

**UNIVERSIDAD DE MATANZAS
FACULTAD DE CIENCIAS TÉCNICAS
DEPARTAMENTO DE INFORMÁTICA**



Proyecto de curso

ACTIVATE, SISTEMA PARA LA DIFUSION DE EVENTOS

Autor:

Alejandro Yanes De la Cruz

Tutor: Jose Enrique Diaz Ramos

Matanzas, 2019

Agradecimientos

A mi familia, por esta a mi lado estos 5 años, en especial a mi papa por soportar que lo utilizara para explicarme yo mismo cada vez que tenía un problema, incluso si él no sabía de qué yo estaba hablando, también por haberme acercado a las computadoras cuando solo tenía 9 años, que, aunque ninguno de los dos lo supimos en ese entonces, fue el inicio de mi atracción por las computadoras.

A mi hermano, por siempre darme una opinión sincera, por hacerme ver mis errores, pero al mismo tiempo otro punto de vista para sacarme de ellos, por siempre impulsarme a no hacer algo mal si sabía que lo podía hacer mejor.

A mi mama, por apoyarme en todas mis decisiones, aun cuando decidí alejarme de ella.

A los grandes amigos que siempre estuvieron cerca, y a los que ya no lo están. A los que me enseñaron que siempre hay algo nuevo que aprender. A los que me enseñaron que la vida es más que trabajo.

A todos los que me tienen como un amigo.

Declaración de Autoridad

Yo, Alejandro Yanes De la Cruz, declaro que soy el único autor de este trabajo de diploma y lo pongo a disposición de la Universidad de Matanzas, Sede "Camilo Cienfuegos", para hacer uso del mismo con el objetivo y finalidad que se estime conveniente.

Firma

RESUMEN

La presente investigación tiene como título: “Actívate, sistema para la difusión de eventos universitarios”. Se realizó en la Universidad de Matanzas, la sede Camilo Cienfuegos, para el Departamento de Comunicación Institucional. Esta investigación surge a partir de la necesidad planteada por el departamento de tener un nuevo medio de difusión para los eventos universitarios, uno que pueda llegar a los estudiantes en una forma más directa, que sea atractivo, acorde con los medios actuales de información, de forma que les sea fácil acceder a la información y poder interactuar con los eventos, ya sea de forma directa con la participación o a través de comentarios. Se llevo a cabo una revisión de la bibliografía existente, así como las posibles herramientas a emplear, de las cuales se escogieron Angular, Express.js y MongoDB para conformar la arquitectura deseada. Se desarrolló un sistema compuesto por dos aplicaciones web para la gestión y difusión de la información referente a los eventos y una aplicación para móviles a través de la cual también se consumir la información.

Abstract

This research is titled: “Activate, sistema para la difusión de eventos universitarios”. It was developed in the Matanzas University, in its Camilo Cienfuegos site, for the Department of Institutional Communication. This research came up from the need stated by the department for having a new broadcast media for the university events, one that can get to the students in a way that is easy for them to access the information and interact with the events, whether is by participating or through comments. A review was carried out about the existing literature, so as the possible tools to use, from which Angular, Express.js and MongoDB where selected to conform the desired architecture. A sistem was developed composed of two web applications for management and diffusion on events and a mobile app to consume the information as well.

TABLA DE CONTENIDOS

CONTENIDO

Introducción	5
Capítulo 1: Marco teórico referencial	9
<i>Marketing Digital</i>	9
<i>Email Marketing</i>	10
Marketing de Redes Sociales	11
<i>Marketing Social</i>	11
Descripción de la Entidad.....	12
Tecnologías empleadas en el desarrollo de la solución.....	13
SCRUM	13
BPMN.....	14
Modelo Vista Controlador	14
JavaScript.....	15
Node.js	15
Angular.....	16
Express.js.....	17
MongoDB	17
Mongoose.....	18
Nginx.....	18
Docker.....	19
Conclusiones parciales.....	22
Capitulo 2: Descripción de la solución propuesta	23
2.1 Propuesta de la aplicación.....	23
2.2 Modelo de Proceso.....	24

2.3 Pila del Producto (Product Backlog)	25
2.4 Planeación de las Entregas (Sprints).....	26
2.5 Descripción de las Entregas	27
2.5.1 Diseño de la Base de Datos	27
2.5.2 Arquitectura inicial	27
2.5.3 Funcionalidad Categorías	27
2.5.4 Funcionalidad Eventos.....	28
2.5.5 Funcionalidad Usuarios	28
2.5.6 Autenticación	29
2.5.7 Crear la aplicación móvil.....	30
2.6 Definición del Equipo.....	30
Conclusiones parciales.....	31
Capítulo 3: Resultados y discusión.....	36
3.1 Descripción de la solución	36
3.2 Pruebas de <i>Software</i>	43
3.2.1 Pruebas de Caja Blanca	43
3.2.2 Pruebas de caja negra.....	50
Pruebas de sistemas web.....	54
3.3 Interfaces de usuario	37
Algoritmos principales	39
Estimación de costo	31
Conclusiones parciales.....	56
Conclusiones.....	57
Recomendaciones	58
Bibliografía	59

INTRODUCCIÓN

La World Wide Web (www) o Internet, como también se le conoce, es la red más expandida por la humanidad, permite a los usuarios estar en contacto con los seres queridos, los ayuda a conocer a nuevas personas y hacer amigos, encontrar nuevos compañeros de negocios, clientes, socializar y mucho más.

Cada vez hay un mayor número de usuarios de internet en el mundo, tanto así que ya superan la mitad de la población global. Eso se resalta en la nueva edición del informe que presentan *We Are Social y Hootsuite* en este 2019, y que año a año muestra las estadísticas, análisis y principales tendencias acerca del número de usuarios de internet, el uso de la telefonía móvil, las redes sociales, así como del *eCommerce* a nivel global.

Mientras que en su informe de 2018 el estudio señalaba un número de usuarios de Internet de 4.021 millones, es decir, el 53% de la población mundial, el informe 2019 asegura que hasta el momento el mundo cuenta con 4.388 millones de internautas, una penetración del 57%.

La evolución en el número de usuarios de Internet en el mundo es notable. En 2014 la cifra total apenas alcanzaba los 2.485 millones de internautas, mientras que para 2015 esta cifra creció un 21%, alcanzando los 3.008 millones.

A partir del ascenso del uso de internet surge un nuevo entorno en el que los usuarios pueden comunicarse, compartir información, puntos de vista, gustos, intereses, metas. Dentro de este entorno las redes sociales han tomado una gran importancia, hasta el punto en que no son capaces de separar sus interacciones con estas de sus rutinas diarias.

Las redes sociales han evolucionado junto al Internet, comenzando por solo permitir a los usuarios publicar fotos de ellos mismos y los lugares que habían visitado, hasta conversar, comentar sobre las fotos de otros, compartir ideas e información, permitiendo acordar y desacordar, e incluso reunir firmas para distintos tipos de peticiones. Cada vez es mayor el número de fotos, videos, comentarios subidos a estas redes sociales. Existen personas que hacen de las redes sociales su trabajo, ejemplos de estos son los youtubers, personas que tienen uno o varios canales en *YouTube*, y se dedican a subir videos de temas variados y cobran una cantidad por cada vista obtenida.

Este ascenso vertiginoso del internet como fuente principal de la información que crean, comparten y consumen los usuarios en el día a día, es lo que paso a la idea de llevar también el *marketing* a esta plataforma, aunque con conceptos e ideas nuevas que permitan acercar a los usuarios y clientes potenciales.

Cada día un mayor número de usuarios de internet prefieren navegar desde sus dispositivos móviles. Este 2019 el total de usuarios de internet activos vía móvil es de 3.986 millones, esto quiere decir que el 52% de la población accede a sus contenidos favoritos y navega por medio de su *smartphone*.

Es así entonces que surge el *marketing* móvil o *mobile marketing*, pues ahora el usuario puede llevar un soporte en el bolsillo más liviano y completo, funcional y adaptable a los requerimientos personales. Cambia la concepción de consumir información, ya que al mismo tiempo se comparte y se produce, facilitando la vida. El teléfono pasa de ser un dispositivo de voz en la comunicación a ser el soporte más interactivo como herramienta íntima funcional a través de la estructura móvil.

Existen muchas otras formas de *marketing* digital, está el *email marketing*, que se trata de mandar correos a los usuarios con noticias o informaciones de su interés, ya sea la existencia de nuevos productos, ofertas o artículos donde se hable de temas a fines a la organización. El *social media marketing*, que utiliza las redes sociales como Facebook, Twitter o Instagram para llegar a miles de usuarios de todo el mundo y mostrarles no solo sus productos sino la idea de que la empresa es más que un negocio, que se preocupan por sus usuarios, que tienen un interés real por su satisfacción y bienestar. Otra forma de marketing digital muy utilizada hoy son los ads o anuncios de Facebook o Google, estos son pequeños carteles en sitios que no necesariamente son de ventas en sí, sino que son blogs que hablan de las características del producto, su utilidad.

Las universidades cubanas se consideran entidades educacionales y sociales, por lo que sus objetivos para con los estudiantes van más allá de su formación como futuros profesionales, también se busca su formación como seres sociales que puedan contribuir a la sociedad de maneras intangibles, influir en su futuro y el de la comunidad. Para ello se desarrollan una serie de eventos y actividades científicas, culturales y sociales, con el fin de formar una conciencia que les permita entender su capacidad para influir a otras personas, ya sea a través de obras teatrales donde expresen su forma de ver la actualidad, o investigaciones destinadas a resolver problemas reales de la comunidad y la ciudad o el medio ambiente.

Con este objetivo todas las universidades cubanas cuentan con un departamento encargado de la gestión y difusión de la información relacionada con todos los eventos a realizarse dentro de las mismas.

El Departamento de Comunicación Social de la Universidad de Matanzas es el encargado de dar a conocer los eventos que se desarrollan en la institución en el transcurso del curso escolar. Estos eventos forman parte del proceso de formación de una cultura general e integral desarrollado por las instituciones educativas cubanas y son ideados y desarrollados con los estudiantes como foco y fuente principal de participación.

En el caso de la universidad, existen los festivales de aficionados, los foros investigativos, los proyectos orientados a rescatar la historia local y su promoción, entre otros.

Todos estos eventos se desarrollan a la par del curso escolar, sin embargo, la participación de los estudiantes ha ido disminuyendo a través de los años y con ella su razón de ser. Debido a esto se desea desarrollar una nueva forma de llegar a los estudiantes a través de las tecnologías actuales de la información y así motivar su participación.

Problema científico:

¿Qué hacer para mejorar la difusión de eventos universitarios en la Universidad de Matanzas?

Hipótesis:

Si se desarrolla el sistema informático, debe mejorar la difusión de los eventos universitarios.

Justificación:

De esta investigación se espera obtener un sistema informático que permita comunicar a los estudiantes de manera más rápida y cercana la información de los eventos universitarios a medidas que estos se van creando y desarrollando, permitiendo así que se involucren en mayor medida en la vida universitaria extra curricular.

Objetivo General:

Desarrollar un sistema informático que permita llegar a los estudiantes de manera más directa y moderna empleando los medios actuales de comunicación.

Objetivos específicos:

- Realizar una revisión bibliográfica que establezca el estado del arte y la fundamentación teórica de la investigación.
- Diseñar un sistema que permita la gestión de la información referente a los eventos universitarios, así como la interacción con los estudiantes a través de un sitio web o aplicación móvil.
- Validar la solución obtenida mediante pruebas de software.

Atendiendo a lo planteado anteriormente, la tesis queda estructurada en introducción, tres capítulos, conclusiones, recomendaciones y referencias bibliográficas, según sigue:

- Una Introducción, donde se caracteriza la situación problemática y se fundamenta el problema científico a resolver
- Un primer capítulo donde se recoge el marco teórico referencial del tema y los principales conceptos que constituyen la base teórica de la investigación, así como el análisis de las herramientas utilizadas en el desarrollo.
- Un capítulo segundo donde se describe el desarrollo e implementación del software a través de la metodología SCRUM.
- Un tercer capítulo donde se muestran las principales interfaces del prototipo inicial y se detallan los resultados de las pruebas funcionales aplicadas al software.
- Un apartado de conclusiones donde se verifica el cumplimiento de los objetivos trazados al inicio de la investigación.
- Las recomendaciones en la cual se plasman una serie de propuestas encaminadas a la continuidad de esta investigación.
- Y las referencias de la bibliografía citada.

