



Proyecto ACME Ytzhak Dawidh Carvajal Contreras

DESCRIPCIÓN GENERAL

El siguiente proyecto que consta principalmente de 4 módulos, será utilizado para:

- Crear una aplicación o sistema que facilite la gestión de datos mediante archivos JSON.
- Garantizar la persistencia de los datos para que las funcionalidades sean fiables y escalables.
- Enfocarse en la facilidad de uso y la eficiencia en el manejo de grandes volúmenes de información.

PROYECTO

✓ ACME

✓ interfaz

🐍 interfaz.py

✓ modelo

🐍 modelo.py

✓ persistencia

🐍 persistencia.py

> util

🐍 main.py

≡ password.txt

ARQUITECTURA DEL SISTEMA

Diagrama del Sistema:

- **Un servidor backend** que se encarga del procesamiento y manejo de la lógica de los datos.
- **Frontend interactivo** (menús de consola) que permite la visualización y manipulación de los datos almacenados en JSON.

Módulos principales:

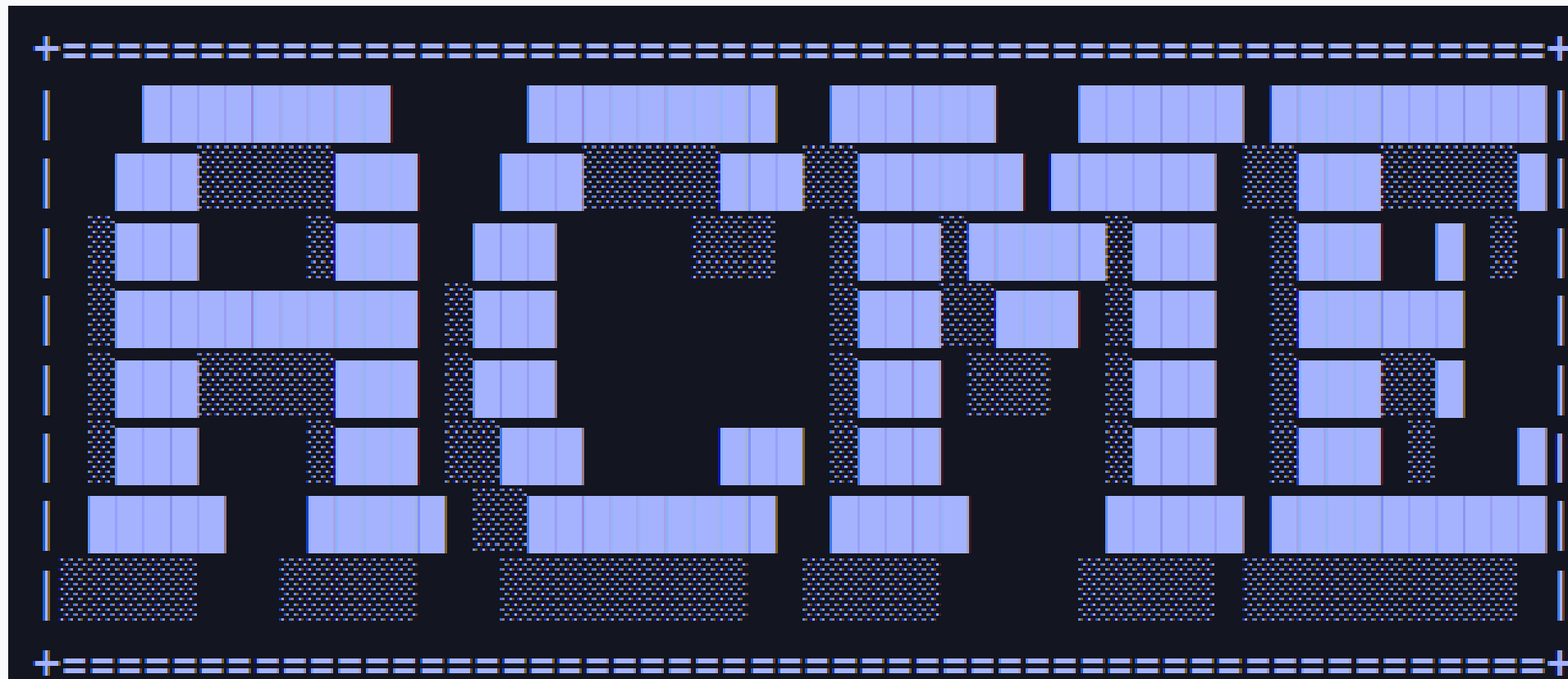
- Autenticación de usuarios.
- Gestión de archivos y datos.
- Persistencia de información en formato JSON.



ESTRUCTURAS Y PERSISTENCIA DE DATOS

Uso de archivos JSON:

- Los datos se guardarán y se cargarán en formato JSON, permitiendo una estructura clara y jerárquica.
- Se implementará un sistema de manejo de datos que permite la búsqueda, edición y eliminación de entradas.
- Se garantizará la persistencia de datos mediante copias de seguridad regulares o mecanismos de respaldo.



```
def guardarGrupos(grupos):  
    with open('ACME/util/registroGrupos.json', 'w') as file:  
        json.dump(grupos, file, indent=4)  
  
    if not file.closed:  
        file.close()  
  
def guardarModulos(modulo):  
    with open('ACME/util/registroModulos.json', 'w') as file:  
        json.dump(modulo, file, indent=4)  
  
    if not file.closed:  
        file.close()  
  
def guardarEstudiantes(estudiante):  
    with open('ACME/util/registroEstudiantes.json', 'w') as file:  
        json.dump(estudiante, file, indent=4)  
  
    if not file.closed:  
        file.close()
```

DECISIONES DE PROGRAMACIÓN

Funciones Clave:

- Función de lectura y escritura en JSON de manera eficiente para manejar grandes volúmenes de datos.
- Módulo de autenticación de usuarios, que incluirá roles y permisos según el tipo de usuario.

Manejo de Usuarios:

- Creación de perfiles de usuarios con diferentes niveles de acceso a la información.
- Control de acceso basado en roles: administradores, usuarios regulares, y usuarios de solo lectura.

```
def guardarGrupos(grupos): ...  
  
def guardarModulos(modulo): ...  
  
def guardarEstudiantes(estudiante): ...  
  
def guardarDocentes(docente): ...  
  
def guardarAsistencia(asistencia): ...  
  
def cargarGrupos(arch): ...  
  
def cargarDocentes(arch): ...  
  
def cargarModulos(arch): ...  
  
def cargarEstudiantes(arch): ...  
  
def cargarAsistencia(arch): ...
```

CONCLUSIÓN DEL PROYECTO

El proyecto ha logrado implementar un sistema robusto para la gestión de datos mediante el uso de archivos JSON, garantizando tanto la persistencia como la integridad de la información. La estructura modular facilita su expansión y mantenimiento, mientras que las decisiones de programación, como el manejo de usuarios y roles, aseguran un control adecuado de acceso a la información. Además, la arquitectura del sistema permite una interacción fluida entre la interfaz de usuario y los datos, optimizando la experiencia para distintos tipos de usuarios. Esto convierte al sistema en una solución eficaz y escalable para la gestión de datos.

