Sistema de Gestión de Deforestación de Cundinamarca y Boyacá.

PROYECTO: TALENTO TECH



DANIEL ALBERTO IBARRA AROCA

RAFAEL BARAHONA BELTRAN

JAIME ANDRES DURAN

MARIBEL ADRIANA RAMIREZ CIFUENTES

JORGE ARMANDO DOTOR DELGADILLO

7 mar 2024

## Objetivo

Desarrollar una plataforma web con contenidos multimedia y capacidad de comentarios, enfocado en noticias, revistas y blogs de temas inherentes a la deforestación en los departamentos de Cundinamarca y Boyacá.

## Licencia

## Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0)

La presente kata de arquitectura de software para un sistema de Gestión de inventario de productos se encuentra bajo la licencia Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0).

De acuerdo con los términos de esta licencia, se permite compartir y distribuir la kata siempre y cuando se otorgue crédito y reconocimiento a su autor original. Sin embargo, no se permite su modificación ni su uso para fines comerciales sin la debida autorización.

Al hacer uso de este documento, usted acepta cumplir con las condiciones y restricciones establecidas por la licencia CC BY-NC-ND 4.0. Para obtener más información sobre los términos y condiciones de esta licencia, visite el siguiente enlace:

## Introducción

* **Plataforma Web**: Es un espacio en Internet que permite la ejecución de diversas aplicaciones o programas. Puede ser utilizado para almacenar información personal o a nivel empresarial.
* **Funcionalidades**: Las plataformas digitales pueden unificar y optimizar operaciones de negocio y sistemas de TI a través del uso de software y tecnología especializada.
* **Diferencia entre Página Web y Plataforma Web**: Mientras una página web muestra información estática, una plataforma web es un sistema más complejo y dinámico que permite interactuar con usuarios y ejecutar diversas funciones.

Esta plataforma presenta noticias e información sobre la deforestación en los Departamentos de Cundinamarca y Boyacá, con el historial y estadísticas de las diferentes causas.

El objetivo principal de la plataforma web es generar conciencia en el lector respecto al impacto causado en la flora, fauna y bosques de la región, de esta manera generar información y compartir del contenido de esta página buscando la participación ciudadana la cual podrá tener acceso a la misma y de esta forma crear un vínculo que busque la protección de nuestros recuerdos.

# Requerimientos Funcionales

## Autenticación y Autorización

### Registro de participación ciudadana

El sistema debe llevar un registro de los comentarios / denuncias de las personas que participan en la plataforma, estos comentarios se validan con el correo electrónico de la persona que realiza el comentario / denuncia.

## Lógica de Negocio

### Registro de Comentarios / Denuncias

La plataforma Web almacena la información enviada por los usuarios, esta información se utilizará de manera pedagógica entre los usuarios que participan e interactúan en la plataforma.

#### Captura de información

La Plataforma web proveerá un formulario para el ingreso de la información enviada por el visitante. El formulario debe ser intuitivo y fácil de usar, con campos claramente etiquetados y validaciones apropiadas para garantizar la calidad de la información ingresada.

#### Validación de datos

La Plataforma Web debe validar la información ingresada por los visitantes antes de almacenarla en la base de datos, asegurando que los datos sean completos, coherentes y no contengan errores y la fuente del artículo y/o escrito cuando aplique. Esto incluye la verificación de campos obligatorios, y la validación por captcha del formulario enviado.

#### Imágenes y multimedia

El sistema debe permitir a los usuarios cargar imágenes u otros archivos multimedia relacionados con el Comentario / Denuncia como fotografías, planos, especificaciones técnicas, entre otros. Estos archivos deben ser almacenados de forma segura y estar accesibles para visualizarlos o descargarlos según corresponda.

#### Historial de cambios

El sistema debe mantener un historial de los comentarios / denuncias de los usuarios llegando de manera efectiva el motivo del rechazo de la publicación cuando aplique.

