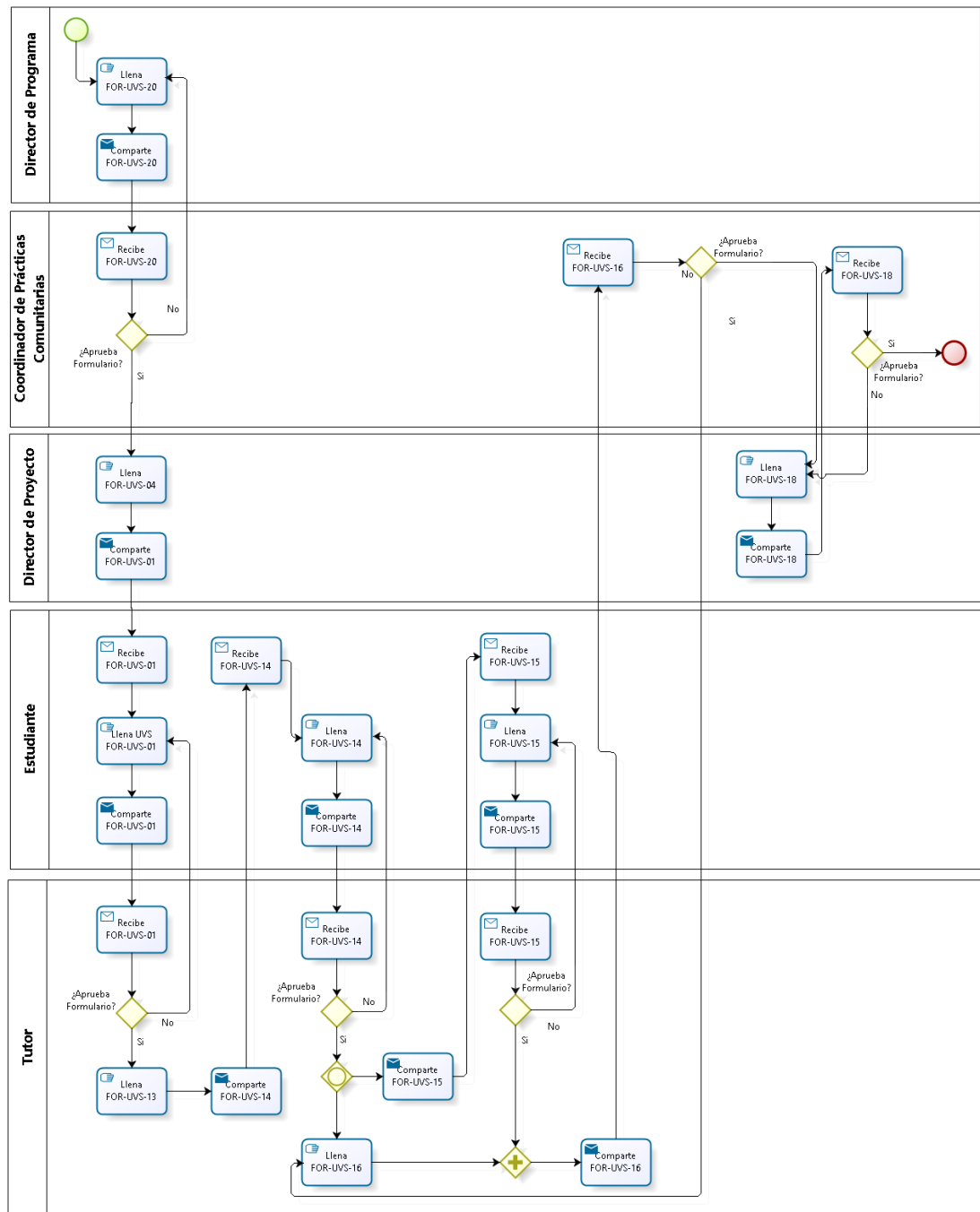


Figura 1.1: Proceso Actual de Prácticas Pre-profesionales Comunitarias

1.2 Propuesta Anterior

Con la finalidad de dar solución a los problemas antes mencionados, estudiantes implementaron una aplicación web que sería capaz de gestionar las actividades en los diversos proyectos de vínculos con la sociedad y, además, permitiría seguir de forma automática el flujo del proceso.

Para lograr comprender el comportamiento del sistema y la interacción de sus usuarios, se realizó un diagrama de caso de uso del sistema (ver Figura 1.2). En este diagrama se muestra todos los roles de usuario que soporta el sistema; así como también las acciones que cada uno de ellos puede realizar mediante el sistema.

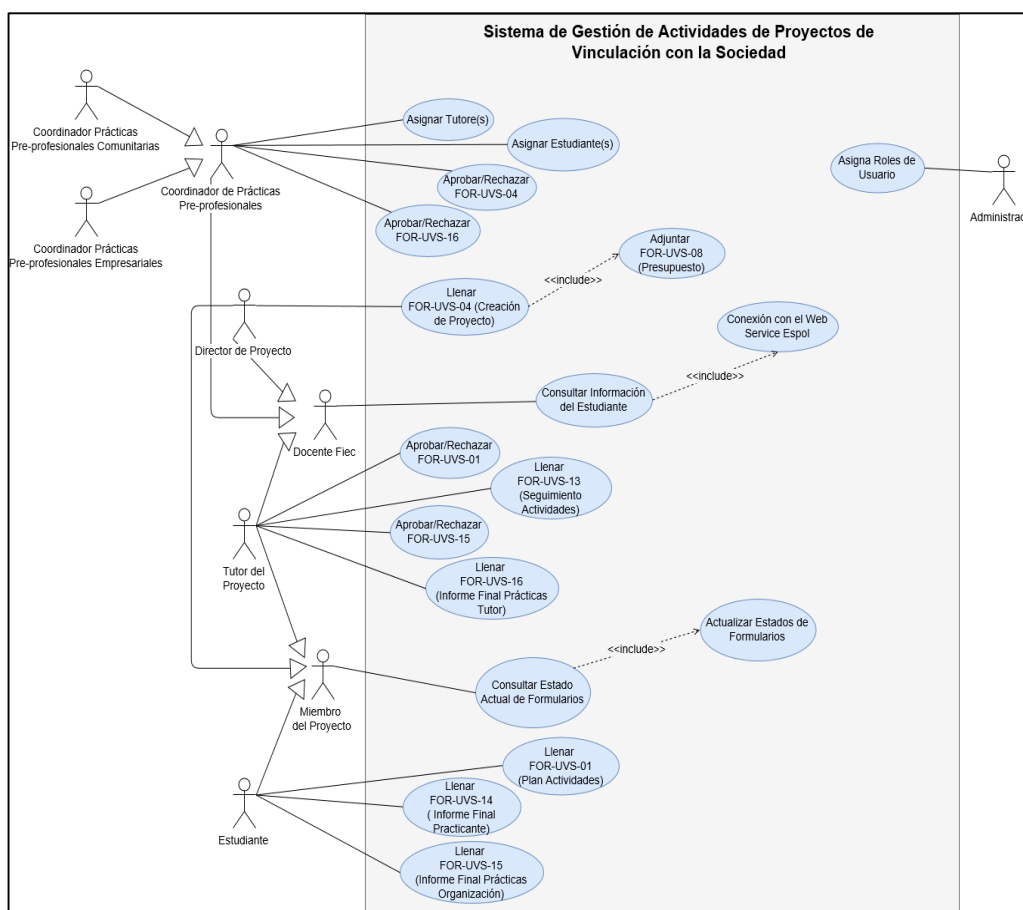


Figura 1.2: Diagrama de Caso de Uso del Sistema Propuesto [2].

En la Figura 1.2 se puede notar qué formularios se manejan mediante la aplicación web. En ese entonces, UVS manejaba el mismo formulario de creación de proyecto (FOR-UVS-04) en ambos tipos de prácticas pre-profesionales.

Cada creación de proyecto involucra dar seguimiento a 5 formularios más que se deben ir llenando de acuerdo con el orden establecido por UVS, (ver Figura 1.1). Se necesita de la aprobación del formulario previo para dar paso al siguiente. Mediante la aplicación web se logra hacerlo de forma automática, y a su vez, controlar el estado de los formularios (pendiente, aprobado o rechazado). De esta manera se crea el flujo que se muestra en la Figura 1.3.

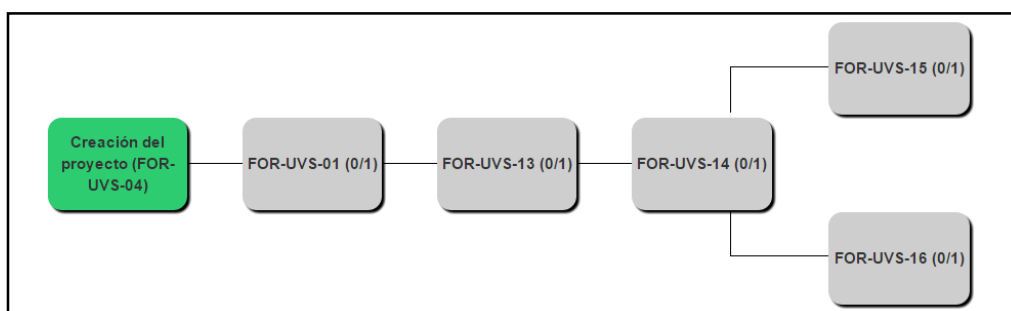


Figura 1.3: Gráfico de Flujo del Proyecto [2].

