

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

ESTRUCTURA DE DATOS

CATEDRÁTICO: ING. NEFTALÍ CALDERÓN

TUTOR ACADÉMICO: RODRIGO PORÓN



MANUAL TÉCNICO

JAVIER ANDRÉS VELÁSQUEZ BONILLA

CARNÉ: 202307775

SECCIÓN: E

GUATEMALA, 1 DE ABRIL DEL 2,024

INTRODUCCIÓN

El desarrollo de software es un campo en constante evolución y adquisición de habilidades fundamentales. En el contexto de este proyecto, nos sumergimos en el mundo de la programación utilizando el lenguaje Java como herramienta principal. Esta práctica se presenta como un paso adicional en el proceso de aprendizaje, donde los estudiantes pueden aplicar los conocimientos previamente adquiridos en el curso de Introducción a la Programación y Computación 1, y al mismo tiempo, familiarizarse con conceptos más avanzados.

OBJETIVOS

1. GENERAL

El propósito principal de esta práctica es familiarizar al estudiante con el lenguaje de programación Java. Se busca que los estudiantes no solo comprendan los conceptos básicos de programación, sino que también puedan aplicarlos de manera práctica en un entorno de desarrollo real. A través de esta práctica, los estudiantes tendrán la oportunidad de elaborar la lógica necesaria para presentar una solución efectiva a la problemática planteada.

2. ESPECÍFICOS

Utilizar Java como herramienta principal para el desarrollo de software, lo que incluye la implementación de conceptos de programación orientada a objetos.

Construir una aplicación con interfaz gráfica que permita una interacción intuitiva con el usuario.

Implementar sentencias de control, ciclos, vectores y librerías de interfaz gráfica para proporcionar funcionalidades específicas.

Aplicar conceptos de programación para crear herramientas administrativas que faciliten la gestión de la aplicación.

Implementar soluciones utilizando hilos para gestionar procesos concurrentes de manera eficiente.

Utilizar la serialización para crear respaldos de información y restablecerla, garantizando la persistencia de los datos incluso después de cerrar la aplicación.

ALCANCES DEL SISTEMA

El presente manual técnico tiene como objetivo principal brindar una guía exhaustiva para los desarrolladores involucrados en el sistema de gestión de viajes en carro implementado en Java. Este documento proporciona una explicación detallada del código fuente, desglosando cada componente, funcionalidad y lógica de programación utilizada en el proyecto. Su finalidad es ofrecer instrucciones claras y precisas que faciliten futuras modificaciones, asegurando la integridad y eficiencia del sistema a lo largo del tiempo.

El propósito fundamental de este manual técnico es dotar al equipo de desarrollo de los recursos necesarios para comprender, mantener y mejorar el sistema de gestión de viajes en carro. A través de una descripción exhaustiva del código fuente, se busca proporcionar una base sólida para la gestión, expansión y evolución efectiva del software. Además, este manual busca establecer un marco de referencia que permita garantizar la coherencia y la calidad del sistema en todas sus etapas de desarrollo y mantenimiento.

ESPECIFICACIÓN TÉCNICA

- **REQUISITOS DE HARDWARE** ○ El equipo que se utilizó cuenta con las siguientes características: - Procesador Ryzen 5 5500U de 6 núcleos, 2.10GHz - 8GB de RAM DDR4 a 3200Mhz - 256GB de almacenamiento SSD
- **REQUISITOS DE SOFTWARE** ○ El equipo que se utilizó cuenta con las siguientes características: - Sistema Operativo Windows 10 Home de 10 Bits

DESCRIPCIÓN DE LA SOLUCIÓN

La solución propuesta es una aplicación de gestión de viajes en carro desarrollada en Java, que ofrece una interfaz gráfica intuitiva para que los usuarios puedan planificar, realizar y monitorear sus viajes de manera eficiente. La aplicación permite a los administradores registrar rutas, a los usuarios generar viajes, y a ambos monitorear el progreso en tiempo real. Además, se mantiene un registro histórico de todos los viajes completados. La persistencia de datos se asegura mediante la serialización, garantizando la integridad y seguridad de la información. En resumen, la solución proporciona una plataforma completa para la gestión de viajes en carro, asegurando una experiencia de viaje segura y satisfactoria.

