Documento de Arquitectura de Software

Proyecto: UYWA

Tabla de contenido

[1. Introducción 3](#_Toc165538972)

[1.1. Propósito 3](#_Toc165538973)

[1.2. Alcance 3](#_Toc165538974)

[1.3. Vista global 3](#_Toc165538975)

[2. Macro Arquitectura 4](#_Toc165538976)

[2.1. Modelo Multicapa 4](#_Toc165538977)

[3. Vista física 4](#_Toc165538978)

[4. Vista Funcional o Lógica 5](#_Toc165538979)

[4.1. Diagrama de Componentes 5](#_Toc165538980)

[4.2. Diagrama de Casos de Uso 6](#_Toc165538981)

[5. Vista de Despliegue 7](#_Toc165538982)

[5.1. Diagrama de despliegue 7](#_Toc165538983)

[6. Vista de Procesos 8](#_Toc165538984)

[6.1. Diagrama de Procesos 8](#_Toc165538985)

1. Introducción
   1. Propósito

El propósito de este documento es definir y dar a conocer la Arquitectura de Software, la cual será utilizada para el desarrollo del proyecto “UYWA”. Se utilizarán las vistas de proceso y física, para describir la estructura y funcionalidad, así como la distribución física, la descripción y la interacción del sistema. Este documento va dirigido al equipo de desarrollo, con el fin de proporcionar las bases arquitecturales, además de una visión global y comprensible del diseño, sobre las cuales el proyecto se define para ser llevado a cabo. Con el documento podrán comprender la organización de procesos del sistema y ver reflejados la aplicación de requisitos no funcionales tales como la mantenibilidad, el rendimiento y la disponibilidad.

* 1. Alcance

Este documento describe cómo se va a desarrollar el sistema que permitirá a los usuarios de la plataforma, reportar incidentes de caza ilegal, secuestro y venta de animales silvestres, así como visualizar eventos de concientización y ganar insignias por sus contribuciones. La aplicación es Web y por tal motivo funcionaría en computadores que tengan un navegador Web. En este documento se especifican las restricciones, los atributos de calidad, las vistas arquitecturales y los patrones de diseño, que se implementarán por parte del equipo de desarrollo.

* 1. Vista global

En el documento se encontrará una macro arquitectura, con especificaciones de las vistas usadas en el proyecto, una lista de metas y restricciones arquitectónicas que mostrarán los atributos de calidad observables y no observables que tiene el equipo, por último, las diferentes vistas con sus respectivos diagramas, que nos ayudarán a encontrar la comprensión deseada.

Imagen que contiene Diagrama

Descripción generada automáticamente

1. Macro Arquitectura

La macro arquitectura que comprende el sistema se basa en 3 vistas calves, la vista lógica, física y de procesos que comprenderán los siguientes diagramas.

* Vista lógica: Diagrama de componentes, debido a la facilidad que tiene junto con su estructura el permitir un análisis completo y a fondo de las diversas etapas de desarrollo de un aplicativo software, expresando las relaciones que se obtendrán las distintas dependencias de un componente.
* Vista física: Diagrama informal, el cual permite mostrar la interacción física necesaria y usada entre los dispositivos que incorpora el proyecto. Diagrama de despliegue, ya que muestra la arquitectura del sistema desde el punto de vista de red de los clientes a los servidores.
* Vista de procesos: Diagrama de actividades. Por su capacidad para ilustrar las acciones internas de un proceso, capturar la especificación de un caso de uso o historia de usuario y reflejarlos en los flujos de procesos del negocio.
  1. Modelo Multicapa

Denominado también 3-tier, es un patrón de arquitectura a de software encargado de separar la lógica de negocio de la interfaz del usuario en varios componentes distintos. Donde el principal objetivo es dividir el software en capas cumpliendo cada capa en una función específica siendo así un sistema modular, ofreciendo una mejor mantenibilidad, alta escalabilidad y flexibilidad al cambio.

