

---

**Universidad La Salle**

---

**Sistema de Detección de Plagio Académico  
PlagiTracker**

**Versión 2.1**

# Documento de Arquitectura de Software

## 1. Introducción

### 1.1 Propósito

Este documento proporciona una visión general arquitectónica del Sistema de Detección de Plagio Académico PlagiTracker. Para ello se presentan diversas vistas arquitectónicas para atender diferentes aspectos del sistema. Con éste se pretende capturar y establecer las decisiones arquitectónicas significativas, que han sido definidas para el desarrollo del sistema.

### 1.2 Alcance

El alcance del presente documento cubre la arquitectura completa del Sistema de Detección de Plagio Académico PlagiTracker.

### 1.3 Definiciones, Acrónimos, y Abreviaturas

**Frontend:** Componente encargado de la interfaz de usuario y la interacción con el cliente. Desarrollado utilizando Vue.js 3 con Typescript.

**Backend:** Componente encargado de la lógica de negocio, la detección de plagio, y la comunicación con la base de datos; desarrollado utilizando .Net.

**Vue.js:** Framework de JavaScript utilizado para el desarrollo del frontend del sistema.

**.Net:** Plataforma de desarrollo de software creada por Microsoft que permite la creación de aplicaciones tanto para web como para escritorio y móviles. Se utiliza en el backend del sistema para gestionar la lógica de negocio del sistema PlagiTracker.

**PostgreSQL:** Sistema de gestión de bases de datos relacionales utilizado para almacenar los datos de la aplicación, incluyendo los códigos fuente a analizar y los resultados de la detección de plagio.

**Selenium:** Herramienta utilizada para la automatización de pruebas y la detección de plagio, permitiendo la comparación automática de código Java en múltiples entornos.

**Java:** Lenguaje de programación orientado a objetos, principal objetivo de la detección de plagio en este sistema.

**Código fuente:** Conjunto de instrucciones y estructuras escritas en Java, objeto del análisis de plagio.

**Web Scraping:** Proceso automatizado de extracción de datos de páginas web, utilizado en el sistema para obtener ejemplos de código Java disponibles en línea, con el fin de realizar comparaciones y detectar posibles casos de plagio.

**API:** Interfaz de programación de aplicaciones (Application Programming Interface), utilizada para permitir la comunicación entre diferentes componentes del sistema, como el frontend y el backend.

**IDE:** Entorno de desarrollo integrado (Integrated Development Environment), software utilizado por los desarrolladores para escribir, compilar y depurar el código Java.

**Git:** Sistema de control de versiones utilizado para gestionar los cambios en el código fuente del sistema de detección de plagio.

