

# **CASOS DE USO**

## **[CodeGuardGT]**

**Versión 1.0**

# Documento de Arquitectura de Software

## 1. Introducción

### 1.1 Propósito

Este documento proporciona una visión general arquitectónica del sistema **CodeGuardGT**, diseñado para la detección de plagio en código SQL. El propósito es presentar una serie de vistas arquitectónicas que abordan diferentes aspectos del sistema. Esto permitirá capturar y establecer las decisiones arquitectónicas significativas tomadas durante el desarrollo, garantizando que el sistema cumpla con sus requisitos funcionales y no funcionales.

### 1.2 Alcance

El alcance de este documento incluye:

- La presentación de la arquitectura del sistema **CodeGuardGT**.
- La descripción de las decisiones arquitectónicas clave.
- La explicación de las vistas arquitectónicas que ilustran cómo el sistema está estructurado y cómo interactúan sus componentes.

### 1.3 Definiciones, Acrónimos, y Abreviaturas

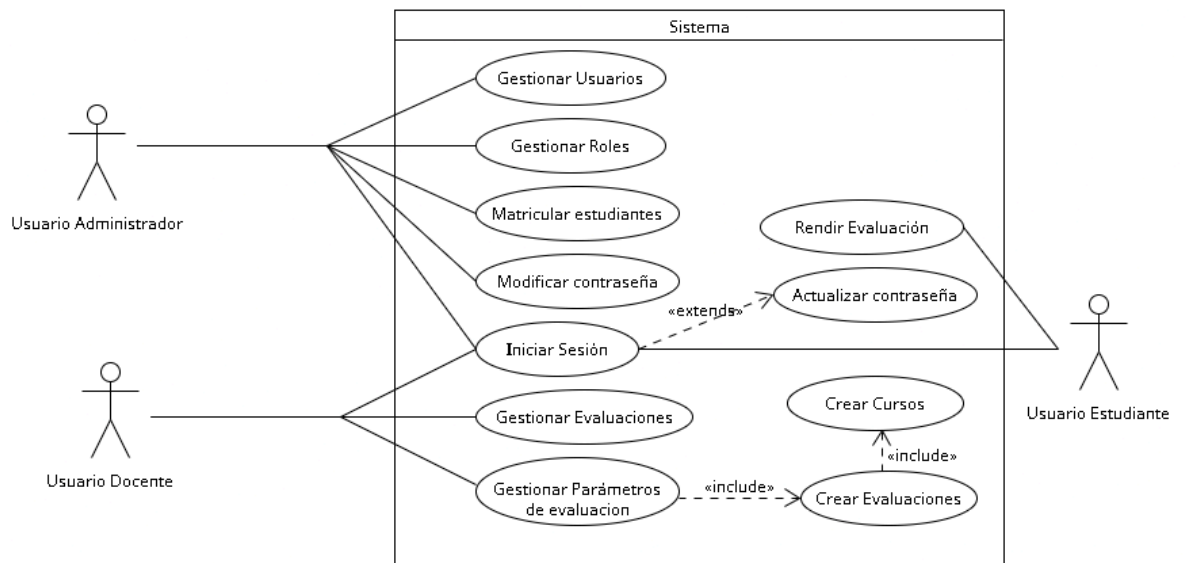
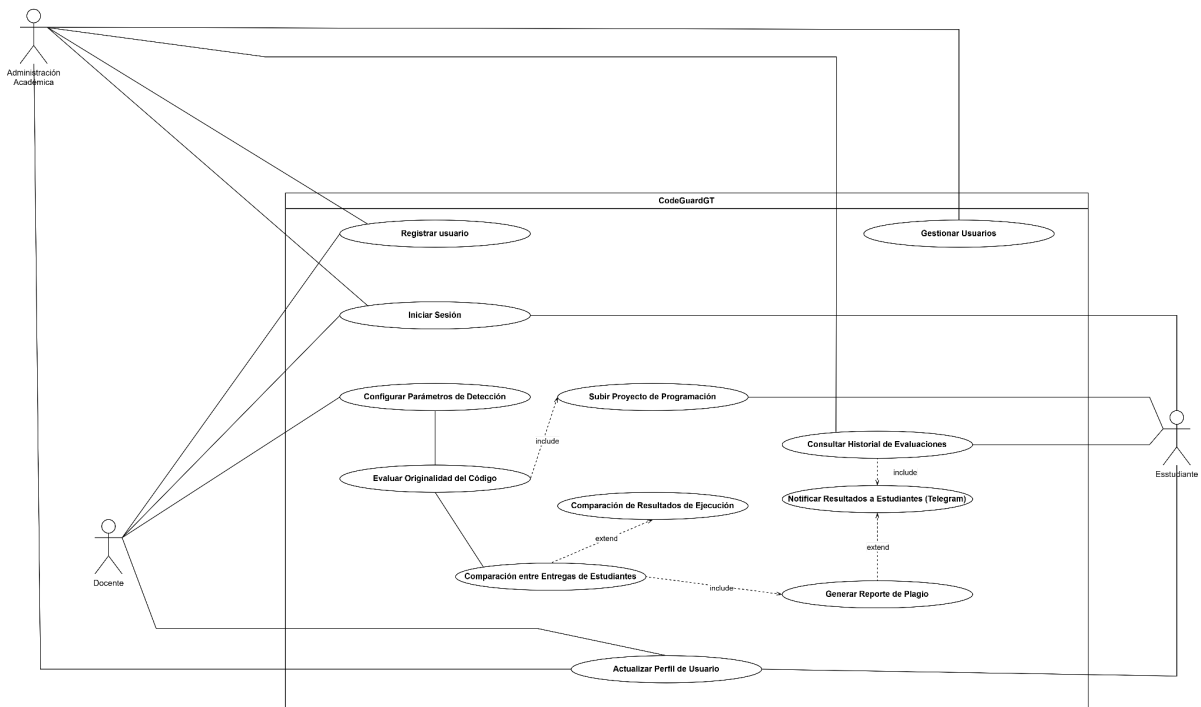
- **CodeGuardGT**: Sistema de detección de plagio en código SQL.
- **SQL**: Structured Query Language (Lenguaje de Consulta Estructurado).
- **SQLFiddle**: Plataforma online para ejecutar y comparar código SQL.
- **GDPR**: General Data Protection Regulation (Reglamento General de Protección de Datos).
- **SSL**: Secure Sockets Layer (Capa de Conexión Segura).