1. Vista física

Diagrama Informal de la Vista Física Este diagrama describe las características físicas de los elementos que componen la arquitectura del proyecto “UYWA”.

El diagrama contempla el uso de la aplicación por parte del usuario que inicialmente ingresa al navegador desde un dispositivo, ya sea un computador de escritorio o dispositivos móviles. El usuario ingresará a la página web de la plataforma que está alojada en un servidor de aplicaciones Node.js. De igual forma, el servidor tiene como gestor de bases de datos PostgreSQL.

Atributos de Calidad Presentes

**Escalabilidad**: La plataforma tiene la capacidad de crecer cuando se incrementa la carga de trabajo (el número de peticiones). Cada máquina tiene una capacidad finita de recursos y por lo tanto sólo puede servir un número limitado de peticiones. Incorpora el patrón de diseño multicapa ya que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario, de este modo cada vez que se requiera implementar un nuevo módulo o funcionalidad a la aplicación se manipulará por sus capas.

**Portabilidad**: La aplicación web es portable ya que implementa el patrón de diseño Responsive Design y se puede adaptar a diferentes dispositivos con múltiples tamaños de pantalla de computadores.

Notación del Diagrama Informal Físico

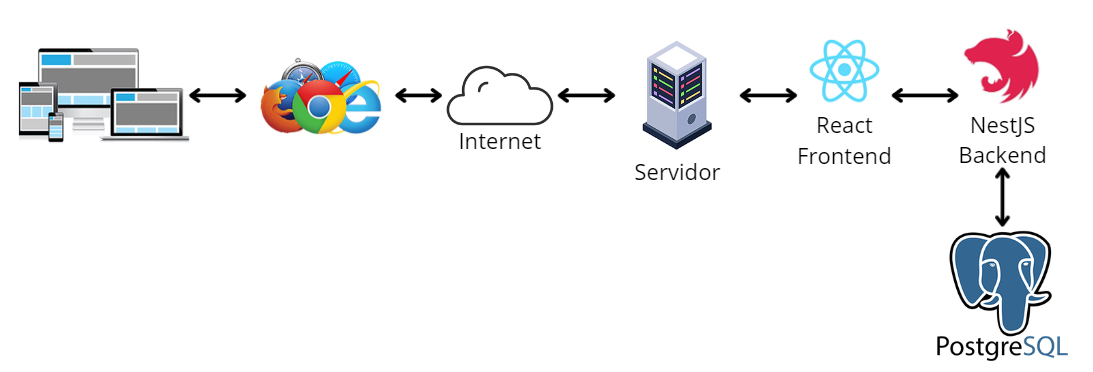
Usuarios: Persona que habitualmente usa el servicio.

**Computador (PC o portátil):** Dispositivo electrónico capaz de procesar datos que llegan desde el exterior (entradas) y proporcionan un resultado (salida). El cual tiene conexión a red permanente o intermitente a una red de datos.

**Internet**: Unión de todas las redes y computadoras distribuidas por todo el mundo, en otras palabras, es una red global en que se conjuntan todas las redes que utilizan el protocolo TCP/IP y que son compatibles entre sí.

**Servidor**: Es un ordenador encargado de suministrar información a una serie de clientes, que puede ser tanto personas como otros dispositivos conectados a él. La información que se puede transferir es múltiple y variada, desde archivos de texto, imagen, vídeo y programas informáticos, base de datos, entre otros.

**Aplicativo web**: Son aquellas herramientas que los usuarios pueden utilizar accediendo a un servidor web a través de internet mediante un navegador.



1. Vista Funcional o Lógica
   1. Diagrama de Componentes

Este diagrama representa el marco de trabajo de la relación entre las diferentes dependencias del stack tecnológico (React, NestJS, Prisma, Passport.js y PostgreSQL) y los componentes que se encuentran dentro de esta arquitectura para la funcionalidad de la plataforma “UYWA”.