CAPÍTULO 1: MARCO TEÓRICO REFERENCIAL

En este capítulo se realiza la fundamentación teórica del trabajo, se explican los conceptos básicos del marketing digital y se proporciona una breve descripción de los tipos más importantes. Además, se realiza un estudio de algunas de las tendencias tecnológicas actuales y se hace un pequeño análisis de las herramientas utilizadas para el desarrollo del sistema Actívate.

Marketing Digital

EL marketing digital es el marketing de productos o servicios usando tecnologías digitales, principalmente el internet, pero también se incluyen los teléfonos inteligentes y cualquier otro medio digital. El desarrollo del marketing digital desde los años 1990s y 2000 ha cambiado la forma en que las compañías y negocios usan la tecnología para el marketing. A medida que las plataformas digitales son incorporadas a la vida diaria y las personas utilizan dispositivos digitales en vez de visitar tiendas reales, las campañas de marketing digital se han vuelto más eficientes. (multiples, 2018)

Métodos de marketing digital como Search Engine Optimization (Optimización para Motores de Búsqueda o SEO por sus siglas en ingles), marketing de contenido, marketing enfocado a los datos, e-commerce marketing, marketing de redes sociales se están volviendo más comunes en el avance de las tecnologías.

Mientras el *marketing* de la década del noventa se centraba en captar cuota de mercado, preferiblemente de los competidores, en la actualidad esta circunstancia prácticamente queda relegada a un segundo plano, ya que el protagonista real es el propio cliente, así como la libertad y agilidad con la que afronta sus decisiones.

Esto constituye una de las principales características que el marketing digital exige en sus planteamientos y se resume en el hecho de que las conversaciones deben tener un carácter bidireccional, para que los clientes puedan responder a los mensajes de las compañías no solo en el acto de la compra, sino a lo largo de un proceso de relación que va más allá, suponiendo el establecimiento de relaciones de largo plazo basadas en este diálogo continuo. (Restrepo Jimenez, et al., 2016)

Por consiguiente, el objetivo fundamental del nuevo *marketing* se basa en la construcción de relaciones duraderas basadas en la confianza con los usuarios finales con la preponderancia

de tener en cuenta: el nivel racional y el nivel emocional. En lo racional, se obvia el hecho de que la oferta proporcione valor añadido mediante los productos y servicios que se ofrecen, y a nivel emocional, generando mayor interactividad con los usuarios mediante la comunicación bidireccional que servirá para fortalecer la relación al establecer un diálogo con el mercado.

Entre las ventajas que ofrece el marketing digital están:

- Su alcance está creciendo cada vez más debido a que el acceso que tienen las personas a internet y las plataformas digitales cada vez es mayor y mejor. (Content, 2018)
- Se centra en el usuario del producto o servicio a promocionar, en conocer y saber de sus intereses y necesidades. (Content, 2018)
- Gran flexibilidad y dinamismo (InboundCycle, 2018), con la posibilidad de realizar cambios sobre la marcha en función de la retroalimentación obtenida y el comportamiento de los usuarios respecto al contenido.
- Medición, cuando se realiza una estrategia de *marketing* digital puede ser medida mucho más fácilmente que las estrategias de *marketing* tradicional. (Llano, 2018)
- Crea comunidad, el *marketing* digital permite crear una comunidad que interactúa entre sí y con la organización, creando un enlace emocional entre esta y sus usuarios. (Llano, 2018)

Email Marketing

El *email marketing* define el envío de correos electrónicos a una base de datos o contactos, como lo pueden ser tus clientes potenciales o prospectos. Se considera una estrategia de comunicación digital y entra dentro de una de las acciones del marketing directo. (2014)

El *mailing* se enfoca en mandar un mensaje con la finalidad de adquirir nuevos clientes, desarrollar la relación con los actuales, crear lealtad, interactuar con los contactos, aumentar ventas, generar confianza hacia un servicio o producto, confirmar una orden de compra, entre otros.

Es una de las técnicas del marketing directo de más bajo coste y el mail que se envía recibe el nombre de boletín de novedades o newsletter. Éstos generan más presencia y recuerdan a los visitantes la existencia de los productos, así como de la empresa, consiguiendo de este modo aumentar ventas a través de ellos.

De acuerdo al uso que se les dé, estas son algunas formas en las que se pueden catalogar los correos electrónicos: (2014)

- Las newsletters o boletines informativos por lo general presentan las noticias de tu compañía (nuevas características o servicios) y/o resumen lo mejor de tu blog. Te permiten permanecer en la mente de sus destinatarios e inspirarlos.
- Las campañas de correo electrónico están más enfocadas en marketing y ventas. Una campaña puede constar de 3 a 10 correos electrónicos durante varios días o semanas. Por ejemplo: una oferta para el día de las madres, ofertas especiales para la temporada de vacaciones, el lanzamiento de un nuevo producto, etc.
- Los correos electrónicos transaccionales, como su nombre lo dicen, son enviados después de una transacción. Estos son, por ejemplo, correos electrónicos de confirmación de reserva, emails de bienvenida cuando alguien se suscribe a tu boletín informativo o restablecer contraseña.

Marketing de Redes Sociales

El marketing de redes sociales es uno de los canales digitales más importantes. Es una herramienta basada en computadoras que permite a las personas crear e intercambiar ideas, información y fotos acerca de los productos o servicios de la compañía. (multiples, 2018) Las redes de marketing social incluyen Facebook, Twitter, LinkedIn y Google+. A través de Facebook, una compañía puede promover eventos relacionados con sus productos o servicios. Con Twitter se puede incrementar la visibilidad de la marca. En LinkedIn, una red social enfocada a los profesionales, en la que un usuario crea un perfil que llena con sus habilidades, títulos, experiencias pasadas, las empresas pueden crear un perfil para que los profesionales puedan obtener más información relacionada con el desenvolvimiento de sus trabajadores, o las áreas del conocimiento en las que está buscando contratar.

Marketing Social

Este no debe ser confundido con el concepto expuesto anteriormente, ya que, aunque comparte nombres similares, sus objetivos y metas son completamente distintos. El marketing social es, de la forma más simple, "la aplicación sistemática del marketing, en

conjunto con otros conceptos y técnicas, para alcanzar metas de comportamientos específicos para beneficio social". (Thornley, et al., 2010) En contraste con el marketing comercial, que apunta a vender productos o servicios por una ganancia comercial, el marketing social promueve cambios de comportamiento voluntarios para mejorar el bienestar personal y de la sociedad. Este tipo de marketing se enfoca en el cambio de comportamientos más que conocimientos o aptitudes. Su enfoque va más allá de reducir comportamientos negativos a enfatizar comportamientos positivos, lo que es consistente con un enfoque para el desarrollo de jóvenes basado en sus fortalezas. Un concepto fundamental en el marketing es el "exchange", que significa que las personas reciben beneficios a cambio de su esfuerzo y el cambio de determinado comportamiento. Por lo que existe el reto para los promotores de estos cambios de maximizar los posibles beneficios a la vez que reduce el posible esfuerzo.

Especialmente cuando el objetivo de estas campañas son los jóvenes, se deben tener en cuenta una serie de conceptos, que, aunque no lo garantizan, si aumentan las posibilidades de llegar a los jóvenes e influir en ellos. Utilizar un enfoque centrado en los jóvenes, "por los jóvenes y para los jóvenes" (Thornley, et al., 2010), donde estos sean incluidos en todos los aspectos de la campaña y su opinión se tome en cuenta y ayude a modelar y refinar la campaña. El uso de canales accesibles, pero, sobre todo, atractivos, así como de mensajes que empoderen la juventud y su necesidad de independencia (Thornley, et al., 2010). El uso mensajes emocionales fuertes e intensos, tanto positivos como negativos. La necesidad de que varias organizaciones se unan y comprometan, para brindar a los jóvenes una experiencia completa, que alcance la mayor cantidad de áreas y aspectos de su vida. El uso de estrategias locales que afecten las zonas más cercanas a los jóvenes, ya sea en el barrio, la escuela, la ciudad, de forma que estos puedan ser partícipes y testigos de los cambios y sus beneficios.

Descripción de la Entidad

La Universidad de Matanzas está integrada por dos sedes "Camilo Cienfuegos" y "Juan Marinello", la Estación Experimental de Pastos y Forrajes "Indio Hatuey", 4 centros universitarios regionales y 9 filiales universitarias municipales. Dispone de diversas instalaciones para dar respuesta a las necesidades docentes, investigativas, de la actividad postgraduada, extensión universitaria y garantizar la vida universitaria en su conjunto. Se estudian 33 carreras universitarias de las Ciencias: Técnicas, Agropecuarias, Económicas,

Sociales, Humanísticas, Pedagógicas y de la Cultura Física. De sus aulas han egresado más de 50 mil estudiantes nacionales y extranjeros. Su claustro de mil profesores lo conforman un 65 % de Doctores en Ciencias y Master. Desarrolla 18 programas de Maestrías o Especialidades y 8 programas de Formación Doctoral. (Cienfuegos, 2019)

Tecnologías empleadas en el desarrollo de la solución.

Hoy en día existen una gran cantidad de metodologías, lenguajes, tecnologías y herramientas para el desarrollo de software, todas con determinadas características que las vuelven indicadas para ciertas situaciones. Estas se pueden combinar para obtener distintos resultados en cuanto a rendimiento, facilidad de uso para el desarrollador, mantenimiento, escalabilidad. Emplear la combinación adecuada para maximizar la facilidad y rapidez a la hora de desarrollar una solución, sin sacrificar la calidad o los requerimientos, es tarea de los desarrolladores.

SCRUM

Scrum es un *framework* de proceso que ha sido utilizado para el trabajo en complejos productos desde los inicios de 1990 (Schwaber, et al., 2017). Scrum no es un proceso, técnica o método definitivo. Mas bien, es un marco de trabajo dentro del cual se pueden emplear varios procesos y técnicas. Scrum establece claramente la eficiencia relativa de la gestión del producto y las técnicas de trabajo de manera que se puede mejorar continuamente el producto, el equipo y el entorno de trabajo (Schwaber, et al., 2017). Está compuesto por equipos y los roles asociados, eventos, artefactos y reglas. Cada componente dentro del *framework* tiene un propósito específico y es esencial para Scrum y su uso. Scrum plantea un desarrollo iterativo e incremental, esto significa que el producto se va desarrollado en pequeñas partes que se prueban como unidad y también integradas al producto ya desarrollado, esta forma de avanzar permite una fácil adaptación ante los posibles cambios que pueden surgir en le transcurso del proceso. Scrum también plantea la necesidad de involucrar al cliente desde el principio de manera que se puedan detectar posibles errores o cambios deseados. Estas dos características hacen de Scrum una metodología apta para el desarrollo de soluciones cuando no existe por parte del cliente un dominio total del negocio.

BPMN

Business Process Model and Notation (BPMN), en español, Modelo y Notación de Procesos de Negocio, es una notación gráfica estandarizada que permite el modelado de procesos de negocio, en un formato de flujo de trabajo (workflow) (2016). Es una notación gráfica que describe la lógica de los pasos de un proceso de negocio. Esta notación ha sido especialmente diseñada para coordinar la secuencia de los procesos y los mensajes que fluyen entre los participantes de las diferentes actividades. BPMN proporciona un lenguaje común para que las partes involucradas puedan comunicar los procesos de forma clara, completa y eficiente. De esta forma BPMN define la notación y semántica de un Diagrama de Procesos de Negocio (Business Process Diagram, BPD).

Se escogió BPMN para especificar el proceso objeto de estudio ya que en las entrevistas con el cliente se pudo determinar que no existen actores establecidos, sino mas bien flujos de información a través de varias etapas. La utilización de BPM proporciona un mejor entendimiento de estos flujos.

Modelo Vista Controlador

El modelo–vista–controlador (MVC) es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres capas distintas que son el modelo, la vista y el controlador, es decir, por un lado, define componentes para la representación de la información, y por otro lado para la interacción del usuario.

La separación del modelo, la vista y el controlador agrega flexibilidad al permitir múltiples vistas del mismo modelo, en caso de que un usuario cambie el modelo a través del controlador, todas las vistas que utilizan el mismo modelo son actualizadas. Toda esta secuencia pasa a través del mecanismo de propagación que debe estar presente en la implementación de este patrón. (Pethuru, et al., 2017)

Este patrón de arquitectura de *software* se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

En el caso de la solución propuesta se pueden establecer las distintas capas de la siguiente forma:

- Controlador: la aplicación de servidor que se encarga de la lógica de almacenamiento y control de permisos
- Modelo: MongoDB.
- Vista: la aplicación cliente que utiliza Angular para mostrar la información contenida en la base de datos y enviada a través de la aplicación de servidor.