### 1.4 Visión general

El documento presenta la arquitectura del sistema a través de diversas vistas que son descritas en las siguientes secciones. Estas vistas proporcionan una visión integral de cómo está estructurado **CodeGuardGT** y cómo se espera que funcione en el entorno de producción.

## 2. Vista de escenarios

### 2.1 Diagrama



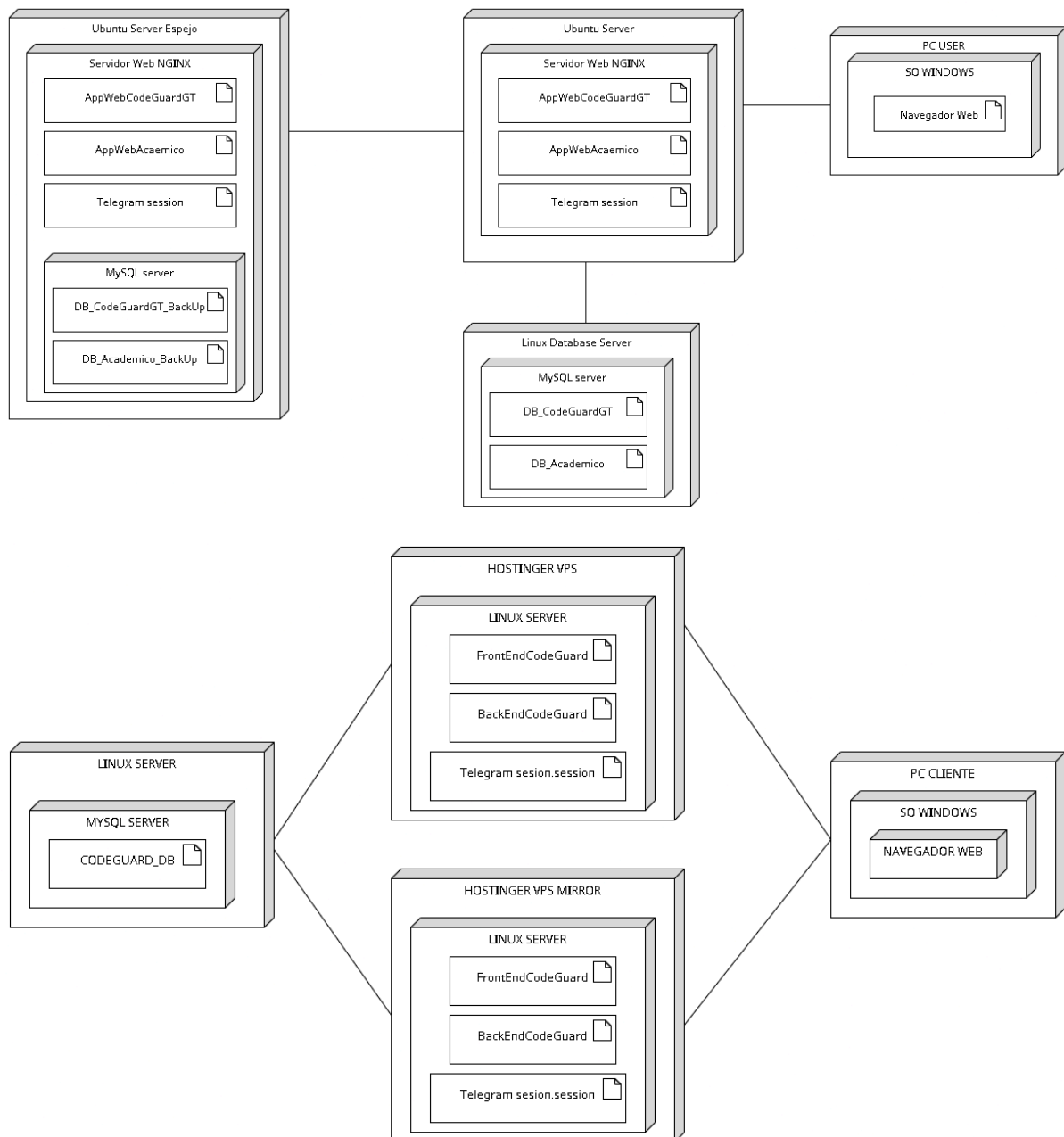
### 2.2

### 2.3 Descripción de la Vista:

1. **Actores:** Los actores son representados por figuras de palitos. Estos pueden ser usuarios o sistemas externos que interactúan con el sistema. En tu diagrama, veo varios actores, como “Usuario”, “Administrador”, y “Sistema Externo”.
2. **Casos de Uso:** Los casos de uso están representados por óvalos y describen las funcionalidades que el sistema proporciona a los actores. Algunos ejemplos en tu diagrama incluyen “Iniciar Sesión”, “Registrar Usuario”, y “Generar Reporte”.
3. **Relaciones:** Las líneas que conectan los actores con los casos de uso indican quién puede realizar qué acciones. Por ejemplo, el “Usuario” está conectado a “Iniciar Sesión” y “Registrar Usuario”, lo que significa que el usuario puede realizar estas acciones.
4. **Sistema:** El rectángulo que engloba los casos de uso representa el sistema en sí. Todo lo que está dentro del rectángulo es parte del sistema que se está modelando.
5. **Extensiones y Inclusiones:** A veces, los casos de uso pueden tener relaciones especiales como “extiende” o “incluye”. Estas relaciones se utilizan para mostrar que un caso de uso puede ser parte de otro o que puede extender su funcionalidad. En tu diagrama, veo algunas de estas relaciones, como “Iniciar Sesión” que incluye “Validar Usuario”.