En este diagrama se ve reflejado que el sistema es multicapa, representado por la capa de presentación (React), la capa de lógica del negocio (NestJS y Passport.js), y la capa de acceso a datos (Prisma y PostgreSQL). Al ser un sistema multicapa nos permite realizar de manera óptima y eficiente la mantenibilidad ya que se puede dar soporte o atención a un punto específico sin necesidad de tocar el resto del sistema, la modificabilidad del sistema permitiendo incluir componentes en una línea de tiempo sin que afecte lo que ya existe, como también permite mayor funcionalidad.

En la capa de presentación se utiliza el framework de diseño React para construir una interfaz de usuario interactiva y dinámica. React permite hacer un diseño web responsivo y portable para diferentes plataformas de computadores y que tengan un navegador Web.

En la capa de lógica de negocio se utiliza NestJS, un marco de aplicación web para Node.js, que proporciona un conjunto robusto de características para las aplicaciones web y móviles. Express maneja las rutas y redirecciona a un controlador donde éste a su vez tiene unas funciones específicas de acuerdo con la ruta invocada. Dentro del controlador tenemos funciones propias del marco de trabajo y la lógica de procesos que ejecuta el sistema. Además, se utiliza Passport.js como un middleware de autenticación para manejar la autenticación de usuarios.

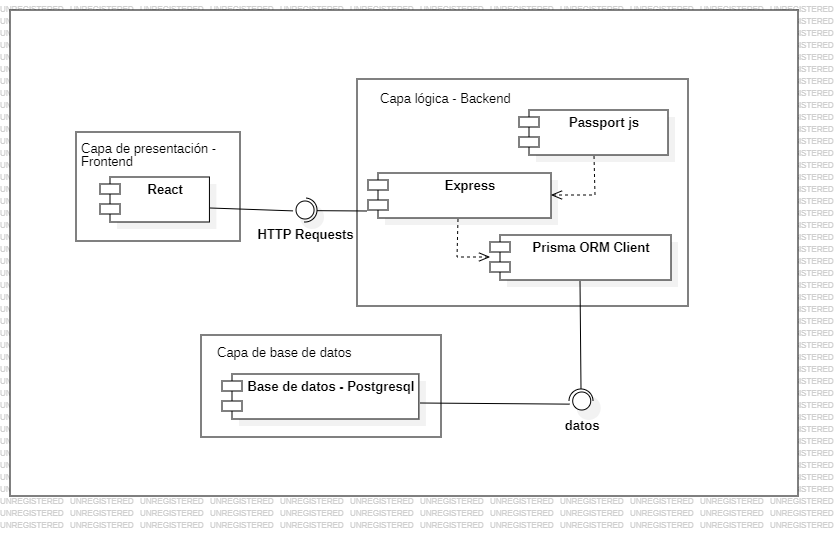
En la capa de acceso a datos se utiliza Prisma, un ORM (Object-Relational Mapping) de código abierto para JavaScript y TypeScript que te permite interactuar con tu base de datos de una manera fácil y segura. Prisma proporciona una API para interactuar con tu base de datos PostgreSQL. En las entidades se implementa el patrón de diseño Singleton permitiendo mayor rendimiento y concurrencia en las peticiones a los servicios de la aplicación dando prioridad a una petición.

Los atributos de calidad presentes son:

**Mantenibilidad** propende a disminuir la alta cohesión de componentes para garantizar que todo el sistema sea más dinámico y adaptable a las necesidades de expansión según lo requiera. Además de dedicar recursos necesarios a cada una de las capas y permitir un procesamiento en paralelo de las responsabilidades de estas, permitiendo que sea manejable en caso una posible expansión del sistema mediante el patrón multicapa el cual tendrá efecto en todo el aplicativo web.

**Modificabilidad** se aplica en la separación de las tareas, permitiendo que la aplicación sea ligera y entendible, además de que se puede realizar cambios en una parte de la aplicación sin afectar los demás componentes mediante el patrón multicapa.