Se puede notar que la aplicación cliente, al utilizar Angular, establece su propia capa MVC, ya que está compuesta por componentes(Controlador), plantillas(Vista) y servicios(Modelo).

JavaScript

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

JavaScript es desarrollado por ECMA, una empresa que desarrolla estándares para las tecnologías, por lo que JavaScript también se conoce como ECMAScript y se utiliza la numeración para nombrar las versiones.

ECMAScript 6 (ES6) fue publicado por ECMA en junio de 2015. También existen referencias a este como “Harmony”, “ES6”, “ES2015” o “ECMAScript 2015”. ES6 representa un avance significativo en JavaScript, y algunos de los mayores problemas de ES5 fueron atendidos en esta nueva versión (Brown, 2016). En conjunto con ES6 también surgieron los transpiladores, programas que interpretan ES6 y producen el código en ES5, como un medio para escribir ES6 para los entornos donde todavía no es soportado.

Node.js

Node.js es un intérprete de JavaScript asíncrono orientado a eventos, está diseñado para construir aplicaciones de red escalables. Esto en contraste con los modelos concurrentes utilizados comúnmente hoy en día, donde se utilizan hilos del sistema. Los sistemas de red

basados en hilos son relativamente ineficientes y bastante difíciles de usar. Por el contrario, los usuarios de Node no necesitan preocuparse por los errores de bloqueos mortales de procesos ya que no hay bloqueos. Casi ninguna función en Node realiza procesos de E/S, por lo que el proceso no se bloquea. (Node.js)

El código JavaScript ejecutado en los navegadores y en Node es compilado utilizando el motor de JavaScript V8, que es el encargado de compilar el código a código de máquina que es mucho más rápido. El código de máquina es un código de bajo nivel que la computadora es capaz de interpretar directamente. (Mead, 2018)

Angular

Es un *framework* de JavaScript de código abierto, mantenido por Google, que ayuda con la gestión de lo que se conoce como aplicaciones de una sola página. Está basado en componentes, que a su vez se agrupan en módulos, permitiendo descomponer la aplicación en pequeñas partes, lo que supone mayor reusabilidad y facilidad de mantenimiento.

Angular trae consigo algunas ventajas significativas a la vez que provee una estructura común para que los desarrolladores en un equipo puedan colaborar más fácilmente. Algunos de los conceptos principales son (Seshadri, 2018):

- Componentes personalizados: Permiten escribir componentes que agrupan funcionalidades y lógica visual en partes pequeñas y reusables.
- Inyección de dependencias: Permite escribir servicios de forma modular e inyectarlos en los componentes donde se necesiten
- Abarcador: Angular es un framework completamente desarrollado y provee soluciones para la comunicación con el servidor, manipulación de rutas dentro de la aplicación.

Otra ventaja significativa de Angular es que está pensado para el paradigma de microservicios, donde los servidores solo se encargan de enviar los datos necesarios y son los dispositivos clientes los que disponen de los datos para mostrarlos al cliente. Este paradigma permite la creación de sistemas multiplataforma, como pueden ser la web y la móvil, minimizando así los tiempos y la complejidad de desarrollo. Todas estas características hacen de Angular una herramienta ideal para el sistema a desarrollar.

Express.js

Express es un *framework* de servidor web minimalista y flexible escrito JavaScript utilizando Node.js que proporciona un conjunto sólido de características para el soporte de las aplicaciones *web* y móviles modernas, implementando una arquitectura de microservicios y el estándar REST para la comunicación.

Esta basado en el modulo http de NodeJS y componentes *Connect*. Esos componentes son llamados *middlewares* y son la piedra angular de la filosofía del *framework*, la cual es configuración sobre convención (configuration over convention), lo que significa que los desarrolladores son libres de escoger cualquier otra librería que necesiten para un proyecto. Este enfoque provee flexibilidad y una alta capacidad de personalización (Mardan, 2014).

Se utilizó Express.js porque está diseñado para implementar una arquitectura de microservicios, la cual es la base del sistema a implementar, además permite un rápido desarrollo utilizando el mínimo necesario de configuraciones, así como permite mantener un solo lenguaje en el desarrollo del sistema y una gran integración con el SGBD MongoDB a través del módulo Mongoose al utilizar los mismos tipos de objetos.

MongoDB

MongoDB es un sistema de gestión de bases de datos diseñada para el rápido desarrollo de aplicaciones web. El modelo de datos y la estrategia de persistencia fueron creados para altas tasas de lectura – escritura y la habilidad de escalar fácilmente (Banker, et al., 2016).

Es una de las numerosas tecnologías de bases de datos no relacionales surgidas a mediados del 2000, utilizadas en aplicaciones de Big Data y otros trabajos de procesamiento que involucran datos que no encajan en un modelo rígido relacional. La arquitectura de MongoDB está compuesta por colecciones de objetos. (Rouse)

MongoDB es una base de datos sin esquemas, lo que significa que se puede almacenar objetos BJSON (Binary JSON), que son una variante de los conocidos objetos JSON pero que aceptan mayor cantidad de tipos de datos; la estructura de estos puede cambiar, ya que no es requerida, como en las bases de datos SQL. Esto es una de las ventajas de usar NoSQL, ya que agiliza el desarrollo de aplicaciones y reduce la complejidad a la hora del despliegue permitiendo que se agreguen nuevos campos sin afectar los objetos ya existentes.

Una entrada en MongoDB es un documento, que es una estructura compuesta por pares <campo, valor>, los valores de los campos pueden incluir otros documentos, arreglos y/o arreglos de documentos, además de los tipos de datos primarios comunes a todos los lenguajes de programación.

Se escogió MongoDB como sistema de bases de datos porque permite el almacenamiento de estructuras complejas, lo que a su vez permite reducir el número de consultas que se realizan y con esto los tiempos de respuesta, también por ser un sistema abierto al cambio en el que pueden cambiar los modelos de datos sin afectar a los datos ya existentes.

Mongoose

Mongoose es una herramienta de modelado para MongoDB y Node.js. Lo que significa, en términos prácticos, que permite definir los modelos de datos en un solo lugar, el código. Esto evita la definición de esquemas en la base de datos. La estructura de los datos se puede definir en formato *JSON* dentro del proyecto. (Holmes, 2013)

Mongoose es utilizado principalmente cuando se quiere interactuar con datos estructurados dentro de MongoDB, también ayuda con muchas de las tareas comunes de MongoDB a la vez que remueve algunos niveles de complejidad que puede causar la comunicación nativa entre MongoDB y una aplicación de NodeJS.

Nginx

Nginx es un servidor HTTP de alto rendimiento, libre y de código abierto. También se puede usar como proxy inverso o servidor proxy para IMAP / POP3. Es conocido por su alto rendimiento, estabilidad, amplio conjunto de características, simple configuración y bajo consumo de recursos. (NGINX)

Nginx ha surgido en la última década como un servidor web de propósito general robusto y escalable. Es una opción para muchos *web masters* e ingenieros de estabilidad gracias a su simple a la vez que expandible arquitectura, fácil configuración y ligero consumo de memoria. Este ofrece un conjunto de funcionalidades manejo de compresión y cache integradas desde el inicio. (Kholodkov, 2015)

Es uno de los servidores escritos para manejar el problema C10K, que consiste en la sobrecarga del servidor a partir de una gran cantidad de peticiones, lo que causa que el servidor colapse y no responda a más peticiones e incluso se apague. Este problema es la base del conocido ataque de denegación de servicios. Distinto a los servidores tradicionales, Nginx no se basa en hilos para gestionar las peticiones, sino que usa una arquitectura basada en eventos (asíncrona) más escalable. Nginx en servidores de alta visibilidad como son Netflix, Airbnb, GitHub, SoundCloud, Heroku y muchas otras. (NGINX).

Docker

Docker es un motor de contenedores que automatiza la creación distribución y despliegue de cualquier aplicación de software a través contenedores, que son ligeros, portables, autosuficientes y pueden ejecutarse en virtualmente cualquier entorno. (Raj, et al., 2015)

Los contenedores permiten a los desarrolladores agrupar una aplicación con todas las partes que necesita, como librerías u otras dependencias, y distribuirla como una sola aplicación. Al hacerlo, los desarrolladores se aseguran de que la aplicación pueda ejecutarse correctamente en cualquier otra máquina Linux sin configuración adicional y sin verse afectada por que pueda tener configuraciones diferentes a las presentes en la maquina donde se desarrolló y probó originalmente la aplicación. (OpenSource)

Docker es, de cierta forma, como una máquina virtual. Pero contrario a una máquina virtual, en vez de crear un sistema operativo completamente nuevo, Docker permite a las aplicaciones usar el mismo kernel de Linux del sistema en el que se están ejecutando, solamente necesita que las aplicaciones sean desplegadas con los requerimientos que no existen en el servidor. Esto posibilita una mejora de rendimiento significativa y reduce el tamaño de la aplicación.

Lighthouse

Lighthouse es una herramienta desarrollada por Google que analiza aplicaciones web y páginas web, recolectando métricas de rendimiento modernas y posibles oportunidades para aplicar buenas prácticas. Lighthouse ejecuta comprobaciones a las aplicaciones web y proporciona una retroalimentación sobre errores, prácticas que están por debajo de los

estándares, consejos para un mejor rendimiento y como arreglarlos (Ayodeji, 2019). Realiza auditorias al rendimiento, la accesibilidad, las potencialidades y capacidades para ser una aplicación web progresiva (PWA) y más. Una vez terminadas las auditorias genera un reporte detallando las puntuaciones obtenidas en cada una de las métricas. Cada una de ellas contiene una documentación de referencia y posibles soluciones.

Otros conceptos importantes

Pruebas de software

Las pruebas de software son una parte integral del ciclo de vida de desarrollo del mismo ya que identifican defectos, fallas y errores en la aplicación. Son de naturaleza iterativa e incremental. (Education, 2012) Objetivos de las pruebas de software:

- Asegurar que la solución cumpla con los requerimientos del negocio, asegurando así la confianza del usuario
- Detecta *bugs*, errores o fallas
- Aseguran que el sistema es estable y está listo para el uso
- Identifica áreas de debilidades en el producto
- Establece un grado de calidad
- Determina la aceptabilidad del usuario.

El alcance de los test puede llegar a ser exhaustivo y comprobar componentes como requerimientos del negocio, requerimientos de diseño, el código y las configuraciones de *hardware*. También se pueden realizar test para comprobar el uso de estándares de la industria, así como buenas practicas profesionales. De esta forma, las pruebas proveen una oportunidad para validar y verificar todos los aspectos de la ingeniería de *software*.

Entre los tipos más básicos y usados de pruebas se encuentran las pruebas estructurales y de funcionalidades, también llamadas pruebas de caja blanca y de caja negra respectivamente. Este tipo de prueba cae en la categoría de pruebas unitarias ya que son realizadas a partes del código, desde las más pequeñas hasta algunas integraciones de distintos componentes.

Pruebas funcionales o de caja negra

Las pruebas de caja negra comprueban si el software funciona según los requerimientos o especificaciones establecidas. Se conocen de esta forma porque se ejecutan las pruebas sin conocer la lógica interna, sino que se enfoca en las salidas generadas a partir de las entradas seleccionadas y las condiciones de ejecución. (Education, 2012)

Este tipo de pruebas permite conocer si el software hace lo que se espera. Las especificaciones funcionales son la fuente de información empleada, por lo que se requiere una especificación detallada de las mismas.

Con el enfoque basado en especificaciones para identificar casos de prueba, la única información utilizada son las especificaciones del software por lo que los casos de usos tienen dos ventajas principales: son independientes de como el software es implementado, por lo que si esta implementación cambia, los casos de pruebas siguen siendo útiles, además, el desarrollo de los casos de prueba puede ocurrir en paralelo a la implementación, lo que reduce el tiempo de desarrollo general del producto. (Jorgensen, 20114)

Entre las ventajas de las pruebas de caja negra se encuentran: (Education, 2012)

- No se necesita entender la lógica interna de la función o componente
- Se interactúa mayormente con la interfaz de usuario por lo que los casos de uso se pueden desarrollar rápida y fácilmente.
- Los casos de uso se pueden diseñar tan pronto como se completan las especificaciones del producto
- Ayudan a encontrar ambigüedades o inconsistencias en las especificaciones del sistema

Pruebas estructurales o de caja blanca

Este tipo de pruebas se enfoca en la estructura del código del software. Se desarrollan casos de uso que para comprobar la lógica funcional del código. Mientras que las pruebas de caja negra responden la pregunta de validación “¿estamos construyendo el software correcto?” las pruebas de caja blanca responden la pregunta de verificación “¿estamos construyendo en software correcto?” (Education, 2012)

En las pruebas de caja negra, cada parte del software es probada independientemente y en con otras partes. Todas las pruebas se desarrollan a nivel de código base, se prueban todos los parámetros del código tales como eficiencia, lógica interna, comunicación con otros componentes, organización de memoria, claridad del código. Los casos de uso deben ser cuidadosamente diseñados para cubrir toda la lógica de la aplicación.

