UNIVERSIDAD DE LOS ANDES

PROYECTO DE GRADO

Presentado ante la ilustre UNIVERSIDAD DE LOS ANDES como requisito final para obtener el Título de INGENIERO DE SISTEMAS

Sistema Digital para el Servicio de Seguridad y Salud en el Trabajo de la Universidad de los Andes - Venezuela

Por

Br. David De Jesús Cedres Becerra

Tutor: Dr. Jany Marisela Suescum Rodríguez

Marzo 2024

Universidad de Los Andes, Mérida, Venezuela

Resumen

La presente investigación tuvo como propósito desarrollar un sistema digital para el servicio de seguridad y salud en el trabajo de la Universidad de los Andes – Venezuela, generando la creación de este sistema a través de herramientas computacionales tales como: Javascript, Typescript, Node.js, React Native, Google Play Store, Railway la cual permitió realizar pruebas con usuarios para los trabajadores universitarios. El enfoque utilizado fue cuantitativo, el diseño fue no experimental, el tipo de investigación fue proyecto factible, ya que se generó un prototipo testeado. Los fines principales de generar la creación del producto fue descubrir como necesidades principales las preocupaciones de una institución u organización tales como; debe ser el control de riesgos que atentan contra la salud de sus trabajadores y contra sus recursos materiales y financieros. Esto debido a que los accidentes de trabajo y enfermedades profesionales son factores que interfieren en el desarrollo normal de la actividad Universitaria, incidiendo negativamente en su productividad y por consiguiente amenazando su solidez y permanencia en las actividades diarias conllevando además graves implicaciones en el ámbito laboral, familiar y social. En consideración a lo anterior, la administración y la gerencia de todo ámbito académico, deben asumir su responsabilidad en buscar y poner en práctica las medidas necesarias que contribuyan a mantener y

mejorar los niveles de eficiencia en las operaciones de la Universidad de los Andes y brindar a sus trabajadores un medio laboral seguro.

Índice

Introducción

La presente investigación tiene como propósito desarrollar un sistema digital para el servicio de seguridad y salud en el trabajo de la Universidad de los Andes Venezuela que contribuya a planificar el trabajo con fines de bienestar para los trabajadores.

La implementación de un sistema digital para el servicio de seguridad y salud en el trabajo es fundamental en la actualidad, ya que contribuye significativamente a la prevención de riesgos laborales, al bienestar de los trabajadores y a la eficiencia en la gestión de la salud ocupacional. Estos sistemas permiten centralizar y gestionar de manera eficaz toda la información relacionada con la seguridad y salud en el trabajo, facilitando la identificación temprana de riesgos, el seguimiento de protocolos de prevención, y la generación de reportes y análisis para una toma de decisiones informada.

En un entorno donde la tecnología avanza rápidamente, la adopción de sistemas digitales para seguridad y salud en el trabajo se convierte en una herramienta estratégica para las empresas y organizaciones que buscan garantizar un ambiente laboral seguro, saludable y productivo. Por tanto, invertir en esta tecnología no solo es una necesidad imperante en el mundo laboral actual, sino también una muestra de compromiso con el bienestar y la seguridad de las personas que hacen vida en la Universidad de los Andes.

Capítulo I

Aspectos básicos de la investigación

1.1 Planteamiento del Problema

En el contexto de la implementación de un sistema digital para el servicio de seguridad y salud en el trabajo de la Universidad de Los Andes enfrenta una problemática importante en la inspección y evaluación en los puestos de trabajo, esto es debido a que los resultados de las inspecciones pueden ser información inexacta causada por errores humanos durante la observación y captura de datos. Además, siempre existe el riesgo de que las anotaciones se extravíen o deterioren, ya que gran parte del proceso actual depende del manejo de documentos físicos, lo que también los hace susceptibles a modificaciones. Por tal motivo de las inexactitudes de los resultados de las inspecciones, el servicio enfrenta un desafío para planificar las inspecciones de manera eficiente y dado a esto, existen áreas con riesgo elevado de accidentes que podrían no ser observadas con la debida frecuencia.

Dada esta problemática existente, se conoce que hay implicaciones importantes en la seguridad de los trabajadores universitarios debido a que los resultados inexactos de los registros pueden dificultar la identificación y el control de los peligros en los

lugares de trabajo, incluso las áreas de alto riesgo pueden aumentar la cantidad de accidentes. Por tal motivo, los aportes que informen, formen y permitan tomar decisiones son muy requeridas para combatir esta problemática. Particularmente, son necesarios los aplicativos computacionales dedicados a la prevención, alerta y cuidado de los trabajadores.

En consecuencias ante la situación mencionada anteriormente se plantean las siguientes inquietudes:

1. ¿De qué manera Diagnosticar la Problemática en SSST-ULA con relación al almacenamiento de la información?
2. ¿Qué modelo de base de datos es apropiado para la organización, de acuerdo a nuestro enfoque y caso de uso?
3. ¿Como Desarrollar plataformas de administrador e inspector, conformadas por un portal web, una aplicación móvil, y una API REST?
4. ¿De qué manera realizar pruebas reales con personal del SSST-ULA y capacitación sobre uso de la plataforma?

1.2 Justificación

A partir de esta problemática integral, se propone desarrollar un Sistema digital para el servicio de seguridad y Salud en el trabajo de la Universidad de los Andes-Venezuela.

Nivel Práctico: Este sistema digital busca ser la solución ágil para la seguridad y salud en el trabajo de la Universidad de los Andes, aporta beneficios tangibles a nivel práctico, como la mejora en la gestión de riesgos laborales, la optimización de procesos de prevención, el seguimiento eficiente de protocolos de seguridad y salud, y la generación de informes detallados para una toma de decisiones fundamentada. Estos aspectos contribuyen directamente a la reducción de accidentes laborales, enfermedades ocupacionales y al fomento de un ambiente laboral seguro y saludable.

Nivel Metodológico: La investigación persigue Proporcionar seguridad a través de un sistema digital, esto implica la aplicación de metodologías innovadoras y tecnologías avanzadas que permiten una gestión más eficaz y precisa de la seguridad y salud en el trabajo. La implementación de este sistema brinda la oportunidad de utilizar herramientas tecnológicas especializadas para recopilar datos, identificar riesgos emergentes y diseñar estrategias preventivas basadas en evidencia.

Nivel teórico: Desde una perspectiva teórica, el proyecto de investigación sobre el desarrollo del sistema digital para seguridad y salud en el trabajo contribuye al avance del conocimiento a nivel computacional de la carrera ingeniería de sistemas con la evidencia de un problema específico de la realidad.

Nivel Social: En un contexto social, la implementación de un sistema digital para seguridad y salud en el trabajo tiene un impacto significativo en la calidad de vida laboral de los trabajadores, promoviendo su bienestar, su seguridad y su salud. Además, al fomentar una cultura preventiva en las organizaciones, se contribuye a la creación de entornos laborales más seguros, sostenibles y responsables socialmente.

En concreto, el desarrollo de este proyecto busca aplicar los conocimientos adquiridos en la carrera de Ingeniería en Sistemas en el bienestar de la sociedad.

1.3 Objetivos

Objetivo general

Desarrollar un sistema digital para el servicio de seguridad y salud en el trabajo de la Universidad de los Andes – Venezuela.

Objetivos específicos

1. Diagnosticar problemática en SSST-ULA con relación al almacenamiento de información.
2. Crear modelo de base de datos de la organización, de acuerdo con nuestro enfoque y caso de uso.
3. Investigar tecnologías apropiadas y proveedores de servicios necesarios.
4. Desarrollar plataforma de administrador e inspector, conformadas por un portal web, una aplicación móvil, y una API REST.
5. Despliegue de los servicios hacia la nube y Google Play.
6. Realizar pruebas reales con personal del SSST-ULA y capacitación sobre uso de la plataforma.

1.4 Alcance del Trabajo

El software se compondrá de tres componentes fundamentales:

- **Una plataforma móvil para inspectores:** Este sistema permitirá a los inspectores recopilar datos sobre los peligros y riesgos en los puestos de trabajo. Los datos recopilados incluyen información sobre las condiciones físicas del lugar de trabajo, los equipos y herramientas utilizados y los procedimientos de trabajo.

- **Una plataforma web para administradores:** Este sistema permitirá a los administradores planificar las inspecciones, administrar a los usuarios, visualizar los datos recopilados, visualizar los informes creados de manera histórica.

- **Una plataforma web para inspectores:** Este sistema permitirá a los inspectores visualizar las inspecciones que deben realizar, así como permitir la redacción de informes una vez la captura de los datos haya sido realizada.

Capítulo II

Marco Teórico

2. 1 Antecedentes

A continuación, se presenta un conjunto de investigaciones, las cuales sirven de base o punto de partida para la presente investigación. A continuación, se detalla cada una de ellas.

La investigación de Morales Paulina, Vintimilla María (2014). Desarrollaron un modelo de plan de seguridad y salud ocupacional enfocado en reducir los riesgos de los trabajadores en la fábrica Sector "Ladrillosa S.A" ubicada en la ciudad de Azogues. Puesto que esta empresa, pese a que permanece 35 años en el sector no sigue la seguridad dentro de sus actividades diarias.

Tapia Marcelo (2010). Desarrolló un Sistema informático de "Salud Ocupacional" con una tecnología de desarrollo que permitiera la portabilidad de la aplicación a distintas plataformas de operación con el fin del manejo de la información de salud de cada trabajador, prevención de accidentes, análisis riesgos y equipo de protección para cada puesto de trabajo calificado dentro de una escala numerada.

Cely Sánchez Diego Leonardo (2016) Realizó un Diseño informático del sistema de gestión de seguridad y salud en el trabajo para la empresa ITEM JC & CIA S.A.S con

el fin de prestarles a sus trabajadores un lugar seguro y con las condiciones necesarias para la realización de las actividades diarias.

Holguín Báez Dolores Grey (2018) Desarrolló un modelo de sistema de gestión de seguridad y salud para el proyecto "CADCA" con el fin de ofrecer calidad de vida y obtener una mejor productividad en la empresa con sus trabajadores y disminuir riesgos accidentales laborales.

Valenzuela Matuz, Luis Alfonso; Valenzuela Galván, Margarita; León Duarte, Jaime Alfonso. (marzo 2021). Desarrollaron un sistema de gestión de seguridad y salud en una empresa agroindustrial de la región de Hermosillo. La metodología se basó en un diagnóstico, desarrollo, implementación y evaluación de un programa de seguridad y salud para la empresa; Con el objetivo de mejorar el bienestar de los trabajadores y se reducir los accidentes y riesgos laborales de la empresa.

2.2 Bases Teóricas

2.2.1 Sistemas de Plataformas educativas

Las plataformas son sistemas educativos buscan simular las experiencias que se encuentran en un salón de clases, enfocándose en el aprendizaje a distancia. Principalmente se utilizan como complemento de la educación presencial, pero actualmente han llegado incluso a sustituir por completo a ésta última en muchas ocasiones. Ejemplos de algunas plataformas educativas son Google Classroom, Moodle, Blackboard y e-College.

2.2.2 Sistemas de Plataformas Sociales

Es un tipo de sistema de plataforma que permite la interacción a distancia entre individuos de igual o diferente localidad, utilizando como mecanismos los mensajes directos, chats grupales, posts y comentarios, videojuegos en línea, e incluso la compartición de imágenes. Algunas plataformas de este tipo son Facebook, Twitter, Discord, entre otras.

