UCAB Universidad Católica ANDRÉS BELLO

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA INFORMÁTICA

MÓDULO DE DIRECCIÓN PARA LA PLATAFORMA DE EDUCACIÓN ONLINE DEL COLEGIO VIRTUAL DAWERE

TRABAJO INSTRUMENTAL DE GRADO

Presentado ante la

UNIVERSIDAD CATÓLICA ANDRÉS BELLO

Como parte de los requisitos para optar al título de

INGENIERO EN INFORMÁTICA

REALIZADO POR	**Vianny Isabel Molero Pérez**
TUTOR EMPRESARIAL	**Álvaro González**
TUTOR ACADEMICO	**Henry Martínez**
FECHA	**Junio, 2016**

UCAB Universidad Católica ANDRÉS BELLO

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA INFORMÁTICA

MÓDULO DE DIRECCIÓN PARA LA PLATAFORMA DE EDUCACIÓN ONLINE DEL COLEGIO VIRTUAL DAWERE

Este Jurado; una vez realizado el examen del presente trabajo ha evaluado su contenido con el resultado: 16 (dieciseis)

JURADO EXAMINADOR

Firma:	Firma:	Firma:
Nombre: Valeria León	Nombre: BISMARCK PONCE	Nombre: Henry Martínez

REALIZADO POR	**Vianny Isabel Molero Pérez**
TUTOR EMPRESARIAL	**Álvaro González**
TUTOR ACADEMICO	**Henry Martínez**
FECHA	**Junio, 2016**

DEDICATORIA

A mis padres por ser el pilar fundamental en todo lo que soy, en mi educación, tanto académica, como de la vida, por su apoyo incondicional que se mantiene a través del tiempo y espacio.

A mi hermano por darme palabras de aliento cada día y creer en mí para culminar las metas que me propongo, más que un hermano eres mi mejor amigo.

A Suki y Viki por sacarme infinidades de sonrisas al acompañarme en muchas noches de estudio, tendrán siempre un lugar especial en mi corazón.

AGRADECIMIENTOS

Agradezco de manera especial al Profesor Henry Martínez por su amabilidad, motivación, disponibilidad y colaboración para ayudarme a finalizar mi carrera profesional con broche de oro.

Quiero expresar mi más sincero agradecimiento a Alfredo Escalante y Héctor Uzcátegui por su importante aporte y participación activa en el desarrollo de Dawere. No cabe duda que su participación ha enriquecido este emprendimiento, además, ha nacido una amistad sincera en esas noches de programación donde le pusimos corazón.

Quiero agradecer a Álvaro González por darme la oportunidad y confianza de participar en este emprendimiento tan especial, le agradezco también por sus siempre atentas y rápidas respuestas a las inquietudes surgidas durante el desarrollo del proyecto. Ha sido capaz de ganarse mi admiración y respeto por aportar su granito de arena en la construcción de un mundo mejor.

Para todos los miembros del equipo de Dawere: producción, diseño, postproducción, gerentes, programación y personal administrativo, tienen mis más sinceros agradecimientos. Debo agradecer de manera especial a Francisco, Jinneth, Adrián, Roeny, Luis Alberto y Fabiola por su cariño, consejos, ánimo y compañía en los momentos más difíciles de mi vida.

Debo agradecer a aquellos amigos que han compartido conmigo, Andreina Verde de quien siempre he recibido palabras de aliento y su divertidas ocurrencias, Alejandra Silva por creer en mí y en lo que puedo llegar a ser,

Gustavo Hernández sin importar donde estemos siempre estaremos para aconsejarnos.

Y, por supuesto, el agradecimiento más profundo va para mi familia. Sin su apoyo e inspiración habría sido imposible llevar a cabo esta meta. A mis padres, Enrique y Valentina por su ejemplo de lucha y honestidad; a mi hermano Enrique por su tenacidad y valentía; a mis primas Verónica y Valeria por su alegría e inocencia.

Para ellos: Gracias totales.

TABLA DE CONTENIDOS

DEDICATORIA I

AGRADECIMIENTOS II

TABLA DE CONTENIDOS IV

ÍNDICE DE FIGURAS VII

INDICE DE TABLAS VIII

GLOSARIO IX

1.1. ENCUESTA IX
1.2. FACTURA IX
1.3. INSCRIPCIÓN ACADÉMICA IX
1.4. INSCRIPCIÓN ADMINISTRATIVA IX
1.5. NOTA DE CRÉDITO IX
1.6. PROMOCIÓN X

SINOPSIS XI

INTRODUCCIÓN 1

CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA 3

1.1. NECESIDAD DE LA EMPRESA 3
1.2. SOLUCIÓN PROPUESTA 4
1.3. OBJETIVO GENERAL 6
1.4. APORTE 7
1.4.1. Aporte Tecnológico 7
1.4.2. Aporte Funcional 7
1.5. ALCANCE Y LIMITACIONES 7
1.5.1. Definir los procesos administrativos y directivos 7
1.5.2. Submódulo de precios y promociones de cursos 7
1.5.3. Gestión de inscripción 8
1.5.4. Diseñar e implementar submódulo de contabilidad 9
1.5.5. Submódulo de Encuestas 9
1.5.6. Diseñar e implementar submódulo de alianzas estratégicas 9
1.5.7. Diseñar e implementar submódulo de diplomas, certificados y constancias 10
1.5.8. Diseñar e implementar submódulo de comunicados 10
1.5.9. Diseñar e implementar la base de datos 10
1.5.10. Aporte 11

CAPÍTULO II: MARCO TEÓRICO 13

3.1. ADMINISTRACIÓN DE LA EDUCACIÓN 13
3.2. API 13
3.3. MERCADOPAGO 13
3.4. APLICACIÓN WEB 13

3.5. FRAMEWORK 14
3.6. GITHUB 14
3.7. INTELLIJ IDEA 14
3.8. BACKEND 14
3.9. FRONTEND 14
3.10. JAVA 14
3.11. JAVASCRIPT 15
3.12. MAVEN 15
3.13. REST 15
3.14. RESTFUL 15
3.15. SERVLET 16
3.16. JERSEY 16
3.17. JSON 16
3.18. MYSQL 16
3.19. HIBERNATE 16
3.20. JQUERY 17
3.21. HTML 17
3.22. CSS 17
3.23. BOOTSTRAP 17
3.24. ANGULARJS 17

CAPÍTULO III: MARCO METODOLÓGICO 18

CAPÍTULO IV: DESARROLLO 24

4.1. SPRINT 0 24
4.1.1. Objetivo 24
4.1.2. Sprint Backlog 24
4.1.3. Desarrollo 25
4.2. SPRINT 1 30
4.2.1. Objetivo 30
4.2.2. Sprint Backlog 30
4.2.3. Desarrollo 30
4.3. SPRINT 2 32
4.3.1. Objetivo 32
4.3.2. Sprint Backlog 32
4.3.3. Desarrollo 32
4.4. SPRINT 3 34
4.4.1. Objetivo 34
4.4.2. Sprint Backlog 34
4.4.3. Desarrollo 35
4.5. SPRINT 4 39
4.5.1. Objetivo 39
4.5.2. Sprint Backlog 40
4.5.3. Desarrollo 40
4.6. SPRINT 5 40
4.6.1. Objetivo 40
4.6.2. Sprint Backlog 40

4.6.3. Desarrollo *41*
4.7. SPRINT 6 42
4.7.1. Objetivo *42*
4.7.2. Sprint Backlog *42*
4.7.3. Desarrollo *42*
4.8. SPRINT 7 45
4.8.1. Objetivo *45*
4.8.2. Sprint Backlog *45*
4.8.3. Desarrollo *45*
4.10. SPRINT 8 47
4.10.1. Objetivo *47*
4.10.2. Sprint Backlog *47*
4.10.3. Desarrollo *47*
4.11. SPRINT 9 48
4.11.1. Objetivo *48*
4.11.2. Sprint Backlog *48*
4.11.3. Desarrollo *48*

CAPÍTULO V: RESULTADOS 50

CONCLUSIONES Y RECOMENDACIONES 55

REFERENCIAS BIBLIOGRÁFICAS 57

APÉNDICE A – ESTRUCTURAS 59
APÉNDICE C – ILUSTRACIONES 66

ÍNDICE DE FIGURAS

Figura 1. Ciclo Scrum. Fuente: [10] 22
Figura 2. Arquitectura de Software. 29
Figura 3. Arquitectura MercadoPago. 37

INDICE DE TABLAS

TABLE 1. OBJETIVOS EN SPRINTS. 23
TABLE 2. EVALUACIÓN DOCUMENTACIÓN Y APRENDIZAJE. 26
TABLE 3. EVALUACIÓN MANEJADOR DE DATA. 26
TABLE 4. EVALUACIÓN PRUEBAS. 27
TABLE 5. EVALUACIÓN DE POPULARIDAD. 28
TABLE 6. EVALUACIÓN DE MANEJADOR DE BASE DE DATOS. 29

GLOSARIO

1.1. Encuesta

La encuesta es un instrumento para recopilar datos por medio de un cuestionario. Sirve para medir la calidad de la aplicación y recibir feedback de los usuarios de la plataforma.

1.2. Factura

Una factura es una constancia de una inscripción administrativa que sirve como comprobante de la realización de la compra, posee información referente a la misma.

1.3. Inscripción académica

La inscripción académica es la acción de un estudiante para registrarse en una materia con la finalidad de interactuar con los elementos que posee la misma.

1.4. Inscripción administrativa

La inscripción administrativa es la realización de un pago por concepto de su inscripción académica para poder tener más beneficios al ver una materia.

1.5. Nota de crédito

Una nota de crédito es la anulación total de la factura, es una devolución de la compra realizada. Es un comprobante para el cliente.

1.6. Promoción

Consiste en implementar un descuento del 100% en una materia mediante códigos.

SINOPSIS

El presente Trabajo Instrumental de Grado titulado "Módulo de Dirección para la Plataforma de Educación Online del Colegio Virtual Dawere", se construyó una aplicación web en Maven y AngularJS que permite administrar inscripciones y retiros de los estudiantes, proveer los precios de los productos, generar facturas, como también generar promociones, alianzas, certificados, constancias, comunicados y encuestas, estas funcionalidades son parte de los requerimientos del emprendimiento educacional Dawere.

El Trabajo Instrumental de Grado durante su construcción se aplicó la metodología ágil Scrum, este se basa en el desarrollo iterativo e incremental. Para la elaboración de la plataforma se realizaron ocho iteraciones, cada iteración pasa por un proceso de análisis, diseño, desarrollo, pruebas, mejoras e inclusión a la versión master del proyecto.

Se elaboró un Servidor API Restful haciendo uso del Framework Maven que utiliza como lenguaje de programación Java, el Cliente fue construido aplicando el Framework AngularJS que implementa JavaScript como lenguaje de programación, se utilizó MySQL como manejador de base de datos, GitHub como manejador de versiones e IntelliJ IDEA como ambiente de desarrollo integrado.

El proyecto cumplió exitosamente los objetivos propuestos al obtener una plataforma web que administre la dirección del futuro colegio virtual llamado Dawere.

INTRODUCCIÓN

En el presente Trabajo Instrumental de Grado, se explican las actividades que se llevaron a cabo para la elaboración del Módulo de Dirección para la Plataforma de Educación Online del Colegio Virtual Dawere, expuesto como solución para fomentar el sistema educacional en la República Bolivariana de Venezuela.

La estructura del Trabajo Instrumental de Grado se encuentra organizado en seis capítulos, su contenido es detallado a continuación:

- **Capítulo I:** Planteamiento del Problema

Se describe la situación actual del sistema educacional en Venezuela, las necesidades del emprendimiento de la empresa, solución al planteamiento del problema, objetivos generales y específicos, alcance y limitaciones del presente Trabajo Instrumental de Grado.

- **Capítulo II:** Marco Teórico

Se describe los conceptos teóricos básicos relacionados con la administración educativa y herramientas utilizadas en el presente Trabajo Instrumental de Grado.

- **Capítulo III:** Marco Metodológico

Se justifica y define la metodología implementada, describiendo cada elemento del ciclo de vida.

- **Capítulo IV:** Desarrollo

Se explica los detalles de las funcionalidades construidas, siguiendo las pautas de la metodología ágil aplicada.

- **Capítulo V:** Resultados

Se presenta el producto final obtenido, cumpliendo las metas propuestas en los objetivos específicos.

- **Capítulo VI:** Conclusiones y Recomendaciones

Se describen las conclusiones obtenidas y contribución de sugerencias para la continuación del presente Trabajo Instrumental de Grado.

CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA

1.1. Necesidad de la empresa

La empresa ***Proyectos Gonmar IPC, C.A.*** se encuentra en el proceso de creación y desarrollo de una plataforma web de educación media del llamado proyecto.

En los últimos años, uno de los problemas con las instituciones de educación media en Venezuela es la falta de profesores. Como consecuencia, los alumnos no reciben los conocimientos necesarios para su desarrollo académico. Sin embargo, los liceos promedian su nota con las materias que sí lograron impartir y los avanzan al siguiente año escolar.