Entre las ventajas que proporciona este tipo de pruebas se encuentra: (Education, 2012)

- Se conoce la lógica interna por lo que se hace fácil el desarrollo de casos de uso que prueben el producto efectivamente
- Las pruebas se realizan a nivel de código por lo que este se puede optimizar con el desarrollo de más pruebas más específicas cada vez.
- Partes innecesarias pueden ser identificadas y eliminadas.

Conclusiones parciales

En este capítulo se expusieron diferentes aspectos que son importantes confrontar para poder introducirse en el tema de esta investigación. Luego de estudiar los antecedentes del proceso objeto de estudio y los aspectos generales de la entidad, así como las herramientas y tecnologías y la metodología de desarrollo de software, se concluye que:

- La utilización de los conceptos y técnicas del *marketing* digital para el desarrollo de la solución puede ayudar a cumplir el objetivo de esta investigación.
- La combinación de las herramientas de desarrollo seleccionadas es la apropiada para desarrollar la aplicación que dará solución al problema de esta investigación.
- La utilización de la metodología SCRUM para el desarrollo del sistema constituye una buena elección por la forma iterativa e incremental que promueve, así como el uso del lenguaje BPMN para detallar el proceso objeto de estudio
- La aplicación de pruebas al software es una buena práctica y de gran necesidad para garantizar calidad en el producto y mejorar los tiempos de desarrollo.

CAPITULO 2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

En este capítulo se explica el flujo de información y cómo va a llegar al estudiante. Se plantean las funcionalidades del sistema, se exponen las etapas del proceso de ingeniería de *software* por las que se transita durante el desarrollo de un proyecto y se detalla el proceso objeto de estudio.

2.1 Propuesta de la aplicación

El sistema *actívate* tiene como objetivo la promoción de eventos universitarios, para ello contará con una aplicación para dispositivos móviles a través de la cual se podrán ver los eventos y comentar sobre ellos, y 2 aplicaciones web, una, para la gestión de la información de dichos eventos por el departamento y otra para los estudiantes, la cual replicaría las posibilidades de la aplicación móvil, permitiendo así un mayor alcance.

La aplicación web de gestión estará compuesta por 3 vistas fundamentales, destinadas a la gestión de la información relacionada con las categorías, los eventos y los usuarios que tienen acceso a ella.

En el caso de los usuarios, solo se permiten los que estén previamente registrados en el dominio de la universidad, asegurando así que sean usuarios reales. Entre los usuarios existen 2 roles, los administradores, encargados de gestionar las categorías y los usuarios que harán función de gestores, y los gestores en sí.

Los administradores pueden cambiar el gestor de un evento determinado, también pueden hacer función de gestores, aunque no se recomienda, ya que impide establecer una clara separación de responsabilidades.

Los gestores son los encargados de publicar y actualizar la información referente a los eventos, así como los comentarios de cada evento, pudiendo ocultarlos o eliminarlos según sea conveniente.

En las aplicaciones para los estudiantes se podrá ver los eventos agrupados por categoría, ordenados ascendentemente por la fecha de inicio, de forma que los primeros eventos sean los más próximos.

El objetivo principal del sistema, son los eventos futuros, pero no se pueden descartar los ya pasados, ya que pueden servir de guía y ejemplo en la creación de futuros eventos, por este motivo existe una categoría extra, que agrupa a dichos eventos.

Existe una vista de detalles en la cual se puede leer toda la descripción de evento, así como los comentarios de otros usuarios y agregar los propios. El sistema cuenta con un mecanismo de filtrado, de manera que los comentarios que contengan palabras no deseadas, ya sean contra revolucionarias u obscenas, no serán publicados, pero si pueden ser vistos por los gestores y administradores, de manera que se puedan tomar medidas. Los usuarios pueden ser bloqueados por los administradores, impidiendo así que puedan interactuar con el sistema.

Por último, cabe destacar que el sistema traerá algunas ventajas adicionales al Departamento de Comunicación, ya que no necesitaría estar al tanto de los detalles del evento para poder responder a los posibles asistentes que tengan alguna duda y podrían centrarse en ayudar a los encargados de los eventos a crear la información de los eventos de la mejor forma posible, dejando a estos la comunicación con los interesados en los eventos.

2.2 Modelo de Proceso

En la figura 2.1 se puede observar representado el proceso de creación y publicación de un evento. Este esta constituido por la entrevista que se realiza a la persona encargada del evento, en esta entrevista se recopilan todos los datos necesarios para la creación de la promoción, desde el nombre, la fecha, el lugar, una descripción del evento y los pasos par participar. También se decide de que forma se va distribuir esta información, las formas mas comunes con a través del correo universitario o con pancartas que se distribuyen en determinados puntos con gran flujo de estudiantes.

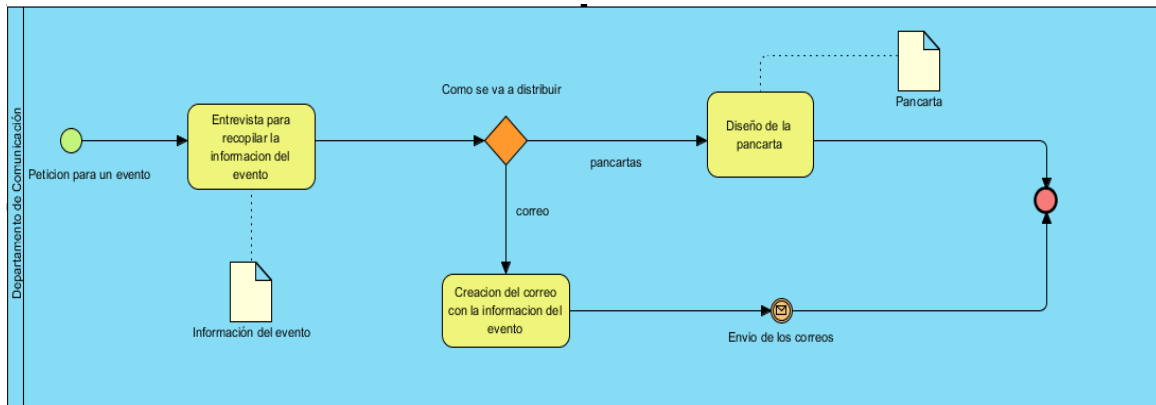


Figura 2.1 Modelo de proceso

2.3 Pila del Producto (Product Backlog)

La Pila del Producto es una lista ordenada de todo lo que se conoce es necesario para el producto. Es la única fuente de requerimientos y cambios para el producto. (Schwaber, et al., 2017) En ella se describen todas las funcionalidades y características del sistema y puede cambiar a medida que se desarrolla el producto y el cliente valora el progreso con respecto al objetivo deseado, que sería su ideal del producto terminado. Cabe destacar que solo el cliente (Product Owner) puede cambiar la pila del producto.

De estas entrevistas se obtuvieron los siguientes requerimientos:

- Gestionar las categorías
- Crear y modificar la información referente a los eventos
- Listar los eventos
- Buscar eventos por nombre
- Seguir y comentar eventos
- Filtrar los comentarios a la hora de publicarlos
- Autenticación con el dominio
- Agregar y modificar los permisos de usuarios como gestores y administradores
- Bloquear usuarios para impedir la interacción con el sistema

2.4 Planeación de las Entregas (Sprints)

En la planeación de las entregas se establecen los plazos dentro de los cuales se debe desarrollar las funcionalidades especificadas. Cada entrega se puede dividir en una serie de historias, que a su vez pueden estar compuestas por tareas. Esta fragmentación permite asignar y desarrollar múltiples tareas al mismo tiempo. Las fechas de las entregas pueden variar a medida que aparecen cambios en los componentes de la pila del producto o algún otro tipo de complicaciones en el desarrollo. En la siguiente tabla se muestran las fechas reales en que se terminaron las entregas.

Tabla 2.1 Planeación de las entregas

Sprint	Fecha de Inicio	Fecha de Fin
Diseño de la Base de Datos	07 enero 2019	11 enero 2019
Arquitectura inicial	14 enero 2019	25 enero 2019
Funcionalidad Categorías	04 marzo 2019	11 marzo 2019
Funcionalidad Eventos	12 marzo 2019	22 marzo 2019
Funcionalidad Usuarios	25 marzo 2019	05 abril 2019
Autenticación	08 abril 2019	19 abril 2019
Crear la aplicación móvil	22 abril 2019	03 mayo 2019

2.5 Descripción de las Entregas

A continuación, se detallan la composición de las entregas en historias y tareas.

2.5.1 Diseño de la Base de Datos

Historia	Tarea	Nombre
1		Diseño de la Base de Datos
	1	Diseño de la Base de Datos

2.5.2 Arquitectura inicial

Historia	Tarea	Nombre
1		Crear la Arquitectura inicial
	1	Crear la aplicación de servidor
	2	Crear la aplicación de gestión
	3	Crear la aplicación para los usuarios finales

2.5.3 Funcionalidad Categorías

Historia	Tarea	Nombre
1		Crear los endpoints en el servidor
	1	Crear el modelo y los métodos GET
	2	Crear los endpoint POST / PUT
	3	Crear el endpoint DELETE
2		Crear los las vistas de gestión
	1	Crear el modelo y el servicio para conectar con el servidor
	2	Crear la vista para listar y eliminar
	3	Crear el formulario para crear / editar

3	Integrar las categorías (AUF)	
	1	Crear el modelo y servicio para conectar con el servidor
	2	Agregar las categorías al menú de navegación

2.5.4 Funcionalidad Eventos

Historia	Tarea	Nombre
1	Crear los endpoints en el servidor	
	1	Crear el modelo y los métodos GET
	2	Crear los endpoint POST / PUT
	3	Crear el endpoint DELETE
2	Crear los las vistas de gestión	
	1	Crear el modelo y el servicio para conectar con el servidor
	2	Crear la vista para listar y eliminar
	3	Crear el formulario para crear / editar
3	Integrar los eventos (AUF)	
	1	Crear el modelo y servicio para conectar con el servidor
	2	Crear las vistas y componentes para representar los eventos
	3	Crear la vista de detalles y el formulario de comentarios

2.5.5 Funcionalidad Usuarios

Historia	Tarea	Nombre
1	Crear los endpoints en el servidor	
	1	Crear el modelo y los métodos GET
	2	Crear los endpoint POST / PUT
	3	Crear el endpoint DELETE

2	Crear los las vistas de gestión
1	Crear el modelo y el servicio para conectar con el servidor
2	Crear la vista para listar, modificar y eliminar
3	Crear el formulario para agregar / editar
3	Integrar los usuarios (AUF)
1	Agregar la funcionalidad de cambiar la foto de perfil

2.5.6 Autenticación

Historia	Tarea	Nombre
1		Crear la configuración en el servidor
	1	Crear el endpoint de autenticación
	2	Crear los servicios para chequear las peticiones y los roles
	3	Crear el servicio para usuarios bloqueados
2		Implementar la integración con LDAP
	1	Implementar la función de autenticación con LDAP
	2	Implementar la función de verificar un usuario en LDAP
	3	Modificar el endpoint de autenticación para utilizar LDAP
3		Implementar la autenticación (AG)
	1	Crear el servicio de autenticación y la vista
	2	Implementar los servicios de autenticación de peticiones y manejo de errores
	3	Proteger las vistas según los roles
4		Implementar la autenticación (AUF)
	1	Crear el servicio de autenticación y la vista
	2	Implementar los servicios de autenticación de peticiones y manejo de errores

2.5.7 Crear la aplicación móvil

Historia	Tarea	Nombre
1		Crear la vista para listar las categorías
	1	Importar el modelo y servicio para conectar con el servidor
	2	Agregar las categorías al menú de navegación
2		Crear la vista para listar los eventos y los detalles
	1	Importar el modelo y servicio para conectar con el servidor
	2	Crear la vista que carga los eventos según la opción escogida en el menú de navegación
	3	Crear la vista de detalles del evento
3		Crear la vista de autenticación
	1	Importar los servicios de autenticación y manejo de errores
	2	Crear la vista de autenticación y conectarla con las funciones que lo requieren
4		Crear la vista de Perfil
	1	Crear la vista de perfil

2.6 Definición del Equipo

Un equipo de Scrum está compuesto por el dueño del producto, el equipo de desarrollo y un Scrum master. Los equipos son auto organizados y multifuncionales. (Schwaber, et al., 2017) Equipos auto organizados escoges como es mejor llevar a cabo su tarea, sin ser dirigidos por otros fuera del equipo. Equipos multifuncionales tienes todas las capacidades necesarias para completar su trabajo sin depender de otros que no pertenecen al equipo.