2.2.3 Sistemas de Plataformas de comercio electrónico

Una plataforma de comercio electrónico permite a los usuarios comprar y vender bienes y servicios a través de un computador con acceso a internet sin tener que salir de su hogar y, muchas veces, sin importar las fronteras físicas. Ejemplos de

algunas plataformas de comercio electrónico conocidas con Amazon, eBay, Mercadolibre, Alibaba, entre otras.

2.2.4 Sistemas de Plataformas especializadas

Son plataformas desarrolladas para satisfacer las necesidades especiales de ciertos grupos de usuarios. Estas plataformas buscan satisfacer las necesidades que otras plataformas no pueden, o que sólo satisfacen parcialmente. Por ejemplo, un individuo podría necesitar ofrecer un servicio o producto virtual especial a través de internet, y dicho producto sólo podría ser usado dentro de los límites de la plataforma. Tal es el caso de los libros virtuales de Amazon, que sólo pueden ser leídos a través de su plataforma, o utilizando un sistema desarrollado por ellos, llamado Kindle, el cual permite sincronizar los libros comprados en todos los dispositivos donde se haya instalado, sin embargo, no es posible descargar los libros fuera de dicho sistema.

2.2.5 Sistemas de Plataformas de Seguridad y Salud

Son plataformas desarrolladas con un sistema de gestión de seguridad y salud en el trabajo (SG-SST) es un conjunto de elementos interrelacionados que tienen como objetivo establecer una política y objetivos de seguridad y salud en el trabajo. Este sistema se basa en un enfoque sistémico para gestionar la seguridad y la salud laboral, identificando peligros, evaluando y controlando riesgos, y promoviendo un entorno laboral seguro y saludable para los trabajadores.

2.3 Marco conceptual

2.3.1 JavaScript

Es un lenguaje de programación interpretado, encaminado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza especialmente para añadir interactividad a las páginas web y crear aplicaciones web. A pesar de su nombre similar, no está relacionado con Java. JavaScript es de alto nivel, dinámico e interpretado, y junto con HTML y CSS, es una de las tres tecnologías esenciales en la producción mundial de páginas web. Es compatible con todos los navegadores modernos sin necesidad de plug-ins y posee una API para trabajar con texto, arrays, fechas y expresiones regulares. JavaScript se ejecuta en el ordenador del visitante a la web, lo que evita descargas constantes desde el sitio web. Además, se utiliza en entornos que no se basan en la web, como documentos PDF, navegadores específicos del sitio y widgets de escritorio, y ha ganado popularidad en aplicaciones web gracias a máquinas virtuales de JavaScript más rápidas y plataformas construidas sobre este lenguaje.

2.3.2 Typescript

TypeScript es un lenguaje de programación fuertemente tipado, orientado a objetos y compilado que se basa en JavaScript, proporcionando una sintaxis adicional para tipos estáticos. En esencia, TypeScript extiende JavaScript al permitir a los desarrolladores agregar tipos a sus códigos, lo que facilita la comprensión de los

datos que se manejan en el código y permite detectar errores de tipo durante la compilación en lugar de en tiempo de ejecución. Esto significa que TypeScript verifica si los tipos especificados coinciden antes de ejecutar el código, lo que ayuda a identificar posibles problemas antes de la ejecución real del programa.

2.3.3 Node.js

Node.js es un medio en tiempo de ejecución multiplataforma, de código abierto, que se basa en el lenguaje de programación JavaScript. Este entorno admite ejecutar código JavaScript del lado del servidor de forma asincrónica, con una arquitectura orientada a eventos. Node.js es conocido por su eficiencia en el manejo de solicitudes concurrentes y por su capacidad para realizar operaciones de entrada/salida de datos de manera eficiente. Además, al ser basado en JavaScript, Node.js utiliza programación asincrónica, lo que significa que puede manejar múltiples tareas sin bloquear el hilo principal de ejecución, lo que lo hace muy eficiente en términos de memoria. Node.js es ampliamente utilizado para generar contenido dinámico en páginas web, manipular archivos en el servidor, recopilar datos de formularios y trabajar con bases de datos, entre otras funcionalidades.

2.3.4 Express.js

Es un marco de aplicación web minimalista y flexible para Node.js que proporciona un conjunto robusto de características para aplicaciones web y móviles. Se utiliza para simplificar las API del servidor y facilitar la creación de aplicaciones web y

APIs de manera eficiente. Express.js permite organizar la funcionalidad de una aplicación con middleware y enrutamiento, además de añadir utilidades útiles a los objetos HTTP de Node. Este marco es conocido por su simplicidad, flexibilidad, escalabilidad y una amplia comunidad activa que contribuye a su mejora constante. Express.js es altamente popular en el desarrollo web y se considera el estándar de facto en cuanto a marcos de servidor para Node.js.

2.3.5 React Native

Es un marco de aplicación móvil de código abierto creado por Meta Platforms, Inc. que permite a los desarrolladores construir aplicaciones móviles nativas para iOS y Android utilizando conocimientos de JavaScript. Este marco se basa en React, la biblioteca de JavaScript de Facebook para construir interfaces de usuario, y en lugar de dirigirse al navegador, se enfoca en las plataformas móviles. React Native permite a los desarrolladores escribir aplicaciones móviles que se ven y se sienten verdaderamente "nativas", todo desde JavaScript. Además, React Native facilita el desarrollo simultáneo para Android e iOS, ya que la mayoría del código escrito se puede compartir entre plataformas. Este marco se destaca por renderizar utilizando las API de renderizado estándar de la plataforma anfitriona, lo que lo diferencia de otros enfoques como Cordova o Ionic que utilizan webviews. React Native traduce el marcado a elementos de interfaz de usuario reales, lo que garantiza un rendimiento óptimo y una apariencia nativa en las aplicaciones móviles. Además, React Native ofrece una experiencia de desarrollo amigable,

herramientas de depuración inteligentes y la capacidad de compartir conocimientos y código entre plataformas, lo que lo convierte en una opción popular para empresas como Instagram, AirBnB y Skype en la construcción de sus aplicaciones móviles.

2.3.7 Postgres

PostgreSQL, comúnmente conocido como Postgres, es un sistema de gestión de bases de datos relacional orientado a objetos y de código abierto. Destaca por su capacidad de adaptación, cumplimiento con el estándar SQL y su amplia funcionalidad, flexibilidad y capacidad de escalamiento. PostgreSQL es considerado uno de los motores de base de datos más avanzados en la industria, ofreciendo opciones avanzadas para gestionar bases de datos relacionales de manera eficiente. Este sistema cuenta con una característica llamada control de concurrencia multiversión (MVCC), que permite una gestión eficiente de transacciones, optimizando el rendimiento del sistema y mejorando la experiencia del usuario. PostgreSQL es ampliamente utilizado en sectores como gobierno, industria, educación y comercio debido a su capacidad para gestionar grandes volúmenes de datos de manera efectiva. Su origen se remonta a los años 80 y ha evolucionado continuamente, manteniendo una sólida reputación en el campo de los sistemas de gestión de bases de datos.

2.3.8 Google Play Store

Google Play Store es un servicio de distribución digital operado y desarrollado por Google que sirve como la tienda oficial de aplicaciones para dispositivos certificados que ejecutan el sistema operativo Android y sus derivados, así como ChromeOS. Permite a los usuarios navegar y descargar aplicaciones desarrolladas con el kit de desarrollo de software de Android (SDK) y publicadas en la plataforma. Además de aplicaciones, Google Play Store ofrece juegos, música, libros, películas y programas de televisión para su descarga. Los contenidos adquiridos en Google Play Movies & TV y Google Play Books pueden ser accedidos a través de un navegador web y mediante aplicaciones específicas. La tienda ofrece aplicaciones gratuitas y de pago que pueden ser descargadas directamente en dispositivos Android a través de la aplicación móvil exclusiva Google Play Store o desplegando las aplicaciones utilizando las capacidades de hardware específicas del dispositivo. Google Play Store ha sido el escenario de múltiples problemas relacionados con la seguridad, incluyendo la aprobación y descarga por parte de los usuarios de software malicioso. Fue lanzado el 6 de marzo de 2012, unificando Android Market, Google Music, Google Movies y Google eBookstore bajo una sola marca, marcando un cambio en la estrategia de distribución digital de Google.

2.3.9 Railway

En el contexto de hacer deploy de backends, Railway se refiere a una plataforma de infraestructura que permite a los desarrolladores aprovisionar infraestructura,

desarrollar localmente con esa infraestructura y luego desplegar en la nube de manera eficiente. Esta plataforma ofrece una forma rápida y sencilla de construir, enviar y monitorear aplicaciones, sin necesidad de un Ingeniero de Plataforma. Railway proporciona primitivas de infraestructura con configuraciones predeterminadas para gestionar secretos, construcciones y despliegues, lo que permite a los desarrolladores comenzar rápidamente y centrarse en el desarrollo de su producto. Además, Railway facilita la escalabilidad operativa a medida que los proyectos crecen, automatizando la interconexión de servicios a medida que los proyectos se expanden, lo que incluye la adición de bases de datos como Postgres y Redis, y la coordinación con Entornos y Conjuntos de Cambios. Railway en el contexto de hacer deploy de backends proporciona una solución integral para la gestión eficiente de la infraestructura y el despliegue de aplicaciones en la nube.

2.3.10 Docker

Docker es una plataforma de software que permite la creación, prueba e implementación rápida de aplicaciones a través de contenedores. Estos contenedores son unidades estandarizadas que incluyen todo lo necesario para ejecutar una aplicación, como bibliotecas, herramientas de sistema, código y tiempo de ejecución. En el contexto computacional, Docker actúa como un sistema operativo para contenedores, permitiendo la virtualización del sistema operativo de un servidor. Al utilizar Docker, los desarrolladores pueden implementar y escalar aplicaciones fácilmente en cualquier entorno, garantizando la portabilidad y

confiabilidad del código. Además, Docker colabora con AWS para facilitar la entrega de aplicaciones modernas a la nube y ofrece herramientas como Docker Compose y Docker Desktop para agilizar el flujo de trabajo de desarrollo y despliegue de aplicaciones.

2.3.11 Redis

Docker Redis se refiere a la implementación y ejecución de Redis, una base de datos NoSQL que trabaja con pares clave-valor, dentro de un contenedor Docker. Esto implica utilizar Docker para crear un entorno aislado y portátil donde se puede ejecutar Redis de manera eficiente y escalable. Al contenerizar Redis con Docker, se aprovechan las ventajas de la virtualización y la facilidad de gestión que ofrece Docker, permitiendo a los desarrolladores desplegar y administrar instancias de Redis de forma rápida y sencilla en diferentes entornos. Además, esta práctica facilita la integración de Redis en flujos de trabajo modernos y su uso en aplicaciones que requieren alta velocidad de acceso a la información, como almacenamiento en caché, almacenamiento de sesiones, procesamiento en tiempo real y mensajería.

