



**UNIVERSIDAD TECNOLÓGICA CENTROAMERICANA**

**FACULTAD DE INGENIERÍA**

**PRÁCTICA PROFESIONAL**

**LAUREATE INTERNATIONAL UNIVERSITIES**

**PREVIO A LA OBTENCIÓN DEL TÍTULO**

**INGENIERO EN SISTEMAS COMPUTACIONALES**

**PRESENTADO POR:**

**11441013 RICARDO JOSÉ GALDÁMEZ ANDINO**

**ASESOR: LIC. TANIA LUCILA MEZA AMADOR**

**CAMPUS TEGUCIGALPA; MAYO, 2020**

## **AGRADECIMIENTOS**

Se agradece a Laureate International Universities por proporcionar un espacio de trabajo para el practicante Ricardo José Galdámez Andino, y a UNITEC por dar soporte y asesoría en el transcurso de la práctica profesional.

## LISTA DE SIGLAS Y GLOSARIO

AI	Inteligencia Artificial
AWS	Servicios en la Nube de Amazon
CRAI	Centro de Recursos para el Aprendizaje y la Investigación
FI	Facultad de Ingeniería
GDPR	Regulación de Protección de Datos Generales
IT	Tecnología de Información
IP	Protocolo de Internet
LMS	Sistema de Gestión de Aprendizaje
LTI	Interoperabilidad de Herramientas de Aprendizaje
OE	Objetivo Específico
OG	Objetivo General
PaaS	Plataforma como Servicio
PBC	Corporación de Beneficio Público
PBSC	Punto de Referencia para el Control de Seguridad del Producto
PI	Incremento de Programa
PP	Práctica Profesional
SQL	Lenguaje de consulta estructurada

TCP    Protocolo de Control de Transmisión

UDP    Protocolo de Datagramas de Usuario

QA    Control de Calidad

Back-end                      Es toda la tecnología necesaria para procesar solicitudes entrantes y enviar respuestas al cliente(*Back-End Web Architecture, s/f*).

CodeBuild                    Servicio de integración continua completamente administrado que compila código fuente, ejecuta pruebas y produce paquetes de software listos para su implementación(*AWS CodeBuild – Servicio de compilación completamente administrado, s/f*).

CodePipeline                Servicio de entrega continua completamente administrado que permite automatizar canalizaciones de lanzamiento para lograr actualizaciones de infraestructura y aplicaciones rápidas y fiables(*AWS CodePipeline | Integración y entrega continuas, s/f*).

Corporación de Beneficio Público    Corporación creada por el gobierno que realiza funciones específicas para el beneficio del público(*Public benefit corporation, s/f*).

DevOps                        Combinación de las prácticas de desarrollo de software y operaciones de tecnología de la información(*¿Qué es DevOps?, s/f*).

Elastic Compute Cloud                  Amazon Elastic Compute Cloud (Amazon EC2) es un servicio web que proporciona capacidad informática en la nube

	segura y de tamaño modificable(AWS   <i>Elastic compute cloud (EC2) de capacidad modificable en la nube</i> , s/f, p. 2).
Elastic Beanstalk	Servicio fácil de utilizar para implementar y escalar servicios y aplicaciones web desarrollados con Java, .NET, PHP, Node.js, Python, Ruby, Go y Docker en servidores familiares como Apache, Nginx, Passenger e IIS(AWS   <i>Elastic beanstalk para aplicaciones web desarrolladas con Java</i> , s/f).
E-Learning	El aprendizaje electrónico se refiere a la enseñanza y aprendizaje virtual en línea, desde internet y por medios tecnológicos(Manzanero, 2003).
Front-end	Sección gráfica de una aplicación web que permite al usuario final interactuar con los datos mostrados( <i>What Is a Front-End Developer?</i> , s/f).
OctoPerf	Plataforma de pruebas de carga en la nube( <i>What is OctoPerf? - Documentation—OctoPerf</i> , s/f).
SCRUM	Un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos("Qué es SCRUM", 2008).
SUMADI	Aplicación de supervisión automatizada que utiliza inteligencia artificial para identificar y corroborar la identidad de estudiantes que estén realizando exámenes en línea ( <i>Product – Sumadi</i> , s/f).

SUMADI Management Site

Aplicación que maneja los contratos para el uso de SUMADI en las instituciones que está instalado SUMADI.

Sprint

Periodo de trabajo establecido en el marco de trabajo de SCRUM("Qué es SCRUM", 2008).

## **RESUMEN EJECUTIVO**

El presente informe se entrega como requisito para optar al título profesional de Ingeniero en Sistemas Computacionales, donde se describen las actividades desarrolladas durante la práctica profesional realizada de octubre del 2019 a marzo del 2020. La práctica se realizó en el departamento de Tecnologías de Información en la sección de sistema de gestión del aprendizaje (LMS) bajo una subsección de arquitectura realizando tareas de apoyo a los desarrolladores como configuraciones en los servidores, cambios en bases de datos y lanzando nuevas versiones de SUMADI al mercado.

En el año 2019 se comenzó el desarrollo de una aplicación con el objetivo de supervisar exámenes en línea sin necesidad de un profesor observando las acciones de los estudiantes y permitir más oportunidades a los alumnos que por alguna razón no pueden llegar a realizar un examen en un lugar en específico. Esta aplicación inicialmente se llamó Laureate Proctor y luego fue renombrada SUMADI, una solución que utiliza inteligencia artificial para supervisar a los alumnos haciendo exámenes y pruebas en línea.

En el departamento de LMS se les ha dado soporte a universidades que son parte de la red de Laureate en cuanto a sus plataformas de manejo de aprendizaje, incluyendo creación de software integrador y la continuidad de estos, específicamente para las universidades que utilizan BlackBoard como su plataforma de aprendizaje en línea.

La subsección de arquitectura, donde se ha ejercido la práctica, ha manejado de igual manera la infraestructura y seguridad de las aplicaciones creadas dentro del departamento de LMS y es donde se maneja SUMADI a nivel de producción, lanzando diferentes versiones de esta al mercado; este grupo ha analizado los diferentes recursos necesarios para que la aplicación pueda ejecutarse sin problemas con un alto nivel de disponibilidad considerando el presupuesto dado para seguir operando.

El equipo de arquitectura ha manejado pruebas de control de calidad asegurándose de minimizar la cantidad de errores encontrados en producción, también se han ejecutado pruebas de carga simulando la cantidad de personas utilizando la aplicación antes de un uso real para verificar una implementación correcta de la misma.

## ÍNDICE DE CONTENIDO

I.	Introducción.....	1
II.	Generalidades de la Empresa.....	3
2.1	Descripción de la Empresa.....	3
2.2	Descripción del Departamento.....	4
2.3	Objetivos del Puesto.....	4
2.3.1	Objetivo General.....	4
2.3.2	Objetivos Específicos.....	4
III.	Marco Teórico.....	5
IV.	Desarrollo.....	20
4.1	Descripción del Trabajo Realizado.....	20
4.1.1	Planeamiento del Último Período de Trabajo de 2019.....	20
4.1.2	Blackboard Training.....	23
4.1.3	Pruebas de Regresión de SUMADI.....	24
4.1.4	Proveer Retroalimentación de Errores.....	25
4.1.5	Obtener Información del Wiki Actual.....	26
4.1.6	Subir Información del Wiki Anterior al Nuevo.....	28
4.1.7	Analizar Políticas de Seguridad.....	30
4.1.8	Crear Políticas de Seguridad y Subirlas a Confluence.....	30
4.1.9	Lanzamientos de la Aplicación SUMADI.....	34
4.1.10	Pruebas de Carga Con OctoPerf.....	37
4.1.11	Control de Calidad con Casos de Uso.....	38
4.1.12	Soporte y Lanzamiento de la Aplicación SUMADI en Electron.....	39
4.2	Cronograma de Actividades.....	42

V. Conclusiones.....	43
VI. Recomendaciones.....	44
Bibliografía.....	45

## ÍNDICE DE ILUSTRACIONES

Ilustración 1 – Habilidades para la Colaboración de un Wiki.....	7
Ilustración 2 – Diagrama de una Base de datos Clave-Valor.....	12
Ilustración 3 – Datos en una Base de Datos de Documentos .....	13
Ilustración 4 – Esquema de un Gráfico .....	15
Ilustración 5 – Aplicación web usando Express .....	16
Ilustración 6 – Aplicación utilizando React.....	16
Ilustración 7 – Diagrama de una Aplicación Web.....	17
Ilustración 8 – Agenda Estándar de la reunión en un PI Planning .....	20
Ilustración 9 – Entrenamiento para el uso del LMS Blackboard.....	23
Ilustración 10 – LMS de Blackboard para prueba de SUMADI .....	24
Ilustración 11 – Aplicación de SUMADI.....	25
Ilustración 12 – Documentación de un Error de SUMADI.....	26
Ilustración 13 – Wiki en Google Sites de la arquitectura LMS de Laureate .....	27
Ilustración 14 – Página de aprendizaje en Wiki de Laureate.....	28
Ilustración 15 – Nuevo Wiki de la arquitectura LMS de Laureate en Confluence .....	29
Ilustración 16 – Nueva página de aprendizaje de la arquitectura LMS de Laureate .....	29
Ilustración 17 - Índice de las Políticas para el Manejo de Cambios.....	31
Ilustración 18 – Índice del Estándar para la Gerencia de Parches .....	32
Ilustración 19 – Página de gerencia de cambio en el Wiki Confluence.....	33
Ilustración 20 – Página de gerencia de parches en el Wiki Confluence.....	34
Ilustración 21 – Documentación de las Funcionalidades Nuevas de SUMADI.....	35
Ilustración 22 – CodeBuilds utilizados para empaquetar código .....	36
Ilustración 23 – Ambientes de SUMADI y Management Site en Elastic Beanstalk.....	37

Ilustración 24 – Pruebas de Carga utilizando Octoperf.....	38
Ilustración 25 – Casos de uso creados en Azure DevOps.....	38
Ilustración 26 – Ejecución de un Caso de Uso .....	39
Ilustración 27 – Creación de una Guía para Actualizar programas en Electron .....	40
Ilustración 28 – Aplicación de SUMADI nativa actualizándose automáticamente .....	41
Ilustración 29 – Diagrama de Gantt .....	42

## **I. INTRODUCCIÓN**

Con la tecnología constantemente evolucionando para enriquecer nuestras vidas día a día, se han creado tantas dependencias sobre esta que se ha vuelto una base fundamental para el trabajo, ocio, transporte, comodidad e incluso educación. Para adaptarse a la tecnología, instituciones escolares han comenzado a integrar dispositivos electrónicos como computadoras, tabletas, proyectores, entre otros. Las computadoras se utilizan para acceder a información por medio de internet y librerías virtuales, para trabajar con editores de texto, para generar archivos y hacer entrega de tareas y proyectos.

Gracias a la tecnología y el deseo de incorporarla a la educación, se han logrado plataformas en la nube accesibles desde cualquier lugar que los maestros y alumnos utilizan para comunicarse, para crear asignaciones que los alumnos deben entregar, compartir recursos y de esta manera generar comunidades que se incorporan en clases de manera parcial o de forma completa. Para facilitar la evaluación de conocimientos de los estudiantes, los maestros de ahora tienen acceso a crear evaluaciones en línea que son más fáciles de analizar y calificar, sin embargo, las aplicaciones recientes en este ambiente, se pueden explotar por parte de los estudiantes cuando no hay sistemas específicamente diseñados para que no se utilicen motores de búsqueda o documentos en línea para encontrar respuestas a estos exámenes virtuales, sobre todo cuando los docentes no están prestando atención constante a los mismos.

Aquí es donde entra en acción el sistema de supervisión a base de inteligencia artificial llamado SUMADI, el cual no necesita a un docente presente para asegurar que el estudiante completará su evaluación en línea sin hacer trampa, lo que al contrario resultaría en que el estudiante obtenga mejores notas que no reflejan sus conocimientos de manera precisa. SUMADI, un sistema para la red de universidades de Laureate que genera muchas más posibilidades para una educación a nivel mundial y completamente virtual, también gracias a este sistema automatizado, se puede confiar más en la tecnología para aplicarse a la educación que no requiera más maestros o supervisores físicos y que inclusive se pueda implementar en niveles superiores de educación.

El Presente informe contiene una descripción de las generalidades de la empresa, donde se habla de Laureate International Universities, se describe el puesto donde se trabajó en esta práctica profesional y también se hacen saber los objetivos generales, así como los objetivos específicos que se desean llevar a cabo.

En el marco teórico, se mencionan las plataformas virtuales, las metodologías utilizadas en la empresa durante la práctica y los avances tecnológicos cada vez involucrándose más en el área educativa.

La sección de desarrollo muestra las actividades realizadas durante la práctica profesional y profundiza los detalles para obtener un mejor entendimiento de las asignaciones.

## 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 aprendices de por vida. De esta manera, somos *Here for Good*. 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*).

Alineados a nuestro compromiso de ser *Here for Good*, 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**

En el departamento de Tecnología de Información (IT) de Laureate, localizado en Honduras, se trabaja con las tecnologías y aplicaciones que se utilizan en la red de universidades de Laureate. Se colabora con Blackboard para entregar a las universidades plataformas virtuales en las cuales los estudiantes pueden ver sus asignaciones y hacer entrega de estas, también dan otros usos que asisten en la pedagogía.

## **2.3 OBJETIVOS DEL PUESTO**

### 2.3.1 OBJETIVO GENERAL

Asistir en la aplicación de supervisión basada en Inteligencia Artificial (A.I.), SUMADI, elevando los estándares de la aplicación en cuanto a seguridad, confiabilidad y precisión para que esta obtenga una calidad relevante en su uso internacional.

### 2.3.2 OBJETIVOS ESPECÍFICOS

- Generar políticas de gestión de cambios y gestión de parches para cumplir con los estándares de seguridad impuestos por Blackboard y permitir la operación continua de SUMADI en su plataforma de aprendizaje virtual.
- Realizar control de calidad del sistema SUMADI y su integración con Blackboard.
- Asistir en el despliegue de aplicaciones y servidores de SUMADI.

### **III. MARCO TEÓRICO**

Para obtener un mejor entendimiento del uso del sistema de supervisión SUMADI, a continuación, se describirán términos y herramientas relacionadas a esta y a su creación.

Una plataforma de enseñanza virtual es una aplicación informática a la que se accede a través de la red, que permite a los profesores elaborar materiales docentes y ponerlos a disposición de los alumnos en internet. Las plataformas virtuales representan herramientas que complementan la enseñanza presencial y, en otro nivel, posibilitan la educación a distancia (Ayllón, 2010)

En la actualidad, existen múltiples plataformas virtuales en el ámbito educativo. Estos espacios establecidos en el internet son implementados en varios centros formativos, incluyendo universidades, centros escolares y como menciona Ayllón (2010), Administraciones públicas, empresas para la formación de su personal, entre otros. Las plataformas tienden a utilizar herramientas similares que frecuentemente permiten la exportación de estas a otras plataformas virtuales.