### Búsqueda y consulta de comentarios / denuncias

La plataforma Web permite a los usuarios buscar y consultar la información de los comentarios / denuncias publicados allí.

#### Función de búsqueda

El sistema debe ofrecer una función de búsqueda y filtrado que permita a los usuarios localizar rápidamente los comentarios / denuncias de interés. Los usuarios deben poder buscar productos por diferentes criterios, como nombre, tipo de denuncia, fecha de publicación, usuario que publicó.

#### Visualización de resultados

Los resultados de la búsqueda deben presentarse en una lista o tabla que muestre la información relevante de los comentarios / denuncias encontrados, descripción, detalle, fecha publicación,.

# Requerimientos No Funcionales

## Rendimiento

La plataforma Web debe ofrecer un rendimiento adecuado y eficiente en su funcionamiento, garantizando tiempos de respuesta rápidos y un uso eficiente de los recursos disponibles.

### Tiempos de respuesta

El sistema debe ser capaz de procesar y responder a las solicitudes de los visitantes de manera rápida y eficiente. Esto incluye, por ejemplo, la búsqueda y consulta de los comentarios. Los tiempos de respuesta deben ser lo suficientemente rápidos como para proporcionar una experiencia de usuario satisfactoria y mantener la productividad de los visitantes que interactúan con el sistema.

### Carga concurrente

El sistema debe ser capaz de manejar múltiples solicitudes de forma simultánea sin degradar significativamente el rendimiento. Esto es especialmente importante en situaciones en las que varios visitantes están interactuando con el sistema al mismo tiempo, o cuando se realizan operaciones que involucran un alto volumen de datos.

### Eficiencia en el uso de recursos

El sistema debe hacer un uso eficiente de los recursos de hardware y software disponibles, como CPU, memoria, almacenamiento y ancho de banda de red. Un uso eficiente de los recursos permite minimizar los costos de infraestructura y garantizar un rendimiento óptimo incluso en entornos con recursos limitados.

### Escalabilidad

El rendimiento del sistema debe mantenerse a medida que aumenta la cantidad de datos, productos y usuarios. Esto puede incluir la capacidad de escalar horizontalmente (añadiendo más servidores o nodos) o verticalmente (incrementando los recursos de un servidor existente) para satisfacer las demandas de rendimiento a medida que crece el sistema.

### Optimización de consultas y transacciones

El sistema debe implementar técnicas de optimización en la realización de consultas a la base de datos y en la gestión de transacciones, lo que permite reducir los tiempos de respuesta y garantizar un rendimiento óptimo en operaciones que involucren la manipulación de datos.

## Seguridad

El sistema de gestión de inventario debe garantizar la seguridad de la información y de los recursos, protegiendo la confidencialidad, integridad y disponibilidad de los datos y del propio sistema. Copias de seguridad periódicas de las bases de Datos y de la plataforma en general.

### Autenticación y autorización

El sistema debe implementar mecanismos sólidos de autenticación y autorización, asegurando que los visitantes publiquen los formularios de manera segura. Esto incluye el uso de transacciones seguras, mecanismos de recuperación de validación de información, si es necesario, autenticación de dos factores o acceso por token.

El sistema debe emplear encriptación para proteger la confidencialidad e integridad de los datos almacenados y transmitidos. Esto incluye la encriptación de datos en reposo (por ejemplo, en la base de datos) y en tránsito (por ejemplo, durante la comunicación entre el cliente y el servidor).

### Protección contra vulnerabilidades y ataques

El sistema debe estar protegido contra vulnerabilidades y ataques comunes, como inyección SQL, cross-site scripting (XSS), ataques de fuerza bruta y otros. Esto implica la adopción de prácticas de desarrollo seguro, la validación y saneamiento de entradas, y la configuración adecuada de los componentes del sistema, entre otros.

### Auditoría y monitorización

El sistema debe incluir mecanismos de auditoría y monitorización que permitan rastrear y analizar la actividad de los usuarios y del propio sistema. Esto facilita la detección temprana de actividades sospechosas o maliciosas y permite tomar medidas para mitigar los riesgos de seguridad.

