

Tesis pregrado GEDC

por Fischer Martinez Dody Adonys Llerena Aguilera

Fecha de entrega: 11-dic-2019 05:23p.m. (UTC-0500)

Identificador de la entrega: 1232631817

Nombre del archivo: extracto_2019126203220.docx (82.42K)

Total de palabras: 8745

Total de caracteres: 46972

RESUMEN

El desarrollo de la plataforma web GEDC LATAM, fue culminado con éxito, esta aplicación permite a los decanos de las escuelas latinoamericanas de ingeniería pertenecientes a GEDC colaborar entre sí, ofreciendo actividades para mejorar el entorno educativo.

La aplicación de la metodología SCRUM demuestra la eficiencia del desarrollo de aplicaciones web con el framework Django. Dentro del Product Backlog se definieron cuatro Sprint de una semana cada uno, donde los primeros tres se centraron en el desarrollo de la plataforma, y el restante en las pruebas, dicho Sprint se realizó en un tiempo inferior al estimado, teniendo en cuenta un período de 8 horas diarias.

Entre otros beneficios que vale la pena destacar es que el marco de Django se basa en la arquitectura del Modelo Vista Controlador (MVC), la misma con la que estamos familiarizados, como desarrolladores y estudiantes del título de Ingeniería en Sistemas Computacionales.

Palabras clave: plataforma web colaborativa, aplicación de SCRUM, Django Framework.

ABSTRACT

The development of the GEDC LATAM web platform, was successfully completed, this application allows the deans of the engineering schools belonging to the GEDC to collaborate with each other, offering activities in order to improve the educational environment.

The application of the SCRUM methodology demonstrates the efficiency of web application development with the Django framework. Within the Product Backlog four Sprint of one week each were defined, where only the first three were focused on the development of the platform, and the remaining in tests, said Sprint were made in a time less than estimated, taking into account a period of 8 hours a day

Among other benefits worth highlighting is that the Django framework is based on the Model Vista Controller (MVC) architecture, the same with which we as developers and students of the Computer Systems Engineering degree are familiar.

Keywords: Collaborative web platform, SCRUM application, Django framework

CAPITULO I

INTRODUCCION

Debido a la necesidad de colaboración entre las escuelas de ingeniería latinoamericanas que pertenecen al GEDC se busca solucionar este problema de mediante el desarrollo de una plataforma web colaborativa, la misma será escrita en lenguaje Python en conjunto de herramientas como JavaScript, CSS, HTML, el framework de Django entre otras, centrándose en ofrecer a los diferentes miembros la oportunidad de promocionar sus eventos académicos.

Una herramienta que simplifica este proceso de desarrollo es el framework de Django, permitiendo a los desarrolladores de la plataforma centrarse netamente en la lógica, y así ahorrar tiempo en el desarrollo, otro punto que cabe destacar es el uso de la metodología de desarrollo ágil SCRUM, para dividir las tareas de manera eficiente en cada reunión.

Además de permitirle a los miembros del directivo del GEDC gestionar los datos de manera efectiva por medio de auditorías de usuario, se podrá administrar cambios en los directivos (al traspasar la presidencia de un miembro a otro), esto facilitará el trabajo de los futuros miembros, evitando que desarrollen una nueva página web para cada evento anual.

1.1. PLANTEAMIENTO DEL PROBLEMA

Los miembros del GEDC (Consejo Global de Decanos de Ingeniería) en español, no cuentan con una plataforma en la web para compartir recursos, contactarse y colaborar entre ellos, por lo consecuente la comunicación de los miembros resulta deficiente y rara vez es posible la colaboración. Por lo general la única forma de comunicación con la que cuentan son los eventos globales del que se realizan una vez cada año.

Existe una enorme gama de ofertas dentro de cada universidad que puede ser aprovechada no solo por sus estudiantes, sino también por el alumnado de academias vecinas dentro del país, e incluso en el extranjero, sin embargo, estas ofertas no se encuentran correctamente definidas, no tienen un alcance bien definido o simplemente esa oportunidad queda bloqueada, y solo unos pocos pueden resultar beneficiados. Los mismos decanos pueden desconocer el potencial que existe en el trabajo colaborativo, los beneficios que aporta este al perfil de los nuevos profesionales, para que se familiaricen con el ámbito profesional mejorando sus relaciones sociales, adaptarse a una nueva infraestructura, otros métodos de formación e investigación.

Como ya se mencionó anteriormente, las universidades crean eventos donde todos pueden participar, pero promocionar dichas actividades puede resultar en gastos, por lo que suelen obviar este proceso y la participación de otras academias resulta solo cuando existen convenios. Todo esto ocurre por la inexistencia de una base de datos de conocimiento al alcance de todos.

Por último, a pesar de que todos los miembros del consejo, y decanos a nivel mundial tienen acceso a internet, no cuentan con una plataforma en la web donde puedan colaborar,

compartir sus ofertas, o simplemente comunicarse, e incluso existen miembros que no pertenecen a carreras de software, y el uso de plataformas complejas resulta ser un problema más que una nueva oportunidad.

1.2. Objetivos

1.2.1. Objetivo General

- Desarrollo de una aplicación web a través del uso de software libre para contribuir a la colaboración entre los decanos de diferentes escuelas de ingeniería.

1.2.2. Objetivos Específicos

- Establecer los tipos de recursos que se van a compartir en la plataforma.
- Diseñar un modelo de base de datos.
- Desarrollar una plataforma cooperativa.

1.3. Justificación

El desarrollo de la plataforma de gestión del GEDC es importante, ya que constituirá un repositorio de información sobre las universidades, además permitirá sus representantes colaborar entre sí, facilitando aportes en investigación y conocimiento, estos beneficios no son exclusivos para los miembros de la organización, la información será pública, para que academias vecinas, puedan informarse, y tengan la oportunidad de participar, siempre y cuando pertenezcan al área de ingeniería, y se comuniquen con los miembros de la directiva del periodo correspondiente.

Los decanos que formen parte de la directiva encargados de la reunión anual del consejo, también serán beneficiados, al contar con la plataforma, ya que podrán gestionarla sin restricciones, haciendo uso de los mantenimientos, para hacer la presentación de su institución, el itinerario y otras actividades, evitando así el desarrollo del sitio web desde cero.

1.4. Antecedentes históricos

En 2008 se inaugura una organización que pretende hacer un esfuerzo por una civilización global, el GEDC, buscando fortalecer la formación de los futuros ingenieros, por medio de la acción colectiva de las universidades, colaborando entre sí para que los estudiantes, puedan ampliar su visión laboral en los campos de ingeniería, una organización donde universidades con un alto grado de tecnología, puedan compartir sus recursos con otras, tanto estudiantes como docentes, pueden pulir sus habilidades, y participar en proyectos que sumen a su archivo profesional.