Es aquí donde entra el Sistema de Gestión de Aprendizaje (LMS) el cual es un software instalado generalmente en un servidor web (puede instalarse en una intranet), que se emplea para crear, aprobar, administrar, almacenar, distribuir y gestionar las actividades de formación virtual (puede utilizarse como complemento de clases presenciales o para el aprendizaje a distancia) (Clarenc et al., 2013)

LMS se construye encima de una plataforma virtual de aprendizaje, permitiendo que no solo se puede utilizar para administrar estudiantes y dar seguimiento al aprendizaje de estos, sino que también según Castro et al. (2013), se integren materiales didácticos y herramientas de comunicación, colaboración y gestión educativa.

Boneu (2007) se refiere a que las plataformas virtuales de aprendizaje deben tener las siguientes características fundamentales:

- Interactividad: Conseguir que la persona que está usando la plataforma tenga conciencia de que es el protagonista de su formación.
- Flexibilidad: Conjunto de funcionalidades que permiten que el sistema de e-learning tenga una adaptación fácil en la organización donde se quiere implantar.
- Escalabilidad: Capacidad de la plataforma de e-learning de funcionar igualmente con un número pequeño o grande de usuarios.
- Estandarización: Hablar de plataformas estándares es hablar de la capacidad de utilizar cursos realizados por terceros; de esta forma, los cursos están disponibles para la organización que los ha creado y para otras que cumplen con el estándar(Boneu, 2007).

Existen diferentes tipos de LMS dependiendo del uso que se le esté dando a estos:

1. LMS de Software Libre: Este tipo de LMS es libre y utilizable en un ambiente comercial.
2. LMS Comercial: LMS el cual se considera un servicio pagado.
3. LMS en la Nube: LMS comercial que está disponible en línea. La base de datos y las políticas son manejadas por el equipo IT de la empresa de la que se origina el LMS.

SUMADI es una herramienta que se integra a los LMS de forma modular para fácilmente ser incluida en las plataformas virtuales utilizadas en varias universidades localizadas en diferentes regiones.

La metodología de trabajo utilizada en Laureate en el área de Tecnología de la Información (IT) es llamada SCRUM, el cual es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos(*¿Qué es Scrum?*, s/f).

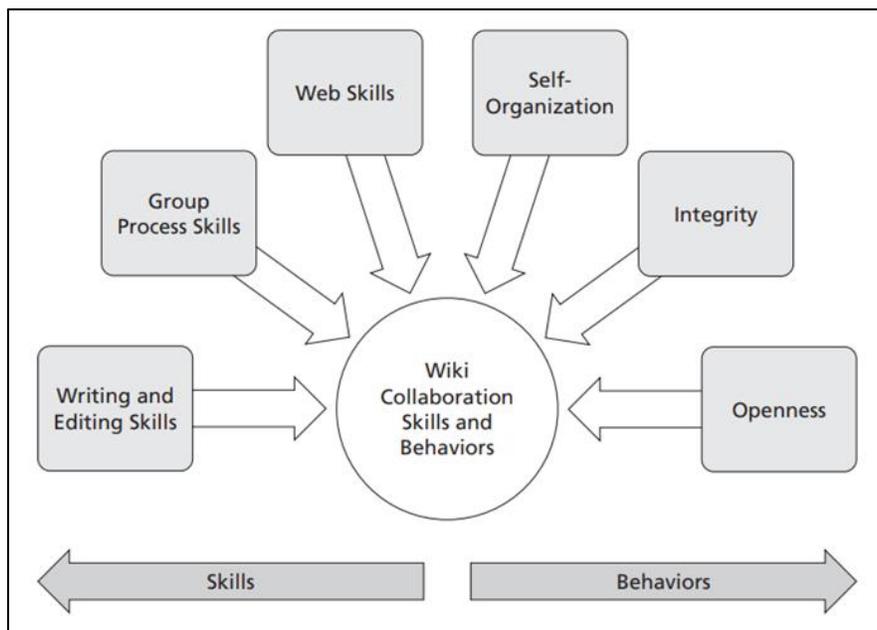
El equipo de LMS de Laureate utiliza un Wiki como una plataforma de colaboración en línea y es ahí donde se manejan diferentes documentos y recursos que son utilizados diariamente.

Mitchell (Mitchell, 2019) define un wiki como una aplicación web cuyo contenido es manejado colaborativamente por sus usuarios.

Un wiki busca involucrar al usuario típico en un proceso de colaboración, lo cual genera cambios constantes en la página web (Cunningham & Leuf, 2001).

Gracias a esto, una gran cantidad de usuarios pueden trabajar en un solo centro de información, sin necesidad de ser un experto en páginas web o un escritor profesional. Es por eso por lo que se usa un wiki en la arquitectura LMS de Laureate, donde se aprovecha la colaboración de todos los miembros.

En la Ilustración 1 se pueden observar las habilidades para la colaboración en el wiki.



**Ilustración 1 – Habilidades para la Colaboración de un Wiki**

Fuente: (West & West, 2009)

Amazon Web Services (AWS) es la plataforma en la nube más adoptada y completa en el mundo, que ofrece más de 175 servicios integrales de centros de datos a nivel global. Millones de clientes, incluyendo las empresas emergentes que crecen más rápido, las compañías más grandes y los organismos gubernamentales líderes, están utilizando AWS para reducir los costos, aumentar su agilidad e innovar de forma más rápida (¿Qué es AWS?, s/f).

Los servicios en la nube de Amazon se utilizan para una variedad de aplicaciones, las más utilizadas en el área de arquitectura en el departamento de LMS son las siguientes:

Amazon Elastic Compute Cloud, el cual se utiliza para manejar instancias de servidores en la nube y eliminar la necesidad de configurar computadoras en un espacio físico en la oficina. Este servicio facilita la integración con otros servicios de Amazon y mitiga problemas de disponibilidad de los servidores al replicar estos en sus regiones asignadas.

- Se utilizan diferentes componentes del servicio Elastic Compute Cloud para la infraestructura de los servidores manejados en el área de LMS como las imágenes virtuales, donde se pueden alterar características de un sistema operativo instalado en algún servidor y este se puede convertir en una imagen predeterminada para instalar en un servidor nuevo o reinstalar en un servidor que ha sido actualizado y ha dejado de funcionar correctamente.
- Las tomas instantáneas de los servidores guardan el estado actual del servidor para restablecerlo en caso de que algún despliegue a producción perjudique el servidor en un momento crítico y exista la necesidad de restablecer el ambiente a su normalidad.
- En Elastic Compute Cloud, se manejan grupos de seguridad, en donde se diseñan los diferentes tipos de acceso a los servidores por medio de puertos del Protocolo de Control de Transmisión (TCP) y el Protocolo de Datagramas de Usuario (UDP).
- Se manejan Protocolos de Internet (IP) estáticos para los servidores que no necesitan estar registrados en un dominio en internet, particularmente para servidores de uso interno como la plataforma de prueba de BlackBoard y servidores de seguridad como Trend Micro
- Al ser los servidores de SUMADI instancias de Elastic Compute Cloud, estos tienen el beneficio de usar los servicios de balanceo de carga de Amazon, lo cual permite manejar la capacidad de pedidos al servidor de manera dinámica; esto implica un uso eficiente de los recursos y un ahorro económico en comparación con otros proveedores o manejando servidores físicos(AWS | *Elastic compute cloud (EC2) de capacidad modificable en la nube, s/f*).

Otro de los servicios que se usan para manejar los servidores de Amazon es llamado Elastic Beanstalk, el cual facilita el despliegue de aplicaciones hacia las instancias de Elastic Compute Cloud de Amazon.

- Elastic Beanstalk reduce la complejidad a nivel estructural del manejo de los servidores al tener este servicio que gestiona automáticamente varias configuraciones de manera óptima, aunque sean opciones predeterminadas y requieran cambios configuracionales para agregar otros servicios o adaptar programas específicos.
- Principalmente LMS utiliza Elastic Beanstalk como plataforma para alojar aplicaciones desarrolladas en Node.js como SUMADI.
- La sección de configuraciones permite agregar variables de entorno que luego se usarán en la aplicación una vez esté presente en la instancia.
- Uno de los servicios más importantes de Elastic Beanstalk es el de tener un registro que permite ver el estado del servidor sin conectarse directamente a él, esto facilita ver si algo está fallando o requiere de más acciones para funcionar efectivamente.

CodeBuild es el servicio utilizado para lanzar aplicaciones hacia ambientes de Elastic Beanstalk, las cuales en realidad son instancias de Elastic Compute Cloud(*AWS CodeBuild – Servicio de compilación completamente administrado, s/f*).

- Codebuild ofrece el uso de computación en la nube para generar y empaquetar programas en un ambiente de pruebas y de producción.
- Para ver el estado del código, se utiliza SonarQube, el cual inspecciona el código y permite identificar errores a temprana edad antes de que estos generen un alto impacto negativo.

CodePipeline es la herramienta que consigue una aplicación generada y optimizada que está lista para ser lanzada a un ambiente. En este caso, CodePipeline consigue la aplicación generada por CodeBuild y la coloca en un ambiente de Elastic Beanstalk, el cual refleja su aplicación en varios minutos dependiendo de la complejidad del programa(*AWS CodePipeline | Integración y entrega continuas, s/f*).

En cuanto a bases de datos y otros servicios para guardar información, se utilizan las siguientes plataformas:

Amazon DynamoDB es una base de datos NoSQL que está diseñado para escalar exponencialmente en cuanto a pedidos por segundo. Es una base de datos muy flexible en cuanto a configuraciones y tipos de documentos que se pueden guardar en tablas, lo que permite usos diferentes y se tienen que analizar los patrones de acceso para aprovechar al máximo la base de datos. Tablas de DynamoDB son utilizadas para la aplicación de Gestión de SUMADI (Management Site). DynamoDB aumenta sus recursos de manera automática para poder seguir operando bajo un uso constante por parte de numerosos usuarios.

Amazon Simple Storage Service es otro servicio de Amazon que se utiliza para almacenar objetos en "cubetas" virtuales que pueden ser accedidos en cualquier parte del mundo.

- Este servicio es escalable y es posible guardar cualquier tipo de archivo en cubetas.
- Las cubetas tienen un nombre único en todo el mundo ya que representan una IP que se puede acceder por medio de un navegador de internet utilizando su nombre en la barra de direccionamiento si se da acceso público o específico a la cubeta.
- Las cubetas pueden utilizarse como plataformas de alojamiento para páginas web estáticas

CosmosDB es una base de datos creada por Microsoft para manejar datos a nivel mundial, este permite uso de varias interfaces para programación de aplicaciones como:

- Lenguaje de Consulta Estructurada (SQL): En 1973, Edgar Codd crea un nuevo modelo para las bases de datos llamada Modelo Relacional el cual apuntaba a mitigar problemas identificados con el paradigma de la navegación: cualquier usuario tenía que navegar una cantidad significativa de complejidad para obtener los datos que buscaba. Su modelo, articulado en "Un modelo relacional de datos para grandes bancos de datos compartidos", era una revolución conceptual: en lugar de concebir los datos como un simple medio de organización, la base de datos podía utilizarse ahora como una herramienta de consulta de datos para encontrar relaciones ocultas en su interior(Avant.org & Fortune, s/f).

IBM desarrolló el lenguaje de consulta estructurado en inglés (SEQUEL) que luego se volvió SQL en 1979 y ahora es aceptado como el modelo definitivo para sistemas de gestión de bases de datos relacionales(*History of SQL*, s/f).

- MongoDB: fue desarrollado por una organización con sede en Nueva York llamada 10gen que ahora se conoce como MongoDB Inc. Fue desarrollado inicialmente como una Plataforma como Servicio (PaaS). Luego, en 2009, se introduce en el mercado como un servidor de base de datos de código abierto que fue mantenido y apoyado por MongoDB Inc. (*MongoDB History—Javatpoint, s/f*).

A mediados del decenio de 1990, el Internet adquirió una popularidad extrema y las bases de datos relacionales simplemente no pudieron seguir el ritmo del flujo de información exigido por los usuarios, así como la mayor variedad de tipos de datos que se produjeron a partir de esta evolución. Esto llevó al desarrollo de bases de datos no relacionales, a menudo denominadas NoSQL. Las bases de datos NoSQL pueden traducir rápidamente datos no estructurados y evitar la rigidez de SQL sustituyendo el almacenamiento "organizado" por una mayor flexibilidad(Foote, 2018).

Las bases de datos NoSQL tienen diferentes maneras de guardas datos en comparación a las tablas de las bases de datos SQL, como:

- Almacenamiento de valores clave
- Almacenamiento de documentos
- Almacenamiento en columna ancha
- Base de datos de gráficos

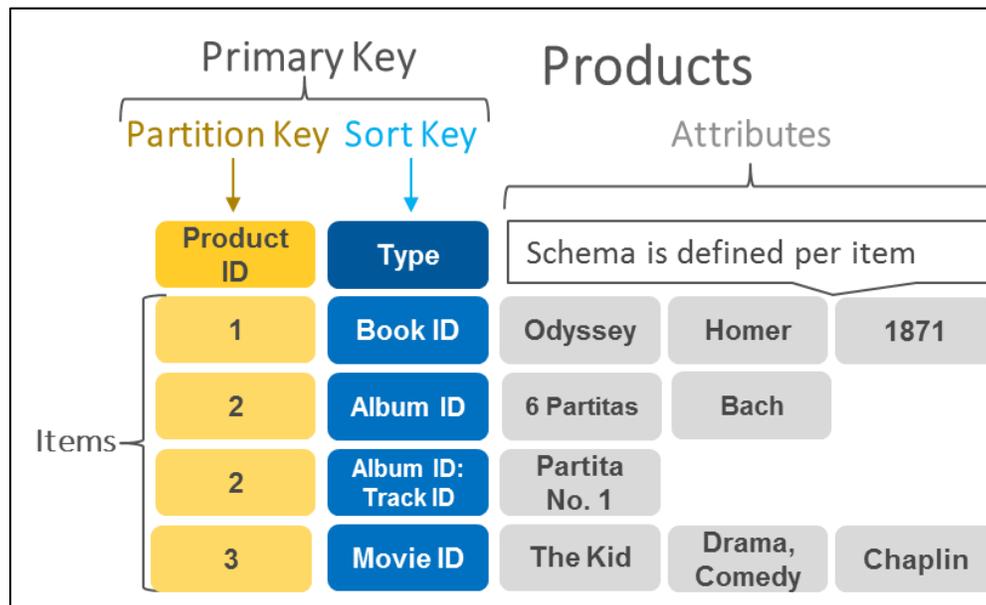
Almacenamiento de Valores Clave: Es un sistema de almacenamiento de datos diseñado para el almacenamiento, recuperación y gestión de "matrices asociativas". Un almacén de valores clave funciona de forma muy diferente a una base de datos relacional. Una base de datos relacional predefine la estructura de los datos, utilizando una serie de tablas que contienen campos con tipos de datos bien definidos. Este almacén de datos elige entre una variedad de opciones óptimas a la hora de clasificar los tipos de datos(Foote, 2018).

