ENSAYO PROYECTO 3

202204496 – Rodrigo Sebastian Castro Aguilar

Resumen

Durante un periodo de dos semanas, me embarqué en el desafío de desarrollar una aplicación web desde cero. Este proceso implicó la creación de un repositorio en una plataforma de alojamiento de código fuente, configuración de versiones para marcar hitos importantes, y desarrollo del backend con Flask para gestionar rutas y solicitudes del cliente. El frontend se diseñó con HTML, CSS y JavaScript, proporcionando una experiencia de usuario interactiva. La lógica del negocio se encapsuló en la clase app_dao, permitiendo el análisis de datos cargados y operaciones de conteo detalladas. Este proyecto fue más que una experiencia técnica; fue un viaje educativo que enseñó sobre el proceso completo de desarrollo de aplicaciones web, desde la planificación hasta la implementación.

Palabras clave

- Desarrollo web
- Flask
- Repositorio
- Frontend interactivo
- Lógica del negocio

Abstract

Over a period of two weeks, I embarked on the challenge of developing a web application from scratch. This process involved creating a repository on a source code hosting platform, configuring releases to mark important milestones, and developing the backend with Flask to manage routes and client requests. The frontend was designed with HTML, CSS and JavaScript, providing an interactive user experience. The business logic was encapsulated in the app_dao class, allowing analysis of loaded data and detailed counting operations. This project was more than a technical experience; It was an educational journey that taught about the entire web application development process, from planning to implementation.

Keywords

- *Desarrollo web*
- *Flask*
- *Repositorio*
- *Frontend interactivo*
- *Lógica del negocio*

Introducción

El desarrollo de aplicaciones web es un proceso complejo que requiere una planificación cuidadosa, habilidades técnicas y una comprensión profunda de las tecnologías involucradas. Durante dos semanas, me embarqué en un emocionante viaje para crear una aplicación web desde cero. Este ensayo explora detalladamente cada fase del desarrollo, desde la creación del repositorio hasta la implementación del código y el lanzamiento final de la aplicación.

Desarrollo del tema

Fase 1: Estableciendo las Bases - Repositorio y Versiones:

El primer paso crucial fue la creación de un repositorio en una plataforma de alojamiento de código fuente. Aquí está el código de cómo se creó el repositorio y se configuraron las versiones:

```
git init
git add .
git commit -m "Primer commit: Configuración inicial"
git branch -M main
git remote add origin <URL_del_repositorio_en_GitHub>
git push -u origin main
```

Fase 2: Desarrollo del Backend con app.py:

El archivo app.py se convirtió en el núcleo del backend de la aplicación. Aquí hay un extracto del código donde se configuraron las rutas y se manejaron las solicitudes del cliente utilizando Flask:

```
from flask import Flask, jsonify, request
app = Flask(__name__)

@app.route('/')
def index():
    return '¡Hola, mundo!'

@app.route('/cargar_datos', methods=['POST'])
def cargar_datos():
    # Lógica para cargar y procesar los datos del cliente
    datos = request.get_json()
    # ... (operaciones de procesamiento de datos)
    return jsonify({'mensaje': 'Datos cargados correctamente!'})
```

Fase 3: Creación del Frontend con HTML, CSS y JavaScript:

El frontend de la aplicación se diseñó con HTML, CSS y JavaScript para proporcionar una experiencia de usuario interactiva. Aquí hay un ejemplo de código HTML y JavaScript para el formulario de carga de archivos:

```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <title>Aplicación Web</title>
</head>
<body>
    <h1>Subir Archivos</h1>
    <input type="file" id="archivo">
    <button onclick="cargarDatos()">Cargar</button>

    <script>
        function cargarDatos() {
            var archivo = document.getElementById('archivo').files[0];
            // Operaciones para enviar el archivo al backend y procesarlo
```

```
    }
  </script>
</body>
</html>
```

Fase 4: Implementación de la Lógica de Negocio con app_dao:
La lógica del negocio se encapsuló en la clase app_dao. Aquí hay un fragmento de código que muestra cómo se utilizó esta clase para analizar los datos cargados y realizar operaciones de conteo:

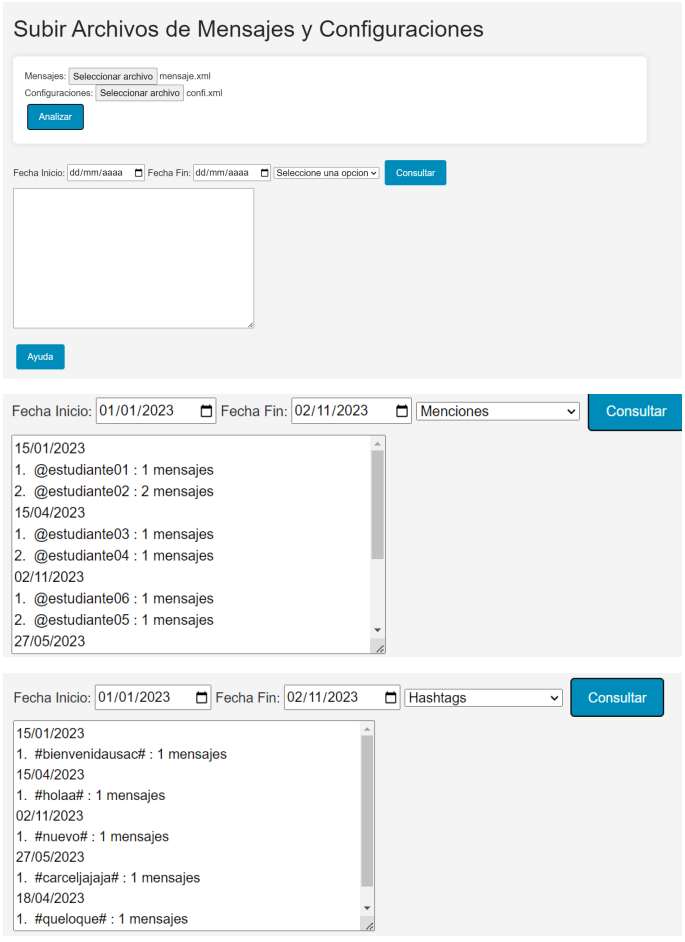
```
class app_dao:
  def __init__(self):
    self.mensajes = []

  def cargar_mensajes(self, datos):
      # Lógica para cargar mensajes desde los datos proporcionados
    # ...

  def contar_menciones(self):
          # Lógica para contar menciones en los mensajes
    # ...
```

