INTEGRICODE

Plan de Iteración 2

Plan de Iteración 2

# Introducción

## Alcance

El presente documento aborda la segunda iteración del proyecto INTEGRICODE, un sistema de detección de plagio que se basa en la revisión de enlaces generados por un editor de código en línea. El sistema permite que los estudiantes suban su código fuente a través de estos enlaces, los cuales serán procesados por el sistema para verificar la originalidad del contenido y detectar posibles casos de plagio.

## Definiciones, acrónimos, y abreviaturas

* INTEGRICODE: Sistema de detección de plagio que analiza códigos fuente enviados mediante enlaces generados por el editor “online-java.com”.
* Plagio: Acción de copiar o imitar el trabajo de otra persona, presentándose como propio, sin dar crédito al autor original.
* Enlace de código (URL): Dirección de un recurso en la web que redirige a un archivo de código fuente enviado para su verificación.
* Similitud: Porcentaje de coincidencia entre dos o más códigos fuente, utilizado para determinar la posible existencia de plagio.
* Usuario: Puede ser el docente o el estudiante, se usa para generalizarse en ocasiones en donde se requiera hablar de ambos.

# Casos de Uso

El listado de casos de uso a implementar y probar en esta iteración es el siguiente:

1. Caso de uso: Generar Reporte de Similitud.
2. Caso de uso: Gestionar Perfil de Usuario.

# Recursos

| **Persona** | **Roles** |
| --- | --- |
| José Alberto Rondón Torres | Líder del Proyecto, Supervisor de calidad, Programador backend, analista. |
| Aaron Adhel Zela Rojas | Supervisor de calidad, Programador frontend, analista. |
| Patrick Jefferson Gonzales Fernandez | Programador frontend, analista. |
| Kevin Dietmar Gonzalez Fernandez | Programador frontend, analista. |
| Guillermo Valentin Ampuero Razuri | Programador backend, analista. |
| Yoel Ccorihuaman Guillen | Programador backend, analista. |
| Diego Fersen Ali Zirena | Programador backend, analista. |

| **Entidades de Data** | **Descripción** |
| --- | --- |
| Roles | Define los diferentes roles dentro del sistema, con un identificador único (ID\_rol), un nombre único (nombre) y una descripción opcional (descripcion), describiendo las funciones de cada rol. |
| Usuarios | Almacena la información de los usuarios, incluyendo su ID único (ID\_usuario), nombres, apellidos, correo (único), contraseña y su rol (ID\_rol), además de su estado y las fechas de creación y actualización. |
| RefreshTokens | Guarda los tokens de actualización para los usuarios con su fecha de expiración y creación, permitiendo la gestión de sesiones seguras. |
| Permisos | Contiene los permisos del sistema con un ID único (ID\_permiso), un nombre único y una descripción que detalla lo que permite hacer cada permiso. |
| Roles\_Permisos | Relaciona los roles con los permisos, permitiendo asignar múltiples permisos a cada rol mediante los campos ID\_rol y ID\_permiso. |
| Cursos | Registra los cursos ofrecidos, con un ID único (ID\_curso), nombre del curso, descripción, código de acceso único, estado, y las fechas de creación y actualización. |
| Usuarios\_Cursos | Relaciona a los usuarios con los cursos en los que están inscritos, indicando el estado de la inscripción y las fechas de inscripción y baja. |
| Actividades | Almacena las actividades asignadas a los cursos, con un ID único (ID\_actividad), nombre, descripción, fechas de inicio y entrega, estado, y fechas de creación y actualización. |
| Codigos | Contiene los códigos fuente enviados por los usuarios para actividades específicas, incluyendo el enlace y la ruta del código, el estado, y las fechas de subida y actualización. |
| Analisis | Registra los análisis de plagio, con un ID único, actividad asociada, estado del análisis, fechas de inicio y fin, y configuración en formato JSON. |
| ResultadosAnalisis | Almacena los resultados generados de los análisis de código, indicando el estado y la fecha de generación. |
| ReportesPlagios | Registra los reportes de plagio entre códigos, con un porcentaje de similitud, detalles, comentarios, el revisor, y el estado de revisión. |
| LogAnalisis | Almacena un registro de eventos del análisis de código, con el tipo de evento, descripción y la fecha en que ocurrió. |
| Componentes | Representa los componentes de la aplicación, incluyendo la ruta y la relación jerárquica entre componentes. |
| Permisos\_Componentes | Relaciona los permisos con los componentes específicos a los que aplican. |
| RecuperacionContrasena | Guarda las solicitudes de recuperación de contraseña para los usuarios, con la fecha de solicitud. |

| **Otros (hardware, ambientes)** | **Descripción** |
| --- | --- |
| Ninguno | Ninguno |

# Plan

| **Actividad** | **Fecha de culminación** |
| --- | --- |
| Implementación de casos de uso | 2024-10-16 |
| Realización de pruebas | 2024-11-06 |
| Ciclo 1: Corrección de defectos / Implementación de cambios | 2024-11-07 |
| Ciclo 2: Realización de pruebas | 2024-11-27 |

# Entregables

| **Artefactos** |
| --- |
| Diagrama de secuencias:   * Flujo básico: Generar Reporte de Similitud   + Flujo alternativo 2.2.1: Selecciona Generar reporte con PDF.   + Flujo alternativo 2.2.2: Selecciona Generar reporte con Robot(IA).   + Flujo alternativo 2.2.3: No se entregaron tareas antes de generar un reporte.   + Flujo alternativo 2.2.4: Descargar reporte. * Flujo básico: Gestionar perfil de usuario   + Flujo alternativo 2.2.1: Cambiar Contraseña.   + Flujo alternativo 2.2.2: Subir Foto de Perfil.   + Flujo alternativo 2.2.3: Subir Foto en formato incorrecto.   + Flujo alternativo 2.2.4: Cancelar Editar Contraseña.   + Flujo alternativo 2.2.5: Eliminar Cuenta.   + Flujo alternativo 2.2.6: Cancelar Eliminar Cuenta. |
| Modelos de datos:   * Generar Reporte de Similitud. * Gestionar Perfil de Usuario. |
| Ejecutables:   * plagio * plagio\_api |

# Criterios de Evaluación

Se deben cumplir los siguientes criterios de evaluación:

1. Cumplir con las pruebas de sistema realizadas para esta iteración.
2. Cumplir con las pruebas unitarias desarrolladas para esta iteración. Tales pruebas deben ser ejecutadas desde una suite de pruebas.
3. Pasar las revisiones externas realizadas por otros equipos.

# Supuestos y restricciones

* Cantidad de estudiantes para enlistarse en un curso(49).
* No se pueden crear actividades que sean de fechas pasadas.
* Necesita un API Key de Gemini para poder generar el reporte de IA.
* El scrapping demora un poco más cuando los códigos son extensos.