

INSTRUCCIONES GENERALES

- El desarrollo de este certamen es en equipos de **máximo 3 integrantes, práctico**, correspondiente a una **entrega con presentación**.
- Debe publicar su proyecto en un repositorio publico GitHub cuyo nombre corresponda a la aplicación a desarrollar.
- El plazo máximo de entrega es lunes 08 de julio a las 9:35 hrs.
- La entrega es exclusivamente vía aula: enlace a GitHub y sitio comprimido en rar.
- Cada grupo es responsable del respaldo de su avance/código del proyecto, de tal forma que estén preparados en caso de fallas o pérdidas de equipos computacionales utilizados para el desarrollo de la aplicación.
- La nota del certamen es asignada en el momento de la presentación. **Si un estudiante no presenta sin motivo justificado sólo obtendrá el 55% de su nota.**

ESPECIFICACIÓN DEL PROYECTO

En este certamen los estudiantes en grupos de máximo tres integrantes deberán construir un sitio web basado en Django Framework que solucione el problema planteado en el caso.

Objetivos

1. Desarrollar aplicaciones que promuevan el desacoplamiento entre sus componentes utilizando un modelo de capas.
2. Desarrollar programas que permitan presentar y modificar información interactuando con bases de datos.
3. Desarrollar programas que realicen llamadas a servicios externos invocando APIs REST.

Descripción del caso

La Empresa Nacional de Combustibles (ENC) necesita implementar un sistema para el registro, control y análisis de su producción diaria. Actualmente, el proceso de registro y análisis se realiza utilizando diferentes medios lo que conlleva a errores y falta de información precisa y oportuna para la toma de decisiones. Además, la Autoridad Fiscalizadora ha solicitado se les entregue información periódica respecto a distintos aspectos de la producción por lo que la implementación debe considerar la entrega de información a los sistemas centrales de la organización fiscalizadora.

Se requiere una aplicación basada en Django que permita a los operadores de la planta registrar la producción de forma eficiente y que proporcione un API para entregar información procesada respecto a la producción, con algunas capacidades de filtrado.

Requisitos Funcionales:

| | |
|-------------|--|
| RF01 | El sistema debe permitir al operador de planta realizar el registro de la producción diaria de su estación desde un formulario en el Front-End de la aplicación (Ver Anexo 1) |
| RF02 | El sistema debe permitir al operador de planta realizar modificaciones en los registros de la producción diaria (sólo los ingresados por él) en el Front-End de la aplicación. En el sistema, debe quedar registro de qué usuario y cuándo se realizó la modificación. |
| RF03 | El sistema debe permitir al supervisor de turno realizar la anulación (eliminación lógica) de registros de producción mediante Django Admin. Debe quedar registro de qué usuario y cuándo se realizó la anulación. |
| RF04 | El sistema debe permitir al operador de planta consultar la producción registrada por él desde el Front End de la aplicación. |
| RF05 | El sistema debe permitir al administrador del sistema realizar la creación de nuevos productos y plantas mediante Django Admin. |
| RF06 | El sistema debe permitir al administrador del sistema realizar la edición de productos y plantas existentes mediante Django Admin |
| RF07 | El sistema debe permitir al administrador del sistema realizar la eliminación de productos y plantas mediante Django Admin. |
| RF08 | El sistema debe permitir a la autoridad fiscalizadora conocer el total de producción de cada combustible, organizada por planta desde un API REST. |
| RF09 | El sistema debe permitir a la autoridad fiscalizadora realizar la consulta de producción de cada combustible filtrada por año y mes en la API REST. |
| RF10 | El sistema debe enviar una notificación al canal #ProductionTracker del Slack de la compañía cada vez que se realice un nuevo registro de producción (Ver Anexo 2) |

ANEXOS

Anexo 1: Reglas del Negocio

Los productos que produce ENC están distribuidos en 3 plantas de producción, donde se refinan los siguientes productos combustibles:

Planta de Refinación de Gasolina (Código: PRG)

- Gasolina 93 Octanos (G93)
- Gasolina 95 Octanos (G95)
- Gasolina 97 Octanos (G97)

Planta de Refinación de Diesel (Código: PRD)

- Diesel convencional (DIE)
- Diesel de alto rendimiento (DIP)

Planta de Refinación de Combustibles para Aviación (Código: PRA)

- Jet A-1 (JA1)
- Av Gas (AVG)

En cada planta trabajan 3 operadores por turno, en 3 turnos diarios (ENC produce 24/7). La dotación de operarios totales de la planta son 40 personas.