Algunos ejemplos de bases de datos con Almacenamiento de Valores Clave son:

- Aerospike

- Apache Ignite
- Berkeley DB
- DynamoDB
- Redis

En la Ilustración 2 se muestra un diagrama de cómo se almacenan datos en una base de datos clave-valor en DynamoDB.



**Ilustración 2 – Diagrama de una Base de datos Clave-Valor**

Fuente: (¿Qué es una base de datos clave-valor?, s/f)

Almacenamiento de documentos: Es un sistema diseñado para el almacenamiento, la recuperación y la gestión de "información orientada a los documentos", que también se denomina datos semiestructurados. Los almacenes de documentos tienen algunas similitudes con los almacenes de valor clave, pero difieren en la forma en que se procesan los datos. Un sistema orientado a los documentos utiliza la estructura interna del documento para su identificación y almacenamiento. Los Almacenes de Documentos guardan toda la información de un elemento determinado como una sola instancia en una base de datos (en lugar de repartirla en tablas, como

en los sistemas relacionales). Esto facilita la asignación de los artículos a la base de datos(Foote, 2018).

Algunos ejemplos de bases de datos en base al Almacenamiento de documentos son:

- CouchDB
- DocumentDB
- CosmosDB
- MongoDB
- PostgreSQL

En la Ilustración 3 se muestra una comparación de los datos guardados en una base de datos relacional genérica y la misma información en una base de datos que maneja documentos utilizando un esquema JSON(MongoDB).

**Relational**

ID	first_name	last_name	cell	city	year_of_birth	location_x	location_y
1	'Mary'	'Jones'	'516-555-2048'	'Long Island'	1986	'-73.9876'	'40.7574'

ID	user_id	profession
10	1	'Developer'
11	1	'Engineer'

ID	user_id	name	version
20	1	'MyApp'	1.0.4
21	1	'DocFinder'	2.5.7

ID	user_id	make	year
30	1	'Bentley'	1973
31	1	'Rolls Royce'	1965

**MongoDB**

```
{
  first_name: "Mary",
  last_name: "Jones",
  cell: "516-555-2048",
  city: "Long Island",
  year_of_birth: 1986,
  location: {
    type: "Point",
    coordinates: [-73.9876, 40.7574]
  },
  profession: ["Developer", "Engineer"],
  apps: [
    { name: "MyApp",
      version: 1.0.4 },
    { name: "DocFinder",
      version: 2.5.7 }
  ],
  cars: [
    { make: "Bentley",
      year: 1973 },
    { make: "Rolls Royce",
      year: 1965 }
  ]
}
```

**Ilustración 3 – Datos en una Base de Datos de Documentos**

Fuente: (What is a Document Database?, s/f)

Almacenamiento en Columna Ancha: Utiliza tablas, filas y columnas, pero a diferencia de las bases de datos relacionales, los nombres y formatos de las columnas pueden cambiar de fila a fila dentro de la misma tabla. Son más flexibles. Los almacenamientos de columnas anchas suelen admitir familias de columnas, que se almacenan por separado. Cada familia de columnas normalmente contiene varias columnas utilizadas conjuntamente. Dentro de una familia de columnas específica, los datos se almacenan fila por fila, y las columnas de una fila específica se almacenan juntas en lugar de que cada columna se almacene individualmente. Las familias de columnas de apoyo se almacenan también en una amplia gama de columnas, que se denominan bases de datos de familias de columnas(Foote, 2018).

Algunos ejemplos de este tipo de base de datos son:

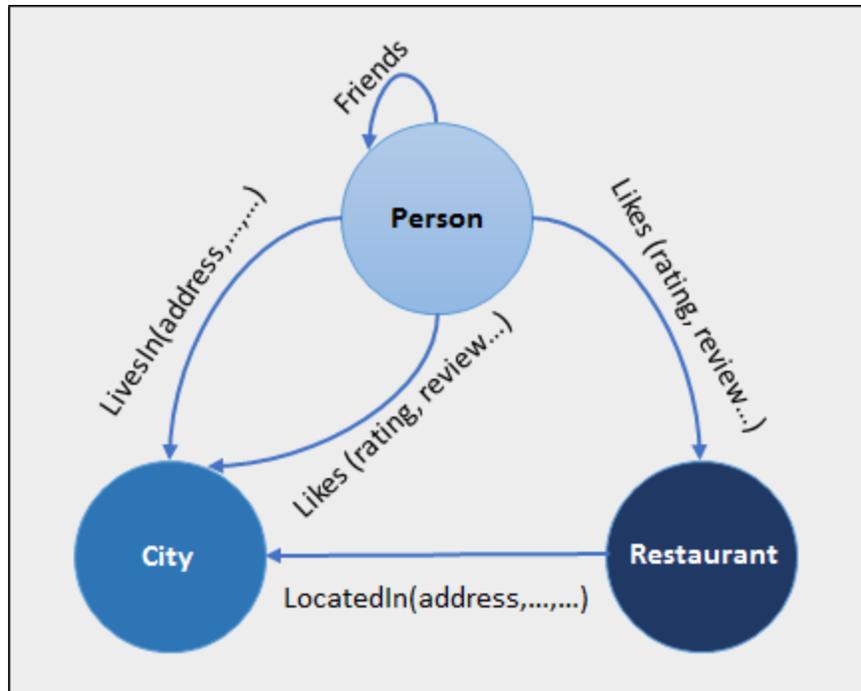
- Cassandra
- CosmosDB
- DynamoDB

Base de Datos de Gráficos: Es esencialmente una colección de relaciones. Cada memoria (un nodo) simboliza una entidad (un negocio, una persona o un objeto). Cada memoria/nodo está conectado a otro. La conexión se denomina "borde" y representa una relación entre dos nodos. Cada nodo dentro de una base de datos de gráficos incluye un identificador único, un conjunto de bordes entrantes y/o salientes y características que se representan como "pares clave-valor". Cada borde también viene con un identificador único, un nodo final y/o un nodo de inicio, y una colección de propiedades(Foote, 2018).

Algunos ejemplos serían:

- Amazon Neptune
- ArangoDB
- Neo4j

En la Ilustración 4 se muestra un esquema genérico de una base de datos de Gráficos para una red social hipotética que maneja personas, restaurantes y ciudades.



**Ilustración 4 – Esquema de un Gráfico**

Fuente: (shkale-msft, s/f)

MongoDB es una base de datos popular de código abierto al público NoSQL que se usa constantemente con aplicaciones creadas en Node.js. Al ser SUMADI una herramienta en línea que requiere manejar diferentes tipos de datos de una manera no estructurada, la base de datos escogida fue CosmosDB con interfaz de control de MongoDB.

Para comprender mejor la tecnología que es utilizada para mantener a SUMADI en línea, hay que primero indagar en el desarrollo de la aplicación en sí.

SUMADI es una aplicación web basado en una lista de herramientas de código abierto llamado MERN Stack, lo cual es un acrónimo de los siguientes componentes:

- MongoDB: Base de datos no relacional utilizado para guardar documentos con un formato JSON.
- Express: Marco de trabajo para el Back-end basado en Node.js.
- React: Librería de JavaScript para crear interfaces de usuario (*Getting Started – React, s/f*).

- Node.js: Entorno para servidores de código abierto(Node.js Introduction, s/f)

En la Ilustración 5 se muestra una aplicación simple utilizando Express como Back-end.

```
1 const express = require('express' 4.17.1 )
2 const app = express()
3 const port = 3000
4
5 app.get('/', (req, res) => res.send('Hello World!'))
6
7 app.listen(port, () => console.log(`Example app listening on port ${port}!`))
```

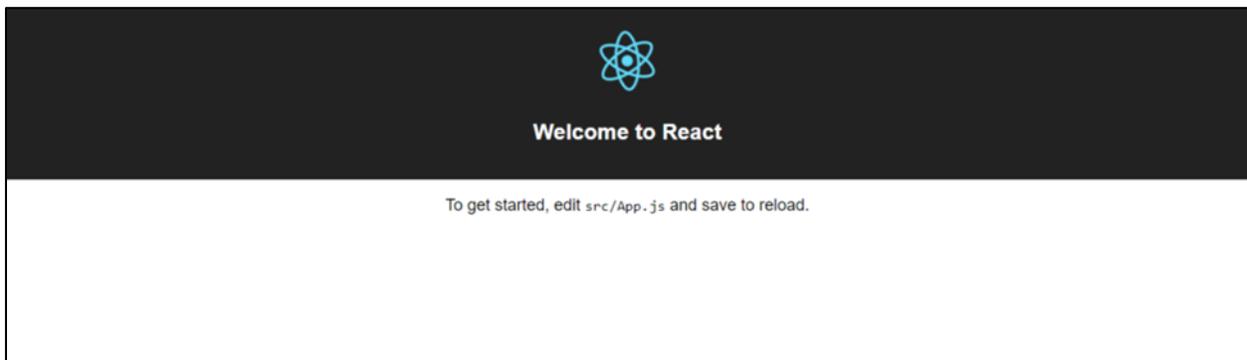
Save on RunKit Node 10 help URL: <https://sq8f8clo3mg2.runkit.sh>

22:00:52.431	▶ Request	GET /robots.txt	ip=	id=eaf786053ee1
22:00:52.432	🗨	"Example app listening on port 3000!"		
22:00:52.432	▶ Request	GET /	ip=	id=c43f63fa9800
22:00:52.445	▶ Response	GET /robots.txt	status=404 duration=	id=eaf786053ee1
22:00:52.448	▶ Response	GET /	status=200 duration=	id=c43f63fa9800
22:00:53.462	▶ Request	GET /favicon.ico	ip=	id=aa6f4a8186fb
22:00:53.466	▶ Response	GET /favicon.ico	status=404 duration=	id=aa6f4a8186fb

**Ilustración 5 – Aplicación web usando Express**

Fuente: (Express "Hello World" example, s/f)

En la Ilustración 6 se muestra una aplicación utilizando React como marco de trabajo del Front-end.

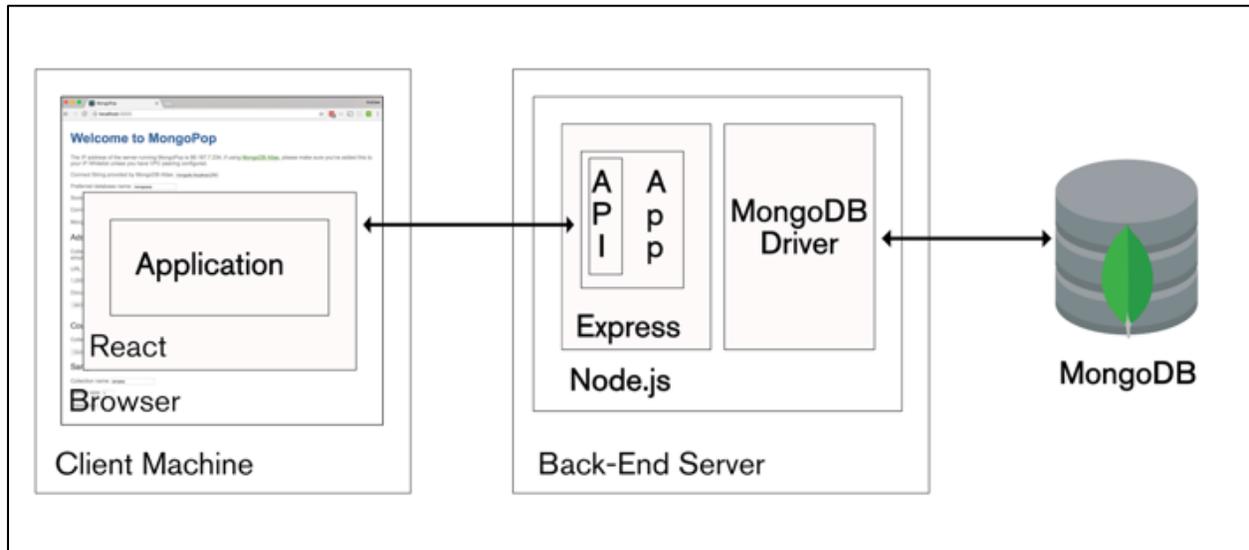


**Ilustración 6 – Aplicación utilizando React**

Fuente: (React – A JavaScript library for building user interfaces, s/f)

Ahora que se entienden mejor las herramientas utilizadas, se puede trabajar con SUMADI comprendiendo las necesidades de esta.

En la Ilustración 7 se muestra el uso de todas estas herramientas en un diagrama



**Ilustración 7 – Diagrama de una Aplicación Web**

Fuente: (*The Modern Application Stack – Part 1, s/f*)

Uno de los deberes del equipo de Arquitectura en el departamento de LMS cuando se da la oportunidad es de probar las aplicaciones desarrolladas de manera extensiva por medio de diferentes operaciones. Ahora para entender las asignaciones de Control de Calidad (QA) se necesita ver algunos de los tipos de prueba que este ejecuta o analiza:

- Prueba de Aceptación: Las pruebas de aceptación son un tipo formal de prueba de software que es realizada por el usuario final cuando las características han sido entregadas por los desarrolladores. El objetivo de esta prueba es comprobar si el software cumple sus necesidades de negocio y los requisitos proporcionados anteriormente. Las pruebas de aceptación se documentan normalmente al principio del sprint y es un medio para que los examinadores y desarrolladores trabajen hacia un entendimiento común y un

conocimiento compartido del dominio empresarial("Types of Software Testing—Complete List", s/f).

- Pruebas Automatizadas: se trata de un enfoque de verificación que utiliza herramientas de programación para ejecutar los casos de prueba diseñados a la medida. La mayoría de las herramientas automatizadas proporcionan la facilidad de captura y reproducción, sin embargo, hay herramientas que requieren escribir extensos guiones o programación para automatizar estos casos("Types of Software Testing—Complete List", s/f).
- Prueba de Caja Negra: es un método de prueba de software en el que no se requiere conocer la codificación o la estructura interna del software. El método de prueba de caja negra se basa en probar el software con varias entradas y validar los resultados contra los resultados esperados("Types of Software Testing—Complete List", s/f).
- Pruebas Funcionales: se centran en la comprobación del software contra el documento de diseño, los casos de uso y el documento de requisitos. La prueba funcional es un tipo de prueba de caja negra y no requiere el funcionamiento interno del software("Types of Software Testing—Complete List", s/f).
- Pruebas de Integración: son uno de los tipos más comunes e importantes de pruebas de software. Una vez que las unidades o componentes individuales son probados por los desarrolladores como trabajo, entonces el equipo de pruebas ejecutará pruebas que comprobarán la conectividad entre estas unidades/componentes o unidades/componentes múltiples("Types of Software Testing—Complete List", s/f).
- Pruebas de Carga: es un tipo de prueba no funcional; se hace para comprobar el comportamiento del software en condiciones de carga normal y de sobrecarga. Las pruebas de carga se realizan generalmente utilizando herramientas de prueba automatizadas y se usan para intentar encontrar cuellos de botella o problemas que impiden que el software funcione como se pretende en sus picos de trabajo("Types of Software Testing—Complete List", s/f).
- Pruebas de Aceptación de Usuario: son realizadas por los clientes/usuarios finales del software. Las pruebas de aceptación de usuario permiten a los expertos en la materia, los clientes, probar el software con sus negocios reales o escenarios del mundo real y

comprobar si el software cumple con los requisitos de su negocio("Types of Software Testing—Complete List", s/f).