2.3.12 API REST

Una API REST, es una interfaz de programación de aplicaciones (API) que sigue el estilo arquitectónico REST (Representational State Transfer). Este tipo de API utiliza solicitudes HTTP para realizar funciones estándar de base de datos, como crear,

leer, actualizar y eliminar registros (conocidas como operaciones CRUD). Una API RESTful se basa en la representación de recursos y utiliza métodos HTTP como GET para recuperar un recurso, PUT para cambiar o actualizar un recurso, POST para crear un recurso y DELETE para eliminarlo. Estas API son flexibles y livianas, lo que las hace ideales para integrar aplicaciones y conectar componentes en arquitecturas de microservicios. Para cumplir con los principios REST, una API REST debe seguir seis restricciones arquitectónicas, como el uso de una interfaz uniforme, ser cliente-servidor, ser sin estado, permitir el almacenamiento en caché, ser un sistema en capas y permitir código bajo demanda. Las API REST son ampliamente utilizadas en la web y en aplicaciones en la nube debido a su simplicidad, flexibilidad y eficiencia en la comunicación con los servidores.

2.3.13 Contenedor/Docker

Un contenedor en términos computacionales, específicamente en el contexto de Docker, es una unidad estandarizada que encapsula una aplicación junto con todas las bibliotecas, archivos de configuración, dependencias y otros elementos necesarios para su funcionamiento. Estos contenedores son ejecutables y portátiles, lo que facilita su despliegue en diferentes entornos de manera consistente. A diferencia de las máquinas virtuales, un contenedor Docker no incluye un sistema operativo propio, sino que comparte el núcleo del sistema operativo del host en el que se ejecuta. Docker utiliza tecnologías como cgroups y namespaces para limitar el acceso de un contenedor a los recursos del sistema y garantizar su aislamiento.

Los contenedores Docker se comunican con el exterior a través de la red, habilitando la interacción con otros servicios a través de puertos específicos. En resumen, un contenedor Docker es una forma eficiente y ligera de empaquetar y ejecutar aplicaciones de manera aislada y reproducible en entornos informáticos.

2.3.14 Biblioteca BullMQ

Es una solución avanzada de Node.js para el manejo de trabajos distribuidos y mensajes en Redis a través de colas de trabajo. Esta biblioteca, construida sobre la base de la biblioteca Bull, mejora el rendimiento y la funcionalidad manteniendo un enfoque en la fiabilidad. BullMQ permite la creación de colas de trabajo esenciales para la gestión de tareas en espera de ser procesadas. Los trabajadores, que son procesos o hilos separados, recogen estos trabajos y los ejecutan. Cada trabajo puede contener datos y opciones, y una vez procesado, puede completarse con éxito o fallar debido a errores. BullMQ ofrece soporte integrado para trabajos retrasados y reintentos automáticos en caso de fallo. Además, emite eventos relacionados con los cambios en el ciclo de vida de los trabajos, como completados, fallidos, estancados, entre otros. Esta biblioteca también admite la programación basada en prioridades y cronogramas, lo que la convierte en una herramienta versátil para la gestión de cargas de trabajo en sistemas distribuidos y arquitecturas de microservicios. BullMQ se destaca por su capacidad de escalar horizontalmente al permitir aumentar el número de trabajadores según la carga de trabajo, su compatibilidad con Redis para una comunicación rápida y confiable, y su facilidad

de uso que la convierte en una opción popular para el manejo de colas de trabajo en entornos informáticos.

2.3.15 Frontend

Un Frontend es la parte de un sistema de información o software a la que los usuarios acceden directamente para interactuar con el producto informático. Esta interfaz de usuario permite a los usuarios enviar consultas, solicitudes y recibir datos desde el sistema anfitrión. Por lo general, un frontend puede ser una aplicación de software, una interfaz gráfica de usuario (GUI) o una combinación de ambos, que facilita la interacción y utilización de un sistema de información subyacente. Los programadores frontend, también conocidos como desarrolladores frontend, son profesionales capaces de crear esta parte del software utilizando lenguajes de programación como HTML, JavaScript y CSS. En resumen, el frontend es la capa visible y accesible para los usuarios finales, mientras que el backend se encarga del procesamiento de datos y la lógica de negocio en un sistema de software o aplicación.

2.3.16 Backend

Es la parte de un sistema informático que se encarga de gestionar la funcionalidad interna y los procesos que no son visibles para el usuario final. Esta parte del sistema se encarga de procesar la información, interactuar con bases de datos, realizar cálculos, gestionar la lógica de negocio y responder a las solicitudes

provenientes del Frontend. En el desarrollo de software, el Backend se compone de servidores, bases de datos, aplicaciones y códigos que trabajan en segundo plano para garantizar el funcionamiento adecuado de una aplicación o sitio web. Los desarrolladores Backend utilizan lenguajes de programación como Java, Python, Ruby, entre otros, para construir y mantener esta parte del sistema. En resumen, el Backend es esencial para que una aplicación funcione correctamente al procesar datos, realizar operaciones complejas y proporcionar la lógica necesaria para la interacción con el usuario a través del Frontend.

2.3.17 Portal web

En términos computacionales, es un sitio en línea que actúa como una puerta de entrada única para ofrecer a los usuarios acceso integrado a una variedad de recursos y servicios relacionados con un tema específico. Estos portales recopilan información de diversas fuentes en una interfaz de usuario centralizada, permitiendo a los usuarios encontrar contenido de manera personalizada y acceder a aplicaciones, documentos, foros, servicios de compra electrónica, entre otros. Los portales web están diseñados para satisfacer necesidades específicas de información y suelen ofrecer acceso personalizado basado en roles de usuario asignados. Ejemplos comunes de portales web incluyen portales de pacientes, portales de gobierno, intranets/extranets, portales de gestión de conocimiento, portales de estudiantes y portales de proveedores. Estos portales pueden ser utilizados en una amplia gama de industrias y contextos, brindando a los usuarios

una experiencia personalizada y facilitando el acceso a recursos relevantes de manera eficiente.

Capítulo III

Aspectos metodológicos

En este capítulo se describe cómo resolver el problema planteado y cómo dar respuestas a los objetivos propuestos en este trabajo de investigación. A continuación, se presentarán las fases de investigación y todo lo que conlleva el cumplimiento de cada una de ellas.

3.1 Enfoque de Investigación

La investigación a utilizar será la investigación cuantitativa la cual en un proyecto de seguridad y salud en el trabajo implica el uso de métodos cuantitativos para recopilar, analizar y presentar datos numéricos con el fin de comprender fenómenos relacionados con la seguridad y la salud laboral. En la Universidad de los Andes, se promueve la formación en métodos cuantitativos e investigación aplicada en seguridad, lo que incluye la participación de trabajadores en la identificación de peligros, valoración de riesgos y determinación de controles.

Este enfoque permite abordar aspectos como la percepción de la seguridad y salud laboral, la identificación de factores de riesgo en los puestos de trabajo y el análisis cuantitativo para mejorar las condiciones laborales desde una perspectiva basada en datos numéricos y evidencia científica.

3.2 Diseño de Investigación

El diseño es no experimental, simplemente se observa el comportamiento de las unidades de análisis, es decir, el investigador obtiene la información, pero no altera las condiciones existentes (Arias, 1999).

3.3 Tipo de Investigación

El tipo de investigación es del tipo proyecto factible, este tipo de investigación se caracteriza por ser un estudio de desarrollo para una propuesta de un modelo de sistema operativo para resolver problemas o necesidades de individuos u organizaciones.

3.4 Fuentes de Información

La investigación es fundamentada dentro de la organización. diagnosticada mediante reuniones y entrevistas con la directora del departamento Jany Suescum, así como los inspectores John Gutiérrez y Juan Jaimes. Si bien los desafíos que expresen pueden ser múltiples se decidirá por limitar el trabajo a algo lo suficientemente pequeño como para abordar en poco tiempo y cuyo valor sea significativo: un sistema enfocado en la digitalización y automatización del proceso de inspecciones de puestos de trabajo.

3.5 Población

Trabajadores de la Universidad de los Andes.

3.6 Técnica a utilizar

Para el desarrollo de la solución planteada, se aplicará el método Kanban. Este método es apropiado para un equipo de desarrollo de una sola persona, ya que permite trabajar de forma flexible y adaptarse a las prioridades cambiantes.

En Kanban, el trabajo se representa mediante tarjetas que se mueven a través de un tablero Kanban. El tablero Kanban se divide en columnas que representan las diferentes etapas de flujo de trabajo. Las tarjetas se mueven de una columna a otra a medida que el trabajo avanza. En este proceso, el tablero Kanban se dividirá en las siguientes columnas:

- **Por hacer:** Las tarjetas se colocan en esta columna cuando identifican nuevas tareas o características.
- **En progreso:** Las tarjetas se colocan en esta columna cuando se comienza a trabajar en ellas.
- **Terminado:** Las tarjetas se colocan en esta columna cuando se completan.

El equipo de desarrollo trabajará en las tareas o características según su prioridad. Las tareas o características con mayor prioridad se colocarán en la parte superior del tablero Kanban.

Kanban es una técnica flexible que permite adaptarse a las prioridades cambiantes. Esto es importante para un equipo de desarrollo de una sola persona, ya que las prioridades pueden cambiar rápidamente.

Capítulo IV

Presentación y Análisis de los Resultados

En este capítulo presentaremos los resultados de la investigación realizada. Estos resultados mostrarán detalles de la creación del sistema digital para SSST-ULA.

4.1 Diagnosticar problemática en SSST-ULA con relación al almacenamiento de información

Para cumplir con este objetivo la problemática dentro de la organización fue diagnosticada mediante tres reuniones con la directora del departamento Jany Suescum, así como los inspectores John Gutiérrez y Juan Jaimes. Si bien los desafíos que expresaron eran múltiples se decidió por limitar el trabajo a algo lo suficientemente pequeño como para abordar en poco tiempo y cuyo valor era significativo: un sistema enfocado en la digitalización y automatización del proceso de inspecciones de puestos de trabajo.

El proceso de inspección previo al sistema digital funcionaba de la siguiente manera:

Se decide que inspección realizar próximamente.

El día de la inspección, el inspector debe llevar consigo hacia el sitio de la inspección, un formato en papel que llenará al momento de realizar las observaciones, este formato le indica qué cualidades evaluar en este puesto de trabajo, y le servirá como guía para identificar su progreso y definir de manera positiva o peligrosa cada uno de las condiciones inseguras. Simultáneamente, el inspector usa su teléfono personal para tomar fotos y videos de los aspectos que vea en mal estado, pues las imágenes serán necesarias a la hora de redactar el informe final.

El inspector una vez en su oficina, haciendo uso del formato que ha llenado previamente y que contiene sus anotaciones de la inspección, y con las fotos y evidencias disponibles en su teléfono personal, redacta un informe final, en el cual se escribe un análisis profundo para cada una de las condiciones inseguras encontradas, en este análisis el inspector expone los riesgos asociados, leyes y estándares correspondientes que no se están cumpliendo, entre otros comentarios que dejan en evidencia la gravedad de cada elemento observado y sus condiciones. Para finalizar, el inspector escribe una conclusión, en donde expresa que acciones tomar para mejorar esos riesgos, si el lugar de trabajo se encuentra en condiciones demasiado peligrosas para los trabajadores, la conclusión podría incluir la solicitud de cerrar el establecimiento, de manera temporal, parcial, o absoluta.