Para la implementación del sistema se usó Laravel, Php, MySQL; además se tomó como referencia algunos sistemas existentes, tales como: Nuxeo, Ecount, y Alfresco, que permiten la administración de contenido y planificación de recursos [2]. La Figura 1.4 representa el esquema básico de la solución propuesta anteriormente.

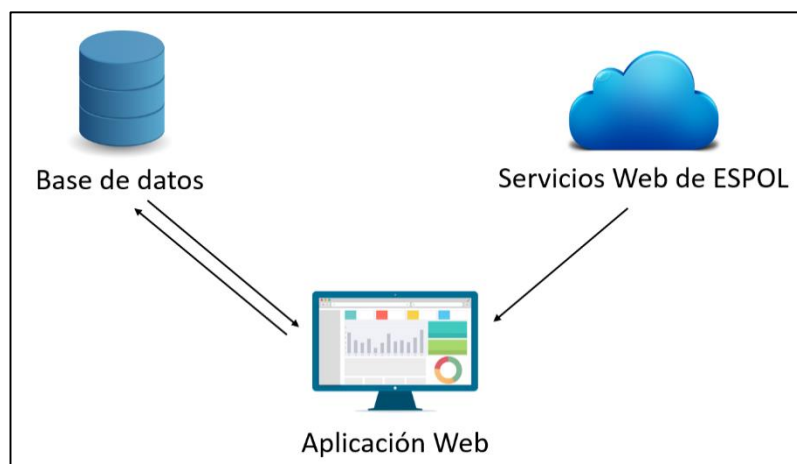


Figura 1.4: Esquema Básico del Sistema [2].

1.3 Problemática

UVS es la unidad encargada de manejar los convenios adquiridos entre la universidad y las organizaciones sin fines de lucro. A su vez, se encarga de generar los formularios correspondientes de cada proyecto, que deben ser llenados, como se mencionó anteriormente, por los estudiantes, tutores y directores involucrados en un determinado proyecto.

A pesar de que se desarrolló un sistema capaz de agilizar el proceso de prácticas comunitarias, este no pudo ser usado debido a que UVS actualizó los formularios en el preciso momento en que el sistema se estaba configurando para quedar en producción. De esta manera surge un problema más: la actualización continua de los formularios por parte de UVS, un problema que, de no resolverse, haría que los mencionados anteriormente continúen.

Debido a que en la actualidad UVS ya cuenta con un sistema capaz de gestionar todo el proceso de las prácticas pre-profesionales empresariales, se decidió enfocar el proyecto al proceso de prácticas pre-profesionales comunitarias, cuyos problemas son:

- Actualización continua de formularios de prácticas pre-profesionales comunitarias.

- Propuesta anterior no cuenta con la gestión de *programas comunitarios*, son los formularios de donde parte toda la información y permite dar flujo a todo el proceso.

1.4 Objetivos

Luego de tener claro cuáles son los problemas que enfrentan las diferentes personas involucradas en el proceso de prácticas comunitarias, se definen los siguientes objetivos:

➤ **Objetivo General**

- Permitir el manejo de versiones de formularios de prácticas comunitarias y la gestión de programas comunitarios, con la finalidad de que sea a partir de éstos, que se generen los proyectos con la información correspondiente al programa al que se vincula.

➤ **Objetivos Específicos**

- Realizar una revisión del sistema actual con la finalidad de asegurar su correcto funcionamiento.
- Evaluar las funcionalidades de aplicaciones web que estén orientadas a la creación de formularios dinámicos.
- Diseñar prototipos low-fi.
- Definir los elementos HTML que podrán formar parte de la creación de los formularios.
- Rediseñar el esquema de la base de datos del sistema actual.
- Evaluar el sistema en producción.

CAPÍTULO 2

2. ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE FORMULARIOS CON CONTENIDO DINÁMICO.

La solución propuesta se basa en aplicaciones similares que han aplicado empresas muy conocidas y exitosas, como Google. El concepto de dinamismo aplicado a los formularios fue el enfoque inicial de este proyecto, y el segundo enfoque, consiste en incorporar los formularios de programas a fin de proveer un sistema mucho más completo para los usuarios involucrados en el proceso de prácticas comunitarias.

Este capítulo contiene una breve explicación de la solución propuesta en la sección 2.1, en base a soluciones similares que plantean el uso de formularios dinámicos, descritos en la sección 2.2. El ambiente de programación, los diseños UML y el rediseño del modelo de la base de datos en el cual se incorporan tanto los programas como la estrategia clave-valor para desarrollar la propuesta, se encuentran en las secciones 2.3 – 2.5. Y finalmente, el método Drag&Drop usado en la implementación de los formularios dinámicos, en la sección 2.6.

2.1 Solución Propuesta

Debido a las continuas actualizaciones en el contenido de los formularios de prácticas pre-profesionales, se optó por usar el dinamismo, de manera que permita agregar o quitar campos de un formulario, generando una nueva versión del mismo.

Hacer dinámico un sistema informático, se refiere a hacer que éste sea configurable y se ajuste a las necesidades del usuario o empresa; de esta manera, se garantiza que el sistema pueda adaptarse ante los diversos cambios que surjan, permitiendo que el costo de mantenimiento sea mínimo [3].

El uso de dinamismo ha sido una técnica muy utilizada en aplicaciones web sencillas que permiten crear encuestas, currículos, formularios de inscripción, etc.

2.2 Soluciones Similares

A continuación, se mencionan algunos ejemplos que proveen una solución similar a la que se decidió usar en este proyecto:

2.2.1 Google forms

Es una aplicación de Google que permite crear formularios de encuestas. Provee una variedad de estilos y una gran cantidad de opciones de preguntas, desde las de tipo test hasta las listas desplegables con escalas numéricas [4]. Todo esto, para que el usuario pueda personalizarlo de acuerdo a su preferencia.

2.2.2 FormBakery

Al igual que Google forms, éste facilita la creación formularios, pero con la diferencia de que permite arrastrar y soltar cada componente para construirlos con un estilo aún más personalizado. Además, provee la opción de exportar el código del formulario para poderlo usar en nuestras propias aplicaciones web [5].

2.2.3 WordPress

WordPress va un poco más allá de la creación de formularios webs. Pues es un software de código abierto que permite la gestión de contenidos, también conocido como un Content Management System (CMS). Está mucho más orientado a la creación de fantásticos blogs y páginas web [6].

WordPress está basado en Hypertext Preprocessor (PHP) y My Structured Query Language (MySQL). Ha tenido una gran acogida debido a sus grandes características [6]: ponerle atención a la estética, estándares web y usabilidad.

2.3 Ambiente de Programación Web

2.3.1 PHP

Es el lenguaje de programación de código abierto, muy usado en el desarrollo web. Es tan popular, que grandes “jugadores” de la web como Wikipedia y Yahoo!, lo utilizan en sus sitios web; al igual que algunos referentes a la edición de contenido como WordPress y Drupal. PHP también es utilizado por los conocidos servidores Apache [8].

Se decidió usar este lenguaje de programación del lado del servidor por las siguientes razones [8]:

- ✓ La curva de aprendizaje es muy corta.
- ✓ Permite implementar programación orientada a objetos.
- ✓ El código PHP se ejecuta desde el servidor, por lo que no se podrá tener acceso a él desde el navegador. Así hay mayor seguridad.
- ✓ Permite el desarrollo de aplicaciones web dinámicas, a partir de los datos almacenados en una base de datos.
- ✓ Gran funcionamiento con bases de datos relacionales de código abierto, como MySQL.

2.3.2 Laravel

Es un framework muy utilizado por los desarrolladores PHP, debido a la extensa documentación que proporciona. Esto último es lo que hace que Laravel sea más valorada, puesto que permite que los desarrolladores encuentren rápidamente soluciones a los errores que cometen [8].

Su objetivo es permitir el uso de una sintaxis elegante y expresiva para crear código de forma sencilla. Entre las principales características, se tiene [9]:

- ✓ Sistema de ruteo
- ✓ Motor de plantilla, Blade.
- ✓ Manejador de dependencias, Composer.

- ✓ Soporte para el caché.
- ✓ Soporte para Modelo-Vista-Controlador (MVC).

2.3.3 MySQL

Es un sistema de gestión de bases de datos relacional. Es debido a su rendimiento, confiabilidad y facilidad de uso, que se ha convertido en la principal opción de bases de datos para muchos sitios web, como Google, WordPress, Facebook, Twitter, y Youtube [10].

MySQL es muy rápido en la lectura de datos, pero lento en cuando a la alta concurrencia en la modificación de datos. Mientras que, en las aplicaciones web, ocurre lo contrario, la concurrencia en modificación de datos es baja, pero el entorno es intensivo en lectura de datos, lo que hace que MySQL se convierta en el gestor de bases de datos ideal para este tipo de aplicaciones [11].