## Usabilidad

El sistema de gestión de inventario debe ser fácil de usar y accesible para los visitantes, ofreciendo una experiencia de usuario eficiente, intuitiva y agradable que facilite la adopción y el uso del sistema en el día a día.

### Diseño de interfaz

La interfaz de usuario del sistema debe ser clara, organizada y fácil de navegar. Debe contar con elementos visuales y de navegación coherentes y utilizar colores, tipografías y estilos adecuados que permitan a los usuarios encontrar y utilizar las funcionalidades del sistema de manera intuitiva.

### Interacciones sencillas

Las interacciones entre el usuario y el sistema deben ser sencillas y eficientes. Esto implica minimizar la cantidad de clics o acciones necesarias para completar tareas comunes y ofrecer atajos y herramientas que faciliten la realización de acciones frecuentes.

### Retroalimentación y mensajes de error

El sistema debe proporcionar retroalimentación adecuada a los usuarios sobre el estado de las acciones y las operaciones realizadas. Esto incluye mensajes de error claros y útiles que informen a los usuarios sobre posibles problemas y cómo resolverlos, así como indicadores de progreso y confirmaciones de acciones exitosas.

## Testing

El sistema de gestión de inventario debe incluir pruebas unitarias y funcionales que garanticen la calidad, fiabilidad y correcto funcionamiento de sus componentes y funcionalidades.

### Pruebas unitarias

Las pruebas unitarias se enfocan en evaluar y validar el correcto funcionamiento de cada componente o unidad del sistema de forma aislada. Este tipo de pruebas ayuda a detectar errores y problemas en etapas tempranas del desarrollo, facilitando la corrección y previniendo la propagación de errores en el sistema.

#### Cobertura de código

El sistema debe contar con una cobertura de ramas de código del 90% mediante pruebas unitarias, abarcando la mayoría de las funciones y clases críticas para el funcionamiento del sistema de inventario.

#### Automatización

Las pruebas unitarias deben ser automatizadas y ejecutadas continuamente durante el proceso de desarrollo e integración, lo que permitirá identificar rápidamente errores y problemas en el código.

### Pruebas funcionales

Las pruebas funcionales se enfocan en validar que las funcionalidades del sistema se comporten de acuerdo a los requerimientos funcionales especificados. Estas pruebas evalúan el sistema en su conjunto, considerando las interacciones entre sus componentes y el cumplimiento de los casos de uso previstos.

#### Casos de prueba

Se deben elaborar casos de prueba que cubran los principales escenarios de uso del sistema de inventario, incluyendo casos de éxito y casos de error, para asegurar que el sistema se comporte de manera adecuada en diversas situaciones.

#### Automatización

Las pruebas funcionales deben ser, en la medida de lo posible, automatizadas y ejecutadas periódicamente, para validar que el sistema continúa cumpliendo con los requerimientos funcionales a medida que evoluciona.

# Aspectos Técnicos

En esta sección se presentan diversas recomendaciones en cuanto a arquitecturas de software, mecanismos de autenticación y autorización, estrategias de cifrado de contraseñas y el uso de ambientes cloud para el desarrollo de la plataforma web. Cabe destacar que se aceptará cualquier propuesta de solución que cumpla con los requerimientos funcionales y no funcionales establecidos; sin embargo, las siguientes recomendaciones se proporcionan como una guía para facilitar la toma de decisiones y asegurar que la solución propuesta se ajuste a las mejores prácticas en la industria.

## Arquitecturas de software

### Arquitectura de microservicios

La arquitectura de microservicios es una de las opciones recomendadas, ya que permite desarrollar y desplegar funcionalidades del sistema como servicios independientes y escalables. Esto facilita la evolución y el mantenimiento del sistema a lo largo del tiempo, así como la integración con otros sistemas y tecnologías.

