



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

IMPLEMENTACIÓN DE LA PLATAFORMA DE VINCULACIÓN DE EGRESADOS Y CURSOS
LIBRES EN LA FACULTAD DE ARQUITECTURA, UNIVERSIDAD DE SAN CARLOS DE
GUATEMALA

Julio Fernando Flores Interiano

Asesorado por el Ing. Hugo Esteban Ajuchán Chis

Guatemala, noviembre de 2017

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**IMPLEMENTACIÓN DE LA PLATAFORMA DE VINCULACIÓN DE EGRESADOS Y CURSOS
LIBRES EN LA FACULTAD DE ARQUITECTURA, UNIVERSIDAD DE SAN CARLOS DE
GUATEMALA**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

JULIO FERNANDO FLORES INTERIANO

ASESORADO POR EL ING. HUGO ESTEBAN AJUCHÁN CHIS

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, NOVIEMBRE DE 2017

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Pedro Antonio Aguilar Polanco
VOCAL I	Ing. Angel Roberto Sic García
VOCAL II	Ing. Pablo Christian de León Rodríguez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Jurgen Andoni Ramírez Ramírez
VOCAL V	Br. Oscar Humberto Galicia Nuñez
SECRETARIA	Inga. Lesbia Magalí Herrera López

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Pedro Antonio Aguilar Polanco
EXAMINADOR	Ing. Marlon Antonio Pérez Türk
EXAMINADOR	Ing. Sergio Leonel Gómez Bravo
EXAMINADORA	Inga. Floriza Felipa Ávila Pesquera
SECRETARIA	Inga. Lesbia Magalí Herrera López

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

IMPLEMENTACIÓN DE LA PLATAFORMA DE VINCULACIÓN DE EGRESADOS Y CURSOS LIBRES EN LA FACULTAD DE ARQUITECTURA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha 26 de febrero de 2017.

A handwritten signature in black ink, consisting of several overlapping loops and a long horizontal stroke at the bottom.

Julio Fernando Flores Interiano

Guatemala, 26 de octubre de 2017

Inga. Christa del Rosario Classon de Pinto
Directora de la Unidad de EPS
Facultad de Ingeniería, USAC
Ciudad Universitaria, Guatemala

Inga. Christa del Rosario Classon de Pinto:

Por este medio le informo que después de revisar los avances del trabajo de EPS titulado "IMPLEMENTACIÓN DE LA PLATAFORMA DE VINCULACIÓN DE EGRESADOS Y CURSOS LIBRES EN LA FACULTAD DE ARQUITECTURA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA", el cual está a cargo del estudiante de Ingeniería en Ciencias y Sistemas de la Universidad de San Carlos de Guatemala, **Julio Fernando Flores Interiano**, que se identifica con el registro académico **201213230** y CUI **2268 10976 0101**, hago constar que el 100% del trabajo escrito ha sido revisado por mi persona, autorizando su publicación sin ningún inconveniente.

Agradeciendo la atención a la presente y quedando a sus órdenes para cualquier información adicional.

Atentamente,



Ingeniero en Ciencias y Sistemas
Hugo Esteban Ajuchán
Colegiado. No. 9531
Asesor

Hugo Esteban Ajuchán Chis
Ingeniero en Ciencias y Sistemas
Col. 9531



Guatemala, 30 de octubre de 2017.
REF.EPS.DOC.755.10.2017.

Inga. Christa Classon de Pinto
Directora Unidad de EPS
Facultad de Ingeniería
Presente

Estimada Ingeniera Classon de Pinto:

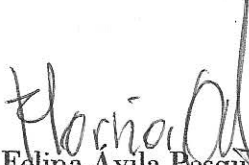
Por este medio atentamente le informo que como Supervisora de la Práctica del Ejercicio Profesional Supervisado, (E.P.S) del estudiante universitario de la Carrera de Ingeniería en Ciencias y Sistemas, **Julio Fernando Flores Interiano, Registro Académico 201213230 y CUI 2268 109760101** procedí a revisar el informe final, cuyo título es **IMPLEMENTACIÓN DE LA PLATAFORMA DE VINCULACIÓN DE EGRESADOS Y CURSOS LIBRES EN LA FACULTAD DE ARQUITECTURA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA.**

En tal virtud, **LO DOY POR APROBADO**, solicitándole darle el trámite respectivo.

Sin otro particular, me es grato suscribirme.

Atentamente,

"Id y Enseñad a Todos"


Inga. Floriza Felipa Avila Pesquera de Medinilla
Supervisora de EPS
Área de Ingeniería en Ciencias y Sistemas



FFAPdM/RA



Guatemala, 30 de octubre de 2017.

REF.EPS.D.457.10.2017.

Ing. Marlon Antonio Pérez Turk
Director Escuela de Ingeniería Ciencias y Sistemas
Facultad de Ingeniería
Presente

Estimado Ingeniero Pérez Türk:


Por este medio atentamente le envío el informe final correspondiente a la práctica del Ejercicio Profesional Supervisado, (E.P.S) titulado **IMPLEMENTACIÓN DE LA PLATAFORMA DE VINCULACIÓN DE EGRESADOS Y CURSOS LIBRES EN LA FACULTAD DE ARQUITECTURA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**, que fue desarrollado por el estudiante universitario **Julio Fernando Flores Interiano**, Registro Académico 201213230 y CUI 2268 109760101 quien fue debidamente asesorado por el Ing. Hugo Esteban Ajuchán y supervisado por la Inga. Floriza Felipa Ávila Pesquera de Medinilla.

Por lo que habiendo cumplido con los objetivos y requisitos de ley del referido trabajo y existiendo la aprobación del mismo por parte del Asesor y la Supervisora de EPS, en mi calidad de Director apruebo su contenido solicitándole darle el trámite respectivo.

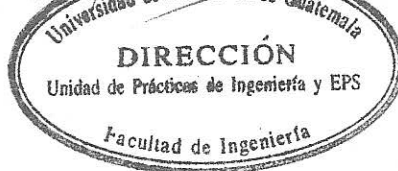
Sin otro particular, me es grato suscribirme.

Atentamente,

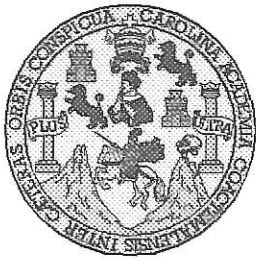
"Id y Enseñad a Todos"


Inga. Christa Classon de Pinto

Directora Unidad de EPS



CCsP/ra



Universidad San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 8 de Noviembre de 2017


Ingeniero
Marlon Antonio Pérez Türk
Director de la Escuela de Ingeniería
En Ciencias y Sistemas

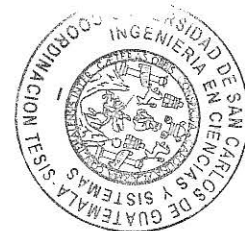
Respetable Ingeniero Pérez:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación-EPS del estudiante **JULIO FERNANDO FLORES INTERIANO** carné 201213230 y CUI 2268 10976 0101, titulado: "IMPLEMENTACIÓN DE LA PLATAFORMA DE VINCULACIÓN DE EGRESADOS Y CURSOS LIBRES EN LA FACULTAD DE ARQUITECTURA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA" y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,


Ing. Carlos Alfredo Azurdia
Coordinador de Privados
y Revisión de Trabajos de Graduación



UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA EN
CIENCIAS Y SISTEMAS
TEL: 24767644

*El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del asesor con el visto bueno del revisor y del Licenciado en Letras, del trabajo de graduación **“IMPLEMENTACIÓN DE LA PLATAFORMA DE VINCULACIÓN DE EGRESADOS Y CURSOS LIBRES EN LA FACULTAD DE ARQUITECTURA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA”**, realizado por el estudiante JULIO FERNANDO FLORES INTERIANO, aprueba el presente trabajo y solicita la autorización del mismo.*

“ID Y ENSEÑAD A TODOS”


Ing. Marlon Antonio Pérez Türk
Director

Escuela de Ingeniería en Ciencias y Sistemas



Guatemala, 08 de noviembre de 2017

Universidad de San Carlos
de Guatemala

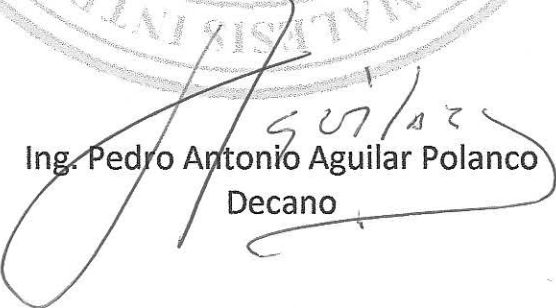


Facultad de Ingeniería
Decanato

DTG. 548.2017

El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al Trabajo de Graduación titulado: **IMPLEMENTACIÓN DE LA PLATAFORMA DE VINCULACIÓN DE EGRESADOS Y CURSOS LIBRES EN LA FACULTAD DE ARQUITECTURA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**, presentado por el estudiante universitario: **Julio Fernando Flores Interiano**, y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:


Ing. Pedro Antonio Aguilar Polanco
Decano

Guatemala, noviembre de 2017

/gdech



ACTO QUE DEDICO A:

Dios	Ser supremo y luz que guía mi vida.
Guatemala	Mi nación, por la cual trabajaré arduamente.
Mis abuelos	Servio Interiano (q. e. p. d.), Julio Flores (q. e. p. d.), Anita Carío, Sonia Domínguez y Herminia Zetina.
Mis padres	Ana María Interiano Carío y Fernando Rodrigo Flores Domínguez

AGRADECIMIENTOS A:

Universidad de San Carlos de Guatemala	Por la educación brindada al pueblo de Guatemala.
Facultad de Ingeniería	Por el conocimiento que me fue impartido.
Facultad de Arquitectura	Especialmente a la Unidad de Bienestar y Desarrollo Estudiantil. Por permitirme realizar mi trabajo de graduación en su institución.
Mis padres	Por su apoyo incondicional durante mis estudios.
Mairim Chávez	Por su cariño y acompañamiento durante la carrera.
Ing. Hugo Ajuchán	Por su asesoramiento durante la realización de mi trabajo de graduación.
Arq. Oscar Enríquez	Por facilitar la realización del proyecto de mi trabajo de graduación y su apoyo en el mismo.
Mis amistades de la Facultad	Por acompañarme a lo largo de la carrera y compartir las experiencias del aprendizaje.
Mis compañeros y amistades laborales.	Por su comprensión y consejo en la realización de mi trabajo de graduación.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES.....	V
GLOSARIO.....	VII
RESUMEN.....	IX
OBJETIVOS	XI
INTRODUCCIÓN.....	XIII
1. FASE DE INVESTIGACIÓN	1
1.1. Antecedentes de la empresa	1
1.1.1. Reseña histórica	1
1.1.2. Misión	1
1.1.3. Visión.....	2
1.1.4. Servicios que realiza.....	2
1.2. Descripción de las necesidades.....	3
1.3. Priorización de las necesidades.....	4
2. FASE TÉCNICO PROFESIONAL.....	5
2.1. Descripción del proyecto.....	5
2.1.1. Vinculación de egresados	5
2.1.1.1. Recopilación de información	5
2.1.1.2. Facilitación de oportunidades laborales	5
2.1.1.3. Divulgación de información	6
2.1.2. Cursos Libres.....	6
2.2. Investigación preliminar para la solución del proyecto	6
2.2.1. Propios de la lógica del proyecto	7

	2.2.1.1.	Bolsa de empleo	7
	2.2.1.2.	Red de contactos profesionales	7
2.2.2.		Propios de la implementación técnica.....	8
	2.2.2.1.	Scrum	8
	2.2.2.2.	Javascript	10
	2.2.2.3.	PHP	10
	2.2.2.4.	Composer	10
	2.2.2.5.	Zend Framework 3.....	11
	2.2.2.6.	Apache HTTP Server.....	12
	2.2.2.7.	MariaDB.....	13
	2.2.2.8.	Doctrine	13
	2.2.2.9.	Git.....	13
	2.2.2.10.	Vagrant.....	14
	2.2.2.11.	Debian	14
	2.2.2.12.	Yarn.....	14
	2.2.2.13.	Sass	15
	2.2.2.14.	Bootstrap	15
2.3.		Presentación de la solución al proyecto	15
	2.3.1.	Especificaciones técnicas.....	15
	2.3.2.	Arquitectura	16
	2.3.3.	Casos de uso	17
	2.3.3.1.	Diagrama de casos de uso para el sistema de vinculación de egresados ..	17
	2.3.3.2.	Casos de uso para el sistema de vinculación de egresados	19
	2.3.3.3.	Historias de usuario para el sistema de vinculación de egresados	31
	2.3.3.4.	Diagrama de casos de uso para el sistema de Cursos Libres	31

2.3.3.5.	Casos de uso para el sistema de Cursos Libres	32
2.3.3.6.	Historias de usuario para el sistema de Cursos Libres	37
2.3.4.	Información de desarrollo.....	38
2.3.4.1.	Scrum.....	38
2.3.4.2.	Git	38
2.3.4.3.	Vagrant	39
2.3.4.4.	Sass	39
2.3.5.	Patrones de diseño aplicados	40
2.3.5.1.	Modelo Vista Controlador	40
2.3.5.2.	Fábricas e inyección de dependencias.....	40
2.3.5.3.	Singleton	40
2.3.5.4.	Domain Driven Design	41
2.3.6.	Estructura de la solución.....	41
2.3.6.1.	Estructura general de un módulo	45
2.3.6.2.	Módulo Application	47
2.3.6.3.	Módulo Autenticacion	48
2.3.6.4.	Módulo CursosLibres	50
2.3.6.5.	Módulo vinculación egresados	52
2.4.	Costos del proyecto	57
2.5.	Beneficios del proyecto.....	58
2.5.1.	Vinculación de egresados	58
2.5.2.	Cursos Libres.....	59
3.	FASE ENSEÑANZA APRENDIZAJE.....	61
3.1.	Capacitación propuesta	61
3.1.1.	Dirigida al personal técnico	61