### 3. Vista física

#### 3.1 Diagrama



#### 3.2 Descripción de la Vista

En esta vista física se observa una infraestructura de red compuesta por varios servidores y un usuario conectado desde una PC con sistema operativo Windows. El usuario accede a las aplicaciones web mediante un navegador, el cual se comunica con un servidor principal basado en Ubuntu.

El servidor Ubuntu tiene configurado un servidor web NGINX que hospeda varias aplicaciones: AppWebCodeGuardGT, AppWebAcademico y una sesión activa de Telegram. Estas aplicaciones están vinculadas a una base de datos MySQL, la cual maneja respaldos de dos bases de datos principales: DB\_CodeGuardGT\_BackUp y DB\_Academico\_BackUp.

Junto al servidor principal de Ubuntu, existe un servidor espejo, también basado en Ubuntu, que replica las mismas aplicaciones web y el servidor de MySQL, asegurando la

disponibilidad de las bases de datos de respaldo DB\_CodeGuardGT\_BackUp y DB\_Academico\_BackUp.

Por otro lado, un servidor dedicado exclusivamente a las bases de datos en Linux maneja las versiones primarias de DB\_CodeGuardGT y DB\_Academico. Todas las aplicaciones web en los servidores Ubuntu, tanto el principal como el espejo, interactúan con este servidor de base de datos, garantizando el acceso a la información de forma centralizada.

#### **4. Vista de desarrollo**

##### **4.1 Diagrama**

##### **4.2 Descripción de la Vista**