- **MySQL Workbench**

Es una herramienta visual que ofrece el modelado de datos, desarrollo de SQL y herramientas de administración completas para la configuración de servidores, administración de usuarios, copias de seguridad, etc [12].

2.3.4 Apache

Es un servidor web HTTP de código abierto. Es usado principalmente para enviar páginas web estáticas y dinámicas en la World Wide Web.

Se decidió usarlo en este proyecto, de forma local, con el propósito de pre-visualizar y probar código a medida que se avanzaba con la implementación.

Las principales ventajas de Apache son las siguientes:

- ✓ Modular
- ✓ Código abierto
- ✓ Multiplataforma

- ✓ Extensible
- ✓ Popular, lo que facilita obtener ayuda o soporte.

2.4 Diseño

2.4.1 Diagramas UML

Diagrama de Caso de Uso

La elaboración de un diagrama de caso de uso fue imprescindible para este proyecto, debido a que se necesitaba comprender las nuevas funcionalidades que se requerían incorporar al sistema actual; así como también los usuarios que intervendrían en estos nuevos cambios. De esta manera, se generó el diagrama de la Figura 2.1.

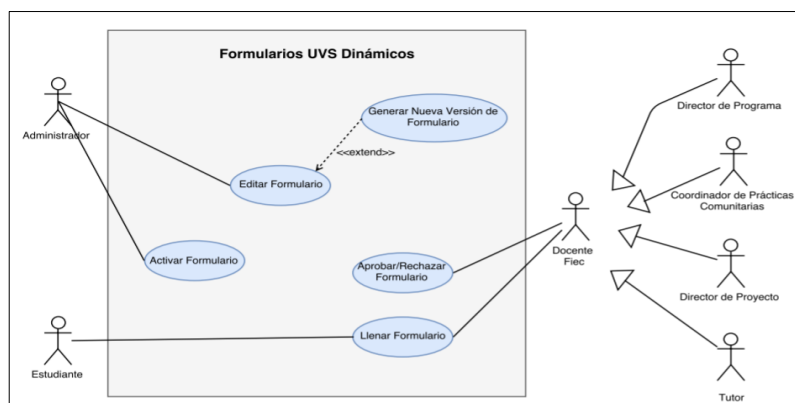


Figura 2.1: Diagrama de Caso de Uso de la Manipulación de Formularios Dinámicos.

Sobre la base de la Figura 2.1, tenemos lo siguiente:

➤ **Actores:**

- Administrador
- Estudiante
- Docente FIEC

- Director de Programa
- Director de Proyecto
- Tutor
- Coordinador de Prácticas Pre-Profesionales de Prácticas Comunitarias

➤ **Casos de Uso:**

- **Editar Formulario**

Consiste en permitir la edición de la estructura del formulario que ha sido previamente guardado, pero no activado. El actor con rol de Administrador es el único con acceso a esta funcionalidad.

- **Generar Nueva Versión de Formulario**

Generar una nueva versión consiste en duplicar la estructura de un formulario ya creado. La duplicación ocurre cuando el actor con rol de Administrador trata de editar la estructura de un formulario que ya fue activado.

- **Activar Formulario**

Una de las funcionalidades que también está a cargo del actor con rol de Administrador es la activación de formularios. Es decir, permitir que el formulario esté habilitado para ser llenado por los demás actores.

- **Llenar Formulario**

Consiste en ingresar información relacionada a un determinado proyecto/programa, la cual está a cargo de los actores de rol Estudiante o Docente FIEC.

- **Aprobar/Rechazar Formulario**

La aprobación o rechazo de un formulario está a cargo de los actores con rol de Docente FIEC, pues son ellos los que dan seguimiento a los proyectos a los que han sido asignados.

Diagramas de Actividades

Los diagramas de actividades permiten visualizar las nuevas funcionalidades de acuerdo con la solución propuesta; es decir, el manejo de formularios dinámicos. En la Figura 2.2 se muestra el flujo de actividades para la edición o creación de una versión de un formulario.

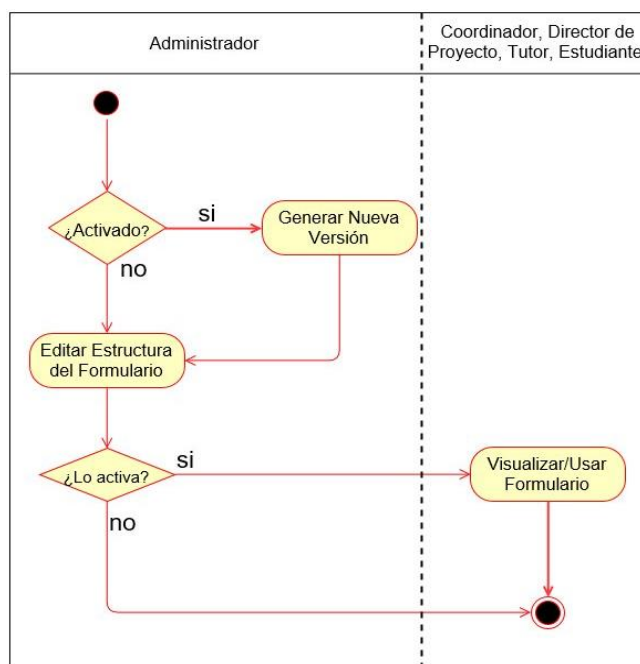


Figura 2.2: Diagrama de Actividad de Edición/Nueva Versión del Formulario.

2.4.2 Diseño de la base de datos

Aplicar la solución propuesta en la sección 2.2, implicó el rediseño de la base de datos del sistema actual. No en su totalidad, pero si en las tablas que guardan información de los formularios, que en la propuesta anterior se había realizado de manera convencional; es decir, tablas con campos fijos.

Dar “infinita libertad” al usuario para crear formularios es un tema complejo, en especial al momento de modelarlo en la base de datos. Sin embargo, se destacan 2 soluciones que podrían aplicarse:

- Guardar en una tabla la estructura del formulario en un formato XML o JSON, y los datos serializados por entidad en otra.
- Usar tablas CLAVE-VALOR, para guardar un número indefinido de campos del formulario, (ver Figura 2.3).

Siendo la segunda solución la más entendible y sencilla de aplicar [7]. La estrategia clave-valor funciona, de acuerdo a la Figura 2.3, de la siguiente manera:

Tabla: formularios			
id	nombre		
1	formulario1		

Tabla: campos				
id	form_id	clave	valor	
1	1	'nombre'	'text'	clave: campo del formulario valor: tipo de campo
2	1	'edad'	'int'	
...				

Tabla: valores				
id	form_id	clave	valor	
1	1	'nombre'	'Eddie'	clave: campo del formulario valor: contenido ingresado en el campo
2	1	'edad'	25	
...				

Figura 2.3: Ejemplo de funcionamiento de tablas Clave-Valor

Aplicando esta estrategia al presente proyecto, y añadiendo otros campos que serán necesarios para la **Edición, Activación, y Manejo de versiones** de los formularios dinámicos, el Diagrama Entidad-Relación que corresponde al manejo de los formularios quedó como se muestra en la Figura 2.4.

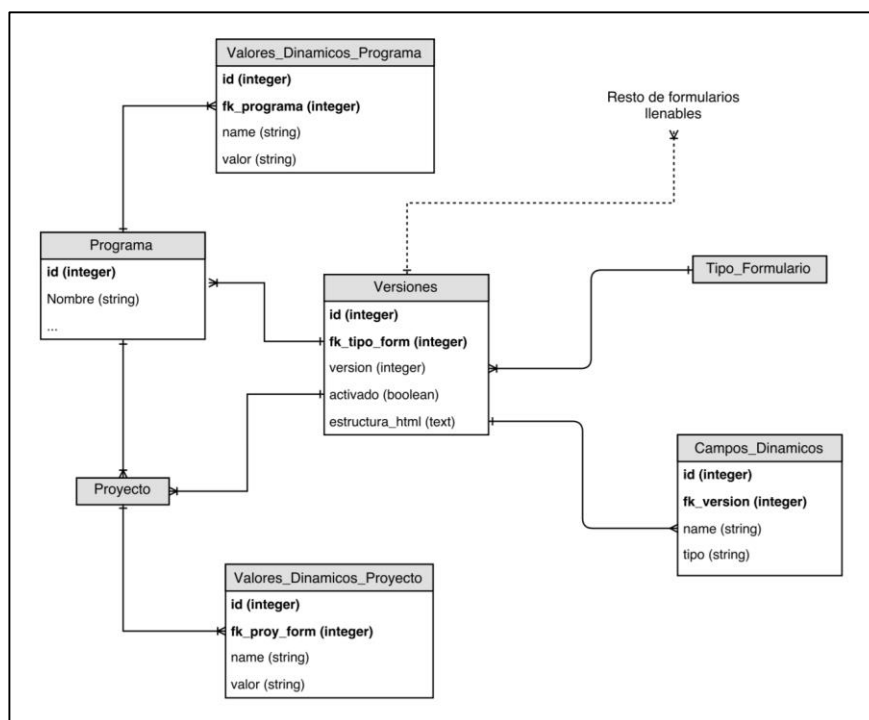


Figura 2.4: Diagrama E.R. del manejo de formularios dinámicos.