3.1.2.	Dirigida al personal administrativo de la plataforma	61
3.2.	Material elaborado	62
3.2.1.	Técnico.....	62
3.2.2.	Información de uso	62
3.2.2.1.	Dirigido al personal administrativo	62
3.2.2.2.	Dirigido al egresado.....	62
3.2.2.3.	Dirigido a las empresas	63
3.2.2.4.	Dirigido a los aspirantes a Cursos Libres.....	63
	CONCLUSIONES	65
	RECOMENDACIONES	67
	BIBLIOGRAFÍA	69
	ANEXOS	73

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Diagrama de arquitectura de la plataforma.....	17
2.	Diagrama de casos de uso para el sistema de vinculación de egresados	18
3.	Diagrama de casos de uso para el sistema de Cursos Libres	32
4.	Estructura general de un proyecto	41
5.	Estructura de carpeta config.....	42
6.	Estructura de carpeta data	43
7.	Estructura carpeta module en sistema de vinculación de egresados....	44
8.	Estructura de carpeta module en sistema de Cursos Libres.....	44
9.	Estructura de carpeta <i>public</i>	45
10.	Ejemplo de estructura general de un módulo	46

TABLAS

I.	Caso de uso VE-1 Registrarse	19
II.	Caso de uso VE-2 Iniciar sesión como egresado	20
III.	Caso de uso VE-3 Recuperar contraseña	20
IV.	Caso de uso VE-4 Editar mi perfil	21
V.	Caso de uso VE-5 Ver oportunidades laborales	22
VI.	Caso de uso VE-6 Ver información académica	22
VII.	Caso de uso VE-7 Buscar egresados desde el catálogo	23
VIII.	Caso de uso VE-8 Enviar mensaje a egresado	23
IX.	Caso de uso VE-9 Iniciar sesión como empresa	24

X.	Caso de uso VE-10 Administrar oferta laboral	25
XI.	Caso de uso VE-11 Editar información de contacto	25
XII.	Caso de uso VE-12 Iniciar sesión como empleado	26
XIII.	Caso de uso VE-13 Administrar información académica	27
XIV.	Caso de uso VE-14 Administrar cuentas de empresas	27
XV.	Caso de uso VE-15 Modificar recursos.....	28
XVI.	Caso de uso VE-16 Descargar listado de egresados.....	29
XVII.	Caso de uso VE-17 Revisar propuestas de oferta laboral.....	29
XVIII.	Caso de uso VE-18 Administrar cuentas de egresados	30
XIX.	Caso de uso CL-1 Visualizar Cursos Libres disponibles	33
XX.	Caso de uso CL-2 Inscribirse al programa.....	33
XXI.	Caso de uso CL-3 Asignarse a un curso.....	34
XXII.	Caso de uso CL-4 Iniciar sesión como administrador	35
XXIII.	Caso de uso CL-5 Administración de Cursos Libres	35
XXIV.	Caso de uso CL-6 Modificar recursos	36
XXV.	Caso de uso CL-7 Generar reportes sobre inscripción	37
XXVI.	Desglose presupuestario	57

GLOSARIO

Controlador de versiones	Herramienta que almacena los cambios a un grupo de archivos a lo largo del tiempo.
CSS	Cascading Style Sheets es un lenguaje utilizado para la definición del aspecto y estilo de una página web.
Cursos Libres	Cursos gratuitos, ofrecidos por la Universidad de San Carlos de Guatemala.
Egresado	Profesional que cerró pensum y se graduó en la Facultad de Arquitectura.
HTML	HiperText Markup Language es un lenguaje utilizado para la definición del contenido de una página web.
Manejador de dependencias	Herramienta de software que almacena un listado de dependencias para un proyecto y se encarga de su instalación y mantenimiento.
Metodologías de desarrollo ágil	Una rama de técnicas de desarrollo en las cuales se priorizan la interacción entre individuos, el software funcional, la colaboración con el cliente y la respuesta ante el cambio.

Open source software	Software cuyo código fuente está disponible al público y puede ser modificado y compartido.
PHP-FIG	PHP-Framework Interoperability Group, es un grupo de desarrolladores de proyectos PHP que buscan trabajar en conjunto y establecer estándares.
Servidor web	Sistema operativo que utiliza el núcleo Linux y una variedad de herramientas GNU.
Sistema Operativo GNU/Linux	Herramienta de software que sirve archivos mediante el protocolo HTTP.

RESUMEN

En este documento se presenta la plataforma de vinculación de egresados y Cursos Libres de la Facultad de Arquitectura de la Universidad de San Carlos de Guatemala.

La plataforma está dividida en dos sistemas informáticos desarrollados durante el año 2017: el sistema de vinculación de egresados y el sistema de Cursos Libres.

El sistema de vinculación de egresados surge con la finalidad de solventar las necesidades que la Facultad de Arquitectura presenta con relación a sus egresados. Este nuevo sistema opera como un sitio en internet, donde se facilita la ubicación del egresado dentro del marco laboral del país.

Dentro del sistema, se recopila información de los egresados, la cual se hace pública para los profesionales y las empresas. Esto fomenta la generación de oportunidades de empleo y redes de contacto. Las empresas pueden publicar directamente ofertas laborales, información que pueden observar los egresados dentro del sistema. Asimismo, la Facultad puede publicar información académica, generando un canal de comunicación con los egresados.

Por su parte, el sistema de Cursos Libres, permite realizar el proceso de gestión e inscripción a los cursos mediante internet, facilitando el acceso a los mismos.

OBJETIVOS

General

Implementar una plataforma informática que ayude al egresado a ubicarse de forma más eficiente dentro del mercado laboral e implementar un sistema para la inscripción de Cursos Libres.

Específicos

1. Facilitar la divulgación de información laboral y académica a los egresados de la Facultad de Arquitectura de la Universidad de San Carlos de Guatemala.
2. Crear una cartera que permita a empresas seleccionadas la rápida ubicación de los profesionales por área de desempeño.
3. Agilizar los procesos de inscripción y facilitar la emisión de diplomas para los Cursos Libres.

INTRODUCCIÓN

Actualmente, las necesidades principales de la Facultad de Arquitectura respecto a los egresados se pueden agrupar en tres categorías:

- Recopilar eficazmente información actualizada sobre los egresados de la Facultad.
- Dar a conocer de manera eficaz y eficiente oportunidades laborales para los egresados.
- Crear un medio de comunicación de información entre la Facultad y los egresados.

Asimismo, la necesidad relacionada con los Cursos Libres es la siguiente:

- Crear un proceso optimizado de inscripción a los Cursos Libres.

En este trabajo se expone la nueva plataforma informática que surge como respuesta a las necesidades presentadas. Esta plataforma cuenta con dos sistemas: el sistema de vinculación de egresados y el sistema de Cursos Libres. Estos funcionan como un sitio web, acorde a las tecnologías que se poseen dentro de la institución.

El sistema de vinculación de egresados está disponible para los egresados y las empresas seleccionadas por la Facultad de Arquitectura. Su finalidad es ayudar al egresado a ubicarse de forma más eficiente dentro del mercado laboral.

Para ello, se busca que las empresas puedan localizar fácilmente a los egresados mediante un catálogo de información. Adicionalmente, se busca que la Facultad tenga un canal más accesible para poder comunicarse con los egresados, mediante un sistema de difusión de información dentro del sistema.

En respuesta a las necesidades del proceso de inscripción para Cursos Libres, se busca implementar el mismo de manera ágil y optimizada, permitiendo que el interesado pueda realizar su proceso de inscripción mediante internet.

1. FASE DE INVESTIGACIÓN

1.1. Antecedentes de la empresa

1.1.1. Reseña histórica

La Facultad de Arquitectura se funda, según consta en el Acta No. 657 del Honorable Consejo Superior Universitario, el día 7 de junio de 1958. El acto Inaugural se realizó el viernes 5 de septiembre de ese mismo año. Con motivo de esa fecha, a solicitud del Colegio de Arquitectos al Congreso de la República, posteriormente, se declaró oficialmente el “Día del Arquitecto”.

El primer Decano Interino de la Facultad fue el Arquitecto Roberto Aycinena Echeverría, convirtiéndose pocos años después en el primer Decano electo.

Fue a partir de 1971 que la Facultad cuenta con edificio propio, el actual edificio T-2. En 1972 se inicia un movimiento transformador en la enseñanza en la facultad de Arquitectura, dando como resultado el Congreso de Reestructuración de Arquitectura -CRA- el 10 de mayo de 1972. A partir del CRA el pensum tuvo un enfoque social humanístico, el cual fue adecuado luego de la experiencia del Terremoto de 1976. A partir del 1982, el pensum tuvo un enfoque tecnológico.

En 1987, en la administración del Arq. Eduardo Aguirre Cantero, se inicia el programa de Técnico en Diseño Gráfico, la carrera tuvo éxito rápidamente, el que se evidencio a través del posicionamiento laboral de sus primeros egresados.

En 1994 se realiza una readecuación sistematizada del pensum de la carrera de Arquitectura, planteada como un trabajo integral y científico, creándose el pensum de estudios para la cohorte 1995-2000, aprobado por el Consejo Superior Universitario 18 de noviembre de 1994.

En 1998 se lleva a cabo una nueva propuesta de readecuación de la Licenciatura en Arquitectura, aprobada el 2 de Mayo de 2002 por Junta Directiva de la Facultad, iniciando su implementación en el año 2003.¹

1.1.2. Misión

En la Unidad Académica, de la Universidad de San Carlos de Guatemala, responsable de ordenar y producir conocimientos, formar profesionales creativos en

¹ Facultad de Arquitectura. <http://farusac.edu.gt/administracion>. Consulta: julio de 2017.

el campo de la arquitectura, el diseño visual y especialidades, con principios éticos comprometidos y competentes, con responsabilidad en la sostenibilidad ambiental, para proponer soluciones a los problemas de la sociedad en su ámbito; desempeñándose en el campo laboral con excelencia y disciplina por el bien de la cultura, el ambiente y el mejoramiento de planificación, organización, desarrollo espacial y comunicación visual.²

1.1.3. Visión

Ser una institución líder en la formación de profesionales creativos y éticos en los campos de la arquitectura, el diseño visual, especialidades y otros que demande la sociedad guatemalteca, con las competencias y principios de responsabilidad en la sostenibilidad Ambiental. Con programas académicos acreditados internacionalmente por su actualización, calidad y excelencia. Con capacidad de proponer soluciones para los problemas nacionales dentro de su ámbito y brindar una respuesta eficaz a los requerimientos del mercado laboral. Con un gobierno democrático, una administración efectiva y con capacidad de gestión y condiciones adecuadas de infraestructura, financiamiento y recursos tecnológicos.³

1.1.4. Servicios que realiza

Formar profesionales de alto nivel académico en el campo de la arquitectura, diseño gráfico y otras especialidades en ramas afines, orientadas a atender con calidad, eficiencia, eficacia y pertinencia, las demandas de la sociedad guatemalteca.

Generar conocimientos científico-tecnológico y social-humanístico, por medio de programas de investigación, en función de las características del medio, oportunidades y necesidades sociales.

Vincular el proceso formativo y la investigación a los programas de extensión para contribuir en la solución de los programas y necesidades de la sociedad guatemalteca en el ámbito de su competencia.

² Facultad de Arquitectura. <http://farusac.edu.gt/administracion>. Consulta: julio de 2017.

³ *Ibíd.*

1.2. Descripción de las necesidades

Actualmente la Facultad de Arquitectura cuenta con un espacio para informar a los egresados sobre ofertas de contratación en el sector laboral, pero no existe una herramienta para establecer la comunicación directa y masiva con los mismos. Por ello, surge la necesidad de crear un sistema informático que dé soporte a la vinculación que la Facultad mantiene con sus egresados.

Adicionalmente, la Facultad de Arquitectura tiene la necesidad de obtener más información relacionada con sus egresados, por lo que surge el requerimiento de crear una base de datos de los egresados que pueda ser llenada con información actual y válida.

Asimismo, se necesita que el perfil del egresado pueda ser dado a conocer entre las empresas del ámbito laboral del país, por lo que se requiere que dentro de esta plataforma exista un espacio para la colocación de estos perfiles.

La Facultad de Arquitectura desea crear un ambiente de cooperación con las empresas del país, por lo que se busca la creación de convenios para que los egresados de la Facultad de Arquitectura tengan mayores oportunidades de posicionarse en el mercado mediante la obtención de las ofertas laborales de las empresas de primera mano.

La Facultad de Arquitectura tiene la intención de incluirse en el programa de Cursos Libres mediante el control sobre los cursos a impartir y el proceso de inscripción a los mismos. De ahí, surge la necesidad de implementar el proceso mediante la creación de una herramienta de software para que los estudiantes tramiten su inscripción de forma personal.

Además, se desea generar informes con base en las estadísticas de inscripción a los Cursos Libres, por lo que la herramienta debe generar ese reporte.

1.3. Priorización de las necesidades

Entre las necesidades presentadas se dará prioridad a la creación de la base de datos de egresados de la Facultad de Arquitectura y la colocación de sus perfiles a la disposición de las empresas con las que se desea establecer cooperación. La necesidad de obtener información acerca del conocimiento que los egresados de la Facultad ponen a disposición del país para implementarlo en el trabajo diario es una prioridad alta para buscar áreas de interés y especialización que, en algún futuro, puedan servir como base para modificaciones curriculares.

Se dará importancia primordial a la publicación de ofertas laborales de las empresas, para que los egresados se informen fácilmente dentro de la plataforma. Esta herramienta confiere importancia a la cooperación con las mayores compañías del mercado nacional, las cuales emplean a los egresados de la Facultad, es decir, se propicia la oportunidad de empleo.

Finalmente, se le dará importancia a la automatización del proceso de inscripción y reportes para los Cursos Libres, tomando en cuenta que no es un proceso crítico de la Facultad, pero requiere atención.

2. FASE TÉCNICO PROFESIONAL

2.1. Descripción del proyecto

El proyecto consiste en la implementación de una nueva plataforma informática dentro de la Facultad de Arquitectura. Esta plataforma cuenta con dos áreas principales:

2.1.1. Vinculación de egresados

Esta funcionalidad se relaciona directamente con los egresados de la Facultad de Arquitectura y tiene tres ejes principales:

2.1.1.1. Recopilación de información

Dentro de la plataforma se recopilan datos de los egresados de la Facultad de Arquitectura al momento de crear un perfil público, esta información se almacena en una nueva base de datos.