Dawere utiliza como referencia la Encuesta Nacional de Juventud 2013 del Instituto de Investigaciones Económicas y Sociales (IIES) de la Universidad Católica Andrés Bello (UCAB), en el cual se aplicó una encuesta a 4022 jóvenes con edades comprendidas entre 15 y 29 años. Los resultados más destacados y de nuestro interés es que el 42% de la población acude a un centro de enseñanza, 1 de cada 3 adolescentes (entre 15 y 19 años) se encuentra desescolarizado, el 60% de los jóvenes han completado la educación media. Estos datos estadísticos son preocupantes por el nivel de desempeño de los estudiantes y las deficiencias estructurales del sistema escolar. Es deseable emprender y apoyar proyectos bien diseñados y factibles que contribuyan a resolver estos problemas.

La visión del proyecto Dawere es ofrecerle a los colegios de Venezuela una herramienta que estará estructurada para impartir clases a través de videos y actividades evaluativas online, con la finalidad de que el alumno adquiera conocimientos de los temas que se imparten en las materias de educación media.

La plataforma de educación online Dawere carece de una estructura de coordinación y directiva, se propone trasladar las operaciones que maneja la dirección de un colegio a un colegio virtual. Dawere requiere de la reducción de los procesos de inscripción donde el alumno deberá inscribir y retirar sus materias, el director necesita manejar la apertura y cierre de los cursos y materias, como también la estructuración del plan de estudios de las materias que se imparten en la educación medial.

El profesor proveerá un material académico (lecturas, libros, ejercicios, actividades, etc) que deberán cargarse en el portal. Se debe manejar un calendario de actividades para el manejo de un cronograma anual, como también anuncios generales, por materias y por cursos. Se requiere el manejo de las contrataciones del personal, el sistema debe manejar reclamos por parte de los usuarios de la plataforma.

1.2. Solución propuesta

Para el problema en cuestión se le propone a la empresa ***Proyectos Gonmar IPC, C.A.*** el diseño y desarrollo de un módulo de dirección para el funcionamiento interno de la plataforma web del futuro colegio online Dawere, y

que cumpla con los requerimientos expuestos. Cabe acotar que el módulo de dirección estará enfocado en la interacción del director y el staff al portal web.

Para el desarrollo del proyecto, es necesario determinar los requerimientos funcionales de los submódulos de contabilidad, inscripción del estudiante, gestión del año académico, encuesta sobre el curso y la plataforma, generar certificados y diplomas, proveer información a los usuarios y hacer reportes.

Se empleará una conexión web services, para la interacción entre cliente-servidor. En la fase de desarrollo se utilizará un certificado autofirmado para el manejo de la seguridad de la plataforma web.

Finalmente, el diseño y desarrollo del proyecto será realizado bajo la metodología de prototipos que se divide en fases y cada fase tiene una o más iteraciones con entregables.

1.3. Objetivo general

Diseñar e implementar un módulo de dirección que permita gestionar el funcionamiento del futuro colegio online Dawere, para los estudiantes de educación media de la República Bolivariana de Venezuela.

1.3.1. Objetivos específicos

- Definir los procesos administrativos y directivos.
- Diseñar e implementar submódulo de precios y promociones de cursos.
- Diseñar e implementar submódulo de inscripción y retiro del estudiante.
- Diseñar e implementar submódulo de contabilidad.
- Diseñar e implementar submódulo de encuestas.
- Diseñar e implementar submódulo de alianzas estratégicas.
- Diseñar e implementar submódulo de diplomas, certificados y constancias.
- Diseñar e implementar submódulo de comunicados.
- Diseñar e implementar la base de datos.

1.4. Aporte

1.4.1. Aporte Tecnológico

- Evaluar los distintos Frameworks para el desarrollo web y seleccionar uno apropiado para la implementación del sistema directivo y la aplicación web.
- Evaluar el manejador de base de datos que mejor se adapte a las necesidades de la empresa.
- Evaluar e implementar la arquitectura de software adecuado para el correcto funcionamiento del sistema a desarrollar.

1.4.2. Aporte Funcional

Definir los procesos administrativos, directivos, planes de estudios y materiales académicos para la plataforma online del futuro colegio Dawere.

1.5. Alcance y Limitaciones

1.5.1. Definir los procesos administrativos y directivos

Para el funcionamiento interno del colegio, no tiene definido los procesos administrativos y directivos. Se realizará un mapa de procesos para su definición.

1.5.2. Submódulo de precios y promociones de cursos

1.5.2.1. Diseñar e implementar una sección de precios

Implementar un manejador para cargar y enlazar los precios de

los cursos de un período académico.

1.5.2.2. Diseñar e implementar una sección de promociones

Las promociones son descuentos para inscribirse en un curso, será manejado con códigos. El sistema debe generar códigos individuales o por lotes de promociones, estos se deben ser enviados por correo. Los códigos de promociones pueden o no estar enlazados a instituciones que tengan alianza con Dawere.

1.5.3. Gestión de inscripción

1.5.3.1. Diseñar e implementar submódulo de inscripción administrativa y académica

Permitir la inscripción del estudiante vía online. La plataforma provee al estudiante de manera amigable la información de los cursos disponibles para realizar la inscripción. Al seleccionar el curso, el sistema automáticamente genera el monto a cancelar.

1.5.3.2. Diseñar e implementar submódulo de pago

Es necesario realizar el pago con tarjeta de crédito, se debe establecer una conexión con el banco y proceder a realizar el pago.

1.5.3.3. Diseñar e implementar submódulo de retiro

Es necesario realizar el manejo de un retiro de un estudiante. Se podrá generar un comprobante de retiro. En el caso de que se retire en un período de tiempo que esté dentro de la fecha de devolución parcial de la inscripción se enviará una notificación al director y generar una

nota de crédito.

1.5.4. Diseñar e implementar submódulo de contabilidad

1.5.4.1. Diseñar e implementar submódulo de factura

Al realizar el pago, el sistema genera una factura automáticamente, este será la constancia de inscripción. El estudiante antes de acceder al aula virtual del curso debe realizar el pago.

1.5.4.2. Diseñar e implementar submódulo de notas de crédito

Si se realiza una devolución se debe generar una nota de crédito para anular la factura y crear una nueva. Reportes varios.

1.5.5. Submódulo de Encuestas

1.5.5.1. Diseñar y desarrollar la encuesta

Se creará un manejador de encuestas para el envío del mismo a los usuarios de Dawere.

1.5.5.2. Diseñar y desarrollar estadísticas de la encuesta

Las encuestas se utilizarán para evaluar la calidad del curso y la plataforma con la finalidad de realizar estadísticas y analizar los datos para mejoras futuras de Dawere.

1.5.6. Diseñar e implementar submódulo de alianzas estratégicas

La alianza estratégica es la participación de una institución académica o financiera en Dawere. Deben estar incluidos en la base de datos para generar reportes de sus aportes.

1.5.7. Diseñar e implementar submódulo de diplomas, certificados y constancias

Al finalizar el curso, el sistema generará un diploma, certificado o constancia para cada estudiante que culminó de manera satisfactoria. El alumno podrá exportar el documento emitido por Dawere. El sistema podrá verificar si el documento es válido o no.

1.5.8. Diseñar e implementar submódulo de comunicados

1.5.8.1. Diseñar y desarrollar una sección de reclamos y observaciones por parte de los usuarios

El sistema se encargará de gestionar y notificar al director y asistentes de los reclamos y observaciones de los representantes, alumnos y profesores. Este se divide en reclamos administrativos (promociones y/o descuentos), acceso al aula virtual, acceso a su sesión, modificaciones de inscripciones, otros.

1.5.8.2. Diseñar y desarrollar notificaciones de información a los correos de los usuarios del sistema

El sistema notificará a través de correo electrónico información de interés.

1.5.9. Diseñar e implementar la base de datos

Para el diseño se utilizará un modelo entidad relación para la visualización y documentación de la base de datos.

1.5.10. Aporte

1.5.10.1. Aporte tecnológico

1.5.10.1.1. Evaluar entre los distintos Frameworks para el desarrollo web y seleccionar uno apropiado para la implementación del sistema directivo y la plataforma

Es necesario realizar una evaluación para la selección de un framework MVC apropiado, para el análisis se eligió previamente CakePHP, Ruby on Rails y Java.

1.5.10.1.2. Evaluar e implementar la arquitectura de software adecuado para el correcto funcionamiento del sistema a desarrollar

Se procederá a evaluar la disponibilidad, seguridad, escalabilidad del proyecto e implementar la tecnología que mejor se adapte y brinde mejor desempeño. Para la evaluación se eligió los manejadores MySQL, Postgresql, SQL Server Light.

Actualmente en la empresa no está definida una arquitectura de software para la producción del sistema, es necesario evaluar entre distintas arquitecturas e implementar la arquitectura que beneficie al sistema y al proyecto.

1.5.10.2. Aporte funcional

1.5.10.2.1. Definir los procesos administrativos, directivos, planes de estudios y materiales académicos para la plataforma online del futuro colegio Dawere.

Actualmente en la empresa no están definidos los procesos para el manejo interno del sistema, es necesario definir los procesos de períodos académicos, inscripciones, retiros, administración y contabilidad, e implementa los procesos que se adapten al desarrollo del sistema.

CAPÍTULO II: MARCO TEÓRICO

3.1. Administración de la educación

La administración de la educación es el proceso de planear, organizar, dirigir, ejecutar y controlar las actividades que se desarrollan en las organizaciones educativas. Básicamente, busca organizar y coordinar los distintos recursos con el fin de lograr los objetivos y metas trazadas por la institución.

3.2. API

Una API (Application Programming Interface) es un conjunto de funciones y procedimientos que cumplen una o muchas funciones con el fin de ser utilizadas por otro software.

3.3. MercadoPago

Es una interfaz de programación de aplicaciones que permite manejar las transacciones de pagos entre los clientes de una empresa online y el banco, siendo MercadoPago el intermediario entre estos 3 participantes.

3.4. Aplicación web

En la Ingeniería de Software se denomina aplicación web a aquellas herramientas que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador.

3.5. Framework

Un Framework es un entorno o ambiente de trabajo para el desarrollo; dependiendo del lenguaje normalmente integra componentes que facilitan el desarrollo de aplicaciones como el soporte de programas, bibliotecas y plantillas.

3.6. GitHub

GitHub es una plataforma para alojar proyectos en la nube utilizando el sistema de control de versiones Git.

3.7. IntelliJ IDEA

IntelliJ IDEA es un ambiente de desarrollo integrado para el desarrollo de programas informáticos.

3.8. Backend

Backend es el lado del Servidor, es la parte del desarrollo web que implementa lógica para recibir y enviar información al cliente. Posee una conexión a base de datos.

3.9. Frontend

Frontend es el lado del Cliente, es la parte del desarrollo web que implementa el diseño de un sitio web, desde su estructura, estilos, animaciones y también implementa lógica.

3.10. Java

Java es un lenguaje de programación orientado a objetos de propósito general, aplicable a cualquier proyecto de programación.

3.11. JavaScript

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas [6].

3.12. Maven

Maven es una herramienta que automatiza el proceso de construcción de un proyecto en Java. Provee un conjunto de estándares de construcción, un modelo de repositorio de artefactos y un motor de software que administra y describe proyectos.

3.13. REST

REST define la interacción entre distintos componentes, las reglas que dichos componentes tienen que seguir como: una arquitectura cliente-servidor, no almacenar datos del cliente, se define la manera de recibir peticiones del cliente, sistema por capas [7].

3.14. RESTful

RESTful hace referencia a un servicio web que implementa la arquitectura REST. Para acceder a un recurso del Servidor (en este proyecto tendrá como nombre endpoint) previamente se define la ruta de la petición del Cliente., el tipo de representación de dicho recurso, operaciones soportadas [7].

3.15. Servlet

El Servlet es un módulo escrito en Java que son utilizados en el lado del Servidor para extender sus capacidades de respuesta a los Clientes al utilizas las potencialidades de Java [8].

3.16. Jersey

Jersey es un framework para Java proporcionada por JAX-RS y permite el manejo de peticiones webservices al poseer anotaciones para producir y consumir elementos REST [3].

3.17. JSON

JSON (JavaScript Object Notation – Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos donde es independiente del lenguaje.

3.18. MySQL

MySQL es un sistema de administración de base de datos de código abierto para base de datos relacionales.

3.19. Hibernate

Hibernate es una herramienta de Mapeo objeto-relacional (ORM) para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) [9].

3.20. jQuery

jQuery es un framework de JavaScript, se usa para facilitar el acceso a los elementos del DOM.

3.21. HTML

HTML es el lenguaje con el que se define el contenido de las páginas web. Posee un conjunto de etiquetas que sirven para definir el texto y otros elementos que compondrán una página web.

3.22. CSS

CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML.

3.23. Bootstrap

Boostrap es un framework de CSS, posee un conjunto de archivos de CSS para incluirlo en la página.