El esquema de la base de datos para el manejo de formularios dinámicos, según la Figura 2.4, está conformado por las siguientes entidades:

- **Programa:** un programa está constituido por un conjunto de proyectos destinados resolver problemas de una misma empresa u organización.
- **Proyecto:** un proyecto siempre estará conformado por muchos formularios, los cuales permiten dar seguimiento a las actividades que se hayan planificado dentro del proyecto.
- **Versiones:** es una tabla creada con el objetivo de permitir el almacenamiento de diferentes versiones de un formulario. Cada versión tiene los siguientes atributos:

- *Versión*: Incrementa cuando se crea una instancia de Version de un mismo formulario, a partir de uno existente.
 - *Activado*: Indica qué versiones están disponibles para ingresar información en ellos.
 - *Estructura_html*: Almacena la estructura del formulario como una plantilla HTML.
- **Tipo Formulario**: tipo de formulario se refiere a una categoría, la cual es asignada a cada formulario que se crea para así identificar qué usuarios tienen acceso a ellos. En otras palabras, el tipo de formulario permite conocer qué usuarios tienen acceso a la modificación del contenido y/o aprobación del contenido. Además, ayudar a construir el flujo del proceso de llenado y aprobación/rechazo de los formularios.
- **Campos Dinámicos**: esta entidad hace referencia a todos aquellos campos de la estructura de una versión de formulario que están destinados a luego mantener información de un determinado proyecto. Cada campo tiene los siguientes atributos (se mencionan los principales):
 - *Name*: Es el nombre del elemento HTML.
 - *Tipo*: Es el tipo de elemento HTML (textarea, input, etc).
- **Valores Dinámicos Programa/Proyecto/ForUVSXX**: son tablas en donde se almacenan los contenidos de los campos dinámicos que se han llenado en los formularios. Es decir, una vez que las versiones de los formularios fueron creadas y activadas por el Administrador, éstas estarán disponibles para que los usuarios encargados de llenar dicho formulario ingresen la información correspondiente. Al finalizar, esa información es guardada en la tabla *Valores_Dinamicos_Programa* si se trata de formularios de

un *Programa*, *Valores_Dinamicos_Proyecto* si se trata de formularios de un *Proyecto*, o *Valores_Dinamicos_ForUVSXX* si se trata de otro tipo de formulario (XX indica el número del formulario). Esto se lo hace utilizando los siguientes atributos, principalmente:

- *Name*: Es el nombre del elemento HTML.
- *Valor*: Es la información ingresada en el elemento HTML.

Adicionalmente, se puede notar que el esquema de la Figura 2.4 presenta un ciclo entre *Programa*, *Proyecto* y *Versiones*. Sin embargo, un ciclo es una condición necesaria, pero no suficiente para concluir que existe una relación redundante [13]. Un ciclo en una base de datos no es algo que esté totalmente prohibido; es decir, si al eliminar la relación redundante NO se puede llegar a la misma asociación, entonces el ciclo es inevitable [14]. En este caso, el ciclo es necesario, puesto que, al eliminar la relación entre *Programa* y *Versiones*, no se podrá saber qué formulario está asociado solamente al Programa. Aunque el Programa exista, se desconocería la información que solamente mediante un formulario se puede almacenar.

2.5 Implementación

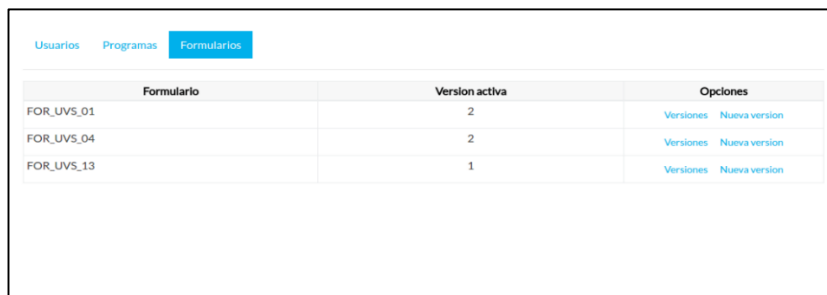
Como se mencionó previamente, el sistema está implementado con el Framework Laravel, por lo que se usa principalmente los lenguajes JavaScript con la librería JQuery y PHP (Front-end y Back-end respectivamente) para programar la solución. También se usan otras librerías de apoyo que se irán mencionando en cada sub-apartado.

2.5.1 Actualización de Formularios

Menú Formularios:

Para la actualización de formularios, se plantea que el encargado de dicha tarea sea un usuario del sistema con rol de Administrador, ya que en ese perfil es donde se alimenta al sistema con varios datos.

Siendo así, al perfil Administrador se le añade un menú “Formularios”, que inicialmente muestra la lista de los formularios que están activos.



The image shows a web interface with a navigation menu at the top containing 'Usuarios', 'Programas', and 'Formularios'. Below the menu is a table with three columns: 'Formulario', 'Version activa', and 'Opciones'. The table lists three forms: FOR_UVS_01, FOR_UVS_04, and FOR_UVS_13. Each row shows the active version number and two links: 'Versiones' and 'Nueva version'.

Formulario	Version activa	Opciones
FOR_UVS_01	2	Versiones Nueva version
FOR_UVS_04	2	Versiones Nueva version
FOR_UVS_13	1	Versiones Nueva version

Figura 2.5: Prototipo del menú Formularios

Versiones de un formulario:

Por cada formulario listado en la tabla, se podrá acceder a las versiones creadas de dicho formulario. Las versiones pueden tener 3 estados posibles:

- **Activo:** indica que es la versión del formulario que es usando actualmente en el sistema.
- **Desactualizado:** indica que es una versión antigua del formulario que fue usada anteriormente.
- **Activar (Botón):** indica que es una versión recién creada del formulario, y que puede ser activada en cualquier momento.

Una versión puede ser *editada* o *eliminada* SÓLO mientras NO haya sido activada; caso contrario, sólo podrá ser vista. Una vez activada, no hay marcha atrás (ver Figura 2.6).

FOR_UVS_01		
Version	Accion	Estado
1		Desactualizado
2		Activo
3		<input type="button" value="Activar"/>

Cerrar

Figura 2.6: Prototipo de la ventana de Versiones de un formulario.

El sistema partirá con los formularios actuales (estáticos) registrados como “**versión 0**”, de tal manera que estos servirán como base para crear nuevas versiones de dichos formularios.

Nueva Versión de un formulario:

Se puede generar una Nueva Versión de un formulario, a partir de la versión actual o activa. En la página se mostrará el formulario junto con un panel de “Herramientas”, la cual contiene todos los campos que se puede agregar al formulario (ver Figura 2.7).

Formularios / Nueva Version

FOR-UVS-XX

Formulario para la creacion de Programas

Nueva version: 2

HERRAMIENTAS

- Seccion
- Etiqueta
- Nota
- Texto corto
- Parrafo
- Editor de texto
- Archivo

1. Seccion 1

Respuesta corta

2. Seccion 2

Respuesta tipo párrafo

3. Seccion 3

No se eligió archivo

4. Seccion 4

-- Seleccione --

Figura 2.7: Prototipo de la vista de Nueva Versión de un formulario.

Los campos tienen su correspondiente formato HTML para poder ser agregados al formulario, y son los siguientes:

- **Sección:** formado por un <h4> y un <hr>.
- **Etiqueta:** formado por un <label>.
- **Nota:** formado por un <p>.
- **Texto corto:** <input> de tipo "text".
- **Párrafo:** formado por un <textarea>.
- **Editor de texto:** <textarea> de clase "ckeditor".
- **Archivo:** <input> de tipo "file".

Para agregar los campos al formulario, se optó por utilizar el método de **Drag&Drop**; es decir, los campos del panel de Herramientas podrán ser arrastrados y soltados dentro del formulario, debajo de una zona en específico.

Para lograr esto se utilizó **JqueryUI** [15], una extensión de JQuery que contiene un conjunto de efectos, widgets, e interacciones de interfaz de usuario. Entre las funciones que posee se encuentran **draggable** y **droppable**, que son las necesarias:

- **Draggable:** permite a elementos del DOM ser movidos usando el mouse.
- **Droppable:** crea objetivos para elementos tipo draggable.

Teniendo claros estos conceptos, se procede a usar estas funciones. En nuestro caso, los elementos de tipo Draggable son los campos del panel de Herramientas (ver Figura 2.8). Luego de identificarlos, se procede a inicializarlos (ver Figura 2.9).

```

<div class="col-md-3">
  <div style="position: fixed; width: 18%; z-index: 1">
    <div id="toolsMenu" class="panel panel-default">
      <div class="panel-heading">
        <h4 class="panel-title">HERRAMIENTAS</h4>
      </div>
      <div class="panel-body">
        <div class="panel-group">
          <div class="panel panel-default">
            <div class="panel-heading">
              <h4 class="panel-title">
                <a data-toggle="collapse" href="#basicos">Básicos <i class="fa fa-lg fa-caret-down pull-right"></i> </a>
              </h4>
            </div>
            <div id="basicos" class="panel-collapse collapse in">
              <div class="panel-body">
                <div class="row">
                  <div class="col-md-6">
                    <div class="btn btn-default addSeccion ui-widget-content">Sección</div>
                  </div>
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>

```

Figura 2.8: Código HTML del Panel de Herramientas.

```

$("#toolsMenu .btn").draggable({ opacity: 0.7, helper: "clone" });

```

Figura 2.9: Inicialización de comportamiento Draggable de los elementos del Panel de Herramientas con JQuery.

