

Documentación con respecto al avance dos

Grupo #1

Universidad Fidélitas

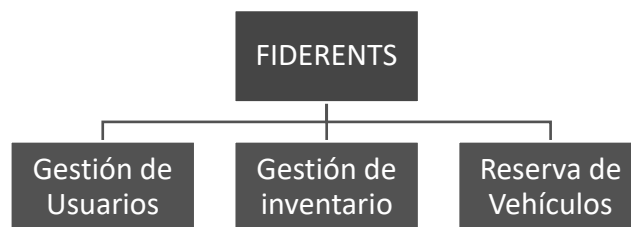
Programación Básica con Python

Ing. José Antonio Ortega González

Marzo 8, 2024

Avance Final

La propuesta asignada es un sistema de gestión de vehículos que permite a los usuarios registrarse, iniciar sesión y realizar diversas acciones relacionadas con la administración de vehículos, como agregar vehículos, inhabilitar vehículos, realizar reservas y mostrar la disponibilidad de vehículos.



En base al siguiente esquema, los requerimientos deseados por la empresa a partir del primer avance son:

1. **Gestión de Usuarios**
 - Registro de nuevos usuarios
 - Inicio de sesión con validación de credenciales
2. **Gestión de Inventario**
 - Agregar nuevos vehículos
 - Inhabilitar vehículos existentes
 - Visualizar vehículos disponibles
3. **Reserva de Vehículos**
 - Permitir a los usuarios realizar reservas
 - Actualizar la cantidad de vehículos disponibles después de una reserva
4. **Cambiar de sede**
 - Permitir a los usuarios cambiar de sede en cualquier momento sin perder ninguna información sobre ello.
 - Permitir al usuario tener varias disponibilidades de sedes dependiendo de la hora.
5. **Persistencia de datos**
 - Permitir a los usuarios siempre estar activos en el programa con la información resguardada en archivos textos planos

El sistema está diseñado para gestionar el inventario y las reservas de vehículos en diferentes sedes de la Universidad Fidelitas. Permite a los usuarios registrados agregar, inhabilitar y reservar vehículos, mientras que los usuarios invitados tienen acceso limitado a estas funcionalidades.

Planeamiento

1. Gestión de Usuarios

- Implementar funcionalidad de registro de nuevos usuarios
- Implementar funcionalidad de inicio de sesión con validación de credenciales

2. Gestión de Inventario

- Implementar funcionalidad para agregar nuevos vehículos al inventario
- Implementar funcionalidad para inhabilitar vehículos existentes
- Implementar funcionalidad para visualizar vehículos disponibles

3. Reserva de Vehículos

- Implementar funcionalidad para que los usuarios puedan realizar reservas
- Actualizar la cantidad de vehículos disponibles después de una reserva

4. Cambiar de sede

- Implementar un sistema para que los usuarios logren cambiar de sede
- Aplicar el sistema para que no se pierda ninguna información de ningún tipo
- En el caso de cambiar sedes nos vamos a tomar el tiempo para explicarlo mejor:

5. Persistencia de datos

- Guardar los datos en un archivo para que los usuarios siempre puedan asistir una vez se hayan creado.
- Tener varios conjuntos que realicen la persistencia en base a los datos.

Comenzamos con un Inicio, que sería el inicio del programa como tal, de una vez pasamos para los cambios de sede, hay cierta elección que preside a la hora de escoger, eso dependerá del usuario totalmente, después comenzamos la fase del menú de registrarse, por ultimo las acciones principales del usuario, que son, iniciar sesión, registrarse, ingresar como invitado, cambiar de sede, o salirse del programa, depende de lo que no escoga, podrá iniciar sesión para seguir el programa, cabe aclarar que no se pierde la información de un usuario amenos de que se salga del programa totalmente. Resumidamente, el código comienza mostrando al usuario las opciones de sede disponibles. Luego, permite al usuario seleccionar una sede y muestra el horario correspondiente. Una vez seleccionada la sede, se accede a un menú de ingreso donde el usuario puede realizar varias acciones según su tipo de acceso. Finalmente, se proporciona la opción de cambiar de sede en cualquier momento, lo que lleva de vuelta al usuario al proceso de selección de sede.

1. Permitir al usuario seleccionar una sede.
2. Mostrar el horario de la sede seleccionada.
3. Proporcionar un menú de ingreso para la sede seleccionada.
4. Implementar acciones específicas para usuarios registrados, nuevos usuarios e invitados.

5. Permitir al usuario cambiar de sede en cualquier momento.
6. Salir del programa cuando el usuario lo solicite.

- Requerimiento: Gestión de Inventario

Actividades:

Diseño del Esquema de Datos: Definición de la estructura de datos para almacenar información sobre los vehículos en el inventario, incluyendo campos como placa, marca, modelo, etc.

Implementación de Funcionalidades: Desarrollo de funciones para agregar y inhabilitar vehículos en el inventario.

Pruebas y Depuración: Verificación de la funcionalidad del módulo de gestión de inventario, identificación y corrección de errores.