El formato ya finalizado es enviado a la directora del departamento y se le comunica al encargado del área inspeccionada cuáles fueron las conclusiones finales, se suele

enviar una copia del informe vía email, en el mejor de los casos, sin embargo, en algunas veces se imprime y lleva en persona.

4.2 Crear un modelo de base de datos de la organización, de acuerdo con nuestro enfoque y caso de uso

Antes de implementar el software en cuestión, se realizó un modelo de entidades junto con todos los atributos que parecieron relevantes para el desarrollo posterior, cada una de las entidades fue implementada como tablas dentro de la base de datos, y cada una de sus características o atributos fueron implementados como columnas dentro de la tabla correspondiente.

Las entidades fueron las siguientes:

- Organización.
- Usuario.
- Área.
- Inspección.
- Observación.
- Reporte.
- Categoría.
- Condición.

- Evidencia.

A continuación, se describen cada uno de esas entidades, y cada uno de sus propiedades, así como las relaciones entre ellas:

4.2.1 Organización

Si bien sólo hará uso de la plataforma el Servicio de Salud de la Universidad de los Andes, se implementó una entidad Organización con el propósito de expandir a otras organizaciones de ser deseado en el futuro.

Atributos:

- ID (Clave primaria).
- Name (nombre de la organización, único).
- Users (one-to-many relationship).
- Areas (one-to-many relationship).

Relaciones:

- Todo usuario dentro de la plataforma pertenece a una organización.
- Una organización tiene un conjunto de áreas asociadas.

4.2.2 Usuario

Representa a cada usuario dentro de la organización, nótese que cada usuario tiene un atributo Role, que representa su posición dentro de la jerarquía de la organización.

Atributos:

- ID (clave primaria)
- Email (texto, campo unico)
- Password (texto)
- Name (texto)

- Role (enum: MANAGER, EMPLOYEE, ADMIN)
- Organization (many-to-one relationship, optional)
- Inspections (one-to-many relationship)
- UpdatedAt (datetime)

Relaciones:

- Un usuario pertenece a una organización.
- Un usuario posee un conjunto de inspecciones (aquellas que debe inspeccionar).

4.2.3 Área

Pretende discretizar los lugares físicos de la Universidad, con el propósito de que siempre que se desee hacer una inspección, haya que especificar primero el Área objetivo, con el paso del tiempo y con la futura recolección de datos, esta entidad nos permitirá realizar estadísticas pues será posible responder preguntas como "¿Cuál es el Área con menor frecuencia de inspecciones?", "¿Qué áreas tienen más de cinco meses sin ser inspeccionadas?", entre otras posibles preguntas.

Atributos

- ID (clave primaria).
- Name (texto, unico).
- Organization (many-to-one relationship).
- Inspections (one-to-many relationship).
- UpdatedAt (datetime).
- Unique composite key on (organizationId, name).

Relaciones

- Un área pertenece a una organización.
- Un área tiene asociadas un conjunto de inspecciones.

4.2.4 Inspección

Representa cada una de las inspecciones a ejecutar.

Atributos:

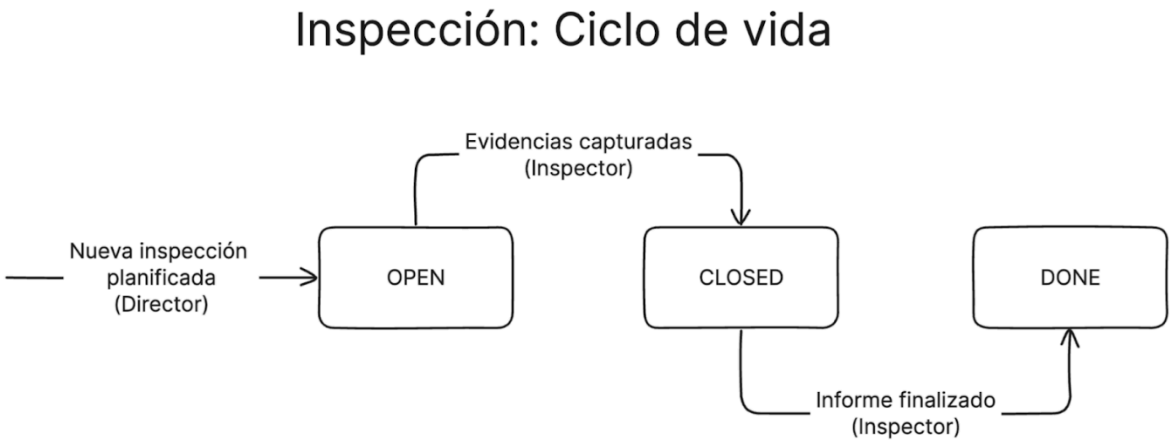
- ID (clave primaria).
- Area (many-to-one relationship).
- Inspector (many-to-one relationship).
- Observations (one-to-many relationship).
- Reports (one-to-many relationship).
- Status (enum: OPEN, CLOSED, DONE).
- Type (enum: ANNOUNCED, UNANNOUNCED).
- Date (datetime).
- UpdatedAt (datetime).

Relaciones:

- Una inspección pertenece a un área (muchas inspecciones pueden tener la misma área).
- Una inspección tiene un inspector (muchas inspecciones pueden tener el mismo inspector).

- Una inspección puede o no tener un reporte (en teoría el esquema soporta varios reportes, pero para efectos prácticos solo existirá un máximo de un reporte por inspección).

Ciclo de vida



4.2.5 Observación

Representa cada una de las observaciones realizadas por el inspector al realizar una inspección de puesto de trabajo.

Atributos:

- ID (clave primaria)
- State (enum: ACCEPTABLE, UNSAFE, MISSING, NEEDS_REPAIR, SKIPPED).
- Inspection (many-to-one relationship).
- Condition (many-to-one relationship).

- Category (many-to-one relationship).
- Evidences (one-to-many relationship).
- Description.
- Analysis (texto, análisis detallado escrito por el inspector a la hora de redactar el informe).
- UpdatedAt (datetime).

Relaciones:

- Una observación pertenece a una inspección (una inspección puede tener muchas observaciones).

- Una observación está relacionada con una condición (una condición puede tener muchas observaciones).
- Una observación está relacionada con una categoría (una categoría puede tener muchas observaciones).
- Una observación tiene asociado un conjunto de evidencias (particularmente necesarias cuando se trata de una observación negativa).

4.2.6 Reporte

Representa el informe que el inspector crea una vez ha realizado todas las observaciones necesarias y ha capturado todas las evidencias requeridas.

Atributos:

- ID (primary key).
- URL (optional).
- Inspection (many-to-one relationship).
- Conclusion.
- CreatedAt (datetime).

4.2.7 Categoría

- ID (primary key).
- Name (unique).
- Conditions (many-to-many relationship).
- Observations (many-to-many relationship).
- UpdatedAt (datetime).

4.2.8 Condición

- ID (primary key).
- Name (unique).
- Observations (many-to-many relationship).
- Categories (many-to-many relationship).
- UpdatedAt (datetime).

4.2.9 Evidencia

- ID (primary key).
- Key.
- URL.
- Observation (many-to-one relationship).
- UpdatedAt (datetime).

4.2.9 Relaciones

- Organización tiene una relación de uno a muchos con Usuario.
- Organización tiene una relación de uno a muchos con Área.
- El usuario tiene una relación de muchos a uno con la organización (opcional).
- Usuario tiene una relación de uno a muchos con Inspección.
- El área tiene una relación de muchos a uno con la organización.
- El área tiene una relación de uno a muchos con Inspección.
- Inspección tiene una relación de muchos a uno con Área.
- Inspección tiene una relación de muchos a uno con Usuario.
- Inspección tiene una relación de uno a muchos con Observación.
- Inspección tiene una relación de uno a muchos con Informe.
- Observación tiene una relación de muchos a uno con Inspección.
- Observación tiene una relación de muchos a uno con Condición.
- Observación tiene una relación de muchos a uno con Categoría.
- Observación tiene una relación de uno a muchos con Evidencia.
- Informe tiene una relación de muchos a uno con Inspección.
- Categoría tiene una relación de muchos a muchos con Condición.

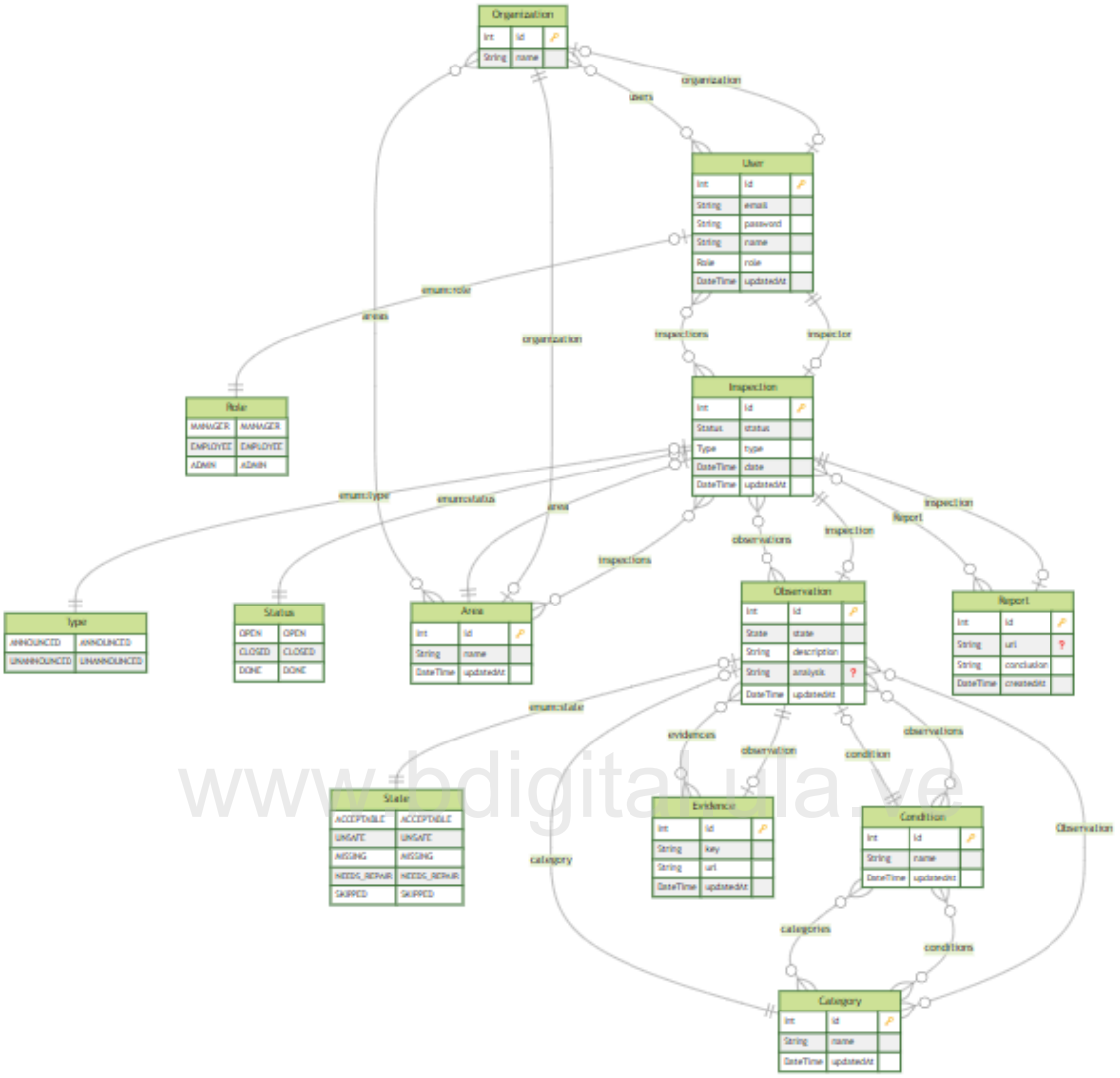
- Categoría tiene una relación de muchos a muchos con Observación.

