

---

**Universidad La Salle**

---

**CodeGuardGT**  
**Caso de Uso: Comparación entre Entregas de Estudiantes**  
**con Algoritmo TF-IDF**  
**Versión:2.2**

CodeGuardGT
Caso de Uso: Comparación entre Entregas de Estudiantes con Algoritmo TF-IDF

## Caso de Uso: Comparación entre Entregas de Estudiantes con Algoritmo TF-IDF

### 1. Comparación entre Entregas de Estudiantes con Algoritmo TF-IDF

#### 1.1 Breve descripción del caso de uso

Este caso de uso describe cómo un docente utiliza el sistema para revisar evaluaciones estudiantiles, enfocándose en detectar plagio en códigos SQL entregados. El docente accede a las evaluaciones, analiza similitudes en los documentos SQL y genera un informe de plagio.

#### 2. Definiciones, Acrónimos, y Abreviaturas

- **CodeGuardGT:** Aplicativo para la detección de plagio en código SQL.
- **SQLAlchemy:** Biblioteca de Python.
- **Selenium:** Herramienta de automatización de navegadores web.
- **Caso de uso:** Descripción de cómo los usuarios interactúan con el sistema para lograr un objetivo.
- **MODAL:** mensaje emergente que retorna la aplicación, sea favorable o errónea.
- **Contraseña:** Palabra “Contraseña” en el aplicativo.
- **Api:** Una API (Interfaz de Programación de Aplicaciones) es un conjunto de reglas y protocolos que permite que diferentes aplicaciones se comuniquen entre sí, facilitando el intercambio de datos y funciones.
- **Botones del sistema:** Entre comillas.
- **Mensajes del sistema:** En negrita.
- **Mensaje satisfactorio:** Verde.
- **Mensaje de error:** Rojo.

### 3. Flujo de Eventos

#### 3.1 Flujo Básico de Comparación entre Entregas de Estudiantes con Algoritmo TF-IDF

ACTOR	SISTEMA
1. El “Usuario Docente” ingresa al sistema.	2. El sistema carga la página de LOGIN cargando un formulario donde solicitará DNI y contraseña.
3. El “Usuario Docente” ingresa sus credenciales.	
4. El “Usuario Docente” hace clic en el botón “Iniciar Sesión”.	5. El sistema verifica las credenciales ingresadas.
	6. El sistema redirecciona a la pantalla principal del “Usuario

	Docente”.
7. El “Usuario Docente” va al menú lateral izquierdo, despliega el apartado “Gestión Evaluaciones” y selecciona el apartado de “Gestión de Evaluaciones”	8. El sistema muestra la pantalla de “Gestión de Evaluaciones” presenta una cartilla principal de <b>Evaluaciones del Docente</b> donde se muestran los cursos y sus respectivas evaluaciones relacionadas con el docente.
9. El “Usuario Docente” hace clic en el botón “Realizar Revisión” de la evaluación que desea revisar	10. El sistema presenta un modal <b>Seleccione el Método de Revisión</b> que permitirá seleccionar uno de los diferentes métodos de revisión de plagio, un amplificador de umbral para medir de 0 a 100 el porcentaje de evaluación de plagio, un espacio para llenar el correo electrónico donde se enviará el reporte y un botón de “Cancelar” y “Aplicar Revisión”.
11. El “Usuario Docente” hace clic en el botón “Aplicar Revisión”.	12. El sistema reúne las entregas de los estudiantes, usando Selenium y ChromeDriver para descargar datos si es necesario. Limpia el contenido eliminando palabras irrelevantes y separa el texto en palabras clave.
	13. Calcula la frecuencia de cada palabra en cada entrega (TF). Calcula la importancia de cada palabra en todas las entregas (IDF) y combina los valores de TF e IDF para cada documento.