Los equipos de Scrum entregan partes del producto de forma iterativa e incremental, maximizando las oportunidades de retroalimentación. Entregas incrementales de producto

“hecho” asegura que siempre exista una versión potencialmente usable del producto funcional.

Tabla 2.6.1 Definición de los roles del equipo de Scrum

Rol	Miembro
Dueño del producto	Departamento de Comunicación
Equipo de desarrollo	Alejandro Yanes De la Cruz
<i>Scrum master</i>	Alejandro Yanes De la Cruz

2.7 Estimación de costo

La estimación del costo de un software es el proceso de predecir la cantidad de esfuerzo requerido para el desarrollo del sistema y el tiempo para ello.

Existen modelos que proveen algoritmos matemáticos para calcular el costo como una función de varias variables como el tamaño (LC o PF) y/o la complejidad (usando complejidad ciclométrica)

La estimación por puntos de función consiste en la suma ponderada de parámetros básicos para dimensionar funcionalmente una aplicación. Considera las características externas relevantes para usuarios. Se usa en etapas tempranas del desarrollo. Son independientes del lenguaje y convierte a Kloc (miles de líneas de código) para lenguajes específicos.

EI:

- 3 procesos con más de 2 archivos y de 5 – 15 atributos (Alta - 6)
- 3 proceso con 1 archivo y 5 – 15 atributos (Baja - 3)
- 3 proceso con 1 archivo y 1 – 4 atributos (Baja - 3)

EO:

- 1 proceso con 1 archivo y 1 – 5 atributos (Baja - 4)

- 1 proceso con 1 archivo y 6 – 19 atributos (Baja - 4)
- 1 proceso con 2 - 3 archivos y 6 – 19 atributos (Media - 5)

Tabla 3.1 Cálculo de los Puntos de Función Sin Ajustar

Componente	Bajo	Medio	Alto	Total
EI	$E_b * 3 = 6$	$E_m * 4 = 0$	$E_a * 6 = 18$	PFT _e
EO	$O_b * 4 = 8$	$O_m * 5 = 5$	$O_a * 7 = _$	PFT _o
EQ	$Q_b * 3 = 1$	$Q_m * 4 = _$	$Q_a * 6 = 1$	PFT _q
ILF	$I_{Fb} * 7 = _$	$I_{Fm} * 10 = _$	$I_{Fa} * 15 = 2$	PFT _{if}
EIF	$E_{Fb} * 5 = _$	$E_{Fm} * 7 = _$	$E_{Fa} * 10 = _$	PFT _{ef}
				PFS _A = 73

Tabla 3.3 Listado de factores de complejidad técnica

N.º de Factor	N.º de Factor	Valor 0...5
1	Comunicación de Datos	3
2	Proceso Distribuido	2

3	Objetivos de Rendimiento	3
4	Configuración de Explotación Compartida	2
5	Tasa de transacciones	1
6	Entrada de Datos en Línea	2
7	Eficiencia con el Usuario Final	5
8	Actualizaciones en Línea	3
9	Lógica de Proceso Interno Compleja	3
10	Reusabilidad del Código	2
11	Conversión e Instalación contempladas	2
12	Facilidad de Operación	1
13	Instalaciones Múltiples	1
14	Facilidad de Cambios	3
	Ajuste de Complejidad Técnica (ACT)	33

Puntos de función ajustados

$$\text{PFA} = \text{PFSA} * [0.65 + [0.01 * \text{ACT}]]$$

$$PFA = 73 * [0.65 + [0.01 * 39]]$$

$$PFA = 71.54$$

Cálculo del Esfuerzo

		Esfuerzo	
Entorno y Lenguaje		Líneas de Código por PF	Horas por PF
Lenguajes VisualXX	4GL:	20	5 a 10

$$\text{LÍNEAS DE CÓDIGO} = PFA * (\text{LINEAS POR PF})$$

$$= 71.54 * (20)$$

$$= 1430.8$$

$$\text{Esfuerzo horas/persona} = PFA / [1 / 8 \text{ persona / hora}]$$

$$= 71.54 / 0.125 = 572.32 \text{ horas / persona}$$

Cálculo de la Duración del Proyecto

DURACIÓN DEL PROYECTO EN HORAS

$$= 572.32 \text{ horas / persona} / 1 \text{ persona} = 572.32 \text{ horas por miembro}$$

DURACIÓN EN MESES

= 572.32 horas / 100 horas/mes = 6 meses

Conclusiones parciales

En este capítulo se realizó una descripción de la solución a desarrollar, se detallaron las funcionalidades que se espera tenga el sistema. Se describió el proceso objeto de estudio utilizando BPMN para enmarcar un lenguaje común entre el cliente y el equipo de desarrollo. Se describió como serán las entregas, qué se debe entregar en cada Sprint y cuando entregarlo.

CAPÍTULO 3: RESULTADOS Y DISCUSIÓN

En este capítulo se presentan los resultados que se obtienen al aplicar las metodologías explicadas en el capítulo 2, así como las pruebas al software realizadas, que permiten conocer el grado de calidad del producto, y de esta forma, comprobar si el sistema es capaz de realizar todas las funcionalidades detalladas anteriormente. Se prueban las características más importantes de la aplicación con el fin de verificar la fiabilidad y calidad de la aplicación como un todo.

3.1 Descripción de la solución

El sistema desarrollado como resultado de la investigación está compuesto por una aplicación de servidor web escrito en JavaScript utilizando el *framework* Express.js, que se comunica con una Base de Datos de MongoDB, todo esto montado en una arquitectura de Docker, lo que permite que el sistema sea de fácil despliegue al tener toda la configuración de la arquitectura y la red necesaria en un archivo de Docker con el cual se puede desplegar o apagar todos los servicios instantáneamente. En la arquitectura de Docker también están los 2 sitios web necesarios para el uso del sistema, estos se desarrollaron en Angular y están desplegados utilizando Nginx, un servidor web de contenido estático de fácil configuración y optimizado para esta función, también permite la redirección de peticiones, capacidad que es utilizada por el sistema ya que el servidor de Nginx es el que recibe todas las peticiones hechas por los navegadores, incluso las hechas a la API (servidor) y estas son redireccionadas internamente para su procesamiento. El sistema tiene acceso al dominio usuarios de la universidad a través de los microservicios desarrollados por el Departamento para la Gestión de la Informatización Universitaria, de esta forma se garantiza que los usuarios que acceden al sistema sean reales.

3.2 Resultados obtenidos

3.2.1 Interfaces de usuario

En este apartado se muestran algunas de las interfaces del sistema. En todas se puede apreciar un diseño sencillo, donde la navegación siempre queda a la izquierda del usuario y el contenido de la página queda ocupando la mayor parte de la página, atrayendo así la atención del usuario para la información que se muestra. Predominan los colores claros, casi siempre entre el blanco y el gris claro y se utilizan los colores principales del sistema para resaltar partes importantes y funcionales, como son la navegación a la izquierda o los botones para crear. En ambas aplicaciones web, la información referente al usuario que está autenticado está en la parte superior derecha de la aplicación, creando así una forma fácil de acceder a ella ya que, aunque el contenido una página determinada pueda exceder el tamaño de la pantalla, la barra superior siempre va a estar visible. En la barra superior también se puede encontrar la barra de búsqueda, esta se puede utilizar para filtrar en las vistas que contienen listas, para buscar el o los elementos deseados. Actualmente solo es posible buscar eventos utilizando el nombre y en el caso de los usuarios se puede utilizar el nombre real o el nombre de usuario del dominio.

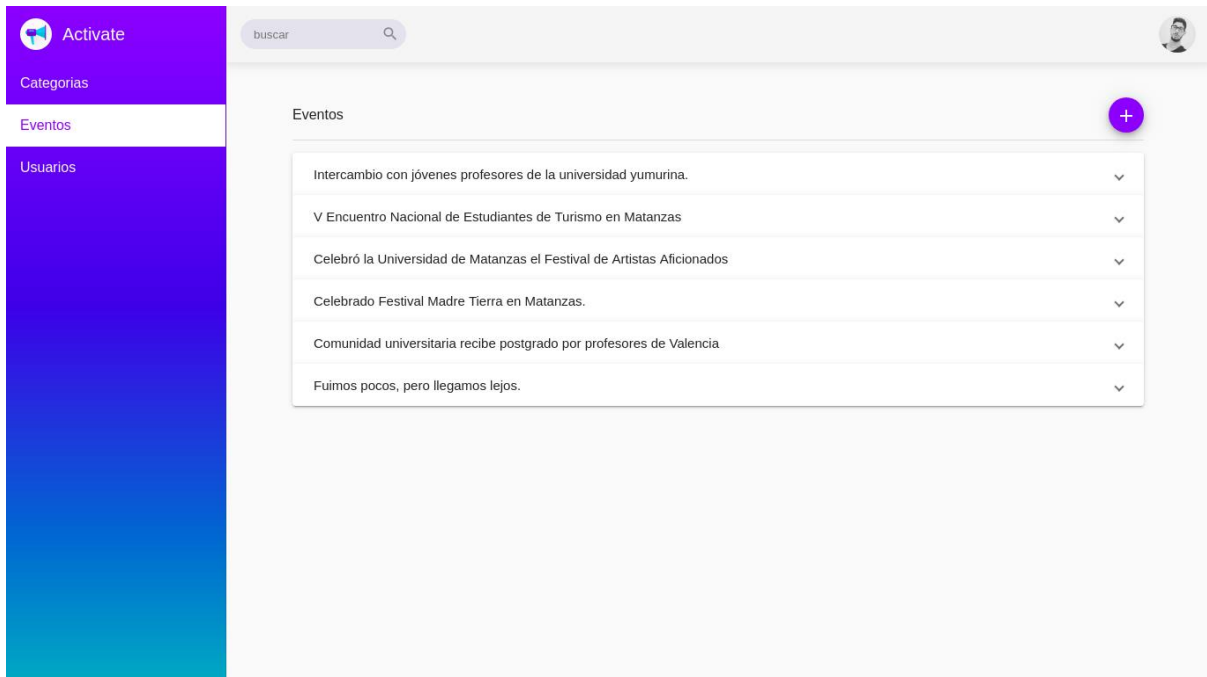


Figura 3.3 Vista de Eventos en la aplicación de gestión

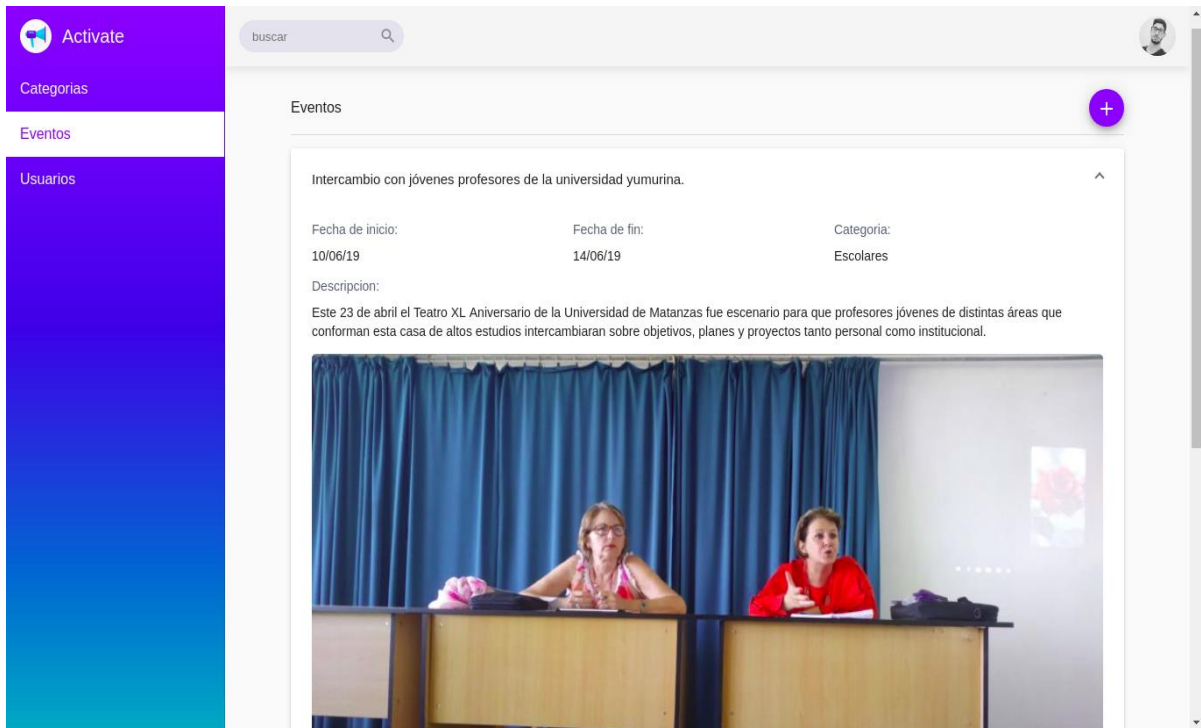


Figura 3.4 Vista de Eventos en la aplicación de gestión con un evento desplegado donde se puede apreciar los datos más relevantes.