3.24. AngularJS

AngularJS es un framework escrito en JavaScript que propone una manera de organizar el frontend de una aplicación web.

CAPÍTULO III: MARCO METODOLÓGICO

Dawere al ser un proyecto innovador en Venezuela, como todo emprendimiento, el cliente tenía una visión medianamente clara de las funcionalidades del producto y al poseer un equipo pequeño de ingenieros y técnicos y la necesidad de ver resultados tangibles a corto plazo, como también la necesidad de permitir cambios de requerimientos y en el desarrollo incluir modificaciones se optó por implementar la metodología ágil Scrum.

Scrum

Scrum es un proceso de la metodología ágil que se usa para minimizar los riesgos del desarrollo de software, a través de las definiciones de los requerimientos y su seguimiento durante la elaboración del producto, con la finalidad de elevar al máximo la productividad del equipo y obtener el mejor resultado posible que sea satisfactorio para el cliente. [1]

Los elementos de la metodología Scrum son los siguientes:

- **Roles**

  Para el desarrollo del Servidor y el Cliente de la plataforma web se formó un equipo de 5 personas. Está constituido por el Product Owner, un Scrum Master y tres desarrolladores. La descripción de cada rol es la siguiente:

- **Product Owner (Dueño del producto)**

  Representa a todos los interesados del producto final. Es el responsable de identificar los requisitos del sistema y poniendo las prioridades de las mismas, es el responsable de la gestión, control, retroalimentación y culminación de la lista de funcionalidades del producto.

- **Scrum Master (Líder del proyecto)**

  Lidera las reuniones, solventa obstaculos que se le presenten al equipo de desarrollo. Se asegura que el proyecto siga las reglas planteadas en el Scrum y progrese según lo previsto.

- **Scrum team (Equipo de desarrollo)**

  Son los encargados de construir el producto. Tienen autoridad para definir las acciones necesarias para el desarrollo de las funcionalidades, aportan soluciones que puedan complementarse con los miembros del equipo.

- **Sprint Planning Meeting (Reunión de Planificación del Sprint)**

La Reunión de Planificación del Sprint se realiza al comienzo de cada Sprint y es conducida por el Scrum Master a la que asisten el propietario del producto y el equipo de desarrollo. Si la duración del Sprint es de 1 a 3 semanas la reunión de planificación debería tener una duración de 2 a 4 horas. Se definen qué se entregará al finalizar el Sprint, cuál trabajo será llevado a cabo y cómo se implementará.

- **Product Backlog (Pila del producto)**

El product backlog es una lista de requerimientos del sistema desde el punto de vista del cliente, es una manera de registrar y organizar funcionalidades del producto de una manera dinámica, constantemente se incluyen necesidades del sistema. El product owner provee información de los requerimientos del producto al scrum master y este se encarga de transformarla en items que posteriormente serán incluidos en la pila, ambos roles llegan a un acuerdo de cuáles son las funcionalidades prioritarias a construir. El product backlog es responsabilidad del product owner.

- **Sprint Backlog (Pila del Sprint)**

El sprint backlog es un subconjunto de items del product backlog, donde se descompone las funcionalidades presentadas con información estrictamente necesaria en un tamaño adecuado para monitorear el avance diario del sprint. Cada item definido posee el tiempo o esfuerzo que se prevee para culminarla, el estatus actual del item y se le asigna un responsable del equipo de programación que se encargará de construirlo y finalizarlo. Es importante la comunicación diaria y directa del equipo.

- **Scrum diario (Daily Scrum Meeting)**

Es una reunión diaria breve que no debe exceder los 15 minutos, donde participa el Scrum Master y todo el equipo de desarrollo. Cada integrante del equipo responde individualmente el avance que tuvo el día anterior, en qué

trabajará ese día y qué ayuda necesita para identificar los problemas u obstáculos que se le presenten. Se busca actualizar la lista del Sprint para verificar cuales tareas están en proceso de desarrollo, cuales fueron culminadas y las que falten por desarrollar del Sprint Backlog.

- **Sprint Review (Revisión del Sprint)**

Al finalizar el Sprint Backlog se realiza una reunión con todos los participantes donde se comprueba la iteración culminada. El propietario del producto revisa y prueba la iteración, este le da feedback relevante del Sprint al equipo de desarrollo donde puedan generar mejoras a la visión del producto.

- **Sprint Retrospective (Retrospectiva del Sprint)**

Scrum busca una mejora continua al finalizar cada iteración. Se identifican los puntos buenos del último Sprint, se puntualiza y priorizan las funcionalidades que se deban mejorar. Asisten el Scrum Master y el equipo de desarrollo.

Véase Figura 1, posee una descripción gráfica de los procesos que se manejan dentro de la metología Scrum, incluyendo a los participantes que la constituyen.

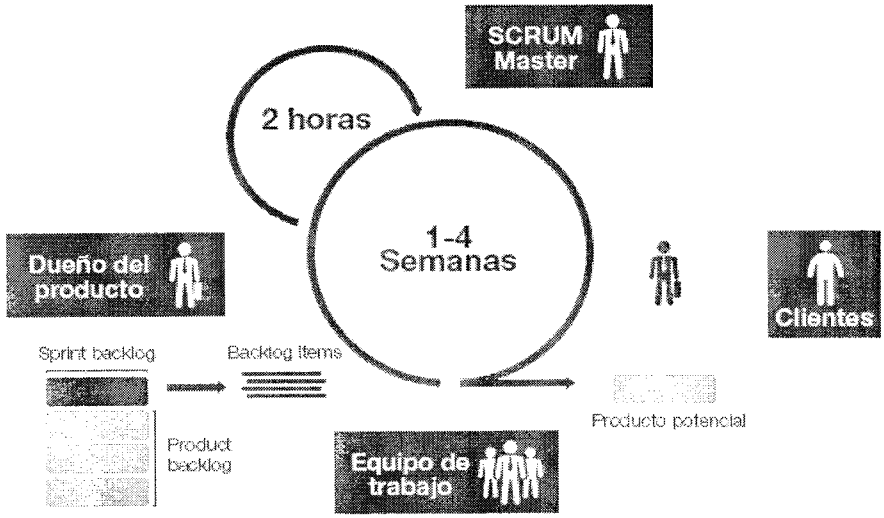

Figura 1. Ciclo Scrum. Fuente: [10]

Los objetivos específicos se divieron en sprints de la siguiente manera:

Objetivos	Sprint
Aporte tecnológico	#0
Diseñar e implementar la base de datos	#1
Diseñar e implementar submódulo de inscripción y retiro del estudiante	#2
Diseñar e implementar submódulo de precios y promociones de cursos	#3
Diseñar e implementar submódulo de alianzas estratégicas	#4
Diseñar e implementar submódulo de contabilidad	#5
Diseñar e implementar submódulo de encuestas	#6
Diseñar e implementar submódulo de comunicados	#7

Diseñar e implementar submódulo de diplomas, certificados y constancias	#8
Diseñar los procesos administrativos y directivos	#9

Table 1. Objetivos en Sprints.

Fuente: Elaboración propia (2016).

CAPÍTULO IV: DESARROLLO

En este capítulo se describe el proceso de diseño e implementación a la solución propuesta en el Trabajo Instrumental de Grado. Para desglosar dicho proceso se aplicó la metodología Scrum mencionado en el marco metodológico, esta metodología se divide en Sprints, la duración de cada Sprint depende de lo extenso que sea un submódulo del producto, obteniendo en cada una nuevas funcionalidades operativas y posteriormente incluyendo mejoras al mismo.

4.1. Sprint 0

4.1.1. Objetivo

Realizar el levantamiento de información, evaluar el ambiente de trabajo, su arquitectura y el manejador de base de datos para el emprendimiento educacional propuesto en este Trabajo Instrumental de Grado.

4.1.2. Sprint Backlog

- Levantamiento de información.
- Evaluar entre los distintos frameworks.
- Evaluar el manejador de base de datos que mejor se adapte al proyecto.
- Evaluar y diseñar la arquitectura del sistema.

4.1.3. Desarrollo

- Se pautó múltiple reuniones para el levantamiento de información donde participaron el dueño del producto, el scrum master y el equipo de desarrollo. Se definió entonces el alcance de la plataforma web Dawere luego de entender y analizar la misión y visión de dicho emprendimiento.
- Se evaluó entre los frameworks CakePHP y Ruby on Rails para la implementación de un Servidor API Restful. Posteriormente se incluyó el lenguaje Java por las múltiples herramientas que posee [12].
    - Documentación y aprendizaje

      CakePHP tiene una documentación bien desarrollada. Ofrece una API con un manual completo donde los desarrolladores pueden compartir código.

      Ruby on Rails ofrece una API como documentación, guías y libros. Tiene una gran comunidad y una wiki.

      Java Oracle provee una documentación amplia de sus herramientas para el desarrollo.

	CAKEPHP	ROR	JAVA
API	Bien	Bien	Bien
BLOGS	Poco	Bastante	Mucho
COMUNIDAD	Pequeña	Bastante	Bien

Table 2. Evaluación documentación y aprendizaje.

Fuente: Elaboración propia (2016).

- Manejador de data

  CakePHP tiene un buen ORM (object relational mapping) framework. No es tan avanzado como RoR o Hibernate.

  Ruby on Rails tiene un mecanismo de migración que permite crear y evolucionar los modelos y el esquema sin destruir la data.

  Java el framework Hibernate facilita el mapeo de atributos entre una base de datos relacional y el modelo de objetos del servidor.

	CAKEPHP	ROR	JAVA
MANEJADOR DE DATA	Bien, es inseguro	Poderoso	Poderoso
ORM	Sí	Sí	Sí

Table 3. Evaluación manejador de data.

Fuente: Elaboración propia (2016).

- Pruebas

  CakePHP el desarrollador debe generar las configuraciones de pruebas.

  Ruby on Rails incluye un framework de prueba, cada vez que se genera un nuevo modelo y controlador, Rails lo incluye en el subdirectorio de pruebas. Crea una base de datos sólo para pruebas.

  Java posee múltiples herramientas para realizar pruebas sobre el servidor.

	CAKEPHP	ROR	JAVA
HERRAMIENTAS DE PRUEBAS	Pocos	Varios	Varios
FRAMEWORK DE PRUEBA	Sí	Sí	Sí

Table 4. Evaluación pruebas.

Fuente: Elaboración propia (2016).

- Escalabilidad

  RoR tiene problemas al tener miles de usuarios en el sistema, tomando en cuenta que Twitter migró a Scala. Empesas como Ebay y Amazon están hechos en Java.

- Popularidad

  TIOBE es una empresa especializada en la evaluación y seguimiento de la calidad de softwares. El rating es basado en el número de ingenieros calificados en todo el mundo, cursos y vendedores a terceros. En el 2015 el rating fue el siguiente:

	PHP	RUBY	JAVA
POPULARIDAD DE LENGUAJE	#6	#10	#1

Table 5. Evaluación de popularidad.

Fuente: Elaboración propia (2016).

- Para evaluar los manejadores de base de datos MySQL, SQL Server y PostgreSQL se investigó sobre los siguientes parámetros.

	SQL SERVER	MYSQL	POSTGRESQL
MODELO DE BASE DE DATOS	Relacional DBMS	Relacional DBMS	Relacional DBMS
LICENCIA	Comercial	Código abierto	Código abierto
TRIGGERS	Sí	Sí	Sí
PARTICIONAMIENTO	Particionamiento horizontal	Particionamiento horizontal	No

REPLICACIÓN	Sí, depende de la versión	MaestroMaestro MaestroEsclavo	MaestroEsclavo

Table 6. Evaluación de manejador de base de datos.

Fuente: Elaboración propia (2016).

El manejador de base de datos que le conviene al proyecto es MySQL por ser de código abierto y por el particionamiento horizontal.

- Para facilitar importaciones y manejo de tareas de Java se utilizó el Framework Maven. el lado del Cliente sería desarrollado en AngularJS. Se instaló las herramientas necesarias para el ambiente de desarrollo, como: el Sistema Operativo, IntelliJ IDEA, MySQL, Java, GitHub, Postman, Tomcat. (Véase Figura 2).