2.1.1.2. Facilitación de oportunidades laborales

Dentro de la plataforma se hacen públicos los perfiles de los egresados. Esto permite que profesionales y empresas que tengan acceso a la plataforma puedan visualizar un catálogo de información de los egresados, y así, se facilite la generación de oportunidades de empleo y redes de contactos.

Adicionalmente, las empresas podrán publicar ofertas de trabajo, las cuales serán reguladas por la administración de la plataforma, y de esta forma, los egresados podrán ver las ofertas directamente dentro de la aplicación.

2.1.1.3. Divulgación de información

Dentro de la plataforma existe un canal de comunicación en el cual la Facultad de Arquitectura puede publicar información para sus egresados, quienes la verán dentro del sistema.

2.1.2. Cursos Libres

Mediante esta funcionalidad, el interesado se inscribirá de forma virtual en los Cursos Libres de la Facultad de Arquitectura. Para ello, debe ingresar su información a una nueva base de datos.

Esta información se presenta al administrador del sistema como una serie de reportes estadísticos.

2.2. Investigación preliminar para la solución del proyecto

Para la implementación del proyecto se investigaron los siguientes conceptos:

2.2.1. Propios de la lógica del proyecto

2.2.1.1. Bolsa de empleo

Una bolsa de trabajo es un instrumento con una finalidad doble: permite a las empresas tener una reserva de contactos que están interesados en ofertas laborales y los aspirantes pueden conocer las ofertas laborales que las empresas publican. De esta forma facilitan la contratación de personal.

Una modalidad empleada en las bolsas de trabajo virtuales incluye el registro del perfil curricular del interesado en la página de la bolsa. Luego, la página le envía las ofertas de trabajo que se registren en relación con su área de interés. Esto se presenta como una ventaja para aquellas personas que no tienen tiempo o no saben cómo elaborar un currículum, ya que los portales suelen poseer un formato particular en el cual solamente deben ingresar sus datos más relevantes. Otra funcionalidad de las bolsas de empleo es la publicación de noticias e información relacionada con su área, a través de publicidad mostrada en estas páginas⁴.

2.2.1.2. Red de contactos profesionales

Una red de contactos profesionales es una herramienta que interconecta a distintos profesionales de una o más áreas, para conformar un grupo de contactos, para que puedan compartir información laboral y generar sociedades, grupos de trabajo o emprendimientos⁵.

⁴ <http://www.degerencia.com/area.php?areaid=10028>. Consulta: marzo de 2017.

⁵ <http://www.enaes.es/blog/que-es-el-networking>. Consulta: marzo de 2017.

Algunos de los beneficios directos que se buscan de esta práctica son:

- Reforzar relaciones laborales.
- Dar a conocer algún producto o servicio de forma eficiente.
- Acceder a personas o entidades, que puedan resultar de interés para el negocio.
- Adquirir un mayor grado de conocimiento acerca del entorno laboral, aprendiendo a valorar sus riesgos y oportunidades.
- Realizar nuevos clientes o socios.
- Aumentar el rango de posibilidades comerciales del negocio.

2.2.2. Propios de la implementación técnica

2.2.2.1. Scrum

Es un marco de trabajo de manejo de proyectos que aplica los principios del movimiento de desarrollo ágil. Scrum sigue una línea de trabajo iterativa e incremental, que espera entregar productos funcionando en cada iteración y, mediante la colaboración con el cliente, enfocarse en el área del proyecto que se requiera.

El equipo de desarrollo Scrum está conformado por distintos roles, entre ellos está el *Scrum Master*, quien es homólogo al administrador del proyecto, pero difiere de él en varias maneras, incluyendo que no da órdenes al equipo ni asigna tareas, sino se encarga de mantener al equipo enfocado y se ocupa de los asuntos externos al desarrollo, para no distraer al resto del equipo. Otro rol del equipo de desarrollo es el de *Product Owner*, quien es el cliente o su representante y tiene como responsabilidad crear la visión del producto y transmitirla al equipo, guiándolos en la dirección correcta durante el desarrollo y

manteniendo relevantes los intereses del cliente. El otro rol es el de miembro de equipo, en el cual todos los miembros tienen el mismo nivel de importancia y cada uno debe contribuir de la manera que puedan para completar el trabajo de la iteración.

El proceso de desarrollo en Scrum se divide en iteraciones llamadas *sprint* las cuales tienen una duración máxima de un mes. Al inicio de cada *sprint* los miembros del equipo determinan cuántos elementos pueden trabajar durante el mismo y crean una lista de tareas llamada *sprint backlog*, que desarrollarán. Durante el *sprint*, el equipo de desarrollo toma la lista de tareas y las desarrolla completamente: codificadas, probadas e integradas al proyecto. Cada día del *sprint* suele llevarse a cabo una reunión rápida del equipo de desarrollo, que no sobrepase 15 minutos, donde se indica cuál fue el trabajo realizado el día anterior, cuál será el trabajo de ese día y qué impedimentos o problemas podrían surgir.

Al final de cada iteración, el equipo lleva a cabo una reunión llamada *sprint review*, en la cual se demuestra la nueva funcionalidad al *Product Owner* o dueño del producto, quien es el encargado, de parte del cliente, o el cliente mismo, de guiar al equipo hacia el producto que realmente quiere el cliente y éste hace la retroalimentación sobre la funcionalidad. Esto se toma en cuenta en las iteraciones consecuentes. Al finalizar la iteración, existe otra reunión del equipo de desarrollo, llamada *sprint retrospective*, la cual sirve para reflexionar sobre el trabajo realizado y buscar oportunidades de mejora⁶.

⁶ <https://www.mountaingoatsoftware.com/agile/scrum>. Consulta: noviembre de 2016.

2.2.2.2. Javascript

“Es un lenguaje ligero e interpretado, orientado a objetos con funciones de primera clase, más conocido como el lenguaje de script para páginas web, pero también usado en muchos entornos sin navegador ... Es un lenguaje script multi-paradigma, basado en prototipos, dinámico, soporta estilos de programación funcional, orientada a objetos e imperativa”⁷.

2.2.2.3. PHP

“PHP, acrónimo de "PHP: Hypertext Preprocessor", es un lenguaje de 'scripting' de propósito general y de código abierto que está especialmente pensado para el desarrollo web y que puede ser embebido en páginas HTML”⁸.

Una de las características principales de PHP es que es ejecutado del lado del servidor, por lo que el cliente recibe el HTML generado pero no conoce el código fuente que lo generó, al contrario de los lenguajes de programación que se ejecutan del lado del cliente cuya fuente es incrustada en el HTML.

2.2.2.4. Composer

Es un manejador de dependencias *open source* para proyectos en lenguaje de programación PHP, por lo que se encarga de instalar y actualizar los paquetes necesarios para el funcionamiento del proyecto. Composer también implementa un sistema de auto carga, suscrito a los estándares del PHP-FIG, con el cual permite incluir las dependencias que instala y manejar espacios de nombres en PHP.

⁷ <https://developer.mozilla.org/es/docs/Web/JavaScript>. Consulta: agosto de 2017.

⁸ <https://secure.php.net/manual/es/preface.php>. Consulta: noviembre de 2016.

Composer instala los paquetes de forma local, incluyendo una carpeta dentro del proyecto basándose en un archivo en el que se declaran las dependencias del proyecto⁹.

Composer utiliza como repositorio principal a Packagist, en el cual se encuentran paquetes públicos que pueden descargarse.

2.2.2.5. Zend Framework 3

Es un marco de trabajo *open source* para el lenguaje de programación PHP, en su versión 5.6 o superior, que incluye una serie de paquetes y componentes orientados a objetos que facilitan el desarrollo de aplicaciones y servicios web.

Zend Framework utiliza Composer como su manejador de dependencias y se suscribe a los estándares de PHP-FIG¹⁰.

Entre los principales paquetes que abarca Zend Framework están¹¹:

- Zend-authentication: se encarga de autenticar a los usuarios mediante adaptadores y validar su identidad dentro de la aplicación.
- Zend-cache: Se encarga de la implementación de caché.
- Zend-component-installer: interactúa con Composer para instalar e inyectar los componentes y su configuración.
- Zend-config: lee y escribe archivos de configuración.
- Zend-crypt: son herramientas de codificación y decodificación.
- Zend-db: es una interfaz entre la abstracción de la base de datos y la aplicación.

⁹ <https://getcomposer.org/doc/00-intro.md>. Consulta: marzo de 2017.

¹⁰ <https://framework.zend.com/about>. Consulta: noviembre de 2016.

¹¹ <http://zendframework.github.io/>. Consulta: agosto de 2017.

- Zend-debug: muestra información de depuración.
- Zend-eventmanager: se encarga de la implementación de eventos.
- Zend-form: se encarga de transformar formularios en objetos.
- Zend-http: se encarga de implementar clientes y abstracciones al protocolo HTTP.
- Zend-json: se encarga de implementar serializaciones de JSON en PHP.
- Zend-mail: se encarga del manejo de correos.
- Zend-modulemanager: se encarga del manejo de módulos dentro de la aplicación.
- Zend-mvc: es una implementación del patrón Modelo Vista Controlador.
- Zend-router: implementa el enrutamiento en la aplicación.
- Zend-serializer: serializa y deserializa estructuras PHP.
- Zend-service manager: implementa el inyector de dependencias basado en fábricas.
- Zend-session: implementa manejo de sesiones PHP.
- Zend-view: implementa capa de vistas.

2.2.2.6. Apache HTTP Server

Es una implementación *open source* de un servidor web de grado comercial y gratuito que utiliza el protocolo HTTP para transmitir archivos a través de la red¹².

Apache puede utilizar PHP para servir el resultado de la ejecución de estos archivos, creando así una plataforma web dinámica.

¹² https://httpd.apache.org/ABOUT_APACHE.html. Consulta: noviembre de 2016.

2.2.2.7. MariaDB

Es un servidor de base de datos relacionales, desarrollado como sustituto *open source* para MySQL, proveyendo acceso a la información mediante SQL¹³.

2.2.2.8. Doctrine

“El Proyecto Doctrine es hogar a varias librerías PHP primariamente enfocadas en el almacenamiento de base de datos y el mapeo de objetos. Los proyectos núcleo son: Object Relational Mapper (ORM) y Database Abstraction Layer (DBAL)”¹⁴.

Doctrine permite el mapeo de tablas de la base de datos a objetos PHP, los cuales pueden utilizarse dentro de los scripts. Además, presenta compatibilidad tanto con Zend Framework 3 como con MariaDB.

2.2.2.9. Git

Git es una herramienta de control de versiones distribuido, gratis y *open source*¹⁵.

Al ser un control de versiones distribuido, Git permite el almacenamiento de repositorios en cada una de las máquinas que se esté utilizando, en lugar de tener solamente un repositorio central.

¹³ <https://mariadb.org/about/>. Consulta: noviembre de 2016.

¹⁴ <http://www.doctrine-project.org/>. Consulta: agosto de 2017.

¹⁵ <https://git-scm.com/>. Consulta: marzo de 2017.

Existen servidores Git públicos y en línea como GitHub y GitBucket, los cuales almacenan el código de los proyectos *open source* para lograr que estén disponibles al público.

2.2.2.10. Vagrant

Es una herramienta para la creación de entornos de desarrollo mediante la virtualización.

Esta herramienta utiliza software de virtualización como VirtualBox para generar las máquinas virtuales y luego provisionarlas, ejecutando pasos predefinidos, para que el proceso de generar un ambiente de desarrollo, sea automatizado y genere el mismo resultado para todas las máquinas destino, por lo que no existen problemas de compatibilidad¹⁶.

2.2.2.11. Debian

El sistema operativo Debian es un sistema operativo GNU/Linux¹⁷. Esta distribución de GNU/Linux valora la estabilidad y la seguridad sobre otros aspectos, por lo que suele especializarse en servidores.

2.2.2.12. Yarn

Es un manejador de dependencias *open source* para paquetes. Los paquetes se especifican en un archivo dentro del proyecto y son instalados localmente¹⁸.

¹⁶ <https://www.vagrantup.com/docs/why-vagrant/>. Consulta: marzo de 2017.

¹⁷ <https://www.debian.org/intro/about>. Consulta: noviembre de 2016.

¹⁸ <https://yarnpkg.com/lang/en/>. Consulta: agosto de 2017.

Yarn tiene como repositorio principal a npm, el cual tiene principalmente paquetes del lenguaje Javascript.

2.2.2.13. Sass

Es un preprocesador de archivos CSS escrito en el lenguaje Ruby, el cual permite extender el lenguaje básico para utilizar variables, reglas de estilo anidadas, *mixins* e importar otras hojas de estilo¹⁹.

2.2.2.14. Bootstrap

Es una librería de componentes de desarrollo *front-end* incluyendo herramientas de HTML, CSS y Javascript (utilizando JQuery) trabajando en conjunto para construir páginas web adaptables²⁰.

La versión de Bootstrap compatible con Sass puede descargarse utilizando Yarn, y permite recompilar los archivos de estilo.

2.3. Presentación de la solución al proyecto

2.3.1. Especificaciones técnicas

El lenguaje de programación principal para el proyecto es PHP versión 5.6. El desarrollo de la plataforma tomará como base el marco de trabajo Zend Framework 3.

¹⁹ http://sass-lang.com/documentation/file.SASS_REFERENCE.html. Consulta: agosto de 2017.

²⁰ <https://getbootstrap.com/>. Consulta: agosto de 2017.

La plataforma utilizará una base de datos MariaDB, la cual es una base de datos con un modelo entidad-relación nuevo y la plataforma utilizará Doctrine ORM para manipular la información almacenada en ella.