- Condición tiene una relación de muchos a muchos con Categoría.

- Condición tiene una relación de muchos a muchos con Observación.

- Evidencia tiene una relación de muchos a uno con Observación.

4.2.10 Modelo entidad-relación



4.3 Investigar tecnologías apropiadas y proveedores de servicios necesarios

4.3.1 Elección de la base de datos

Para la solución de persistencia, se limitó a la siguiente pregunta: ¿se estará trabajando con datos altamente relacionados? Como el modelo realizado es altamente relacional tuvo sentido elegir una base de datos de tipo relacional, PostgreSQL que fue una buena opción dentro de esta categoría, cualquier otro gestor de bases de datos relacional también pudo haber funcionado a la perfección.

4.3.2 Elección de tecnologías para el Backend

Para la solución de backend, se eligió NodeJS como runtime para el lenguaje Javascript/Typescript, la biblioteca Express que permitió la creación de una API Restful, así como la biblioteca Prisma, que permite una fácil comunicación con el gestor de base de datos, usando interfaces y clases triviales en Javascript, limitandonos así a no tener que escribir código en SQL que habría sido necesario para comunicarse con PostgreSQL.

Para algunos flujos como la creación de archivos de PDF, se necesitó agregar un sistema de cola, por lo tanto, la biblioteca BullMQ fue utilizada, quien a su vez necesito un servidor de Redis, ambos permiten al sistema la ejecución de tareas de manera asíncrona.

4.3.3 Elección de tecnologías para el Frontend

Se construyeron dos clientes, un portal web, y uno móvil.

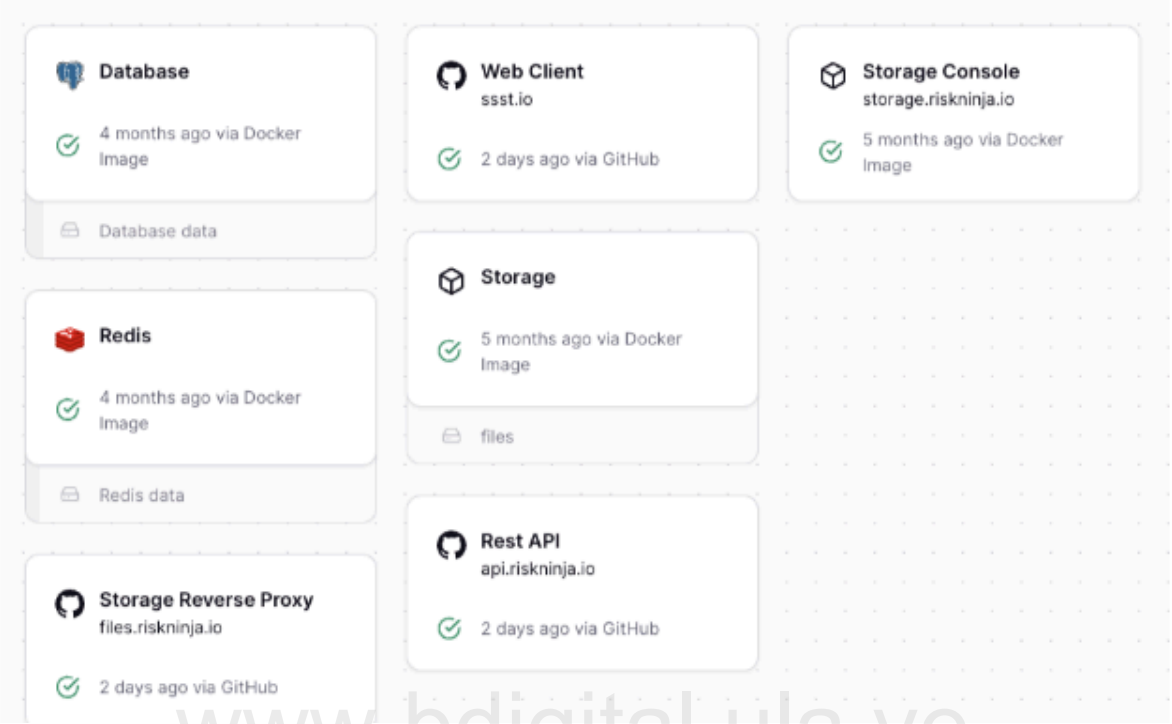
El portal web fue construido usando Typescript como lenguaje, con la ayuda de la biblioteca React para la construcción de las interfaces de usuario, botones, y formularios, El portal para teléfonos fue construido usando Typescript como lenguaje, con la ayuda de la biblioteca React Native para la construcción de las interfaces de usuario, botones y formularios.

4.3.4 Investigación de plataformas para el despliegue de los servicios

Considerando la necesidad los siguientes servicios:

- Una base de datos.
- Un servidor de Redis.
- Un servidor de archivos estáticos (para el almacenamiento de imágenes de evidencias).
- Un servidor que mantenga la API Rest disponible 24/7.
- Un servidor que exponga los archivos estáticos de HTML/Javascript para la página web.
- Dos servicios complementarios para el manejo de archivos estáticos.

Se utilizó Railway como proveedor de infraestructura en línea, a continuación se muestra el diagrama de los servicios almacenados en la nube.



Desarrollar plataforma de administrador e inspector, conformadas por un portal web, una aplicación móvil, y una API REST.

Para lograr este objetivo se utilizaron técnicas de desarrollo de Frontend y backend.

4.1 Diseño y Planificación

Arquitectura

Se utilizó la arquitectura cliente-servidor como punto de partida, facilitando el movimiento de datos dentro de la plataforma, al tener un cliente web, y una aplicación móvil, ambas conectadas al mismo origen de datos, que tuvo la forma de una API Rest.

Funcionalidades

Las funcionalidades de la API Rest fueron aquellas de proteger los datos de usuarios desconocidos, permitir que ambos clientes (web y móvil), puedan acceder a información relevante como inspecciones, usuarios, observaciones, evidencias e

informes, así como permitir la subida de nuevos datos según sea necesario en los flujos correspondientes.

Flujos de usuario como manager

Un usuario que posea el rol "MANAGER" debe ser capaz de realizar las siguientes acciones dentro de la plataforma web y dentro de su organización:

1. Iniciar sesion en el portal web https://ssst.io
2. Agregar nuevos usuarios de tipo "EMPLOYEE", que representan a los inspectores, especificando sus credenciales.
3. Planificar inspecciones, construyendo así una reserva de inspecciones para ser ejecutadas en el futuro por los diferentes inspectores (al momento de la planificación se debe especificar quién es el inspector responsable).
4. Visualizar el estado de todas las inspecciones.
5. Visualizar un listado de todos los informes generados por los inspectores para las diferentes inspecciones, en formato PDF.

Flujos de usuario como inspector

Un usuario que posea el rol "EMPLOYEE" debe ser capaz de realizar las siguientes acciones:

1. Iniciar sesión en la aplicación móvil "Seguridad y Salud ULA" con sus credenciales.

2. Descargar todas las inspecciones de estado “ABIERTO”, que el inspector debe ejecutar en el futuro inmediato.

3. Utilizar el dispositivo móvil como medio de captura de datos, calificando cada condición de seguridad e higiene, como “Aceptable”, “No Seguro”, “No Posee”, “Reparar”.

4. Sincronizar sus observaciones con la nube cuando el dispositivo móvil tenga conexión a internet de nuevo (pudo no haber tenido conexión cuando se capturaron los datos, útil para casos en donde la inspección se realiza en un lugar remoto).

5. Iniciar sesión en el portal web https://ssst.io.

6. Visualizar un resumen de las categorías observadas y sus calificaciones.

Crear un informe detallado, describiendo un análisis detallado para cada una de las condiciones de seguridad de tipo “No Seguro”, “No Posee”, “Reparar” que fueron capturadas en la inspección, finalizando con una conclusión.

7. Visualizar una versión en formato PDF del informe, una vez creado.

4.2 Desarrollo de la API REST

Desarrollo

Se utilizó la biblioteca Express.js para la fácil implementación de controladores y endpoints que fueron capaces de exponer a la red los recursos necesarios para el funcionamiento del sistema, sirviendo así la única fuente de verdad.

Seguridad

Un desafío clave en el desarrollo de cualquier API es no permitir a un usuario acceder a la información de otro usuario dentro del mismo sistema, para prevenir este filtrado de datos, los usuarios deben primero autenticarse utilizando mail y contraseña, obteniendo así un JSON Web Token (JWT) que los clientes deberán presentar cada vez que quieran acceder a datos o alterarlos; de esta manera la API es capaz de identificar que usuario desea acceder a que recurso, y es capaz de rechazar una solicitud de naturaleza fraudulenta.

4.3 Implementación de la plataforma WEB

Desarrollo

La plataforma brinda diferentes funcionalidades dependiendo del rol que posea el usuario que ha iniciado sesión, se habla entonces de una estrategia de control de acceso basado en roles (RBAC):

Un usuario que posea el rol “Manager” dentro de la organización, es capaz de planificar nuevas inspecciones y agregar nuevos inspectores al sistema, sin embargo, un usuario que posea el rol “Inspector” dentro de la organización, es capaz de visualizar las inspecciones que tiene asignadas, y es capaz de redactar informes detallados una vez la inspección y captura de datos haya tomado lugar.

De igual manera la API Rest está protegida bajo la estrategia RBAC y restringe las acciones que los usuarios pueden realizar, basándose en sus roles.

La plataforma web sirve como punto de partida para los flujos que serán ejecutados, pues es el lugar en donde las inspecciones son planificadas, acción necesaria para que los inspectores puedan realizar su trabajo utilizando la aplicación móvil.

4.4 Creación de la aplicación móvil

Desarrollo

Fue desarrollada utilizando Expo, un toolchain que permite el fácil desarrollo y despliegue de la aplicación.

Un requisito particular y gran pilar de la arquitectura e implementación fue la necesidad de que la aplicación funcione incluso cuando no existe conexión a internet, fundamentada en que algunas inspecciones son llevadas a cabo en puestos de trabajo remotos, en donde la señal es nula.

Al utilizar React Native como biblioteca de Javascript se aumenta la compatibilidad pues ambas plataformas móvil y web utilizan React como biblioteca para la construcción de interfaces gráficas, de igual manera se vuelve trivial la reutilizacion de codigo de Javascript/Typescript en caso de ser necesario en futuras iteraciones del producto.

Despliegue de los servicios hacia la nube y Google Play.