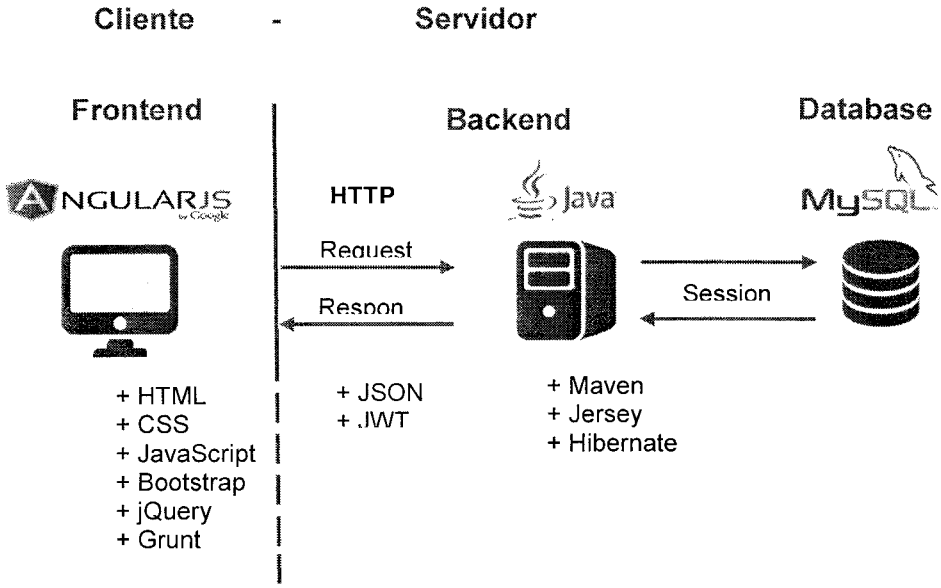

Figura 2. Arquitectura de Software.

Fuente: Elaboración propia (2016).

4.2. Sprint 1

4.2.1. Objetivo

Implementar las conexiones de los múltiples frameworks utilizados en el proyecto.

4.2.2. Sprint Backlog

- Crear un nuevo proyecto en Maven para el Servidor API.
- Utilizar el Framework Jersey para la implementación Restful del Servidor.
- Establecer conexión del Servidor API con MySQL mediante el Framework Hibernate.
- Generar y desarrollar métodos genéricos en backend que permitan crear, obtener, guardar y eliminar data en la base de datos con clases tipo T.

4.2.3. Desarrollo

- IntelliJ IDEA posee la opción de crear un proyecto Maven [2], donde genera un nuevo módulo con un directorio base y archivos estándar para una aplicación web. (Véase Apéndice A-1). La carpeta *main* posee el código de las funcionalidades implementadas del proyecto, la carpeta *test* posee las pruebas de los métodos creados en la carpeta main, el archivo *pom.xml* es el manejador de dependencias

(librerías) donde se declaran las librerías y son descargadas automáticamente.

- Para la implementación de Jersey se utilizó la dependencia *jersey.server* [4]. Se creó una carpeta *Resources* en la carpeta java de backend, esta carpeta representa los endpoints del Servidor, es decir, todas las peticiones que realiza el Cliente llegarán a esta ruta y el recurso (clase) se encargará de producir o consumir las mismas. Jersey posee anotaciones que serán declaradas con una arroba antes de la implementación del método como la ruta, si produce o consume un JSON o el tipo de petición webservices (post, put, delete, get).
- En la estructura del proyecto existe una carpeta "src > resources" donde se implementan las configuraciones del Servidor. Para configurar el **Framework Hibernate** se debe crear un archivo xml con las propiedades de la conexión entre Maven y MySQL, una de las propiedades es declarar el nombre y contraseña de la base de datos, el pool de conexiones y las entidades que posee.
- Se implementó una interfaz donde están declarados los métodos globales que una conexión a base de datos posee, como obtener un elemento, obtener una lista, eliminar, guardar y modificar. Estos métodos reciben parámetros de tipo "T", es decir, pueden recibir todo tipo de objeto que sea un modelo y se encarga de realizar el mapeo con las entidades de base de datos.

4.3. Sprint 2

4.3.1. Objetivo

Diseñar e implementar una inscripción y retiro del estudiante.

4.3.2. Sprint Backlog

- Realizar la inscripción académica de un estudiante en una materia desde su usuario.
- Generar inscripción como complemento académico.
- Cambiar la inscripción del estudiante a "Reforzamiento Premium" cuando se compra una materia con éxito.
- Incluir en el dashboard de estudiante las materias que tiene inscrita por tipo de inscripción.
- Elaborar retiro del estudiante.

4.3.3. Desarrollo

- Se creó una tabla "*Enrolments*" en base de datos donde se definió la relación entre de Materia – Usuario – Tipo de inscripción. El usuario hace la petición de inscripción mediante el botón de inscribir. (Véase Apéndice C-1). En backend se desarrolló el endpoint que maneja la lógica de validar si el estudiante puede ser inscrito en la materia seleccionada, si es válido se guarda en base de datos la inscripción del usuario, la materia y el tipo de inscripción inicialmente es un reforzamiento básico.

- Para describir una inscripción por código de promoción se utilizó la definición de "Complemento Virtual" donde es un nuevo tipo de inscripción y es guardada en la tabla "*Enrolments*". Todas las inscripciones se generan como reforzamiento básico para posteriormente cambiar a otro tipo de inscripción. (Véase Apéndice C-4). Se cambia a complemento virtual si el usuario tiene el código de promoción para acceder a este tipo de inscripción. (Véase Apéndice C-5). El botón enviar de la vista válida que el campo no este vacío.
- En el endpoint de success implementado para actualizar el estado de la orden de compra descrita previamente en el Sprint 3, a la tabla Enrolments se le actualizó el campo de tipo de inscripción por el enum que posee el string "*reforzamiento premium*".
- Para mostrar las materias por tipo de inscripción del estudiante desde su dashboard, se crearon dos peticiones desde el controlador *dashboard.js* a backend, donde backend responde a la petición *complementEnrolment* la lista de materias que son de tipo complemento virtual y la petición *tutoringEnrolment* las materias que son de tipo reforzamiento premium o reforzamiento básico que el estudiante tenga inscrito a la fecha. El controlador *dashboard.js* recibe dos JSON con ambas listas y con la directiva $scope se pasan a la vista y se iteran las listas mediante la directiva ng-repeat en la etiqueta *div* de HTML. (Véase Apéndice C-7).

- La elaboración del retiro del estudiante se realiza en la vista de inscripción de la materia, donde se incluyó un botón para que el Cliente haga la petición de retiro al Servidor. En el Servidor se le hace un update a la tabla *Enrolments* y la columna enrolmentType a "inactive" si son de tipo reforzamiento básico o complemento académica y a la columna cancelled se modifica a "true". Si el estudiante quiere volver a inscribirse, se genera una nueva inscripción y la anterior deja de ser válida, para implementar esta regla de negocio se le agregó el condicional cancelled false al query del método *checkForEnrolment* que verifica si está inscrito el estudiante.

4.4. Sprint 3

4.4.1. Objetivo

Diseñar e implementar submódulo de precios, pagos y promociones de cursos.

4.4.2. Sprint Backlog

- Crear CRUD de producto.
- Generar CRUD de promoción para la materia. Generar códigos de descuento. Listar los códigos de promoción.
- Validar que un usuario puede utilizar el código de promoción, registrar el usuario que utilizó el código.

- Crear cuenta de Dawere en MercadoPago.
- Crear tablas que contengan la información de orden de pago, intento de pago.
- Realizar la compra del producto desde MercadoPago. Guardar información pertinente de la compra.
- Generar redirecciones a la aplicación Dawere donde se manejen el "Success", "In_Pending" y "Failure" de MercadoPago.
- Incluir vista de resumen de compras cuando vas a pagar.

4.4.3. Desarrollo

- Las funciones de crear, modificar, eliminar o ver la lista de precios de los productos se realizan mediante la sesión del rol administrador.
- El rol administrador tiene la opción de registrar una nueva promoción para una alianza, como también modificarla y consultar una lista que posee la relación entre alianza, sus promociones y los códigos de esa promoción. (Véase Apéndice C-3). Registrar promoción tiene incluida la funcionalidad de definir la cantidad de códigos que se desean generar de esa promoción. En backend se registra la promoción en la tabla "*Promotions*" y los códigos en la tabla "*Codes*" de base de datos, donde esta posee una clave foránea de la promoción asociada. Para el identificador string de los códigos se elaboró un randomize con la función *rnd* de Java, previamente se declaró en una variable los caracteres que se incluirán de manera aleatoria en el string a generar.

- En la tabla Users_Codes se enlaza qué usuario utilizó el código asignado.
- Para los pagos en la plataforma se hizo mediante el API de MercadoPago. Se registra una cuenta de correo en MercadoPago, posteriormente crear una preferencia que tenga la información de número de cuenta, razón social, nombre, RIF y teléfono. En esta preferencia llegarán los pagos que se hayan realizado en MercadoPago de los productos de Dawere.
- La tabla orden de pago "*Purchase_Orders*" tiene información pertinente al pago, una orden puede tener un estado *approved*, *pending* o *failure*, código de la preferencia generado por MercadoPago, fechas, monto que posee el producto a la fecha, entre otras columnas. La tabla intento de compra "*Purchase_Attempts*" representa la cantidad de veces que un usuario intenta una compra del mismo producto, es decir, sólo existe una orden de compra del producto y si el intento de pago es fallido aplica la misma orden pero con otro intento de pago.
- El usuario estudiante está interesado en realizar el pago del producto, se le muestra el monto y el botón pagar. (Véase Apéndice C-6). Al accionar el botón pagar en la vista empieza la interacción entre el Cliente Dawere, Servidor de Dawere y el Servidor de MercadoPago. La explicación de la comunicación entre el Cliente Dawere, Servidor de Dawere y el Servidor de MercadoPago está descrita siguiendo la

documentación del API de MercadoPago, sigue el siguiente patrón: (Véase Figura 5).

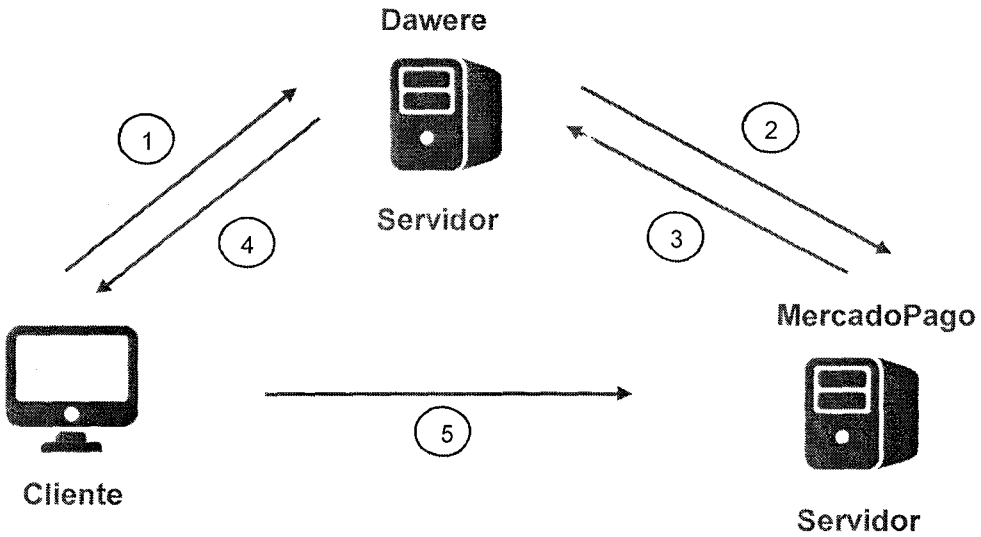

Figura 3. Arquitectura MercadoPago.

Fuente: Elaboración propia (2016).

El Cliente Dawere hace una petición HTTP al Servidor Dawere para comprar el producto seleccionado. Valida que el usuario puede comprar el producto y procede a crear una orden de compra y guardarlo en base de datos. El Servidor Dawere posteriormente a crear la orden de compra procede a realizar una petición POST al Servidor de MercadoPago con la finalidad de crear una nueva preferencia de pago. A través de un JSON se le envía a MercadoPago información referente a la compra como el monto, el nombre del producto, cantidad y los URL donde MercadoPago va a redireccionar si es exitoso, pendiente o falla de la transacción. MercadoPago crea la nueva preferencia y envía en un JSON al Servidor de

Dawere información como el id de la preferencia, monto, el tipo de pago y el URL de MercadoPago donde se procederá a realizar el pago. El Servidor Dawere guarda en la tabla "*MP_Preferences*" los datos de la nueva compra y tiene incluido una clave foránea de la tabla "*Purchase_Orders*". El Servidor Dawere le responde la petición al Cliente con el URL que envió MercadoPago en la preferencia. Desde el controlador *purchaseSummary.js* se valida que la respuesta del Servidor Dawere sea exitoso. El Cliente de Dawere al obtener el URL, desde el controlador se procede a redireccionar con la función *$window.location.href* de AngularJS a la página de MercadoPago. En la página de MercadoPago el Cliente puede seleccionar qué tipo de pago quiere realizar, puede encontrar las opciones de tarjeta de crédito, tarjeta de débito, depósito y transferencias. (Véase Apéndice C-11).