Queda definir los elementos de tipo Droppable; es decir, las zonas en donde se ejecute un evento al soltar los elementos de tipo Draggable (ver Figura 2.12). Para lograrlo se tuvo que rediseñar las vistas (html) de los formularios base, de tal manera que cada “h4”, “form-group”, u otro elemento correspondiente al Panel de Herramientas, quede dentro de un <div> con clase “droppable” (ver Figura 2.10). De esta manera se podrán inicializar posteriormente como elementos Droppable (ver Figura 2.11).

```

<!-- SECCION 1 DATOS GENERALES DEL PROGRAMA -->
<div class="droppable seccion restrict">
  <h4 class="titulo-seccion">1. DATOS GENERALES DEL PROGRAMA</h4>
  <hr>
</div>

<div class="form-group droppable">
  <div class="etiqueta restrict">
    {!!Form::label('titulo', '1.1. Título del programa:')}!!
  </div>
  {!!Form::text('titulo', null, ['id'=>'titulo', 'class'=>'form-control'])!!
</div>

<div class="form-group carreras-proyecto droppable">
  <div class="etiqueta restrict">
    {!!Form::label('carrerasProyecto', '1.2. Carrera(s) que participa(n) en el programa:')}!!
  </div>
  <div class="etiqueta restrict">
    {!!Form::label('carrera_principal', 'Carrera principal', ['class' => 'top-space'])!!
  </div>
  {!!Form::text('carrera_principal', null, ['id'=>'carrera_principal', 'class'=>'form-control'])!!
  <div class="table-responsive">
    <table class="table sv-list-table" id="tabla_carrera_principal">
    </table>
  </div>
</div>

```

Figura 2.10: Rediseño del HTML de un formulario.

```
$(".droppable").droppable(dropOptions);
```

Figura 2.11: Inicialización de elementos tipo Droppable.

```
var dropOptions = {
  classes: {
    "ui-droppable-hover": "dropBelow"
  },
  drop: function( event, ui ) {
    if( $(ui.draggable).hasClass("addSeccion") ){
      $(this).after( getSeccionSel(TRASH, "", "") );

      $(this).next().after('\
        <div class="droppable seccion editable" style="display:none">\
          <h4 class="titulo-seccion"></h4>\
          <hr>\
        </div>\
      ');

      $(this).next().next().droppable(dropOptions);
    }
    else if( $(ui.draggable).hasClass("addEtiqueta") ){
      $(this).after( getEtiquetaSel(TRASH, "" ) );

      $(this).next().after('\
        <div class="droppable etiqueta editable" style="display:none">\
          <label class="control-label" style="margin-bottom:20px"></label>\
        </div>\
      ');

      $(this).next().next().droppable(dropOptions);
    }
    else if( $(ui.draggable).hasClass("addNota") ){
```

Figura 2.12: Comportamiento del Drop.

Los elementos del Panel de Herramientas al ser colocados en el formulario tendrán la clase “editable”, lo que hará que al pasar el mouse encima de dichos elementos, estos tengan un fondo color Celeste y permitirá que su contenido pueda ser editado o eliminado en cualquier momento al darle click (ver Figura 2.13).

FOR-UVS-20
FORMULACIÓN DE PROGRAMAS DE VINCULACIÓN CON LA SOCIEDAD
Nueva versión: 1

1. DATOS GENERALES DEL PROGRAMA

Etiqueta de prueba

1.1. Título del programa:

1.2. Carrera(s) que participa(n) en el programa:

Carrera principal

Carreras adicionales

1.3. Área geográfica que cubre el programa: [\(Ver FOR-UVS-12\)](#)

Figura 2.13: Elementos que pueden ser Editados o Eliminados.

Cabe recalcar que solo podrán ser *eliminados* los elementos que NO pertenezcan al formulario base, es decir a la versión 0. Estos elementos tendrán la clase “restrict” y un fondo de color Rojo al pasar el mouse sobre ellos, lo que indica que sólo podrán *editarse*, pero no *eliminarse* (ver Figura 2.14).

FOR-UVS-20
FORMULACIÓN DE PROGRAMAS DE VINCULACIÓN CON LA SOCIEDAD
Nueva versión: 1

1. DATOS GENERALES DEL PROGRAMA

1.1. Título del programa:

1.2. Carrera(s) que participa(n) en el programa:

Carrera principal

Carreras adicionales

1.3. Área geográfica que cubre el programa: [\(Ver FOR-UVS-12\)](#)

Provincia:

Figura 2.14: Elementos restringidos de Eliminar

Guardar una nueva versión de un formulario

Aparte de la información que se registra en la base de datos, al guardar se genera un archivo tipo Vista de Laravel (blade.php) que contiene toda la estructura HTML del formulario.

Este archivo será guardado en la carpeta de vistas del sistema con el nombre "***FOR-UVS-XX_#.blade.php***", en donde ***XX*** es el número de formulario y ***#*** es la versión del formulario, y será usado automáticamente cuando dicha versión sea activada en el menú Formularios.

CAPÍTULO 3

3. RESULTADOS OBTENIDOS.

Con la finalidad de demostrar el cumplimiento de los objetivos planteados, en este capítulo se explican algunos esquemas y escenarios para mostrar el comportamiento actual del sistema web. Se inicia con un contraste entre las funcionalidades que antes brindaba el sistema y las que brinda ahora, y se finaliza con los escenarios principales de la gestión de programas.

3.1 Esquema del Funcionamiento del Sistema

3.1.1 Propuesta Anterior vs. Propuesta Actual

El esquema de la Figura 3.1 muestra cuál era el funcionamiento de la solución propuesta anteriormente. Ésta consistía en manejar formularios estáticos, cuyo contenido podía ser manipulado por 3 roles: Coordinador, Director y Estudiante. Al decir que estos formularios son estáticos, nos referimos a que la estructura inicial no podía ser alterada por ningún usuario, a menos que esa estructura sea actualizada por medio de mantenimientos continuos al sistema.

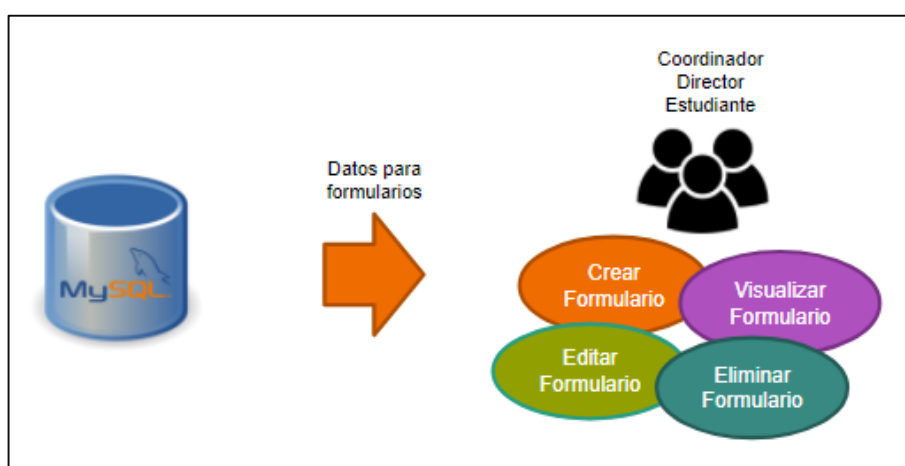


Figura 3.1: Esquema de Funcionamiento de la Propuesta Anterior.

El esquema de la Figura 3.2 muestra el funcionamiento actual del sistema. En esta nueva propuesta, se incluye la generación de nuevas versiones de los formularios a cargo del usuario con rol de *Administrador*. Estas versiones posteriormente son accedidas, mediante scripts, por los 3 roles restantes: Coordinador, Director y Estudiante. Cabe mencionar que, sólo puede activarse una versión de un formulario a la vez; sin embargo, se mantendrá un historial de las versiones que se hayan generado por cada formulario.

De esta manera, el funcionamiento actual brinda una ventaja más, la disminución del costo de mantenimiento a futuro del sistema.

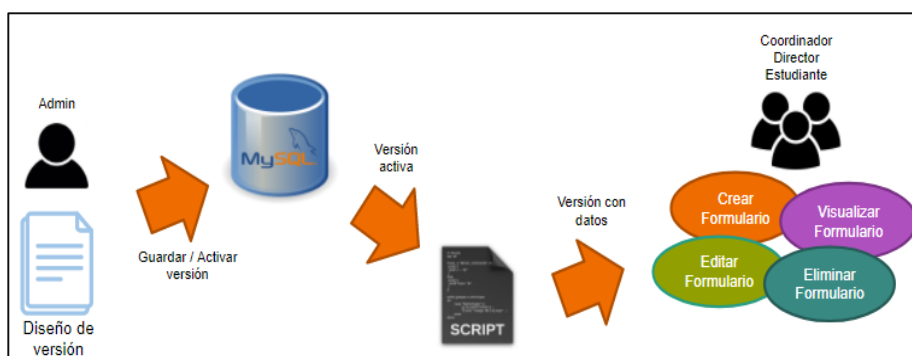


Figura 3.2: Esquema de Funcionamiento de la Propuesta Actual

3.1.2 Gestión de Programas

El segundo objetivo de este proyecto consiste en permitir la gestión de programas comunitarios, los cuales son formularios que mantienen toda la información necesaria para iniciar el proceso de prácticas comunitarias. De esta manera la Figura 3.3 muestra un esquema jerárquico de los formularios, es decir, la forma en que éstos se deberán llenar para seguir el flujo del proceso, según la Figura 1.3.

