UNIVERSIDAD TECNOLÓGICA CENTROAMERICANA

FACULTAD DE INGENIERÍA

PRÁCTICA PROFESIONAL

LAUREATE INTERNATIONAL UNIVERSITIES

PREVIO A LA OBTENCIÓN DEL TÍTULO

INGENIERO EN SISTEMAS COMPUTACIONALES

PRESENTADO POR:

11541292 SAMIR GERARDO FACUSSÉ ESPINAL

ASESOR: LIC. TANIA LUCILA MEZA AMADOR

CAMPUS TEGUCIGALPA; 28 DE OCTUBRE, 2020

RESUMEN EJECUTIVO

El presente informe tiene el propósito de exponer el trabajo realizado en la empresa Laureate Education, en él se describen las actividades completadas durante la práctica profesional realizada del 18 de mayo al 16 de octubre del 2020. La práctica se realizó en el departamento de Tecnologías de Información en la sección de sistemas de gestión del aprendizaje (LMS) en la subsección de arquitectura, aquí el practicante ha realizado tareas relacionadas al control de calidad de SUMADI como pruebas de regresión, pruebas de carga y de estrés.

El proyecto más importante en el que se trabajó durante el desarrollo de la práctica fue la aplicación SUMADI, esta es una solución que utiliza inteligencia artificial para monitorear a los estudiantes al momento de realizar exámenes en línea. En el departamento de arquitectura se realizaron numerosas pruebas de regresión para poder capturar errores en la aplicación, también se trabajó en conjunto con el departamento de desarrollo para poder minimizar la cantidad de problemas encontrados en los nuevos lanzamientos del producto.

El equipo de arquitectura también ha tenido que realizar pruebas de carga para determinar los límites de SUMADI en términos de infraestructura, es decir, la carga que puede ser manejada por los servidores, las bases de datos, y servicios externos que son utilizados en el funcionamiento de la aplicación. También se han ejecutado pruebas de estrés con el propósito de validar nuevas mejoras que han sido implementadas para incrementar la capacidad de SUMADI. Todas estas fueron realizadas utilizando programas creados en JMeter, una herramienta que simula los llamados a las funciones de SUMADI.

El equipo también ha sido responsable por manejar los servicios administrativos de la nube de Amazon, entre estos se puede mencionar Codebuild, una herramienta utilizada para administrar los nuevos lanzamientos de SUMADI, y CloudFormation, un servicio que permite crear la infraestructura necesaria por medio de archivos de texto.

Por último se ha trabajado para mantener los estándares de seguridad de la empresa por medio de herramientas como Trend Micro y tenable.io para encontrar vulnerabilidades en SUMADI y proveer soluciones viables para las mismas.

LISTA DE SIGLAS Y GLOSARIO

AI	Inteligencia Artificial.
API	Interfaz de Programación de Aplicaciones
CRAI	Centro de Recursos para el Aprendizaje y la Investigación
FI	Facultad de Ingeniería
ICFES	Instituto Colombiano de Educación Superior
IT	Tecnología de Información
LMS	Sistema de Gestión de Aprendizaje
PP	Práctica Profesional
REST	Transferencia de Estado Representacional

API	Interfaz de programación de aplicaciones, es utilizada para desarrollar e integrar el software de las aplicaciones (Gazarov, 2019).
Branches	Puntero movible que apunta a un commit (Heller, 2018).
Middleware	Software que ayuda a una aplicación a comunicarse con otras (Introducción a Express/Node, s. f.).
PI Planning	Evento en el que los grupos planifican las actividades por realizar para la siguiente entrega del producto (What is SAFe, s. f.).
REST	Un estilo de arquitectura que establece estándares para la comunicación entre los sistemas en la web (What Is REST?, s. f.).
SAFe	Metodología ágil diseñada para mejorar la productividad y eficiencia de las empresas (What is SAFe, s. f.).
SCRUM	Marco de trabajo en el que se pueden desarrollar y mantener productos complejos (Schwaber & Sutherland, 2016).
Sprint	Periodo de trabajo entre las entregas en SCRUM (Schwaber & Sutherland, 2016).

INTRODUCCIÓN

Cada año se puede ver cómo la tecnología toma un rol indispensable en el desarrollo de la humanidad, entrelazándose con todas las áreas de la sociedad, desde la salud y los negocios hasta la educación y la cultura. Esta actúa como una herramienta que ha logrado ayudar a muchos a superar limitantes como la distancia, el tiempo y la situación económica para cumplir con metas académicas, profesionales y personales. En vista de la presente crisis ante la pandemia, las personas se han visto obligadas a recurrir a vivir sus vidas de manera virtual con clases en línea y trabajos remotos.

Esto ha forzado a las escuelas y universidades a tomar en cuenta ahora más que nunca la posibilidad de que sus alumnos exploten las vulnerabilidades de la educación en línea. Para poder enfrentar este problema, Laureate ha estado trabajando en un sistema de supervisión a base de inteligencia artificial llamado SUMADI, este logra monitorear las acciones de los estudiantes durante sus exámenes, eliminando la necesidad de maestros para supervisar a los alumnos. De esta manera se eleva el estándar de la educación, mitigando el riesgo de que los alumnos puedan engañar al sistema.

En este informe se presenta una descripción de las generalidades de la empresa, Laureate International Universities, también se describe el puesto que se ocupó en esta práctica profesional con sus objetivos generales y específicos.

En el marco teórico se habla de las metodologías utilizadas en la empresa junto con las herramientas y avances tecnológicos que han sido utilizados en el desarrollo de esta aplicación.

En la sección de desarrollo se podrán ver los detalles de las actividades que se realizaron a lo largo de esta práctica profesional.

II. GENERALIDADES DE LA EMPRESA

2.1 DESCRIPCIÓN DE LA EMPRESA

Creemos en el poder de la educación para cambiar vidas. Laureate Education, Inc. Es un proveedor líder de educación superior de calidad a través de su red global de instituciones de educación superior otorgadoras de títulos y de sus relaciones colaborativas de vanguardia con empresas líderes de la industria (*Sobre Laureate*, s. f.).

Laureate se enorgullece de ser parte de un movimiento global dedicado a hacer del mundo un lugar mejor. Estamos comprometidos con tener propósito en todo lo que hacemos, y de transmitir conocimientos y habilidades que ayuden a nuestros estudiantes a convertirse en agentes de cambio y apéndices de por vida. Ya sea a través de nuestra red de clínicas de salud dirigidas por estudiantes que ofrecen servicios gratuitos o de bajo costo a más de 150,000 pacientes cada año, o con nuestros ganadores del premio *Here for Good* que han recibido reconocimiento mundial por su excepcional impacto social, nuestras instituciones, estudiantes, personal administrativo y docentes juegan un papel importante en cambiar el mundo (*Sobre Laureate*, s. f.).

Estamos orgullosos de ser una de las Corporaciones B Certificadas® y Corporaciones de Beneficios Públicas (PBC) más grandes del mundo. Además, nuestra PBC es la empresa más grande de este tipo que se cotiza públicamente en cualquier bolsa de valores del mundo. Ser una PBC y una B Corp nos permite integrar legalmente nuestra misión en todo lo que hacemos, lo que nos empuja a encontrar formas significativas de mejorar nuestro desempeño social y ambiental. Nos ayuda a equilibrar nuestro propósito y beneficio, cerrando la brecha entre nuestra comunidad académica y nuestras operaciones comerciales. Nos esforzamos por ser no solo los mejores del mundo, sino también los mejores para el mundo. Sin embargo, lo que nos diferencia es la escala y la plataforma que tenemos para equipar a nuestros estudiantes a nivel mundial con las habilidades y la conciencia para ser embajadores en este movimiento, preparándolos para liderar empresas y comunidades innovadoras que están decididas a efectuar un cambio global (*Sobre Laureate*, s. f.).

2.2 DESCRIPCIÓN DEL DEPARTAMENTO

El departamento de Tecnología de Información (IT) de Laureate se encarga de dar mantenimiento y soporte a las tecnologías y aplicaciones que son utilizadas en las universidades de la red de Laureate. Se trabaja con la plataforma de Blackboard, la cual les facilita a varios estudiantes y profesores la tarea de realizar cursos en línea de manera parcial o total proporcionando un espacio en el que se pueden compartir recursos y completar asignaciones. Considerando el crecimiento en el número de asignaciones virtuales, se ha dado prioridad al desarrollo de aplicaciones que aseguren la integridad del trabajo de los estudiantes.

2.3 OBJETIVOS DEL PUESTO

2.3.1 OBJETIVO GENERAL

Colaborar en el desarrollo y progreso de la aplicación de supervisión basada en Inteligencia Artificial (A.I.), SUMADI, asegurando que las nuevas versiones de la misma cumplan con los estándares de seguridad y calidad ya establecidos.

2.3.2 OBJETIVOS ESPECÍFICOS

- Asistir en el proceso de control de calidad de las aplicaciones desarrolladas en el Learning Management System (LMS).
- Participar en la creación de documentación de estándares de Arquitectura para todo el equipo de LMS.
- Realizar tareas administrativas de operaciones en la nube en los servidores de Amazon Web Services (AWS).

III. MARCO TEÓRICO

A continuación se presentan varios conceptos y herramientas que han sido utilizados en el desarrollo de la aplicación SUMADI para poder tener un mejor entendimiento del propósito y funcionamiento de la misma.

Se puede decir que una de las consecuencias de la globalización en el mundo actual han sido los grandes avances en la tecnología y la comunicación, uno de estos es el auge de la educación virtual, la cual ha sido facilitada por el Internet. Esta herramienta provee un medio capaz de asegurar el transporte de información y la interacción, permitiendo la formación de una comunidad interconectada de estudiantes y profesores (Gallardo & Buleje, 2011).

Este cambio ha tenido un enorme impacto en la educación durante el siglo XXI, transformando la relación entre el estudiante y el profesor de un esquema vertical en el que el conocimiento es proporcionado únicamente por el profesor a una forma circular y participativa que incentiva a los alumnos a tomar una actitud proactiva y autónoma en el proceso de aprendizaje (Gallardo & Buleje, 2011).

Para poder implementar este modelo, muchas instituciones han decidido utilizar los LMS, o Sistema para la Gestión de Aprendizaje. Estos sistemas permiten la creación de aulas virtuales en las que se pueden realizar todas las actividades de una clase normal, como realizar evaluaciones, intercambiar archivos y comunicarse con otros usuarios (Universidad Internacional de Valencia, s. f.).