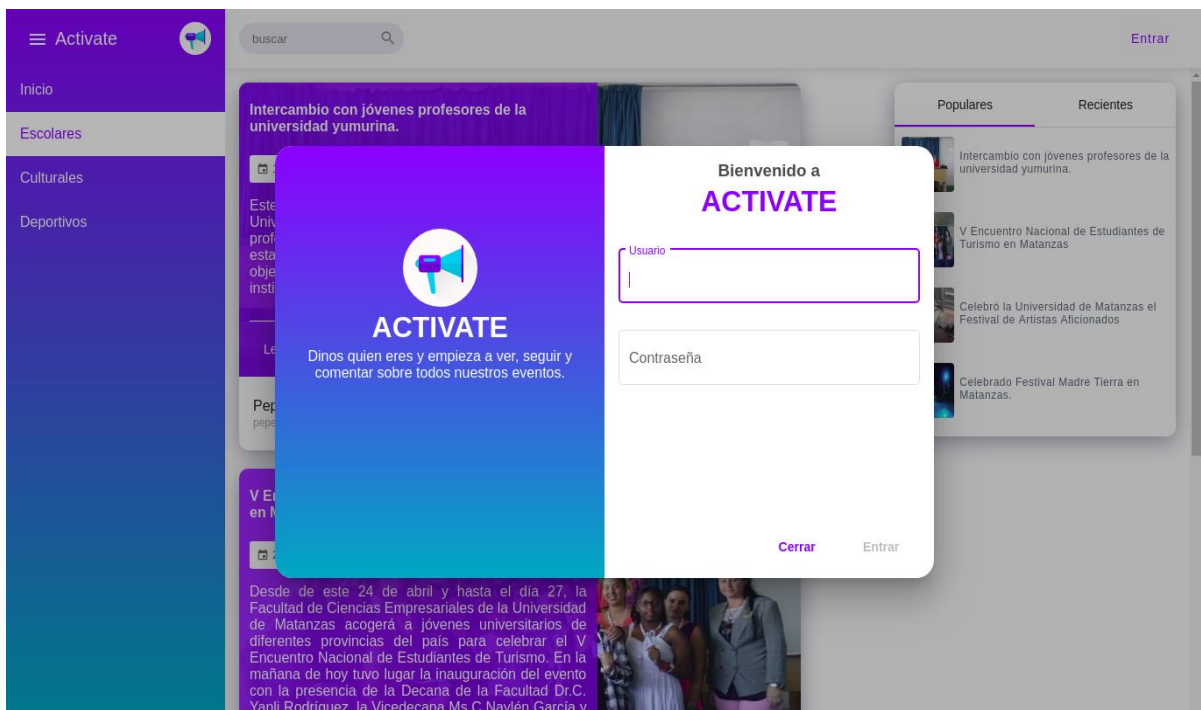


Figura 3.5 Vista del dialogo de inicio de sesión en la aplicación para los estudiantes

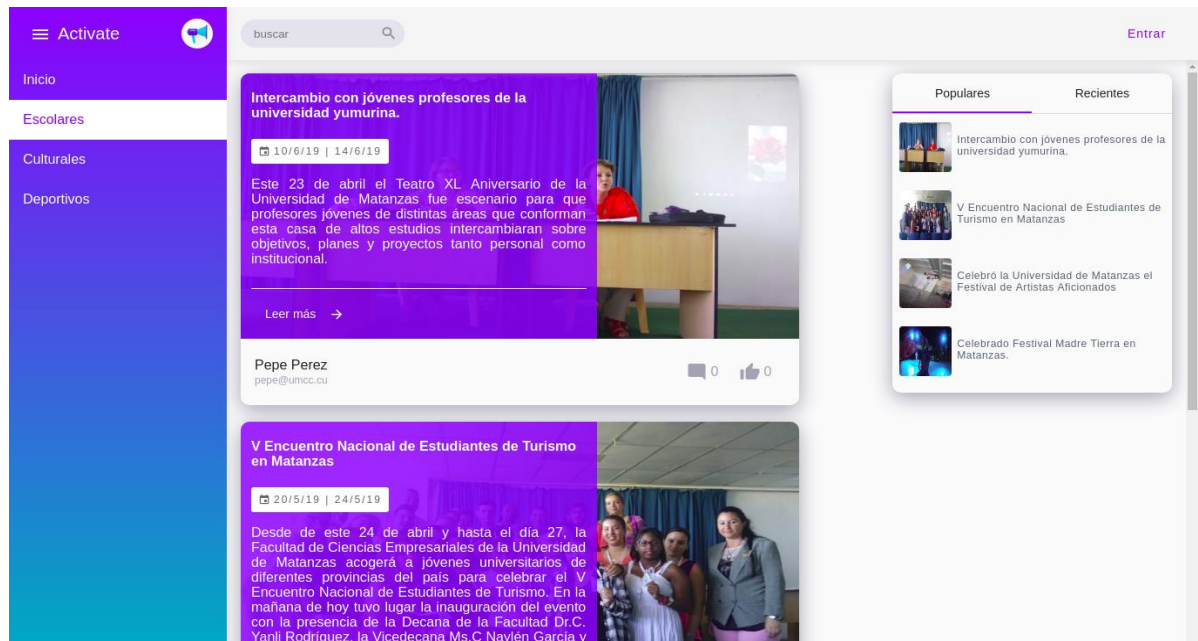


Figura 3.6 Vista de los eventos en la aplicación para los estudiantes, aquí se puede apreciar la simplicidad de la interfaz en la que los eventos tienen la mayor atención

3.2.2 Algoritmos principales

A continuación, se muestran algunos de los métodos más importantes del sistema como son la autenticación en el dominio y la creación de la información de usuario necesaria a partir de la respuesta de este.

En la Figura 3.7 se muestra el código utilizado para hacer la petición al microservicio de comunicación con el dominio, se puede observar como se construye la petición, utilizando los atributos necesarios, como la dirección *IP:PORT*, que establece en que maquina real se encuentra alojado el servicio y el puerto por el cual comunicarse con él. Se puede observar que en la parte de las cabeceras o *headers*, se utiliza una en especifico que esta oculta a propósito, ya que consiste en la llave de la API (*API Key*), que es una cadena de caracteres que identifican a la aplicación que esta tratando de utilizar el servicio, en este caso el sistema Actíivate. Esta clave se obtiene del departamento DGIU cuando este registra la información sobre el sistema, así como los servicios a los que tiene acceso y quien es el usuario


```

17 function createUserFromLDAP(ldapResponse, login) {
18     const data = parseResponse(ldapResponse.dn);
19     const name = data[0];
20     const isEstudent = data.findIndex(
21         element => element.toLowerCase() === 'estudiantes'
22     ) !== -1;
23     const emailSuffix = isEstudent ? 'est.umcc.cu' : 'umcc.cu';
24     const email = `${login}@${emailSuffix}`;
25     return {
26         login,
27         name,
28         email,
29         authorities: {
30             admin: false,
31         }
32     };
33 }
34
35

```

Figura 3.8 Algoritmo para crear un usuario a partir de la información que devuelve el dominio

En la Figura 3.9 se muestra el método utilizado para procesar la petición de *up-like* a un evento, que no es más que cuando un usuario presiona el botón de *like* o seguir, a un evento. En este método se comprueba que el usuario no este ya siguiendo el evento, de ser así, se agrega el evento a la lista de eventos que sigue el usuario y se incrementa el campo en el evento que denota la cantidad de usuarios que lo siguen. Es importante mencionar que aunque en la aplicación para los estudiantes también se tiene en cuenta y se evita que se pueda marcar como seguido un evento varias veces por el mismo usuario, es necesario realizar la comprobación del lado del servidor también ya que la arquitectura del sistema es de microservicios, por lo que el servidor puede recibir peticiones sin distinción de quien o que aplicación las haga, lo que lo hace susceptible a peticiones malignas hechas por usuarios o aplicaciones que no son las enmarcadas para el uso del sistema.

```

37 router.get(
38   '/api/events/up-like/:id',
39   checkAuth.required,
40   isBanned,
41   logTraces,
42   async (req, res) => {
43     const { id: eventId } = req.params;
44     const { id: userId } = req.payload;
45     const event = await Event.findById(eventId);
46     const user = await Users.findById(userId);
47     const followed = user.following.some(e => e.id == eventId);
48     if (!followed) {
49       user.following.push({
50         id: event._id,
51         nombre: event.nombre,
52         fecha: event.fecha,
53       });
54       await user.save();
55       event.likes = parseInt(event.likes) + 1;
56       const updatedEvent = await event.save();
57       res.status(201).json(updatedEvent);
58     } else {
59       res.status(200).json(null);
60     }
61   });
62

```

Figura 3.9 Método para actualizar la información de un evento cuando un usuario decide seguir el evento

Por último, en la Figura 3.10 se observa el método para modificar el rol de gestor de un usuario. Para que esta operación se efectúe con éxito, se deben tener en cuenta una serie de condiciones que deben cumplirse, las cuales son:

- El usuario que realiza la petición debe ser un administrador del sistema o el usuario *root*.
- El usuario que realiza la petición no se puede estar modificando a sí mismo.
- Si el usuario que realiza la petición es un administrador, no puede modificar a otro administrador.
- Solo el usuario *root* puede modificar a un administrador.

Si todas las condiciones anteriores se cumplen, se procede a la modificación del rol de gestor del usuario especificado.

```

64 router.put(
65   '/api/user/set-manager/:id',
66   checkAuth.required,
67   isBanned,
68   isRootOrAdmin,
69   logTraces,
70   (req, res) => {
71     const userId = req.params.id;
72     const user = req.body;
73     const { authorities: handlerAuthorities, id: handlerId } = req.payload;
74
75     const handlerIsDifferentFromUser = userId !== handlerId;
76     const handleIsNotRoot = !handlerAuthorities.root;
77     const handlerAndUserAreAdmins = handlerAuthorities.admin && user.authorities.admin;
78
79     if (handlerIsDifferentFromUser && handleIsNotRoot && handlerAndUserAreAdmins) {
80       return res.status(400).send({ error: ERROR_USER_DOES_NOT_MEET_REQUIRED_ROLES });
81     } else {
82       Users.findOne({ _id: userId }).then(dbUser => {
83         dbUser.authorities.manager = user.authorities.manager;
84         return dbUser.save();
85       }).then(savedUser => {
86         res.status(200).json(savedUser);
87       });
88     }
89   });
90

```

Figura 3.10 Método para actualizar la información de un usuario que hace de gestor para el sistema.

3.3 Pruebas de *Software*

3.3.1 Pruebas de Caja Blanca

Para aplicar esta prueba es necesario utilizar la técnica denominada prueba del camino básico que permite obtener una medida de la complejidad lógica de un diseño y definir un conjunto básico de caminos de ejecución que garantizan que se ejecuten al menos una vez cada sentencia del programa. Además, es necesario precisar la complejidad ciclomática, que es una métrica que proporciona una medida cualitativa de la complejidad lógica de un programa.

Para mostrar la aplicación de esta prueba en sistema se seleccionó el método para obtener la lista de atributos nominales de una fuente de datos.

V(G): Marca el límite de casos de prueba para un programa. Cuando $V(G) > 10$ la probabilidad de defectos en el módulo o programa crece mucho, entonces quizás se deba dividir el módulo. Los pasos para calcular y verificar V(G) son:

1. $V(G) = a - n + 2$.

2. $V(G) = r$.
3. $V(G) = c + 1$

Donde:

- a: número de arcos o aristas del grafo.
- n: número de nodos del grafo.
- r: número regiones cerradas del grafo.
- c: número de nodos prediados del grafo.