PlagiTracker
Documento de Arquitectura de Software

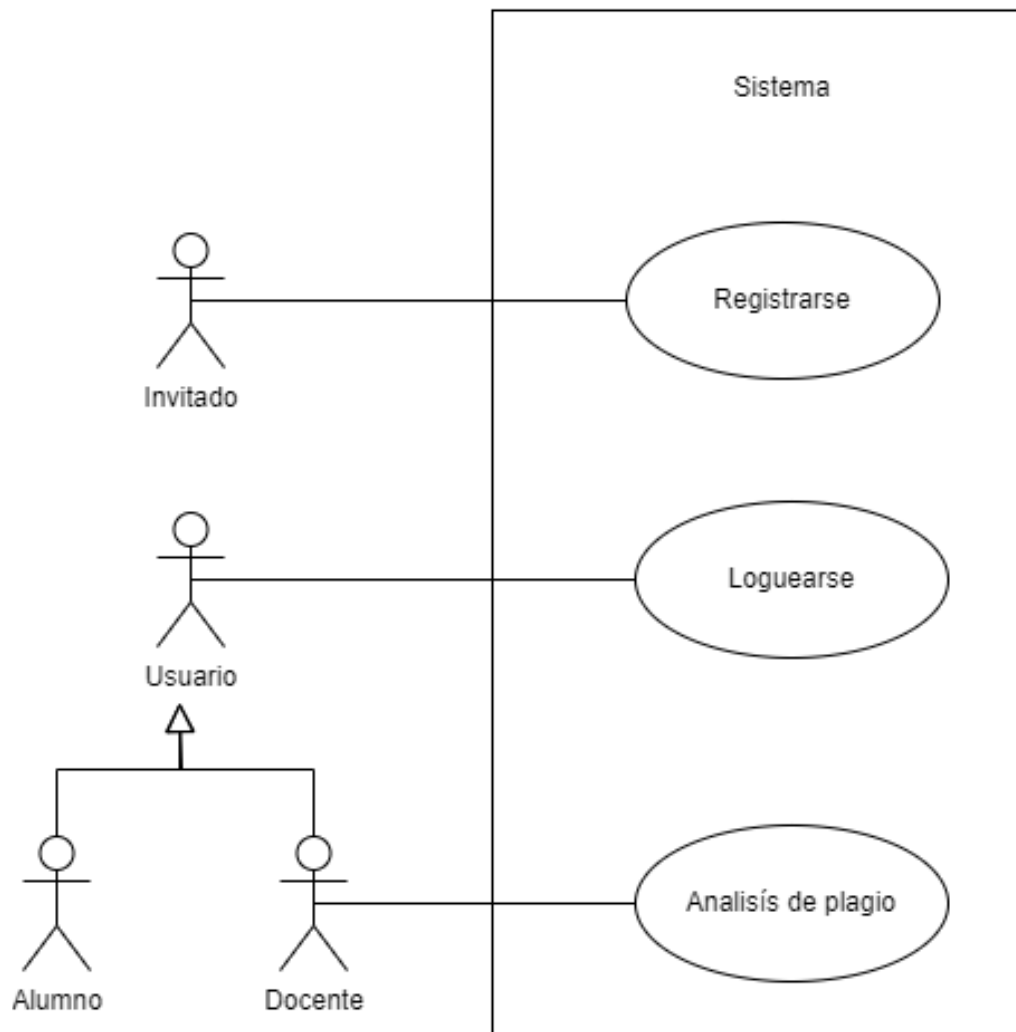
**Docker:** Proyecto de código abierto que automatiza el despliegue de aplicaciones dentro de contenedores de software, proporcionando una capa adicional de abstracción y automatización de virtualización de aplicaciones en múltiples sistemas operativos.

#### 1.4 Visión general

El documento presenta la arquitectura a través de diversas vistas que son descritas en cada una de las siguientes secciones.