- Prueba de Regresión: Es un tipo de prueba de software que se lleva a cabo por el equipo de control de calidad como pruebas de regresión funcional y los desarrolladores como pruebas de regresión de la unidad. El objetivo de estas pruebas es encontrar los defectos que se introdujeron para corregir los defectos o introducir nuevas características. Las pruebas de regresión son candidatos ideales para la automatización("Types of Software Testing—Complete List", s/f).

Principalmente se manejarán las siguientes pruebas durante la práctica:

- Pruebas de Caja Negra: Realizando pruebas de las aplicaciones sin tener un profundo entendimiento del código, especialmente durante los primeros meses, lo cual ayudará a probar los programas sin esperar resultados ya conocidos.
- Pruebas Funcionales: Generando nuevos casos de uso para características nuevas que se agreguen a las aplicaciones.
- Pruebas de Integración: Analizando todos los componentes existentes y los nuevos para ver sus relaciones y cómo estos se impactan entre ellos.
- Pruebas de Carga: Simulando el uso de múltiples usuarios utilizando las aplicaciones de manera concurrente, se implementa una vez ya se ejecutaron las pruebas de funcionalidad, regresión e integración.

El servicio de software utilizado para crear y ejecutar pruebas de carga es OctoPerf, el cual es una herramienta que simula peticiones enviadas hacia un servidor por medio de usuarios virtuales(*What is OctoPerf? - Documentation—OctoPerf, s/f*).

Finalmente, la versión 2 de SUMADI es una aplicación nativa para Windows, lo que permite más control para la supervisión de los alumnos y elimina la dependencia de tener una extensión en el navegador web para recopilar datos en vivo. Para crear la aplicación nativa se utilizó el marco de trabajo Electronjs.

Electronjs permite la creación de aplicaciones nativas que usan tecnologías para desarrollo web, esto incluye JavaScript, HTML, CSS y cualquier librería basado en estos(*Electron, s/f*).

## IV. DESARROLLO

### 4.1 DESCRIPCIÓN DEL TRABAJO REALIZADO

#### 4.1.1 PLANEAMIENTO DEL ÚLTIMO PERÍODO DE TRABAJO DE 2019

En la Ilustración 8 se pueden observar los diferentes pasos que se realizaron para organizar el trabajo restante del año 2019.



**Ilustración 8 – Agenda Estándar de la reunión en un PI Planning**

Fuente: (PI Planning – Scaled Agile Framework, s/f)

En esta reunión se dedicaron 2 días en el que se discutieron las características que se desean aplicar en el siguiente periodo de trabajo, conocido como Incremento de Programa (PI). Este periodo inició a la mitad de octubre y termina a finales de diciembre del año 2019. Durante esta reunión, todo el equipo de IT definió historias de usuario que se separaban en diferentes áreas para poder trabajar concurrentemente en las diferentes características que se desean aplicar en el ambiente de producción, en aplicaciones, servidores y en la administración.

Inicialmente, la reunión se comenzó con una descripción general de las aplicaciones en las que se están trabajando y también se habló del estado general actual de los usos de las aplicaciones y el deseo de expandir el uso de estas de manera mundial.

Después se inició la fase de reuniones grupales en la cuales los equipos comenzaron a definir historias de usuario para luego obtener asignaciones de estas mismas. Se trabajó con el equipo de arquitectura para discutir los siguientes puntos:

- Soporte a los demás equipos:  
El equipo de arquitectura debe dar soporte a los demás equipos por medio del control de calidad de las aplicaciones, antes de que estas sean desplegadas a un ambiente de producción. También debe hacer despliegues de aplicaciones una vez que estas sean aprobadas y generar roles de seguridad para el acceso necesario a las bases de datos.
- Seguridad de la aplicación SUMADI:  
Se desea incrementar el nivel de seguridad de la aplicación de SUMADI para que este pueda comercializarse mundialmente y minimizar las posibilidades de ataque y sabotaje por entidades externas con intenciones malignas.
- Control de Calidad de SUMADI:  
Se deben generar casos de prueba los cuales luego se utilizarán para ejecutar pruebas extensivas en las nuevas características agregadas a la aplicación SUMADI, también se harán pruebas de regresión e integración para verificar que todo lo anterior funcione sin más problemas.
- Despliegue de la nueva versión de SUMADI en producción:

Se han agendado 3 despliegues de nuevas versiones de SUMADI para este PI, se desean desplegar en las fechas establecidas para actualizar a los usuarios actuales de SUMADI en las fechas prometidas y mantener una buena relación con los clientes y que estos puedan referenciar la aplicación SUMADI a otros clientes potenciales.

- Migración de Base de Datos Cosmos:

Se desea mantener el uso de la base de datos Cosmos, sin embargo, se desea <generar otra instancia de esta base de datos que pueda usarse de anfitrión para diferentes bases de datos a un menor costo que el actual.

- Uso del Punto de Referencia para el Control de Seguridad del Producto (PBSC) para SUMADI:

Blackboard proporciona guías para utilizar en cuanto al control de seguridad, que se deben aplicar con el programa SUMADI ya que es una herramienta interoperable de aprendizaje (LTI) que se utiliza principalmente en Blackboard.

- Manejo de Políticas de Seguridad para SUMADI:

Se desean implementar políticas de seguridad para el equipo de SUMADI para que se lleve un proceso mejor monitoreado y seguro, por medio de mejores prácticas de acceso y seguridad.

- Pruebas de Control de Calidad y de Carga, así como Despliegues de Programas:

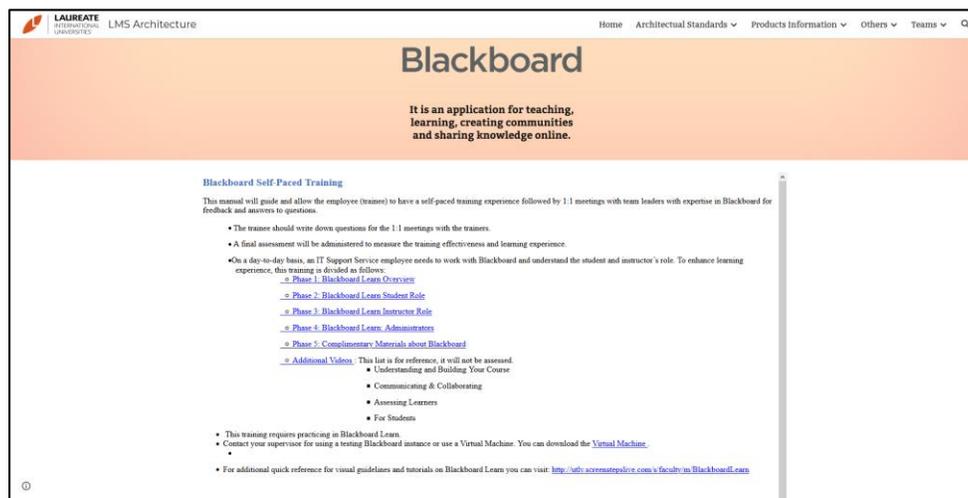
Se deben ejecutar pruebas de calidad al agregarse una característica importante en la aplicación de SUMADI, también verificar que todos los componentes anteriores funcionen correctamente y sin errores agregados. Al encontrarse errores, se debe dar una retroalimentación al equipo desarrollador que trabaja en la característica donde se encontró el mismo, asignar a un desarrollador que se vuelve el responsable de resolverlo y verificar que no se encuentra en el programa después de aplicar un parche que lo resuelve. Una vez que se ha confirmado que no haya errores críticos en la aplicación y se esté preparando para desplegarse en un ambiente de producción, se debe aplicar una prueba de carga para simular el estado de la aplicación en un ambiente comercial real, en el cual luego se evaluará el resultado y se decidirá si lanzar el programa o arreglar errores si estos surgen.

- Escaneos de Seguridad y Aplicaciones de Parches:

Se desea aplicar un agente escáner de vulnerabilidades en los ambientes de producción para mantener un mejor control de seguridad y reparar vulnerabilidades críticas cuando surjan. Se desea generar políticas de escaneo de seguridad y también asignar responsables para que estas políticas sean implementadas correctamente.

#### 4.1.2 BLACKBOARD TRAINING

En la Ilustración 9 se puede observar parte del manual autodidacta para aprender a usar la plataforma de aprendizaje virtual Blackboard.

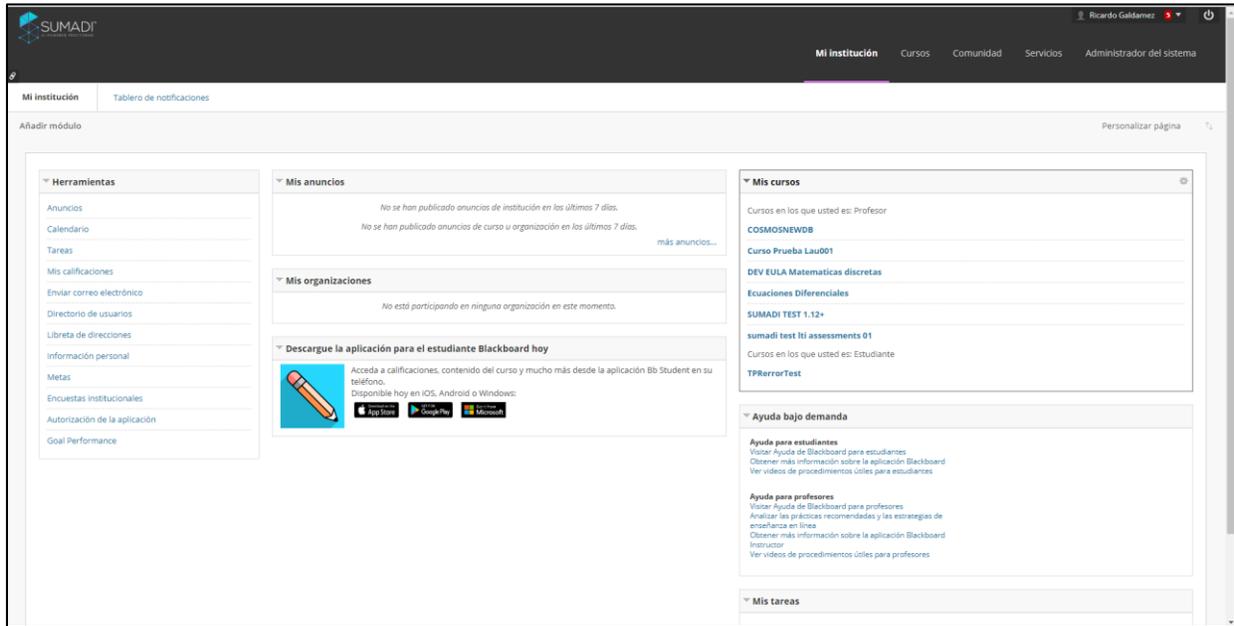


**Ilustración 9 – Entrenamiento para el uso del LMS Blackboard**

Fuente: (LMS Architecture, s/f)

Actualmente el LTI SUMADI está implementado en Blackboard para su uso principal y comercial, por lo cual es necesario la realización de un entrenamiento del Blackboard LMS para comprender la interoperabilidad de Blackboard y SUMADI. Se entrenó el uso de Blackboard simulando la experiencia de un estudiante, de un maestro y finalmente como un administrador de la plataforma.

En la Ilustración 10 se puede observar una instancia de Blackboard que se utiliza para probar las nuevas características de la aplicación SUMADI.

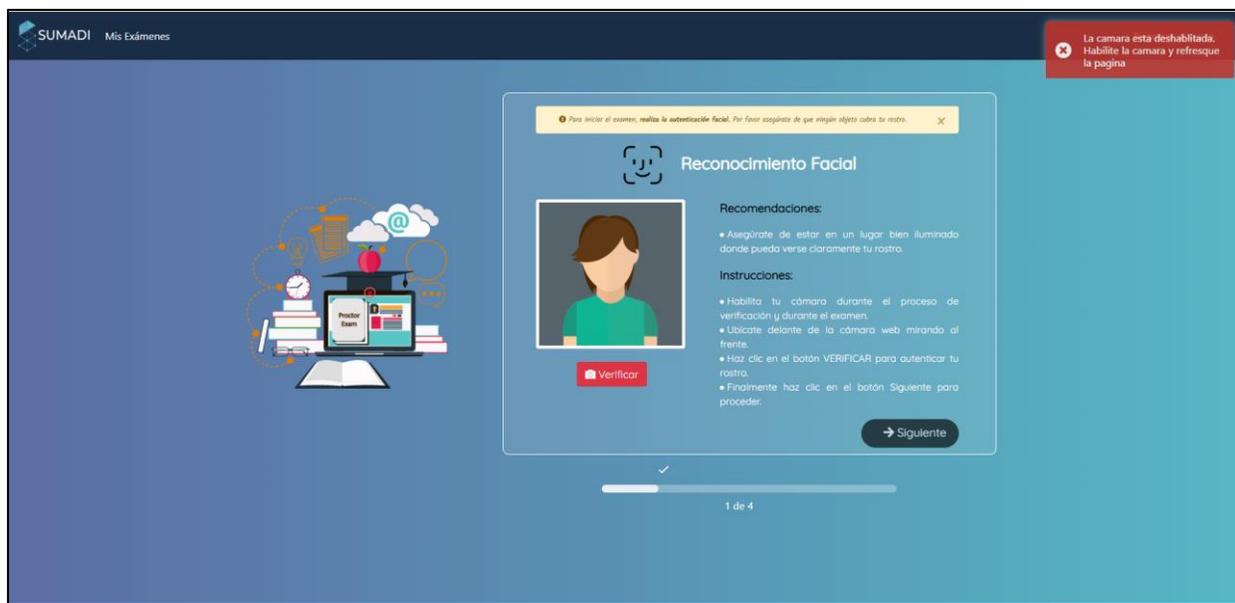


**Ilustración 10 – LMS de Blackboard para prueba de SUMADI**

Fuente: (Blackboard Learn, s/f)

#### 4.1.3 PRUEBAS DE REGRESIÓN DE SUMADI

En la Ilustración 11 se muestra una página de la aplicación de SUMADI.