**Funcionalidad** se aplica en todo el aplicativo la estandarización de componentes que se desarrollaron de forma separados por capas para ser integrados con otros, y así tener subcategorías y la interoperabilidad del aplicativo.

****

NestJS

* 1. Diagrama de Casos de Uso

Este diagrama describe las interacciones entre los usuarios (o actores) y el sistema, en términos de casos de uso. Un caso de uso es una secuencia de acciones que proporciona un valor medible a un actor.

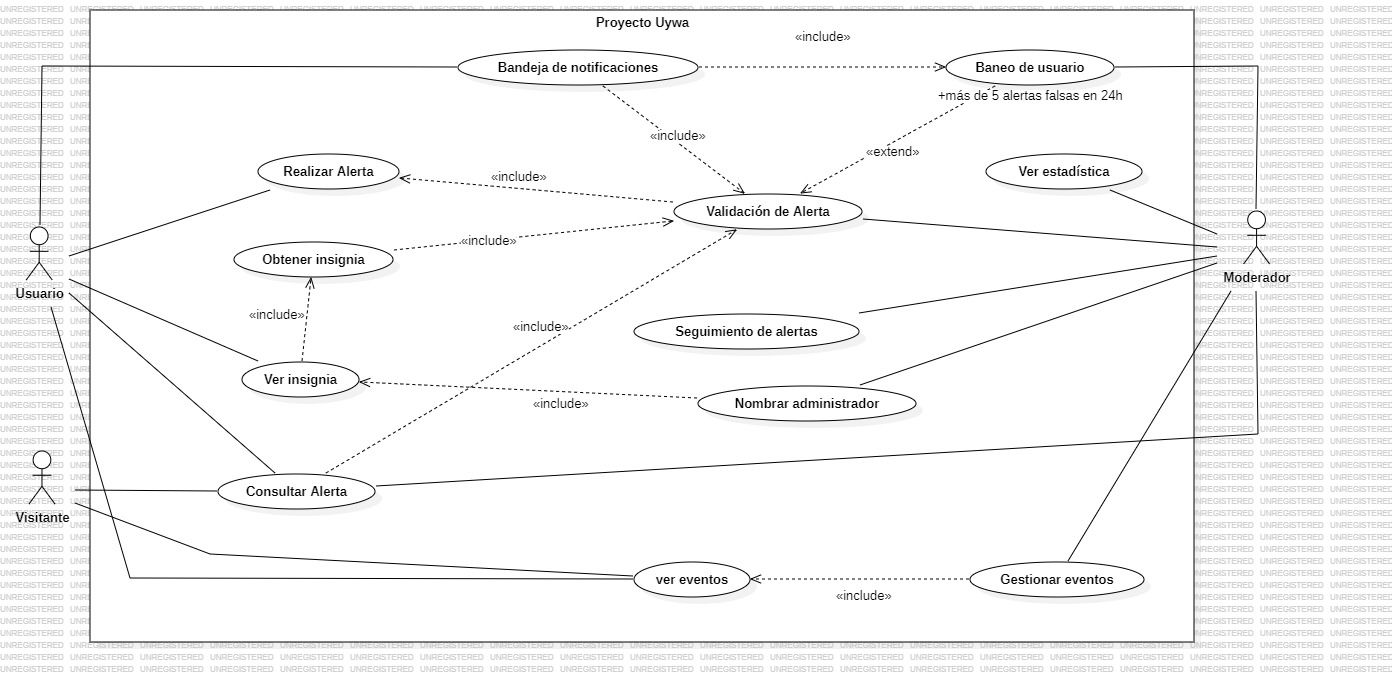
**Actores** Los actores representan a los usuarios o sistemas externos que interactúan con el sistema. En tu caso, los actores podrían ser:

**Usuario**: Este actor está registrado en el sistema y puede ver alertas y eventos, reportar incidentes y ganar insignias por sus contribuciones.