La información que se debe registrar es la siguiente:

- Código combustible
- Litros de combustible producidos en el turno
- Fecha producción
- Turno (mañana (AM), tarde (PM), noche (MM))
- Hora registro
- Operador que realizó el registro



NOTAS PARA LA DEMOSTRACIÓN

Considerar la creación de las cuentas de usuario respectivas. Utilizar modelo de Usuarios Django.
Se asumirá, por asuntos de tiempo, que la gestión de turnos está resuelta

Anexo 2: Canal Slack y notificaciones

ENC cuenta con un espacio de trabajo en Slack donde distintos sistemas reportan alertas o incidencias. La aplicación debe enviar un mensaje al canal #ProductTracker cada vez que se realice un nuevo registro de producción.

El mensaje debe mostrarse en Slack siguiendo el formato:

Fecha hora CÓDIGO PLANTA – Nuevo Registro de Producción – CÓDIGO COMBUSTIBLE litros registrados | Total Almacenado: litros totales

Ejemplo



23-06-2024 18:48 PRG - Nuevo Registro de Producción - G93 1090 lts. | Total Almacenado: 10500 lts.

EVALUACIÓN

Restricciones

1. Aplicaciones que no compilen tendrán como nota máxima un 20.
2. Archivos con código que están en el proyecto pero que no son utilizados por la aplicación no tienen puntaje.
3. El repositorio Github del equipo debe registrar contribución de dificultad/esfuerzo equivalente de cada integrante. Repositorios con un único commit o commit de solo algunos integrantes del equipo tendrán un descuento de 20 puntos.
4. El puntaje por preguntas al estudiante es individual.
5. El estudiante o grupo obtiene nota cero si:
 - a. Copia código. La copia no está permitida, ya sea facilitando código a otros grupos, copiando código de otros grupos o copiando aplicaciones (o partes de ella) desde internet u otro tipo de terceras personas.
 - b. No entrega aplicación en el plazo establecido.

Pauta De Corrección

| | ASPECTO A EVALUAR | PUNTAJE |
|----|---|------------|
| 1. | Construye un proyecto Django, organizando los archivos respetando el patrón MVT y los principios de desacoplamiento utilizando apps para cada módulo del sistema. | 5 |
| 2. | Genera Modelos que cumple con las necesidades de la organización. | 5 |
| 3. | Realiza las migraciones correspondientes para traspasar el set de modelos a la base de datos. | 5 |
| 4. | Construye las Vistas necesarias para cumplir con los requerimientos del cliente. | 5 |
| 5. | Construye una app basada en Django REST Framework que permita desplegar la información solicitada a otros sistemas externos. | 15 |
| 6. | Utiliza API de Terceros para integrar la aplicación desarrollada con otros sistemas externos. | 15 |
| 7. | El equipo cuenta con un repositorio GIT para el control de versiones del sistema, demostrando trabajo colaborativo mediante commits, branch o merge. El aporte al repositorio de cada uno de los integrantes del equipo es balanceado y se aprecia claramente las tareas realizadas por cada miembro del equipo. (El puntaje de este item se distribuirá en proporción al aporte de cada integrante del equipo al Repositorio GIT). | 20 |
| 8. | Responde a las preguntas realizadas en la presentación demostrando conocimiento respecto a las tecnologías utilizadas, las características del patrón de arquitectura o el detalle de implementación de alguno de los componentes en los que trabajó. | 30 |
| | PUNTAJE TOTAL | 100 |