

Sistema de Gestión de Colectivos

Proyecto Final – Programación II

Matías Agustín Sepúlveda
Carlos Miyen Brandolino
Enzo Sebastián Riera

¿Qué es el Proyecto?

Descripción

Una aplicación Java diseñada para consultar rutas de transporte público urbano.

Soporta dos interfaces de usuario:

- Consola de comandos (CLI)
- Interfaz gráfica (GUI) con JavaFX

Características Principales

- Cálculo de recorridos: directos, con transbordos y conexiones a pie.
- Flexibilidad de datos: Consume datos desde TXT, PostgreSQL y archivos binarios.
- Objetivo: Optimizar la planificación de viajes en transporte público para el usuario.

Metodología y Herramientas



Gestión

Comunicación: Discord

Tareas: Trello

Versiones: Git y GitHub



Desarrollo

IDEs: Eclipse, IntelliJ,

VSCode

Sistemas: Windows y Linux

Build: Maven



División de Tareas

Miyén: Arquitectura, Git, DAO

Secuencial.

Enzo: Datos, DAO PostgreSQL, Logs

(Log4j).

Agustín: Algoritmos, JavaFX,

Mapas.

Arquitectura y Patrones

Un vistazo a la estructura interna del sistema.

Arquitectura Inicial (Inspirada en 'Subte')

El Desafío 'Subte'

El proyecto "Subte" de la cátedra añadía patrones progresivamente.

Nuestro Desafío: Implementar **todos** los patrones relevantes (DAO, Factory, MVC, Facade) desde el inicio en una arquitectura unificada.

Restricciones del Diseño

- No modificar el paquete **Modelo**.
- Respetar nombres de paquetes originales.
- Reutilizar clases utilitarias de "Subte".

Patrones Clave de la Arquitectura



Factory + Registry

Permite la selección dinámica de implementaciones (DAO y UI) editando `factory.properties`, sin necesidad de recompilar el código.



Service Layer (Facade)

Actúa como punto de entrada a la lógica de negocio. El Service pide instancias (ej. "INTERFAZ") al Factory sin conocer la implementación concreta.



Ciudad (Singleton) y Coordinador

Ciudad: Es un Repository y Singleton. Carga todos los datos una sola vez al inicio para optimizar el rendimiento.

Coordinador: Orquesta la inicialización. **No** es un Singleton.

Arquitectura Final y Flujo

Ventajas Clave

- Desacoplamiento total entre capas.
- Configuración dinámica (cambio de DAOs/UI) sin recomilar.
- Alta extensibilidad y mantenibilidad.
- Testabilidad independiente por capa.
- Responsabilidades claras (SRP).

Flujo de Consulta

Usuario → Interfaz → Coordinador → Calculo → EstrategiaBusqueda
→ Ciudad (Repository)

Funcionalidades Avanzadas

Patrón Strategy

Se implementó la interfaz `EstrategiaBusqueda` con varias implementaciones (Directa, Caminando, Transbordo).

Aunque actualmente se ejecutan secuencialmente, el patrón permite gran flexibilidad a futuro (ej. "ruta más rápida" vs. "menos caminata").

Desafío Clave en JavaFX

Problema: `InterfazJavaFX` (que `extends Application`) pierde el contexto del Coordinador inyectado debido a su ciclo de vida.

Solución: Se utilizó un `static Coordinador` para preservar el contexto entre la inicialización y el método `start()` de JavaFX.

Gestión de Datos y DAOs

De los archivos de texto a una base de datos robusta.

Implementación de Datos



SecuencialDAO

Datos: Archivos `txt` con delimitador `;`.

Desafío: Parsing de coordenadas (`;` vs `:`).

Solución: Normalización con `String.replace(';', ':')`.



PostgreSQLDAO

Diseño del esquema de la base de datos.

Creación de script de carga.
Configuración de conexión JDBC.
Validación cruzada vs.
SecuencialDAO.



Log4j e I18N

Log4j: Configuración de `log4j.properties` para registrar historial de consultas y eventos.

I18N: Uso de ` ResourceBundle` para cargar textos en (ES/EN).