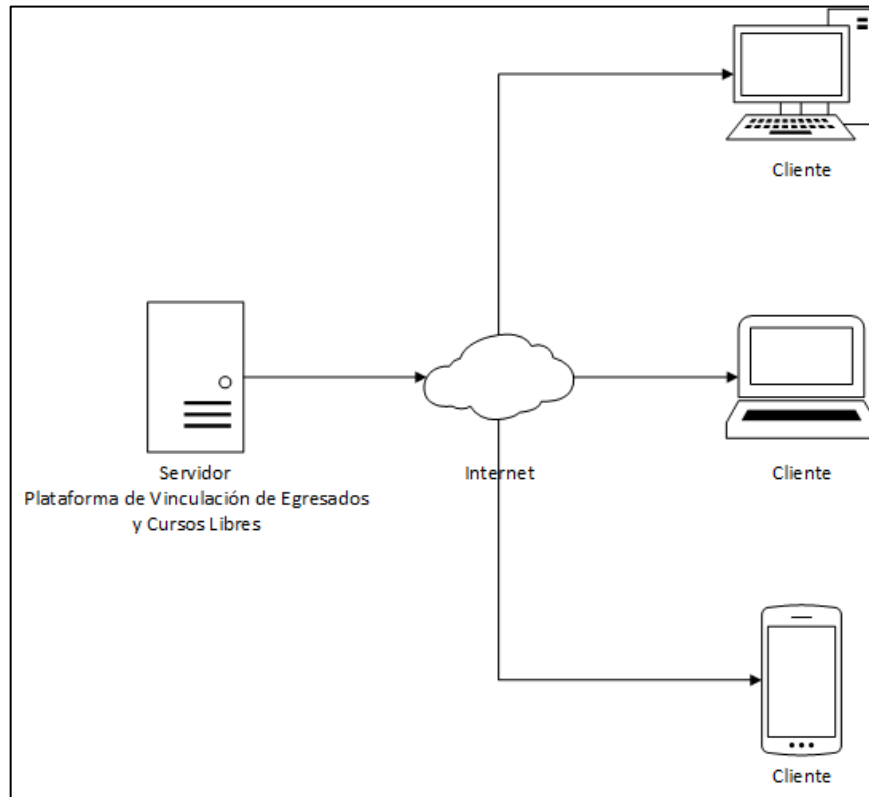
La plataforma será desplegada sobre un servidor Debian, y utilizará un servidor web Apache.

Para el *front-end* la plataforma utiliza la librería Bootstrap para proveer un diseño base y el lenguaje de programación Javascript en combinación con JQuery en algunas de sus páginas para proveer la funcionalidad deseada.

2.3.2. Arquitectura

La arquitectura utilizada es Cliente-Servidor. Se sirve la plataforma desde un servidor interno en la Facultad de Arquitectura y se coloca para su acceso mediante internet, donde los clientes podrán acceder con el navegador de su dispositivo (ver figura 1).

Figura 1. **Diagrama de arquitectura de la plataforma**



Fuente: elaboración propia.

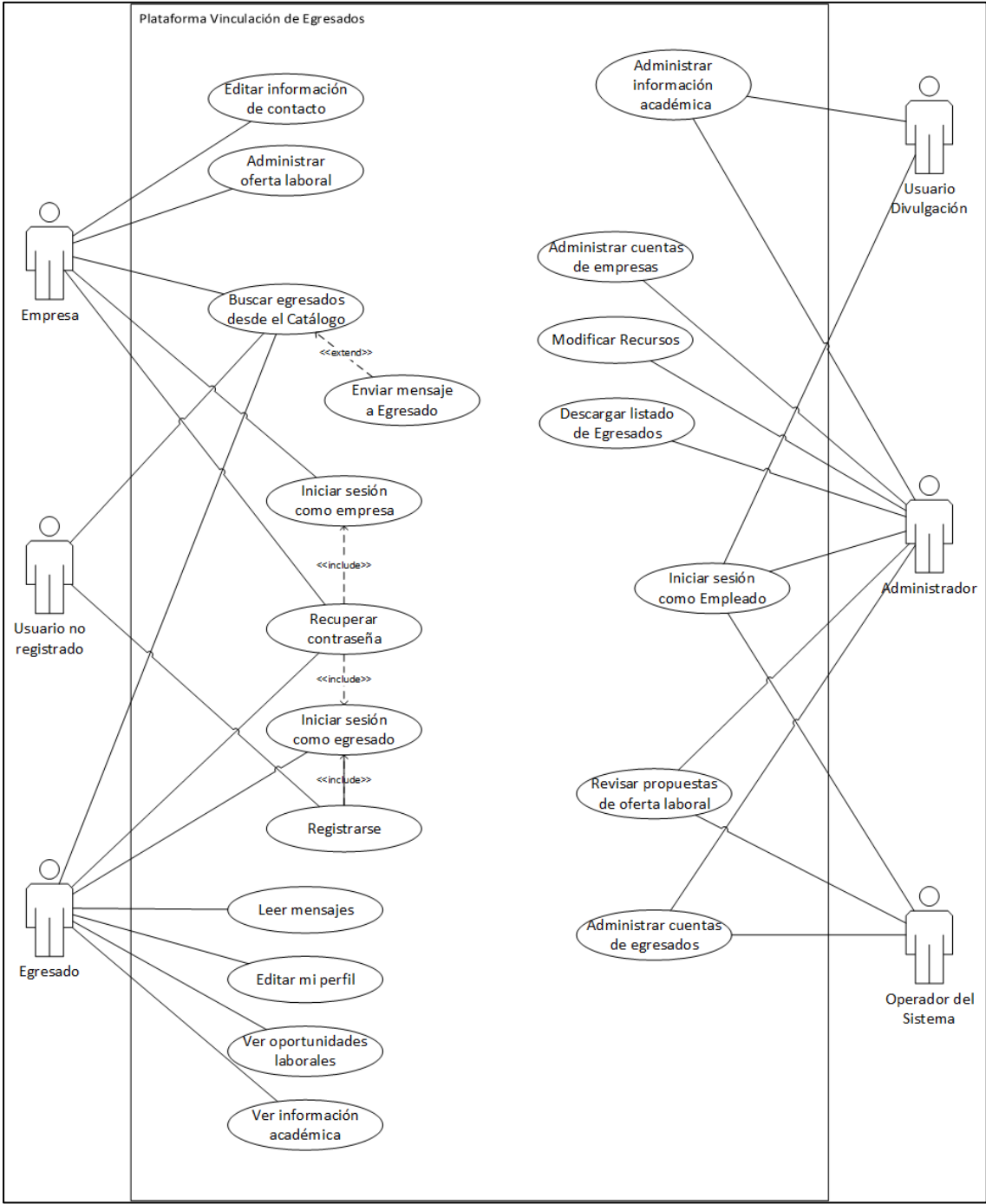
2.3.3. Casos de uso

A continuación, se presentan los casos de uso para la plataforma:

2.3.3.1. Diagrama de casos de uso para el sistema de vinculación de egresados

El diagrama general de casos de uso para el sistema de vinculación de egresados (ver figura 2):

Figura 2. Diagrama de casos de uso para el sistema de vinculación de egresados



Fuente: elaboración propia.

2.3.3.2. Casos de uso para el sistema de vinculación de egresados

A continuación, se muestran los casos de uso para el sistema de vinculación a egresados (ver tablas I a XVIII).

Tabla I. **Caso de uso VE-1 Registrarse**

Nombre Caso de Uso	Registrarse
Flujo básico	1. El usuario solicita registrarse en el sistema.
	2. El sistema solicita el número de carné y la contraseña del sistema de FARUSAC ya existente.
	3. El usuario ingresa su número de carné y su contraseña.
	4. El sistema llama al web <i>service</i> de autenticación para validar que sea un egresado.
	5. El sistema obtiene del web <i>service</i> el nombre, la carrera y el año de graduación del egresado.
	6. El sistema muestra al usuario su información y solicita nuevos datos.
	7. El usuario ingresa su número de CUI, su correo electrónico y su nueva contraseña.
	8. El sistema guarda el usuario e inicia su sesión, redirigiéndolo al formulario de creación de perfil.
	9. Fin del caso de uso.
Flujo alterno	5.a.1 El web <i>service</i> indica que la información proporcionada no es correcta.
	5.a.2 El sistema muestra un mensaje al usuario indicando que la información es incorrecta.
	5.b.1 El web <i>service</i> falla por algún motivo externo.
	5.b.2 El sistema muestra un mensaje de error.
	8.1 El sistema no puede guardar al usuario debido a información errónea o repetida y muestra un error.
Requerimientos especiales	El caso de uso debe acoplarse al servicio web que valida los datos ingresados.
Precondiciones	El usuario no debe de estar registrado anteriormente.
Poscondiciones	El usuario ahora puede ingresar al sistema.
Puntos de extensión	Al registrarse, se incluye el Inicio de Sesión de Egresado.

Fuente: elaboración propia.

Tabla II. **Caso de uso VE-2 Iniciar sesión como egresado**

Nombre Caso de Uso	Iniciar sesión como egresado
Flujo básico	1. El usuario solicita iniciar sesión en el sistema.
	2. El sistema solicita el CUI y la contraseña al usuario.
	3. El usuario ingresa su CUI y su contraseña.
	4. El sistema valida que las credenciales sean correctas.
	5. El sistema inicia la sesión del egresado y lo redirige a la página de inicio de egresados.
	6. Fin del caso de uso.
Flujo alterno	4.1 El sistema determina que el CUI y contraseña ingresados no son válidos y muestra un mensaje al usuario.
Requerimientos especiales	No
Precondiciones	El usuario no debe tener una sesión ya activa.
Poscondiciones	El usuario inicia sesión en el sistema.
Puntos de extensión	No

Fuente: elaboración propia.

Tabla III. **Caso de uso VE-3 Recuperar contraseña**

Nombre Caso de Uso	Recuperar contraseña
Flujo básico	1. El usuario solicita recuperar su contraseña.
	2. El sistema solicita el CUI si es egresado o el correo electrónico si es empresa al usuario.
	3. El usuario ingresa su CUI o su correo.
	4. El sistema valida que el usuario exista y no haya solicitado cambiar contraseña actualmente.
	5. El sistema envía un correo electrónico a la cuenta de correo asociada al usuario con un enlace hacia la página de recuperación de contraseña.
	6. El usuario revisa su correo y presiona el enlace.
	7. El sistema solicita al usuario su nueva contraseña.
	8. El usuario ingresa su nueva contraseña.
	9. El sistema asigna la nueva contraseña y realiza el inicio de sesión del usuario.
	10. Fin del caso de uso.

Continuación tabla III.

Flujo alterno	4.1. El sistema no encuentra el usuario o determina que el usuario ya solicitó recuperar su contraseña actualmente y muestra un mensaje indicándolo
Requerimientos especiales	El sistema debe enviar el correo electrónico utilizando su propia cuenta.
Precondiciones	El usuario no debe tener una sesión ya activa.
Poscondiciones	El usuario cambia su contraseña.
	El usuario inicia sesión en el sistema.
Puntos de extensión	Incluye el caso de uso Iniciar sesión como egresado.
	Incluye el caso de uso Iniciar sesión como empresa.

Fuente: elaboración propia.

Tabla IV. **Caso de uso VE-4 Editar mi perfil**

Nombre Caso de Uso	Editar mi perfil
Flujo básico	1. El usuario solicita editar su perfil.
	2. El sistema solicita al egresado la información de su perfil.
	3. El usuario ingresa la información de su perfil.
	4. El sistema valida la información ingresada.
	5. El sistema almacena la información del perfil del egresado.
	6. Fin del caso de uso.
Flujo alterno	4.1. El sistema determina que la información no es correcta o no puede ser almacenada y muestra un mensaje indicándolo
Requerimientos especiales	Si el usuario no ha editado su perfil por primera vez no puede acceder a las otras funcionalidades del sistema.
Precondiciones	El usuario debe haber iniciado sesión.
Poscondiciones	El usuario edita su información de perfil.
Puntos de extensión	No.

Fuente: elaboración propia.

Tabla V. **Caso de uso VE-5 Ver oportunidades laborales**

Nombre Caso de Uso	Ver oportunidades laborales
Flujo básico	1. El usuario solicita ver las oportunidades laborales.
	2. El sistema muestra las oportunidades laborales actualmente publicadas.
	3. El usuario selecciona una oferta laboral.
	4. El sistema muestra la información relacionada a la oferta laboral seleccionada.
	5. El usuario puede retornar al paso 3.
	6. Fin del caso de uso.
Flujo alterno	2.1. El sistema determina que no hay ofertas laborales publicadas y muestra un mensaje indicándolo.
Requerimientos especiales	No.
Precondiciones	El usuario debe haber iniciado sesión.
	Debe existir al menos una oferta laboral publicada.
Poscondiciones	No.
Puntos de extensión	No.

Fuente: elaboración propia.

Tabla VI. **Caso de uso VE-6 Ver información académica**

Nombre Caso de Uso	Ver información académica
Flujo básico:	1. El usuario solicita ver la información académica.
	2. El sistema muestra las publicaciones existentes.
	3. El usuario selecciona una publicación.
	4. El sistema muestra la información relacionada a la publicación seleccionada.
	5. El usuario puede retornar al paso 3.
	6. Fin del caso de uso.
Flujo alterno	2.1. El sistema determina que no hay publicaciones existentes y muestra un mensaje indicándolo.
Requerimientos especiales	No.
Precondiciones	El usuario debe haber iniciado sesión.
	Debe existir al menos una publicación académica.
Poscondiciones	No.
Puntos de extensión	No.

Fuente: elaboración propia.

Tabla VII. **Caso de uso VE-7 Buscar egresados desde el catálogo**

Nombre Caso de Uso	Buscar egresados desde el catálogo
Flujo básico	1. El usuario solicita buscar egresados desde el catálogo.
	2. El sistema muestra un filtro por áreas de desempeño y especialización.
	3. El usuario selecciona el filtro que desea o ninguno.
	4. El sistema muestra los resultados de la búsqueda de egresados.
	5. El usuario puede regresar al paso 2.
	6. El usuario selecciona un perfil de egresado.
	7. El sistema muestra el perfil del egresado seleccionado.
	8. El usuario puede regresar al paso 4.
	9. Fin del caso de uso.
Flujo alternativo	4.1 El sistema muestra que no encontró ningún resultado para el filtro especificado.
Requerimientos especiales	El usuario que no está registrado dentro de la plataforma solamente podrá ver un perfil limitado, que no contiene la información completa del egresado.
Precondiciones	Debe existir al menos un perfil de egresado.
Poscondiciones	No.
Puntos de extensión	Si el usuario es de tipo no registrado, podrá enviar un mensaje para contactar al egresado (VE-8).

Fuente: elaboración propia.