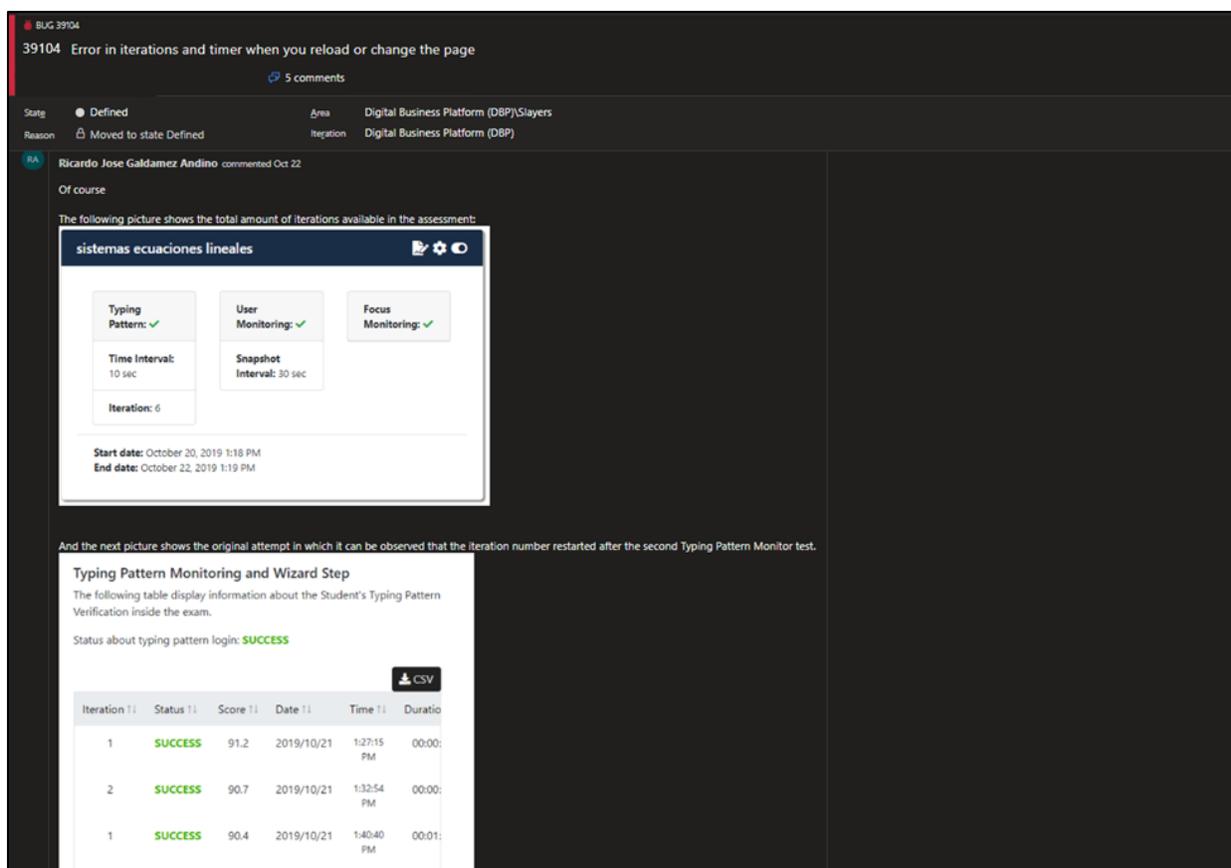
**Ilustración 11 – Aplicación de SUMADI**

Fuente: (Sumadi – AI-Powered Proctoring, s/f)

Se deseaba hacer pruebas de regresión de la aplicación SUMADI, probando componentes que previamente ya habían sido probados, pero se deseaba tener un punto de vista nuevo; es aquí donde el practicante comenzó a ejecutar pruebas de componentes anteriores. Se encontraron algunos errores y luego se documentaron en la plataforma digital de negocios de IT Azure DevOps para luego dar retroalimentación.

#### 4.1.4 PROVEER RETROALIMENTACIÓN DE ERRORES

En la Ilustración 12, se encuentra la típica documentación de un error y su socialización hacia los desarrolladores utilizando la plataforma de Azure.



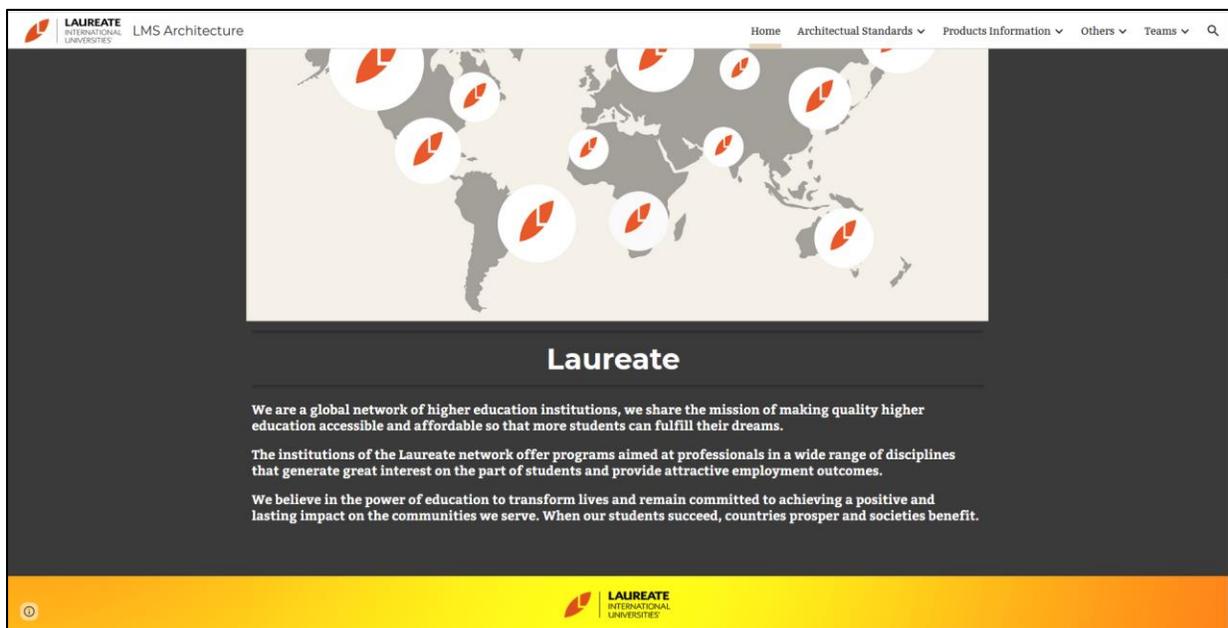
## Ilustración 12 – Documentación de un Error de SUMADI

Fuente: (Azure DevOps Services | Microsoft Azure, s/f)

Después de concluir con las pruebas de regresión, se encontraron errores críticos y no tan críticos. Estos se documentaron y luego se dio retroalimentación al scrum master para explicar a detalle los errores y la replicación de estos para que los desarrolladores puedan hacer una depuración de manera precisa, ya que de esta manera sabrán en qué parte del código de SUMADI se encuentran los problemas.

### 4.1.5 OBTENER INFORMACIÓN DEL WIKI ACTUAL

En la Ilustración 13 se puede observar una página del wiki actualmente utilizado.

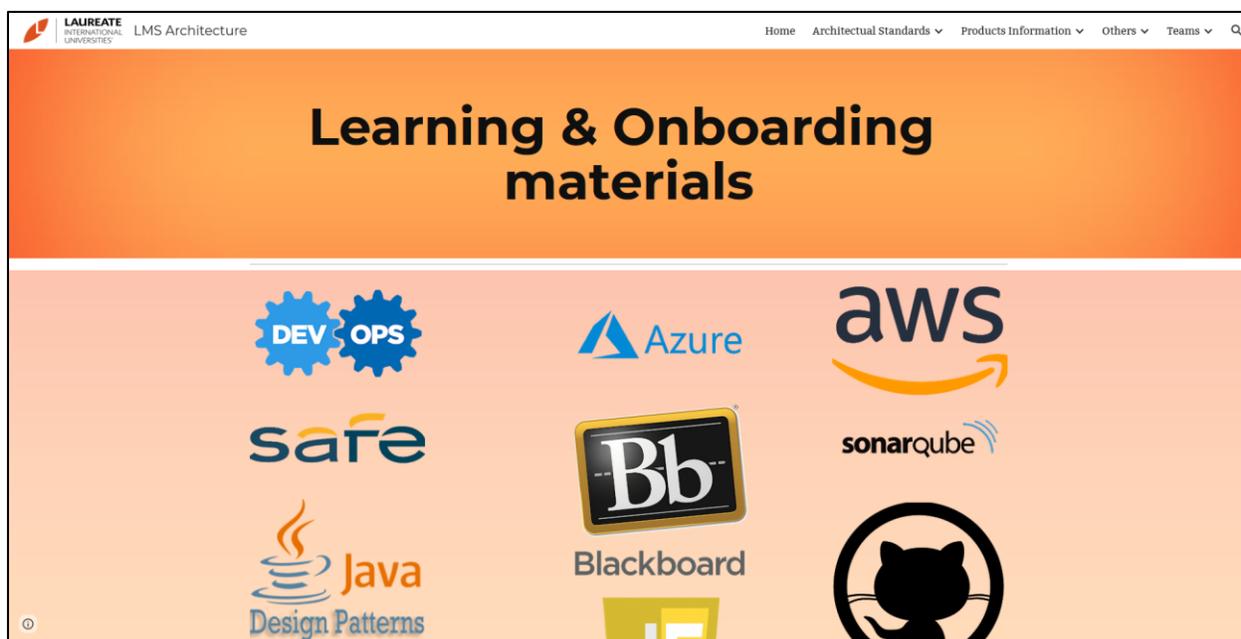


**Ilustración 13 – Wiki en Google Sites de la arquitectura LMS de Laureate**

Fuente: (LMS Architecture, s/f)

Actualmente, la información de la arquitectura LMS de Laureate se encuentra en sites.google.com, y se desea migrar toda esta información a otro sitio que esté diseñado para usarse como wiki. Después de investigar diferentes sitios con capacidad de funcionar como servidores para un wiki, el equipo de arquitectura decidió que Confluence, servicio de Atlassian, era una opción óptima para migrar la información. Una vez migrada la información, se desean actualizar todos los datos obsoletos y dar un mejor nivel de atención y mantenimiento al nuevo wiki.

En la Ilustración 14 se muestra parte de los recursos de aprendizaje que se proveen a los miembros de la arquitectura LMS de Laureate.



**Ilustración 14 – Página de aprendizaje en Wiki de Laureate**

Fuente: (LMS Architecture, s/f)

La página de aprendizaje del wiki se utiliza para explicar las herramientas utilizadas en el ambiente de trabajo de IT. Los practicantes, interinos y trabajadores recién contratados utilizan lo necesario de esta página para familiarizarse con nuevas tecnologías de una manera rápida y a su propio paso para eventualmente trabajar en conjunto con el resto del departamento de IT.

#### 4.1.6 SUBIR INFORMACIÓN DEL WIKI ANTERIOR AL NUEVO

En la Ilustración 15 se muestra la nueva organización de los recursos obtenidos del wiki anterior en el nuevo wiki de Confluence.

All Spaces			Filter
Space	Description	Categories	
IT-LMS Operation - Public		knowledge-bases	ⓘ ☆
LMS Architecture	here is all the information about LMS architecture, for example: Architectural standards, templates, security standards, etc...	knowledge-bases	ⓘ ☆
SUMADI - Admin		documentation	ⓘ ☆
SUMADI - Public Information			ⓘ ☆
SUMADI - Security			ⓘ ☆

**Ilustración 15 – Nuevo Wiki de la arquitectura LMS de Laureate en Confluence**

Fuente: (Confluence, s/f)

Se movió la información del wiki en Google Sites a Confluence, la información principal movida por el practicante fue la página de aprendizaje.

En la Ilustración 16 se muestra la nueva sección de aprendizaje para el grupo LMS de Laureate.

The screenshot shows a Confluence page titled "Learning and Onboarding" in the "LMS Architecture" space. The page is created by Ricardo Jose Galdamez Andino and was last updated on Oct 25, 2019. It features a section titled "Take a look at popular Topics!" with a list of tags: architecturalstandards, code, coding, devops, express, featured, javascript, kb-how-to-article, learning, mongo, node, onboarding, reptilians, slayers, and sumadi. Below this is a search bar and a grid of links to various topics: Amazon Web Services, Azure, BlackBoard, Clean Code, ExpressJS, Git & Github, HTTP, Java Design Patterns, Javascript, MongoDB, NodeJS, NPM, React, REST API, SAFe, SonarQube, Visual Studio Code, and What is DevOps?. At the bottom left, there is a "Like" button with the text "Be the first to like this". At the bottom right, there are navigation links for "learning", "onboarding", "knowledge", and "knowledgecenter".

**Ilustración 16 – Nueva página de aprendizaje de la arquitectura LMS de Laureate**

Fuente: (Confluence, s/f)

#### 4.1.7 ANALIZAR POLÍTICAS DE SEGURIDAD

Al ser SUMADI una aplicación LTI que se ejecuta dentro la plataforma de aprendizaje de Blackboard, esta debe cumplir con una lista de estándares establecida por ellos y conlleva énfasis en seguridad operacional para que la aplicación pueda seguir operando como un servicio. Fue una necesidad crear las políticas de gestión de cambios y las políticas para la gestión de aplicación de parches de seguridad.

Para comenzar a crear las políticas de seguridad, se analizaron las políticas de seguridad de Laureate International Universities y otras empresas para tomarlas como ejemplo en cuanto a aplicación de parches y peticiones de cambio. Se desea generar estándares para mantener un mejor registro de los cambios aplicados a los ambientes de pruebas y producción con el fin de mejorar el control de seguridad y ser capaces de demostrarlo.

#### 4.1.8 CREAR POLÍTICAS DE SEGURIDAD Y SUBIRLAS A CONFLUENCE

Se crearon políticas de seguridad para SUMADI y se utilizó la tipografía y colores de esta para generar documentos más propios de la solución de software. Se crearon diagramas que explican el proceso de gerencia de cambios y el proceso de gerencia de parches.

En la Ilustración 17 se puede ver el índice del documento creado.