**Visitante**: Este actor será aquel que no está registrado en el sistema, pero puede realizar ciertas acciones dentro de la misma.

**Administrador**: Este actor puede verificar los informes de incidentes y administrar la plataforma, así como eventos.

Las **relaciones** representan las interacciones que tiene los actores t los casos de uso.



1. Vista de Despliegue
   1. Diagrama de despliegue

Este diagrama describe las configuraciones de redes físicas, sobre las cuales el software será desarrollado. Se indican los nodos físicos que interactúan con la aplicación web tales como un nodo para el dispositivo donde se accede la aplicación web en un PC con acceso a Internet y un navegador. Éste envía las peticiones por HTTP para mayor seguridad de información accediendo al nodo del servidor web donde se aloja el servidor NestJS y el ORM Prisma con el componente de base de datos PostgreSQL.

Dentro del servidor NestJS, tenemos la funcionalidad de la aplicación dividida por capas tales como la capa de presentación (React), lógica del negocio (NestJS y Passport.js), y acceso a datos (Prisma y PostgreSQL), permitiéndonos tener mejor mantenibilidad, modificabilidad y funcionalidad.

Por otra parte, la lógica de negocios integra las interfaces de APIs la cual sería el backend de la lógica del negocio creada por microservicios. En este caso, NestJS actúa como el microservicio que maneja las solicitudes HTTP y proporciona las APIs para la aplicación.

Los atributos de calidad presentes son:

**Modificabilidad** se aplica en la separación de las tareas, permitiendo que la aplicación sea ligera y entendible, además de que se puede realizar cambios en una parte de la aplicación sin afectar los demás componentes mediante el patrón multicapa.

Diagrama

Descripción generada automáticamente

**NestJSS**

NestJS**S**

1. Vista de Procesos
   1. Diagrama de Procesos

En esta vista se muestran los procesos que hay en el sistema y la forma en la que se comunican estos procesos; es decir, se representa desde la perspectiva de un integrador de sistemas, el flujo de trabajo paso a paso de negocio y operacionales de los componentes que conforman el sistema. Para completar la documentación de esta vista se puede incluir el diagrama de actividad de UML.

El diagrama de actividades muestra un proceso de negocio o un proceso de software como un flujo de trabajo a través de una serie de acciones. Las personas, los componentes de software o los equipos pueden realizar estas acciones.

Los atributos de calidad presentes son:

La **confidencialidad** en tu proyecto “UYWA” asegura la capacidad del software de garantizar que cualquiera de sus características y/o contenidos estén enmascarados u ocultos de entidades no autorizadas.

El diagrama de Realizar Alertas describe como el usuario registrado podrá subir las alertas al sistema.

Diagrama: Realizar Alertas

Diagrama

Descripción generada automáticamente

Descripción de Actividades

Tabla

Descripción generada automáticamente

El diagrama de Ver Alertas describe el proceso que el usuario o visitante debe realizar para poder ver las alertas registradas.