Se hace gran énfasis en la investigación e innovación para solucionar los constantes desafíos que enfrentamos en la actualidad, tales como el déficit de la calidad de agua y alimentación, la limitación que supone la energía eléctrica, la salud pública y seguridad. Cada escuela puede aportar en este esfuerzo, debido a su misión y visión única, y sus esfuerzos por mejorar la calidad del recurso humano, además de su constante actualización e inversión en tecnología para mantenerse en la vanguardia. (Statement, Engineering, & Council, 2008)

El perfil del ingeniero es importante, este debe de proyectarse a medio y largo plazo, la capacitación continua es la clave, además de contar con la infraestructura adecuada.

Doboli, Doboli y Currie (2009) describen un enfoque educativo, para que estudiantes de universidades extranjeras emulen los fundamentos de la ingeniería global, para poder desarrollar diseños de equipos complejos (hardware). (IFEES, 2015)

La colaboración entre escuelas de ingeniería es la clave para que todos los estudiantes tengan las mismas oportunidades de fortalecer sus habilidades, y cuenten con la capacidad de desenvolverse en el ámbito profesional.

La colaboración en entornos web, es un tema que llama mucho la atención, los textos relacionados a esta forma de trabajo han obtenido muy buenos resultados, por ejemplo (Xu, Du, & Fan, 2015) por medio de su estudio sobre el uso de entornos web colaborativos para el aprendizaje, reveló que el uso de dichas plataformas influye de manera positiva en los estudiantes, mejorando su retroalimentación. Tanto individual como grupalmente accedían a cursos, o buscaban ayuda dentro de estos entornos. El trabajo hace énfasis en el uso de las plataformas que fomentan el trabajo en equipo, y lo que representan para el futuro.

Sin importar su nivel de grado, estos beneficios son evidentes también en estudiantes de postgrado, tal como lo demuestran en su estudio (Castillo, Heredia, & Gallardo, 2017), quienes por medio de un programa de maestría online plantean una relación entre el nivel de desempeño de sus estudiantes en el ámbito académico y el trabajo colaborativo. Las variables con las que trabajaron estaban enfocadas en el desempeño de los estudiantes, y se quedaba registrada dentro de los foros. Entre los resultados se confirmó el aumento de

rendimiento de los estudiantes, y un gran apogeo por la colaboración. Por último, hacen un hincapié en la aplicación de estas herramientas y técnicas para el trabajo conjunto.

Se suma a esta lista otro estudio que se enfoca en estudiantes con dificultades auditivas, (Schley & Stinson, 2016) investigaron sobre el uso de otras alternativas dentro del aula de clase para que se fomente el trabajo colaborativo. Los principales participantes de este estudio fueron estudiantes con problemas auditivos de posgrado en educación docente. Estos interactuaban en un laboratorio, por medio de lenguaje de señas y chat, sin embargo, el medio más popular para trabajar fue una herramienta colaborativa, donde redactaron un documento en equipo. Los resultados son los mismos que estudios anteriormente revisados, un aumento representativo en el rendimiento de los estudiantes, en este caso con problemas auditivos gracias al uso de equipos informáticos y herramientas colaborativas, y la respectiva recomendación de aplicación de herramientas de colaboración esta vez para estudiantes con algún tipo de discapacidad.

En cuanto a las plataformas en sí, tenemos como ejemplo el trabajo de (Cardinale & Chbeir, 2015), la plataforma EDiM, que busca emular el servicio que brindan las redes sociales, pero con un valor agregado: la correcta gestión de datos. Un medio en donde el procesamiento de contenido multimedia es posible. La finalidad de este proyecto es el intercambio de multimedia en un entorno colaborativo, basado en arquitecturas que benefician a todos los miembros que interactúan en dicho entorno. La plataforma aún se encuentra en desarrollo, y se ha mostrado la efectividad de sus módulos principales.

Otra propuesta interesante expuesta por (Battaglia, Neil, De Vincenzi, & Martínez, 2016), es el uso de una herramienta CASE ya dentro de una plataforma web de colaboración, esta herramienta no está pensada para reemplazar la plataforma, sino para que como complemento, les ofrezca a los usuarios un entorno colaborativo más completo, para que tanto estudiantes como maestros se coordinen, y se ajusten mejoras en términos de evaluación y control de los trabajos que se elaboran, para la obtención de resultados de manera continua.

Lo más parecido a una plataforma colaborativa, es el principio de una red social, donde todos los usuarios comparten opiniones formando continuos debates, y no se excluye a nadie de participar.

La implementación de un tipo de red social con fines académicos no es algo nuevo, de hecho (Estrada, Cabada, & Lugo, 2018) diseñaron una Red Social de Aprendizaje colaborativo (RSAC), la cual estaba destinada a la construcción de conocimiento por medio de la colaboración entre los mismos estudiantes, como herramientas colaborativas definidas en el proyecto se encontraban las Wikis para plasmar conocimientos, foros para compartirlo, comunidades para el debate, y el respectivo chat para comunicación en privado. Estas herramientas están pensadas, para que los docentes asignen trabajos a los estudiantes, y que estos hagan uso de ellas para hacer efectiva la colaboración, y que todo quede evidenciado dentro de la red.

Siguiendo en el contexto educativo, las mejoras en la forma de evaluar a los estudiantes es algo que no se debe dejar atrás, por eso (Briede, Cabello, Pérez, & Arriagada, 2016) proponen una plataforma en donde se haga una gestión de las evidencias

proporcionadas en los talleres de Diseño Industrial en la Universidad del Bío-Bío en Chile, todo esto para gestionar y transparentar el enorme conjunto de datos e información de los proyectos. Tal cual se trabajará en este proyecto, se enfocan en el uso de herramientas de desarrollo *Open* *Source*.

Los resultados que arrojaron el uso de esta plataforma, demostraron que la correcta gestión de los datos, hace mucho más fácil los procesos de evaluación y control de los proyectos.

GEDC Exchange es la plataforma digital oficial del GEDC, sin embargo, solo trabaja como un foro para el intercambio de información para los miembros de la organización, aquí se publica la información sobre las oportunidades que brindan las universidades que tienen la intención de compartir su infraestructura u otros recursos.

LinkedIn en cambio es una plataforma más abierta, a pesar de ser una herramienta e-recruitment, cuenta con un gestor de anuncios, y como red social con perfiles tanto de quienes ofertan, y para quienes va dirigida cada publicación, se puede acceder a las publicaciones desde la pantalla “*feed*”, cada publicación cuenta con información sobre el perfil de quien realizó la publicación, un pequeño detalle donde se detalla la información relevante para los interesados, además el usuario que va a solicitar el trabajo, puede comprobar si es apto para solicitar el puesto de acuerdo a los datos de su perfil profesional, y si está interesado, puede guardar el anuncio, para visitarlo en el futuro, o hacer una solicitud para el contacto.

1.5. Marco teórico

1.5.1. Conceptos de programación

Para entrar en contexto, y tener claro que son las herramientas que vamos a utilizar en el desarrollo de nuestra aplicación debemos conocer los siguientes temas.