# SUMADI

## Change Management Policy

### Contents

Objective.....	2
Scope.....	2
Definitions.....	3
Roles and Responsibilities.....	5
Types of Changes.....	5
Emergency Changes.....	5
Unplanned Changes.....	6
Planned Changes.....	7
Change Documentation.....	7
Change Review and Authorization.....	8
Change Implementation.....	9
Policy Application.....	10
Compliance and Exceptions.....	10
Change Advisory Board.....	10
Request For Change ID Format.....	11
Examples.....	11
Request For Change Ticket Format.....	12
RFC Template.....	12
Ticket Filling Legend.....	13
Change Management Records Table.....	14
Meeting Minutes Form.....	15
Change Management Workflow.....	17

### **Ilustración 17 - Índice de las Políticas para el Manejo de Cambios**

Fuente: (Elaboración propia)

En la Ilustración 18 se puede observar el índice del estándar en el que se trabajó.

# SUMADI

## Patch Management Standard

### Contents

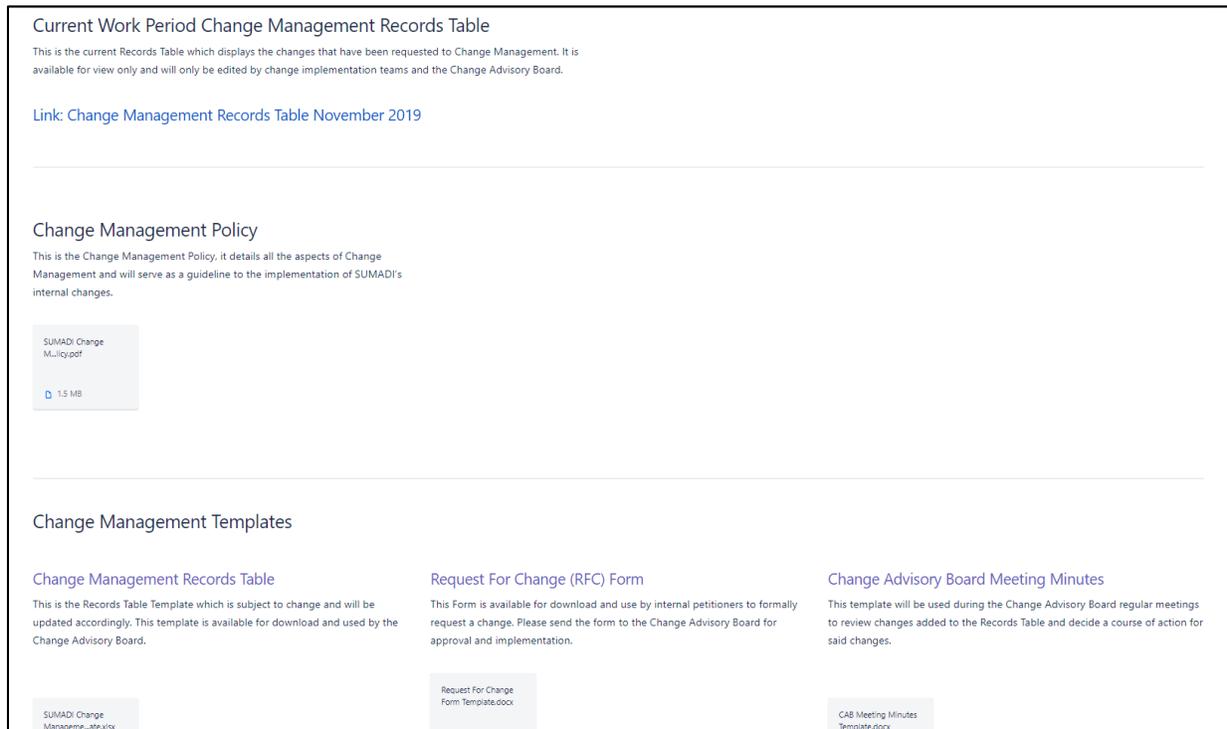
Objective.....	2
Scope.....	2
Definitions.....	3
Severity of Vulnerabilities.....	5
Deployment Schedule for Security Patches.....	6
Patch Implementation Log.....	7
Scan Frequency.....	8
Vulnerability Evaluation.....	8
Deployment Schedule for Operational Patches.....	9
Patch Rollback.....	9
Verification.....	9
Standard Application.....	9
Compliance and Exceptions.....	10
Patch Management Workflow.....	10

### Ilustración 18 – Índice del Estándar para la Gerencia de Parches

Fuente: (Elaboración propia)

Las políticas de seguridad necesitan archivos que serán utilizados para llevar un control de cambios y parches aplicados, por lo cual se crearon plantillas las cuales se deben subir al nuevo wiki.

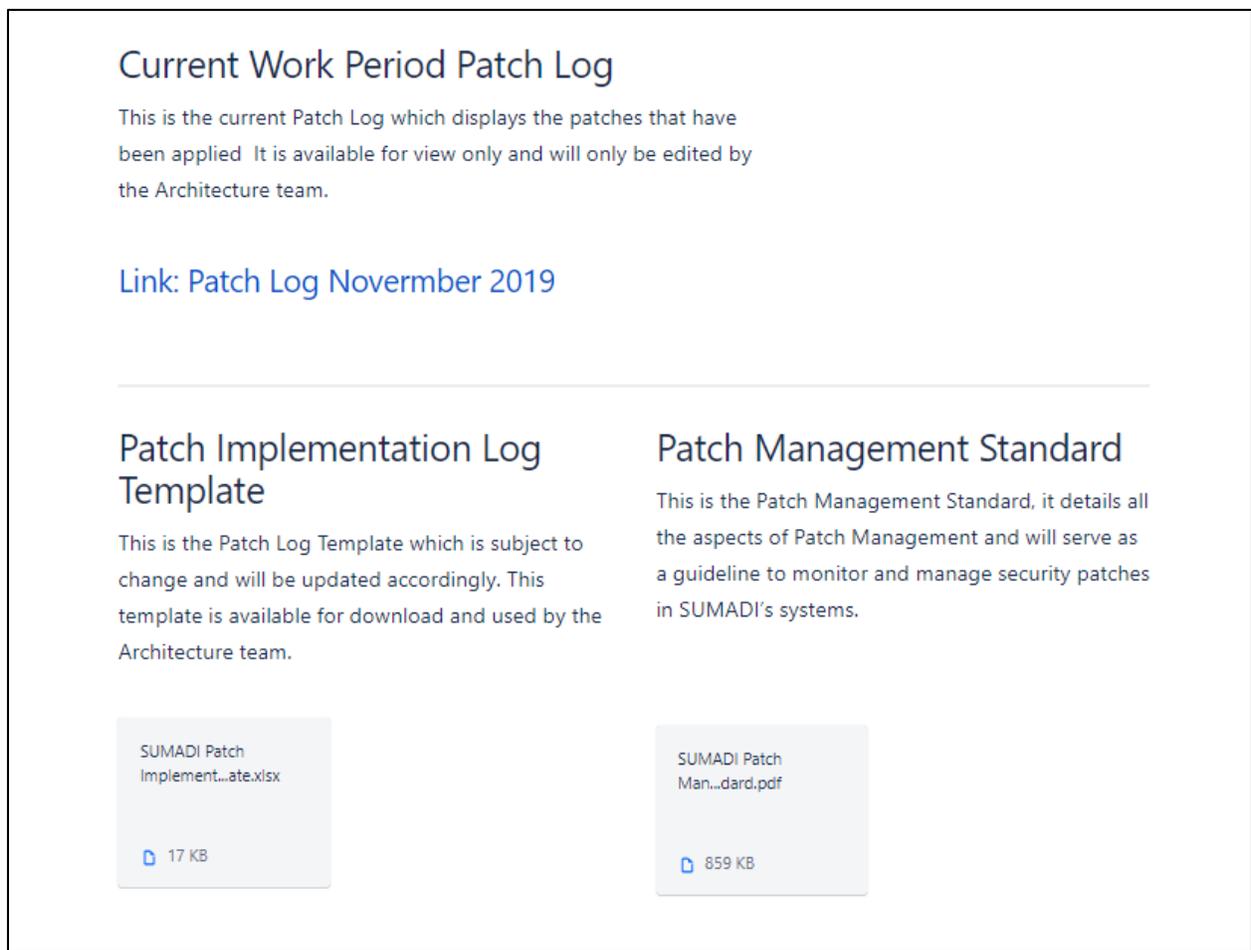
En la Ilustración 19 se puede observar la estructura de la página de Confluence en la cual están las políticas de cambios y las plantillas de estas.



### Ilustración 19 – Página de gerencia de cambio en el Wiki Confluence

Fuente: (Confluence, s/f)

En la Ilustración 20 se puede observar otra sección en el wiki de Confluence que se apartó para gestionar la disponibilidad de los documentos relacionados a la gestión de parches. Se pueden ver subidas las plantillas de las políticas de seguridad, las cuales definen el formato que tomarán los datos que se ingresarán y definen cómo estos datos se guardaran mensualmente en un lugar seguro.



**Ilustración 20 – Página de gerencia de parches en el Wiki Confluence**

Fuente: (Confluence, s/f)

#### 4.1.9 LANZAMIENTOS DE LA APLICACIÓN SUMADI

Se hicieron 3 despliegues con nuevas versiones de la aplicación SUMADI. Estas publicaciones fueron comunicadas al equipo para que este les comunique a los clientes que ya está desplegada una nueva versión de la aplicación que contiene más funcionalidades y parches.

En la Ilustración 21 se puede observar la documentación de los despliegues de la aplicación SUMADI junto con las funcionalidades agregadas y las fechas en las cuales se lanzaron públicamente.

Created by [Ricardo Jose Galdamez Andino](#)  
Last updated less than a minute ago by Ricardo Jose Galdamez Andino

**Date:** 16 Jan 2020

**Version 1.23**

**New Features**

1. **Security:** Added CSRF token that will be generated on each request. Added CORS origin header to verify source of requests.
2. **Management Site:** Created new site to manage SUMADI in a friendly, administrative way, which is also connected directly to the SUMADI app.
3. **Chatbot:** Included a conditional Chatbot embed in the face and typing registration end in the wizard view.

---

**Date:** 19 Nov 2019

**Version 1.19**

**New Features**

1. **User Agreement:** End user must now accept a defined user agreement to be able to use the SUMADI application.
2. **Security:** Session token and request origin set from cookies. HTTP and HTTPS Server created.
3. **Dashboard:** Friendly User Interface to help analyze students' Performance during an assessment evaluation.
4. **AI Tagging:** User emotions being shown in the overall report.
5. **Shortcuts Control:** Added events object in Assessment model to determine keyboard and mouse events availability.

---

**Date:** 23 Oct 2019

**Version 1.12**

**New Features**

1. **Proctor Setting Modularity:** SUMADI now supports enabling/disabling features at an account level per institution & exam.
2. **Typing Pattern Recognition:** A proctor feature that recognizes the typing pattern of a user.
3. **Edusoft Placement Test Support:** Integration with the Edusoft platform for using SUMADI for their placement tests.

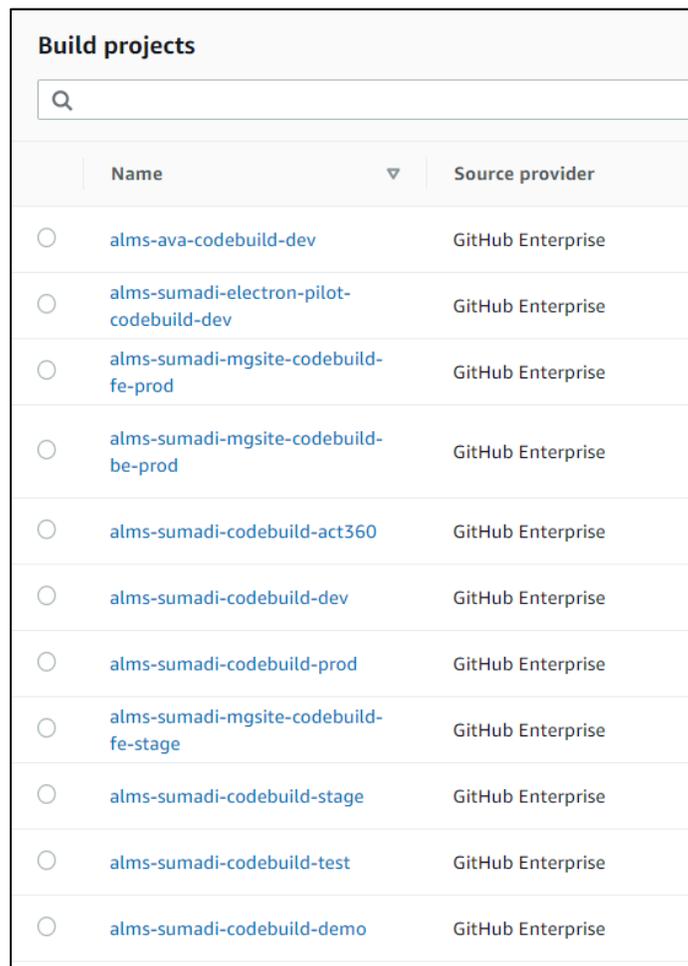
**Ilustración 21 – Documentación de las Funcionalidades Nuevas de SUMADI**

Fuente: (Confluence, s/f)

Para hacer despliegues de la aplicación SUMADI, se han utilizado los Servicios en la Nube de Amazon (AWS). Principalmente se utilizan CodeBuild, CodePipeline y Elastic Beanstalk.

- CodeBuild se utiliza para obtener el código fuente y empaquetarlo para luego desplegar este código a un servidor.
- CodePipeline se encarga de obtener el código empaquetado y desplegarlo a un servidor.
- Elastic Beanstalk maneja directamente los servidores y el escalamiento de estos.

En la Ilustración 22 se pueden observar los codebuilds de Amazon que se utilizan para los distintos ambientes de SUMADI.