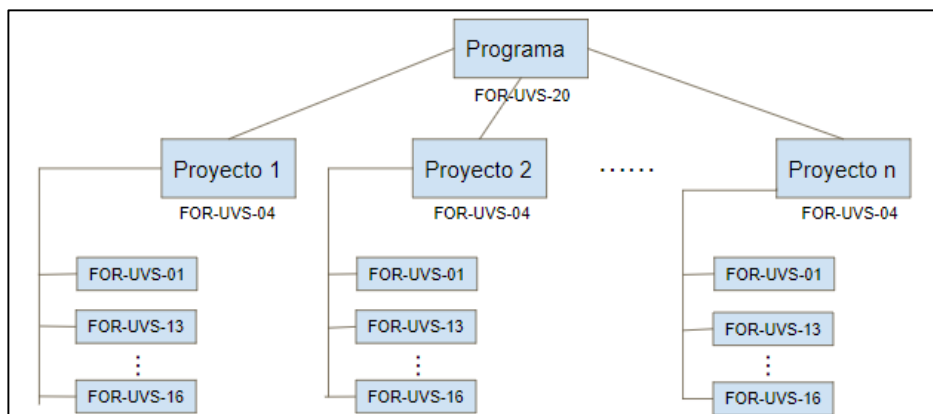


Figura 3.3: Esquema Jerárquico de Formularios.

Tal como se muestra en la Figura 3.3, el sistema está hecho para iniciar con la creación de programas, mediante el formulario FOR-UVS-20, desde el cual se crean también n proyectos.

Dado que nuevos proyectos surgirán en un corto plazo, se brinda la facilidad de que éstos puedan vincularse a un programa mediante una lista desplegable de los programas existentes. Esto último será mejor comprendido en la sección 3.3

3.2 Actualización de Formularios

3.2.1 Menú Principal de los Formularios

La Figura 3.4 permite visualizar los formularios que se encuentran disponibles en el sistema. En cada uno de ellos es visible la versión activa y 2 opciones disponibles para el usuario: ver el listado de versiones creadas del formulario y crear una nueva versión.

Lista de Formularios			
Formulario	Descripción	Versión Activa	Opciones
FOR-UVS-01	PLAN DE ACTIVIDADES	0	
FOR-UVS-04	FORMULACIÓN DE PROYECTOS DE VINCULACIÓN CON LA SOCIEDAD	0	
FOR-UVS-13	INFORME PARCIAL SEGUIMIENTO DE ACTIVIDADES TUTORIADAS	0	
FOR-UVS-14	INFORME FINAL DEL PRACTICANTE	0	
FOR-UVS-16	INFORME FINAL DEL DOCENTE TUTOR	0	
FOR-UVS-20	FORMULACIÓN DE PROGRAMAS DE VINCULACIÓN CON LA SOCIEDAD	0	

Figura 3.4: Lista de Formularios de Prácticas Comunitarias.

3.2.2 Nueva Versión y Versiones Anteriores

La Figura 3.5 muestra una interfaz mejorada, pero siguiendo la misma idea que se propuso inicialmente, en el prototipo descrito en la sección 2.6. En este menú el usuario va a poder arrastrar y soltar (Drag&Drop) nuevos componentes, desde el *Panel de Herramientas* al formulario; creando así una nueva versión del mismo, una vez que decida guardar los cambios realizados.

Formularios / Nueva versión

HERRAMIENTAS

Básicos

Sección Grupo

Individuales

Sección Etiqueta

Nota

Texto Corto Párrafo

Editor de Texto Archivo

FOR-UVS-01
PLAN DE ACTIVIDADES
Nueva versión: 1

1. DATOS GENERALES DEL PROYECTO

1.1. Título De La Actividad/servicios:

1.2. Organización Beneficiaria:

1.3. Número De Personas Como Beneficiarios Directo:

1.4. Número De Personas Como Beneficiarios Indirectos:

1.5. Area Geográfica Que Cubre:

Provincia: -
Cantón: -

Figura 3.5: Creación de una nueva versión del formulario FOR-UVS-01.

Cabe recalcar que el *Panel de Herramientas* que hace posible la edición de los formularios se encuentra disponible para todos los formularios de prácticas comunitarias, como se puede ver en la Figura 3.4.

3.2.3 Activación de Nueva Versión

El usuario con rol de *Administrador*, luego de crear una nueva versión, puede pasar a activarla. Como podemos ver en este caso (Figura 3.6), en todos los formularios se ha creado una nueva versión, 1, y se puede comparar con cómo se mostraba inicialmente en la Figura 3.4.

The screenshot shows a web interface with a modal window titled 'FOR-UVS-01'. The modal contains a table with the following data:

Versión	Acción	Estado
0		Desactualizado
1		ACTIVO

Below the modal, a 'Cerrar' button is visible. In the background, a table lists various forms with their descriptions and active versions. A red arrow points from the 'Opciones' column of the background table to the modal window.

Formulario	Descripción	Versión Activa	Opciones
FOR-UVS-01	PLAN DE ACTIVIDADES	1	
FOR-UVS-04	FORMULACIÓN DE PROYECTOS DE VINCULACIÓN CON LA SOCIEDAD	1	
FOR-UVS-13	INFORME PARCIAL SEGUIMIENTO DE ACTIVIDADES TUTORIADAS	1	
FOR-UVS-14	INFORME FINAL DEL PRACTICANTE	1	
FOR-UVS-16	INFORME FINAL DEL DOCENTE TUTOR	1	
FOR-UVS-20	FORMULACIÓN DE PROGRAMAS DE VINCULACIÓN CON LA SOCIEDAD	1	

Figura 3.6: Visualización de versiones anteriores y Activación de nueva versión.

Un formulario una vez activado, no puede ser editado. Si el usuario desea editarlo, generará por defecto una nueva versión. Dicha versión podrá ser modificada siempre y cuando permanezca en un estado de “Activar” (ver Figura 3.7). Por otro lado, una nueva versión de formulario que permanece en estado de “Activar” cuenta con las 3 acciones que se pueden realizar sobre él: visualizar o vista previa, editar y eliminar.

Versión	Acción	Estado
0		Desactualizado
1		ACTIVO
2		Activar

Cerrar

Figura 3.7: Estados de las Versiones del Formulario.

3.3 Programas y Proyectos

3.3.1 Menú Principal de Programas

El menú principal de programas permite visualizar cuáles son los programas comunitarios existentes, con su respectivo estado y acción (ver Figura 3.8).

Titulo Programa	Estado de Formulario	Accion
Programa 1	Aprobado	

+ Nuevo Programa

Figura 3.8: Lista de Programas Comunitarios.

Los estados de un programa pueden ser:

- No enviado: estado que tendrá por defecto al recién crearse.
- Enviado (Pendiente de Aprobación): cambiará a este estado cuando el formulario sea enviado al Coordinador de Prácticas Comunitarias.
- Aprobado: cambiará a este estado cuando el formulario sea aprobado por el Coordinador de Prácticas Comunitarias.

- Rechazado: cambiará a este estado cuando el formulario sea rechazado por el Coordinador de Prácticas Comunitarias.

Las acciones que se pueden realizar varían de acuerdo con el rol de usuario. Hasta el momento, son 2 usuarios los que tienen permisos para acceder a los programas. Para el Director de Programas, por ejemplo, existen 3 acciones: *Crear*, *Editar*, *Visualizar* y *Eliminar* (ver Figura 3.9). Mientras que el Coordinador de Prácticas Comunitarias sólo puede *Visualizar* y *Aprobar/Rechazar* el formulario de programas (ver Figura 3.10)

Programas			+ Nuevo Programa
Título Programa	Estado de Formulario	Acción	
Programa 1	Aprobado		
Desarrollo de Sistemas Informáticos Educativos	No enviado		

Figura 3.9: Acciones a cargo del Director de Programas.

Programas		
Título Programa	Estado de Formulario	Ver
Desarrollo de Sistemas Informáticos Educativos	No enviado	

Figura 3.10: Acciones a Cargo del Coordinador de Prácticas Comunitarias.

La creación de un programa implica, a su vez, la creación de los proyectos que se realizarán a lo largo del tiempo de duración del programa. Desde este formulario, se describen los campos amplios, específicos, y detallados por cada proyecto (ver Figura 3.11), información que luego será cargada en el FOR-UVS-04 (Formulario del Proyecto).