En muchos casos las nuevas herramientas para la educación virtual han sido incorporadas para apoyar el proceso de la formación presencial. Sin embargo, la creación de estos sistemas que actúan como plataformas de software especializadas ha facilitado la transición a un modelo de estudios a distancia. Normalmente, estos siguen los modelos tradicionales de instalación, esto implica que las instituciones deben contar con un equipo de IT que se encargará de ofrecer soporte a los usuarios, además se debe contar con la infraestructura necesaria para poder disponer del almacenamiento, las redes y recursos necesarios para el funcionamiento de un LMS (Herrera-Cubides et al., 2019).

Los costos pueden disuadir a muchos de decidir invertir en un LMS, pero estos traen varios beneficios como los que nos menciona la Universidad Internacional de Valencia (s. f.):

- Eliminan las limitantes de la distancia y el tiempo, permitiéndole a los estudiantes estudiar en su propio tiempo y a su propio ritmo.
- Son fáciles de usar, en teoría cualquier persona con un nivel básico de conocimientos sobre el funcionamiento del Internet sería capaz de usarlos.
- Reduce los costos y maximiza la flexibilidad de las capacitaciones de las personas.
- Logra un aprendizaje constante y actualizado gracias a la constante interacción entre los alumnos y sus profesores.

Para lograr todas estas ventajas, la Universidad Internacional de Valencia (s. f.) ha listado las características que todos los entornos virtuales de aprendizaje han adoptado:

- Centralizan y automatizan la gestión del aprendizaje.
- Debe ser flexible, permitiendo que la plataforma pueda incorporar los planes de estudio, contenidos y estilos pedagógicos de cualquier organización.
- Proveen servicios guiados que elevan el grado de interactividad con el estudiante, permitiéndole a este convertirse en el protagonista de su propio aprendizaje.
- Permiten el uso de contenido creado por terceros, haciendo que sea posible reutilizar el conocimiento de otras fuentes.
- Soportan cantidades variables de usuarios según las necesidades de la institución.
- Cuentan con varias funcionalidades, adaptando la plataforma a los requerimientos y necesidades de los usuarios.
- Deben poder ser integradas con otras aplicaciones empresariales utilizadas por otros departamentos de las empresas, esto facilita el proceso de contabilizar el impacto, la eficacia y el costo de las actividades de formación.

Existen varias opciones al momento de escoger un LMS, el más utilizado por Laureate es Blackboard el cual es conocido por lograr convertir el aprendizaje en atractivo, accesible y valioso, proporcionándole a los estudiantes la oportunidad de aprender a su propio ritmo y estilo (Universidad Internacional de Valencia, s. f.).

El equipo de desarrollo de la arquitectura de LMS en Laureate de Honduras está dividido en cuatro grupos, cada grupo trabaja con la metodología Scrum pero se utiliza la metodología SAFe para coordinar el trabajo de todo el departamento.

Scrum es un marco de trabajo en el que se pueden desarrollar y mantener productos complejos. Este marco se basa en la teoría del empirismo, este dice que la experiencia es la fuente del conocimiento y las decisiones se deben tomar basándose en el conocimiento. Para poder implementar esta teoría Schwaber & Sutherland (2016) mencionan tres principios que se deben seguir:

- Transparencia: Todo el equipo debe conocer y estar de acuerdo con los conceptos establecidos, esto hace que se pueda crear un estándar en el que todos entienden las metas y todos están de acuerdo en cómo se verá el producto terminado.
- Inspección: Se debe supervisar el progreso de los proyectos de manera frecuente para poder detectar problemas de manera temprana. Es importante notar que estas inspecciones no deben ser lo suficientemente frecuentes como para interferir con el trabajo.
- Adaptación: Es importante que el proceso o el material que está siendo utilizado sea modificado en caso de que se encuentre una desviación de los estándares establecidos durante alguna de las inspecciones. Para dar mayor flexibilidad al desarrollo del producto el proceso es dividido en iteraciones o Sprints, comúnmente estas duran dos semanas.

Para poder seguir estos principios se deben realizar cuatro eventos formales a lo largo de cada iteración, según Schwaber & Sutherland (2016) estos son:

- Planificación del Sprint: Reunión en la cual se especifican las metas que se desean cumplir durante ese sprint.

- Scrum Diario: Este evento es usado para monitorear el progreso del equipo, cada miembro da un avance de su trabajo de ese día y anuncia la tarea que planea desarrollar hasta el siguiente Scrum Diario.
- Revisión del Sprint: En esta reunión se presenta un reporte de las tareas terminadas y las que no se pudieron lograr, esta facilita un espacio para recibir retroalimentación y fomentar la colaboración.
- Retrospectiva del Sprint: Es una oportunidad para que el equipo evalúe su desempeño y encuentre posibilidades de mejora que serán implementadas en el siguiente sprint.

SAFe, al igual que Scrum, es una de las metodologías ágiles, está diseñada para ayudar a las empresas a lograr alcanzar sus metas dentro de las fechas propuestas. Se ha visto que esta metodología ha logrado grandes mejoras en la productividad y calidad del trabajo de muchas empresas a nivel mundial (*What is SAFe*, s. f.). A diferencia de Scrum, que trabaja con grupos pequeños, SAFe se utiliza para coordinar el trabajo de grupos grandes, una de las herramientas que usa para esto es el PI Planning.

El PI Planning es un evento importante en el que los grupos planifican las actividades que se realizarán hasta la siguiente entrega del producto que se está desarrollando. Según (Beck et al., 2001), la manera más eficiente de transmitir información en un equipo de desarrollo es por medio de conversaciones cara a cara, esto es una parte vital de las sesiones de PI Planning, las cuales incluso hacen uso de video llamadas para lograrlo.(«PI Planning», s. f.).

Otro aspecto importante de la filosofía de trabajo del equipo es el uso de un Wiki como una manera para compartir la información que puede ser útil para todos los miembros del equipo.

En la página *Wikis* (2012) se define una Wiki como una herramienta colaborativa que le permite a todos sus usuarios contribuir y modificar las páginas que se encuentran dentro de la misma, es decir, funciona como una página web con un sistema de edición.

A diferencia de una wiki pública como Wikipedia, las wikis en entornos empresariales deben utilizar ciertas herramientas para asegurar que el contenido que es agregado a estas sea útil y confiable.

Por esto Yeo & Arazy (2012) aseguran que una de las maneras que ha mostrado ser igual de efectiva en ambientes corporativos y públicos ha sido el uso de herramientas para el control de calidad hecho por otros miembros.

Otra de las herramientas utilizadas por el equipo LMS de Laureate es Github. Github actúa como un repositorio de código que funciona a través de servidores en la nube. Por medio de Git, un sistema de control de versiones, Github tiene la capacidad de guardar proyectos y mantener una bitácora de los cambios que han sido realizados sobre estos (Heller, 2018).

En la Ilustración 1 se puede observar un ejemplo de la página en la que Github guarda el historial de los cambios realizados.

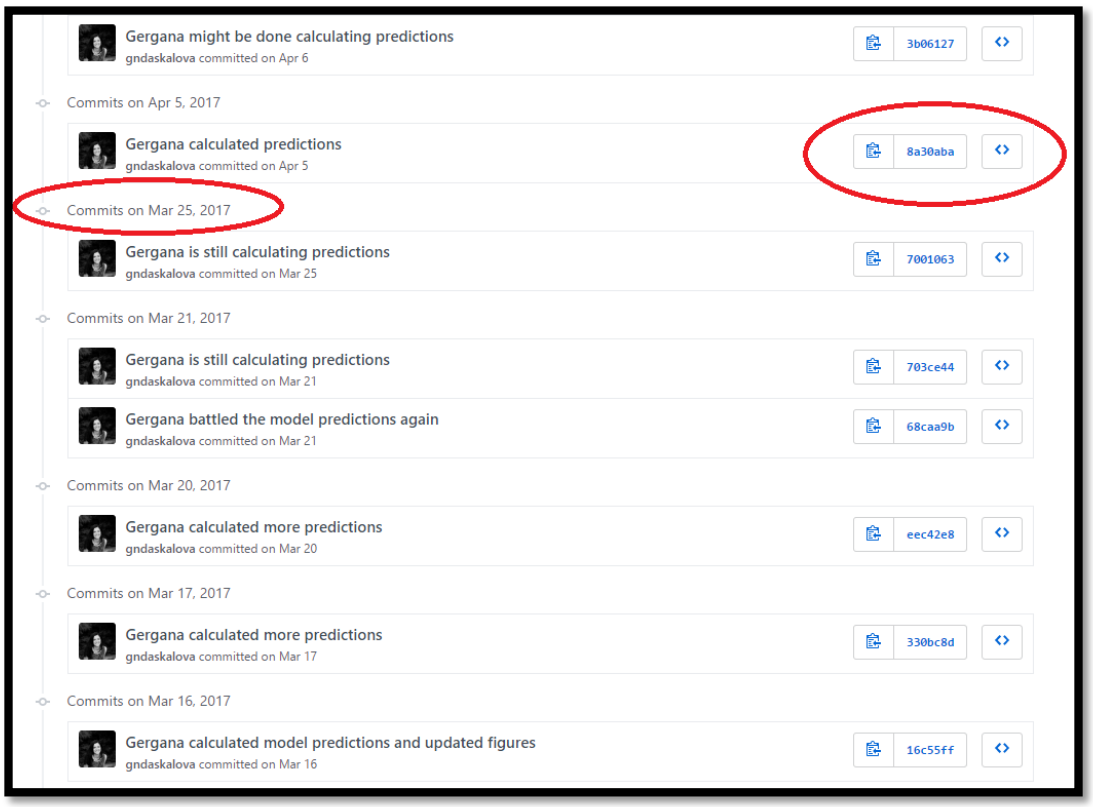

Ilustración 1 - Historial de Cambios en Github

Fuente: (Gergana, s. f.)

Github también facilita la colaboración entre grupos de cualquier tamaño por medio del uso de branches, esto hace que los miembros del grupo puedan trabajar en versiones secundarias del proyecto, manteniendo los cambios que ellos realicen en sus versiones separadas de la versión principal del proyecto. Para poder unir su versión a la versión

principal, el usuario tendría que combinar los cambios realizados en su versión con la versión principal por medio de una unión entre ambas.