## 2. Vista de escenarios

### 2.1 Diagrama



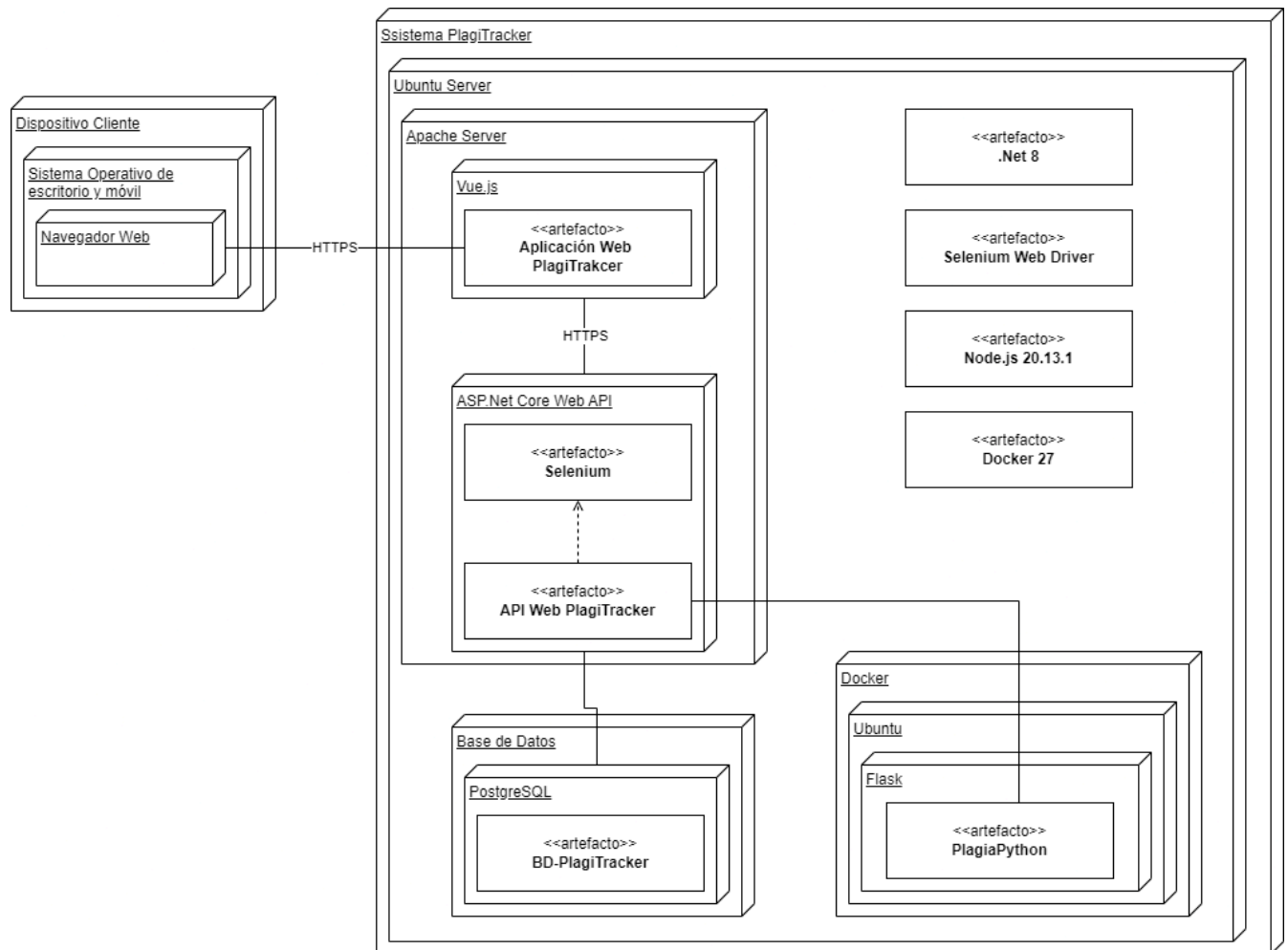
### 2.2 Descripción de la Vista

- **Análisis de plagio:** El caso de uso describe el análisis de plagio que se inicia cuando los alumnos ingresan las URLs de la página "Codiva" en un campo específico de la actividad. Al finalizar el tiempo de la actividad, el sistema habilita un botón 'Analizar' para el docente. Al hacer clic en este botón, se inicia el análisis de las URLs, generando un informe detallado en formato PDF. Una vez completado el análisis, se envía una notificación al docente informando que el informe está listo para ser descargado.

- **Loguearse en el sistema:** El caso de uso describe el proceso mediante el cual un usuario se autentica en el sistema de detección de plagio para acceder a las funcionalidades del sistema, como la visualización de reportes de plagio.
- **Registrarse en el sistema:** El caso de uso describe el proceso donde el docente y el alumno se registran en el sistema de detección de plagio. El objetivo es permitir al profesor crear una cuenta para acceder a las funcionalidades del sistema, como la detección de similitudes en código fuente y al alumno le permite la visualización de los cursos y asignaciones.

### 3. Vista física

#### 3.1 Diagrama



#### 3.2 Descripción de la Vista

##### Dispositivo Cliente (usuario):

Puede ser una computadora de escritorio, una laptop o incluso un dispositivo móvil. No hay limitaciones con el Sistema Operativo dado que se trata de una aplicación web.