Lista de Programas / Nuevo

FOR-UVS-20
FORMULACIÓN DE PROGRAMAS DE VINCULACIÓN CON LA SOCIEDAD

1. DATOS GENERALES DEL PROGRAMA

1.1. Título del programa:
Desarrollo de Sistemas Informáticos Educativos

1.2. Carrera(s) que participa(n) en el programa:

Carrera principal
Ingeniería en Ciencias Computacionales

Ingeniería en Ciencias Computacionales

Carreras adicionales

Figura 3.11: Creación de Nuevo Programa Comunitario (FOR-UVS-20).

3.3.2 Asignación de Director a Proyectos Comunitarios

Una vez que el programa ha sido aprobado por el Coordinador de Prácticas Comunitarias, éste puede asignar un Director de Proyecto a cada uno (ver Figura 3.12).

Sistema de Ges

Asignar Docente Responsable del Proyecto

Selección docente director:
-- Seleccione --
-- Seleccione --
Jaine Alberto Macías Guerrero
Dario Sergio Leon Pasmay
Miriam Rocio Naula Rios
Fernando Galo Pelaez Ramos

coordinador2 | Cerrar sesión

Lista de Programas Lista de Proyectos

Proyectos

Ingrese título del proyecto

Titulo Proyecto

Proyecto 2

Estado

Pendiente

Figura 3.12: Asignación de Proyectos Creados en el FOR-UVS-20.

El sistema maneja notificaciones dentro del mismo. Por ejemplo, continuando con la explicación de asignación de proyecto de la Figura 3.12, el sistema es capaz de notificar inmediatamente al Docente que ha sido seleccionado como Director en un determinado proyecto (ver Figura 3.13)

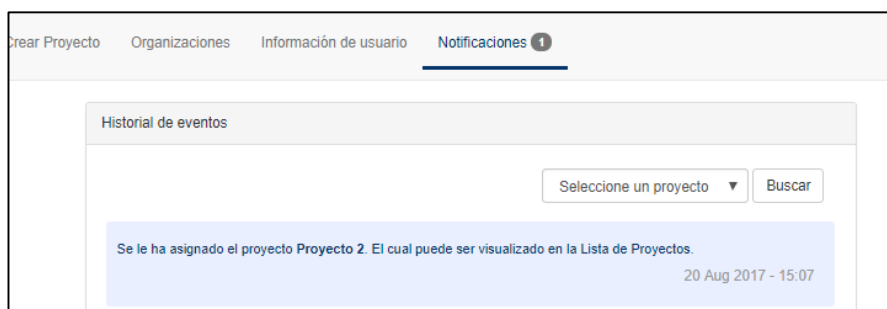


Figura 3.13: Notificación de Asignación de Proyectos.

3.3.3 Edición de Proyecto Asignado

El Director del Proyecto tiene acceso a la edición del formulario del proyecto asignado, pero como se puede ver en la Figura 3.14, mucho del contenido es generado mediante el programa al que está vinculado y esa información no puede ser modificada.

Figura 3.14: Edición de FOR-UVS-04.

3.3.4 Vinculación de Nuevo Proyecto a Programa Existente

Debido a que existe la posibilidad de que, habiendo iniciado un determinado programa se requiera vincular un nuevo proyecto a dicho programa, y teniendo en cuenta que todo programa al ser aprobado no puede ser editado, se redefinió el concepto de “Nuevo Proyecto”. Antes, el usuario encargado de crear un nuevo proyecto tenía que ingresar por completo toda la información; ahora, basta con seleccionar el programa al que se desea vincular el proyecto para que de esa manera parte del contenido sea cargado desde dicho programa (ver Figura 3.15).

Crear Proyecto Organizaciones Información de usuario Notificaciones 0 director

FOR-UVS-04

FORMULACIÓN DE PROYECTOS DE VINCULACIÓN CON LA SOCIEDAD

1. DATOS GENERALES DEL PROYECTO

1.1. Título del proyecto:

1.2. Programa al cual pertenece el proyecto:

* Si su programa no se encuentra en la lista, solicite al administrador del Sistema que lo agregue.

1.3. Carrera(s) que participa(n) en el proyecto: +

Carrera principal
Ingeniería en Ciencias Computacionales

Carreras adicionales

1.4. Área geográfica que cubre el proyecto: (Ver FOR-UVS-12)

Provincia:

Figura 3.15: Creación de Nuevo Proyecto.

3.3.5 Visualización del FOR-UVS-20 entre los detalles del proyecto

Aunque la visualización del contenido de los formularios de programas es una opción que ya estaba habilitada para los usuarios con rol *Director de Programas* y *Coordinador de Prácticas Comunitarias*, quienes están encargados de crear y aprobar/rechazar los programas respectivamente, se decidió también colocar esta opción entre los detalles de cada proyecto (ver Figura 3.16) para que el *Director de Proyectos* pueda estar bien informado sobre todo lo relacionado al proyecto que se le haya asignado y pueda controlar el cumplimiento a cabalidad de todos los requerimientos del mismo.

Módulo de Prueba	
Estado: Iniciado	
Programa al que pertenece	
Formulario	Acción
Formulación del Programa (FOR-UVS-20)	

Figura 3.16: Visualización de la información del Programa al que pertenece un determinado Proyecto.

3.4 Sistema Web en Producción

Con la finalidad de comprobar el correcto funcionamiento del sistema en producción, se utilizó el paquete XAMPP que contiene principalmente 3 elementos necesarios para la puesta en producción de este sistema: el gestor de bases de datos MySQL, el servidor web Apache y el lenguaje PHP.

3.4.1 Apache

Apache es un servidor web muy poderoso y completamente libre. Es muy popular debido a su estabilidad y seguridad, pero la mayor ventaja es que es un servidor web multiplataforma; es decir mantiene un excelente rendimiento sobre cualquier sistema operativo. Entre las principales características de Apache [16]:

- Soporte de seguridad Secure Sockets Layer (SSL) y Transport Layer Security (TLS).
- Puede utilizar autenticación de datos utilizando Sistemas Gestores de Bases de Datos (SGBD).
- Puede dar soporte a diferentes lenguajes, como Perl, PHP y Python.

3.4.2 Pruebas del Flujo del Proceso

Debido a que el proceso de prácticas comunitarias es muy largo, se mostrarán imágenes del proceso terminado, indicando las funciones claves en las que intervienen los diferentes roles de usuario.

Rol: Director de Programa

Funciones:

- **Creación de Proyectos mediante Programas**

El usuario crea los proyectos comunitarios desde el formulario de programas o FOR-UVS-20. Desde dicho formulario puede definir los campos amplios, específicos y detallados de cada uno de los proyectos. Ver Figura 3.17.

1.6. Campos de educación y capacitación CINE (UNESCO):			
Proyecto	Campo Amplio	Campo Especifico	Campo Detallado
Módulo de Prueba	campo amplio Proyecto 1	campo específico	campo detallado
Módulo de Aulas del Conocimiento	campo amplio	campo específico	campo detallado

Figura 3.17: Proyectos Creados desde el Formulario FOR-UVS-20.

Rol: Coordinador de Prácticas Comunitarias

Funciones:

- **Aprobación/Rechazo de FOR-UVS-20**

El usuario es el encargado de aprobar o rechazar el formulario del programa creado. En caso de aprobar, se notifica inmediatamente al *Director de Programas* que el formulario fue aprobado; sino se notifica que fue rechazado junto con el motivo de rechazo (ver Figura 3.18).

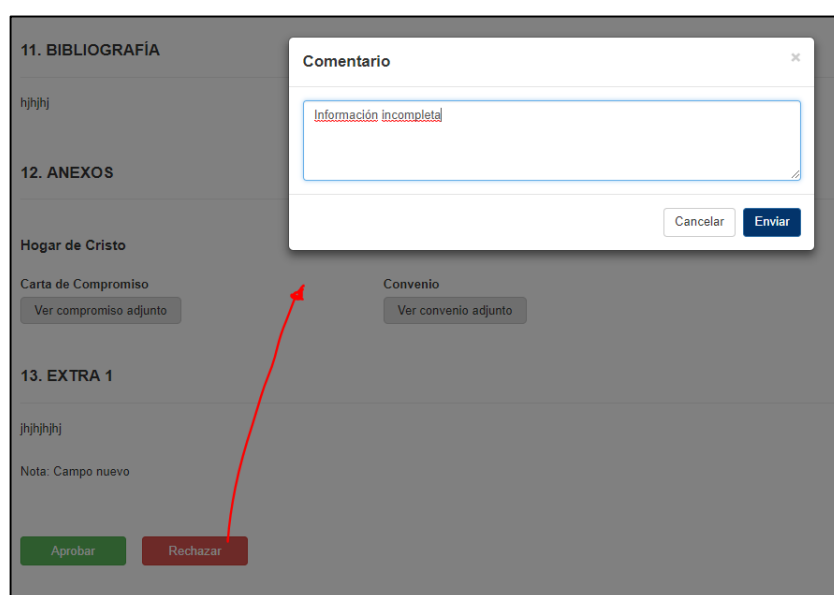


Figura 3.18: Opciones de Aprobación/Rechazo de FOR-UVS-20.