En la Ilustración 2 se puede ver una representación gráfica del sistema de branches de Github.

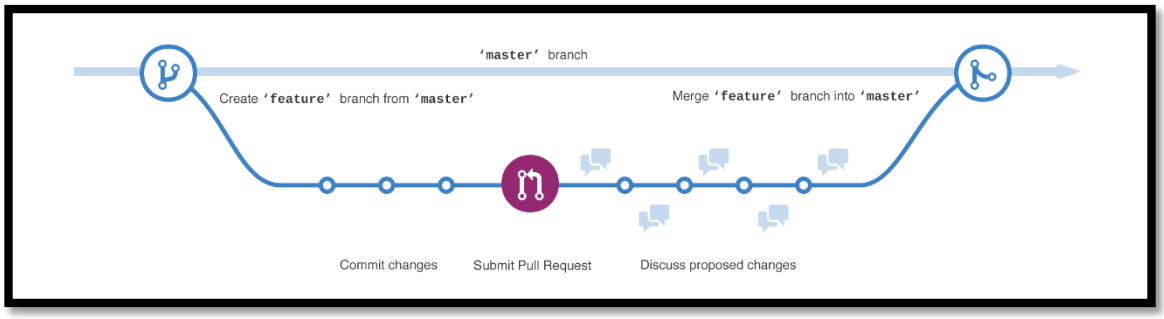

Ilustración 2 - Branches en Github

Fuente: (*What is Github?*, 2016)

Para hacer que sea portable, SUMADI ha sido diseñada como una herramienta modular que puede integrarse a las diferentes plataformas virtuales utilizadas por las distintas universidades dentro de la red de Laureate. Además de esto, ha sido diseñada como una aplicación web, cuenta con su front end y back end.

El front end de una página web es la parte con la que el usuario interactúa, es aquí donde se encuentran todos los componentes que son visibles, el texto, los colores, estilos, imágenes, gráficos, tablas, botones, y menús de navegación. Los lenguajes utilizados para el desarrollo en front end son HTML, CSS y Javascript. Los dos objetivos más importantes para los desarrolladores del front end son lograr que sus páginas web sean responsivas (es decir, que los componentes de la aplicación se acomoden al tamaño y la forma de la pantalla del dispositivo en el que se está viendo) y eficientes, ya que una aplicación con tiempos de espera largos es poco atractiva para los usuarios. Algunos de los frameworks utilizados en esta sección son AngularJS, React.js, jQuery y SASS («Frontend vs Backend», 2019).

El back end de una página funciona como la conexión con el servidor de esta. Se encarga de guarda y arreglar los datos, asegurando que el front end tenga acceso a los recursos que necesite. No es visible para los usuarios ya que no se supone que ellos tengan contacto con esta. En esta sección de la aplicación se incluyen acciones como el uso de

APIs, librerías, o incluso programas científicos. Los lenguajes utilizados para el back end son PHP, C++, Java, Python, Javascript y Node.js. Algunos de los frameworks utilizados en esta sección son Express, Django, Rails, Laravel y Spring («Frontend vs Backend», 2019).

Como se mencionó anteriormente, las APIs son una parte importante de las aplicaciones web modernas. Estas son útiles porque la división de las aplicaciones en front end y back end también se pueden ver como una división entre cliente y servidor. El cliente es el dispositivo del usuario, aquí es donde funciona el front end, el back end está ligado al servidor, la información y los recursos de la página se encuentran ahí. Gazarov (2019) dice que un API es la parte del servidor que recibe peticiones y devuelve respuestas.

Una mejora que se ha hecho en este diseño es la adición de REST, un estilo de arquitectura que establece estándares para la comunicación entre los sistemas en la web. La novedad de este estilo es que el cliente y el servidor ya no necesitan saber el estado del otro al momento de enviar y recibir mensajes. Considerando que cada mensaje ahora debe ser realizado en un formato específico, estos pueden ser entendidos de manera individual (*What Is REST?*, s. f.).

Para el front end de la aplicación se trabajó con React, una librería desarrollada por Facebook. Se creó para cumplir con las necesidades que surgieron durante el desarrollo de la red social, entre estas la más importante sería elevar el rendimiento de las aplicaciones (Álvarez, 2019).

Otros factores que ponen a React sobre los demás frameworks como jQuery es el hecho que puede ser usado para desarrollar aplicaciones web de manera más ordenada y con menos código. También se puede mencionar la habilidad de encapsular las funciones disponibles en la interfaz de usuario dentro de componentes, estos contienen el funcionamiento y la presentación del mismo (Álvarez, 2019).

El factor que ha logrado mejorar la eficiencia de las aplicaciones desarrolladas con este framework es el hecho de haber asociado las vistas con los datos, logrando que se actualicen las vistas al momento de cambiar datos de manera automática. Esto lo logra por medio del DOM Virtual, Robie (s. f.) define el Document Object Model (DOM), Modelo de Objetos del Documento en español, como la estructura lógica de los documentos y el modo en el que se accede y manipula un documento.

Al utilizar un DOM virtual (VDOM), React logra mantener una representación del DOM en memoria (*DOM virtual y detalles de implementación* – React, s. f.). Esto hace que sea posible actualizar solo las partes del DOM que deben ser cambiadas ya que se guardan los cambios deseados en el DOM virtual y se comparan con el DOM del navegador, ahorrándose la necesidad de actualizar la vista entera (Álvarez, 2019).

Además de esto, al ser un framework bastante popular, se encuentra rodeado por un ecosistema de módulos, herramientas y componentes, facilitando el desarrollo de funcionalidades complejas con relativamente poco esfuerzo. Por último cabe mencionar que React es isomórfico, este es un concepto nuevo que significa que una aplicación es capaz de utilizar el mismo código para generar HTML en el servidor y en el cliente (Álvarez, 2019).

En la Ilustración 3 se puede ver un ejemplo de cómo se ve una aplicación desarrollada con React.

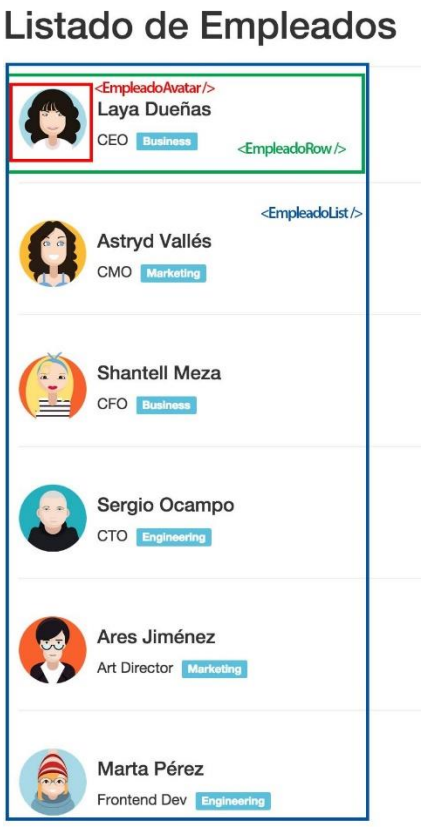

Ilustración 3 - Ejemplo de una Aplicación en React

Fuente: (Azaustre, 2015)

Para el back end se trabajó con Express. Express es el framework web más popular de Node, para tener un mejor entendimiento de este se debe conocer un poco más de Node.

Node funciona como un entorno que trabaja en tiempo de ejecución, es utilizado por los desarrolladores para construir aplicaciones del lado del servidor y aplicaciones en JavaScript. Una de las características más importantes de Node es que está diseñado para ser utilizado fuera del contexto de un explorador web, puede trabajar directamente desde una computadora o sistema operativo de servidor (*Introducción a Express/Node*, s. f.).

Además de esto, Node cuenta con un gran número de ventajas, algunas de ellas son mencionadas en el artículo *Introducción a Express/Node*, (s. f.):

- Fue diseñado para crear aplicaciones eficientes y escalables.
- Como su código está escrito en JavaScript, un lenguaje relativamente nuevo, muchos otros lenguajes nuevos populares se compilan o convierten a JavaScript, esto hace que se puedan utilizar lenguajes como CoffeeScript, ClosureScript, Scala, LiveScript, etc.
- NPM, el gestor de paquetes de Node provee al desarrollador con cientos o miles de paquetes reutilizables. También puede lograr automatizar en gran parte la cadena de herramientas de compilación.
- Es portable, tiene versiones que funcionan con varios sistemas operativos como Windows, OS X, Linux, Solaris, FreeBSD, OpenBSD, WebOS y NonStop OS.
- Tiene una comunidad bastante activa llena de gente que está dispuesta a ayudar.

Como se había mencionado anteriormente, Express es el framework web más popular de Node, en el artículo *Introducción a Express/Node* (s. f.) se habla sobre los mecanismos que se pueden encontrar dentro de este framework:

- Peticiones con diferentes verbos HTTP en diferentes rutas.
- Integración de motores para generar vistas que pueden crear respuestas a través de la introducción de datos en plantillas.

- Modificar los ajustes de las aplicaciones web, entre ellos el puerto que se usa para realizar la conexión y la localización de las plantillas que se usan para generar respuestas.
- Agregar procesamiento de peticiones de middleware adicionales en cualquier punto dentro del manejo de una petición.

Además de todo esto, Express cuenta con una gran cantidad de paquetes de middleware que pueden ser usados para enfrentar casi cualquier problema de desarrollo web. Se pueden encontrar librerías para trabajar con cookies, sesiones, inicios de sesión de usuario, parámetros URL, datos de POST, cabeceras de seguridad y muchos más (*Introducción a Express/Node*, s. f.).

En la Ilustración 4 se puede ver un ejemplo de una aplicación desarrollada usando Express.

```
/*
    Comenzar proyecto con Express y Node.Js

    parzibyte.me/blog
*/
const express = require("express"),
   path = require("path"),
   app = express(),
   puerto = 3000;

app.get('/', (peticion, respuesta) => {
   // Podemos acceder a la petición HTTP
   let agenteDeUsuario = peticion.header("user-agent");
   respuesta.send("La ruta / solicitada con: " + agenteDeUsuario);
});
```

Ilustración 4 - Ejemplo de una aplicación usando Express

Fuente: (Cabrera, s. f.)