- Cuando el Cliente de Dawere compra el producto desde MercadoPago, MercadoPago se encarga de redireccionar al Cliente de Dawere de nuevo a la página de Dawere. El Servidor Dawere le indicó en la preferencia (Paso 2) a MercadoPago a donde va a redireccionar a los Clientes cuando finalicen la compra. Existen 3 tipos de redirecciones al frontend de Dawere que serían success, pending y failure. MercadoPago al redireccionar a Dawere en el URL agregar los parámetros collection_id, collection_status y preference_id indicando el producto que el Cliente canceló, esta data es cacheada en el controlador e incluida en la petición a backend. La funcionalidad de cada redirección es la siguiente:

- **Success**: El pago fue exitoso, se le indica al usuario que puede acceder al producto comprado inmediatamente. En el Servidor Dawere existe un endpoint para cuando el pago es success, en ese método se cambia el status de la orden de compra a *approved*.
- **In_pending**: El pago está en proceso, MercadoPago tiene de 1 a 3 días hábiles para confirmar la transacción si fue exitosa o no. El Servidor Dawere tiene habilitado un endpoint destinado a las respuestas de MercadoPago sin pasar por los filtros, donde MercadoPago pasa un topic_id y un id, mediante el id el Servidor Dawere hace una petición POST a MercadoPago para preguntar por el estado de la compra y actualizar el estado de la orden de la compra.
- **Failure**: MercadoPago redirecciona al Cliente a la página de failure. Desde el Cliente se le hace una petición al Servidor Dawere y se actualiza a failure la orden de compra.

- La vista de resumen de compra contiene una previsualización de la compra, cual es el producto, el monto, descripción. El controlador se encarga de hacer una petición al Servidor para obtener los datos pertinentes.

4.5. Sprint 4

4.5.1. Objetivo

Diseñar e implementar submódulo de alianzas estratégicas.

4.5.2. Sprint Backlog

- Generar CRUD de la tabla "Users" desde el rol administrador. Incluir el rol Alianza (institución académica).

4.5.3. Desarrollo

- Se le incluyó al rol administrador la funcionalidad de crear, ver, modificar y eliminar un usuario tipo estudiante o alianza. (Véase Apéndice C-2). Cada rol tiene campos propios, se muestran los campos dependiendo del rol seleccionado. Se elaboró todo el manejo de backend de los modelos, recursos y servicios de las entidades "*Students*" y "*Alliances*".

4.6. Sprint 5

4.6.1. Objetivo

Diseñar e implementar submódulo de contabilidad.

4.6.2. Sprint Backlog

- Elaborar una factura como constancia si la compra se realizó con éxito.
- Lista de facturas de las compras de reforzamiento premium.
- Generar una nota de crédito si el estudiante realizó una compra y esta compra se encuentra dentro de los días hábiles.

4.6.3. Desarrollo

- En el endpoint de success se implementó también la generación de una factura, se guarda el número de control, el total cancelado, retención, el método de pago y el orden de pago que pertenece en la tabla "*Invoices*" de la base de datos. El número de control es una iteración, se busca el último guardado en base de datos para fijar el siguiente número de control. Al monto del producto de la materia se le calcula el 12% para ser sumado al monto final.
- El rol administrador desde su dashboard puede seleccionar en la pestaña "Compras > Facturas" un listado de facturas, donde se implementó un filtro por fechas y por usuarios para la búsqueda de una factura con la función *filter* que provee AngularJS. Al seleccionar el botón ver factura se hace una petición HTTP con el id de la factura al Servidor, en el Servidor se realizan querys a las tablas de Invoices, Purchase_Orders, Products_Purchases y Products para obtener la data que se mostrará en la vista del Cliente. Se elaboró un diseño de la factura y fue implementado en una vista.(Véase Apéndice C-10).
- La nota de crédito sólo aplica para los estudiantes que tienen una inscripción de tipo reforzamiento premium y tengan solo 10 días de haberse inscrito. En backend se válida estas reglas de negocio para hacerle un update en la tabla Enrolments y cambie el tipo de inscripción a credit_note. El administrador puede verificar las notas de crédito desde su sesión, en su dashboard "Compras > Notas de

crédito" le aparecen las solicitudes de notas de crédito, el administrador se encarga de aprobar la generación de una nueva nota de crédito en el botón generar nota de crédito. La petición al Servidor implementa un método donde registra en la tabla "*Credit_Notes*" la nueva nota de crédito con el monto, la fecha, la factura que tiene asociada, número de control y cambiar el estatus a cancelado de la factura en la tabla "*Invoices*".

4.7. Sprint 6

4.7.1. Objetivo

Diseñar e implementar submódulo de encuestas.

4.7.2. Sprint Backlog

- Implementar manejador de encuesta donde se pueda crear la encuesta, guardar preguntas y opciones.
- Implementar que el estudiante pueda presentar una encuesta. Guardar en base de datos la data presentada.
- Listar las encuestas.
- Realizar una estadística de la encuesta mediante gráficas con highcharts.

4.7.3. Desarrollo

- La encuesta posee un título, una descripción, preguntas y opciones. El usuario administrador tiene acceso a crear una encuesta, en su

dashboard se encuentra la opción "Encuesta > Crear" para hacer uso de este requerimiento. (Véase Apéndice C-9). El administrador llena el formulario título, descripción de la vista y la materia que se le va a pasar la encuesta. Al hacer click en el botón de agregar nueva pregunta se llama una función *addQuestions()* que se encuentra en el controlador *createSurveyController.js* donde se agrega nueva pregunta en una lista con el método *push* propia de AngularJS, esta lista es recorrida nuevamente en la vista automáticamente con la directiva ng-repeat de la variable $scope.questions mostrando el tipo de pregunta y descripción. Para agregar una nueva opción de la pregunta se selecciona el botón "agregar nueva opción", al hacer click llama a la función *addOptions(question)* donde se le pasa la pregunta que pertenece y se hace *push* a la variable *question.optionList* para incluir un nuevo elemento de tipo opción a la lista. En la vista HTML se encuentra el iterador de la directiva ng-repeat de la variable *question.optionList*. Al seleccionar el botón aceptar llama a la función *send(survey,questions)* que se encuentra en el controlador de la vista y se encarga de enviar la petición a backend con la información referente a las preguntas y respuestas de la encuesta.

- Para presentar una encuesta desde estudiante, el controlador de la vista *submitSurvey.html* carga la función *renderSurvey*, este llama al servicio *surveyService.getSurvey(surveyId)* para realizar la petición HTML al backend y este retorne la encuesta con la lista de preguntas

y opciones. En la vista mediante las directivas ng-repeat y ng-if se despliegan la lista de preguntas y opciones de acuerdo a su tipo de opción múltiple o simple. Al terminar la encuesta el estudiante selecciona Aceptar y llama a la función *sendAnswers(survey.questionList)* donde están guardadas todas las respuestas seleccionadas por el estudiante, estas son enviadas al endpoint *survey/submitSurvey* del Servidor, se encarga entonces de iterar la lista de opciones respondidas y guardar las respuestas en la tabla *Answers_Surveys*, esta tabla tiene las foráneas fk_question_survey, fk_option_survey, fk_user y la fecha que realizó la encuesta.

- El administrador puede acceder a las estadísticas de la encuesta seleccionando "Encuesta > Index", el método *getSurveys()* que se encuentra en el controlador tiene la funcionalidad de llamar a *surveyService.getSurveysList* para hacer la petición GET a backend y retornar la lista de encuestas que se hayan realizado. El botón ver reporte tiene una redirección con la directiva *ui-sref* a *app.survey.showReport({id:survey.id}).* En la vista *showSurveyReport* se muestra las estadísticas de cada pregunta, el controlador hace una petición a backend mediante la función *surveySelected*, se le pasa el id de la encuesta seleccionada. El backend responde con un JSON la data de la encuesta con sus preguntas, opciones y respuestas. Si la pregunta tiene menor o igual a 3 opciones se utiliza una gráfica de torta y si es mayor a 3 una gráfica de barras.

- Se utilizó la librería HighCharts de JavaScript para construir gráficas, mediante un JSON que representa la configuración de la gráfica a mostrar. A la configuración se le pasa la información de las opciones al eje "x" y la cantidad de respuestas que posee la opción al eje "y", el estilo que poseerá la torta o la barra, la leyenda, título, entre otros. La etiqueta SVG que es implementada en el documento HTML recibe la configuración JSON y se encarga de construir el gráfico.

4.8. Sprint 7

4.8.1. Objetivo

Diseñar e implementar submódulo de comunicados.

4.8.2. Sprint Backlog

- Configurar SendGrind en backend para envíos de correos.
- Colocar todos los emails como templates y sustituir cuando sea necesario.
- Incluir correos en las funcionalidades desarrolladas previamente.
- Implementar formulario de quejas.

4.8.3. Desarrollo

- Para la configuración de SendGrind se incluyó la dependencia "*sendgrid-java*" en el archivo *pom.xml*, esta dependencia posee una clase SendGrid donde se le incluye el username y contraseña de Dawere, la cuenta es creada previamente desde el portal de

SendGrid. El método "*SendGrid.Email*" tiene como finalidad crear el nuevo correo a enviar, posee las propiedades: *addTo* para incluir el destinatario, *setFrom* para incluir el remitente del correo, *setHtml* es la descripción del correo, *setFromName* es el título del correo y el método *send* es para enviar el correo a SendGrid y este se encarga de realizar las conexiones pertinentes con los distintos proveedores de correos.

- Se implementó el método de leer un HTML en el Servidor mediante un BufferedReader, esta función se encargará de interpretar el archivo HTML que posee el estilo y etiquetas de un documento HTML y con el método *parseHtmlToString* realizar la conversión a string. Se elaboró la conversión a string porque los HTML poseen variables tipo "$$xxx$$" que representan un tag con la finalidad de ser reemplazadas mediante la función *replace* de Java por la data que le hace referencia.
- Se elaboró el método de enviar email por tipo de inscripción, se le informa al estudiante la materia, el profesor, el año y el URL para ingresar al aula virtual. Se implementó el envío de correo a la alianza con su lista de códigos de promoción, la promoción que le hace referencia y el nombre de la alianza. (Véase Apéndice 8). Se implementó el envío de correo con el link de la factura donde el estudiante podrá visualizarlo en la plataforma, la nota de crédito posee la misma lógica. Se elaboró un formulario de quejas para que los usuarios realizaran observaciones, constará de tipo de

observación y descripción. (Véase Apéndice C-12). Existe una tabla Complaints en base de datos donde se guarda los campos del formulario más una columna de visto que es un booleano. El administrador podrá ver las quejas desde la pestaña "Quejas > Index" de su dashboard, y podrá seleccionar cual ver.

4.10. Sprint 8

4.10.1. Objetivo

Diseñar e implementar submódulo de comunicados.

4.10.2. Sprint Backlog

- Generar certificados al terminar la materia y listas los que existan.

4.10.3. Desarrollo

- En la vista "Mis logros" se encuentra el listado de cursos con certificados, donde en el query de backend valido si el estudiante culminó de manera exitosa el curso de complemento académico. Para construir el certificado se utilizó la librería jspdf de JavaScript, donde la librería crea el archivo PDF, se parsea el documento HTML a base64 y para abrirlo en el browser se pasa el string generado por base64 a la función *window.open* de JavaScript para la visualización del mismo. Esta librería posee la opción de descargar archivo.

4.11. Sprint 9

4.11.1. Objetivo

Definir los procesos administrativos y directivos.

4.11.2. Sprint Backlog

- Elaborar un diagrama de flujo indicando el proceso de inscripción administrativa y académica del estudiante.
- Elaborar un diagrama de flujo indicando el proceso para el retiro del estudiante.
- Elaborar un diagrama de flujo representando el proceso de generar códigos de promoción

4.11.3. Desarrollo

- El proceso inicial para generar una inscripción es realizar una inscripción académica donde se enlazan la materia con el usuario. (Véase Apéndice B-1). La inscripción cuando es de tipo reforzamiento básico no posee una inscripción administrativa. La inscripción cuando es de tipo reforzamiento premium posee una inscripción administrativa al realizar la compra de una materia. (Véase Apéndice B-2). La inscripción cuando es de tipo complemento virtual posee una inscripción administrativa sólo al hacer uso del código de promoción. (Véase Apéndice B-3). La diferencia entre los 3 tipos de inscripciones es la permisología que pueden tener al ingresar al aula virtual de la materia.

- El retiro es igual para los 3 tipos de inscripciones, se le cambia el estatus a la inscripción. (Véase Apéndice B-4).
- Para generar códigos de promoción debe haber iniciado sesión desde un rol administrador. (Véase Apéndice B-5).

CAPÍTULO V: RESULTADOS

En este capítulo se define los submódulos obtenidos de los objetivos propuestos para este Trabajo Instrumental de Grado titulado "Módulo de Dirección para la Plataforma de Educación Online del Colegio Virtual Dawere".

- **Definir los procesos administrativos y directivos**

Se definió los procesos administrativos y directivos mediante la representación de diagrama de flujos de la inscripción y retiro del estudiante por tipo de inscripción (reforzamiento básico, reforzamiento premium y complemento virtual), retiro de una inscripción de la materia, generar códigos de promoción por alianza. (Véase Sprint 9).

- **Diseñar e implementar submódulo de precios y promociones de cursos**

Se diseñó e implementó el crear, ver, modificar y eliminar los precios de los productos de una materia, el administrador posee la permisología para esta funcionalidad. La implementación para el enlace del precio con la materia se hace en backend al crear una nueva materia.