### Arquitectura de capas

Otra opción recomendada es la arquitectura de capas, que organiza el sistema en módulos o capas separadas por responsabilidades, como la capa de presentación, la capa de lógica de negocio y la capa de acceso a datos. Esta arquitectura facilita la reutilización de componentes, la separación de responsabilidades y la adaptabilidad a cambios en los requerimientos.

## Mecanismos de autenticación y autorización

### OAuth 2.0 y OpenID Connect

Estos protocolos estandarizados permiten la autenticación y autorización de usuarios mediante tokens de acceso y la delegación de permisos entre aplicaciones y servicios. Su uso garantiza la seguridad y la interoperabilidad del sistema, así como la integración con proveedores de identidad externos como Google, Facebook o sistemas corporativos.

### JSON Web Tokens (JWT)

Los JWT son tokens de autenticación y autorización que pueden ser firmados y cifrados, lo que permite validar y proteger la información de las sesiones de usuario. Su uso en conjunto con OAuth 2.0 y OpenID Connect proporciona una solución completa y segura para la gestión de usuarios y permisos en el sistema.

## Estrategias de cifrado de contraseñas

### Funciones de derivación de contraseñas

El uso de funciones de derivación de contraseñas, como bcrypt, scrypt o Argon2, es fundamental para garantizar la seguridad de las contraseñas almacenadas en el sistema. Estas funciones aplica un cifrado sólido y un proceso de "salting" y "hashing" iterativo, lo que dificulta significativamente los ataques de fuerza bruta y la obtención de contraseñas a partir de los hashes almacenados.

## Uso de ambientes cloud

### Plataformas como servicio (PaaS) y contenedores

El uso de plataformas como servicio, como AWS Elastic Beanstalk, Google App Engine o Azure App Service, y tecnologías de contenedores, como Docker y Kubernetes, permite desplegar y escalar la Plataforma web de manera eficiente y flexible en ambientes cloud. Esto facilita la administración de recursos, la monitorización del rendimiento y la adaptabilidad a cambios en la demanda.

### Bases de datos en la nube

Utilizar bases de datos en la nube, como Amazon RDS, Google Cloud SQL o Azure SQL Database, proporciona una solución escalable, segura y de alto rendimiento para el almacenamiento y la plataforma web. Además, permite la implementación de copias de seguridad automáticas, la replicación de datos y la recuperación ante desastres, garantizando la disponibilidad y la integridad de la información en todo momento.

SQL-Lite ???

### Almacenamiento en la nube

Utilizar servicios de almacenamiento en la nube, como Amazon S3, Google Cloud Storage o Azure Blob Storage, permite guardar y gestionar archivos y documentos relacionados con la plataforma web, como imágenes de productos o archivos de importación y exportación. Estos servicios ofrecen una solución escalable, segura y de alto rendimiento para el almacenamiento y la distribución de archivos en la nube.

### Servicios de seguridad en la nube

La implementación de la plataforma web en ambientes cloud permite aprovechar los servicios de seguridad ofrecidos por los proveedores de la nube, como la protección contra ataques DDoS, la monitorización de la seguridad y la gestión de certificados SSL/TLS. Estos servicios ayudan a garantizar la seguridad y la privacidad de la información del inventario y los usuarios del sistema.

ComuniWebSpot

MediaInteractHub

VoxNewsWeb

OpinioSphere

CrearNotiBlog

WebInnovaCom

MultiContHub

ComentaWebHub

OptiMagNews

InsightBlogHub

ActiveMediaConnect

LuzVerde

| **ComuniWebSpot** |  |
| --- | --- |
| **MediaInteractHub** |  |
| **VoxNewsWeb** |  |
| **OpinioSphere** |  |
| **CrearNotiBlog** |  |
| **WebInnovaCom** |  |
| **MultiContHub** |  |
| **ComentaWebHub** |  |
| **OptiMagNews** |  |
| **InsightBlogHub** |  |
| **ActiveMediaConnect** |  |
| **LuzVerde** |  |