Una de las estrategias utilizadas para mitigar los costos de mantener la infraestructura necesaria para poder utilizar un LMS es a través del uso de servicios en la nube. La nube más utilizada en el mundo es la de Amazon Web Services (AWS).

Hoy en día AWS ofrece sus servicios como una plataforma de infraestructura escalable, confiable y de bajo costo a clientes en 190 países de todo el mundo. Las soluciones más comunes ofrecidas en esta plataforma son el alojamiento de aplicaciones, sitios web, almacenamiento de datos y respaldos de los mismos así como bases de datos escalables, entre otras (*Acerca de AWS*, s. f.).

AWS ofrece una gran variedad de servicios, el equipo en el departamento de LMS utilizó los siguientes:

Amazon Elastic Compute Cloud, este le proporciona capacidad de computación escalable en la nube de AWS, eliminando la necesidad de invertir en hardware por medio del uso de servidores virtuales. Además de estos beneficios, se presentan los demás beneficios que provee este servicio en (¿Qué es Amazon EC2? - Amazon Elastic Compute Cloud (s. f.)):

- Existen varias ubicaciones físicas para los recursos divididas en regiones y zonas de disponibilidad.
- Es posible utilizar un firewall con el que se pueden especificar los protocolos, los puertos y los rangos de direcciones IP que pueden alcanzar las instancias por medio de grupos de seguridad.
- Se pueden crear direcciones IPv4 estáticas dentro de la nube dinámica, también conocidas como direcciones IP elásticas.

Otro de los servicios es Elastic Beanstalk, este es utilizado para implementar y escalar aplicaciones web en la nube. Solo se debe cargar el código y Elastic Beanstalk administra de manera automática la implementación. Esto resulta en varios beneficios para los clientes, entre ellos se encuentran el equilibrio de carga y el escalado automático (*AWS | Elastic beanstalk para aplicaciones web desarrolladas con Java*, s. f.).

El servicio de Codebuild se encarga de ejecutar pruebas y producir paquetes de software que estén listos para ser implementados. Es posible utilizar entornos de compilación pero también existe la opción de crear personalizar estos para que puedan utilizar herramientas designadas por el cliente *(AWS CodeBuild – Servicio de compilación completamente administrado*, s. f.).

CodePipeline es un servicio que permite automatizar canalizaciones de lanzamiento para lograr actualizaciones optimizadas de infraestructura y aplicaciones. Esta herramienta también se puede integrar con servicios externos como Github u otros complementos personalizados (*AWS CodePipeline | Integración y entrega continuas*, s. f.).

CloudFormation provee un lenguaje común para crear modelos que son utilizados para preparar todos los recursos necesarios de AWS para las aplicaciones por medio de archivos generados con lenguajes de programación o texto simple (AWS CloudFormation - Infraestructura como código y aprovisionamiento de recursos de AWS, s. f.).

Sonarqube es una herramienta utilizada en conjunto con el resto de los servicios de AWS para encontrar errores y malas prácticas en el código, esto ayuda a los equipos de desarrollo a implementar mejores prácticas en sus proyectos (About | SonarQube, s. f.).

Por otro lado también es importante mencionar los siguientes servicios que son usados para el almacenamiento de datos.

Amazon DynamoDB es una base de datos NoSQL de clave-valor y documentos diseñada para mantener un rendimiento en milisegundos por transacción a cualquier escala. Cuenta con copias de seguridad, restauración y seguridad integradas, siendo esta una excelente opción para las empresas que están experimentando un crecimiento rápido *(AWS | Servicio de base de datos gestionada NoSQL (DynamoDB)*, s. f.). También se puede mencionar que es altamente flexible gracias a que admite estructuras de datos de valor de clave y documento (iNBest, s. f.-a).

Amazon Simple Storage Service es un servicio de almacenamiento de objetos que ofrece los siguientes beneficios:

- Sencillez: Es fácil de utilizar desde un dispositivo móvil con una consola de administración web.
- Durabilidad: Con su infraestructura duradera, está diseñado para almacenar datos importantes.
- Seguridad: Es posible transferir datos a través de SSL y cifrarlos de forma automática una vez que se han cargado.

- Disponibilidad: Garantiza la disponibilidad de los datos durante un periodo de un año.
- Bajo costo.

(iNBest, s. f.-b)

CosmosDB es un servicio de Microsoft de base de datos NoSQL que garantiza eficiencia y disponibilidad de datos en cualquier parte del mundo. Cuenta con API de código abierto para MongoDB y Cassandra (*Azure Cosmos DB*, s. f.).

MongoDB es una base de datos orientada a documentos, estos son almacenados en BSON, una representación binaria de JSON. A diferencia de las bases de datos relacionales, los documentos en MongoDB no necesitan seguir un esquema, esto permite que existan varios documentos dentro de una misma colección (similar a la tabla en las bases de datos relacionales) que están compuestos de campos diferentes, ya sea por el número de campos o los tipos de datos que se encuentran en ellos (rubenfa, 2014).

En la Ilustración 5 se puede ver un ejemplo de una colección en MongoDB en la cual los documentos tienen campos diferentes.

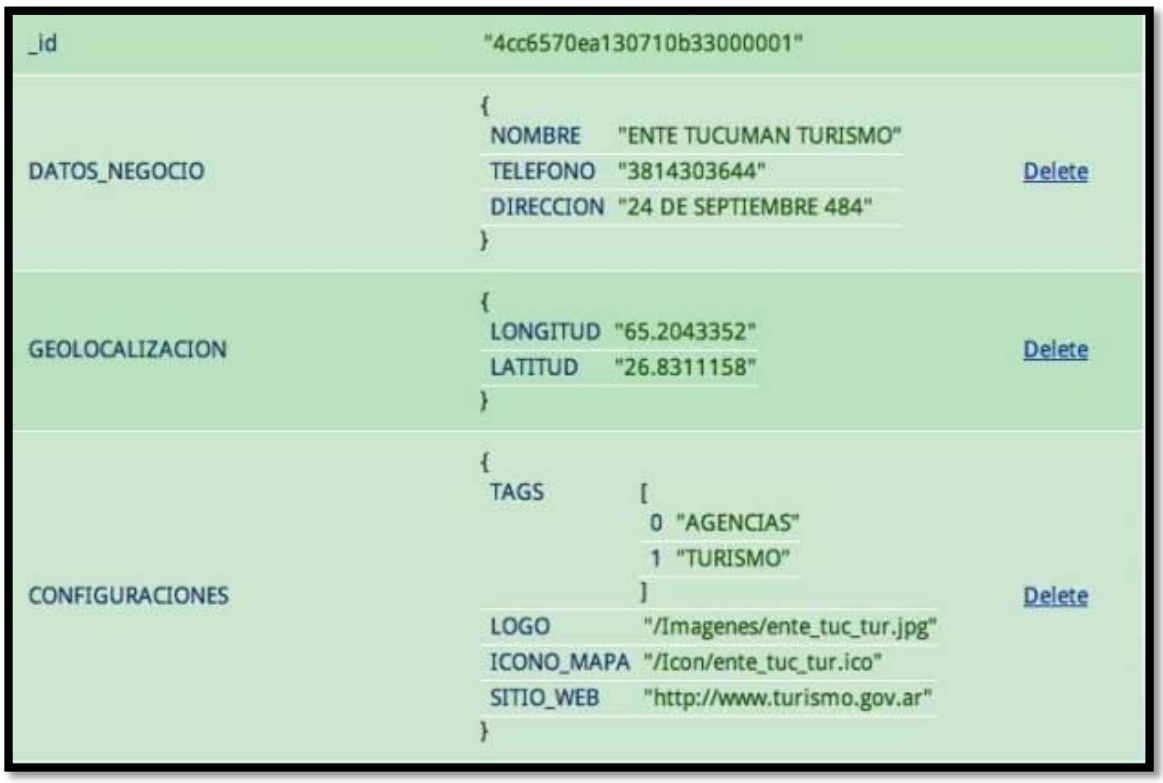

Ilustración 5 - Documentos en una Colección en MongoDB

Fuente: («Experiencia MongoDB – 2da Parte», 2010).

En teoría MongoDB se puede utilizar en muchos proyectos que se desarrollan en la actualidad, es especialmente útil en los proyectos que necesitarán escalabilidad. Se puede notar que aunque los documentos en esta base no necesitan seguir esquema, es común que estos terminen siguiendo uno para el funcionamiento de la aplicación, normalmente el esquema extraoficial es decidido por las consultas que se realicen con más frecuencia en la aplicación (rubenfa, 2014).

Es importante notar que las bases de datos presentadas anteriormente han sido NoSQL o no relacionales, esto es porque en el pasado se han encontrado algunos problemas con los modelos relacionales:

- Leer datos es costoso, considerando que en el modelo relacional los datos están almacenados en tablas relacionadas entre sí. Las consultas generalmente necesitan unir las tablas, esta operación es cara en términos de cómputo.
- El uso de transacciones interfiere con el rendimiento por el uso de bloqueos a nivel de datos, esto hace que los modelos relacionales se vean limitados en

situaciones en las que la eficiencia es más importante que la integridad de los datos.

- Los modelos relacionales no son escalables, tratar de llenar una tabla con más registros hace que el servidor de base de datos utilice cada vez más recursos de hardware. Es posible almacenar las tablas en varios servidores pero como estas están relacionadas se necesitarían unir desde distintos sistemas, lo cual sería una operación bastante cara.

  (Camacho, s. f.)

Estos problemas son solucionados en las bases no relacionales eliminando las relaciones entre las tablas para reducir el costo de leer datos e incrementar la escalabilidad de las bases de datos. También se eliminan las transacciones para mejorar el rendimiento. Además de estas soluciones, Camacho (s. f.) señala que existen los siguientes tipos de bases de datos no relacionales:

- Almacenes Clave-Valor: Estas bases de datos guardan una clave y el valor asociado, este valor es guardado como un arreglo de bytes. Esto hace que no tenga esquema y que sea bastante eficiente al momento de hacer lecturas y escrituras. Su limitante es que no se pueden realizar búsquedas avanzadas ya que el único tipo de consulta que se puede hacer es por una clave.
- Bases de Datos Columnares: Estas bases de datos guardan los datos en columnas en lugar de renglones, esto hace que las lecturas sean más rápidas ya que es más eficiente realizar una consulta con un número reducido de columnas. Sin embargo no es muy eficiente al momento de realizar escrituras.
- Bases de Datos Orientadas a Documentos: Estas bases de datos son similares a los almacenes clave-valor, sin embargo tienen una diferencia importante. El campo de valor no se guarda como un campo binario sino que se utiliza un formato que puede ser entendido. Este concepto le permite al usuario realizar operaciones y consultas avanzadas sobre los datos y hasta crear relaciones entre los mismos, aunque no permite que se unan las tablas para evitar las limitantes de las bases relacionales.