1.5.1.1. Entorno de desarrollo integrado (IDE)

Es una herramienta de software visual que permite la construcción de aplicaciones a partir de componentes, estas aplicaciones le ofrecen al programador una gran cantidad de elementos para que pueda trabajar entre ellos: Una o más paletas donde se muestran en forma de iconos cada uno de los componentes disponibles para el proyecto; un contenedor donde se colocan los componentes que se relacionan; editores específicos para configurar y especializar componentes; visores que ayudan a localizar los componentes de acuerdo a ciertos criterios de búsqueda; una ventana desplegable con los directorios de los componentes; acceso a editores, interpretes, compiladores y depuradores para el desarrollo de nuevos complementos; por último el acceso a herramientas de gestión y control de proyectos para grandes proyectos de software. (Fuentes & Troya, 2016)

1.5.1.2. Programación Orientada a Objetos (POO)

Metodología de programación que se refiere a transformar el mundo real en código. (Jaime & Uriarte, 2012)

La finalidad de la programación orientada a objetos es la de desarrollar aplicaciones que puedan ser reutilizadas, abiertas en cualquier ordenador, además de robustas, sus

elementos básicos, son los objetos, métodos y clases.(Barzola Mariscal, Tapia Solórzano, Falconi San Lucas, & Tobar Morán, 2017) Entre sus características se pueden definir:

Abstracción: Cada objeto sirve como un agente abstracto y se puede comunicar con otros objetos sin revelar sus características.

Encapsulamiento: Agrupa elementos de una misma entidad.

Polimorfismo: Los objetos reciben diferentes comportamientos, sin embargo, estos pueden tener el mismo nombre.

Herencia: Las clases pueden compartir sus características con otras llamadas clases hijas.

1.5.1.3. Interfaz de programación de aplicaciones (API)

Este concepto es definido por (Lujan, 1999) como un: “ Conjunto de constantes, funciones y protocolos que permiten programar aplicaciones, una API facilita la tarea de desarrollar aplicaciones, ya que otorga al programador todas las piezas para trabajar, y este se encarga de unirlos tal cual requiera su sistema.”

1.5.1.4. Framework

Este término de acuerdo a la documentación de (Gutiérrez, 2014) Se refiere a una estructura de software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación, es decir una aplicación que puede ser complementada por el desarrollador. En cambio, un *framework* web, se puede definir como el conjunto de

componentes (clases, archivos XML, etc.), de diseño reutilizable, que facilitan el desarrollo de sistemas web.

Otros autores como (Molina Ríos, Loja Mora, Zea Ordóñez, & Loaiza Sojos, 2016) definen al framework como un almacén, que vendría a ser como una estructura el cual contiene técnicas mediante la utilización de todos los elementos que sean necesarios para beneficio del ser humano

Python es una gran alternativa para el desarrollo de sitios web debido a que cuenta con un framework muy eficiente, y popular entre los programadores.

1.5.1.5. Servidor web

No hay que confundir, un servidor web no es lo mismo que un servidor. El servidor web es un software en cargado de hacer las solicitudes de los clientes de la World Wide Web, este puede contener uno o más páginas web. Se encarga del procesamiento de solicitudes de red entrantes por medio del protocolo HTTP entre otros.

Entre las funciones de un servidor web se encuentran el almacenamiento, procesamiento y entrega de páginas web a los clientes en forma de documentos HTML interpretados por el navegador.

1.5.1.6. Sistema Gestor de Base de Datos (SGBD)

Herramienta de software que se encarga de asegurar la accesibilidad, integridad y consistencia de los datos, convirtiendo la gestión y acceso a los mismos en una aplicación cerrada, haciéndose cargo de todos los problemas de mantenimiento, explotación y

comprobación de datos, colocando una pared entre los usuarios y los ficheros. La separación de los usuarios de los archivos consiste en colocar toda la carga de trabajo en el sistema, y que el usuario pueda acceder a los datos por medio de un lenguaje conceptual sencillo.

Los SGBD pueden encargarse de cantidades de datos que jamás podríamos imaginar, estas herramientas son capaces de eliminar inconsistencia de datos por medio de un control exhaustivo, permiten ahorrar espacio de almacenamiento, el acceso a los datos resulta casi inmediato, aseguran nuestros datos frente a desastres y el mal uso de los mismos.

1.5.1.7. Modelo entidad relación

Modelado de base de datos donde se forma el esqueleto de la misma, permite al programador organizar sus ideas, plasmarlas en objetos y estructurar por completo la base.

Es una técnica de ingeniería de información que se usa para el desarrollo de calidad de un modelo de datos. Este modelo ofrece un estándar para definir nuestros datos y las relaciones que estos tienen para los diferentes sistemas de información, la razón por la que se realiza recae en la mejora de la calidad de nuestro software, y su productividad. (Barker, 1994)

1.5.2. Herramientas tecnológicas

Conociendo ahora si los conceptos clave, detallamos a continuación cada una de las herramientas que utilizaremos en el desarrollo.

1.5.2.1. Hypertext Transfer Protocol (HTML)

Lenguaje de etiquetas para programación web, junto a CSS permite que una página web cuente con la estructura adecuada.

Es considerado un lenguaje de marcas de hipertexto, donde se define la estructura base para la construcción de un sitio web, este lenguaje es capaz de ser interpretado por casi cualquier navegador web en la actualidad, debido a su estándar (SGML Sistema de Marcado generalizado Estándar en español) en cualquiera de sus versiones.

Se puede reescribir las páginas escritas en HTML, brindando al usuario facilidad en su manejo, las mismas páginas pueden diseñarse usando distintos tipos de elementos, ya sean colores, listas, tablas imágenes, etc. Esto se hace por medio del uso de etiquetado que indica donde ubicar cada elemento en la página con su respectivo formato. (Barzola Mariscal et al., 2017)

1.5.2.2. Cascading Style Sheets (CSS)

Estándar para páginas escritas en HTML, publicado en W3C (*World Wide Web Consortium*). CSS está dedicado al diseño gráfico de los sitios web, permite ahorrar tiempo, da consistencia, y facilita su escritura. (Durango & Academy, 2015)

1.5.2.3. PyCharm

Es un IDE que proporciona no solo herramientas de desarrollo, sino una interfaz muy cómoda, y facilita mucho el proceso de desarrollo, con características como el autocompletado de código inteligente, resaltado de los errores de código, advertencias de

errores, y la refactorización de dichos problemas, además ofrece soporte para *Python*, *JavaScript*, *CSS*, lenguajes de plantilla entre otros.

1.5.2.4. JavaScript

Lenguaje de programación, generalmente usado en web, permite al programador desarrollar un sitio sumamente dinámico.

Con este lenguaje el programador puede crear actividades realmente complejas en su sitio web, desde mostrar contenido en tiempo real, hasta interactuar con archivos multimedia, creación de mapas de bits y gráficos en 2D, 3D entre otros.