Se diseñó e implementó el crear y modificar una promoción desde el administrador, donde al crear una promoción se enlaza o no con una alianza y se incluye la cantidad de códigos que serán propios de la promoción, si la promoción tiene una relación con la alianza se le envía a la alianza el listado de códigos solicitados. (Véase Sprint 3).

- **Diseñar e implementar submódulo de inscripción y retiro del estudiante**

Se diseñó e implementó la inscripción y retiro del estudiante, donde la inscripción puede ser de tipo reforzamiento básico, reforzamiento premium y complemento virtual, cada una posee distintas características, el reforzamiento básico se genera cuando seleccionas inscribir la materia, el reforzamiento premium para cuando se realiza un pago a través de MercadoPago de una materia y el complemento virtual para cuando se utiliza un código de promoción. La implementación de retiro es igual para los tres tipos de inscripciones, se le cambia el status a la inscripción. (Véase Sprint 2).

- **Diseñar e implementar submódulo de contabilidad**

Se diseñó e implementó la emisión de facturas y notas de crédito, el estudiante podrá acceder a la constancia de inscripción administrativa cuando se inscribe como reforzamiento premium, el sistema le enviará un link al correo del estudiante para ir a la plataforma Dawere y visualizar su factura. La nota de crédito posee una implementación similar sólo que es generada cuando el administrador habilita la generación de la nota de crédito posteriormente a la petición de retiro del estudiante.

Como administrador se puede visualizar el reporte de facturas a través de "Compras > Facturas" y el reporte de retiros en "Compras > Nota de Crédito". (Véase Sprint 5).

- **Diseñar e implementar submódulo de encuestas**

Se diseñó e implementó el crear una encuesta con sus preguntas y opciones, al crear se selecciona la materia y el servidor se encarga de enviarle a través un correo la encuesta que el estudiante tiene que presentar, el estudiante puede llenar la encuesta mediante la plataforma. Como administrador en la pestaña "Encuesta > Estadística" se encuentra las gráficas tipo torta y barras como representación de las cantidades de respuestas que posee una opción de la pregunta de la encuesta. (Véase Sprint 6).

- **Diseñar e implementar submódulo de alianzas estratégicas**

Se diseñó e implementó un submódulo de alianza, donde la alianza es la representación de un colegio público o privado, es creada como un usuario tipo alianza y posee su formulario pertinente. Como administrador se puede visualizar a la alianza y sus promociones, códigos y los estudiantes de su colegio. (Véase Sprint 4).

- **Diseñar e implementar submódulo de diplomas, certificados y constancias**

Se diseñó e implementó la emisión de documentos avalados por Dawere, donde un certificado o diploma se obtiene sólo cuando finaliza todos las evaluaciones de la materia y puede ser visualizado en la vista "Mis Logros" (este documento puede ser exportado por el estudiante), la constancia es la

factura del pago realizado por el estudiante cuando se inscribe como reforzamiento premium y el estudiante recibe un correo con el link a la vista que contiene toda la información referente a la factura. (Véase Sprint 8).

- **Diseñar e implementar submódulo de comunicados**

Se diseñó e implementó un submódulo de quejas para los usuarios de rol estudiante, pueden seleccionar el tipo de observación y escribir el problema que estén presentando. Estas quejas pueden ser vistas por el administrador en la pestaña "Quejas > Index" de su dashboard.

Se implementó SendGrid para el envío de correos electrónicos, el estudiante recibe correos cuando realiza una inscripción sea reforzamiento básico, reforzamiento premium o complemento virtual, cuando realiza una compra de una materia se le envía un link con el URL de la factura, cuando se crea una nueva alianza, cuando el administrador genera una encuesta y esta es enviada a los estudiantes de la materia, cuando la alianza genera una lista de códigos de promoción se le envía la lista de códigos a la alianza. (Véase Sprint 7).

- **Diseñar e implementar la base de datos**

Se diseñó un modelo entidad relación (MER) para la solución propuesta, se obtuvo mediante el levantamiento de información realizada en cada Sprint y dentro del Sprint se analiza e incluía nuevas entidades al modelo de entidad relación base.

Para su implementación se construyó una configuración con el framework Hibernate, en esta configuración se incluyen los objetos que serán mapeados de Maven a MySQL, pool de conexiones con la base de datos, usuario y contraseña de MYSQL. Se elaboró métodos globales de mapeo en el Servidor para no repetir la implementación de los obtener, eliminar, modificar y guardar de una base de datos. (Véase Sprint 1).

- **Aporte tecnológico**

Debido a la escalabilidad del proyecto de llevar la educación a latinoamerica, se optó por Java por lo robusto y por las múltiples herramientas que posee, por su popularidad en el mercado y por los conocimientos de los programadores en este lenguaje de programación.

- **Aporte funcional**

Al definir los procesos de la empresa se puede evaluar y optimizar las implementaciones desarrolladas.

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

- Mediante los resultados descritos previamente del presente Trabajo Instrumental de Grado, se obtuvo las siguientes conclusiones:
- La plataforma es fácil de utilizar para las inscripciones académicas y administrativas del estudiante, como también la visualización de las mismas por parte del administrador del sistema.
- Los nuevos procesos diseñados para la gestión de la aplicación permiten ordenar la estructura de la plataforma y sus funcionalidades.
- La implementación de frameworks permitió obtener una aplicación web con código más fácil de mantener y minimizó el tiempo empleado para la construcción del mismo.
- La implementación de encuestas facilita recolectar información para aplicar mejoras en la plataforma.
- El submódulo de notificaciones mediante el envío de correos permite informar al estudiante sobre la transacción que haya realizado dentro de la plataforma.
- El uso de la metodología ágil Scrum permitió tener al dueño del producto involucrarlo en cada una de las reuniones de los Sprints, garantizando un mínimo de errores en el producto y permitiendo una visualización previa de la plataforma.

- Este emprendimiento ha posicionado a la empresa Dawere como pionera de la educación en línea en Venezuela al contar con una plataforma que permite manejar procesos e información de una entidad educativa.

RECOMENDACIONES

- Se recomienda crear una lista de correos de usuarios a quienes se le desea invitar a participar en la plataforma.
- El estudiante debería poder generar su propio código de promoción o invitar a una alianza a la plataforma.
- Al culminar los estudiantes las materias, se recomienda enviar una notificación a su representado.
- Los correos que son enviados a Gmail a través de Sendgrid, aleatoriamente entran a la bande de promociones. Se recomienda incluir la llegada a la bandeja principal del usuario.
- Desarrollar la aplicación para los dispositivos móviles Android, iOS para todo público, que permita visualizar en forma general sin tantos detalles el aula virtual.

REFERENCIAS BIBLIOGRÁFICAS

[1] Palacio, J. (2015). Gestión de proyectos Scrum Manager (Scrum Manager I y II). Recuperado el Abril de 2015. Desde el sitio web scrummanager: http://www.scrummanager.net/files/gestion_proyectos_scrum_manager.pdf

[2] Maven. (2016). Introduction to the Standard Directory Layout. Recuperado el 10 de Mayo de 2016. Desde el sitio web apache:https://maven.apache.org/guides/introduction/introduction-to-the-standard-directory-layout.html

[3] Java. (2016). Chapter 16 Security. Recuperado el 11 de Marzo de 2016. Desde el sitio web java: https://jersey.java.net/documentation/latest/security.html

[4] Java. (2016). Chapter 2 Modules and dependencies. Recuperado el 11 de Marzo de 2016. Desde el sitio web java:https://jersey.java.net/documentation/latest/modules-and-dependencies.html

[5] Sanchez, Jan. (2014). Grunt Tutorial Básico y Primeros Pasos. Recuperado el 3 de abril de 2014. Desde el sitio web frontendlabs: http://frontendlabs.io/146--grunt-js-espanol-tutorial-basico-primeros-pasos

[6] Eguiluz, Javier. (2010). Introducción a JavaScript. Recuperado el 20 de Mayo de 2010. Desde el sitio web librosweb:http://librosweb.es/libro/javascript/capitulo_1.html

[7] Fernández, Alberto. (2013). Servicios web RESTful con HTTP. Parte I: Introducción y bases teóricas. Recuperado el 12 de noviembre de 2013. Desde el

sitio web adwe: http://www.adwe.es/general/colaboraciones/servicios-web-restful-con-http-parte-i-introduccion-y-bases-teoricas

[8] Barrios, Juan. (2001). Java Servlets. Recuperado el 30 de Noviembre de 2001. Desde el sitio web ddc:http://users.dcc.uchile.cl/~jbarrios/servlets/general.html

[9] Hibernate. (2016). Hibernate Getting Started Guide. Recuperado el 11 de Febrero de 2016. Desde el sitio web jboss:http://docs.jboss.org/hibernate/orm/5.1/quickstart/html_single/

[10] Guitierrez, Carolina. (2014). ¿Para qué sirve el Scrum en la Metodología Ágil?. Recuperado el 16 de Mayo de 2014. Desde el sitio web: http://www.bing.com/search?q=metodologia+agil+para+que+sirve&src=IE-TopResult&FORM=IETR02&conversationid=

[11] JWT. (2015). Introduction to JSON Web Tokens. Recuperado el 8 de Mayo de 2015. Desde el sitio web: https://jwt.io/introduction/

Apéndice A – Estructuras

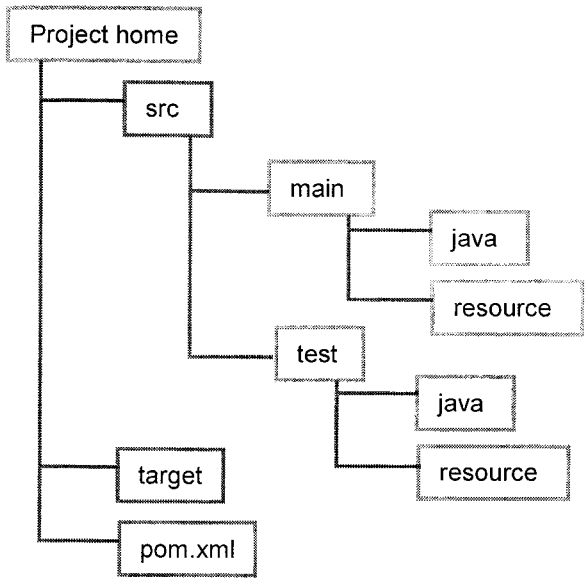

Figura A-1. Estructura de un proyecto Maven.

Fuente: Elaboración propia (2016).

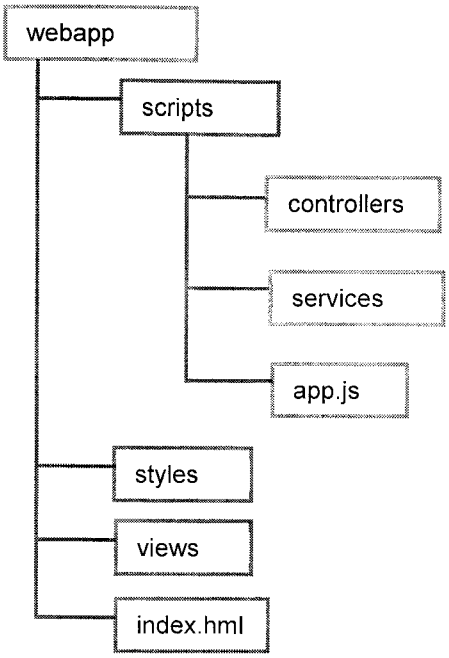

Figura A-2. Estructura de AngularJS.

Fuente: Elaboración propia (2016).

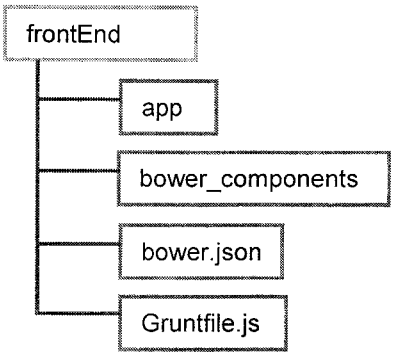

Figura A-3. Estructura frontend modificada.

