

Manual de Usuario del Sistema - Proyecto de Colectivos Urbanos

Índice

1. Introducción.....	2
2. Requisitos previos.....	2
3. Instalación y ejecución.....	2
3.1 Clonar el repositorio en tu máquina.....	2
3.2 Importar el proyecto en tu IDE.....	2
4. Configuración del sistema.....	3
4.1 Archivo application.properties.....	3
4.2 Archivo jdbc.properties.....	3
4.3 Archivo factory.properties.....	4
5. Ejecución de la aplicación.....	4
6. Interfaz gráfica.....	5
6.1 Parada de origen.....	5
6.2 Parada de destino.....	6
6.3 Día.....	7
6.4 Hora.....	7
6.5 Recorrido.....	8
6.6 Volver.....	9
7. Persistencia de datos.....	9
7.1 Registro de eventos.....	9
8. Ejemplos de ejecución.....	9
9. Enlaces de referencia.....	11

1. Introducción

El sistema ISFPP 2025 es una aplicación desarrollada en Java bajo el patrón Modelo–Vista–Controlador (MVC), que permite calcular y visualizar distintos recorridos posibles entre paradas del servicio de transporte público.

La aplicación integra un cálculo genérico (Calculo), acceso a datos mediante DAO Factory, y una interfaz gráfica JavaFX para la interacción con el usuario.

El usuario tiene la capacidad de consultar recorridos al especificar el origen, el destino, el día y la hora deseados. La información necesaria para estas consultas se obtiene tanto de archivos de texto como de una base de datos PostgreSQL.

2. Requisitos previos

- Java (JDK 17 o superior) instalado.
- Eclipse IDE o entorno compatible con proyectos Java.
- PostgreSQL 13 o superior, si se utiliza la persistencia de base de datos.
- Librerías incluidas en la carpeta lib/ (JavaFX y Log4j2).
- Acceso de lectura a los archivos de datos en doc/ o conexión a la base de datos configurada en jdbc.properties.

3. Instalación y ejecución

3.1 Clonar el repositorio en tu máquina

git clone https://github.com/moises-kutnich/Colectivo_POO_2025

3.2 Importar el proyecto en tu IDE

Importa el proyecto en tu IDE preferido. El IDE cargará automáticamente el código fuente, las librerías y los recursos de configuración.

4. Configuración del sistema

4.1 Archivo application.properties

Ubicado en config/application.properties.

Contiene las rutas a los archivos de datos (si se usa modo Secuencial) y los parámetros de ejecución de la aplicación.

Ejemplo:

```
datos.parada=doc/parada_PM.txt
datos.linea=doc/linea_PM.txt
datos.tramo=doc/tramo_PM.txt
datos.frecuencia=doc/frecuencia_PM.txt
```

Las rutas de los archivos en el ejemplo (parada_PM.txt) utilizan el sufijo _PM, que corresponde a la configuración inicial de los datos para la ciudad de Puerto Madryn.

Si su aplicación requiere utilizar un conjunto de datos diferente para otra localidad, es obligatorio que reemplace el sufijo _PM por el identificador de la nueva localidad en las cuatro líneas mostradas.

4.2 Archivo jdbc.properties

Ubicado en config/jdbc.properties.

Define la conexión a la base de datos PostgreSQL, incluyendo host, nombre de base, usuario y contraseña.

Ejemplo:

```
jdbc.url=jdbc:postgresql://localhost:5432/isfpp
jdbc.user=postgres
jdbc.password=admin
```

4.3 Archivo factory.properties

Ubicado en la raíz del proyecto (src/factory.properties).

Determina qué tipo de implementación utilizará el sistema para los DAO.

```
PARADA=colectivo.dao.postgresql.ParadaDatabaseDAO  
LINEA=colectivo.dao.postgresql.LineaDatabaseDAO  
TRAMO=colectivo.dao.postgresql.TramoDatabaseDAO  
FRECUENCIA=colectivo.dao.postgresql.FrecuenciaDatabaseDAO
```

Para cambiar al modo secuencial (archivos de texto), basta con reemplazar las líneas por:

```
PARADA=colectivo.dao.secuencial.ParadaSecuencialDAO  
LINEA=colectivo.dao.secuencial.LineaSecuencialDAO  
TRAMO=colectivo.dao.secuencial.TramoSecuencialDAO  
FRECUENCIA=colectivo.dao.secuencial.FrecuenciaSecuencialDAO
```

El sistema detecta automáticamente la configuración y crea las instancias DAO correspondientes mediante la Factory genérica.