Este lenguaje es sumamente sencillo, sus programas son denominados scripts, y son empleados en las páginas HTML ejecutadas en el navegador web. Fue desarrollado por Netscape, a partir del lenguaje Java. Java al ser un lenguaje de programación puede ejecutarse fuera de los navegadores web, lo cual marca su diferencia. (Barzola Mariscal et al., 2017)

1.5.2.5. JQuery

Es una biblioteca de JavaScript con una enorme cantidad de funciones, permite la manipulación de documentos en HTML, manejar eventos, Ajax y animación, todo esto por medio de una API, que es funcional en la mayoría de navegadores del mercado. (Álvarez, 2009)

Su arquitectura se basa en una matriz compuesta por diferentes elementos, puede agrandarse por medio de plugin que le agregan funcionalidades para la manipulación de

páginas web. JQuery puede ser muy útil, ya que cuenta con funciones únicas, a partir de su arquitectura permite a los usuarios finales el uso de extensiones, widgets y otro tipo de utilidades que ayudan a obtener información de su navegador web para hacer el respectivo control.(Sisalima Tapia & Tenelema Arias, 2014)

1.5.2.6. PostgreSQL

SGBD relacional de código abierto, este usa el lenguaje SQL y lo combina con otras características, para que se encarguen de almacenar grandes cargas de trabajo de datos, entre sus características están: la integridad de los datos, recuperación ante desastres, seguridad, extensibilidad entre otras. (The PostgreSQL Global Development Group, 1996)

PostgreSQL permite una gestión sencilla de base de datos, y trabaja muy bien acompañado del lenguaje Python.

1.5.2.7. Python

Básicamente es un lenguaje de programación. Este lenguaje cuenta con estructuras de datos de alto nivel, y a pesar de esto tiene un enfoque muy simple a POO. Permite una fuerte comprobación de tipificación en tiempo real, permitiendo así un desarrollo de aplicaciones muy rápido, en una gran cantidad de plataformas. Tanto el intérprete de Python como su biblioteca se encuentran de manera libre en sus respectivas plataformas, a dicho intérprete se le pueden agregar funcionalidades. Agregado a esto Python puede ser usado como lenguaje de extensiones para aplicaciones personalizables. (Duque, 2011)

Los beneficios de usar este lenguaje de programación radican en la simpleza del mismo, sin embargo, puede permitimos desarrollar aplicaciones complejas, este nos permite separar el software a desarrollar en módulos que a su vez pueden ser reutilizados por otros programas en el mismo lenguaje, además de que ya viene con una variedad de módulos incrustados, en caso de que necesitemos ahorrarnos tiempo de desarrollo o versatilidad en el sistema.

Entre las características definidas por (Molina Ríos et al., 2016), *Python* es:

- Legible: Evita la escritura de código ofuscado, permitiendo que este sea entendible para todos los que conozcan el lenguaje.
- Simple y poderoso: Este lenguaje soporta estructuras de alto nivel como listas, cadenas de caracteres, diccionarios, entre otros.
- Scripting: Se pueden utilizar constantes y variables sin necesidad de declararlas previamente.
- De código interoperable: El código de Python se puede usar en múltiples plataformas, como por ejemplo *Python* (donde se ejecuta *Python* desde la máquina virtual de *Java*).
- De código abierto: Permitiendo a los interesados en colaborar en el desarrollo de una herramienta de software la capacidad de acceder al código fuente.

¿Por qué elegir *Python*?

Python además de ser una herramienta de desarrollo eficiente, fácil de usar y de código abierto, cuenta con un *framework* que aprovecha cada una de sus características,

permitiendo a los programadores encontrar soluciones inmediatas dentro de su gran comunidad, y es considerado como un lenguaje de propósito general, es decir, se puede hacer todo lo que se podría hacer con cualquier otro lenguaje.

Tabla 1-1: Comparativa de lenguajes de programación.
Fuente: Elaboración propia

PHP goza de muchas características que lo convierten en uno de los lenguajes por excelencia para desarrollar aplicaciones dirigidas a la web, sin embargo, existe un complemento que le genera valor agregado a Python, y la razón principal por la que su uso es muy popular, el *framework* Django.

1.5.2.8. Django

Framework de aplicaciones web de alto nivel escrito en Python, cuyo propósito es agilizar el desarrollo de software y que este a su vez sea más limpio, además es gratis y de código abierto. (Núñez Cují & Viteri Bravo, 2017)

Django permite al desarrollador trabajar de una forma tan simple que la el proceso de creación de una aplicación se resume en la reutilización de código que ya se ha trabajado anteriormente por otros miembros de la comunidad y adaptarlo a nuestras necesidades, ahorrándonos el tiempo que tomaría desarrollarlo desde cero.

Historia de Django

Basados desde el libro de Django podemos tener una visión más clara de lo que representa este framework, es su historia. Los autores de “La guía definitiva de Django” (Holovaty & Kaplan Moss, 2015), explican que la creación de Django fue la necesidad de

automatizar el mismo proceso de desarrollo de un sitio web, reutilizando parte del código. Nace en el 2003 cuando Adrián Holovaty y Simón Willison empiezan a usar el lenguaje Python para desarrollar sus aplicaciones web. Debido al tiempo en el que tenían que publicar las noticias para la empresa periodística en la que trabajaban, estos personajes crean el *framework* para hacer el proceso de desarrollo más eficaz, y así ahorrar tiempo y recursos. Debido al éxito que representó para la empresa, tanto que casi todo se construía en base a ese framework, el equipo de desarrollo de la empresa en el año 2005 libera Django como un software de código abierto. Con las constantes actualizaciones que ha recibido, Django se ha convertido en lo que hoy conocemos, pasando por muchas versiones.

Versiones

En la Figura 1-1 se exponen todas las versiones de Django que se han ido puliendo con el tiempo desde su primera versión en el 2005, en el presente trabajo se utilizará la versión 1.7 porque es la más estable para trabajar en el IDE de *PyCharm*, y el SGBD de PostgreSQL, además de que las versiones desde la 1.5 en adelante son mucho más rápidas que las versiones previas, y contienen características muy útiles para proyectos en la web.

Figura 1-0-1: Versiones de Django.
Fuente: (Solórzano Ávila, 2018)

Arquitectura MVC

La plataforma virtual del GEDC, contará con una arquitectura modelo vista controlador, y esta es la misma arquitectura en la que se inspira Django, sin embargo, esta es denominada Modelo Vista *Template*.

Modelo: La aplicación contará con la constante sincronización entre ella y la Base de datos, para que la información que requiera el usuario pueda ser visualizada en cualquier momento.

El usuario también tendrá el poder de modificar los registros de la base de datos en cada uno de los mantenimientos.

Vista: La vista cumple la función de recibir cada una de las peticiones que se envían al navegador, puede comunicarse con el modelo para realizar la extracción de algún valor o un conjunto de dichos valores desde la BD.

Plantilla: Es el archivo que van a visualizar los usuarios finales, visitantes del sitio entre otros, esta embebido en código HTML, permite la navegación en el sitio web.