Tabla VIII. **Caso de uso VE-8 Enviar mensaje a egresado**

Nombre Caso de Uso	Enviar mensaje a egresado
Flujo básico	1. El usuario solicita enviar un mensaje al egresado.
	2. El sistema muestra el formulario para ingreso del mensaje.
	3. El usuario ingresa la información solicitada.
	4. El sistema verifica que la información sea válida y procede a guardar el mensaje.
	5. Fin del caso de uso.
Flujo alternativo	4.1. El sistema determina que la información no es válida o no puede almacenarse y muestra un mensaje indicándolo.

Continuación tabla VIII.

Requerimientos especiales	El formulario debe incluir validaciones para prevenir que se realice spam a los egresados.
Precondiciones	El usuario debe haber buscado en el catálogo.
	El usuario debe ser un usuario no registrado.
Poscondiciones	El egresado tiene un nuevo mensaje en su bandeja de entrada.
Puntos de extensión	No.

Fuente: elaboración propia.

Tabla IX. **Caso de uso VE-9 Iniciar sesión como empresa**

Nombre Caso de Uso	Iniciar sesión como empresa
Flujo básico	1. El usuario solicita iniciar sesión en el sistema.
	2. El sistema solicita el correo electrónico y la contraseña al usuario.
	3. El usuario ingresa su correo electrónico y su contraseña.
	4. El sistema valida que las credenciales sean correctas.
	5. El sistema inicia la sesión de la empresa y redirige al usuario a la página de inicio de empresa.
	6. Fin del caso de uso.
Flujo alterno	4.1 El sistema determina que el correo electrónico y contraseña ingresados no son válidos y muestra un mensaje al usuario.
Requerimientos especiales	No
Precondiciones	El usuario no debe tener una sesión ya activa.
Poscondiciones	El usuario inicia sesión en el sistema.
Puntos de extensión	No

Fuente: elaboración propia.

Tabla X. **Caso de uso VE-10 Administrar oferta laboral**

Nombre Caso de Uso	Administrar oferta laboral
Flujo Básico	1. El usuario solicita administrar las ofertas laborales. 2. El sistema muestra las ofertas laborales existentes de la empresa. 3. El usuario selecciona crear, modificar o eliminar una oferta. 4. El sistema muestra el formulario apropiado para la acción solicitada. 5. El usuario ingresa la información apropiada para la acción solicitada. 6. El sistema verifica la información y procede a ejecutar la acción solicitada. 7. Fin del caso de uso.
Flujo alterno	6.1 El sistema determina que la información ingresada no es válida y muestra un mensaje al usuario.
Requerimientos especiales:	No
Precondiciones	El usuario debe haber iniciado sesión como empresa.
Poscondiciones	La oferta laboral es creada, modificada o eliminada según corresponda.
Puntos de extensión	No

Fuente: elaboración propia

Tabla XI. **Caso de uso VE-11 Editar información de contacto**

Nombre Caso de Uso	Editar información de contacto
Flujo básico	1. El usuario solicita editar la información de la empresa. 2. El sistema muestra el formulario para la edición de información de la empresa. 3. El usuario ingresa la información solicitada. 4. El sistema valida que la información sea válida y procede a guardar la información de contacto de la empresa. 5. Fin del caso de uso.
Flujo alterno	4.1 El sistema determina que la información ingresada no es válida y muestra un mensaje al usuario indicándolo.

Continuación tabla XI.

Requerimientos especiales	No
Precondiciones	El usuario debe haber iniciado sesión como empresa.
Poscondiciones	La información de la empresa es actualizada.
Puntos de extensión	No

Fuente: elaboración propia.

Tabla XII. **Caso de uso VE-12 Iniciar sesión como empleado**

Nombre Caso de Uso	Iniciar sesión como empleado
Flujo básico	1. El usuario solicita iniciar sesión en el sistema.
	2. El sistema solicita el registro personal y la contraseña al usuario.
	3. El usuario ingresa su registro personal y su contraseña.
	4. El sistema valida que las credenciales sean correctas mediante el consumo de un servicio web y recibe el rol del usuario.
	5. El sistema inicia la sesión del usuario y redirige al usuario a la página de inicio de su respectivo rol.
	6. Fin del caso de uso.
Flujo alterno	4.a. El servicio web indica que las credenciales son inválidas y se muestra un mensaje indicándolo.
	4.b. El servicio web no puede ser alcanzado por lo que se muestra un mensaje de error.
Requerimientos especiales	La autenticación depende del consumo del servicio web externo al sistema.
Precondiciones	El usuario no debe tener una sesión ya activa.
Poscondiciones	El usuario inicia sesión en el sistema.
Puntos de extensión	No

Fuente: elaboración propia.

Tabla XIII. **Caso de uso VE-13 Administrar información académica**

Nombre Caso de Uso	Administrar información académica
Flujo básico	1. El usuario solicita administrar la información académica. 2. El sistema muestra las publicaciones existentes. 3. El usuario selecciona crear, modificar o eliminar una publicación. 4. El sistema muestra el formulario apropiado para la acción solicitada. 5. El usuario ingresa la información apropiada para la acción solicitada. 6. El sistema verifica la información y procede a ejecutar la acción solicitada. 7. Fin del caso de uso.
Flujo alterno	6.1 El sistema determina que la información ingresada no es válida y muestra un mensaje al usuario.
Requerimientos especiales	No
Precondiciones	El usuario debe haber iniciado sesión con rol administrador o divulgación.
Poscondiciones	La publicación es creada, modificada o eliminada según corresponda.
Puntos de extensión	No

Fuente: elaboración propia.

Tabla XIV. **Caso de uso VE-14 Administrar cuentas de empresas**

Nombre Caso de Uso	Administrar cuentas de empresas
Flujo básico	1. El usuario solicita administrar las cuentas de empresas. 2. El sistema muestra las cuentas empresariales existentes. 3. El usuario selecciona crear o eliminar una cuenta. 4. El sistema muestra el formulario apropiado para la acción solicitada. 5. El usuario ingresa la información apropiada para la acción solicitada. 6. El sistema verifica la información y procede a ejecutar la acción solicitada. 7. Fin del caso de uso.

Continuación tabla XIV.

Flujo alterno	6.1 El sistema determina que la información ingresada no es válida y muestra un mensaje al usuario.
Requerimientos especiales	No
Precondiciones	El usuario debe haber iniciado sesión como administrador.
Poscondiciones	La cuenta empresarial es creada o eliminada según corresponda.
Puntos de extensión	No

Fuente: elaboración propia.

Tabla XV. **Caso de uso VE-15 Modificar recursos**

Nombre Caso de Uso	Modificar recursos
Flujo básico	1. El usuario solicita modificar los recursos del sistema. 2. El sistema muestra el formulario para la modificación de recursos. 3. El usuario ingresa la información solicitada. 4. El sistema valida la información y procede a modificar los recursos del sitio. 5. Fin del caso de uso.
Flujo alterno	4.1 El sistema determina que la información ingresada no es válida y muestra un mensaje al usuario.
Requerimientos especiales	No
Precondiciones	El usuario debe haber iniciado sesión como administrador.
Poscondiciones	Los recursos del sitio web son modificados.
Puntos de extensión	No

Fuente: elaboración propia.

Tabla XVI. **Caso de uso VE-16 Descargar listado de egresados**

Nombre Caso de Uso	Descargar listado de egresados
Flujo básico	1. El usuario solicita descargar el listado de egresados. 2. El sistema muestra el formulario para la descarga. 3. El usuario ingresa la información solicitada. 4. El sistema valida la información y procede a generar y gestionar la descarga del archivo Excel. 5. Fin del caso de uso.
Flujo alternativo	4.1 El sistema determina que la información ingresada no es válida y muestra un mensaje al usuario.
Requerimientos especiales	Genera un archivo tipo hoja de cálculo descargable.
Precondiciones	El usuario debe haber iniciado sesión como administrador.
Poscondiciones	No.
Puntos de extensión	No

Fuente: elaboración propia.

Tabla XVII. **Caso de uso VE-17 Revisar propuestas de oferta laboral**

Nombre Caso de Uso	Revisar propuestas de oferta laboral
Flujo básico	1. El usuario solicita revisar las propuestas de oferta laboral. 2. El sistema muestra las ofertas laborales. 3. El usuario selecciona una oferta laboral. 4. El sistema muestra la oferta laboral y el formulario para aprobar o reprobar la oferta laboral. 5. El usuario aprueba o reprueba la oferta laboral. 6. El sistema verifica la información del formulario. 6.a. Si la oferta fue aprobada, el sistema la publica. 6.b. Si la oferta fue reprobada el sistema envía un correo electrónico a la empresa, notificándole. 7. El usuario puede regresar al paso 2. 8. Fin del caso de uso
Flujo alternativo	5.1 El sistema determina que la información ingresada no es válida y muestra un mensaje al usuario.

Continuación tabla XVII.

Requerimientos especiales	El caso de uso depende del envío de correos y debe enviar el correo electrónico utilizando su propia cuenta.
Precondiciones	El usuario debe haber iniciado sesión como administrador u operador.
Poscondiciones	El estado de la oferta laboral es actualizado.
	Se envía un correo en caso de reprobarse la propuesta.
Puntos de extensión	No

Fuente: elaboración propia.

Tabla XVIII. **Caso de uso VE-18 Administrar cuentas de egresados**

Nombre Caso de Uso	Administrar cuentas de egresados
Flujo básico	1. El usuario solicita administrar las cuentas de egresados.
	2. El sistema muestra las cuentas existentes.
	3. El usuario selecciona eliminar una cuenta.
	4. El sistema muestra el formulario apropiado para eliminar la cuenta.
	5. El usuario confirma la eliminación de la cuenta.
	6. El sistema verifica la información y procede a eliminar la cuenta.
	7. El usuario puede regresar al paso 2.
	8. Fin del caso de uso.
Flujo alterno	6.1 El sistema determina que la información ingresada no es válida o no puede ejecutar la acción y muestra un mensaje al usuario.
Requerimientos especiales	No
Precondiciones	El usuario debe haber iniciado sesión con rol administrador u operador.
Poscondiciones	La cuenta de egresado es eliminada.
Puntos de extensión	No

Fuente: elaboración propia.

2.3.3.3. Historias de usuario para el sistema de vinculación de egresados

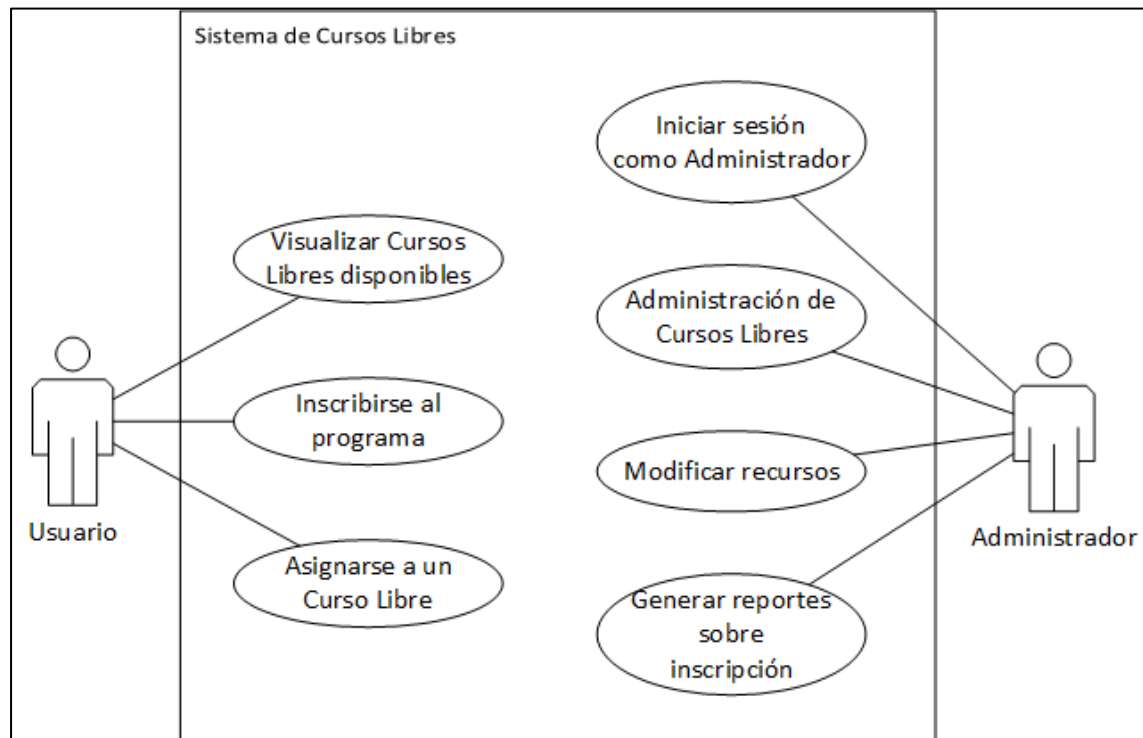
Historias de usuario referentes a la vinculación de egresados:

- El egresado puede registrarse en la plataforma.
- El egresado puede ingresar y mantener actualizada su información en la plataforma para crear un perfil de egresado.
- El egresado puede visualizar la información académica dentro de la plataforma.
- El egresado puede buscar y visualizar las ofertas laborales dentro de la plataforma.
- El usuario de divulgación puede publicar y administrar información académica dentro de la plataforma.
- La empresa puede ver el catálogo de perfiles de egresados.
- La empresa puede solicitar la publicación de ofertas laborales.
- El administrador puede revisar y publicar las ofertas laborales.
- El administrador puede administrar perfiles de usuario.
- El administrador puede administrar los recursos del sitio.

2.3.3.4. Diagrama de casos de uso para el sistema de Cursos Libres

El diagrama general de casos de uso para el sistema de Cursos Libres (ver figura 3).

Figura 3. Diagrama de casos de uso para el sistema de Cursos Libres



Fuente: elaboración propia.

2.3.3.5. Casos de uso para el sistema de Cursos Libres

A continuación, se muestran los casos de uso extendidos para el sistema de Cursos Libres (ver tablas XIX a la XXV).