- Bases de Datos Orientadas a Grafos: Este modelo almacena los datos en forma de grafos. Esto hace que las relaciones entre los nodos tengan gran importancia, de hecho estas pueden hasta tener atributos propios. Estas bases son útiles en situaciones en las que los datos pueden ser representados como una red.
- Bases de Datos Orientadas a Objetos: Este modelo está orientado a objetos y no al modelo relacional, por esto es capaz de representar relaciones jerárquicas. También reemplaza las claves primarias con OID (identificadores únicos de objetos).

Otro aspecto importante del trabajo del equipo de LMS es realizar pruebas de control de calidad (QA) antes del lanzamiento de cualquier aplicación. Ares (2017) presenta los tipos de pruebas que se pueden realizar:

- Pruebas de software funcionales, también llamadas pruebas de Caja Negra, en estas se evalúa el comportamiento del sistema, subsistema o componente de software. Normalmente son pruebas ya documentadas con requisitos o casos de uso ya establecidos. En esta categoría se pueden encontrar las pruebas de Regresión, las pruebas de Aceptación de Usuario, pruebas de Interfaz de Compatibilidad, pruebas de Instalación y pruebas de Interfaz de Usuario.
- Pruebas de software no funcionales, estas buscan evaluar el comportamiento externo del sistema. En estas encontramos las pruebas de Rendimiento, de Carga, de Estrés, de Usabilidad, Mantenibilidad, Fiabilidad y de Portabilidad.
- Pruebas de software estructurales, estas buscan evaluar el comportamiento interno de la aplicación probando los componentes y la integración de estos. También son conocidas como pruebas de Caja Blanca.
- Pruebas de software de regresión o re-pruebas, se realizan cuando un error detectado ha sido corregido para asegurar que este ha sido eliminado. Es importante volver a revisar todos los componentes, no solo el modificado, para asegurar que no hayan nuevos errores que surjan como consecuencia de los cambios realizados.

Considerando el incremento de usuarios en SUMADI, se debieron realizar pruebas de carga utilizando la herramienta JMeter. Apache JMeter es una aplicación gratuita que fue

diseñada en Java para poder realizar pruebas de estrés y evaluar el rendimiento de las aplicaciones web, esto se logra simulando enormes cantidades de solicitudes en un servidor o una red (Apache JMeter - Apache JMeterTM, s. f.).

En la Ilustración 6 se puede ver la interfaz gráfica de un programa de prueba en JMeter.

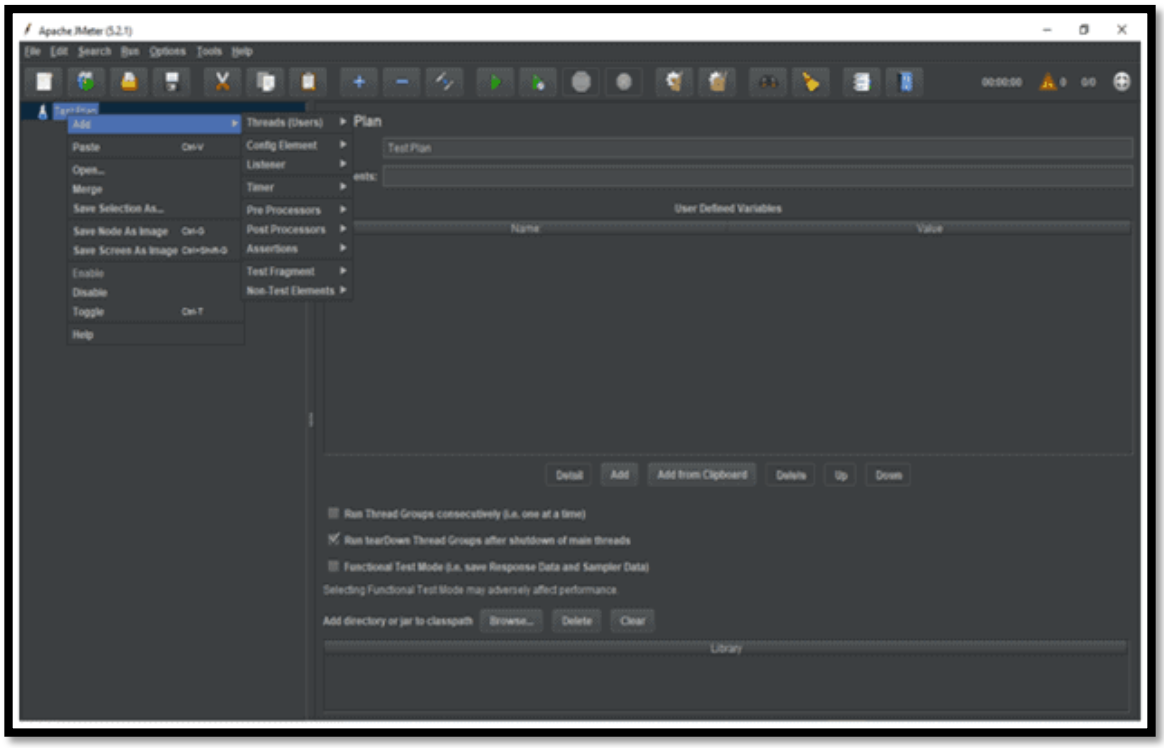

Ilustración 6 - Interfaz Gráfica de JMeter

Fuente: («Performance Testing Software | Introduction of Apache Jmeter», 2020)

IV. DESARROLLO

4.1 DESCRIPCIÓN DEL TRABAJO REALIZADO

4.1.1 ENTRENAMIENTO EN BLACKBOARD

El primer paso para poder aprender a utilizar SUMADI fue entender el funcionamiento de la plataforma de Blackboard. Se utilizaron varios tutoriales para conocer la experiencia de maestro y de administrador.

En la Ilustración 7 se muestra el entrenamiento en Blackboard encontrado en la wiki.

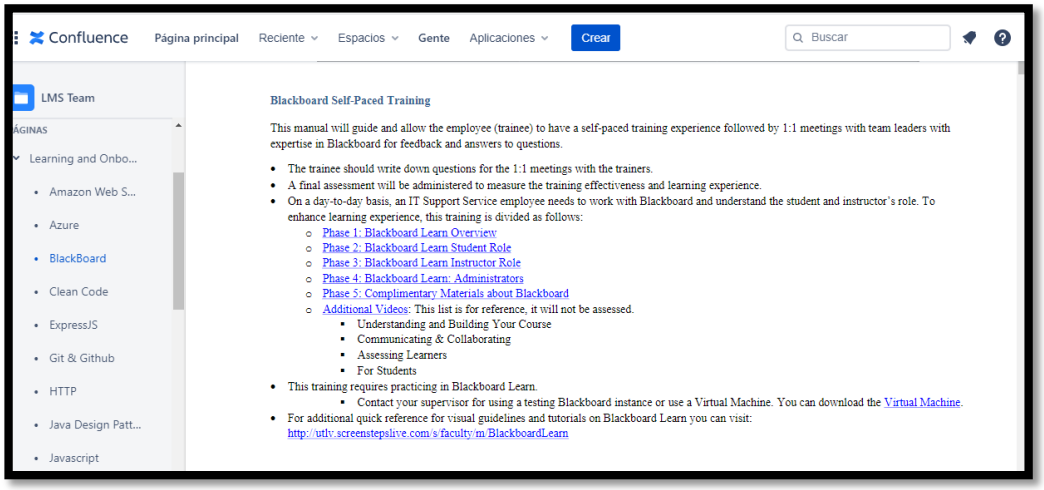

Ilustración 7 - Entrenamiento de Blackboard

Fuente: (*BlackBoard - LMS Team - Confluence*, s. f.)

4.1.2 PRUEBAS DE REGRESIÓN DE SUMADI

Se planeaba realizar pruebas de regresión en la aplicación SUMADI en su versión web y en la Electron app, el practicante realizó pruebas primero en la versión web en la plataforma de Blackboard y luego en la Electron app utilizando la interfaz de varias instituciones. Luego se preparó un reporte para presentar los errores encontrados al equipo de desarrollo.

Entre los errores encontrados se pueden mencionar, problemas de la interfaz gráfica, como pantallas no responsivas y botones con colores poco amigables. También se pueden mencionar problemas que intervienen con las funcionalidades de la aplicación como datos erróneos en los reportes de los exámenes, que la cámara se mantenga activa

y que el contador de tiempo del examen siga corriendo aún después de que el alumno abandone la evaluación.

Por otro lado también se pueden mencionar algunas vulnerabilidades del sistema, entre ellas está el hecho que la aplicación aún no detecta el uso de la función de duplicación de pantallas, esto permite que se controle una computadora de manera remota desde otra computadora o celular. También es posible que al desactivar las opciones de monitoreo de la cámara web y de verificación de identificación, los estudiantes puedan entrar sin ninguna verificación a un examen.

Para poder explorar a fondo el proceso de pruebas de SUMADI se debe hablar de las diferentes versiones que existen. Entre estas se pueden contar:

- Electron App: Esta es una de las versiones de escritorio, se enfoca en la experiencia de estudiante e instructor de la institución Cambridge.
- Q: Esta es la segunda versión de escritorio, se enfoca en la experiencia de estudiante de SUMADI para las instituciones IPP y Teclab.
- OpenLMS: Esta es la tercera versión de escritorio, ofrece una experiencia más completa a los estudiantes que las dos versiones anteriores con una interfaz que les permite entrar a los cursos en los que están matriculados, pudiendo participar en actividades dentro de los mismos y obtener acceso a recursos compartidos en la plataforma, además de poder tomar sus exámenes.
- Mern: Esta es la versión en línea de la aplicación, en esta se encuentra la experiencia del instructor en el Q y las experiencias de estudiante, instructor y administrador en Blackboard.

En la Ilustración 8 se muestra la interfaz gráfica de la Electron app.