CAPITULO II

2. METODOLOGÍA Y EJECUCIÓN DEL PROYECTO

Luego de un previo estudio y revisión bibliográfica, llegamos a la conclusión de que la metodología de desarrollo Ágil Scrum se adaptaba a nuestras necesidades, y cubría exactamente el tiempo que disponíamos para desarrollar la herramienta.

2.1. Metodología de desarrollo Ágil SCRUM

SCRUM es una metodología de desarrollo de software ágil, muy usado en la actualidad, debido a su flexibilidad ante los cambios, en esta se agrupan una gran cantidad de técnicas que la convierten en una herramienta considerada como una de las más completas, y esta al mismo tiempo da la opción de implementar otras herramientas CASE, para que se adapte a las necesidades del grupo de trabajo o del proyecto en sí. Dicha metodología es muy variable, le crea posibilidades al gestor del proyecto, permitiéndole el contacto con el resto del equipo frente a cualquier cambio, sin que se pierda el sentido de la estructura.

Para este trabajo, se hizo la revisión de “La guía del SCRUM” escrita por (Schwaber & Sutherland, 2017), quienes definen que: “Scrum es un marco dentro del cual puede emplear diversos procesos. Scrum deja en claro la eficacia relativa de la gestión de su producto y las técnicas de trabajo para mejorar continuamente el producto, equipo y entorno de trabajo. (p.04)”

2.1.1. Funcionamiento del SCRUM

Scrum depende de todos sus miembros para lograr la calidad de producto deseada, esta metodología tiene éxito porque la comunicación es constante, y permite que sus miembros trabajen en cualquier momento siempre y cuando muestren evidencias, está compuesto por:

- *Product Owner* o dueño del producto: Es el encargado de comunicarse directamente con el cliente, y así obtener la información necesaria para que el *Scrum Master* haga su trabajo.
- *Scrum Master*: A partir de la información que obtiene de la conversación con el dueño del producto, modela las necesidades del cliente, y también es el encargado de guiar a los miembros del equipo de desarrollo o *development team*.
- *Product Backlog*: Conocido en español como Pila de producto, Es un listado de requerimientos que se ha creado por medio del levantamiento de información del dueño del producto.
- *Sprint Backlog*: Listado de requerimientos descompuesto en pequeñas tareas, que se deben cumplir en un plazo de una a cuatro semanas.
- *Development Team* o equipo de desarrollo: Son los encargados de cumplir con el Sprint establecido.
- *Sprint*: Corazón del Scrum, es el proceso de desarrollo (construcción de las necesidades del cliente), este se encuentra dividido en pequeños módulos funcionales, por un producto incremental. Cada que se finaliza un sprint se

debe iniciar con uno nuevo, tomando nuevos requisitos o funcionalidades del *Product Backlog*.

- *Daily Scrum*: Se refiere a las reuniones diarias que permiten dar seguimiento al Sprint, dentro de la reunión se define una retroalimentación y se consultan pequeños avances, dura aproximadamente 15 minutos.
- *Sprint Review*: Aquí es donde se corrobora que se hayan cumplido cada uno de los objetivos establecidos en el sprint, todo esto para garantizar la entrega del producto en al cliente.
- *Sprint Retrospective*: Cuando se termina un sprint, entra en juego esta etapa, aquí se analizan los resultados del sprint anterior para encontrar problemas, falencias en el proceso, o mejoras que se puedan aplicar al nuevo sprint.

El ciclo de procesos de la metodología Scrum se hace efectiva gracias al cumplimiento de cada uno de los *Sprint*, esto le permite al cliente interactuar con el producto a pesar de que no esté terminado, en la figura 2-1.

Figura 2-1: Ciclo de la metodología Scrum
Fuente: (Schwaber & Sutherland, 2017)

2.2. Factibilidad

A pesar de que no se va a implementar el software, podemos detallar un posible alcance general, en cuanto a la factibilidad desde diferentes perspectivas: técnica y operativa, además de económica. Este estudio a más de mostrar si la herramienta será factible, demostrar que no se necesita un presupuesto muy elevado para ponerlo en funcionamiento.

2.2.1. Factibilidad Técnica y operativa

Este apartado está compuesto por el recurso humano competente para el proceso de desarrollo y la aplicación de la metodología de desarrollo, el recurso de hardware utilizado, para el desarrollo y pruebas, y el recurso de software utilizado dentro de dicho hardware, a pesar de que se utilizó equipos personales, los datos corresponden a las especificaciones, mínimas y las óptimas.

2.2.1.1. Recurso humano

Como nuestra tutora formó parte de la metodología de desarrollo, es considerada en este apartado.

MSc. Jessenia Cárdenas Cobo

Rol: Product Owner

Responsabilidad: Especificación de requerimientos, comunicación directa con los miembros del GEDC (Futuros usuarios de la plataforma)

Contacto: jcardenasc@unemi.edu.ec

Dody Adonys Fischer Martínez

Rol: Desarrollador (miembro Team Development)

Responsabilidad: Desarrollo de la plataforma

Contacto: dfischerm2@unemi.edu.ec

Héctor Aaron Llerena Aguilera

Rol: Desarrollador (miembro Team Development)

Responsabilidad: Desarrollo de la plataforma

Contacto: hllerena@unemi.edu.ec

2.2.1.2. Recurso Software

Las herramientas mencionadas a continuación son totalmente gratuitas.

- Python v2.7
- Django v1.7
- PostgreSQL
- IDE PyCharm Community
- Navegador web

2.2.1.3. Recurso hardware

Los requerimientos técnicos para el desarrollo de la plataforma no son muy exigentes, para compilarlo se necesita:

En cuanto al equipo de Escritorio requerido

Para compilar el programa se puede utilizar un IDE cualquiera o simplemente un editor de código fuente como *SublimeText* o Notepad++, sin embargo, tomamos como referencia el más popular, PyCharm el cual necesita un equipo con:

- Un Sistema Operativo desde Windows 7 en adelante con arquitectura de 64 bits.

- 4 GB de RAM como mínimo y 8 GB para un rendimiento optimo
- En cuanto al procesamiento, *Python* no se desenvuelve muy bien en multihilo, por lo que un procesador de un núcleo sería suficiente, sin embargo, el IDE necesita varios núcleos para trabajar, entonces como recomendación para el procesamiento, como opción en Intel un CPU de la familia Core i de segunda generación en adelante, como referencia el más básico el Core i3 2120 que cuenta con dos núcleos de trabajo y 4 hilos, a una frecuencia 3.3Ghz por núcleo.
- Para la salida de video un monitor con una resolución superior a 1024x768.
- Un mouse y teclado para la entrada de datos.

No se tiene en cuenta equipos móviles para hacer las pruebas del diseño responsivo del sitio web, debido a que los navegadores ya cuentan con esa opción.

2.2.2. Factibilidad Económica

2.2.2.1. Hardware

Se detallan a continuación en la tabla 2-1 los costos del hardware que se necesita para el desarrollo de la aplicación, de acuerdo al detalle de la factibilidad técnica y operativa.