- Requerimiento: Gestión de Reservas

Actividades:

Diseño del Proceso de Reserva: Definición de los pasos necesarios para que un usuario pueda realizar una reserva de vehículo.

Implementación de Funcionalidades: Desarrollo de funciones para realizar reservas, verificar disponibilidad de vehículos, etc.

Pruebas y Depuración: Verificación de la funcionalidad del módulo de gestión de reservas, asegurando que las reservas se realicen correctamente y se actualice el inventario adecuadamente.

- Requerimiento: Interfaz de Usuario

Actividades:

Diseño de la Interfaz de Usuario: Creación de prototipos de pantalla para las diferentes funcionalidades del sistema, incluyendo el ingreso de usuarios, la gestión de inventario y reservas.

Implementación de la Interfaz de Usuario: Desarrollo de la interfaz de usuario utilizando tecnologías como Python y bibliotecas gráficas como Tkinter.

Pruebas de Usabilidad: Evaluación de la usabilidad de la interfaz de usuario, identificando posibles mejoras en la navegación y la experiencia del usuario.

1. Uso de estructuras de datos adecuadas: Se utilizan diccionarios ("inventario", "sedes", "reservas_activas") y listas ("usuarios", "contrasenas", "cedula") para almacenar la información de forma organizada y eficiente.

2. Funciones bien definidas: Las funciones están bien definidas y modularizan el código, lo que facilita su comprensión y mantenimiento.

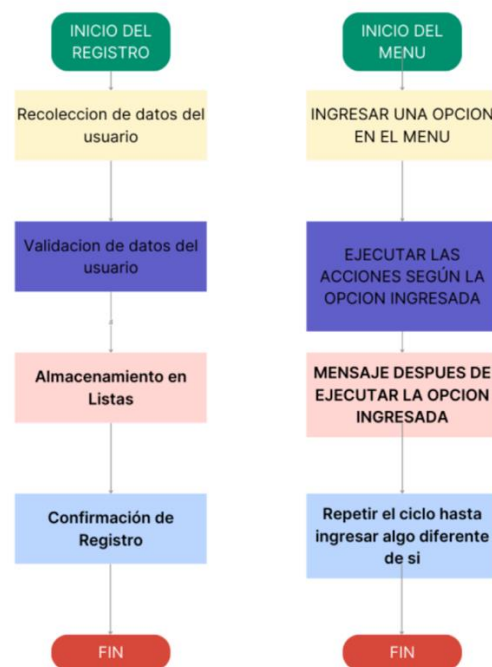
3. Manejo de archivos: Se utiliza la apertura y escritura de archivos ("usuarios.txt", "contrasenas.txt", etc.) para guardar y cargar la información de forma persistente.
4. Menús claros: Los menús están bien presentados y facilitan la navegación por el programa al usuario.
5. Validación básica de entrada: Se utiliza la función input() para obtener información del usuario y se realizan algunas validaciones básicas (por ejemplo, que la opción ingresada sea un número entero dentro del rango válido).

Requerimiento	Actividades	Fecha de Entrega	Responsable
Gestión de Usuarios	Implementar registro de nuevos usuarios e implementar el inicio de sesión.	Semana 3 y 4	<ul style="list-style-type: none"> • Steilyn Cedeño García • Juan Andrey Vindas Matarrita
Gestión de Inventario	Agregar nuevos vehículos, Inhabilitar vehículos y visualizar los disponibles.	Semana 4 y 5	<ul style="list-style-type: none"> • Julián Clot Córdoba • Stephanie Dayan Cruz García • Sebastián Cajina Calvo
Reserva de Vehículos	Realizar reservas y actualizar dependiendo de su disponibilidad.	Semana 5 y 6	<ul style="list-style-type: none"> • Julián Clot Córdoba • Steilyn Cedeño García
Cambio de sede	Cambiar de sede según los deseos del usuario sin perder ninguna información pertinente.	Semana 6 y 7	<ul style="list-style-type: none"> • Julian Clot Cordoba • Steilyn Cedeño García
Integración y Pruebas	Pruebas de funcionalidad. Persistencia de datos	Semana 6 y 12	<ul style="list-style-type: none"> • Julián Clot Córdoba

			<ul style="list-style-type: none"> Steilyn Cedeño García
Documentación y Entrega Final	Preparar la documentación junto a la entrega total del código. Reservas incluidas y activas	Semana 8	<ul style="list-style-type: none"> Julián Clot Córdoba Steilyn Cedeño García

A continuación, mostraremos un diagrama de flujo principal del código, que abarca las funciones principales y generales del código, consiguiente a eso se estará adjuntando el repositorio (GitHub) del código fuente.

Diagrama de flujo



El siguiente enlace conducirá hacia el repositorio donde se encuentra el código fuente del primer avance: https://github.com/JulianClotCordoba/GRUPO_1_PF

Referencias

1. JulianClotCordoba. (s. f.). *GitHub - JulianClotCordoba/GRUPO_1_PF: Este repositorio es el codigo de todo el proyecto de introduccion a la programacion*. GitHub.

https://github.com/JulianClotCordoba/GRUPO_1_PF