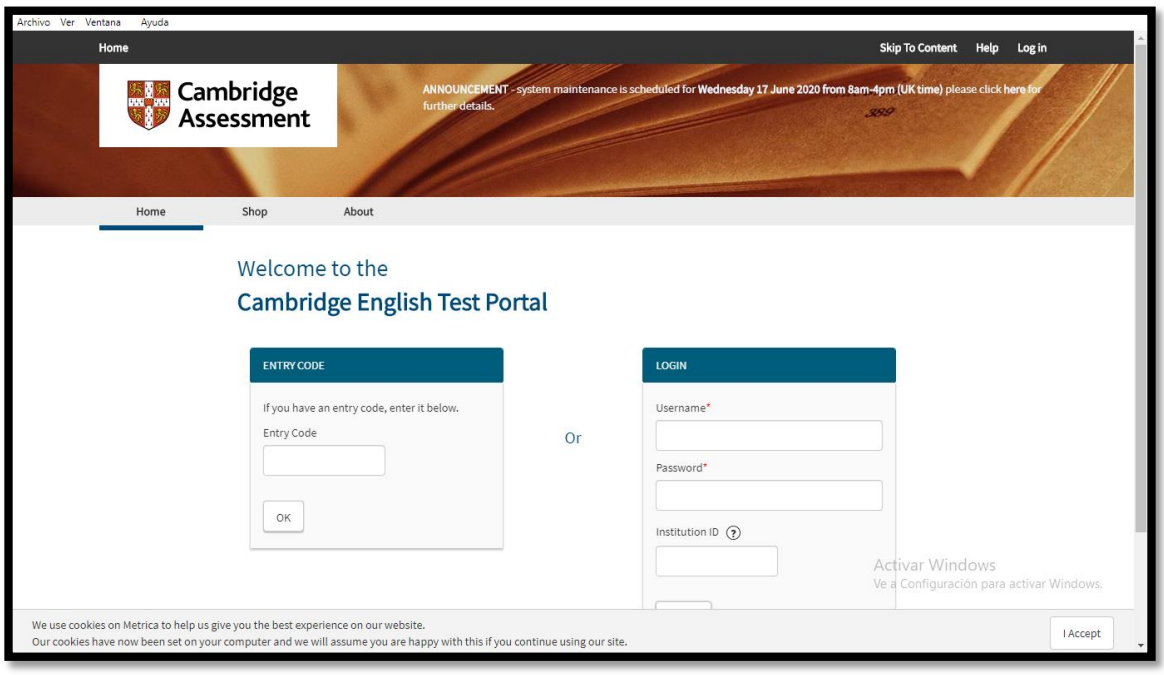

Ilustración 8 - Interfaz Gráfica Electron App

Fuente: (Laureate International Universities, s. f.-c)

En la Ilustración 9 se muestra la interfaz gráfica de la versión Q de SUMADI.

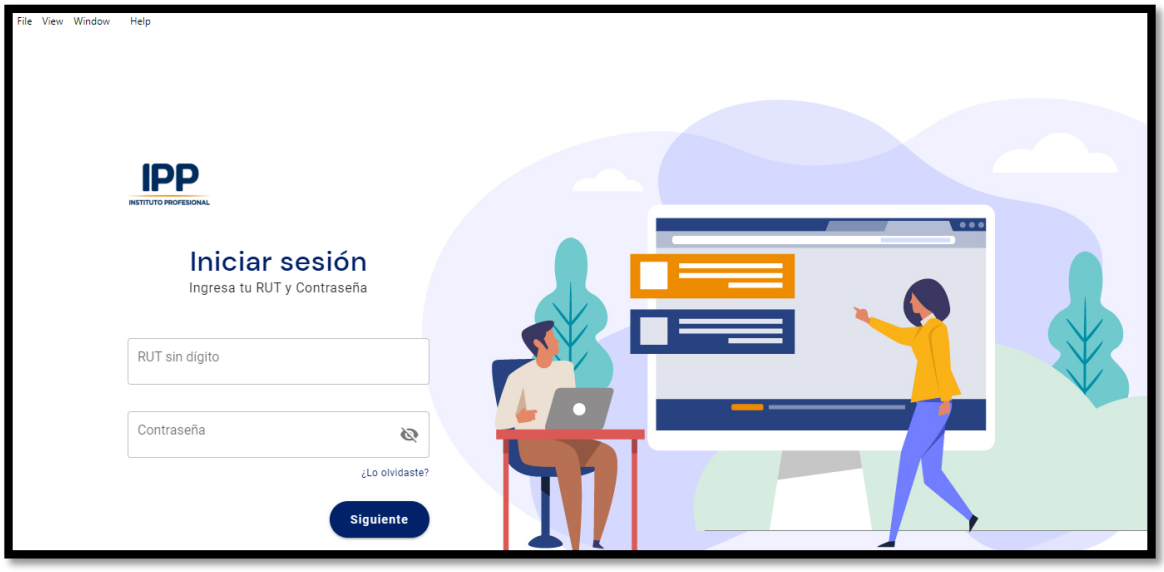

Ilustración 9 - Interfaz Gráfica de SUMADI Q

Fuente: (Laureate International Universities, s. f.-b)

4.1.3 PRUEBAS DE REGRESIÓN DE BOOKING

Booking es una aplicación que consiste de un sistema de reservas en el cual se pueden crear salas y actividades, esto les provee a los profesores una herramienta para poder notificar a los alumnos de actividades, como exámenes o laboratorios, con la fecha y el lugar en donde estos están planeados. Además, evita que se generen conflictos entre diferentes docentes queriendo utilizar el mismo espacio en el mismo momento.

En las pruebas realizadas por el practicante se encontraron algunos problemas, entre ellos se puede mencionar la posibilidad de modificar y eliminar salas que tienen actividades ligadas a ella. Por otro lado es importante mencionar que en su estado actual la aplicación ofrece tres diferentes interfaces, una para editores, otra para profesores y una tercera para estudiantes. Los editores y profesores pueden crear, modificar y eliminar salas y reservaciones, los estudiantes pueden ver las reservaciones en las que han sido incluidos y tienen la opción de aceptar o cancelar estas invitaciones. Actualmente la interfaz de profesor no es funcional.

En la Ilustración 10 se muestra la vista de editor en Booking.

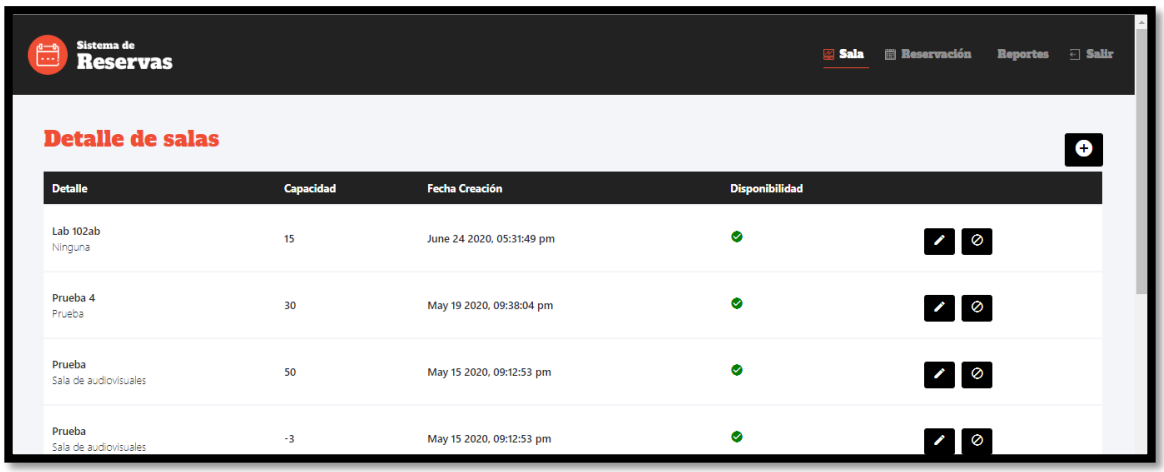

Ilustración 10 - Interfaz de Editor en Booking

Fuente: (Laureate International Universities, s. f.-a)

4.1.4 RETROALIMENTACIÓN DE ERRORES

Luego de preparar un informe detallado de los errores encontrados, el equipo de QA tuvo reuniones con el grupo de desarrollo para discutir los problemas, asignando un valor de importancia a cada uno de los errores y dividiéndolos entre errores y oportunidades de mejora. Los errores eran considerados importantes y debían ser corregidos lo más pronto posible, las oportunidades de mejora se podían posponer.

Para poder tener acceso a información en tiempo real sobre los defectos reportados y el estado en el que estos se encuentran, se comenzó a utilizar la herramienta de reporte de errores en Azure DevOps. Por medio de esta es posible reportar los errores y asignarlos a los equipos de desarrollo, luego los desarrolladores pueden reportar los errores como resueltos y, una vez que estos sean aceptados, es posible visualizar las estadísticas de los errores (cuántos fueron reportados, qué porcentaje fue resuelto, etc.).

4.1.5 PLANEACIÓN DEL TERCER PERÍODO DE TRABAJO DE 2020

En la Ilustración 11 se muestran los pasos que se siguieron durante la organización del trabajo que se realizará a lo largo del tercer período del 2020.

Ilustración 11 - Agenda Estándar de una reunión de PI Planning

Fuente: («PI Planning», s. f.)

En esta reunión se dedicaron 2 días para discutir las tareas y los proyectos que los equipos planean cumplir en este siguiente periodo de trabajo. Durante este proceso, los equipos de IT se reunieron para discutir las metas, se trabajó por medio de la herramienta en línea Miro para crear un espacio virtual en el que cada equipo pudo reportar sus objetivos como notas en una pizarra.

El equipo de arquitectura definió las siguientes metas para este periodo:

- Soporte a otros equipos:

  El equipo de arquitectura debe apoyar a los demás equipos por medio de control de calidad de las aplicaciones antes de que estás puedan pasar al ambiente de producción. También se trabaja en los despliegues de las aplicaciones una vez que estás hayan sido probadas. Adicionalmente, se trabaja en los manuales de usuario, asegurando que estos estén al día con las funciones que los nuevos despliegues contienen.