Un total de 7 servicios fueron desplegados hacia la infraestructura otorgada por Railway, los servicios fueron los siguientes:

1. Una base de datos PostgreSQL.
2. Un almacén de estructura de datos de valores de clave en memoria Redis.
3. Un servidor web para el funcionamiento del cliente web, propulsado por Caddy.
4. Un servidor de Javascript/Node.js para la ejecución de la API Rest.
5. Un servidor de almacenamiento de archivos estáticos MinIO.
6. Un servicio de proxy inverso utilizando Caddy, para exponer los archivos estáticos de manera segura.
7. Un servicio de visualización y manejo administrativo para MinIO.

Los servicios de Base de datos, y Redis fueron generados y administrados vía interfaz gráfica en Railway, sin embargo, el resto de los servicios fueron desplegados haciendo uso de contenerización, pues Railway es compatible con esta

tecnología, de esta manera se reducen los problemas de poca compatibilidad entre el entorno de desarrollo local y el de producción.

Todos los servicios, a nivel de implementación y detalles técnicos se encuentran ubicados en el mismo repositorio de código (git), usando una organización de archivos conocida como monorepo, lo cual permite tener cierta independencia entre los servicios, pero manteniéndolos en un único repositorio, que está a su vez almacenado en la nube Github.

Railway fue configurado para escuchar cambios en el repositorio en Github, por lo tanto, se habla ahora de un enfoque de integración continua y distribución continua (CI/CD).

Las aplicaciones móviles son intrínsecamente diferentes de los otros servicios ya mencionados, pues termina siendo el dispositivo del usuario quien suministra los recursos necesarios para la ejecución, se reemplaza entonces el despliegue por un proceso de distribución de archivos binarios hacia las tiendas de aplicaciones móviles, en nuestro caso Google Play Store; se utilizó la plataforma de Expo para la generación de archivos APK/AAB de manera automática (CI), sin embargo la distribución de esos archivos hacia Google Play Store es un trabajo que sigue en progreso, por lo tanto sería incorrecto decir que se ha logrado la distribución continua (CD)en el caso particular de la aplicación móvil.

Para sobrellevar la limitante de la distribución de la aplicación móvil, los archivos APK se han instalado en los dispositivos móviles de miembros de SSST ULA de

manera manual, lo cual fue suficiente para la validación del producto mínimo viable desarrollado, pero que limita la posibilidad de desplegar posibles mejoras y actualizaciones de manera sencilla, motivo por el que la distribución continua con Google Play Store es un objetivo que aún debe ser ejecutado.

Realizar pruebas reales con personal del SSST-ULA y capacitación sobre uso de la plataforma

El 13 de marzo, en la facultad de Odontología de la Universidad de Los Andes, a las 9:00 AM, se realizó una prueba piloto en la Clínica de Odontología Operatoria, y se utilizó la plataforma por primera vez en un ambiente real.

- El dispositivo móvil utilizado fue cargado dos horas antes con la última versión de la aplicación, se procedió a iniciar sesión con las credenciales del inspector John Gutierrez.

- El dispositivo móvil no tuvo acceso a internet durante la totalidad del tiempo que duró la inspección, validando así, el objetivo de que funciona en lugares remotos en donde no hay acceso a internet.

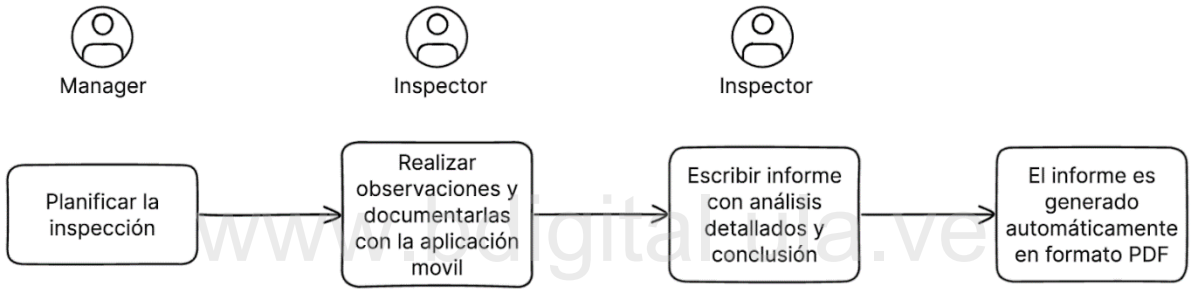
- Los inspectores Juan Jaimes y John Gutierrez utilizaron la herramienta por primera vez en un ambiente real, siendo la aplicación quien guiase qué aspectos observar en el espacio de trabajo, y no permitiendo finalizar la inspección hasta que todos los aspectos fuesen calificados (u omitidos de manera explícita).

- Al finalizar la inspección, se procedió a sincronizar los datos con la nube, cuando se tuvo de nuevo acceso a internet.

Manual de Usuario

En esta sección se desarrolló un manual de uso de la plataforma para Manager, inspectores y usuarios regulares.

El siguiente diagrama ilustra la secuencia de pasos necesarios para el uso correcto de la plataforma, se necesitan de dos actores (un manager y un inspector), así como el uso de la plataforma web y móvil para completar las inspecciones.



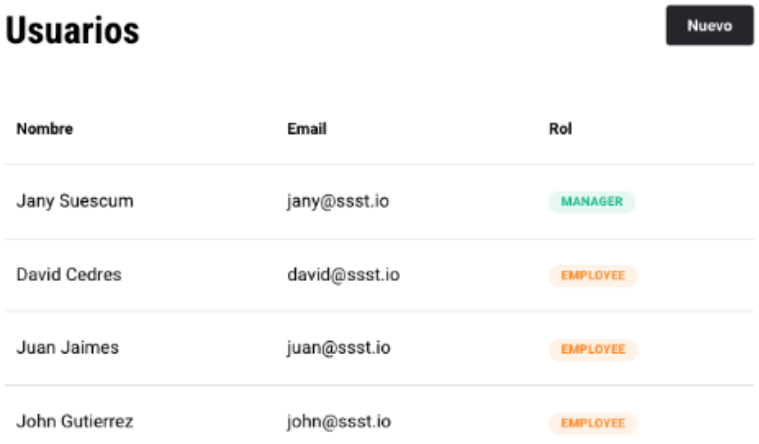
¿Cómo visualizar y agregar nuevos usuarios a la organización?

La organización soporta dos tipos de usuarios: Manager e Inspector.

El usuario de tipo Manager, puede iniciar sesión en la plataforma web https://ssst.io, y encontrará en el lado izquierdo de la pantalla, en el panel de navegación, un elemento que le permitirá navegar hacia la pantalla de manejo de usuarios.

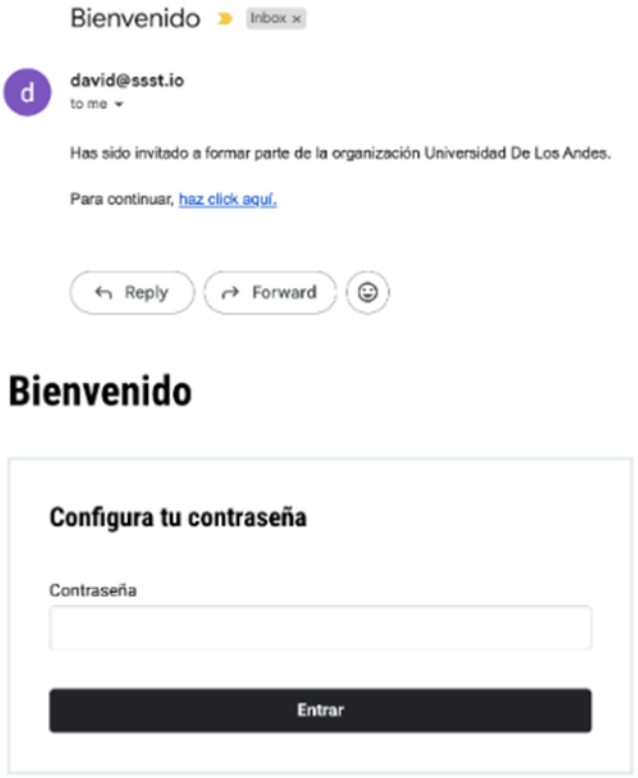
El usuario de tipo Manager puede entonces:

Visualizar los usuarios que conforman la organización.



Agregar un nuevo usuario a la organización, especificando su nombre y correo electrónico.

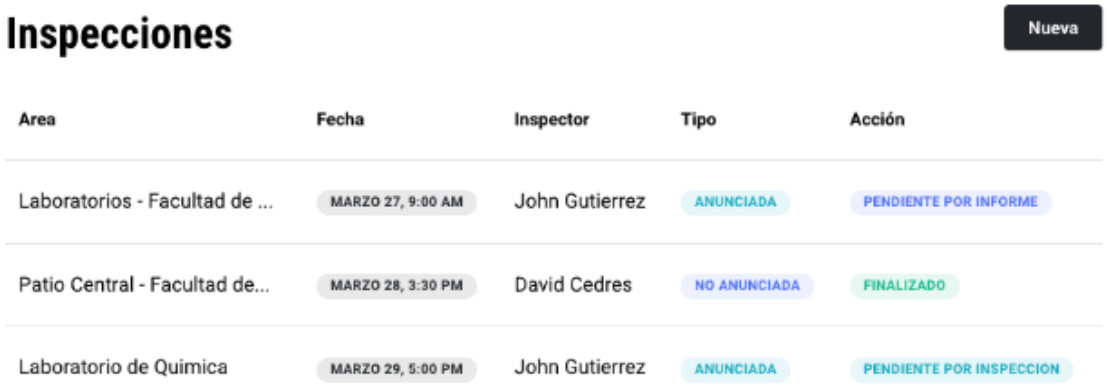
El nuevo inspector recibirá un correo electrónico, en donde podrá continuar el proceso y configurar su contraseña.



¿Cómo planificar nuevas inspecciones?

El usuario de tipo Manager, una vez inicie sesion en https://ssst.io sera presentado con la pantalla de inspecciones, en donde podrá:

Visualizar todas las inspecciones en el sistema.



Planificar una nueva inspección.

¿Cómo documentar una inspección con la aplicación móvil?

El usuario de tipo Inspector, puede iniciar sesión en la aplicación móvil, y será presentado con la pantalla de inspecciones.

En esta pantalla el inspector podrá visualizar las inspecciones asignadas, cada una representada por una tarjeta que identifica el nombre del área a inspeccionar, la fecha y hora de la inspección, así como una etiqueta que expresa si es una inspección anunciada o no anunciada.

El inspector puede tocar la tarjeta correspondiente, lo cual lo llevará a la pantalla de Inspección:

Esta pantalla permite al inspector visualizar una lista de categorías que deberá inspeccionar en el área, así como un contador de condiciones que ha o no observado en cada una de las categorías.

En la medida de que el inspector toque cada una de las categorías, se le presentará la pantalla de Categoría.

En esta pantalla el inspector podrá omitir la categoría y todas sus posibles condiciones si considera que no tiene sentido para la inspección en cuestión, esto es útil si por ejemplo se está inspeccionando una cancha de atletismo sin techo.

El inspector puede tocar cada una de las condiciones a observar, al hacerlo se le presentará la pantalla de Evaluación de Condición.

