

Reglas Entrega Final (Proyecto).

Elaborado por: Mg. Jeisson Ibarguen Maturana

Objetivos

1. Evaluar la implementación de reportes funcionales y la interfaz CRUD del sistema de gestión Clínica.
2. Fomentar la colaboración en la integración del frontend y backend.
3. Valorar la comprensión individual del proyecto a través de la sustentación de las funcionalidades.
4. Asegurar el cumplimiento de los requisitos de generación de reportes y funcionalidades.
5. Promover buenas prácticas en la implementación del código.

Entregable => Segundo entregable y final.

1. Funcionalidades de Registro de Información:

- **Frontend:**
 - Completar los formularios interactivos necesarios para gestionar la información de pacientes y citas médicas, asegurando una captura eficiente de datos.
 - Los formularios deben estar completamente funcionales, sin errores, y correctamente integrados con el backend, permitiendo validaciones en tiempo real.
- **Backend (Django):**
 - Crear y optimizar las APIs necesarias para gestionar toda la información capturada en los formularios del frontend y almacenarla correctamente en la base de datos.

2. Generación de Reportes:

- **Frontend:**
 - Implementar las vistas y funcionalidades que permitan generar y visualizar todos los reportes requeridos en el proyecto sin errores.
 - Exportar los reportes en formato PDF desde el sistema, sin usar el Django admin.
- **Backend (Django):**
 - Desarrollar las APIs necesarias para que todos los datos sean enviados correctamente desde el backend al frontend, asegurando que todos los reportes sean funcionales y no contengan errores.
- **Reportes requeridos (todos los reportes del proyecto):**
 - Reporte de Información Básica de los Paciente.
 - Reporte de Historial de Citas Médicas.
 - Reporte de Facturación y Copagos.
 - Reporte de Inventarios de Medicamentos.
 - Reporte de Historias Clínicas.

Todos los reportes deben estar completamente funcionales y sin errores, con la posibilidad de exportarlos en formato PDF.

3. Pruebas Unitarias:

- Implementar pruebas unitarias tanto para el frontend como para el backend, utilizando herramientas como pytest para el backend en Django.
- Asegurarse de que cada módulo del sistema tenga cobertura de pruebas para garantizar la estabilidad y calidad del código.

4. Documentación:

- Proporcionar documentación detallada del código y de la API utilizando Swagger u otra librería adecuada.

5. Despliegue:

- Desplegar el sistema en un entorno de producción, preferiblemente en una plataforma web gratuita como Heroku o Vercel para el frontend, asegurando que la conexión entre el frontend y el backend esté funcionando correctamente.

Restricciones:

- No utilizar el Django Admin para ninguna de las funcionalidades. Todo debe gestionarse a través del frontend y APIs.

Instrucciones para la Entrega

El líder del equipo debe enviar el Pull Request (PR) al enlace <https://github.com/JEISSON25/tendenciasGClinica> correspondiente hasta la fecha límite asignada.

Fecha de Entrega:

- **Fecha límite:** 2 de Noviembre, 09:59 a.m.
- **Fecha de Sustentación del proyecto por grupo:** 10:00 a.m. hasta las 2:00 p.m.
- **Tiempo asignado por sustentación:** 20 minutos por grupo.

RUBRICA DE EVALUACIÓN

Calificación Grupal (5%)