Conclusiones

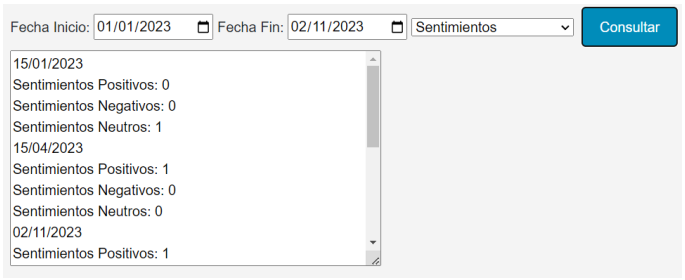
Conclusión:
Este proyecto fue más que una experiencia técnica; fue un viaje educativo que me enseñó sobre el proceso completo de desarrollo de aplicaciones web. Desde la configuración del repositorio hasta la implementación de la lógica del negocio y el diseño del frontend, cada línea de código fue una oportunidad para aprender y crecer como desarrollador.

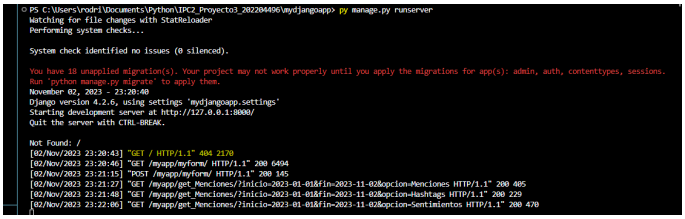
Este viaje no solo fue un ejercicio de codificación, sino también una lección valiosa sobre la importancia de la planificación, la perseverancia y la pasión en el desarrollo de aplicaciones web modernas. En resumen, este proyecto no solo fue un logro técnico, sino también una demostración de las habilidades adquiridas durante este emocionante viaje de dos semanas en el mundo del desarrollo web.

Anexos



Universidad de San Carlos de Guatemala
Escuela de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería
Introducción a la programación y computación 2, 2do. Semestre 2023.






```python
def consultar_sentimientos(self, fecha_inicio, fecha_fin):
    response = ""
    fecha_inicio = datetime.strptime(fecha_inicio, "%Y-%m-%d")
    fecha_fin = datetime.strptime(fecha_fin, "%Y-%m-%d")

    for texto in self.mensajes:
        self.sentimientos_manage.reset()
        cadena = texto.texto.lower()
        fecha_mensaje = datetime.strptime(texto.fecha, "%d/%m/%Y")

        # Comprobar si la fecha de mensaje está dentro del rango de fechas
        if fecha_inicio <= fecha_mensaje <= fecha_fin:
            usuarios_leidos = re.findall(r'(@\w+)', texto.texto.lower())
            hashtags_leidos = re.findall(r'#(.*?)#', texto.texto.lower())

            for user in usuarios_leidos:
                cadena = cadena.replace(user, '')

            for hs in hashtags_leidos:
                cadena= cadena.replace('#' + hs + '#', '')

            response += texto.fecha+"\n"
            self.buscar_sentimiento(cadena)
            response += "Sentimientos Positivos: "+str(self.sentimientos_manage.positivo)+"\n"
            response += "Sentimientos Negativos: "+str(self.sentimientos_manage.negativo)+"\n"
            response += "Sentimientos Neutros: "+str(self.sentimientos_manage.neutro)+"\n"
    return response

def buscar_sentimiento(self,cadena):
    positivo = 0
```

```
</style>
</head>
    <body>
        <div class="container mt-4">
        <h1>Subir Archivos de Mensajes y Configuraciones</h1>

        <div id="upload-container" class="form-group mt-4">
            <form method="POST" enctype="multipart/form-data" id="myform">
                {% csrf_token %}
                Mensajes: <input type="file" id="file-mensaje-input" name="file_mensaje" required><br>
                Configuraciones: <input type="file" id="file-config-input" name="file_config" required><br>
                <button id="submit-button" type="button">Analizar</button>
            </form>
        </div>
        <div id="response" class="form-group mt-4">
            <label for="fecha-inicio">Fecha Inicio:</label>
            <input type="date" id="fecha-inicio" name="fecha-inicio" required>

            <label for="fecha-fin">Fecha Fin:</label>
            <input type="date" id="fecha-fin" name="fecha-fin" required>


            <select id="opcion" name="opcion" required class="form-select form-select-sm" aria-label="Small select example">
                <option selected>Seleccione una opcion</option>
                <option value="Menciones">Menciones</option>
                <option value="Hashtags">Hashtags</option>
                <option value="Sentimientos">Sentimientos</option>
            </select>

            <button type="button" id="getResponseButton">Consultar</button>
```