- Definir Planes de Recuperación de Desastres:

  Se deben definir planes de contingencia para poder hacer respaldos de las bases de datos. Para lograr crear planes efectivos se deben especificar las condiciones bajo las cuales se crearán respaldos, es decir, cuánto tiempo habrá entre cada respaldo, y también el tiempo que tardaría recuperar los datos en estas copias de seguridad.

- Control de Calidad de SUMADI y Booking:

  Se deben crear casos de prueba para las varias versiones de SUMADI, estos facilitarán las pruebas de regresión al momento de agregar nuevas características a la aplicación. Luego, estos mismos casos de prueba serán utilizados para reportar errores encontrados por medio de Azure DevOps.

- Despliegue de Aplicaciones:

  El equipo debe ayudar a los desarrolladores a desplegar las aplicaciones en los diferentes ambientes por los que estas deben pasar. Cada nuevo lanzamiento debe pasar por los ambientes de prueba para que una vez que estos sean aprobados, estos puedan pasar al ambiente de producción.

- Seguridad de la aplicación SUMADI:

  Se debe asegurar que se sigan los lineamientos definidos para garantizar la seguridad en las aplicaciones de la empresa. Esto se logra trabajando con listas semanales para monitorear y modificar las políticas de seguridad.

4.1.6 Actualización de Manuales de Usuario

Los manuales son utilizados para facilitar la capacitación de los usuarios finales de SUMADI (los estudiantes e instructores de las instituciones), estos deben ser actualizados para incluir las nuevas características que son parte de los lanzamientos. Se trabajó en los manuales para las diferentes implementaciones de SUMADI, entre ellas se encuentran las versiones en línea (Act360, Edusoft y Blackboard) y las versiones de escritorio (Electron y Q).

4.1.7 Pruebas de Carga con JMeter

Para poder asegurar la escalabilidad de SUMADI, se planearon pruebas de carga y de estrés, estás fueron realizadas por el practicante con la herramienta JMeter. Estas pruebas fueron diseñadas para simular el proceso que se ejecuta en el backend cada vez que un usuario realiza un examen utilizando el monitoreo de SUMADI, se diseñaron dos programas de prueba:

- Registro y Verificación: En este programa se simula el proceso que los estudiantes siguen al crear su perfil la primera vez que entran a SUMADI. Primero, el estudiante debe tomarse tres fotos para crear su perfil facial, una vez que este se ha creado de manera exitosa, se pasa a la verificación facial. Aquí el estudiante debe tomarse una foto y enviarla, luego debe tomarle una foto a algún documento oficial que muestre su nombre y su rostro como una identidad o pasaporte.
- Examen Regular: Este programa simula el proceso de un estudiante realizando una prueba de media hora, primero se pasa por el paso de verificación facial, luego se envían capturas hechas por la cámara web del estudiante cada treinta segundos.

Cada uno de los pasos mencionados en estos programas envía una solicitud HTTP a los servidores, tres pruebas fueron realizadas, la primera con mil usuarios, la segunda con cuatro mil y la tercera con ocho mil. Para poder crear todos estos usuarios, se diseñó otro programa en JMeter para enviar solicitudes con los datos de los mismos.

Al realizar las pruebas se monitorearon tres aspectos:

- El estado de los servidores, esto incluye datos como el estado de los mismos, la cantidad de solicitudes que recibían por segundo, la cantidad de bytes enviados y recibidos, entre otros.
- El estado de la base de datos, esto incluye las unidades de solicitud (RU), estas miden el costo de las operaciones realizadas por la base de datos.
- El estado de los servicios de reconocimiento de Amazon, estos son los que realizan las operaciones de verificación facial. Aquí se observaba la cantidad de llamados que eran recibidos y, cuando estos servicios alcanzaban su límite, se observaba la cantidad de solicitudes que no pudieron ser procesadas.

Al terminar las pruebas se concluyó que se debía aumentar la capacidad de los servicios de reconocimiento de Amazon ya que estos fueron los primeros en fallar, mientras los servidores y la base de datos fueron capaces de manejar las solicitudes de más de seis mil usuarios concurrentes, los servicios de Amazon solo pudieron procesar aproximadamente un 40% de estas solicitudes.

4.1.8 Pruebas de Estrés

Al terminar las pruebas de carga, SUMADI fue utilizado para las pruebas del Instituto Colombiano de Educación Superior (ICFES), estas son realizadas a nivel nacional por alrededor de 60,000 estudiantes de manera concurrente, en estas se descubrió que el sistema no estaba preparado para soportar esta cantidad de usuarios. Los exámenes fueron pospuestos y se decidió agregar varias mejoras al sistema para poder alcanzar este nivel de carga, entre ellas se encuentran las siguientes:

- Cambiar las bases de datos de CosmosDB a DocumentDB.
- Crear colas en las que las imágenes enviadas por los alumnos puedan quedar en espera de ser procesadas en caso de que los servicios de Amazon estén saturados.
- Agregar mejoras a la interfaz gráfica para mejorar la experiencia de los estudiantes.

Para poder validar la efectividad de estas soluciones, se realizaron nuevas pruebas de estrés en JMeter con el propósito de simular la carga esperada el día de estos exámenes. Se actualizó el script de la experiencia del estudiante para incluir las solicitudes HTTP que recuperan los datos utilizados para cargar las páginas en el frontend. También se creó un nuevo script para simular la experiencia del monitor, incluyendo todos los pasos que se espera que estos usuarios sigan. Entre estos se encuentran:

- La autenticación del usuario.
- Configuración del monitoreo de un examen.
- Vista del reporte general de un examen.
- Vista del reporte individual del intento de un estudiante en un examen determinado.
- Vista del reporte global del curso en el que se puede ver un resumen de los intentos de los estudiantes en todos los exámenes del curso.

4.1.9 Integración de SonarQube en Codebuild

La inclusión de Sonarqube en los procesos de Codebuild fue realizada con el propósito de proporcionar una importante herramienta para encontrar errores, vulnerabilidades y malas prácticas en el código presentado por el equipo de desarrollo. Esto eleva la calidad del producto final presentado en los nuevos lanzamientos de SUMADI.

Para poder implementar esta herramienta en Codebuild, simplemente se debió configurar el archivo buildspec.yaml, agregando un conjunto de comandos que descargan y configuran los archivos necesarios para este proceso. Luego, los usuarios solamente deben entrar a la página web de SonarQube, ahí se encuentran todos los proyectos que han sido subidos a Github por el equipo de desarrollo, cada uno con las mejoras propuestas por esta herramienta.

4.1.10 Ejecución de Modelos en CloudFormation

Para poder crear la infraestructura necesaria para el funcionamiento de SUMADI, se utilizó el servicio de CloudFormation de AWS. Este provee una interfaz en la que se pueden usar modelos predeterminados para crear los servicios necesarios. El equipo de arquitectura ha determinado el orden en el que estos servicios deben ser creados para que puedan funcionar correctamente:

- Primero se deben crear los servicios de red que serán utilizados en este ambiente.
- Luego se deben generar los roles de usuarios que pueden existir y los permisos que cada uno de estos puede tener.
- Después se debe crear el servicio de almacenamiento que será usado para los datos que manejará SUMADI.
- Por último se corre un último modelo que busca integrar los servicios creados anteriormente, haciendo referencia a ellos para que puedan estar entrelazados.

4.1.11 Security and Compliance

Para asegurar que las aplicaciones manejadas por el equipo de arquitectura cumplen con los estándares de seguridad establecidos por la empresa, se deben realizar revisiones semanales para encontrar vulnerabilidades y problemas existentes. Para poder encontrar estas debilidades se utilizan dos herramientas:

- Trend Micro: un servicio de seguridad cibernética que provee soluciones para vulnerabilidades encontradas en redes y servicios de la nube. Este es proporcionado por la empresa Trend Micro Inc. (About Us, s. f.).
- Tenable.io: un servicio gestionado en la nube con la capacidad de sugerir qué problemas de seguridad tienen mayor prioridad y deben ser resueltos primero (Tenable.io, 2017).

4.1.12 Validación de Datos para Entregar al ICFES

Luego de que se realizaran las pruebas del ICFES, se necesitaba enviar la información de la base de datos para que el gobierno de Colombia tuviera acceso a esta. Esta actividad se realizó con el apoyo del equipo de Operaciones, ellos prepararon un script desarrollado en Python para guardar los datos de SUMADI en archivos CSV. Para verificar que este proceso se realizó de manera exitosa, el practicante debió comparar la información que se planeaba enviar al cliente con la que se encontraba en la base de datos.

4.1.13 Migración de Repositorios en Github

Desde el último PI Planning, realizado el 18 de agosto, la compañía decidió migrar todos sus archivos de la cuenta actual en Github Enterprise a una organización nueva creada en Github. Para poder realizar esta migración, el practicante recopiló una lista de todos los proyectos en la cuenta vieja que pertenecían al equipo LMS de Honduras, luego se creó una lista de todos los usuarios que debían ser agregados a la nueva organización, por último se importaron todos los repositorios.

4.2 CRONOGRAMA DE ACTIVIDADES

En la Ilustración 12 se puede observar el cronograma de actividades realizadas durante la práctica profesional.

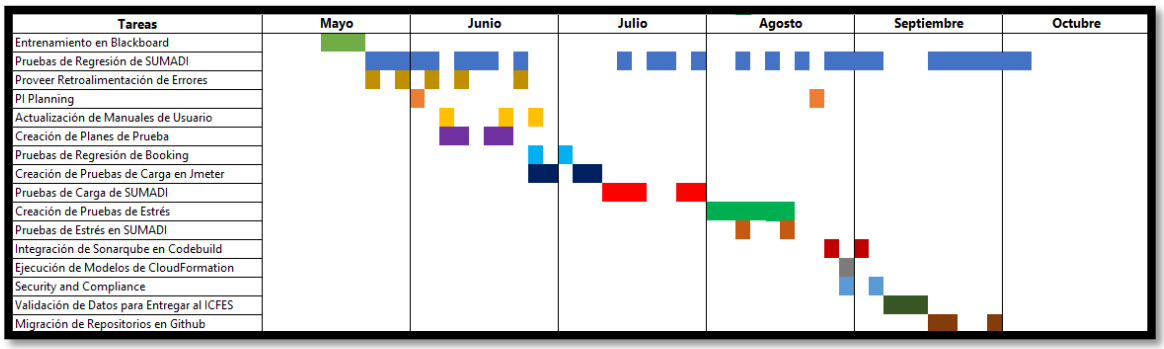

Ilustración 12 - Diagrama de Gantt

Fuente: (Elaboración Propia)

V. CONCLUSIONES

I. Se realizaron numerosas pruebas de calidad junto con revisiones semanales, estas fueron importantes para poder encontrar vulnerabilidades de seguridad que fueron mitigadas periódicamente.