Tabla XIX. **Caso de uso CL-1 Visualizar Cursos Libres disponibles**

Nombre Caso de Uso	Visualizar Cursos Libres disponibles
Flujo básico	1. El usuario solicita ver los cursos disponibles.
	2. El sistema muestra los cursos que están disponibles para el año y periodo actual.
	3. El usuario selecciona uno de los cursos.
	4. El sistema muestra los detalles del curso seleccionado.
	5. El usuario puede o no regresar al paso 3
	6. Fin del caso de uso
Flujo alternativo	2.1. Si no existen cursos disponibles el sistema muestra un mensaje indicándolo.
Requerimientos especiales	No
Precondiciones	Al menos, un curso se encuentra disponible para el año y periodo actual
Poscondiciones	No
Puntos de extensión	No

Fuente: elaboración propia.

Tabla XX. **Caso de uso CL-2 Inscribirse al programa**

Nombre Caso de Uso	Inscribirse al programa
Flujo básico	1. El usuario solicita inscribirse al programa, enviando su CUI.
	2. El sistema verifica el dato y muestra el formulario para la inscripción. Si el CUI ya está inscrito muestra la información previamente almacenada.
	3. El usuario ingresa la información solicitada.
	4. El sistema verifica la información y procede a inscribir al estudiante.
	5. Fin del caso de uso.
Flujo alternativo	2.1. El sistema determina que la información no es válida y muestra un mensaje al usuario indicándolo.
	4.1. El sistema determina que la información no es válida y muestra un mensaje al usuario indicándolo.

Continuación tabla XX.

Requerimientos especiales	No
Precondiciones	No
Poscondiciones	El usuario se inscribe.
Puntos de extensión	Ahora el usuario puede asignarse cursos (CL-3).

Fuente: elaboración propia.

Tabla XXI. **Caso de uso CL-3 Asignarse a un curso**

Nombre Caso de Uso	Asignarse a un curso
Flujo básico	1. El usuario solicita asignarse.
	2. El sistema muestra los cursos que están disponibles para el año y periodo actual.
	3. El usuario selecciona uno de los cursos.
	4. El sistema muestra los detalles del curso seleccionado y el formulario para la asignación.
	5. El usuario llena el formulario para asignarse.
	6. El sistema verifica la información y si puede realizar la operación, luego procede a asignar al usuario.
	7. El usuario puede regresar al paso 2
	8. Fin del caso de uso
Flujo alterno	2.1. Si no existen cursos disponibles el sistema muestra un mensaje indicándolo.
	6.1. Si la información es inválida o no existe cupo o no puede realizarse el almacenamiento se muestra un mensaje indicando al usuario que no se realizó su asignación.
Requerimientos especiales	Para asignarse un curso se verifica que el curso tenga cupo disponible.
Precondiciones	El usuario debe estar inscrito.
Poscondiciones	El usuario queda asignado al curso.
Puntos de extensión	No

Fuente: elaboración propia

Tabla XXII. **Caso de uso CL-4 Iniciar sesión como administrador**

Nombre Caso de Uso	Iniciar sesión como administrador
Flujo básico	1. El usuario solicita iniciar sesión en el sistema.
	2. El sistema solicita el registro personal y la contraseña al usuario.
	3. El usuario ingresa su registro personal y su contraseña.
	4. El sistema valida que las credenciales sean correctas mediante el consumo de un servicio web y recibe el rol del usuario.
	5. El sistema inicia la sesión del usuario y redirige al usuario a la página de inicio de administrador.
	6. Fin del caso de uso.
Flujo alterno	4.a. El servicio web indica que las credenciales son inválidas y se muestra un mensaje indicándolo.
	4.b. El servicio web no puede ser alcanzado por lo que se muestra un mensaje de error.
Requerimientos especiales	La autenticación depende del consumo del servicio web externo al sistema.
Precondiciones	El usuario no debe tener una sesión ya activa.
Poscondiciones	El usuario inicia sesión en el sistema.
Puntos de extensión	No

Fuente: elaboración propia.

Tabla XXIII. **Caso de uso CL-5 Administración de Cursos Libres**

Nombre Caso de Uso	Administración de Cursos Libres
Flujo básico	1. El usuario solicita administrar los Cursos Libres.
	2. El sistema muestra los cursos existentes.
	3. El usuario selecciona crear, modificar o eliminar un curso.
	4. El sistema muestra el formulario apropiado para la acción solicitada.
	5. El usuario ingresa la información apropiada para la acción solicitada.
	6. El sistema verifica la información y procede a ejecutar la acción solicitada.
	7. Fin del caso de uso.
Flujo alterno	6.1 El sistema determina que la información ingresada no es válida y muestra un mensaje al usuario.

Continuación tabla XXIII.

Requerimientos especiales	No
Precondiciones	El usuario debe haber iniciado sesión con rol administrador.
Poscondiciones	El curso es creado, modificado o eliminado según corresponda.
Puntos de extensión	No

Fuente: elaboración propia.

Tabla XXIV. **Caso de uso CL-6 Modificar recursos**

Nombre Caso de Uso	Modificar recursos
Flujo básico	1. El usuario solicita modificar los recursos del sistema.
	2. El sistema muestra el formulario para la modificación de recursos.
	3. El usuario ingresa la información solicitada.
	4. El sistema valida la información y procede a modificar los recursos del sitio.
	5. Fin del caso de uso.
Flujo alterno	4.1 El sistema determina que la información ingresada no es válida y muestra un mensaje al usuario.
Requerimientos especiales	No
Precondiciones	El usuario debe haber iniciado sesión como administrador.
Poscondiciones	Los recursos del sitio web son modificados.
Puntos de extensión	No

Fuente: elaboración propia.

Tabla XXV. **Caso de uso CL-7 Generar reportes sobre inscripción**

Nombre Caso de Uso:	Generar reportes sobre inscripción
Flujo básico	1. El usuario solicita descargar el reporte de inscripción. 2. El sistema muestra el formulario para la descarga. 3. El usuario ingresa la información solicitada. 4. El sistema valida la información y procede a generar y gestionar la descarga del archivo Excel. 5. Fin del caso de uso.
Flujo alterno	4.1 El sistema determina que la información ingresada no es válida y muestra un mensaje al usuario.
Requerimientos especiales	Genera un archivo tipo hoja de cálculo descargable.
Precondiciones	El usuario debe haber iniciado sesión como administrador.
Poscondiciones	No.
Puntos de extensión	No

Fuente: elaboración propia.

2.3.3.6. Historias de usuario para el sistema de Cursos Libres

Las historias de usuario referentes al sistema de Cursos Libres:

- El usuario del sistema puede visualizar la información sobre los Cursos Libres disponibles.
- El usuario del sistema puede inscribirse a un Curso Libre.
- El administrador puede generar reportes sobre las inscripciones a los Cursos Libres.
- El administrador puede gestionar los cursos disponibles para su inscripción.

2.3.4. Información de desarrollo

Durante el desarrollo de la plataforma se utilizaron las siguientes metodologías y herramientas:

2.3.4.1. Scrum

Durante los primeros cuatro meses del proyecto se manejaron *sprints* de dos semanas de duración, mientras que en los últimos dos meses se redujeron a una duración de una semana.

Al final de cada *sprint* se realizó la reunión *sprint review* y la *sprint retrospective*, junto al *Product Owner*, el arquitecto Oscar Enríquez y su equipo de trabajo, quienes proveían la retroalimentación que sería aplicada en el próximo *sprint*.

El arquitecto Enríquez también cumplió con algunas atribuciones del *Scrum Master*, al tratar con los trámites externos que involucrasen al proyecto, para no desenfocar al desarrollador.

2.3.4.2. Git

Desde el inicio del proyecto se versionó el código utilizando la herramienta Git, y un repositorio privado en la plataforma Bitbucket, para mantener el código disponible únicamente para el desarrollador y el personal técnico de la Facultad de Arquitectura de la Universidad de San Carlos de Guatemala.

Gracias a esto se logró que el código fuente esté almacenado de manera segura y los futuros desarrolladores tengan acceso al mismo fácilmente.

2.3.4.3. Vagrant

Durante los primeros 5 meses de desarrollo, no se contaba con un entorno de desarrollo dentro de FARUSAC, por lo que se creó un entorno de desarrollo utilizando la herramienta Vagrant, simulando las características del servidor de producción dentro de una máquina virtual.

Esta máquina virtual es portátil y puede ser versionada utilizando Git, además, permite generar un entorno de desarrollo automáticamente con las configuraciones preestablecidas para su funcionamiento, encargándose de instalar el código fuente del proyecto, PHP, Apache, MariaDB y los complementos extra necesarios.

Gracias a esto se logró que los futuros desarrolladores puedan reutilizar esta máquina virtual para simular el comportamiento del servidor de desarrollo desde cualquier ubicación y computadora.

2.3.4.4. Sass

Debido a la cantidad de cambios en el diseño de la aplicación se optó por utilizar la herramienta Yarn para descargar el código fuente de Bootstrap en su versión compatible con Sass y luego recompilar la librería con las modificaciones al estilo realizadas sobre el código fuente.

Gracias a esto, se facilitó el diseño de la página web y es posible que futuros desarrolladores puedan modificar los estilos de la página completa de manera sencilla.

2.3.5. Patrones de diseño aplicados

Dentro de Zend Framework 3 existen varios patrones de diseño en el marco de trabajo²¹ a continuación se muestra la aplicación de los mismos dentro del proyecto:

2.3.5.1. Modelo Vista Controlador

El patrón Modelo Vista Controlador se aplica teniendo por separado los Controladores, las Vistas y la lógica del negocio. En la plataforma, esto se distingue claramente al existir todos los controladores dentro de la carpeta src/Controller de cada módulo; todas las vistas dentro de la carpeta view de cada módulo y posteriormente separadas por controlador; y todos los archivos de lógica del negocio repartidos entre Manejadores, Servicios y Entidades.

2.3.5.2. Fábricas e inyección de dependencias

El patrón de fábricas se aplica al existir una fábrica para cada controlador y clase del modelo de cada módulo, lo cual permite inyectarle una instancia de cada dependencia al momento de instanciar el objeto.

2.3.5.3. Singleton

El patrón Singleton se aplica en el manejador de servicios de Zend. Este almacena una instancia de cada objeto registrado y reutiliza el mismo objeto cada vez que se solicita uno por lo que no se vuelve a instanciar el objeto.

²¹ <https://olegkrivtsov.github.io/using-zend-framework-3-book/html>. Consulta: julio de 2017. cap. 1.10

Esto se aplica en conjunto con las fábricas para inyectar la misma instancia del objeto existente dentro de cada clase que lo necesita (a menos que se requiera una nueva instancia). De esta manera aumenta el rendimiento del sistema.

2.3.5.4. Domain Driven Design

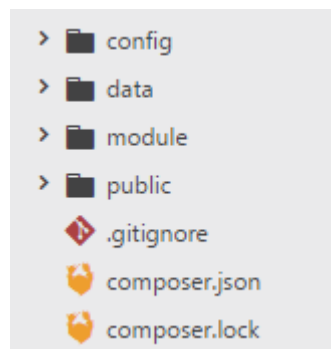
El patrón *Domain Driven Design* es aplicado en la división de clases de la lógica del negocio, dependiendo de su funcionalidad: las entidades, los servicios y los manejadores van agrupados por separado para diferenciar sus roles específicos.

2.3.6. Estructura de la solución

La solución se creó utilizando Zend Framework 3, por lo que la estructura del código está basada en el modelo utilizado por el marco de trabajo.

Dentro del código existen dos proyectos principales: el sistema de vinculación de egresados y el sistema de Cursos Libres, cada uno de ellos posee su propia carpeta proyecto y se estructuran de la siguiente manera (ver figura 4).

Figura 4. Estructura general de un proyecto



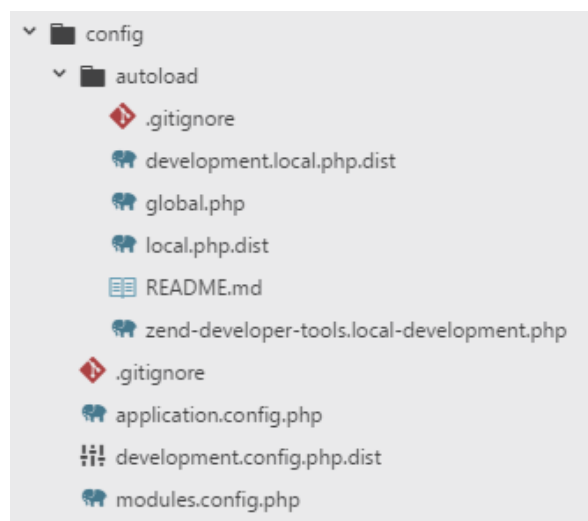
Fuente: elaboración propia.

Existen cuatro carpetas principales: config, que almacena la información de configuración; data, que almacena la información de caché, imágenes y archivos temporales; module, que contiene los módulos que conforman el proyecto y public, la cual, como su nombre lo indica, es pública, por lo que sus archivos están disponibles para el usuario. Adicionalmente, existen archivos propios de Git y Composer:

- El archivo gitignore almacena una lista de archivos y carpetas que deben ser ignorados por el control de versiones (este archivo se presenta múltiples veces dentro del proyecto, pero solamente se mencionará esta vez).
- El archivo composer.json almacena la información de las dependencias del proyecto y datos generales del proyecto en formato JSON.
- El archivo composer.lock es creado automáticamente al momento de instalar las dependencias mediante Composer y guarda las últimas versiones de las dependencias instaladas.

La carpeta config se estructura de la siguiente forma (ver figura 5).

Figura 5. **Estructura de carpeta config**



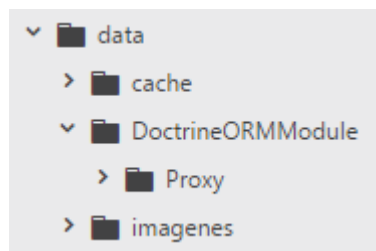
Fuente: elaboración propia.

Dentro de la carpeta config existen tres archivos principales y una carpeta. Estos archivos son: application.config.php, que almacena la configuración general de Zend Framework; development.config.php, el cual almacena configuración adicional de Zend Framework que se aplica cuando el proyecto está en modo de desarrollo y modules.config.php, que incluye una lista de los módulos a cargar para la aplicación. Existen también archivos con extensión. dist los cuales son redistribuibles que se almacenan en el control de versiones como una plantilla de sus homólogos sin la extensión los cuales se ignoran para evitar guardar información irrelevante o confidencial.