Diagrama: Ver Alertas

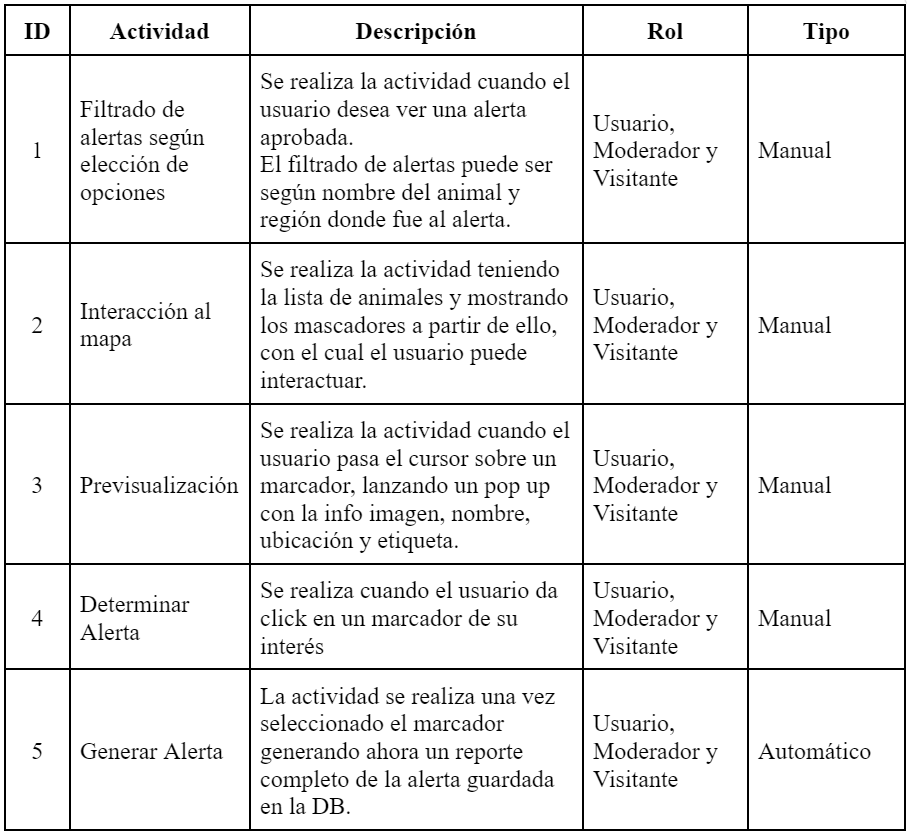
Diagrama

Descripción generada automáticamente

Diagrama

Descripción generada automáticamente

Descripción de actividades



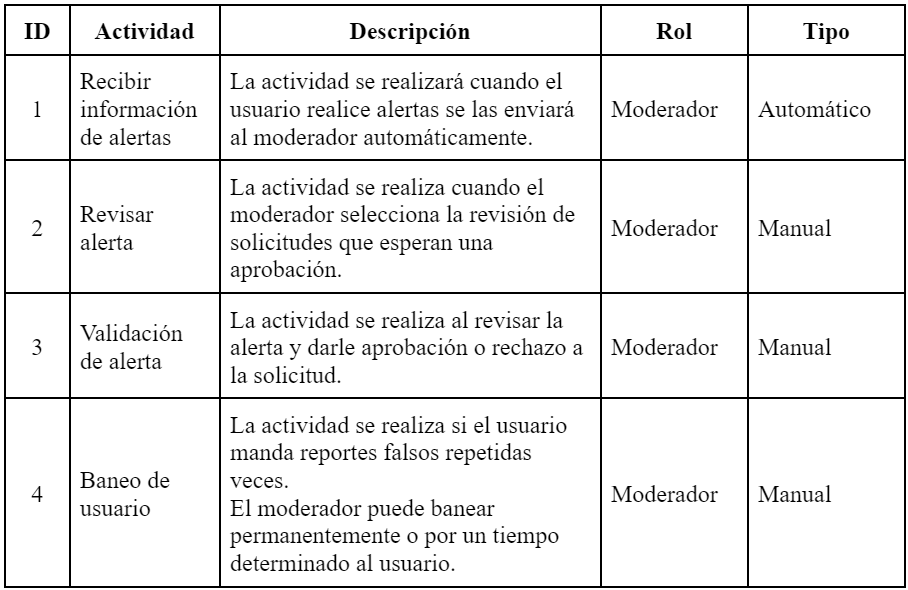
El diagrama de Control de alertas describe el proceso por el cual el moderador validará las alertas y realizará el control de los usuarios del sistema.

Diagrama: Control de alertas

Diagrama

Descripción generada automáticamente

Descripción de actividades



El diagrama de Promover Usuario describe el proceso por el cual el usuario puede ser promovido a moderador dentro del sistema.

Diagrama: Promover Usuario

Diagrama

Descripción generada automáticamente

Descripción de actividades