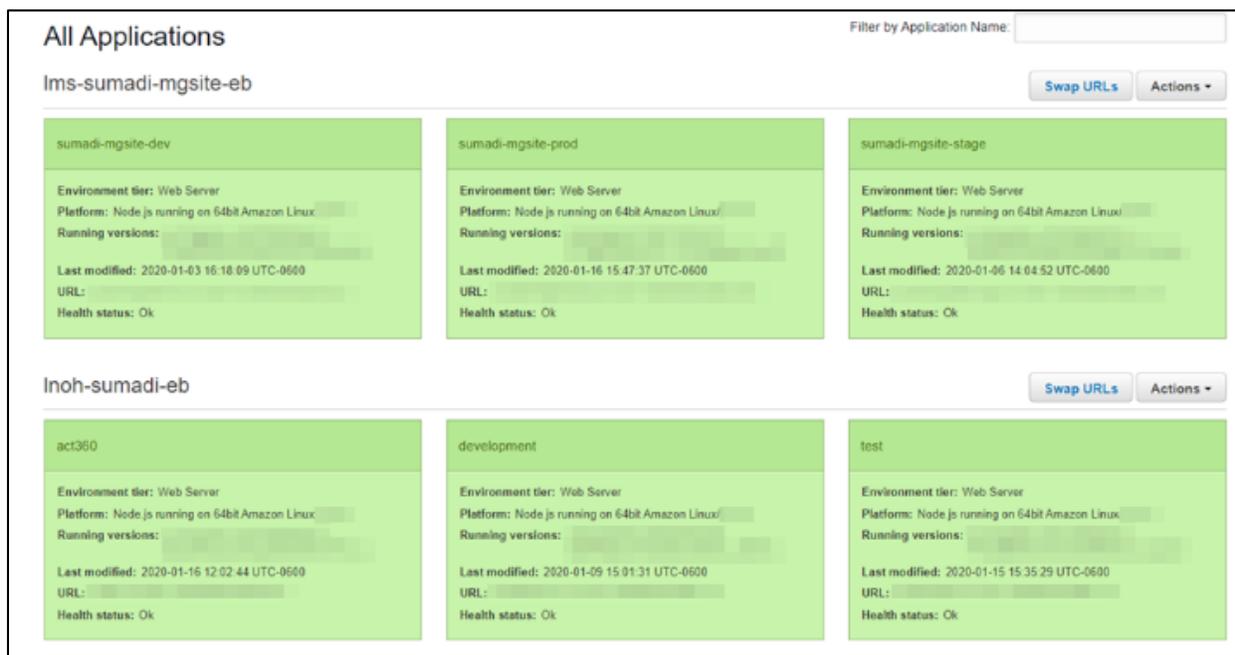
The screenshot shows the 'Build projects' page in the AWS CodeBuild console. It features a search bar at the top and a table listing various build projects. Each row includes a radio button for selection, the project name, and the source provider, which is consistently 'GitHub Enterprise' for all listed projects.

	Name	Source provider
<input type="radio"/>	alms-ava-codebuild-dev	GitHub Enterprise
<input type="radio"/>	alms-sumadi-electron-pilot-codebuild-dev	GitHub Enterprise
<input type="radio"/>	alms-sumadi-mgsite-codebuild-fe-prod	GitHub Enterprise
<input type="radio"/>	alms-sumadi-mgsite-codebuild-be-prod	GitHub Enterprise
<input type="radio"/>	alms-sumadi-codebuild-act360	GitHub Enterprise
<input type="radio"/>	alms-sumadi-codebuild-dev	GitHub Enterprise
<input type="radio"/>	alms-sumadi-codebuild-prod	GitHub Enterprise
<input type="radio"/>	alms-sumadi-mgsite-codebuild-fe-stage	GitHub Enterprise
<input type="radio"/>	alms-sumadi-codebuild-stage	GitHub Enterprise
<input type="radio"/>	alms-sumadi-codebuild-test	GitHub Enterprise
<input type="radio"/>	alms-sumadi-codebuild-demo	GitHub Enterprise

**Ilustración 22 – CodeBuilds utilizados para empaquetar código**

Fuente: (AWS CodeBuild – Servicio de compilación completamente administrado, s/f)

En la Ilustración 23 se muestran los ambientes que están siendo utilizados en Elastic Beanstalk. Estos ambientes crecen automáticamente dependiendo del tráfico que se está recibiendo.



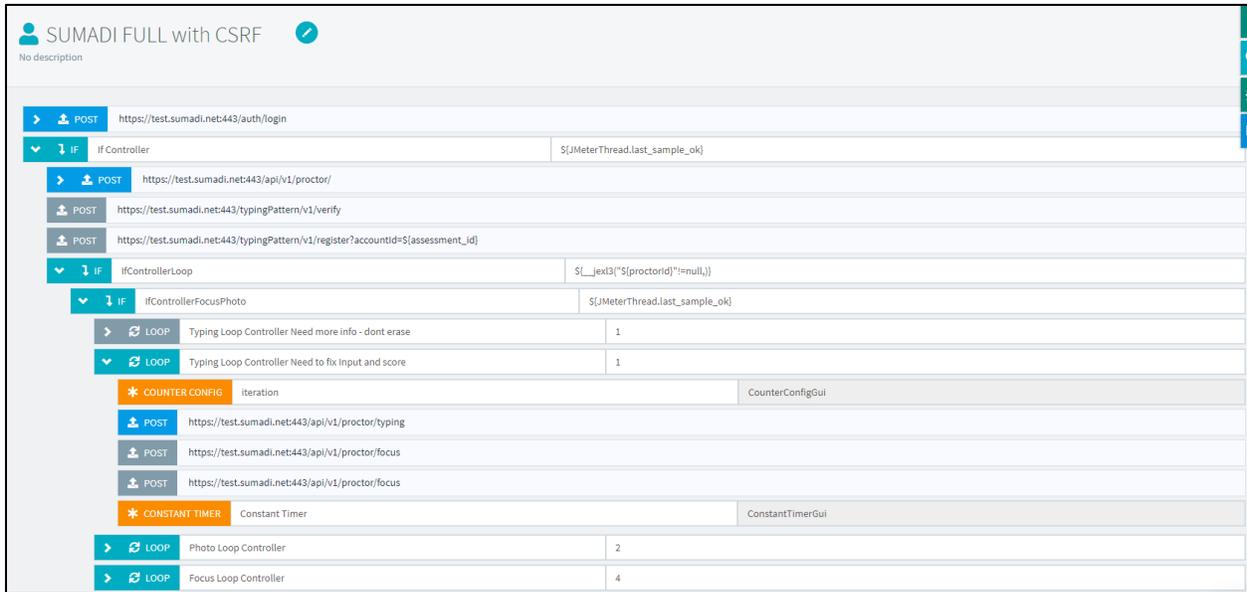
**Ilustración 23 – Ambientes de SUMADI y Management Site en Elastic Beanstalk**

Fuente: (AWS | Elastic beanstalk para aplicaciones web desarrolladas con Java, s/f)

#### 4.1.10 PRUEBAS DE CARGA CON OCTOPERF

Se realizaron pruebas de Carga utilizando OctoPerf para simular el uso de cientos de estudiantes en uno de los servidores de prueba de SUMADI. Realizando este tipo de pruebas se verifica si la aplicación es óptima y no afectará al usuario final si otros usuarios están usando la aplicación de manera concurrente. También se utilizaron las pruebas de carga para generar reportes dentro de la aplicación y verificar que todos los datos se mostraran correctamente.

En la Ilustración 24 se muestra la prueba de carga de SUMADI utilizada para llenar la base de datos con datos de prueba, simulando muchos estudiantes realizando exámenes por medio de dicha plataforma.



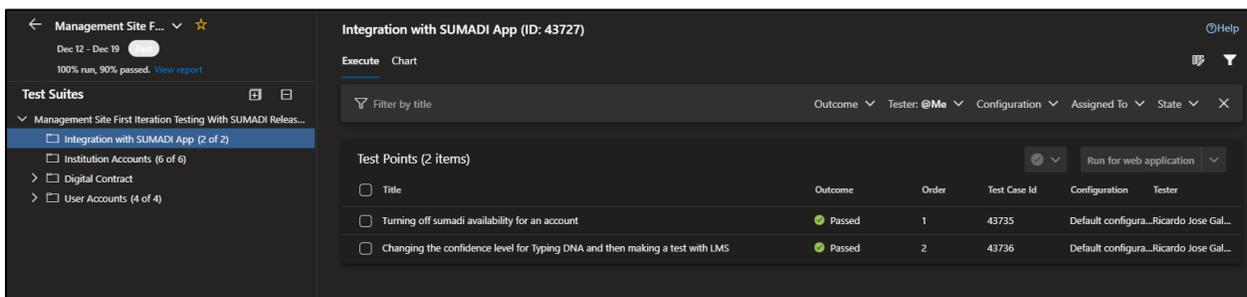
**Ilustración 24 – Pruebas de Carga utilizando Octoperf**

Fuente: (What is OctoPerf? - Documentation—OctoPerf, s/f)

#### 4.1.11 CONTROL DE CALIDAD CON CASOS DE USO

Una de las tareas actuales del equipo de arquitectura es la creación y ejecución de casos de uso para las aplicaciones generadas en el departamento de LMS.

En la Ilustración 25 se muestran algunos casos de uso que se crearon para probar la integración del sitio de Gestión de SUMADI (Management Site) con la aplicación principal.

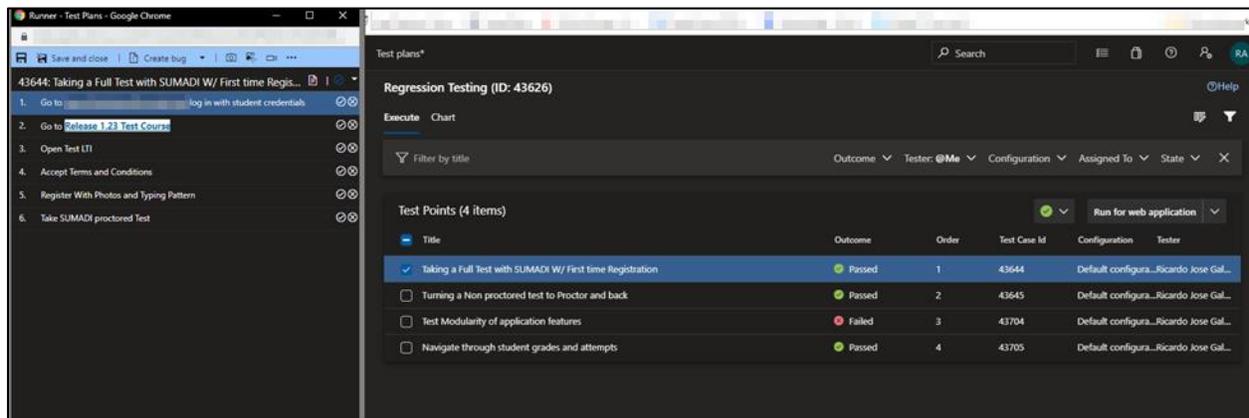


**Ilustración 25 – Casos de uso creados en Azure DevOps**

Fuente: (Azure Test Plans | Microsoft Azure, s/f)

Se deben realizar casos de uso para probar todas las funcionalidades añadidas a SUMADI antes de actualizar una nueva versión en los ambientes de producción. Para esto existe una instancia virtual en la nube que se utiliza para ejecutar los casos de uso creados en Azure DevOps.

En la Ilustración 26 se muestra la ejecución de una prueba de regresión que consiste en simular el uso normal de SUMADI y aparecen los pasos previamente definidos que se deben seguir para completar la prueba.



**Ilustración 26 – Ejecución de un Caso de Uso**

Fuente: (Azure Test Plans | Microsoft Azure, s/f)

#### 4.1.12 SOPORTE Y LANZAMIENTO DE LA APLICACIÓN SUMADI EN ELECTRON

Se adaptó la aplicación SUMADI de manera nativa como aplicación de escritorio, como una prueba piloto que indicará la dirección que tomará el desarrollo de la aplicación en el futuro. Se asignó dar soporte a los desarrolladores en cuanto a buscar opciones para actualizar de manera automática el programa en Electron, también la creación de un constructor de código que empaquetará el código del servidor de SUMADI hacia un servidor en la nube.

En la Ilustración 27 se muestra la documentación en Confluence de cómo actualizar programas en Electron de manera automática. Se investigaron diferentes opciones y se consideró esta como la óptima para las circunstancias dadas.

# How to set Automatic Updates for an Electron Application



Created by Ricardo Jose Galdamez Andino  
Last updated Jan 28, 2020

Here are the steps to apply automatic updates using electron-builder.

## Instructions

1. Use a GitHub Enterprise repository as the main repository where the source code will be located.
2. Use a public GitHub (<https://github.com/>) account where the releases will be deployed. **Do not store any code in the public GitHub Repository.**
3. Set a README or any other file in the **public GitHub** account to be able to use for releases.
4. Get a GitHub Token from the **public GitHub** account with **repo** permissions.
5. Add environment Variable to the computer or server that will be packing and deploying the application and name it "**GH\_TOKEN**" set as value the actual token e.g.  
0bc595df0c6ab5d18d67820506e33b6c646ca0d2
6. Include in the package.json the following packages: (update versions if necessary)

```
1  "dependencies": {  
2    "electron-log": "^4.0.3",  
3    "electron-updater": "^4.2.0"  
4  },  
5  "devDependencies": {  
6    "electron-builder": "^22.2.0",  
7    "electron": "^7.1.9"  
8  },
```

7. In the scripts add the following

## Ilustración 27 – Creación de una Guía para Actualizar programas en Electron

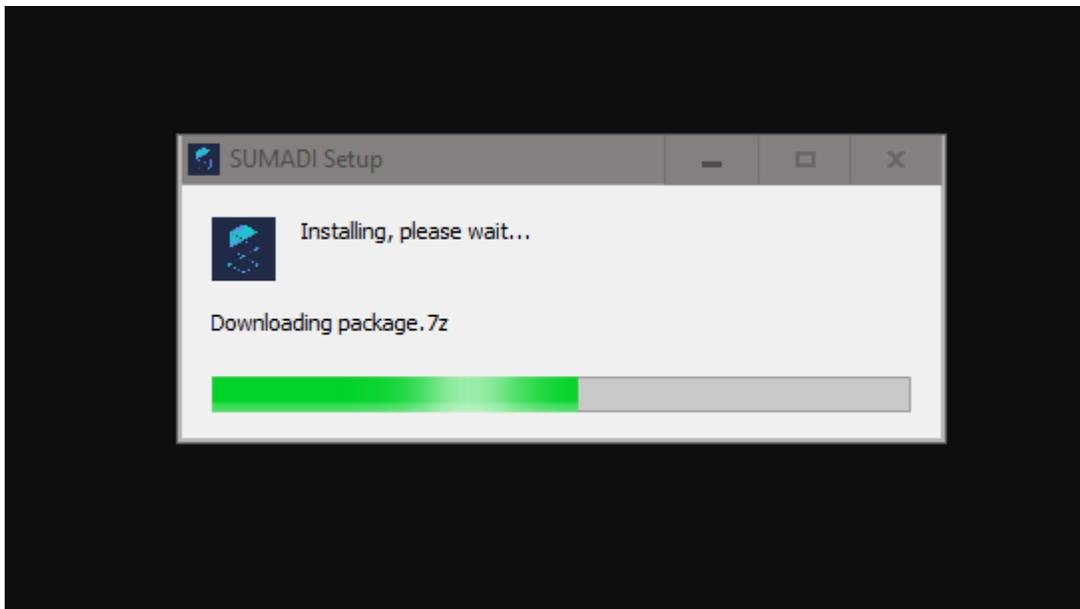
Fuente: (Elaboración Propia)

Fue necesario realizar una nueva investigación porque la forma anterior de lanzar nuevas versiones solo dependía de servidores privados en la nube, mientras que una aplicación que se instala de manera nativa en una computadora requiere otra estrategia para actualizarse después de lanzar una nueva versión al público.