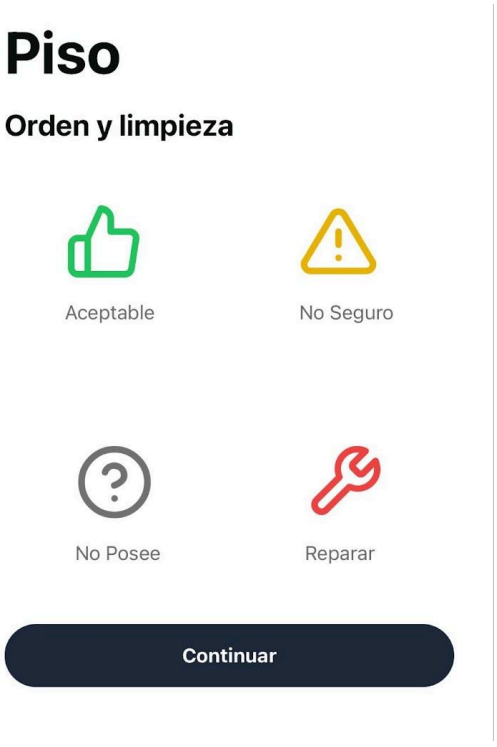
En esta pantalla el inspector puede calificar la condición en alguna de las siguientes opciones:

1. **Aceptable**: la condición está en buen estado y no representa un riesgo para la salud de los trabajadores.

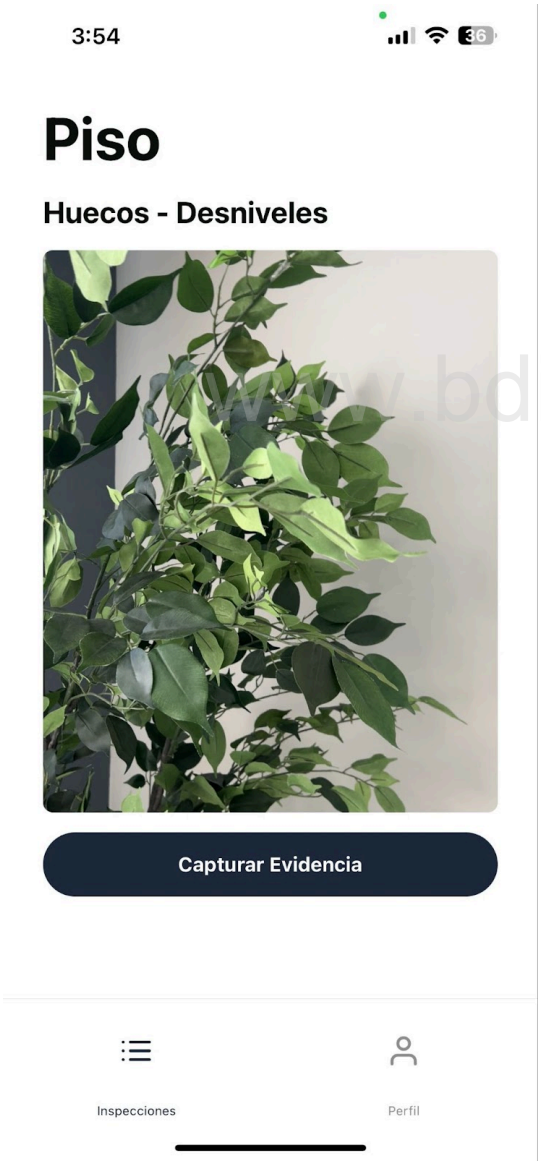
2. **No Seguro**: la condición no está en condiciones óptimas y representa un riesgo laboral; por ejemplo, un techo notablemente agrietado.

3. **No Posee**: la condición no está presente, por ejemplo, debería haber un filtro de agua, pero no hay filtro alguno.

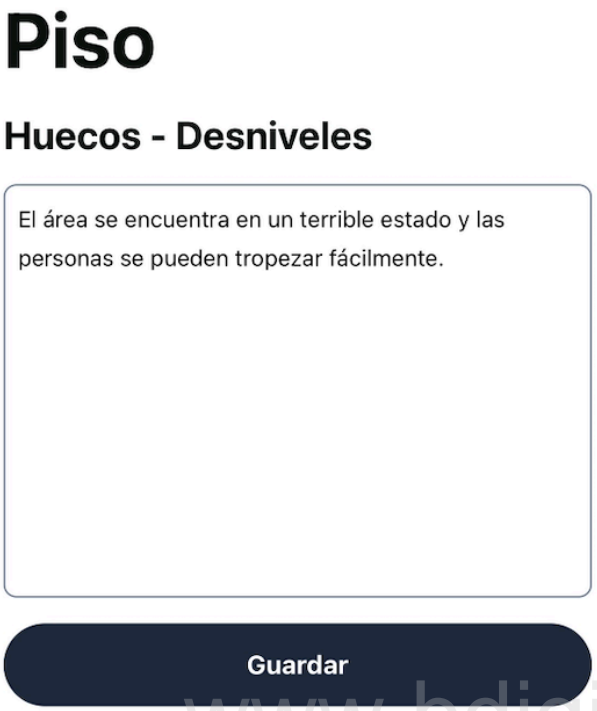
4. **Reparar**: la condición está presente, pero en mal estado y debe ser reparada.



Una vez el inspector califique la condición, pueden pasar dos cosas: si la condición fue **Aceptable**, entonces se le presentará de nuevo la pantalla de **Categoría** para que pueda continuar calificando el resto de condiciones; sin embargo, si el inspector calificar la condición como negativa (no seguro, no posee, o reparar), se le presentará la pantalla de **Captura de evidencia**, en donde usando la cámara del dispositivo podrá tomar una fotografía que confirme su calificación.



Una vez el inspector ha capturado la evidencia, se le pedirá que escriba una anotación corta, que más adelante será la base para escribir un análisis completo.



Una vez el inspector ha escrito la anotación, será presentado de nuevo con la pantalla de **Categoría**, en donde evaluarán las condiciones restantes.

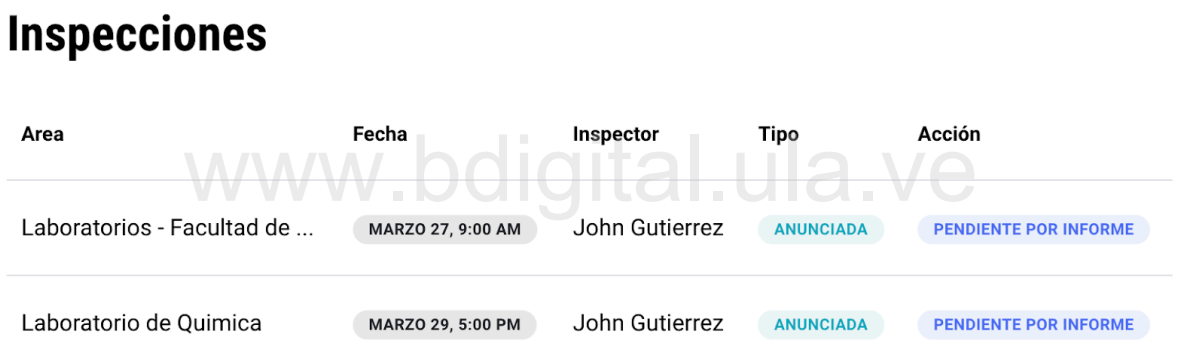
Una vez el inspector ha terminado de evaluar todas las condiciones en todas las categorías aparecerá un botón que permite terminar la inspección.

Una vez el inspector ha terminado la inspección, los datos permanecen almacenados en el dispositivo, y cuando el dispositivo tenga conexión a internet se podrá iniciar la subida de datos hacia la nube, desbloqueando así el siguiente paso en el proceso: Redactar el informe correspondiente.

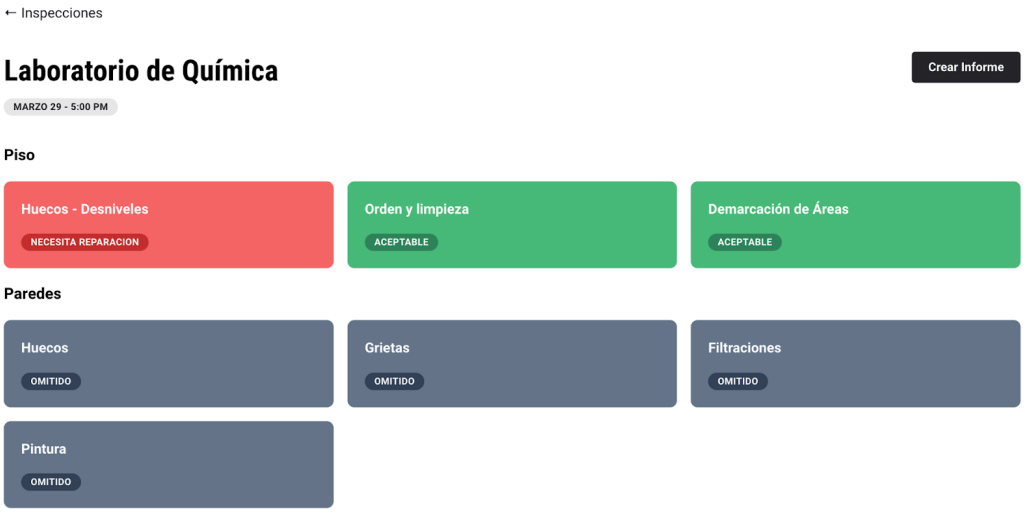
¿Cómo escribir el informe asociado a una inspección ya documentada?

El usuario de tipo **Inspector**, puede iniciar sesión en la aplicación la plataforma web https://ssst.io, y será presentado con la pantalla de **inspecciones.**

En esta pantalla el inspector podrá visualizar las inspecciones asignadas, si ya se ha realizado la evaluación de puesto de trabajo utilizando la aplicación móvil, entonces la inspección aparecerá en la lista bajo el estado **pendiente por informe**.



El inspector puede clickear cualquiera de las filas de la tabla, lo cual lo llevará a la pantalla de **Inspección**, en donde puede visualizar en forma de matriz, un resumen de todas las observaciones, positivas o negativas, para todas las condiciones en todas las categorías, que se llevaron a cabo durante la captura de datos usando la aplicación.



El inspector debe clickear el botón **Crear informe**, será presentado con la pantalla de creación de informes.

Dentro de esta pantalla, el inspector será presentado con un formulario que contiene tantas entradas como condiciones de riesgo fueron encontradas en el momento de la inspección, cada una con la evidencia correspondiente y la anotación corta escrita en el momento de la inspección, el inspector tendrá la posibilidad de redactar un análisis completo sobre la condición y sus factores de riesgo.

Nuevo Informe

Correspondiente a la inspección de ***Laboratorios - Facultad de Odontología*** ejecutada el ***27 de marzo a las 9:00 am*** por el inspector ***John Gutierrez.***

Piso

Demarcación de Áreas

INSEGURO

Las unidades no poseen demarcación con una línea amarilla

Analisis *

Escriba su análisis sobre el estado de Piso: Demarcación de Áreas.

Una vez el inspector ha terminado con la redacción de análisis para cada uno de los factores de riesgo encontrados, tendrá que escribir una conclusión, necesaria para finalizar el informe.