Prueba de caja blanca al método recibir la petición de autenticación al sistema

El método "recibir petición de autenticación al sistema" se puede dividir en secciones, en una primera parte se comprueba que se hayan enviado el nombre de usuario y la contraseña respectivamente, en caso de que no, se envía un error al usuario para que introduzca el dato faltante, luego se comprueba que el usuario no esté en la lista de usuarios bloqueados, en caso contrario se envía un mensaje de que el usuario esta bloqueado. Por último, se comprueba desde que aplicación se esta intentando autenticar el usuario, en caso de ser desde la aplicación para los estudiantes, solamente se pasa a comprobar las credenciales en el dominio, si es en la aplicación de gestión se comprueban las credenciales en el dominio y a demás se comprueban los roles del usuario.


```

120 router.post(
121   "/api/auth/signin/:authClient",
122   auth.optional,
123   (req, res) => {
124     const { user: { userName, password } } = req.body;
125     const { authClient } = req.params;
126     const bannedUsers = bannService.getBannedUsers();
127
128     if(!userName) {
129       return res.status(422).json({
130         error: ERROR_USERNAME_REQUIRED,
131       });
132     }
133     if(!password) {
134       return res.status(422).json({
135         error: ERROR_PASSWORD_REQUIRED,
136       });
137     }
138     if (bannedUsers.some(bannedUser => bannedUser === userName)) {
139       return res.status(400).json({
140         error: ERROR_USER_BANNED,
141       });
142     }
143     if (authClient === 'blog-site') {
144       checkUserCredentials(res, userName, password, false);
145     } else {
146       checkUserCredentials(res, userName, password, true);
147     }
148   });
149

```

Figura 3.2.1.1 Método para recibir la petición de autenticación al sistema

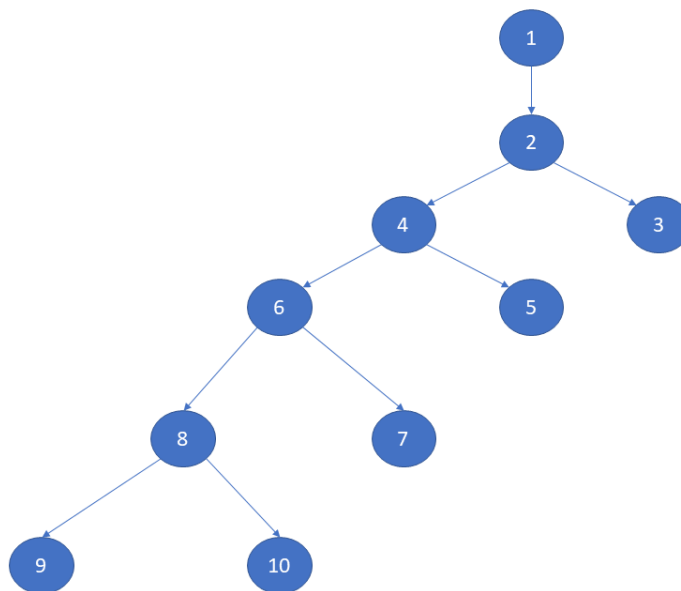


Figura 3.2.1.2 Grafo de caminos posibles

Resultados al aplicar la prueba

$a = 9, n = 10, r = 1$ y $c = 4$

$V(G) = a - n + 2 = 9 - 10 + 2 = 1$

$$V(G) = r = 1$$

$$V(G) = c + 1 = 4 + 1 = 5$$

Camino independientes:

- 1, 2, 4, 6, 8, 9
- 1, 2, 4, 6, 8, 10
- 1, 2, 4, 6, 7
- 1, 2, 4, 5
- 1, 2, 3

Tabla 3.2.1.1 Casos de prueba para el método recibir la petición de autenticación al sistema

Camino	Descripción	Resultado esperado
1, 2, 4, 6, 8, 9	Se recibe la petición de autenticación, se comprueba que se haya recibido un nombre de usuario y una contraseña, se comprueba que el usuario no esté bloqueado, se pasa a autenticar el usuario con el dominio	Se pasa a comprobar las credenciales del usuario con el dominio especificando que no hay que comprobar los roles del usuario
1, 2, 4, 6, 8, 10	Se recibe la petición de autenticación, se comprueba que se haya recibido un nombre de usuario y una contraseña, se comprueba que el usuario no esté bloqueado, se pasa a autenticar el usuario con el dominio	Se pasa a comprobar las credenciales del usuario con el dominio especificando que si hay que comprobar los roles del usuario
1, 2, 4, 6, 7	Se recibe la petición de autenticación, se comprueba que se haya recibido un nombre de usuario y una contraseña,	Se manda un mensaje de error al usuario de que esta boqueado en el sistema

	se comprueba que el usuario no esté bloqueado	
1, 2, 4, 5	Se recibe la petición de autenticación, se comprueba que se haya recibido un nombre de usuario y una contraseña	Se manda un mensaje de error al usuario de que debe proporcionar una contraseña
1, 2, 3	Se recibe la petición de autenticación, se comprueba que se haya recibido un nombre de usuario	Se manda un mensaje de error al usuario de que debe proporcionar un nombre de usuario

Prueba de caja blanca al método para confirmar las credenciales de un usuario

El método “confirmar las credenciales de un usuario” se encarga de utilizar el microservicio de comunicación con el dominio LDAP de la universidad para verificar que el usuario que está en el proceso de autenticación sea un usuario real de la universidad, si el servicio responde con un código 200, significa que el usuario existe por lo que se pasa a verificar si existe en la base de datos del sistema, de ser así, se comprueba si se está entrando a la aplicación de gestión para ver si es necesario comprobar los roles del usuario, si no es necesario, se envía al usuario el token de autenticación para utilizar en las peticiones futuras. Si el usuario no existe en la base de datos del sistema, se considera que es su primera vez utilizándolo por lo que se pasa a crear un registro con sus datos para su posterior uso. Esta parte del proceso evita la necesidad de un flujo de registro en el que el usuario tenga que introducir todos sus datos y permite tener una sola fuente de información con respecto a los usuarios. Si por el contrario, el servicio del dominio no responde con un código 200, se procede a una última verificación, ya que el usuario que está intentando autenticarse puede ser el usuario primario o *root*, que es un usuario que se crea inicialmente con el propósito de poder gestionar los usuarios, para esto, se busca el nombre de usuario en la base de datos y si es correcto se verifica también la contraseña, si estas verificaciones son correctas se autentica el usuario, sino, no quedan más alternativas válidas por lo que se devuelve al usuario un error de que las credenciales enviadas son inválidas.

```

89 function checkUserCredentials(res, userName, password, authenticatingManager) {
90   ldapService.checkUserCredentials(userName, password).then(ldapResponse => {
91     if (ldapResponse.code === 200) {
92       Users.findOne({ login: userName })
93         .then(user => {
94           if (user) {
95             if (authenticatingManager) {
96               return checkUserAuthorities(res, user);
97             }
98             return res.status(200).json({...user.toAuthJSON() });
99           } else if (!authenticatingManager) {
100             const ldapUser = ldapService.createUserFromLDAP(ldapResponse, userName);
101             const user = new Users(ldapUser);
102             user.save().then(savedUser => {
103               return res.status(200).json({...savedUser.toAuthJSON() });
104             });
105           }
106         });
107     } else {
108       Users.findOne({ login: userName }).then(user => {
109         if (!user || !user.validatePassword(password)) {
110           return res.status(401).json({ error: ERROR_USERNAME|PASSWORD_INVALID });
111         }
112         else {
113           return res.status(200).json({...user.toAuthJSON() });
114         }
115       });
116     }
117   })
118 }
119

```

Figura 3.2.2.1 Método para confirmar las credenciales de un usuario

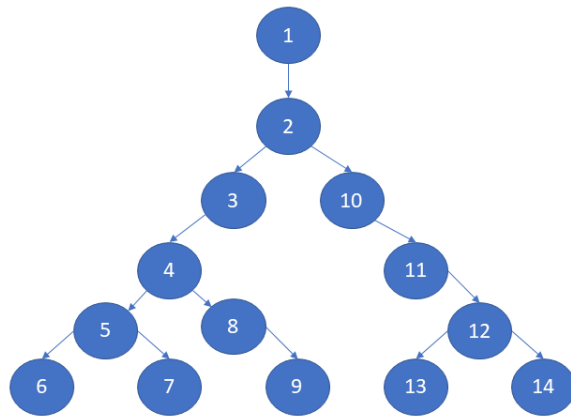


Figura 3.2.2.2 Grafo de caminos posibles

Resultados al aplicar la prueba

$$a = 13, n = 14, r = 1 \text{ y } c = 5$$

$$V(G) = a - n + 2 = 13 - 14 + 2 = 1$$

$$V(G) = r = 1$$

$$V(G) = c + 1 = 5 + 1 = 6$$

Caminos independientes:

- 1, 2, 3, 4, 5, 6
- 1, 2, 3, 4, 5, 7
- 1, 2, 3, 4, 8, 9
- 1, 2, 10, 11, 12, 13
- 1, 2, 10, 11, 12, 14

Tabla 3.2.2.1 Casos de prueba para el método para confirmar las credenciales de un usuario en LDAP

Camino	Descripción	Resultado esperado
1, 2, 3, 4, 5, 6	Se comprueban las credenciales del usuario en el dominio, si el dominio responde correctamente, se busca al usuario en la DB, si el usuario existe, está entrando al sitio de gestión y posee las credenciales necesarias.	El usuario es autenticado correctamente y se le da acceso a la aplicación.
1, 2, 3, 4, 5, 7	Se comprueban las credenciales del usuario en el dominio, si el dominio responde correctamente, se busca al usuario en la DB, si el usuario existe y no está entrando al sitio de gestión	El usuario es autenticado correctamente y se le da acceso a la aplicación.
1, 2, 3, 4, 8, 9	Se comprueban las credenciales del usuario en el dominio, si el dominio responde correctamente, se busca al usuario en la DB, si el usuario no existe y no está entrando al sitio de gestión, se crea una entrada en la DB	El usuario es autenticado correctamente y se le da acceso a la aplicación.

1, 2, 10, 11, 12, 13	Se comprueban las credenciales del usuario en el dominio, si el dominio responde con un error, se busca al usuario en la DB, si el usuario existe y la contraseña es valida	El usuario es autenticado correctamente y se le da acceso a la aplicación.
1, 2, 10, 11, 12, 14	Se comprueban las credenciales del usuario en el dominio, si el dominio responde con un error, se busca al usuario en la DB, si el usuario no existe o la contraseña es valida	Se envía un mensaje de error al usuario

3.3.2 Pruebas de caja negra

Durante el proceso de creación de la solución, en la medida que se desarrolló el sistema, se crearon un grupo de pruebas funcionales para comprobar aspectos importantes, estas quedaron reflejadas en la documentación del sistema.

Prueba a la funcionalidad de Crear un comentario

Tabla 3.2.1 Prueba funcional #1 a Crear un comentario

Caso de prueba: Comentar un evento correctamente
Descripción: Se accede a la vista de detalles de un evento y se agrega un comentario
Condiciones de ejecución: El usuario está autenticado
Entrada: comentario
Resultado esperado: Se agrega el comentario a la lista correctamente
Evaluación de la prueba: Satisfactoria.

Tabla 3.2.2 Prueba funcional #2 a Crear un comentario

Caso de prueba: Comentar un evento sin estar autenticado
Descripción: Se accede a la vista de detalles de un evento y se agrega un comentario
Condiciones de ejecución: El usuario no está autenticado
Entrada: comentario
Resultado esperado: Se muestra el dialogo de autenticación, si el usuario es autenticado correctamente, el comentario se agrega a la lista.
Evaluación de la prueba: Satisfactoria.

Tabla 3.2.3 Prueba funcional #3 a Crear un comentario

Caso de prueba: Comentar un evento con palabras que están marcadas como prohibidas
Descripción: Se accede a la vista de detalles de un evento y se agrega un comentario
Condiciones de ejecución: El usuario esta autenticado
Entrada: comentario que contenga una palabra marcada como prohibida
Resultado esperado: Se agrega el comentario a la lista, pero no se muestra
Evaluación de la prueba: Satisfactoria.

Prueba a la funcionalidad Autenticación en la aplicación de gestión

Tabla 3.2.3 Prueba funcional #1 a Autenticación en la aplicación de gestión

Caso de prueba: Autenticación de un usuario permitido en la aplicación de gestión.
Descripción: Se accede a la aplicación de gestión y se introducen las credenciales de un usuario permitido
Condiciones de ejecución: El usuario es un gestor o administrador en el sistema, o es el usuario <i>root</i>
Entrada: las credenciales del usuario
Resultado esperado: Se autentica el usuario y se le da acceso a la aplicación
Evaluación de la prueba: Satisfactoria.