1. Generación de reportes funcionales (2%):
 - 2%: Los reportes de Información de Profesores y su Carga Académica y de Comunicaciones y Notificaciones Enviadas están completamente implementados, son funcionales y pueden ser exportados en formato PDF desde la interfaz del sistema.
 - 1%: Los reportes están implementados pero con algunos errores menores, o uno de los reportes no está completamente funcional.
 - 0%: No se han generado reportes funcionales o no están disponibles para su exportación desde la interfaz.
 2. Desarrollo de la interfaz funcional (1%):
 - 1%: La interfaz está completamente funcional con los formularios requeridos para la captura de información (CRUD completo), incluyendo la gestión de estudiantes, profesores y cursos desde el frontend. Las APIs del backend gestionan correctamente la información capturada.
 - 0.5%: La interfaz está parcialmente funcional, pero presenta algunos errores en la captura de información o en la integración con las APIs del backend.
 - 0%: La interfaz tiene varias limitaciones y no permite realizar correctamente la mayoría de las operaciones CRUD.
 3. Documentación (1%):
 - 1%: La documentación del sistema está completa, explicando las principales funcionalidades, incluyendo la API, y está generada con una herramienta adecuada como Swagger.
 - 0.5%: La documentación está incompleta o tiene algunas partes que no describen claramente las funcionalidades o la API.
 - 0%: No se ha proporcionado documentación adecuada para el sistema.
 4. Pruebas Unitarias (0.5%):
 - 0.5%: Las pruebas unitarias están completamente implementadas y cubren todas las funcionalidades del sistema, tanto del frontend como del backend.
 - 0.25%: Las pruebas unitarias están implementadas, pero no cubren todas las funcionalidades del sistema o presentan fallas en algunos casos.
 - 0%: No se han implementado pruebas unitarias.
 5. Despliegue (0.5%):
 - 0.5%: El sistema ha sido desplegado correctamente en un entorno de producción funcional, permitiendo la interacción con todas las funcionalidades del sistema.
 - 0.25%: El sistema ha sido desplegado, pero presenta algunos errores que impiden la interacción con algunas funcionalidades.
 - 0%: El sistema no ha sido desplegado correctamente.
-

Calificación Individual (15%)

1. Comprensión de los reportes y su generación (3%):
 - 3%: Explica claramente cómo se generan los reportes de Información de Profesores y su Carga Académica y de Comunicaciones y Notificaciones Enviadas, describiendo tanto la lógica de negocio como la integración entre frontend y backend.
 - 2%: Explica la mayor parte del proceso, pero tiene algunas lagunas en cuanto a la implementación o el flujo de los datos entre frontend y backend.

- 1%: Explicación insuficiente o con deficiencias importantes sobre la generación de reportes o su implementación.
- 2. Explicación de la interfaz y funcionalidades principales (2%):
 - 2%: Describe y sustenta con claridad el funcionamiento de los formularios de registro de información (CRUD), las funcionalidades de la interfaz y cómo se conecta con las APIs del backend.
 - 1%: Tiene una comprensión parcial de las funcionalidades del frontend o de su integración con las APIs.
 - 0%: No logra explicar correctamente el funcionamiento de la interfaz o la gestión de los formularios.
- 3. Documentación y comprensión de la API (2%):
 - 2%: Explica y demuestra una comprensión clara de la documentación generada, incluyendo cómo se utiliza la API del sistema para gestionar la información.
 - 1%: Explica la documentación, pero tiene algunas lagunas en cuanto a la comprensión del funcionamiento de la API.
 - 0%: No demuestra comprensión suficiente de la documentación ni de la API.
- 4. Explicación de las pruebas unitarias (3%):
 - 3%: Explica detalladamente cómo se implementaron las pruebas unitarias en el sistema, tanto para el frontend como para el backend, y la cobertura de dichas pruebas.
 - 2%: Explica parte de las pruebas unitarias, pero tiene lagunas en cuanto a la cobertura o implementación de algunas funcionalidades.
 - 1%: La explicación de las pruebas unitarias es insuficiente o tiene deficiencias importantes en la cobertura.
- 5. Respuesta a preguntas sobre las funcionalidades y despliegue (5%):
 - 5%: Responde correctamente a todas las preguntas relacionadas con las funcionalidades del sistema, incluyendo la integración entre frontend y backend, la captura de información, la generación de reportes, y el proceso de despliegue.
 - 3%: Responde correctamente a la mayoría de las preguntas, pero presenta algunos errores o respuestas incompletas.
 - 1%: Responde incorrectamente o de forma insuficiente a la mayoría de las preguntas.