Documento de Alcance para ISFPP 2025

Miembros del grupo: Santiago Gabet, Damian Silva, Lizandro Marino.

Nombre de proyecto: Simulación de consultas de recorrido para colectivos

Objetivo General  
El objetivo de este proyecto es desarrollar un programa que simula consultas para un sistema de colectivos urbanos con el fin de aplicar lo aprendido con respecto a Programación Orientada a Objetos.

Descripción del Sistema

El sistema permitirá a los usuarios consultar recorridos de colectivos urbanos, ingresando datos como el día de la semana, parada de origen, parada de destino y hora de salida. Mostrará qué líneas pueden realizar el recorrido, posibles combinaciones de líneas, horarios, duración de viajes y diferencias de frecuencia según día de la semana y feriados. Además, la interfaz gráfica será intuitiva y permitirá la gestión, parametrización, persistencia de datos y generación de informes.

Alcance Funcional  
El sistema realizará las siguientes funciones:

Para el usuario final:

* Permitir al usuario indicar día de la semana, parada de origen, parada de destino y hora con el fin de mostrar al usuario:
* Líneas de colectivos disponibles para el recorrido consultado.
* Horarios de paso por cada parada y línea.
* Duración estimada del viaje.
* Posibles combinaciones entre líneas si no existe un recorrido directo.
* Diferencias de frecuencia para días de semana y feriados.
* Proveer una interfaz gráfica adecuada para facilitar la consulta y visualización de resultados (incluyendo mapas, recorridos, paradas).

Para el administrador:

* Permitir la gestión de datos y configuración mediante archivos de texto y/o base de datos.
* Generar informes y manejar hilos en la implementación grafica.

Fuera de Alcance  
El sistema **NO** realizará:

* Simulaciones de seguimiento en tiempo real de colectivos.
* Integraciones con sistemas externos de transporte.
* Compra o gestión de boletos.
* Gestión de pasajeros.
* Optimización de rutas considerando tráfico en tiempo real.
* Simulaciones de tráfico.
* Visualización de recorridos en mapas interactivos en línea (solo mapas simples o esquemáticos).
* Soporte para ciudades sin datos estructurados según el formato definido.

Usuarios Destinatarios

El sistema será utilizado por usuarios finales, los cuales realizaran consultas para obtener información relativa a los parámetros escritos y será utilizado por administradores, que podrán generar informes y almacenar la información en diferentes tipos de almacenamiento.

Restricciones o Supuestos

* El sistema debe ser genérico, permitiendo consultas para cualquier ciudad con datos disponibles.
* Se deben respetar las declaraciones indicadas en los diagramas UML presentados en el documento de presentación de la ISFPP.
* Los datos se entregan en archivos de texto y cada grupo selecciona una ciudad distinta respetando el formato dado.
* La interfaz debe ser suficientemente intuitiva para usuarios sin conocimientos técnicos.
* El código debe ser portable y ejecutable sin editar líneas (configuración en archivos externos).
* Se debe implementar por capas (paquetes) para facilitar mantenibilidad y escalabilidad.
* Solo se utilizarán herramientas y tecnologías permitidas por la cátedra (Java, JUnit, UML, etc.).

Definiciones técnicas y de diseño

1) Arquitectura del Sistema:  
Dentro del paquete colectivo, el sistema contara con las siguientes capas: controlador, conexión, dao, dao.secuencial, doc, interfaz, lógica, modelo, servicio, util.

* Controlador: Capa principal que coordinara todos los demás archivos para el funcionamiento del programa mediante MVC.
* Conexión: Capa donde se encontrara el patrón Factory para decidir el origen de los datos
* DAO, Secuencial y MySQL: Capas que manejaran la manera en la que se accede a los datos.
* Doc: Capa donde se encontrará toda la documentación del programa.
* Interfaz: Capa que se encargara de la interfaz gráfica de usuario.
* Logica: Capa que se encargara de realizar los cálculos de las consultas.
* Modelo: Capa donde se encontrará la estructura de los objetos usados en el programa.
* Servicio: Capa que actuara como intermediario entre DAO y Factory
* Util: Capa donde se encontrarán los métodos auxiliares del programa.

1. Patrones de diseño: El sistema utilizara los patrones de diseño MVC, Singleton, DAO y Factory para su funcionamiento.

2) Tecnologías y Herramientas:

Lenguaje: Java SE.

IDEs: Eclipse y VSCode.

Persistencia: Archivos de texto y bases de datos.

Control de versiones: Git con repositorios en GitHub.

3) Control de calidad:

El sistema incluirá pruebas para la lógica principal (Tests con JUnit).

El código tendrá documentación en formato Javadoc y se proveerán diagramas UML para describir la estructura del sistema.