Tabla 2-1: Costo del Hardware.
Fuente: Elaboración propia

2.2.2.2. Software

Se detallan a continuación en la tabla 2-2 los costos del hardware que se necesita para el desarrollo de la aplicación, de acuerdo al detalle de la factibilidad técnica y operativa.

Tabla 2-2: Costo del software.
Fuente: Elaboración propia

2.2.2.3. Otros

En la tabla 2-3 se especifican otros costos que se hacen presentes en el trabajo de desarrollo.

Tabla 2-3: Otros costos.
Fuente: Elaboración propia

2.2.2.4. Valor total

Se detalla en la tabla 2-4 el valor total que representa el desarrollo en 3 meses.

Tabla 2-4: Valor total del desarrollo
Fuente: Elaboración propia

CAPITULO III

3. RESULTADOS

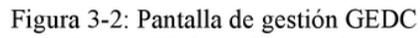
3.1. Manual de usuario

3.1.1. Acceso al sistema

El ingreso a la plataforma de gestión web del GEDC, será posible desde el siguiente enlace temporalmente: “<http://gedc.2temp.com/seguridad/login/?next=/panel/>”, en esta parte el usuario debe autenticarse, siempre y cuando esté registrado.

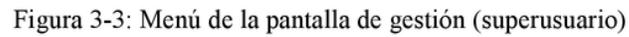

Figura 3-1: Pantalla de inicio de sesión
Fuente: Elaboración propia

Una vez se hayan ingresado los datos de autenticación, tendremos acceso a la pantalla de la Figura 3-2.


Figura 3-2: Pantalla de gestión GEDC
Fuente: Elaboración propia

Del lado izquierdo de la plataforma podemos observar un pequeño menú desde el cual se podrá acceder a diferentes módulos del sistema.

Dependiendo de los roles de usuario, existirán variaciones, como restricciones para quienes no sean superusuario(presidente(a), secretario(a) ejecutivo(a), o secretario(a)).


Figura 3-3: Menú de la pantalla de gestión (superusuario)
Fuente: Elaboración propia

3.1.2. Gestión de actividades

El apartado de gestión de actividades tal como se muestra en la figura 3-4, permite al superusuario gestionar las actividades de todos los miembros, mientras que a los usuarios regulares solo las propias.

 Figura 3-4: Pantalla de gestión de actividades
Fuente: Elaboración propia

Desde el botón de “acción” se puede realizar la visualización de los eventos, editarlos, o darlos de baja, en este último se puede anexar un documento que certifique la finalización del evento. Podemos publicar una nueva actividad, desde el botón de “Registro nuevo” de la figura 3-4.

Figura 3-5: Parte del formulario de Actividad
Fuente: Elaboración propia

Dentro del formulario de actividades que se muestra en la figura 3-5, se puede agregar un documento formal, o un fichero .PDF que muestre todos los detalles del evento.

La actividad puede publicarse como “Activo” que denotará que los usuarios se puedan inscribir. En caso de que esta casilla no se active la actividad se tomará como culminada, o como un evento pasado.

La actividad se puede marcar como “Publica” o no, esto solo afectará la visibilidad del evento en la página de presentación, siempre y cuando esté activa.

PROPUESTA DE SOLUCIÓN

3.2. Tema

Desarrollo de una aplicación web a través del uso de software libre para contribuir a la colaboración entre los decanos de diferentes escuelas de ingeniería.

3.3. Descripción de la propuesta de solución

Se plantea crear una plataforma de colaboración entre decanos, donde los miembros sean capaces de realizar una constante actualización de información relevante sobre eventos dentro de su organización. Los decanos de las escuelas de ingeniería podrán mantenerse al tanto de cada uno de estos eventos con el acceso a la plataforma, sin necesidad de que acudan a las reuniones que se organizan anualmente por el GEDC. Esto con el fin de mejorar la calidad académica y hacer efectiva la colaboración entre escuelas, por medio de la interacción del alumnado con una infraestructura diferente, acercándose a otras universidades a recibir charlas, lo mismo con el material docente. También se abre la posibilidad de compartir conferencias de alto nivel e incluso de postgrado.

3.4. Especificación de requerimientos (Especificaciones técnicas)

3.4.1. Requerimientos funcionales

Antes de empezar, para la gestión de los diferentes módulos se usará el acrónimo CRUD que corresponde a las iniciales de *Create*, *Read*, *Update*, *Delete* que al español significan Crear, Leer, Modificar, Eliminar correspondientemente.

En primer lugar, se establece el diseño tanto de la página de presentación como la página de gestión.

El sistema va a estar compuesto por los siguientes módulos:

- **Módulo de usuarios:** Gestión de los usuarios registrados.
 - El usuario administrador puede realizar el CRUD del registro de usuarios.
 - Respecto a la creación del usuario, posterior a la solicitud de un registro: Se debe de ingresar una foto de 350x350p, un nombre de usuario, contraseña, nombres y apellidos completos, nivel de acceso, fecha de registro, datos de contacto (telf., telf. Oficina, email, país y universidad a la que representa, títulos obtenidos, otro tipo de experiencia adquirida.
 - El usuario tendrá acceso al listado de los datos de todos los miembros que sean o hayan sido registrados en la plataforma, este listado estará comprendido por el nombre de usuario, la foto, los nombres completos, la universidad a la que pertenece y su respectivo país, fecha de incorporación y de último ingreso a la plataforma, podrá verificar si aún es miembro y su respectivo nivel de acceso.
 - Todos estos datos pueden ser modificados o eliminados.
- **Módulo de carreras:** Gestión de las carreras universitarias.

- El usuario administrador puede realizar el CRUD de las carreras:
 - Para la creación de una nueva carrera, deberá ingresar un código para identificar la carrera y el nombre respectivo, por ejemplo: Ingeniería en Sistemas Computacionales.
 - Podrá acceder al listado de las carreras, el listado comprende el código, el nombre y el estado de la carrera.
 - Todos estos datos pueden ser modificados o eliminados.
- **Módulo de país:** Gestión de países.
 - El administrador puede realizar el CRUD de país
 - Para agregar un país se debe detallar su nombre, código (estándar de iniciales), y su código telefónico.
 - El listado comprende el nombre, código, código telefónico y estado.
 - Todos estos datos pueden ser modificados o eliminados.
- **Módulo Universidad:** Gestión de Universidades:
 - El administrador puede realizar el CRUD de universidades
 - Para agregar una universidad se debe subir un logo que la identifique, un código, nombre en siglas, descripción que detalle el nombre completo de la institución, dirección, país, longitud y latitud, un teléfono de contacto y el e-mail.
 - El listado comprende el logo, siglas, nombre completo, dirección, país, estado.