CodeGuardGT
Caso de Uso: Comparación entre Entregas de Estudiantes con Algoritmo TF-IDF

	14. Compara las entregas utilizando una fórmula que mide similitudes entre documentos. Identifica posibles casos de plagio si la similitud supera un porcentaje definido.
	15. Genera un reporte con los resultados y lo envía al correo del docente.
	16. El sistema muestra un modal con el mensaje <b>La revisión de códigos SQL utilizando el método TF-IDF se completó con éxito. El reporte ha sido generado en formato PDF y enviado correctamente al correo asociado.</b> y un botón “Cerrar”.
17. El “Usuario Docente” visualiza y hace clic en el botón de “Cerrar”.	

#### 4. Flujos Alternativos

##### 4.1 Flujo alternativo 1: *Una entrega o no hay entregas para revisión (paso 11)*

ACTOR	SISTEMA
1. Desde el punto 11 del flujo básico.	2. El sistema presenta un aviso de error <b>Debe haber al menos dos códigos entregados a la evaluación. Count: 1</b> y un botón de “Cerrar”
4. El “Usuario Docente” hace clic en el botón “Cerrar”.	
	3. Regresa al punto 11 del flujo básico.

CodeGuardGT
Caso de Uso: Comparación entre Entregas de Estudiantes con Algoritmo TF-IDF

#### 4.2 Flujo alternativo 2: *Error al comunicarse con el servidor (paso 11)*

ACTOR	SISTEMA
1. Desde el punto 11 del flujo básico.	
	2. El sistema presenta un aviso de error <b>Error al comunicarse con el servidor.</b> y un botón de “Cerrar”
4. El “Usuario Docente” hace clic en el botón “Cerrar”.	
	4. Regresa al punto 10 del flujo básico.

CodeGuardGT
Caso de Uso: Comparación entre Entregas de Estudiantes con Algoritmo TF-IDF

## 5. Requisitos Especiales

- **Rendimiento:** CodeGuardGT debe ser capaz de comparar hasta 20 entregas de código SQL en menos de 5 minutos para garantizar la eficiencia del proceso de revisión.
- **Seguridad:** El análisis debe realizarse garantizando la confidencialidad e integridad de los datos, evitando accesos no autorizados o modificaciones a las entregas de los estudiantes.
- **Conexión a Telegram:** El sistema debe establecer una conexión segura con la API de Telegram para enviar notificaciones sin interrupciones, asegurando la correcta entrega del reporte de evaluación.
- **Integridad de Datos:** Las comparaciones de plagio y las consultas a la base de datos deben realizarse sin comprometer la integridad de los datos de los estudiantes ni la confidencialidad de la información.

## 6. Precondiciones

- Las entregas de los estudiantes deben estar completas y accesibles en la plataforma donde se gestionan (por ejemplo, un LMS o base de datos).
- CodeGuardGT debe contar con acceso a dichas entregas para realizar el análisis.
- El “Usuario Docente” debe haber iniciado sesión con credenciales válidas.
- El sistema debe tener acceso a las APIs correspondientes, tanto la API interna para la detección de plagio como la API externa de Telegram para notificaciones.
- El sistema debe estar correctamente configurado para realizar consultas a las bases de datos (tablas curso, evaluación, documentos) mediante SQLAlchemy y acceso a Selenium para la extracción de datos.
- El “Usuario Docente” debe tener una conexión a internet estable (se recomienda tener al menos una velocidad de internet de 1 Mbps; sin embargo, para una experiencia más fluida y sin interrupciones, una velocidad de 3 a 5 Mbps sería lo ideal) para acceder al aplicativo web.

## 7. Postcondiciones

- Cualquier entrega que haya sido omitida por problemas técnicos o de acceso es claramente indicada en el informe, junto con la razón de la omisión.
- Un informe detallado sobre la similitud entre los documentos SQL entregados por los estudiantes debe ser generado y accesible para el “Usuario Docente”, ya sea por la plataforma o por un canal alternativo (correo, descarga directa).
- Cualquier error en el proceso de generación del informe o el envío de la notificación debe ser registrado en el aplicativo web, y el “Usuario Docente” debe ser informado con opciones para reintentar.