Dentro de la carpeta autoload existen dos archivos principales: global.php, el cual almacena la configuración para toda la aplicación y local.php el cual almacena configuración específica para el entorno donde se está ejecutando. Este no se versiona por seguridad, para mantener las credenciales fuera del control de versiones.

La carpeta data se estructura de la siguiente forma (ver figura 6).

Figura 6. **Estructura de carpeta data**



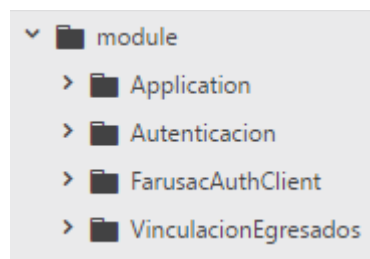
Fuente: elaboración propia.

Dentro de la carpeta data existen tres carpetas: cache, donde se almacena automáticamente la información de caché para optimizar la velocidad de respuesta de la aplicación; DoctrineORMModule/Proxy, donde se almacenan

archivos temporales generados por Doctrine para manipular las entidades e imagenes, donde se almacenan las imágenes que son cargadas a la aplicación.

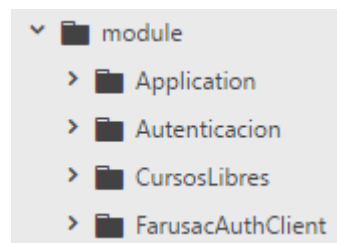
La carpeta module difiere para cada uno de los sistemas y se estructuran de la siguiente forma (ver figuras 7 y 8).

Figura 7. **Estructura carpeta module en sistema de vinculación de egresados**



Fuente: elaboración propia.

Figura 8. **Estructura de carpeta module en sistema de Cursos Libres**

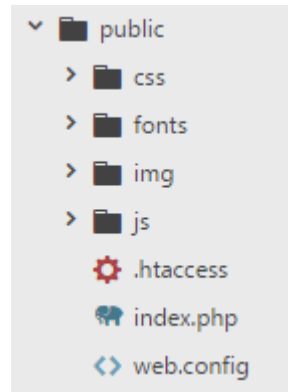


Fuente: elaboración propia.

Dentro de la carpeta module están cada uno de los módulos principales del proyecto. Debido a la extensión y relevancia de cada módulo, se explicarán por separado en la siguiente sección.

La carpeta *public* se estructura de la siguiente forma (ver figura 9).

Figura 9. **Estructura de carpeta *public***



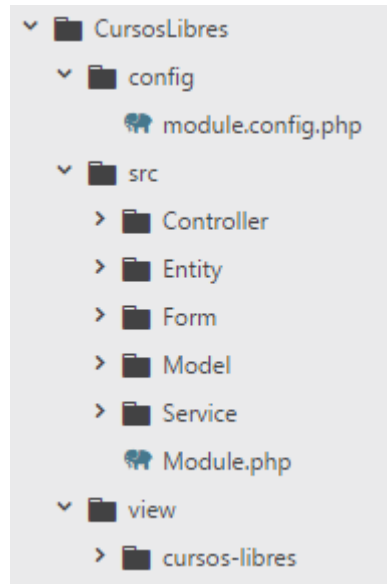
Fuente: elaboración propia.

Dentro de la carpeta *public* existen cuatro carpetas y tres archivos: la carpeta *css*, que incluye los archivos de estilos; la carpeta *fonts*, que incluye las fuentes que utilizará la aplicación; la carpeta *img*, que contiene las imágenes estáticas de la aplicación; la carpeta *js*, que contiene el javascript propio de la aplicación; el archivo *htaccess*, que contiene información de configuración para el directorio para uso de Apache; el archivo *index.php*, que es el único punto de acceso para la aplicación, donde se instancia la aplicación y se cargan las configuraciones; y el archivo *web.config*, que contiene configuraciones adicionales para el servidor.

2.3.6.1. Estructura general de un módulo

Cada módulo que se define a continuación está estructurado de la siguiente forma (ver figura 10).

Figura 10. **Ejemplo de estructura general de un módulo**



Fuente: elaboración propia.

Dentro de cada módulo existen tres carpetas principales: config, la cual contiene al archivo `module.config.php` el cual contiene la configuración para este módulo (que se unirá a la configuración global); src, la cual contiene el código de los controladores y modelos del módulo; y `view/nombre-del-modulo`, la cual contiene las vistas agrupadas en carpetas con el nombre de cada controlador.

Dentro de la carpeta src existen varias carpetas: Controller, la cual almacena los controladores de las vistas; Entity, la cual almacena las entidades utilizadas por Doctrine; Form, la cual almacena las clases de formularios utilizados por Zend-Form; Model, la cual almacena las clases del modelo de la lógica del negocio dentro de la aplicación; Service, que almacena las clases de servicios que provee el módulo; y el archivo `Module.php`, que carga la configuración del módulo.

2.3.6.2. Módulo Application

El módulo Application se crea de forma predeterminada al instaurar un proyecto utilizando Zend Framework. Posee la página de plantilla principal de diseño de todas las vistas de los otros módulos y la página que se muestra si existe un error de la aplicación. Además, se encarga de filtrar la autorización de los usuarios que desean acceder a un controlador.

Dentro del módulo Application existen los siguientes componentes:

- config/module.config.php, El archivo de configuración, aquí se define cuáles serán las páginas de error predeterminadas de la aplicación.
- src/Module.php. La clase de carga del módulo, en ella se dispara una función de validación cada vez que llega una petición a la aplicación, para determinar si el usuario que inició sesión tiene los permisos necesarios para acceder al controlador destino, de lo contrario no le permite acceso.
- view/
 - application/
 - error/
 - ✓ 404.phtml: La vista que se muestra en caso de un error 404 (página no encontrada).
 - ✓ index.phtml: La vista que se muestra en caso de otro tipo de error.
 - layout/
 - ✓ layout.phtml: La vista maestra que define el estilo y diseño de la aplicación, todas las otras vistas se insertan dentro de esta.
 - ✓ navbar.phtml: Una vista parcial que define la barra de navegación.

2.3.6.3. Módulo Autenticacion

El módulo Autenticacion se encarga de registrar usuarios nuevos y autenticar a los ya existentes. El módulo Autenticacion contiene los siguientes componentes:

- config/module.config.php. El archivo de configuración. En este se definen las fábricas del controlador, los archivos del modelo de negocio y los servicios. Se definen, además, las rutas del módulo.
- src/
 - Controller/
 - AuthController.php. El controlador encargado de las acciones de registro, inicio y fin de sesión de la aplicación.
 - Factory/AuthControllerFactory.php. La fábrica para el controlador AuthController.
 - Form/
 - EnviarCorreoForm.php. El formulario para el envío de correos de recuperación de contraseña (Solamente en Vinculación de Egresados).
 - LoginAdminForm.php. El formulario para el inicio de sesión del personal Administrativo.
 - LoginEgresadoForm.php. El formulario para el inicio de sesión del egresado (solamente en vinculación de egresados).
 - LoginEmpresaForm.php. El formulario para el inicio de sesión de las empresas asociadas (solamente en vinculación de egresados).

- ReestablecerContraseniaForm.php. El formulario para reestablecer la contraseña (solamente en vinculación de egresados).
 - RegistroForm.php. El formulario para el registro de Egresados (solamente en vinculación de egresados).
 - Service/
 - AuthAdapter.php. El adaptador utilizado por el servicio de autenticación.
 - AuthManager.php. El manejador del servicio de autenticación.
 - ServicioCorreo.php. El servicio encargado de enviar correos (solamente en vinculación de egresados).
 - ServicioValidacion.php. El servicio encargado de validar la información mediante un web service.
 - Factory/*: La carpeta con las fábricas de los servicios.
 - Module.php: La clase de carga del módulo.
- view/autenticacion/auth/
 - cl-login.phtml. La vista de inicio de sesión para el Administrador del módulo Cursos Libres (solamente en Cursos Libres).
 - login-admin.phtml. La vista de inicio de sesión para el personal administrativo del módulo vinculación egresados (solamente en vinculación de egresados).
 - login-egresado.phtml. La vista de inicio de sesión para el egresado del módulo vinculación egresados (solamente en vinculación de egresados).
 - login-empresa.phtml. La vista de inicio de sesión para las empresas del módulo vinculación egresados (solamente en vinculación de egresados).

- recuperacion-egresado.phtml. La vista de envío de correo electrónico para egresados (solamente en vinculación de egresados).
- recuperacion-empresa.phtml. La vista de envío de correo electrónico para empresas (solamente en vinculación de egresados).
- reestablecer-egresado.phtml. La vista para reestablecer la contraseña (solamente en vinculación de egresados).
- registro.phtml. La vista de registro para egresados del módulo vinculación egresados (solamente en vinculación de egresados).

2.3.6.4. Módulo CursosLibres

El módulo CursosLibres lo contiene el sistema de Cursos Libres. Contiene los siguientes componentes:

- config/module.config.php: El archivo de configuración, en este se definen las fábricas del controlador, los archivos del modelo de negocio y los servicios. Se definen además las rutas del módulo.
- src/
 - Controller/
 - AjaxController.php. Controlador encargado de responder a peticiones AJAX.
 - CursosLibresAdminController.php. Controlador encargado de las acciones del usuario administrador.
 - CursosLibresController.php. Controlador encargado de las acciones de inscripción y asignación de Cursos.
 - ImagenesController.php. Controlador encargado de obtener las imágenes solicitadas.

- Factory/*. Fábricas para los controladores.
 - Entity/*. Las entidades para mapeo de la base de datos
 - Form/
 - AdministracionSitioForm.php. Formulario para administración de recursos del sitio web.
 - CursoLibreForm.php. Formulario para creación y edición de Cursos Libres.
 - InscripcionForm.php. Formulario para la inscripción de estudiantes.
 - ReporteAsignacionesForm.php. Formulario para la generación de reportes de asignaciones.
 - Model/
 - ManejadorCursoLibre.php. Clase encargada de realizar todas las acciones sobre los Cursos Libres.
 - ManejadorUsuario.php. Clase encargada de realizar todas las acciones sobre los usuarios.
 - Factory/*. Fábricas de los manejadores.
 - Service/
 - AnioPeriodoService.php. Servicio encargado de manejar años y periodos para la disponibilidad de Cursos Libres.
 - ImagenesManager.php. Servicio encargado de obtener las imágenes desde los archivos.
 - ReporteService.php. Servicio encargado de crear el reporte en formato xls.
 - Module.php. La clase de carga del módulo.
- view/cursos-libres/
 - cursos-libres/
 - asignacion.phtml. Vista utilizada para la asignación de Cursos Libres.

- asignado.phtml: Vista utilizada para confirmar la asignación.
- cursos-libres.phtml: Vista utilizada para listar los Cursos Libres.
- home.phtml: Vista inicial del sistema, con formulario para iniciar inscripción y vista de los cursos disponibles.
- inscripcion.phtml: Vista utilizada para la inscripción al programa.
- cursos-libres-admin/
 - administracion-sitio. Vista utilizada para editar los recursos.
 - curso-libre.phtml. Vista utilizada para editar o crear un curso.
 - cursos-libres.phtml. Vista utilizada para gestionar los cursos existentes.
 - home.phtml. Vista inicial de las funcionalidades de administración.
 - reportes.phtml. Vista utilizada para la generación de reportes.

2.3.6.5. Módulo vinculación egresados

El módulo vinculación egresados está contenido en el sistema de vinculación de egresados. Contiene los siguientes componentes:

- config/module.config.php. El archivo de configuración. En este se definen las fábricas del controlador, los archivos del modelo de negocio y los servicios. Se definen además las rutas del módulo.
- src/
 - Controller/
 - AdminController.php. Controlador encargado de las acciones del administrador.

- AjaxController.php. Controlador encargado de responder a las peticiones AJAX.
- DifusionController.php. Controlador encargado de las acciones de usuarios administrativos de divulgación.
- EgresadoController.php. Controlador encargado de las acciones de usuarios egresados.
- EmpresaController.php. Controlador encargado de las acciones de usuarios empresariales.
- ImagenesController.php. Controlador encargado de obtener las imágenes solicitadas.
- VinculacionEgresadosController.php. Controlador encargado de las acciones que no requieren registrarse.
- Factory/*. Fábricas para los controladores.
- Entity/*. Las entidades para mapeo de la base de datos.
- Form/
 - CatalogoForm.php. Formulario para búsqueda por catálogo.
 - CrearEventoForm.php. Formulario para creación de eventos académicos.
 - CrearOfertaLaboralForm.php. Formulario para creación de ofertas laborales.
 - EmpresaForm.php. Formulario para creación de usuarios empresariales.
 - InfoEmpresaForm.php. Formulario para edición de detalles de la empresa.
 - MensajeForm.php. Formulario para envío de mensajes a egresados.
 - PerfilEgresadoForm.php. Formulario para edición de perfil de egresado.

- ReporteEgresadosForm.php. Formulario para la generación de reportes.
 - RevisarOfertaLaboralForm.php. Formulario para la revisión de ofertas laborales.
 - Model/
 - ManejadorEmpresa.php. Clase de modelo de negocio encargada del manejo de las empresas.
 - ManejadorEventoAcademico.php. Clase de modelo de negocio encargada del manejo de los eventos académicos.
 - ManejadorOfertaLaboral.php. Clase de modelo de negocio encargada del manejo de las ofertas laborales.
 - ManejadorPerfilEgresado.php. Clase de modelo de negocio encargada del manejo de los perfiles de egresado.
 - Factory/*. Fábricas para las clases de modelo.
 - Service/
 - ImagenesManager.php. Servicio encargado de obtener las imágenes desde los archivos.
 - ReporteService.php. Servicio encargado de la generación de archivos de reporte en formato .xls.
 - Module.php. Clase de carga del módulo.
- view/vinculacion-egresados/
 - admin/
 - catalogo.phtml. Vista utilizada para el catálogo de egresados.
 - crear-empresa.phtml. Vista utilizada para la creación de empresas.
 - empresas.phtml. Vista utilizada para gestionar cuentas de empresas.
 - index.phtml. Vista inicial del rol administrador.
 - perfil.phtml. Vista utilizada para ver perfiles de egresado.