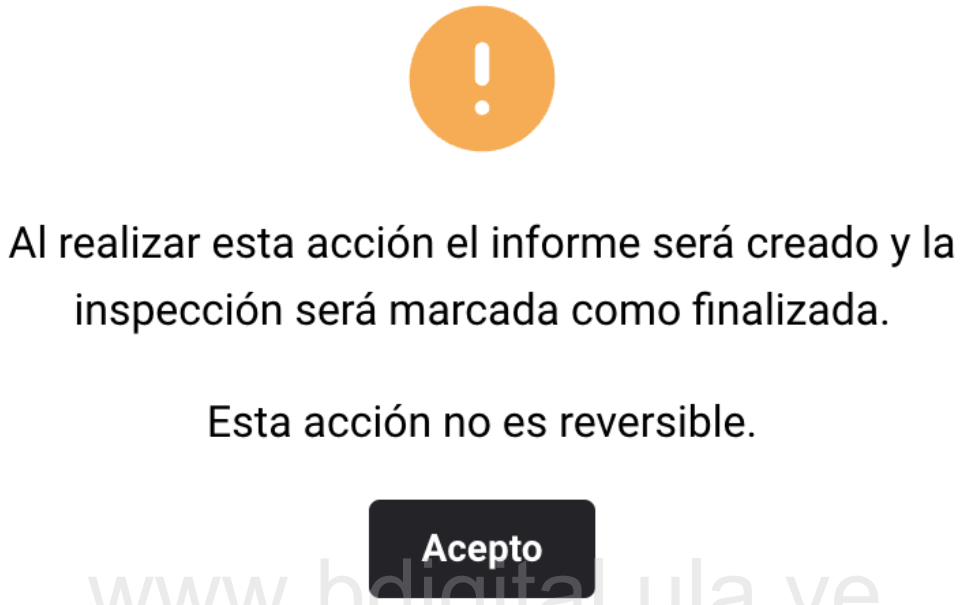
Conclusión

Conclusión final *

Escriba sus comentarios de cierre para finalizar el reporte.

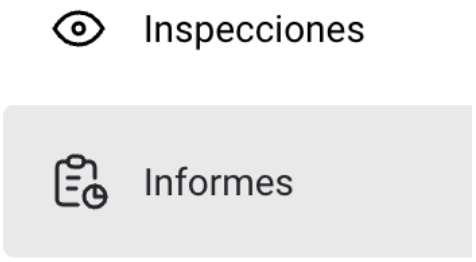
Guardar Cambios

Una vez el inspector clickee el botón **Guardar cambios,** será presentado con un diálogo de confirmación, dejando claro que el informe una vez creado ya no podrá ser alterado.

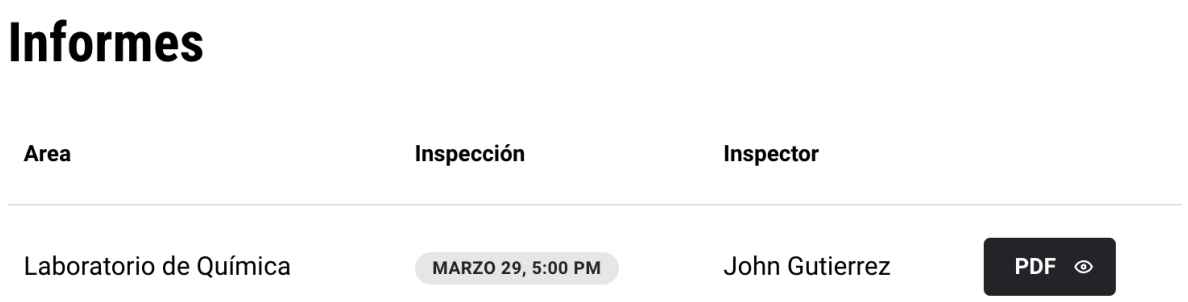


¿Cómo visualizar los informes?

Un usuario de tipo **Manager** o **Inspector** puede iniciar sesión en la plataforma web https://ssst.io, y encontrará en el lado izquierdo de la pantalla, en el panel de navegación, un elemento que le permitirá navegar hacia la pantalla de inspecciones.



En la pantalla de informes, el usuario será capaz de visualizar una tabla con todos los informes de la organización, así como un botón que permite descargar una versión en PDF.



Capítulo V

Conclusiones y Recomendaciones

Una vez desarrollados todos los aspectos en el presente estudio se muestran de manera ordenada las siguientes conclusiones y recomendaciones.

En síntesis, se llegó a las siguientes conclusiones:

- ✔ Realizar un diagnóstico de la problemática del sistema de seguridad y salud para los trabajadores de la Universidad de Los Andes relacionado con el almacenamiento de información es fundamental para identificar deficiencias y áreas de mejora.

- ✔ El proceso de diagnóstico permite identificar posibles vulnerabilidades en el almacenamiento de información relacionada con la seguridad y salud laboral de la Universidad de los Andes, lo que contribuye a fortalecer la protección de los datos sensibles.

- ✔ El análisis de la problemática en el almacenamiento de información facilita el cumplimiento de normativas en materia de seguridad y salud laboral, evitando errores y asegurando el bienestar de los trabajadores.

- ✔ Al diagnosticar y resolver problemas relacionados con el almacenamiento de información, se optimizan los recursos tanto humanos como tecnológicos,

permitiendo una gestión más eficiente y efectiva de la seguridad y salud en el trabajo.

- ✔ La creación de un modelo de base de datos adaptado a las necesidades específicas de la Universidad de los Andes en el ámbito de seguridad y salud en el trabajo garantiza una gestión eficiente y personalizada de la información relevante para esta institución.

- ✔ El diseño del modelo de base de datos permite una organización estructurada y eficaz de la información relacionada con la seguridad y salud laboral en la universidad, facilitando la consulta, actualización y análisis de los datos de manera ágil y precisa.

- ✔ Al alinear el modelo de base de datos con los casos de uso particulares de la Universidad de los Andes en materia de seguridad y salud en el trabajo, se logra una mayor coherencia y relevancia de la información almacenada, contribuyendo a una toma de decisiones más informada y estratégica.

- ✔ La implementación de un modelo de base de datos diseñado para las necesidades específicas de la universidad optimiza los procesos operativos relacionados con la seguridad y salud laboral, reduciendo tiempos de respuesta y minimizando posibles errores en la gestión de la información.

- ✔ El modelo de base de datos creado para la Universidad de los Andes en el contexto de seguridad y salud en el trabajo promueve una cultura

organizacional orientada a la prevención, el bienestar de los trabajadores y el cumplimiento de normativas, fortaleciendo así el compromiso con la seguridad laboral.

- ✔ La investigación de tecnologías adecuadas para un sistema de seguridad y salud en el trabajo en la Universidad de los Andes es fundamental para garantizar la eficacia y la adaptabilidad del producto a las necesidades específicas de la institución.

- ✔ El desarrollo de una plataforma de administrador e inspector que incluye un portal web, una aplicación móvil y una API REST para la Universidad de los Andes representa una integración tecnológica avanzada que facilita la gestión eficiente de la seguridad y salud en el trabajo en la institución.

- ✔ La combinación de un portal web y una aplicación móvil brinda accesibilidad y flexibilidad a los administradores e inspectores de la universidad, permitiéndoles acceder a la plataforma desde diferentes dispositivos y en cualquier momento, lo que agiliza los procesos de supervisión y gestión.

- ✔ La API REST actúa como un puente de comunicación entre el portal web y la aplicación móvil, permitiendo una interconexión fluida de datos y funcionalidades, lo que mejora la sincronización de la información y optimiza la experiencia de usuario.

- ✔ La plataforma de administrador e inspector proporciona herramientas y datos clave para la toma de decisiones informadas en materia de seguridad y salud en el trabajo, permitiendo una supervisión más efectiva y una respuesta rápida ante posibles riesgos laborales.

- ✔ La implementación de esta plataforma tecnológica en la Universidad de los Andes no solo mejora la eficiencia operativa en la gestión de la seguridad y salud laboral, sino que también promueve una cultura preventiva y proactiva en beneficio de la comunidad universitaria.

- ✔ La realización de pruebas reales con el personal del SSST-ULA (Servicio de Seguridad y Salud en el Trabajo de la Universidad de los Andes) permite validar la efectividad y la usabilidad de la plataforma de seguridad y salud en el trabajo en un entorno real, identificando posibles mejoras y ajustes necesarios.

- ✔ La interacción directa con el personal del SSST-ULA durante las pruebas proporciona una valiosa retroalimentación sobre la experiencia de uso, las funcionalidades requeridas y las necesidades específicas del usuario, lo que contribuye a optimizar la plataforma para satisfacer las demandas del personal.

- ✔ A través de las pruebas reales, se pueden identificar áreas de mejora tanto en la plataforma como en los procesos de capacitación, permitiendo ajustes que garanticen una implementación exitosa y una adopción efectiva por parte del personal del SSST-ULA.

Recomendaciones

- ✔ Es fundamental proporcionar una capacitación detallada y personalizada al personal del SSST-ULA sobre el uso de la plataforma, asegurando que estén familiarizados con todas las funcionalidades y puedan aprovechar al máximo sus beneficios.

- ✔ Establecer un sistema de soporte técnico eficiente y accesible para atender cualquier consulta o problema que pueda surgir durante la utilización de la plataforma.

- ✔ Implementar actualizaciones y mejoras de manera regular para garantizar que la plataforma se mantenga actualizada y cumpla con las necesidades en constante evolución del SSST-ULA.

- ✔ Capacitar al personal del SSST-ULA sobre las mejores prácticas en el almacenamiento de información sensible, concientizando sobre la importancia de proteger los datos relacionados con la seguridad y salud en el trabajo.

- ✔ Realizar evaluaciones periódicas de la plataforma en colaboración con el personal del SSST-ULA para recopilar retroalimentación y sugerencias de mejora.

- ✔ Realizar pruebas rigurosas de la plataforma antes de su formalidad para identificar y corregir posibles fallos o deficiencias.

Referencias

- ✔ Cambridge University Press. (s. f.). Significado de PLATFORM. Cambridge Dictionary. Recuperado 16 de marzo de 2024, de https://dictionary.cambridge.org/es/diccionario/ingles/platform.
- ✔ Estrela, D., Batista, S., Martinho, D., & Marreiros, G. (2017). A Recommendation System for Online Courses. Researchgate. https://www.researchgate.net/profile/David-Estrela/publication/315848045_A_Recommendation_System_for_Online_Courses/links/6038d496a6fdcc37a8520283/A-Recommendation-System-for-Online-Courses.pdf
- ✔ Hasan, M. (2019). SEKHAO: AN E-LEARNING PLATFORM OF VIRTUAL CLASSROOM FOR REMOTE EDUCATION. Dspace Repository. http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/3783/P15020%20%2813_%29.pdf?sequence=1&isAllowed=y
- ✔ IBM. (s.f.). ¿Qué es Docker? Recuperado de https://www.ibm.com/mx-es/topics/docker
- ✔ McKenzie, C. (2018, noviembre). Git vs. GitHub: What is the difference between them? The Server Side. https://www.theserverside.com/video/Git-vs-GitHub-What-is-the-difference-between-them

✔ Railway. (s.f.). Deploy BullMQ with BullBoard on Railway. Recuperado de https://railway.app/template/odzp-I

✔ UOC - Universitat Oberta de Catalunya (UOC). (2017, 7 de marzo). Benvinguts a la Biblioteca de la UOC [vídeo en línea]. YouTube. https://www.youtube.com/watch?v=fL3EdXW-Hpg