- **Asignación de Tutores y Estudiantes a un Proyecto**

Aunque el *Director de Proyecto* también es responsable de asignar estudiantes al proyecto, el *Coordinador* también tiene esta misma función, pero adicionalmente él es quien asigna el/los docente(s) que desempeñarán el rol de *Tutor*. Por supuesto, cada *Tutor* debe ser responsable de monitorear las actividades del/los estudiante(s) que tenga a cargo. Ver Figura 3.19.

Módulo de Prueba					
Estado: Iniciado					
Flujo del proyecto Detalles del Proyecto Asignar Tutores Asignar Estudiantes	Lista de Estudiantes				
	Ingrese matrícula		Agregar		
	Nombre	Matricula	Correo	Tutor	Acción
	Janna Elizabeth Barreto Castro	201008877	jbarreto@espol.edu.ec	Sergio Roberto Wong Diaz (Cambiar Tutor)	✖

Figura 3.19: Asignación de Tutores y Estudiantes al Proyecto.

- **Aprobación de FOR-UVS-16**

Este formulario es el informe final, el cual es entregado por el Tutor. Mediante este formulario se declara si el estudiante ha cumplido a cabalidad cada una de las actividades que se le ha asignado y el número de horas invertidas en la práctica. El Tutor se encarga de enviar este informe al Coordinador y éste decide si aprobarlo o rechazarlo. En caso de aprobarlo, se da por terminada la práctica comunitaria del estudiante. Ver Figura 3.20.










Módulo de Prueba	
Estado: Iniciado	
Detalle del proyecto / Información Estudiante	
Nombres y Apellidos: Janna Elizabeth Barreto Castro	
Matricula: 201008877	
Correo Electrónico: jbarreto@espol.edu.ec	
Estado del estudiante: Actividades terminadas	
Formularios Aprobados	
Formulario	Acción
FOR-UVS-01	 
FOR-UVS-14	 
FOR-UVS-13	 
FOR-UVS-15	
FOR-UVS-16	 

Figura 3.20: Finalización de Prácticas Comunitarias

Rol: Tutor

Funciones:

- **Formularios a cargo**

El usuario con rol de *Tutor* es el docente encargado de dar seguimiento a las actividades asignadas a los estudiantes durante todo el proceso de prácticas comunitarias. Durante el proceso, existen 3 formularios que el usuario debe llenar por cada estudiante que tenga a cargo. Ver Figura 3.21.

Módulo de Prueba		
Estado: Iniciado		
Formularios		
Lista		
Formulario	Estado	Accion
FOR-UVS-13 - Janna Elizabeth Barreto Castro	REVISADO	 
FOR-UVS-15 - Janna Elizabeth Barreto Castro	Aprobado	
FOR-UVS-16 - Janna Elizabeth Barreto Castro	Aprobado	

Figura 3.21: Formularios Llenados por el Tutor.

Rol: Estudiante

Funciones:

- **Formularios a cargo**

Cada practicante es responsable de llenar 2 formularios relacionados al proyecto en el que participan. Dichos formularios le permitirán continuar con el flujo del proceso para, finalmente, cumplir con sus horas de prácticas comunitarias. Ver Figura 3.22.



Módulo de Prueba		
Estado del proyecto: Iniciado		
Organización: Hogar de Cristo		
Director: Jaime Alberto Macias Guerrero		
Tutor asignado: Sergio Roberto Wong Diaz		
Estado del estudiante: Actividades terminadas		
Formularios		
Tipo	Estado	Acción
FOR-UVS-01	Aprobado	
FOR-UVS-14	Aprobado	

Figura 3.22: Formularios Llenados por el Practicante.

Rol: Director de Proyecto, Coordinador de Prácticas Comunitarias y Tutor

Funciones:

- **Visualización del Flujo**

Finalmente, la manera de comprobar el avance de un proyecto de prácticas comunitarias es mediante un grafo donde cada nodo representa los formularios que ha sido aprobados. Cuando el grafo presente todos sus nodos en color verde, se entiende que el proceso ha terminado. Ver Figura 3.23.

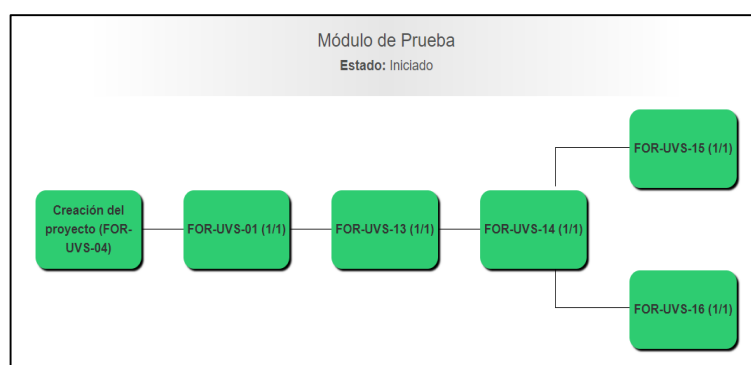


Figura 3.23: Culminación del Proceso de Prácticas Comunitarias.

CONCLUSIONES Y RECOMENDACIONES

El sistema web soporta la actualización continua de formularios. Mediante el concepto de dinamismo no sólo se ha permitido que el usuario tenga la facilidad de configurar el formulario como lo desee, sino que también se ha disminuido el costo de mantenimiento, a futuro, del sistema.

La estrategia utilizada para el dinamismo fue el uso de tablas clave-valor, la cual ha permitido que tanto el almacenamiento como la obtención de datos en la base de datos, sea mucho más rápido.

La incorporación de formularios de programas en el sistema web agiliza aún más el proceso de prácticas comunitarias, debido a que es a partir de éste, de donde se genera toda la información necesaria para el flujo de actividades entre los usuarios enrolados en los diversos proyectos de prácticas comunitarias.

El sistema web puede ser mejorado mediante la incorporación de formularios dinámicos en el proceso de prácticas empresariales. De esta manera se beneficiaría a un grupo más grande de estudiantes y docentes que se vinculen en las prácticas pre-profesionales.

El inicio y culminación de un proyecto de prácticas comunitarias se evidencia mediante el gráfico del flujo dentro del sistema web, que permite conocer el avance del proyecto. Sin embargo, se recomienda agregar un menú de reportes que genere la lista de estudiantes que han cumplido con sus horas de prácticas comunitarias por periodo. Esto último con la finalidad de agilizar la convalidación de horas de las prácticas comunitarias realizadas por el estudiante.

Una última recomendación que podría ser de mucha ayuda para los involucrados en el proceso de prácticas comunitarias, es permitir la impresión directa de los formularios con el formato que indica la UVS para que posteriormente sean entregados al responsable de archivar dichos documentos.

BIBLIOGRAFÍA

- [1] Consejo de Educación Superior (CES). (2010, Octubre 12). Ley Orgánica de Educación Superior [Online]. Disponible en: http://www.ces.gob.ec/index.php?option=com_phocadownload&view=category&id=11:ley-organica-de-educacion-superior&Itemid=137
- [2] J. A. Romero, J. L. Monar, "Desarrollo de una Aplicación web para la Gestión de Actividades de Proyectos de Vinculación con la Sociedad", Tesis de Ingeniería, Fiec, Espol, Guayaquil, EC, 2016.
- [3] P. Alzuri, G. Moreira, El qué, cuándo y cómo de los formularios dinámicos [Online]. Disponible en: <https://es.slideshare.net/genexus/062-el-que-cuando-y-como-de-los-formularios-dinamicos>.
- [4] Google, Google forms [Online]. Disponible en: <https://www.google.com/forms/about/>
- [5] FormBuilder [Online]. Disponible en: <http://formbakery.com/>
- [6] WordPress [Online]. Disponible en: <https://es.wordpress.org/>
- [7] LibrosWeb (2014, Mayo 30). Formularios Dinámicos Generados A Partir De La Información De La Base De Datos [Online]. Disponible en: <http://librosweb.es/foro/pregunta/147/formularios-dinamicos-generados-a-partir-de-la-informacion-de-la-base-de-datos/>
- [8] BBVA (2015, Julio 15). Herramientas Básicas para los Desarrolladores en PHP [Online]. Disponible en: <https://bbvaopen4u.com/es/actualidad/herramientas-php-para-desarrolladores>
- [9] Wikipedia (2017, Abril 5). Laravel [Online]. Disponible en: <https://es.wikipedia.org/wiki/Laravel>
- [10] Wikipedia (2017, Mayo 30). MySQL [Online]. Disponible en: <https://es.wikipedia.org/wiki/MySQL>

[11] Oracle. La Base de datos de código abierto más popular del mundo [Online]. Disponible en: <https://www.oracle.com/lad/mysql/index.html>

[12] MySQL (2017). MySQL Workbench [Online]. Disponible en: <https://www.mysql.com/products/workbench/>

[13] L. Valencia (2013). Bases de Datos [Online]. Disponible en: http://www.cs.us.es/blogs/bd2012/files/2012/09/BD-Tema-1_Adjunto.pdf

[14] S. Shlaer y N. Lang, "Shlaer-Mellor Method: The OOA96 Report", Project Technology Inc., Berkely, CA, 2560 Ninth Street. Suit 214, 1996.

[15] JQuery User Interface [Online]. Disponible en: <http://jqueryui.com/droppable/>

[16] Culturacion [Online]. Disponible en: <http://culturacion.com/que-es-apache/>