Después de haberse comunicado con el soporte de GitHub Enterprise, se nos notificó que no es posible guardar paquetes en un repositorio de ellos, por lo cual se decidió optar por la siguiente alternativa:

GitHub Enterprise se utilizó para guardar todo el código fuente de la aplicación de Electron y se creó una cuenta de GitHub personal para enviar los paquetes de las siguientes versiones de Electron para que las aplicaciones nativas se puedan actualizar automáticamente.

En la Ilustración 28 se muestra el resultado final de la aplicación nativa de SUMADI utilizando el marco de trabajo Electronjs y actualizaciones automáticas.

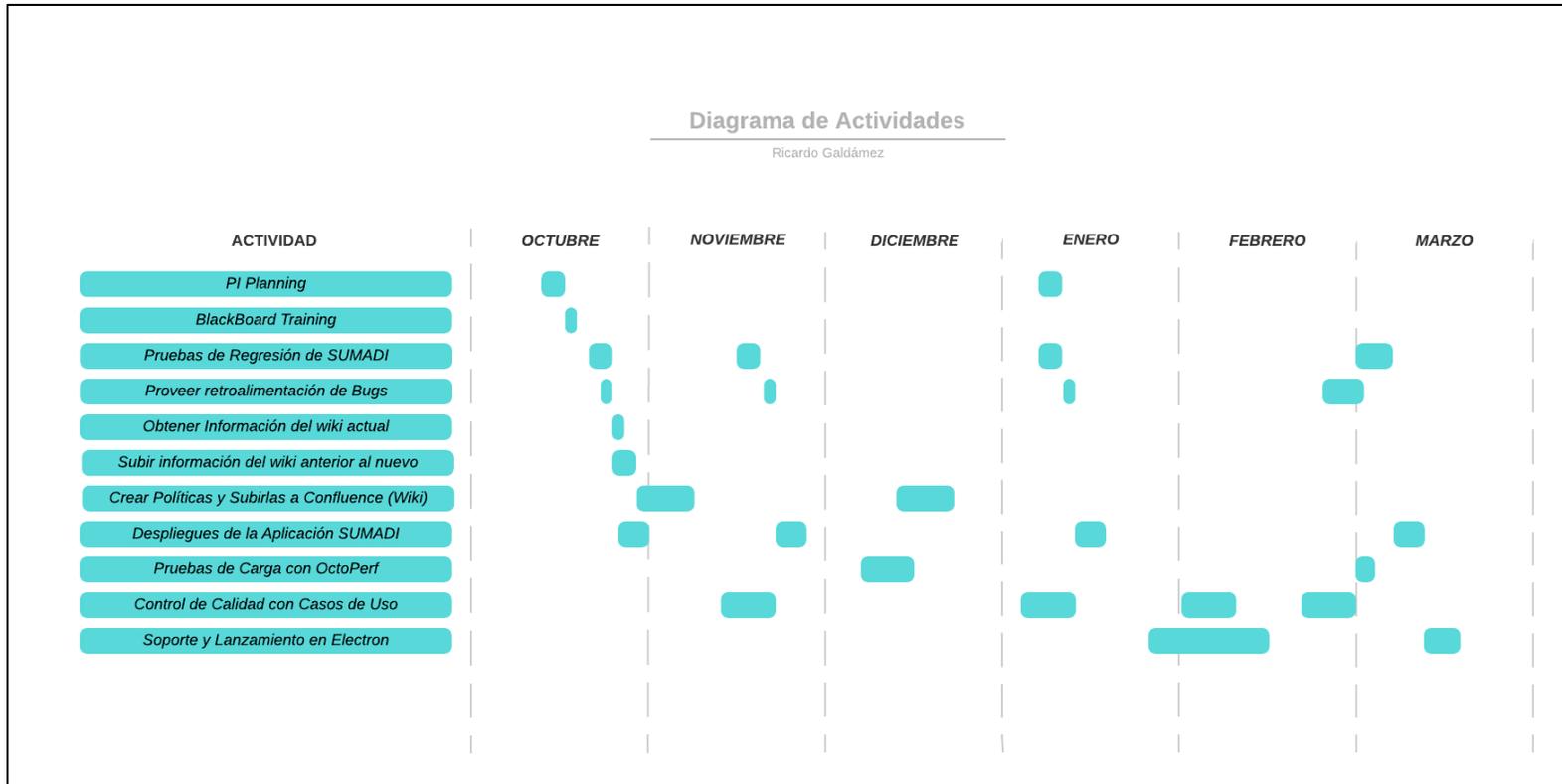


**Ilustración 28 – Aplicación de SUMADI nativa actualizándose automáticamente**

Fuente: (Sumadi – AI-Powered Proctoring, s/f)

## 4.2 CRONOGRAMA DE ACTIVIDADES

En la Ilustración 29 se puede observar el Cronograma de Trabajo utilizado para el desarrollo de las actividades realizadas durante la práctica profesional.



**Ilustración 29 – Diagrama de Gantt**

Fuente: (Elaboración Propia)

## V. CONCLUSIONES

- I. Logrando conseguir auditorías externas de Laureate corporativo y Blackboard para analizar la estructura y seguridad de los servidores mantenidos por LMS, se realizaron diferentes actividades incluyendo la creación de políticas para manejar los cambios en las configuraciones de servidores de producción y la reestructuración de las plataformas en AWS para asegurar el nivel de acceso a los recursos. Esto ayudó a obtener niveles de seguridad más profesionales, aumentar la disponibilidad de los servicios de LMS y asegurar una asociación con Blackboard que seguirá permitiendo la comercialización de SUMADI en su plataforma virtual.
- II. Después de crear políticas de seguridad para lanzar SUMADI a nivel internacional, se iniciaron las operaciones escritas en estas para cumplir con las normas de seguridad y privacidad. Se agendaron varias asignaciones para revisar que las políticas siguen cumpliéndose luego de agregar nuevas funcionalidades al programa de SUMADI. Junto al resto del departamento, se logró llegar al estándar establecido por Blackboard y se puede continuar la operación de SUMADI como una aplicación LTI en sus plataformas.
- III. Se realizaron casos de uso para probar las diferentes funcionalidades de SUMADI en Azure DevOps. Las pruebas de regresión y de aceptación por el usuario se realizaron cada vez que se lanzó una nueva versión de la aplicación, asegurando el funcionamiento normal de la aplicación, así como la calidad de este.
- IV. Se levantaron exitosamente más servidores al querer modularizar la aplicación SUMADI y dejar que clientes realicen su conexión a esta. Se crearon 3 servidores aparte para la aplicación en sí y se lanzaron 3 diferentes versiones del software mencionado junto con la versión de la aplicación Management Site de SUMADI.

## **VI. RECOMENDACIONES**

1. Se deben de socializar las políticas de seguridad más a menudo y dedicar un espacio de tiempo para conseguir certificados SSL para la aplicación del Management Site de SUMADI ya que este maneja SUMADI directamente y no se encripta tráfico sensible sobre la red al usar esta aplicación.
2. Se debe estimar más tiempo y conseguir más personal para realizar las pruebas de calidad ya que consume la capacidad ya establecida del equipo de arquitectura para realizar otras tareas.
3. Se deben migrar los dominios de SUMADI hacia los servicios de Amazon para centralizar los servicios utilizados y reducir costos de la empresa.
4. Se deben documentar correctamente los pasos para hacer un despliegue de la aplicación de SUMADI con el fin de minimizar errores de lanzamiento.

## BIBLIOGRAFÍA

1. Avant.org, & Fortune, S. (s/f). *A Brief History of Databases*. Recuperado el 8 de febrero de 2020, de <http://avant.org/project/history-of-databases/>
2. AWS | *Elastic beanstalk para aplicaciones web desarrolladas con Java*. (s/f). Amazon Web Services, Inc. Recuperado el 16 de enero de 2020, de <https://aws.amazon.com/es/elasticbeanstalk/>
3. AWS | *Elastic compute cloud (EC2) de capacidad modificable en la nube*. (s/f). Amazon Web Services, Inc. Recuperado el 8 de febrero de 2020, de <https://aws.amazon.com/es/ec2/>
4. AWS CodeBuild – *Servicio de compilación completamente administrado*. (s/f). Recuperado el 16 de enero de 2020, de [https://aws.amazon.com/es/codebuild/?nc1=h\\_ls](https://aws.amazon.com/es/codebuild/?nc1=h_ls)
5. AWS CodePipeline | *Integración y entrega continuas*. (s/f). Amazon Web Services, Inc. Recuperado el 16 de enero de 2020, de <https://aws.amazon.com/es/codepipeline/>
6. Ayllón, J. M. (2010). *EL USO DE LAS PLATAFORMAS DE ENSEÑANZA VIRTUAL PARA IMPARTIR ASIGNATURAS JURÍDICAS*. 1, 49–60.
7. Azure DevOps Services | *Microsoft Azure*. (s/f). Recuperado el 1 de diciembre de 2019, de <https://azure.microsoft.com/en-us/services/devops/>
8. Azure Test Plans | *Microsoft Azure*. (s/f). Recuperado el 9 de febrero de 2020, de <https://azure.microsoft.com/en-us/services/devops/test-plans/>
9. *Back-End Web Architecture*. (s/f). Codecademy. Recuperado el 9 de febrero de 2020, de <https://www.codecademy.com/articles/back-end-architecture>
10. *Blackboard Learn*. (s/f). Recuperado el 1 de diciembre de 2019, de <https://blackboard2.sumadi.net/>

11. Boneu, J. (2007). *Plataformas abiertas de e-learning para el soporte de contenidos educativos abiertos*.
12. Clarenc, C., Castro, S., Lenz, C., Moreno, M., & Tosco, N. (2013). *Analizamos 19 plataformas de eLearning: Investigación colaborativa sobre LMS*.
13. *Confluence*. (s/f). Recuperado el 1 de diciembre de 2019, de <https://lmsarchitecture.atlassian.net/wiki>
14. Cunningham, W., & Leuf, B. (2001). *The Wiki Way: Quick Collaboration on the Web*.
15. *Electron | Build cross-platform desktop apps with JavaScript, HTML, and CSS*. (s/f). Recuperado el 5 de marzo de 2020, de <https://www.electronjs.org/>
16. *Express "Hello World" example*. (s/f). Recuperado el 9 de febrero de 2020, de <https://expressjs.com/en/starter/hello-world.html>
17. Foote, K. D. (2018, junio 19). A Brief History of Non-Relational Databases. *DATAVERSITY*. <https://www.dataversity.net/a-brief-history-of-non-relational-databases/>
18. *Getting Started – React*. (s/f). Recuperado el 9 de febrero de 2020, de <https://reactjs.org/docs/getting-started.html>
19. *History of SQL*. (s/f). Recuperado el 8 de febrero de 2020, de [https://docs.oracle.com/cd/B13789\\_01/server.101/b10759/intro001.htm](https://docs.oracle.com/cd/B13789_01/server.101/b10759/intro001.htm)
20. *LMS Architecture*. (s/f). Recuperado el 1 de diciembre de 2019, de <https://sites.google.com/view/laureate-wiki/>
21. Manzanero, J. (2003). *El E-Learning en España Modelos Actuales y Tendencias de actuación* (1a ed.). UNION EUROPEA. <https://glifos.unitec.edu/library/index.php/107564>

22. Mitchell, S. (2019). *Easy Wiki Hosting, Scott Hanselman's blog, and Snagging Screens*.  
<https://docs.microsoft.com/en-us/archive/msdn-magazine/2008/july/easy-wiki-hosting-scott-hanselman-s-blog-and-snagging-screens>
23. *MongoDB History—Javatpoint*. (s/f). [www.javatpoint.com](http://www.javatpoint.com). Recuperado el 9 de febrero de 2020, de <https://www.javatpoint.com/mongodb-history>
24. *Node.js Introduction*. (s/f). Recuperado el 9 de febrero de 2020, de [https://www.w3schools.com/nodejs/nodejs\\_intro.asp](https://www.w3schools.com/nodejs/nodejs_intro.asp)
25. *PI Planning – Scaled Agile Framework*. (s/f). Recuperado el 7 de noviembre de 2019, de [/pi-planning/](#)
26. *Product – Sumadi*. (s/f). Recuperado el 27 de noviembre de 2019, de <https://sumadi.net/product/>
27. *Public benefit corporation*. (s/f). LII / Legal Information Institute. Recuperado el 27 de noviembre de 2019, de [https://www.law.cornell.edu/wex/public\\_benefit\\_corporation](https://www.law.cornell.edu/wex/public_benefit_corporation)
28. *¿Qué es DevOps? Explicación de DevOps | Microsoft Azure*. (s/f). Recuperado el 27 de noviembre de 2019, de <https://azure.microsoft.com/es-es/overview/what-is-devops/>
29. *¿Qué es Scrum?* (s/f). Scrum.Org. Recuperado el 6 de noviembre de 2019, de <https://www.scrum.org/resources/blog/que-es-scrum>
30. *Qué es SCRUM*. (2008, agosto 4). *Proyectos Ágiles*. <https://proyectosagiles.org/que-es-scrum/>
31. *¿Qué es una base de datos clave-valor?* (s/f). Amazon Web Services, Inc. Recuperado el 9 de febrero de 2020, de <https://aws.amazon.com/es/nosql/key-value/>

32. *React – A JavaScript library for building user interfaces.* (s/f). Recuperado el 9 de febrero de 2020, de <https://reactjs.org/>
33. shkale-msft. (s/f). *SQL Graph Database Sample—SQL Server.* Recuperado el 9 de febrero de 2020, de <https://docs.microsoft.com/en-us/sql/relational-databases/graphs/sql-graph-sample>
34. *Sobre Laureate.* (s/f). Laureate. Recuperado el 5 de noviembre de 2019, de <https://www.laureate.net/es/about-laureate/>
35. *Sumadi – AI-Powered Proctoring.* (s/f). Recuperado el 10 de noviembre de 2019, de <https://sumadi.net/>
36. *The Modern Application Stack – Part 1: Introducing The MEAN Stack | MongoDB Blog.* (s/f). MongoDB. Recuperado el 9 de febrero de 2020, de <https://www.mongodb.com/blog/post/the-modern-application-stack-part-1-introducing-the-mean-stack>
37. *Types of Software Testing—Complete List.* (s/f). *Testing Excellence.* Recuperado el 9 de febrero de 2020, de <https://www.testingexcellence.com/types-of-software-testing-complete-list/>
38. West, J., & West, M. (2009). *Using Wikis for Online Collaboration* (1a ed.). Jossey-Bass.
39. *What is a Document Database?* (s/f). MongoDB. Recuperado el 9 de febrero de 2020, de <https://www.mongodb.com/document-databases>
40. *What Is a Front-End Developer?* (s/f). Recuperado el 9 de febrero de 2020, de <https://frontendmasters.com/books/front-end-handbook/2018/what-is-a-FD.html>

41. *What is OctoPerf? - Documentation—OctoPerf.* (s/f). Recuperado el 17 de enero de 2020,  
de <https://doc.octoperf.com/>