- Todos estos datos pueden ser modificados o eliminados.
- **Módulo Títulos Académicos:** Gestión de títulos académicos.
 - El administrador puede realizar el CRUD de títulos Académicos:
 - Para agregar un nuevo título se debe ingresar el código, la abreviatura y la descripción.
 - El listado comprende el código, abreviatura, la descripción y estado.
 - Todos estos datos pueden ser modificados o eliminados.
- **Módulo de auditoría:** Visualización de los cambios realizados por los usuarios.
 - El usuario tendrá acceso al listado de actividades de todos los usuarios dentro del sitio, por ejemplo: El registro de un nuevo usuario. El administrador podrá comprobar, la hora en la que se registró a ese nuevo usuario, desde que otro modulo se hizo este registro, el nombre del registro, el usuario que realizó dicho registro de usuario, a que organización o universidad pertenece, y que tipo de acción realizó (agregar, eliminar, o modificar).
- **Historial de directivos:** Este módulo presenta al administrador, los cambios hechos por la directiva a los diferentes tipos de usuario, permite controlar los niveles de acceso, y mantener un control de los nuevos registros de usuario, aquí se puede visualizar la fecha de los cambios realizados, a que usuario fue dirigido dicha modificación, la universidad a la que pertenece, el nivel de

acceso previo y posterior, y el nombre del usuario que hizo dicha modificación.

- **Módulo de galería:** Gestión de las fotografías publicadas en la página de presentación.
 - El administrador puede realizar el CRUD de la galería
 - Para agregar fotografías a la galería se debe de subir dicha foto, y algún detalle o información sobre la actividad que se realizó, o personas involucradas.
 - El listado comprende la foto, el detalle y estado.
 - Todos estos datos pueden ser modificados o eliminados.
- **Módulo Tipo de Actividad:** Gestión de los tipos de actividades que van a realizar las universidades con el fin de colaborar.
 - El administrador puede realizar el CRUD de Tipo de Actividad:
 - Para agregar un nuevo tipo de actividad, se debe de ingresar un código, un icono que la identifique, un nombre, y una pequeña descripción que pueda explicar en pocas palabras de que tratará dicho tipo de actividad.
 - El listado comprende el código, el icono, el nombre, la descripción y el estado.
 - Todos estos datos pueden ser modificados o eliminados.
- **Módulo de Actividades guardadas:** Corresponde a la visualización de las actividades en las que el usuario estuvo interesado.

- El usuario tiene acceso al listado de actividades en las que ha interactuado y ha enviado una solicitud, el listado corresponde al nombre de la actividad, tipo de actividad, universidad y país que la oferta, nombre del usuario que la creó, estado del evento, y confirmación de participación.
- **Módulo de actividad:** Corresponde a las actividades ofertadas por los miembros del GEDC.
 - El usuario tiene puede realizar el CRUD de la actividad
 - Para crear una actividad se debe de ingresar el tipo de actividad, un nombre o título que la identifique, un detalle que explique a cabalidad los beneficios de la participación, cupos disponibles, fecha del evento, fecha de caducidad, una imagen informativa de 690x560p, si se ha hecho un detalle más elaborado, subirlo en formato PDF, tendrá dos estados activos, y público en caso de que quieran mostrarlo en la página de presentación.

Los datos ingresados en cada módulo deberán de ser validados correspondiendo de su tipo y veracidad.

Características de los usuarios

Ya que el sistema contará con una variedad de roles de usuario, a continuación, se va a detallar el papel que cumple cada uno dentro del sistema.

Un detalle que cabe recalcar es que los usuarios serán notificados por correo electrónico, en caso de que se hagan cambios en su cuenta.

Previo a esto necesitamos crear los usuarios.

 Figura 3-6: Registro de usuario
Fuente: Elaboración propia

 Figura 3-7: Asignación de roles
Fuente: Elaboración propia

 Figura 3-8: CRUD y usuarios que interactúan
Fuente: Elaboración propia

Secretario ejecutivo, presidente (Administrador): Es el usuario único que tiene la autorización de realizar la gestión y administración completa del sitio, este tipo de usuario puede trabajar con los siguientes módulos:

 Figura 3-9: Diagrama de caso de uso Administrador
Fuente: Elaboración propia

Secretario: Este usuario tendrá un nivel de acceso que se asemeja al del presidente sin embargo contará con cambios en ciertos aspectos, no podrá cambiar los roles luego de que estos sean asignados. Este usuario tendrá múltiples accesos, sin embargo, en módulos como el de galería solo tendrá acceso de lectura.

 Figura 3-10: Diagrama de caso de uso secretario(a)
Fuente: Elaboración propia

Decano y Miembro directivo: Este usuario tiene los niveles de acceso más bajo, y puede acceder a un número limitado de módulos para su gestión, tendrá acceso a:

Figura 3-11: Diagrama de caso de uso Decano(a).
Fuente: Elaboración propia

3.4.2. Requerimientos no funcionales

3.4.2.1. Interfaz de usuario

La plataforma tendrá el nombre de GEDC LATAM EXCHANGE, la estructura de la interfaz estará compuesta por una página de gestión y otra de presentación.

3.4.2.2. Interfaz de hardware

El usuario debe de contar con el equipo de hardware ya sean ordenadores, teléfonos inteligentes, tabletas, con tarjetas de red integradas, que les permitan conectarse a internet.

Este se debe implementar en un servidor dedicado sobre la infraestructura del área de TICS de la UNEMI para sus pruebas respectivas.

3.4.2.3. Interfaz de software

La plataforma puede ser visitada, por todos los usuarios, a pesar de no estar registrados, por medio de un navegador web.

El sitio es responsivo por lo que puede ser visualizado desde cualquier dispositivo que soporte el uso de un navegador web.

3.4.2.4. Interfaz de comunicaciones

La comunicación cliente/servidor se va a efectuar por medio del protocolo HTTP, y las peticiones entre cliente/servidor y viceversa por medio del protocolo TCP/IP.

3.5. Restricciones

Toda persona que desee tener acceso a la aplicación y disfrutar de sus servicios, necesitará de conexión a internet, un navegador web, además del hardware que cuente con la tecnología necesaria para acceder al navegador.

Como usuarios deben de estar registrados en el sitio.

Opcional: Contar con el software para manipular el código fuente entre ellos el IDE, y el SGBD.

3.6. Cronograma de actividades

Presentación de la planificación de desarrollo del proyecto y de la documentación respectiva, siguiendo la metodología Scrum.

Figura 3-12: Cronograma de actividades.
Fuente: Elaboración propia

3.7. Aplicación de la metodología (Product Backlog)

Aplicando los conceptos de la metodología de desarrollo Ágil Scrum, se desarrolló la respectiva pila del producto (*Product backlog*) de nuestra herramienta de colaboración web, como se muestra en la Figura 3-13, se definieron cinco *sprint* que comprendían

periodos de trabajo desde 16 a 56 horas semanales, de acuerdo a los requerimientos, la primera semana fue el periodo con mayor carga de trabajo debido a la implementación de la capa de presentación de la plantilla de gestión del sitio, login, y los módulos necesarios para crear a los usuarios.

