

---

**Universidad La Salle**

---

**CodeGuardGT**

**Caso de Uso: Evaluar Originalidad del Código**

# Caso de Uso: Evaluar Originalidad del Código

## 1. Evaluar Originalidad del Código

### 1.1 Breve descripción del caso de uso

Este caso de uso se centra en verificar que el código desarrollado sea original y no una copia de otros trabajos. Es crucial para asegurar la integridad y la propiedad intelectual del proyecto.

## 2. Flujo de Eventos

### 2.1 Flujo Básico de Inicio de CodeGuardGT

ACTOR	SISTEMA
1. El docente o administrador abre la aplicación de CodeGuardGT.	2. CodeGuardGT carga la pantalla de inicio - "login con Google".
3. El usuario proporciona sus credenciales (usuario y contraseña) en la ventana de google auth a la que será redireccionado.	4. Google valida las credenciales y redirecciona a CodeGuardGT con los datos de la cuenta.
	5. CodeGuardGT carga las funcionalidades y herramientas según el rol del usuario.
6. El usuario selecciona la opción de "Configuración" en el menú principal para definir parámetros específicos del análisis (por ejemplo, periodo de entrega, tipo de análisis, criterios de detección).	7. CodeGuardGT muestra la pantalla de configuración y permite al usuario realizar ajustes.
8.	9. El sistema guarda las configuraciones y vuelve a la pantalla principal, dejando el sistema listo para ejecutar análisis.

### 2.2 Flujo Básico de Cierre de CodeGuardGT

ACTOR	SISTEMA
1. El usuario selecciona la opción "Cerrar Sesión" en el menú principal.	2. CodeGuardGT pregunta al usuario si desea guardar cualquier configuración o trabajo no guardado.
3. El usuario confirma la acción de cierre o guarda los cambios pendientes.	4. CodeGuardGT guarda los datos (si corresponde) y finaliza cualquier proceso en ejecución.
	5. CodeGuardGT cierra la sesión del usuario y redirige a la pantalla de

Universidad La Salle
Caso de Uso: Evaluar Originalidad del Código

	inicio de sesión.
	6. El sistema desconecta cualquier conexión activa con plataformas como SQLFiddle.
7. El usuario cierra la aplicación.	8. CodeGuardGT finaliza todos los procesos en segundo plano y libera los recursos del sistema.

## 2.3 Flujo Básico

ACTOR	SISTEMA
1. El estudiante sube el código al sistema.	2. El sistema recibe el código y lo almacena temporalmente.
	3. El sistema ejecuta un algoritmo de detección de plagio.
	4. El sistema compara el código subido con una base de datos de códigos existentes.
	5. El sistema genera un reporte de originalidad.
	6. El sistema notifica al docente sobre los resultados del análisis.

## 2.4 Flujos Alternativos

### 2.4.1 Primer Flujo Alternativo: Enlace no válido.

ACTOR	SISTEMA
1. El estudiante sube el link de SQLFiddle al sistema..	2. El sistema detecta que el enlace subido no es válido.
	3. El sistema notifica al estudiante sobre el error y solicita que suba el archivo nuevamente.
4. El estudiante vuelve a subir el link de SQLFiddle al sistema.	

### 2.4.2 Segundo Flujo Alternativo: El alumno solicita una revisión.

ACTOR	SISTEMA
1. El estudiante sube el código al sistema.	2. El sistema detecta que el código subido es idéntico a uno existente en la base de datos.
	3. El sistema notifica al estudiante sobre el posible plagio y solicita una revisión.
4. El estudiante solicita una revisión adicional de su entrega.	

## 3. Requisitos Especiales

### 3.1 Primer requerimiento especial:

- El sistema debe ser capaz de manejar archivos de código de SQLFiddle.
- El algoritmo de detección de plagio debe ser actualizado regularmente para mejorar su precisión.
- El sistema debe cumplir con las normativas de privacidad y protección de datos.

## 4. Precondiciones

### 4.1 Primera precondición

- El estudiante debe estar registrado en el sistema y tener acceso autorizado.
- El sistema debe tener acceso a una base de datos actualizada de códigos existentes para realizar la comparación.

## 5. Postcondiciones

### 5.1 Primera postcondición

- El estudiante recibe un reporte de originalidad del código.
- El sistema almacena los resultados del análisis para futuras referencias.