Fuente: Elaboración propia (2016)

Apéndice B – Diagramas de Flujo

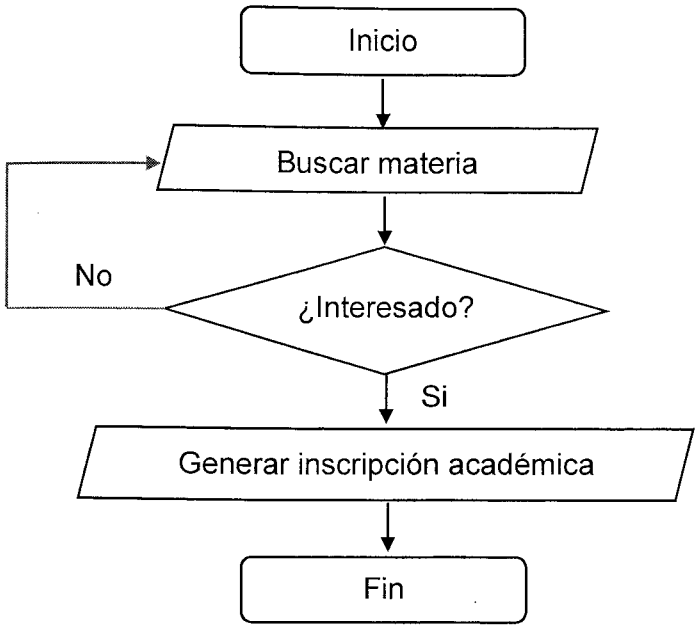

Figura B-1. Diagrama de Flujo: Inscripción académica.

Fuente: Elaboración propia (2016).

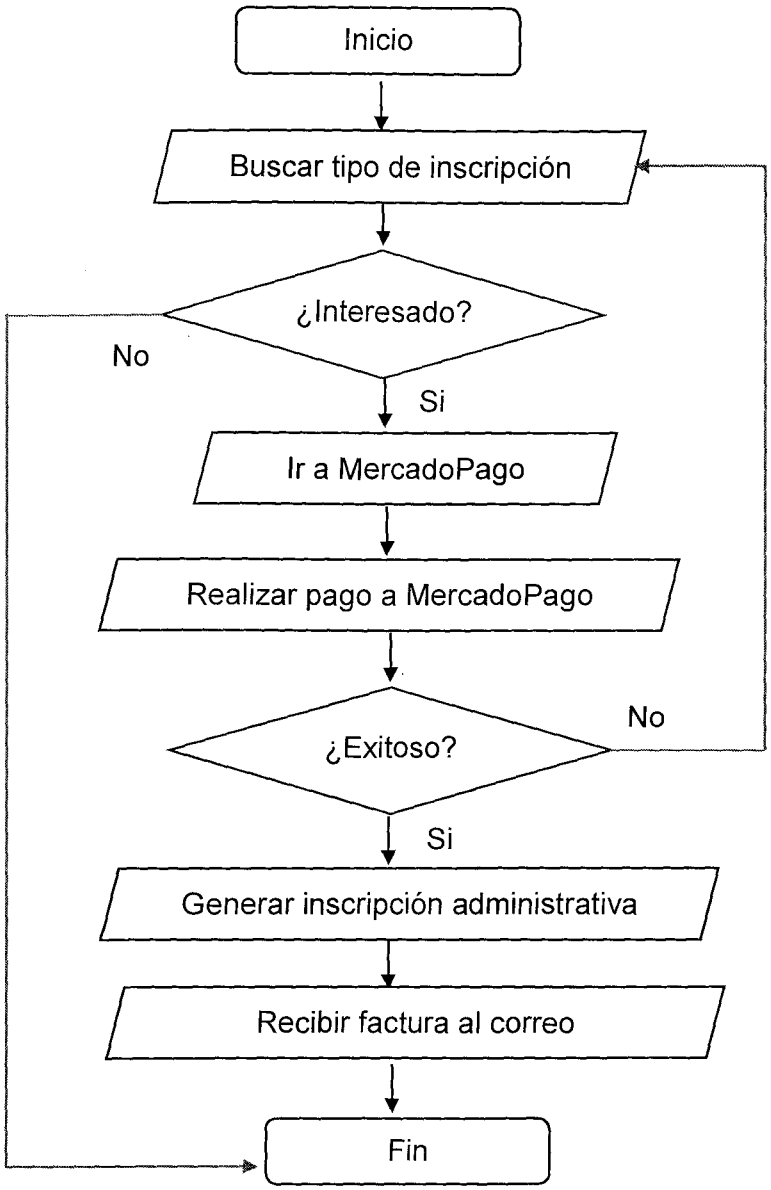

Figura B-2.

Diagrama de Flujo: Inscripción administrativo tipo Reforzamiento Premium.

Fuente: Elaboración propia (2016).

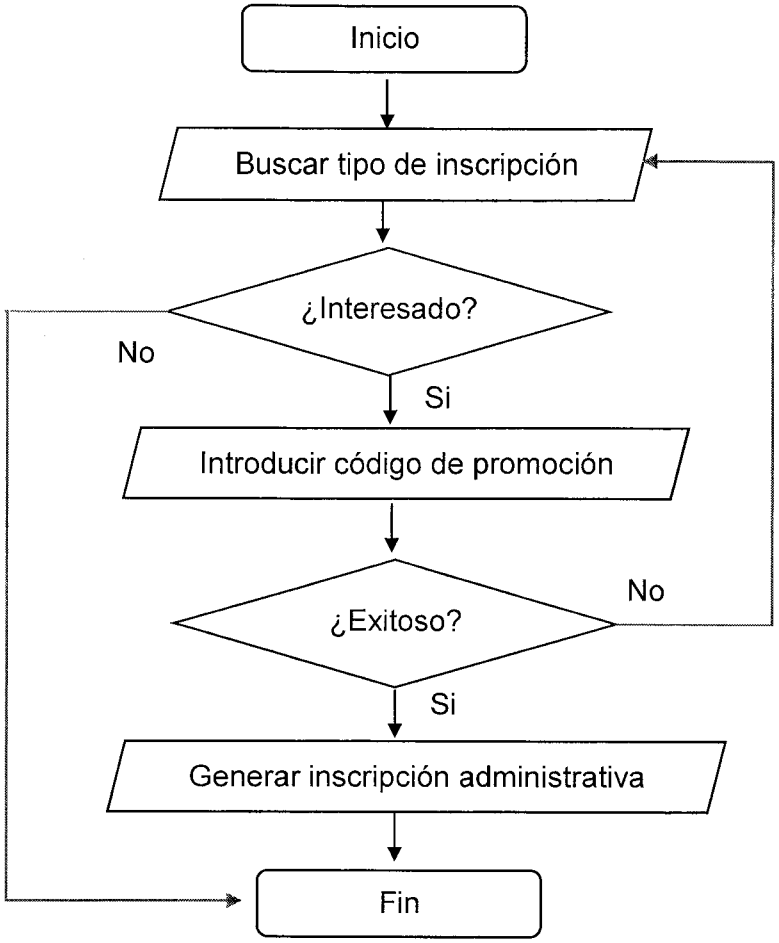

Diagrama de Flujo: Inscripción administrativo tipo complemento virtual.

Fuente: Elaboración propia (2016).

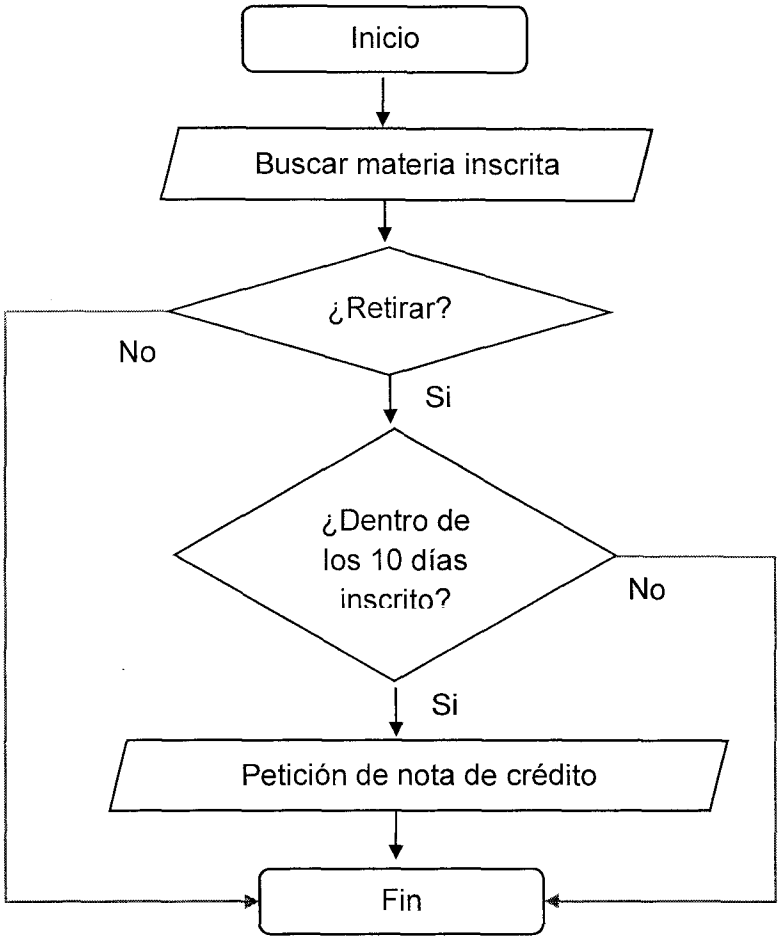

Figura B-4.

Diagrama de Flujo: Retiro de inscripción.

Fuente: Elaboración propia (2016).

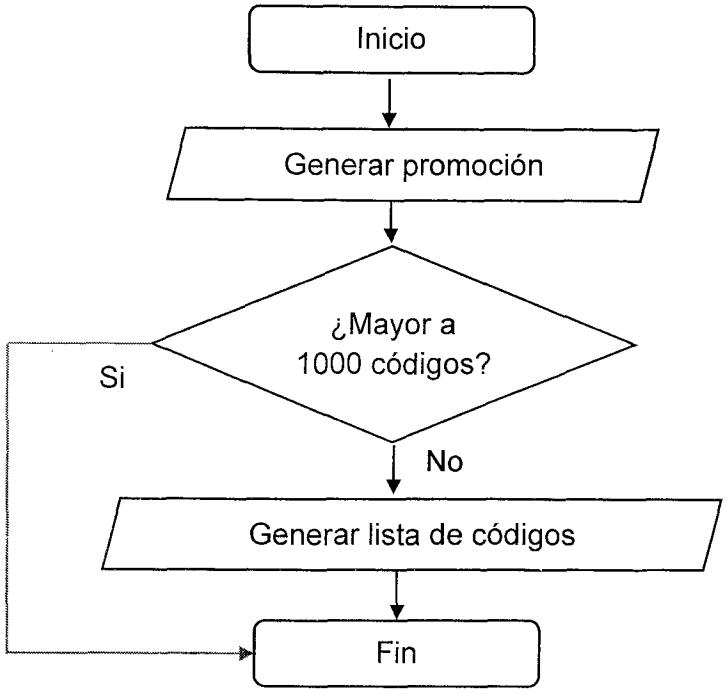

Figura B-5.

Diagrama de Flujo: Generar códigos de promoción desde Administrador.

Fuente: Elaboración propia (2016).

Apéndice C – Ilustraciones

Figura C-1.

Vista: Ver descripción de materia a inscribir.

Fuente: Elaboración propia (2016).

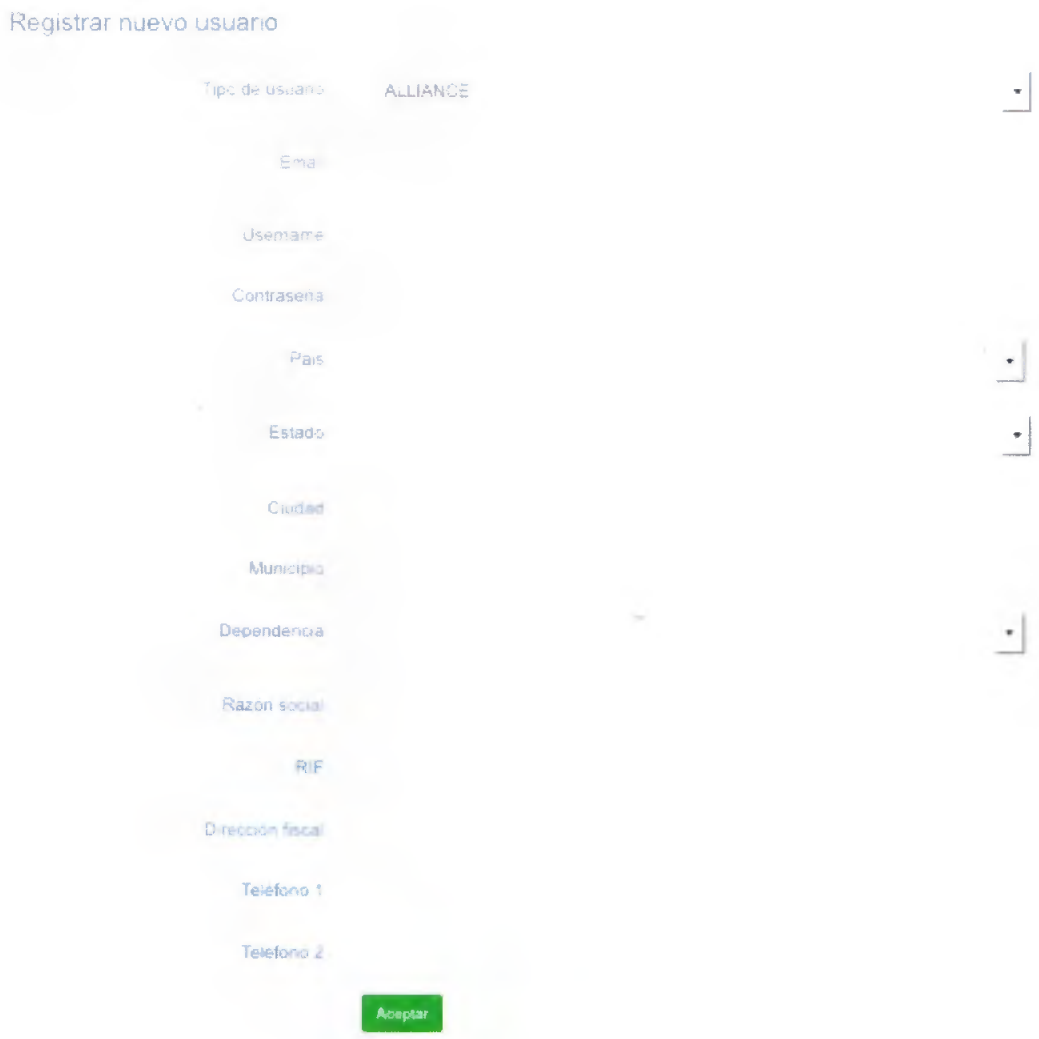

Figura C-2.

