
Universidad La Salle

CodeGuardGT

Caso de Uso: Comparación entre Entregas de Estudiantes

Caso de Uso: Comparación entre Entregas de Estudiantes

1. Comparación entre Entregas de Estudiantes

1.1 Breve descripción del caso de uso

Este caso de uso describe el proceso mediante el cual un docente utiliza el sistema para revisar evaluaciones de estudiantes, enfocándose en la detección de plagio entre los códigos SQL entregados. El sistema permite al docente acceder a las evaluaciones disponibles, extraer los documentos SQL de los estudiantes, analizar similitudes entre ellos, y generar un informe de plagio. Además, el sistema ofrece la opción de enviar una notificación por Telegram con el informe generado. En caso de fallos durante el proceso de generación del informe o envío de la notificación, el sistema ofrece métodos alternativos como reintentos.

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.

3. Flujo de Eventos

3.1 Flujo Básico de Comparación entre Entregas de Estudiantes

ACTOR	SISTEMA
1. El "Usuario Docente" ingresa al aplicativo web.	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" despliega el menú lateral izquierdo y selecciona el apartado “Revisión de Evaluaciones”.	8. El sistema muestra el apartado de revisión de evaluaciones, cargando las evaluaciones relacionadas con el "Usuario Docente” logueado.
9. El "Usuario Docente" hace clic en el botón “Revisar” en la evaluación que desea revisar.	10. El sistema presenta una vista de carga mientras se inicia el proceso de revisión.

CodeGuardGT
Caso de Uso: Comparación entre Entregas de Estudiantes

	11. El sistema realiza una consulta interna utilizando Consulta la tabla cursor para obtener la información del curso asociado. Consulta la tabla evaluación para obtener la evaluación seleccionada. Consulta la tabla documentos para obtener los documentos entregados por los estudiantes. Comunicándose mediante query.
	12. El sistema utiliza Selenium WebDriver y Chromedriver para extraer los códigos SQL de las URLs asociadas a los estudiantes en la evaluación seleccionada.
	13. El sistema consume la API de detección de plagio, enviando los documentos obtenidos para analizar las similitudes entre ellos.
	14. La API devuelve la respuesta de satisfacción de la evaluación indicando que el análisis de plagio se ha completado con éxito.
	15. El sistema muestra un modal con el mensaje de evaluación realizada correctamente y los detalles del análisis de plagio.
	16. El sistema genera un reporte con los documentos (utiliza la url para extraer el código SQL) y el porcentaje de similitud entre ellos, indicando cualquier plagio detectado.
	17. El sistema carga un botón de notificación por Telegram para enviar el reporte al "Usuario Docente".
18. El "Usuario Docente" hace clic en el botón de "Notificación".	19. El sistema consume la API de Telegram y utiliza un bot para enviar el reporte al "Usuario Docente" por el chat de Telegram.

CodeGuardGT
Caso de Uso: Comparación entre Entregas de Estudiantes

	20. El sistema muestra un modal confirmando que el reporte ha sido enviado correctamente por Telegram.
--	--

3.2 Flujos Alternativos

3.2.1 Flujo alternativo 1: No Hay Registro de Estudiantes o Evaluaciones (paso 8)

ACTOR	SISTEMA
	1. Desde el punto 8 del flujo básico.
	2. El sistema no muestra evaluaciones en la tabla de revisión.
	3. El sistema muestra una opción para generar nuevas evaluaciones.

3.2.2 Flujo alternativo 2: No Se Genera el Informe de Plagio (paso 18)

ACTOR	SISTEMA
1. Desde el punto 18 del flujo básico.	2. El sistema intenta generar el informe, pero ocurre un error (por ejemplo, fallo en la API de detección de plagio o problema de red).
	3. El sistema muestra un mensaje de error en un modal indicando que no se ha podido generar el informe.
4. El "Usuario Docente" intenta realizar nuevamente la acción o actualiza la página.	5. El sistema permite al "Usuario Docente" reintentar la generación del informe después de verificar si el problema persiste.
	6. Si el problema continúa, el sistema sugiere contactar al soporte técnico mediante un modal.

3.2.3 Flujo alternativo 3: No Llega el Mensaje a Telegram (paso 18)

ACTOR	SISTEMA
-------	---------

CodeGuardGT
Caso de Uso: Comparación entre Entregas de Estudiantes

1. Desde el punto 18 del flujo básico.	2. El sistema intenta enviar el mensaje, pero ocurre un fallo en la conexión con la API de Telegram o el bot no responde.
	3. El sistema muestra un mensaje de error indicando que el mensaje no ha sido enviado correctamente a Telegram.
	4. El sistema muestra los detalles del análisis de plagio.

4. 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.

CodeGuardGT
Caso de Uso: Comparación entre Entregas de Estudiantes

5. 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.

6. 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.