1er Incremento (Funcionalidad base)

```
InterfazConsola instanciada correctamente.

Ingrese código de parada origen: 44
Ingrese código de parada destino: 47
Ingrese día de la semana (1=lunes, 2=martes, ..., 7=domingo/feriado): 1
Ingrese hora de llegada (HH:MM): 10:35
[1] L1I
[2] L5R

===== Resultados =====

Parada origen: Parada [codigo=44, direccion=España, 1660]
Parada destino: Parada [codigo=47, direccion=España, 928]
Llega a la parada: 10:35
-----
Línea: Línea 1 Ida
Paradas: [Parada [codigo=44, direccion=España, 1660], Parada [codigo=43, direccion=España, 1450], Parada [codigo=47, direccion=España, 928]
Hora de Salida: 10:50
Duración: 00:03
-----
Duración total: 00:18
Hora de llegada: 10:53
-----
```

2er Incremento

Lemos, 3900]  Simulación Colectivo

Consulta de Recorridos - Colectivos

Seleccione los parámetros de consulta

Parada Origen: 31 - Bv. Almirante Brown, 3053

Parada Destino: 66 - Los Alamos, 3900

Día de la semana: 1 - Lunes

Hora de llegada: 10:35 (formato HH:MM)

Consultar Recorridos

Opciones de Recorrido: Es necesario llenar todos los campos para poder hacer una búsqueda

Detalle de los recorridos...

Fallo la consulta

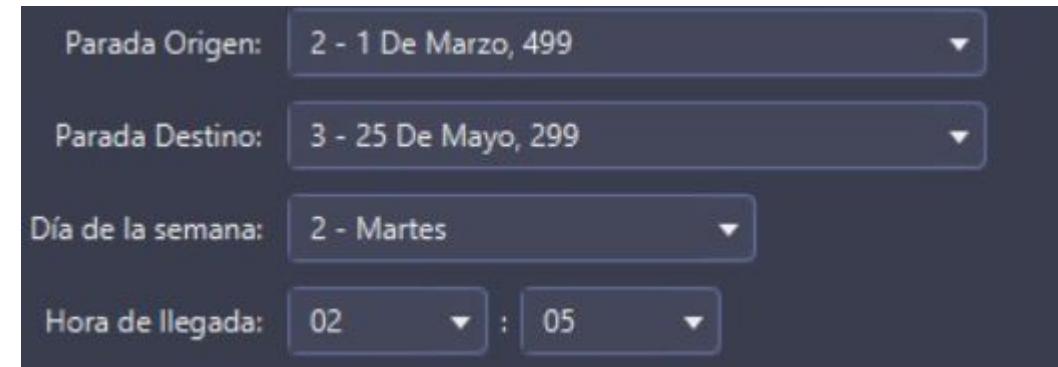


Parada Origen: 2 - 1 De Marzo, 499

Parada Destino: 3 - 25 De Mayo, 299

Día de la semana: 2 - Martes

Hora de llegada: 02 : 05



3er Incremento

Consulta de Recorridos - Colectivos
Seleccione los parámetros de consulta

Parada Origen: 44 - España 1/2a, 1660
Parada Destino: 47 - España 1/2a, 928
Día de la semana: 3 - Miércoles
Hora de llegada: 16 : 41

Consultar Recorridos

Vista ES

Opciones de Recorrido:

[1] L1I

[2] L5R

Consulta completada

The map displays a detailed street layout of Mar del Plata, Argentina, with numerous streets labeled in Spanish. Two bus routes are highlighted: L1I (blue line) and L5R (yellow line). Route L1I starts at Parada 44 (España 1/2a, 1660), goes north along Avenida Gales, turns east onto 20 de Junio, then south onto Sarmiento, ending at Parada 47 (España 1/2a, 928). Route L5R starts at Parada 47, goes north along 20 de Junio, then west onto Estivariz, ending at Parada 44. Other labeled locations include Plaza España, Estación Central, and various schools and institutions like Colegio 703 José Toschke and Escuela N° 780 "Leandro N. Alem". The map also shows several parks and landmarks.