LÓGICA DEL PROGRAMA

Aquí tienes la especificación de para qué funcionan cada una de las partes enumeradas:

Main.java

1. **Librerías:**

- Se utilizan librerías de Swing y AWT para crear la interfaz gráfica de usuario.

2. **Variables Globales:**

- No hay variables globales definidas en este código.

3. **Clases:**

- Contiene la clase `Main`, que es la clase principal de la aplicación.

4. **Métodos y Funciones Utilizadas:**

- `main(String[] args)`: Este método es el punto de entrada de la aplicación. Se encarga de inicializar la interfaz gráfica y manejar eventos de cierre de ventana.

CarreraInterfaz.java

1. **Librerías:**

- Se utilizan librerías de Swing, AWT y otras para crear la interfaz gráfica y realizar operaciones de entrada/salida.

2. **Variables Globales:**

- Define varias variables globales como mapas para almacenar información sobre vehículos, carreras y posiciones guardadas, así como constantes para dimensiones de la ventana y botones.

3. **Clases:**

- Contiene la clase `CarreraInterfaz`, que representa la interfaz gráfica para gestionar carreras de vehículos.

4. **Métodos y Funciones Utilizadas:**

- ``CarreraInterfaz()`: Constructor de la clase que inicializa la interfaz.
- ``initComponents(Map<String, VehiculoInterfaz> estadoInicial)`: Inicializa los componentes de la interfaz.
- ``guardarEstado()`: Guarda el estado de los vehículos en un archivo.
- ``cargarEstado()`: Carga el estado de los vehículos desde un archivo.

GenerarViajeInterfaz.java

1. ****Librerías:****

- Se utilizan librerías de Swing, AWT y otras para crear la interfaz gráfica y realizar operaciones de entrada/salida.

2. ****Variables Globales:****

- Define varios componentes de la interfaz como combos, botones y etiquetas, así como un mapa para almacenar información sobre vehículos.

3. ****Clases:****

- Contiene la clase ``GenerarViajeInterfaz``, que representa la interfaz gráfica para generar nuevos viajes.

4. ****Métodos y Funciones Utilizadas:****

- ``GenerarViajeInterfaz()`: Constructor de la clase que inicializa la interfaz.
- ``initComponents()`: Inicializa los componentes de la interfaz.
- ``initListeners()`: Inicializa los listeners para manejar eventos de botón.
- ``validarPilotosDisponibles()`: Valida la disponibilidad de pilotos para el viaje.
- ``cargarDatosDesdeCSV(String rutaArchivo, JComboBox<String> comboBox, int columnIndex)`: Carga datos desde un archivo CSV en los combos.

VehiculoInterfaz.java

1. ****Librerías:****

- Se utilizan librerías de Swing, AWT y otras para crear la interfaz gráfica de los vehículos.

2. ****Variables Globales:****

- Define varias variables para almacenar información sobre el vehículo, como nombre, posición, combustible, etc.

3. ****Clases:****

- Contiene la clase `VehiculoInterfaz`, que representa la interfaz gráfica de un vehículo.

4. ****Métodos y Funciones Utilizadas:****

- Diversos métodos para inicializar, avanzar, retroceder, recargar combustible y actualizar la interfaz del vehículo.

Carrera.java

1. ****Librerías:****

- Se utilizan librerías relacionadas con la serialización y clases de la aplicación.

2. ****Variables Globales:****

- Define variables para almacenar referencias al vehículo y al botón de iniciar carrera.

3. ****Clases:****

- Contiene la clase `Carrera`, que implementa la lógica para controlar el avance de un vehículo durante una carrera.

4. ****Métodos y Funciones Utilizadas:****

- `run()`: Implementa la lógica para avanzar el vehículo durante la carrera.