Las iteraciones semanales fueron definidas en horas, debido a que la evidencia de desarrollo era más contundente al ser plasmada en tiempo, y no en puntos.

Es evidente la reducción de tiempo en el desarrollo de los módulos después del primer sprint, debido a la reutilización de código, tanto para definir las vistas como las plantillas.

Figura 3-13: Product Backlog

Fuente: Elaboración propia

3.7.1. Sprint 1: Seguridad

Luego de descomponer los requerimientos funcionales, de manera que se pueda desarrollar la plataforma en cierto periodo de tiempo, por lo que se dividieron los módulos de manera en que pudieran desarrollarse más de uno por sprint.

Como se muestra en la Figura 3-14, para el sprint 1, se desarrollaron los módulos de país, universidad, carrera, título ya que eran necesarios para la creación de un nuevo usuario y además se creó dicho módulo, sin embargo, no se culminaron las plantillas ni las vistas necesarias, que quedaron pendientes para el sprint 2.

“**Requisito**” representa el requisito funcional descompuesto de los módulos que ya se habían establecido en las reuniones.

“**ID**” es el número de la actividad.

“**Estado**” representa el estado de la actividad (L) Listo, (P) Pendiente, (D) en desarrollo.

“**Tarea**” descripción de lo que se estaba desarrollando

“**Componente**” archivo que se modificaba en el IDE.

Figura 3-14: Sprint 1: Seguridad
Fuente: Elaboración propia

3.7.2. Sprint 2: Mantenimientos

Durante este sprint, se añaden dos nuevos módulos: auditoría e histórico, y adicional a esto, se terminan de desarrollar vistas y plantillas referentes a los perfiles de usuario y el perfil personal, además de incluir plantillas de los módulos definidos en este sprint, se puede ver esto más a detalle dentro de la Figura 3-15.

Figura 3-15: Sprint 2 Mantenimientos
Fuente: Elaboración propia

3.7.3. Sprint 3: Gestión

Este es el último sprint de desarrollo en la que se asignó la creación de los módulos restantes para la gestión. En este caso nos encargamos del desarrollo del tipo de actividad, y la actividad en sí, el MVC completo, en la figura 3-16 se muestra más a detalle las tareas definidas para este sprint.

Adicional a esto, para plantillas se definió la pantalla de listado y de gestión y una adicional para mostrar las actividades en las que un usuario estuvo interesado, es decir en las que interactúa con la actividad por medio del botón “Participar”.

Figura 3-16: Sprint 3 Gestión
Fuente: Elaboración propia

3.7.4. Sprint 4: Pruebas y validación

Este sprint se enfocó en crear las plantillas, y las instrucciones para evitar que el usuario final se encuentre con errores del sistema, o que inserte datos erróneos en específico. Otra parte en específico es la del envío de correos electrónicos que se muestran a los usuarios ya sea por cambio de contraseña, notificación para el cambio de estado de una actividad, la publicación de una nueva actividad, registro de usuario por parte de un superusuario, y registro de usuario por medio de una solicitud. Por último, se añadió la plantilla para la página principal que ya estaba definida, sin embargo las actividades como tal en el sprint eran la comunicación de dichas plantillas con las vistas.

Figura 3-17: Sprint 4 Pruebas y validación
Fuente: Elaboración propia

Entre las validaciones que destacan para este sprint fue el registro exclusivo de las casillas, ya sea para un tipo de carácter en específico.

3.7.4.1. Caso de prueba 1

Tabla 3-1: Caso de prueba 1 “Ingreso de correo electrónico
Fuente: Elaboración propia

3.7.4.2. Caso de prueba 2

Tabla 3-2: Caso de prueba 2 "Ingreso de número telefónico"
Fuente: Elaboración propia

3.7.4.3. Caso de prueba 3

Tabla 3-3: Caso de prueba 3 "Registro de eventos"
Fuente: Elaboración propia

4. CONCLUSIONES

Se desarrolló entorno web de colaboración del GEDC, el resultado, tal como se muestra en el manual de usuario es una plataforma simple, donde los usuarios pueden trabajar sin sentirse ofuscados, el sitio cuenta con tiempos de carga muy cortos, el sitio cuenta con una enorme cantidad de validaciones detrás por lo que se considera robusto, ya resulta muy poco susceptible a errores y caídas. Cabe destacar que el código es fácil de entender, lo que permitirá a futuros desarrolladores, leerlo y modificarlo sin tener problemas.

Luego de cada sesión se establecían requerimientos que se iban adaptando al software en desarrollo, pero existía una finalidad para este “la colaboración”, por lo que era muy importante definir las actividades que se iban a realizar, y que clase de recursos se iban a compartir desde el sitio. Una de las actividades clave fue la de ofrecer la facilidad de transportar tanto a docentes como estudiantes de una institución a otra para fortalecer su perfil académico, a esta se la denominó “Movilidad estudiantil o docente” dependiendo el caso.

Otra actividad, y no menos importante es la de “Conferencias” donde se establecen las fechas para nuevas reuniones de los miembros en el futuro.

Nuestra base de datos se alimenta de toda la información que se ingresa en la página de gestión, ya sea el registro eventos o de usuarios gracias a las tablas de auditoría, y quienes hacen estos posible son los mismos miembros interactuando entre sí. Toda esta información podrá ser procesada y analizada por los directivos más altos del GEDC, ya sean el presidente o el secretario ejecutivo.

Se creó una plataforma adaptable, rápida, simple de usar por medio de la metodología de desarrollo ágil Scrum, donde se definió desde el Product Backlog cuatro sprint para el desarrollo, el primero se centró en la seguridad y registros de usuarios, el segundo constaba de mantenimientos y auditorías para usuarios, sin dejar atrás otros módulos, aunque no se mostrarían visibles, el último sprint de desarrollo fue el tercero, que fue enfocado a la creación de las actividades y sus tipos, para finalizar se realizaron las pruebas respectivas y se crearon las validaciones para que el sitio web.

Si bien existe mucha información relevante en internet sobre plataformas colaborativas, todas se limitan al entorno académico, ya sea para mejorar la forma en la que se evalúan a los estudiantes, o simplemente contar con evidencias de los trabajos que se plantean, sin embargo, no existe una plataforma como tal en la que diferentes directivos de escuelas se comuniquen para compartir sus recursos, y conocimiento.

Tesis pregrado GEDC

INFORME DE ORIGINALIDAD

2%

INDICE DE SIMILITUD

1%

FUENTES DE
INTERNET

0%

PUBLICACIONES

2%

TRABAJOS DEL
ESTUDIANTE

ENCONTRAR COINCIDENCIAS CON TODAS LAS FUENTES (SOLO SE IMPRIMIRÁ LA FUENTE SELECCIONADA)

1%

★ Submitted to Universidad Católica de Santa María

Trabajo del estudiante

Excluir citas

Activo

Excluir coincidencias

< 15 words

Excluir bibliografía

Activo