PlagiTracker
Documento de Arquitectura de Software

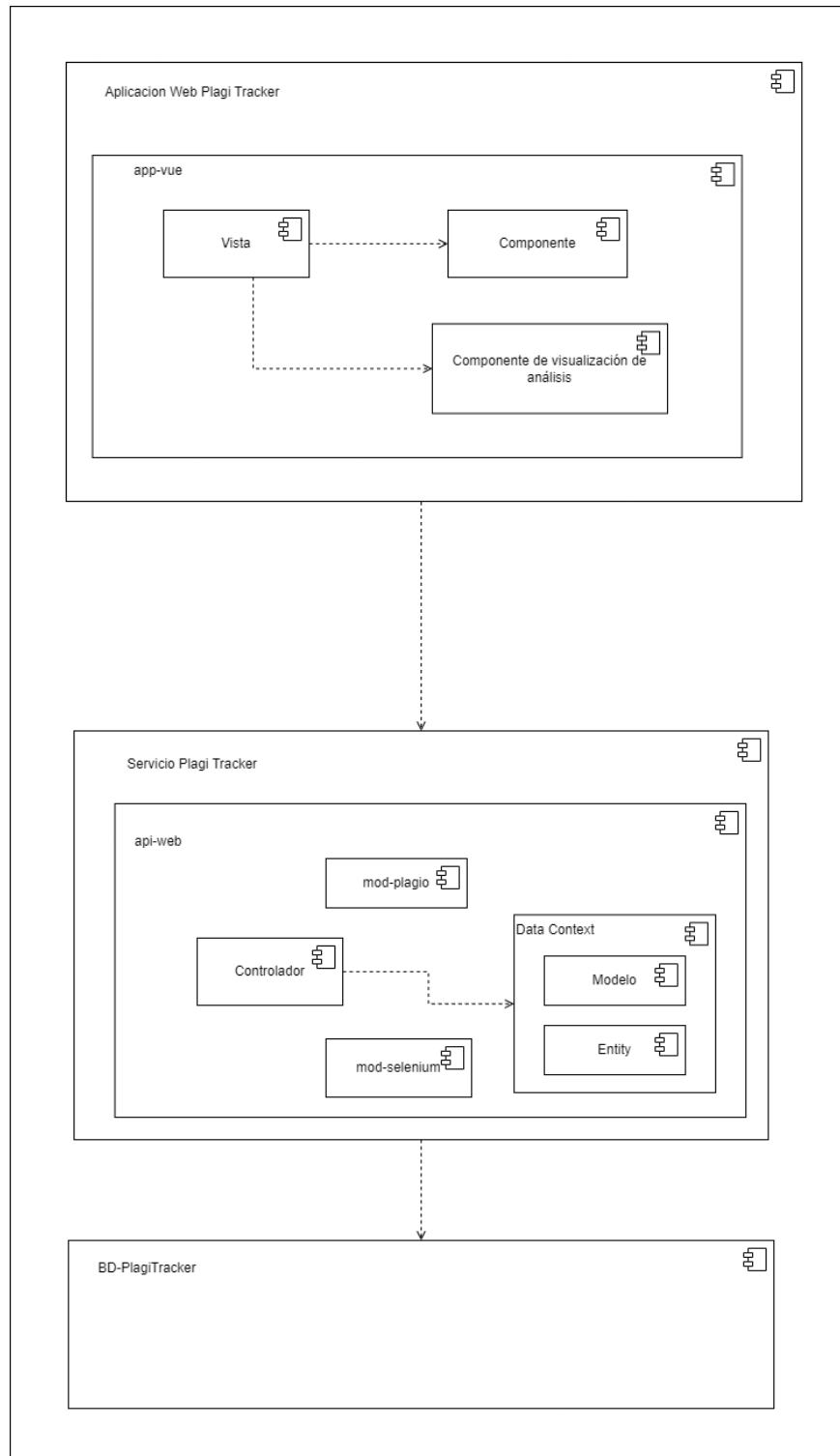
### Sistema PlagiTracker:

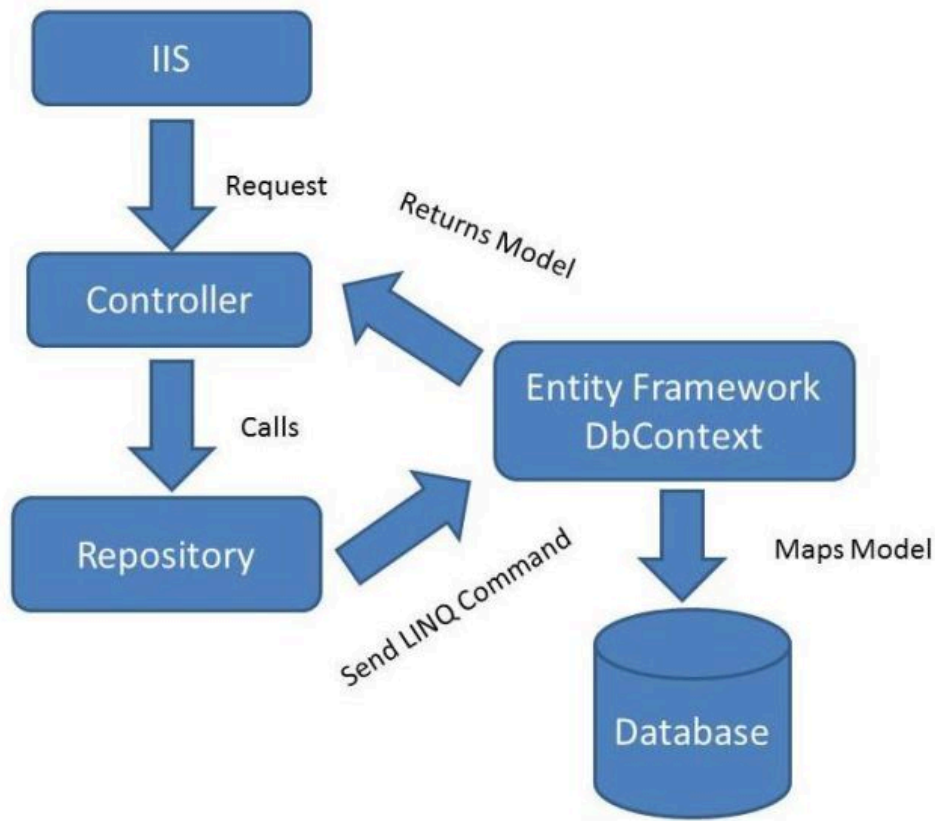
Está desplegado en el sistema operativo Ubuntu Server, usa un servidor web Apache para alojar la aplicación web y la API web. Para que pueda desplegar la aplicación web se necesita Node.js 20.13.1 para desplegar la API web necesita .Net 8 y Selenium Web Driver.

- La **Aplicación Web** está hecha en Vue.js, que se comunica de forma bidireccional con el cliente por medio del protocolo HTTPS.
- La **API Web** está hecha en ASP.Net Core, que a su vez, usa Selenium para el web scraping. La API Web se comunica con la aplicación web por medio del protocolo HTTPS.
- La **Base de Datos**, que también está dentro del Ubuntu Server, usa PostgreSQL y se comunica únicamente con la API Web.

#### 4. Vista de desarrollo

##### Diagrama





#### 4.1 Descripción de la Vista

##### Aplicación web PlagiTracker (Interfaz de usuario):

La capa superior muestra el módulo de la interfaz de usuario, donde se utiliza un framework frontend llamado app-vue.

Dentro de este módulo hay dos componentes principales:

- **Vista:** Es la interfaz con la que interactúan los usuarios para realizar acciones en el sistema.
- **Componente:** Este parece ser un componente reutilizable o parte modular del frontend.

##### Servicio PlagiTracker (Lógica del Negocio):

En el centro del diagrama está el módulo api-web, que representa el servicio backend del sistema.

Los elementos incluyen:

- **Controlador:** Gestiona las solicitudes que provienen de la interfaz de usuario, dirigiéndolas al módulo correcto.
- **Entity-Framework:** Es un intermediario que realiza el mapeo entre relaciones del sistema.
- **Modelo:** Representa la lógica de datos del sistema, probablemente gestionando la interacción



PlagiTracker
Documento de Arquitectura de Software

con la base de datos.

- mod-plagio: Módulo encargado del análisis de plagio.
- mod-selenium: Indica que se está utilizando Selenium para realizar pruebas automatizadas o la verificación de contenido.

### **BD-PlagiTracker (Base de Datos):**

En la parte inferior se encuentra la base de datos, representando el almacenamiento de los datos relevantes para el sistema.