Tabla 3.2.3 Prueba funcional #2 a Autenticación en la aplicación de gestión

Caso de prueba: Autenticación de un usuario no permitido en la aplicación de gestión
Descripción: Se accede a la aplicación de gestión y se introducen las credenciales de un usuario no permitido
Condiciones de ejecución: El usuario no es un gestor o administrador en el sistema
Entrada: las credenciales del usuario
Resultado esperado: Se muestra un error de que el usuario no posee los roles necesarios para acceder al sistema
Evaluación de la prueba: Satisfactoria.

Prueba a la funcionalidad Editar un evento en la aplicación de gestión

Tabla 3.2.4 Prueba funcional #1 a Editar un evento en la aplicación de gestión

Caso de prueba: Editar un evento existente con una fecha de inicio pasada
Descripción: Se accede a la vista de detalles de un evento y se modifica uno de los campos
Condiciones de ejecución: El usuario esta autenticado y posee el rol de gestor y es el gestor del evento
Entrada: se modifica algún campo en el formulario
Resultado esperado: Se muestra un error de que la fecha de inicio no puede ser una fecha anterior al día de edición y se deshabilita el botón de guardar.
Evaluación de la prueba: Satisfactoria.

Tabla 3.2.4 Prueba funcional #2 a Editar un evento en la aplicación de gestión

Caso de prueba: Editar un evento existente cuando el usuario no es el gestor del evento
Descripción: Se accede a la vista de detalles de un evento y se modifica uno de los campos
Condiciones de ejecución: El usuario esta autenticado y posee el rol de gestor y no es el gestor del evento
Entrada: se modifica algún campo en el formulario
Resultado esperado: Se deshabilita el botón de guardar y se muestra un error de que el usuario no es el gestor del evento.
Evaluación de la prueba: Satisfactoria.

3.3.3 Pruebas de sistemas web

Esta prueba se realizó con el fin de medir requerimientos inherentes a las aplicaciones web y sus características, como son el rendimiento, tiempo de carga, entre otras. Para esta prueba se utilizó la herramienta Lighthouse.

La figura 3.1 muestra las métricas utilizadas por la herramienta Lighthouse para medir la calidad de una aplicación web, estas son, el rendimiento, que se trata de los tiempos que toma la aplicación antes de que se renderice algún contenido o está lista para que el usuario interactúe con ella. La accesibilidad mide mayormente la integración con los *screen readers* o lectores de pantallas, estos son software utilizados por las personas con discapacidades visuales o que le impiden interactuar con los dispositivos como el resto de las personas, para una correcta integración se utilizan etiquetas especiales que permiten a los softwares leer al usuario la página y de esta forma este pueda entender la disposición y el objetivo de los componentes y como interactuar con ellos. Las *Best Practices* o buenas prácticas son soluciones a problemas comunes que están probadas son las mejores posibles hasta el momento, ya sea que aumentan el rendimiento, la usabilidad o el mantenimiento del código, esta métrica se encarga de verificar la implantación de dichas prácticas en el sitio. El *SEO* u optimización para motores de búsqueda (Search Engine Optimization), mide el uso de las etiquetas de *keywords* o palabras claves existentes en el sitio web que ayudan a los *web crawlers* o indexadores web que dichos motores emplean para analizar el contenido de la página y así determinar el objetivo y poder posicionarlo mejor en los resultados de las búsquedas. Por último, el *Progressive Web App* o aplicación web progresiva son sitios web que utilizan la tecnología de *Service Workers* para indexar y almacenar en el dispositivo del usuario todos los archivos necesarios para el funcionamiento del sitio completamente desde el dispositivo, lo que permite que solo se necesiten hacer las peticiones necesarias para obtener los datos e incluso que la aplicación funcione sin estar conectada a internet. Esta métrica mide el nivel de implementación de dichas técnicas.

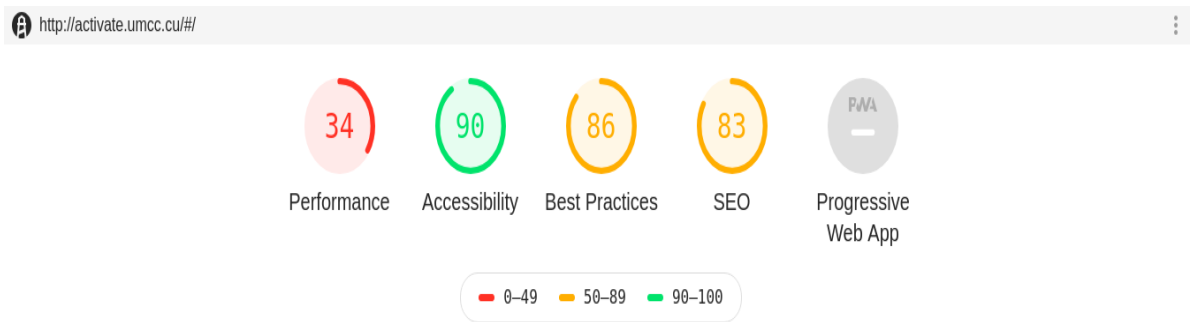


Figura 3.1 Resultados generales de las mediciones aplicadas por la herramienta Lighthouse

En la Figura 3.2 se observan los parámetros utilizados por la herramienta Lighthouse para medir el rendimiento de una aplicación web. La métrica *First Contentful Paint* o Primer Dibujo de contenido mide cuanto tiempo pasa desde que el navegador realiza la petición hasta que aparece en la pantalla el maquetado de la aplicación con los estilos aplicados. Por otro lado, la métrica *First CPU idle* o primer estado ocioso de la CPU mide cuanto tiempo pasa la CPU procesando todos los archivos y la lógica necesaria para el primer pintado con sentido, o sea, el momento en que la aplicación esta funcional.

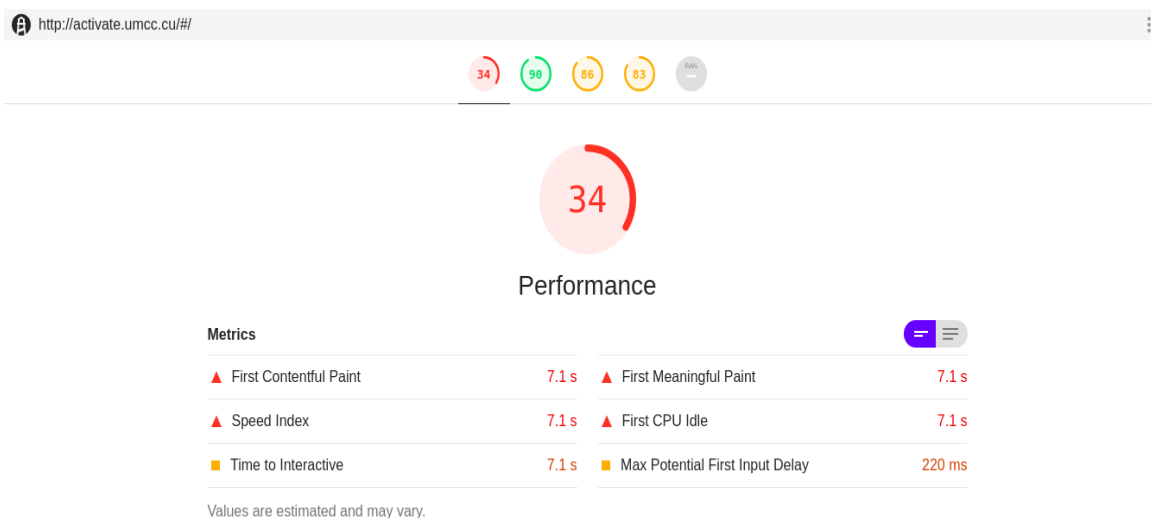


Figura 3.2 Resultados específicos referentes al rendimiento

Conclusiones parciales

Las pruebas que se realizaron a partir de las técnicas de caja blanca y caja negra fueron de gran importancia para demostrar el buen funcionamiento del software, debido a que la evaluación de las pruebas mostró resultados satisfactorios. Las pruebas también detectaron posibles mejoras en el sistema, lo que constituye al mismo tiempo, una validación del software desarrollado y una muestra de lo que queda por hacer. Con la descripción de los métodos mas importantes del sistema se puede validar la confiabilidad del sistema y con el análisis de sus interfaces, mostrar que es una solución factible, no solo desde el punto de vista de las funcionalidades, sino también de cara a un usuario que, aunque no tiene por qué conocer en detalle el sistema, sí se espera que lo utilice de forma continua. La estimación realizada establece que es posible realizar el desarrollo de la solución dentro del periodo enmarcado.

CONCLUSIONES

Con el desarrollo de esta investigación se arribó a las siguientes conclusiones:

- La solución planteada y desarrollada cumple con los requisitos del cliente y debe ser capaz de aumentar la difusión de los eventos universitarios.
- La investigación de los antecedentes, el estado actual de la temática y la bibliografía relacionada, permitió aportar los elementos necesarios para dar solución a la problemática planteada.
- Se elaboró una solución para la difusión de los eventos universitarios que cumple con todos los requisitos exigidos por el cliente y cuenta con la aceptación del mismo.
- Las pruebas que se realizaron y el análisis de los casos de estudio demostraron que el sistema desarrollado posee un nivel de calidad y estabilidad suficientes para la explotación del mismo.

RECOMENDACIONES

Desde el punto de vista del alcance del presente trabajo y teniendo en cuenta el momento de desarrollo del mismo, se proponen las siguientes recomendaciones:

- Implementar funcionalidades estadísticas que permitan una mejor comprensión de la interacción de los estudiantes con el sistema, que tipos de eventos son más seguidos, etc.
- Continuar el desarrollo de la aplicación móvil para que se convierta en la plataforma principal.
- Aumentar la capacidad multimedia del sistema para que permita videos y una descripción de eventos enriquecida.

BIBLIOGRAFÍA

- 2016.** ¿QUÉ ES BPMN Y PARA QUÉ SIRVE? *NEXTECH*. [En línea] 16 de Septiembre de 2016. [Citado el: 20 de Abril de 2019.] <http://nextech.pe/que-es-bpmn-y-para-que-sirve>.
- 2014.** ¿Qué es el mailing o e-mail marketing? *MisterBox*. [En línea] 18 de agosto de 2014. [Citado el: 20 de Abril de 2019.] <http://www.misterbox.es/que-es-el-mailing-o-e-mail-marketing>.
- Banker, Kyle, y otros. 2016.** *MongoDB in Action Second Edition*. 2016.
- Brown, Ethan. 2016.** *Learning JavaScript*. 2016.
- Holmes, Simon. 2013.** *Mongoose for Application Development*. 2013.
- Kholodkov, Valery. 2015.** *Nginx Essentials*. 2015.
- Mardan, Azat. 2014.** *Pro Express.js*. 2014.
- Mead, Andrew. 2018.** *Learning Node.js Development*. 2018.
- multiples, Autores. 2018.** *Digital Marketing & Social Media*. 2018.
- NGINX.** Welcome to NGINX Wiki! *NGINX*. [En línea] [Citado el: 19 de Abril de 2019.] <https://www.nginx.com/resources/wiki/#>.
- Node.js.** About Node.js. *Node.js*. [En línea] [Citado el: 19 de Abril de 2019.]
- Nuestra historia. *Univerisad de Matanzas Camilo Cienfuegos*. [En línea] http://www.umcc.cu/nuestra_historia.
- OpenSource.** What is Docker? *opensource.com*. [En línea] [Citado el: 19 de Abril de 2019.] <http://opensource.com/what-is-docker>.
- Pethuru, Raj, Raman, Anupama y Subramanian, Harihara. 2017.** *Architectural Patterns*. 2017.
- Raj, Pethuru y Singh, Vinod. 2015.** *Learning Docker*. 2015.
- Restrepo Jimenez, Luis Carlos y Muñoz Hernandez, Helmer. 2016.** *Marketing digital: una mirada al pasado, presente y futuro*. 2016.
- Rouse, Margaret.** MongoDB. *Tech Target*. [En línea] [Citado el: 19 de Abril de 2019.]
- Schwaber, Ken y Sutherland, Jeff. 2017.** *The Scrum Guide*. 2017.

Seshadri, Shyam. 2018. *Angular Up & Running.* 2018.

Thornley, Louise y Marsh, Kate. 2010. *What Works in Social Marketing to Young People?* 2010.