Vista: Registrar nuevo usuario Alianza.

Fuente: Elaboración propia (2016).

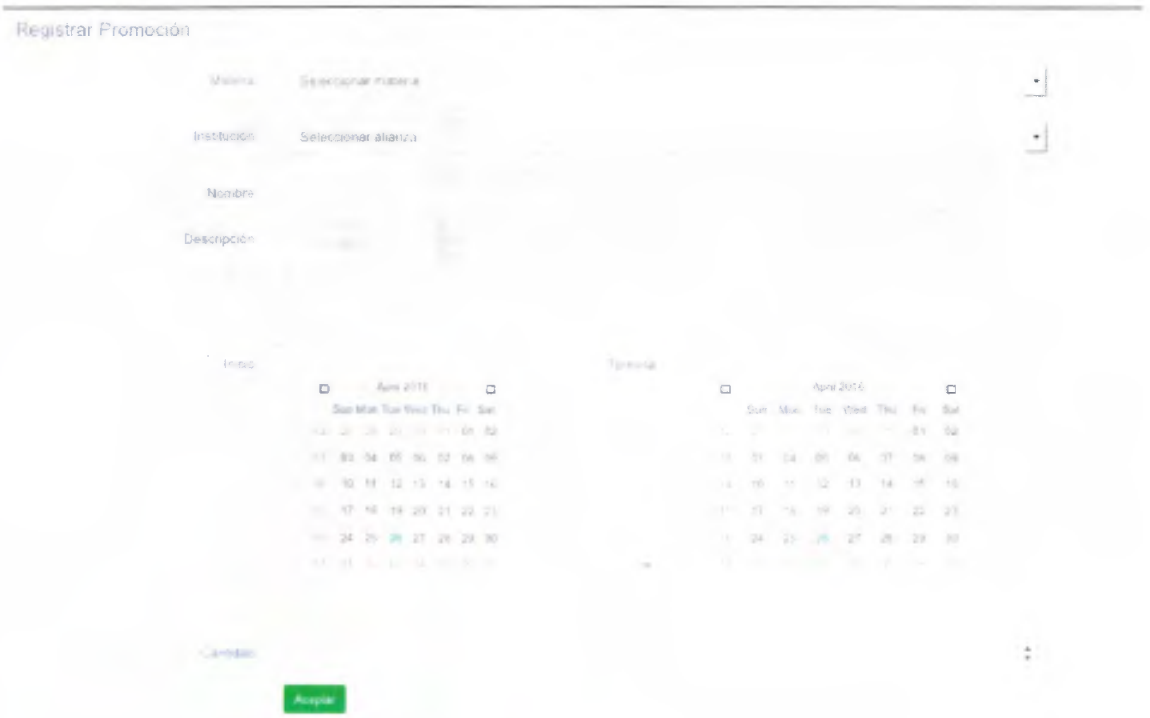

Figura C-3.

Vista: Registrar nuevo promoción con códigos.

Fuente: Elaboración propia (2016).

Figura C-4.

Vista: Tipo de inscripciones para ver una materia.

Fuente: Elaboración propia (2016).

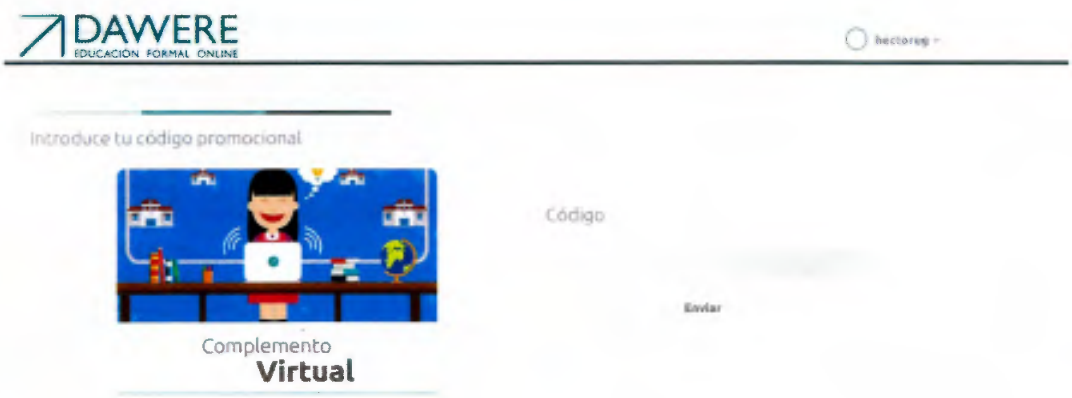

Figura C-5.

Vista: Validar y hacer uso del código de promoción.

Fuente: Elaboración propia (2016).

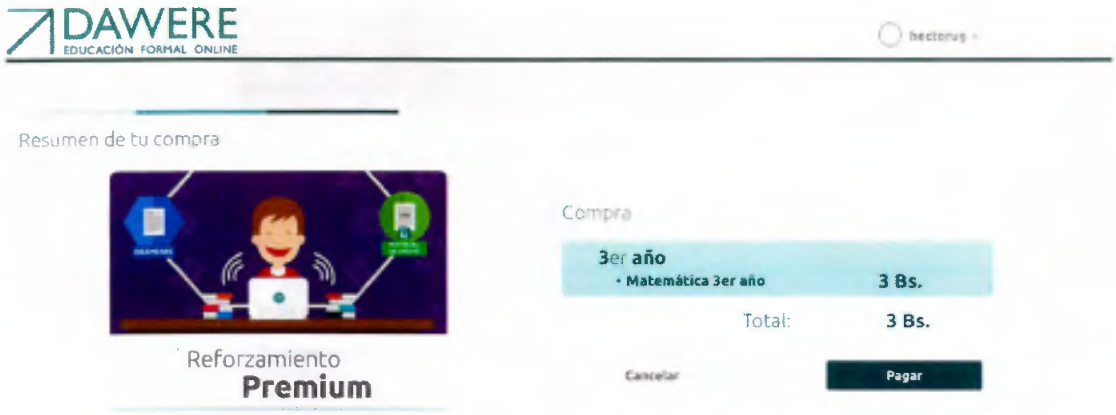

Figura C-6.

Vista: Visualizar la materia a pagar.

Fuente: Elaboración propia (2016).

Figura C-7.

Vista: Dashboard de estudiante.

Fuente: Elaboración propia (2016).

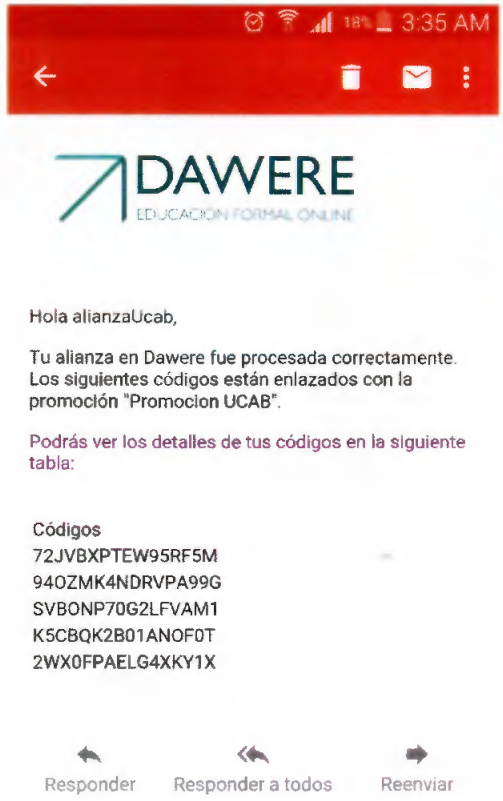

Figura C-8.

Vista: Lista de códigos de promoción para la alianza.

Fuente: Elaboración propia (2016).

Figura C-9.

Vista: Registrar nueva encuesta.

Fuente: Elaboración propia (2016).

DAWERE
EDUCACIÓN FORMAL ONLINE

Dawere, C.A. RIF: J-40669421-5 Av. 10, Casa San Judas Tadeo, Nro 483B, Urb. Alto Prado, Baruta, Caracas, Miranda, Zona Postal 1080.
Tlf: 0212 7815979

RESUMEN DE PAGO FACTURA FISCAL N° 3630058

FACTURA FISCAL NÚMERO:	3630058
FECHA DE EMISIÓN:	01-02-2016
FORMA DE PAGO:	Cheque / Transferencia/TDC
NÚMERO DE CONTROL	00-05305453

PRODUCTO:	EDUCACIÓN			
SERIE:	A			
CLIENTE:	67134	PETRA MENDEZ	CP	1050
DIRECCIÓN:	AV. 16 PASEO LOS JABILLOS, MIRAFLORES, CAPITOLIO. MUNICIPIO LIBRTADOR		Nro. NIT:	
			Nro. RIF:	J-25678345
TELÉFONO:	0212 5677890			
CIUDAD:	CARACAS -VENEZUELA			

RESUMEN:

Número	Fecha	Producto	Cantidad	Precio unitario	Total
2029579	05-01-2016	Materia	1	35.00	35.00
	Total	1			35.00

BASE IMPONIBLE Bs.	IVA 12 %	TOTAL FACTURA Bs.	MENOS RETENCION ISLR SOBRE SERVICIO 2 %:	MONTO A PAGAR Bs.
35.00	4.20	39.20	-0.70	38.50

NOTA:
XXXX

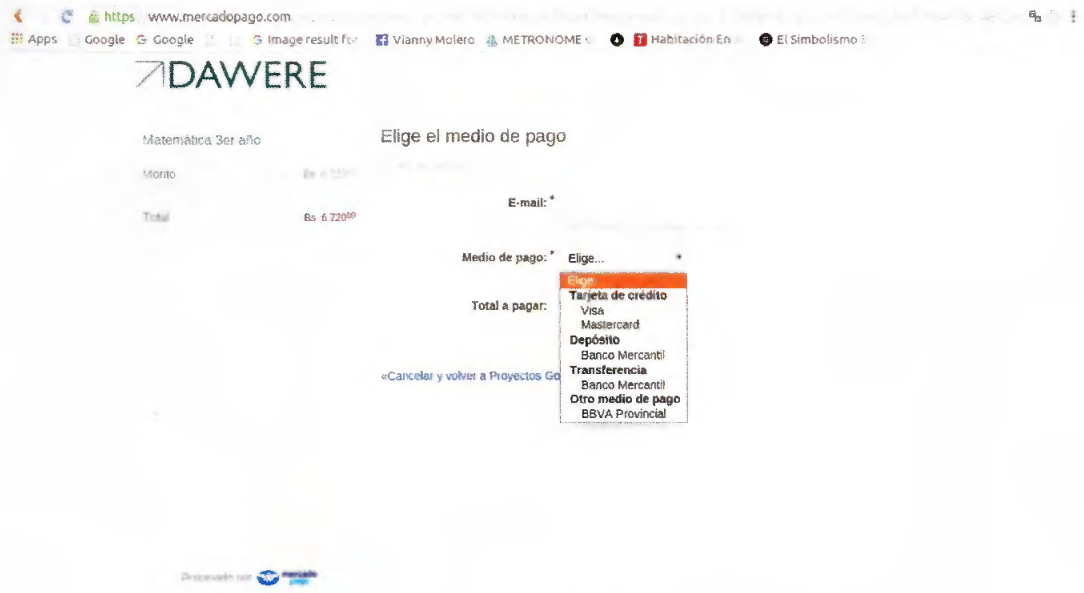

Figura C-11.

Vista: Redirección a MercadoPago.

Fuente: Elaboración propia (2016).

Figura C-12.

Vista: Formulario de quejas.

Fuente: Elaboración propia (2016).