- propuesta.phtml. Vista utilizada para visualizar una propuesta de oferta laboral.
- reportes.phtml. Vista utilizada para la generación de reportes.
- resultados.phtml. Vista utilizada para visualizar los resultados de búsqueda de egresados.
- revisar-propuestas.phtml. Vista utilizada para listar las propuestas existentes.
- js/*: Scripts para las vistas.
- difusion/
 - crear.phtml. Vista utilizada para crear un evento académico.
 - index.phtml. Vista utilizada para visualizar los eventos disponibles.
 - ver.phtml. Vista utilizada para visualizar un evento académico.
 - js/*.Scripts para las vistas.
 - parciales/*. Vistas parciales.
- egresado/
 - catalogo.phtml. Vista utilizada para el catálogo de egresados.
 - edit.phtml. Vista utilizada para editar el perfil de egresado.
 - index.phtml. Vista utilizada para visualizar el perfil propio.
 - mensajes.phtml. Vista utilizada para visualizar mensajes y administrarlos.
 - noticia.phtml. Vista utilizada para visualizar un evento académico.
 - noticias.phtml. Vista utilizada para visualizar todos los eventos académicos.
 - oferta-laboral.phtml. Vista utilizada para visualizar una oferta laboral.

- ofertas-laborales.phtml. Vista utilizada para visualizar todas las ofertas laborales.
- perfil.phtml. Vista utilizada para ver perfiles de egresado.
- resultados.phtml. Vista utilizada para visualizar los resultados de búsqueda de egresados.
- js/*. Scripts para las vistas.
- parciales/*. Vistas parciales.
- empresa/
 - catalogo.phtml. Vista utilizada para buscar entre el catálogo de egresados.
 - crear.phtml. Vista utilizada para crear una propuesta de oferta laboral.
 - editar-info.phtml. Vista utilizada para editar la información de la empresa.
 - index.phtml. Vista utilizada para visualizar las propuestas de oferta laboral existentes.
 - perfil.phtml. Vista utilizada para ver perfiles de egresado.
 - propuesta.phtml. Vista utilizada para visualizar propuestas laborales.
 - resultados.phtml. Vista utilizada para visualizar los resultados de búsqueda de egresados.
 - js/*. Scripts para las vistas.
- vinculacion-egresados/
 - catalogo.phtml. Vista utilizada para buscar entre el catálogo de egresados.
 - enviar-mensaje.phtml. Vista utilizada para el envío de mensajes.
 - index.phtml. Vista inicial del sistema.
 - perfil.phtml. Vista utilizada para ver perfiles de egresado.

- resultados.phtml. Vista utilizada para visualizar los resultados de búsqueda de egresados.
- parciales/*. Vistas parciales.

2.4. Costos del proyecto

El desglose presupuestario del proyecto se presenta a continuación (ver tabla XXVI).

Tabla XXVI. **Desglose presupuestario**

Costos mensuales			
Recursos	Cantidad	Costo unitario	Subtotal mensual
Estudiante de Ciencias y Sistemas con cierre de pensum como Programador	4 horas al día entre semana	Q80,00 por hora	Q6 400,00
Estudiante de Ciencias y Sistemas como Encargado de Sistemas	2 hora al día entre semana	Q80,00 por hora	Q3 200,00
Ingeniero en Ciencias y Sistemas como Encargado de Sistemas	2 horas al día entre semana	Q150,00 por hora	Q6 000,00
Ingenieros en Ciencias y Sistemas como Consultores	3 horas a la semana cada uno	Q200,00 por hora x 2 consultores	Q4 800,00
Espacio de trabajo dentro de la institución	10 horas a la semana	Q10,00 por hora	Q400,00
Internet dentro de la institución	10 horas a la semana	Q0,50 por hora	Q20,00

Continuación de tabla XXVI.

Energía eléctrica dentro de la institución	10 horas a la semana	Q0,30 por hora	Q12,00
Subtotal mensual (1 mes)			Q20 832,00
Subtotal mensual (6 meses)			Q124 992,00
Costos únicos			
Material audiovisual para capacitación de usuarios	1	Q3 000,00	Q3 000,00
Subtotal único			Q3 000,00
Costo total del proyecto			Q127 992,00

Fuente: elaboración propia.

2.5. Beneficios del proyecto

Los beneficios de la plataforma se dividen dentro de sus dos funcionalidades.

2.5.1. Vinculación de egresados

Se recopila una nueva base de datos con información sobre los egresados de la Facultad, con información actualizada. Esta información contiene datos como la especialización de los egresados y puede utilizarse para procesos de minería de datos para descubrir nuevas áreas de interés que en un futuro pueden ser de utilidad en una modificación curricular. Además, se obtienen datos sobre la tasa de empleo y solicitud de los egresados de la Facultad de Arquitectura.

Se obtiene un espacio donde los profesionales pueden exponer sus perfiles y, de esta manera, darse a conocer en el ámbito laboral, proporcionando un impulso a la tasa de empleo de los egresados de la Facultad de Arquitectura.

Se obtiene un medio de transmisión de información entre las empresas que quieren dar a conocer sus oportunidades laborales y los egresados que las buscan.

Se obtienen acuerdos de cooperación y alianzas entre las compañías y la Facultad de Arquitectura que ofrecen mejores oportunidades a los egresados.

Se obtiene un medio de comunicación entre la Facultad y los egresados, mediante el cual puede divulgarse información.

2.5.2. Cursos Libres

Se implementa un proceso de inscripción eficiente ya que los interesados pueden realizar su trámite personalmente utilizando internet.

Se facilita el acceso a la información estadística sobre el proceso y lista de estudiantes asignados.

3. FASE ENSEÑANZA APRENDIZAJE

3.1. Capacitación propuesta

Dentro de la capacitación que se propone hay dos grupos objetivo: el personal técnico de Control Académico y el personal administrativo de la plataforma.

3.1.1. Dirigida al personal técnico

La capacitación al personal de Control Académico debe ser de carácter técnico. En ella, se deben indicar los detalles del funcionamiento del proyecto, ya que dicho personal le dará mantenimiento y realizará cambios al sistema si surgiera el requerimiento. Durante esta capacitación se les hace entrega del manual técnico de la aplicación.

3.1.2. Dirigida al personal administrativo de la plataforma

La capacitación al personal administrativo de la plataforma debe indicar todas las funcionalidades y usos que pueden dársele, ya que este personal se encargará de las funciones diarias del sistema y la resolución de dudas relacionadas con el mismo. Durante esta capacitación se les entrega el manual de usuario dirigido a personal administrativo.

3.2. Material elaborado

El material elaborado se divide en dos categorías: material técnico y material de información de uso.

3.2.1. Técnico

Este material corresponde al manual técnico, documento en el que se indican los detalles técnicos de la plataforma, incluyendo la estructura de archivos del proyecto, las variables de configuraciones locales, las extensiones requeridas, las dependencias del proyecto y una guía para instalación del mismo.

3.2.2. Información de uso

El material de información para los usuarios finales se divide de acuerdo con el rol que tendrán los usuarios dentro de la plataforma.

3.2.2.1. Dirigido al personal administrativo

El material dirigido al personal administrativo incluye el manual de usuario propio para cada sistema (vinculación de egresados y Cursos Libres). En él se indican las funcionalidades, alcances y limitaciones de la plataforma, para que se tenga una guía de referencia respecto a su uso.

3.2.2.2. Dirigido al egresado

El material dirigido a los egresados incluirá videos, que creará el personal de Bienestar Estudiantil de la Facultad de Arquitectura, basados en el manual de

usuario. Estos mostrarán, mediante ejemplos audiovisuales, el uso apropiado del sistema de vinculación de egresados.

3.2.2.3. Dirigido a las empresas

El material dirigido a las empresas incluye videos, que creará el personal de Bienestar Estudiantil de la Facultad de Arquitectura. Se basarán en el manual de usuario, que muestran, mediante ejemplos audiovisuales, el uso apropiado del sistema; y un manual de usuario orientado a empresas. Este documento indica las funcionalidades, alcance y limitaciones del sistema enfocado al usuario de rol empresa que podrá descargarse desde el sistema de vinculación de egresados.

3.2.2.4. Dirigido a los aspirantes a Cursos Libres

El material dirigido a los egresados incluye un documento de ayuda, que podrá ser descargado desde la página de Cursos Libres y muestra el uso apropiado del sistema.

CONCLUSIONES

1. Se implementó una plataforma informática que permite al egresado publicar un perfil en línea, lo que lo ayuda a ubicarse de forma más eficiente dentro del mercado laboral.
2. Se implementó un sistema para realizar la inscripción de los Cursos Libres en línea.
3. Se facilitó la divulgación de información laboral y académica a los egresados de la Facultad de Arquitectura de la Universidad de San Carlos de Guatemala, mediante la publicación de noticias, eventos y ofertas laborales que los egresados pueden ver dentro de la plataforma.
4. Se creó una cartera, mediante la recolección de información de los egresados y la publicación de sus perfiles. Esto permite, a las empresas seleccionadas, la rápida y eficaz ubicación de los profesionales por su área de desempeño.
5. Se agilizó el proceso de inscripción al realizarse en línea y se facilitó la emisión de constancias de participación mediante la creación de listados de inscritos por curso para los Cursos Libres.

RECOMENDACIONES

1. Designar a una persona del personal técnico informático para que se encargue de dar mantenimiento y seguimiento a la plataforma, monitoreando su uso y verificando si existe alguna oportunidad de mejora para la misma.
2. Dar a conocer el sistema de vinculación de egresados de forma extensiva ya que, debido a la naturaleza del mismo, cuantos más egresados y empresas estén inscritos, el sitio estará más completo y variado.
3. Realizar revisiones periódicas del sistema de vinculación de egresados para verificar que los perfiles de usuario no presenten contenido inapropiado y las noticias y propuestas laborales no se acumulen dentro del sistema luego de que pierdan su relevancia.
4. Mantener comunicación y proveer apoyo constante con las empresas que utilicen el Sistema de vinculación de egresados, para mantener una buena relación con las mismas y obtener retroalimentación del funcionamiento del sistema.

BIBLIOGRAFÍA

1. Apache Software Foundation. *The Apache HTTP Server Project. About the Apache HTTP Server Project.* [en línea]. <https://httpd.apache.org/ABOUT_APACHE.html>. [Consulta: noviembre de 2016].
2. CATLIN, Hampton. et al. *Sass: Syntactically Awesome Style Sheets.* [en línea]. <http://sass-lang.com/documentation/file.SASS_REFERENCE.html>. [Consulta: agosto de 2017].
3. Composer. *GetComposer. Getting Started.* [en línea]. <<https://getcomposer.org/doc/00-intro.md>>. [Consulta: marzo de 2017].
4. DeGerencia. *Bolsas de Empleo.* [en línea]. <<http://www.degerencia.com/area.php?areaid=10028>>. [Consulta: marzo de 2017].
5. Doctrine Team. *Doctrine Home.* [en línea]. <<http://www.doctrine-project.org/>>. [Consulta: agosto de 2017].
6. Facultad de Arquitectura, Universidad de San Carlos de Guatemala. *Administración.* [en línea]. <<http://farusac.edu.gt/administracion>>. [Consulta: julio de 2017].

7. HashiCorp. *Why Vagrant?*. [en línea]. <<https://www.vagrantup.com/docs/why-vagrant/>>. [Consulta: marzo de 2017].
8. KRIVTSOV, Oleg. *Using Zend Framework 3*. [en línea]. <<https://olegkrivtsov.github.io/using-zend-framework-3-book/html>>. [Consulta: julio de 2017].
9. MariaDB Foundation. *About MariaDB*. [en línea]. <<https://mariadb.org/about/>>. [Consulta: noviembre de 2016].
10. Mountain Goat Software. *Scrum Methodology and Project Management*. [en línea]. <<https://www.mountaingoatsoftware.com/agile/scrum>>. [Consulta: noviembre de 2016].
11. Mozilla. et al. *Mozilla Development Network Web Docs. Javascript*. [en línea]. <<https://developer.mozilla.org/es/docs/Web/JavaScript>>. [Consulta: agosto de 2017].
12. OTTO, Mark. et al. *Bootstrap*. [en línea]. <<https://getbootstrap.com/>>. [Consulta: agosto de 2017].
13. PANTOJA, Diego. *ENAE Business School. ¿Qué es el Networking?* [en línea]. <<http://www.enaes.es/blog/que-es-el-networking>>. [Consulta: marzo de 2017].
14. PHP Group. *PHP: Hypertext Preprocessor*. [en línea]. <<https://secure.php.net/manual/es/preface.php>>. [Consulta: noviembre de 2016].

15. Software Freedom Conservancy. *Git*. [en línea]. <<https://git-scm.com/>>. [Consulta: marzo de 2017].
16. Software in the Public Interest, Inc. *Debian - Acerca de Debian*. [en línea]. <<https://www.debian.org/intro/about>>. [Consulta: noviembre de 2016].
17. Yarn. *Yarn*. [en línea]. <<https://yarnpkg.com/lang/en/>>. [Consulta: agosto de 2017].
18. Zend, a Rogue Wave Company. *Zend Framework - About*. [en línea]. <<https://framework.zend.com/about>>. [Consulta: noviembre de 2016].
19. ————. *Zend Framework Components*. [en línea]. <<http://zendframework.github.io>>. [Consulta: agosto de 2017].

ANEXOS

Anexo 1. Material publicitario promocional del sistema de vinculación de egresados 1



Fuente: Unidad de Bienestar y Desarrollo Estudiantil, Facultad de Arquitectura, Universidad de San Carlos de Guatemala.

Anexo 2. **Material publicitario promocional del sistema de vinculación de egresados 2**

USAC
TRICENTENARIA
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE
ARQUITECTURA

Te invitamos a conocer la
NUEVA PÁGINA DE
EGRESADOS
ARQUITECTURA / DISEÑO GRÁFICO

Auditorio
Viernes 25 de Agosto • 17:00 horas

Fuente: Unidad de Bienestar y Desarrollo Estudiantil, Facultad de Arquitectura, Universidad de San Carlos de Guatemala.