II. Se realizaron pruebas de calidad cada vez que se planeó un lanzamiento de una nueva versión de SUMADI, también se reportaron los errores encontrados y se trabajó con el equipo de desarrollo para asegurar que la aplicación fuera desplegada sin errores que afectaran la experiencia de los usuarios de manera significativa.

III. Se documentaron los casos de prueba en los cuales se explica el flujo de las diferentes versiones de SUMADI para facilitar el proceso de control de calidad en el futuro, también se crearon y actualizaron los manuales de usuarios que serán importantes al momento de capacitar a los docentes y estudiantes de las instituciones que utilizarán esta aplicación.

IV. Se manejaron aplicaciones de AWS como Codebuild para manejar los nuevos lanzamientos de SUMADI y CloudFormation para crear la infraestructura necesaria por medio de archivos de texto.

VI. RECOMENDACIONES

1. Se deben realizar reuniones entre los diferentes equipos de trabajo para poder sincronizar las actividades que serán realizadas cuando los equipos deben realizar tareas de manera complementaria.
2. Se debe dedicar más tiempo a la solución de errores dentro del proceso de desarrollo, actualmente solo se planean dos o tres días para las pruebas de calidad, esto hace que muchos errores que no son considerados críticos sean minimizados.
3. Se deben establecer mejores prácticas en la documentación, actualmente las notificaciones de errores son genéricas, al utilizar mensajes más descriptivos, el equipo podrá dar reportes más detallados a los desarrolladores al realizar pruebas para que puedan encontrar soluciones de manera eficiente.

BIBLIOGRAFÍA

1. *About | SonarQube*. (s. f.). Recuperado 6 de septiembre de 2020, de https://www.sonarqube.org/about/
2. *About Us*. (s. f.). Trend Micro. Recuperado 6 de septiembre de 2020, de https://www.trendmicro.com/en_us/about.html
3. *Acerca de AWS*. (s. f.). Amazon Web Services, Inc. Recuperado 1 de junio de 2020, de https://aws.amazon.com/es/about-aws/
4. Álvarez, M. (2019, febrero 25). *Qué es React. Por qué usar React*. https://desarrolloweb.com/articulos/que-es-react-motivos-uso.html
5. *Apache JMeter—Apache JMeter™*. (s. f.). Recuperado 20 de julio de 2020, de https://jmeter.apache.org/
6. Ares, L. (2017, octubre 26). Testing: Su importancia y los distintos tipos de pruebas de software. *Visual Engineering*. http://visual-engin.com/2017/10/26/importancia-pruebas-de-software-testing/
7. *AWS CloudFormation—Infraestructura como código y aprovisionamiento de recursos de AWS*. (s. f.). Amazon Web Services, Inc. Recuperado 6 de septiembre de 2020, de https://aws.amazon.com/es/cloudformation/
8. *AWS | Elastic beanstalk para aplicaciones web desarrolladas con Java*. (s. f.). Amazon Web Services, Inc. Recuperado 1 de junio de 2020, de https://aws.amazon.com/es/elasticbeanstalk/
9. *AWS | Servicio de base de datos gestionada NoSQL (DynamoDB)*. (s. f.). Amazon Web Services, Inc. Recuperado 1 de junio de 2020, de https://aws.amazon.com/es/dynamodb/
10. *AWS CodeBuild – Servicio de compilación completamente administrado*. (s. f.). Amazon Web Services, Inc. Recuperado 1 de junio de 2020, de https://aws.amazon.com/es/codebuild/

11. *AWS CodePipeline | Integración y entrega continuas*. (s. f.). Amazon Web Services, Inc. Recuperado 1 de junio de 2020, de https://aws.amazon.com/es/codepipeline/

12. Azaustre, C. (2015, junio 23). *Ejemplo de aplicación con React.js en ECMAScript 6*. carlosazaustre.es. https://carlosazaustre.es

13. *Azure Cosmos DB: Servicio de base de datos multimodelo | Microsoft Azure*. (s. f.). Recuperado 1 de junio de 2020, de https://azure.microsoft.com/es-es/services/cosmos-db/

14. Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Schwaber, K., Mellor, S., Sutherland, J., & Thomas, D. (2001). *Manifesto for Agile Software Development*. https://agilemanifesto.org/

15. *BlackBoard—LMS Team—Confluence*. (s. f.). Recuperado 1 de junio de 2020, de https://lmsarchitecture.atlassian.net/wiki/spaces/LMS/pages/1376326/BlackBoard

16. Cabrera, L. (s. f.). *Node.Js + Express: Ejemplo de creación de proyecto—Parzibyte's blog*. Recuperado 1 de junio de 2020, de https://parzibyte.me/blog/2019/05/27/node-js-express-ejemplo-creacion-proyecto/

17. Camacho, E. (s. f.). *NoSQL la evolución de las bases de datos*. SG Buzz. Recuperado 1 de junio de 2020, de https://sg.com.mx/revista/28/nosql-evolucion-bases-datos

18. *DOM virtual y detalles de implementación – React*. (s. f.). Recuperado 1 de junio de 2020, de https://es.reactjs.org/docs/faq-internals.html

19. Experiencia MongoDB – 2da Parte. (2010, octubre 29). *Tag in News*. https://taginnews.wordpress.com/2010/10/29/experiencia-mongodb-2da-parte/

20. Frontend vs Backend. (2019, julio 11). *GeeksforGeeks*. https://www.geeksforgeeks.org/frontend-vs-backend/

21. Gallardo, L. M. G., & Buleje, J. C. M. (2011). *IMPORTANCIA DE LOS PROGRAMAS VIRTUALES EN LA EDUCACIÓN SUPERIOR PERUANA*. 14.

22. Gazarov, P. (2019, diciembre 19). *What is an API? In English, please*. freeCodeCamp.org. https://www.freecodecamp.org/news/what-is-an-api-in-english-please-b880a3214a82/

23. Gergana. (s. f.). *Intro to Github for version control*. Recuperado 31 de mayo de 2020, de https://ourcodingclub.github.io/tutorials/git/

24. Heller, M. (2018, abril 2). *What is GitHub? More than Git version control in the cloud*. https://search.proquest.com/docview/2020617639/D622F368CD2C437FPQ/5?accountid=35325

25. Herrera-Cubides, J. F., Gelvez-García, N. Y., & López-Sarmiento, D. A. (2019). LMS SaaS: Una alternativa para la formación virtual. *Ingeniare: Revista Chilena de Ingenieria; Arica*, *27*(1), 164-179.

26. iNBest. (s. f.-a). *Amazon DynamoDB*. Recuperado 1 de junio de 2020, de https://www.inbest.cloud/amazon-dynamodb

27. iNBest. (s. f.-b). *Amazon Simple Storage Service*. Recuperado 1 de junio de 2020, de https://www.inbest.cloud/amazon-simple-storage-service

28. *Introducción a Express/Node*. (s. f.). Documentación web de MDN. Recuperado 1 de junio de 2020, de https://developer.mozilla.org/es/docs/Learn/Server-side/Express_Nodejs/Introduction

29. Laureate International Universities. (s. f.-a). *Booking*. Recuperado 20 de julio de 2020, de https://booking-fe-staging.ed-tech.cloud/lti/room

30. Laureate International Universities. (s. f.-b). *SUMADI Q v0.6.5*.

31. Laureate International Universities. (s. f.-c). *SUMADI v0.6.2*.

32. Performance testing software | Introduction of Apache Jmeter. (2020, enero 21). *Devstringx Technologies*. https://www.devstringx.com/introduction-of-apache-jmeter-open-source-performance-testing-software

33. PI Planning. (s. f.). *Scaled Agile Framework*. Recuperado 31 de mayo de 2020, de https://www.scaledagileframework.com/pi-planning/

34. *¿Qué es Amazon EC2? - Amazon Elastic Compute Cloud*. (s. f.). Recuperado 1 de junio de 2020, de https://docs.aws.amazon.com/es_es/AWSEC2/latest/UserGuide/concepts.html

35. Robie, J. (s. f.). *¿Qué es el Modelo de Objetos del Documento?* Recuperado 1 de junio de 2020, de http://html.conclase.net/w3c/dom1-es/introduction.html

36. rubenfa. (2014, febrero 3). *MongoDB: Qué es, cómo funciona y cuándo podemos usarlo (o no)*. Genbeta. https://www.genbeta.com/desarrollo/mongodb-que-es-como-funciona-y-cuando-podemos-usarlo-o-no

37. Schwaber, K., & Sutherland, J. (2016). *La Guía Definitiva de Scrum: Las Reglas del Juego*. https://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-Spanish.pdf#zoom=100

38. *Sobre Laureate*. (s. f.). Laureate. Recuperado 30 de mayo de 2020, de https://www.laureate.net/es/about-laureate/

39. *Tenable.io*. (2017, enero 20). Tenable®. Recuperado 6 de septiembre de 2020 https://es-la.tenable.com/products/tenable-io

40. Universidad Internacional de Valencia. (s. f.). *Características, tipos y plataformas más utilizadas para estudiar a distancia | VIU*. Recuperado 31 de mayo de 2020, de https://www.universidadviu.com/caracteristicas-tipos-y-plataformas-mas-utilizadas-para-estudiar-a-distancia/

41. *What is Github?* (2016, abril 7). https://guides.github.com/activities/hello-world/

42. *What is REST?* (s. f.). Codecademy. Recuperado 1 de junio de 2020, de https://www.codecademy.com/articles/what-is-rest

43. *What is SAFe*. (s. f.). Scaled Agile. Recuperado 31 de mayo de 2020, de https://www.scaledagile.com/enterprise-solutions/what-is-safe/

44. *Wikis*. (2012, junio 8). Vanderbilt University. https://cft.vanderbilt.edu/guides-sub-pages/wikis/

45. Yeo, M., & Arazy, O. (2012). *What Makes Corporate Wikis Work? Wiki Affordances and Their Suitability for Corporate Knowledge Work*. *7286*. https://doi.org/10.1007/978-3-642-29863-9_14