5. Ejecución de la aplicación

Para iniciar el sistema, ejecutar la clase:

```
colectivo.aplicacion.AplicacionPrincipal
```

La clase inicializa el CoordinadorApp, carga la configuración, crea los objetos de negocio, conecta con los DAO y lanza la interfaz JavaFX.

6. Interfaz gráfica

Una vez ejecutada la aplicación, se abrirá la interfaz gráfica. Esta misma permite al usuario realizar consultas sobre los recorridos posibles entre paradas del sistema de transporte. A continuación se detallan las funcionalidades disponibles.

Sistema de Colectivos Urbanos

Archivo

Consulta de recorridos de colectivos urbanos

Parada origen:

Parada destino:

Día:

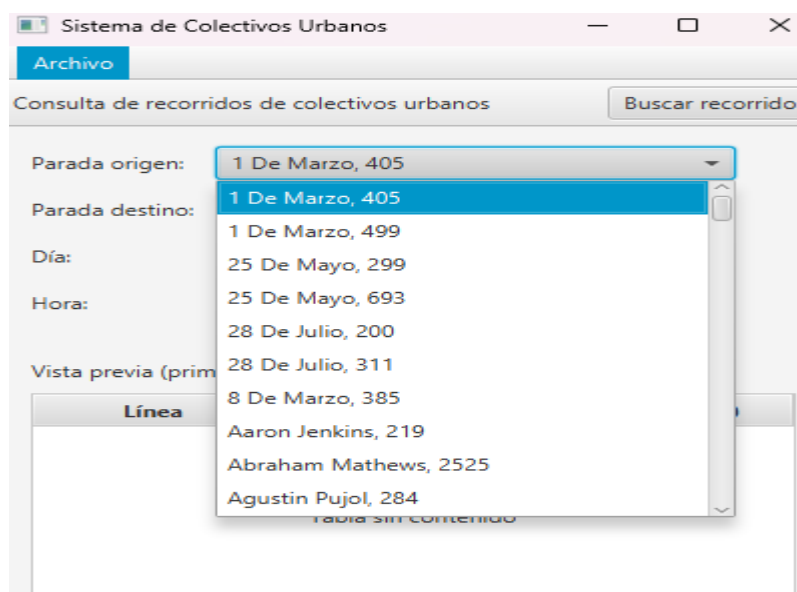
Hora: :

Vista previa (primera alternativa)

Línea	Hora Salida	Duración (min)
Tabla sin contenido		

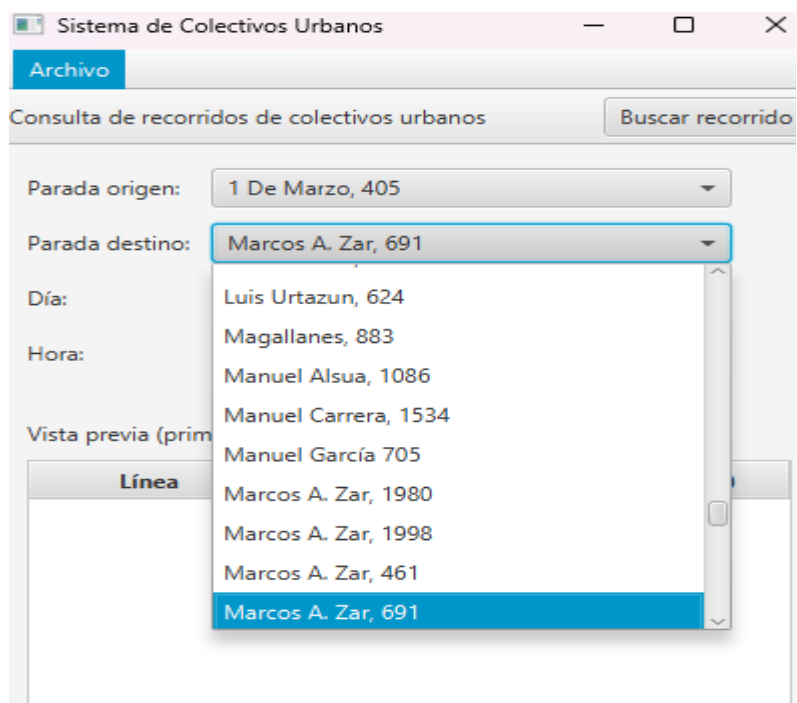
6.1 Parada de origen

El campo "Parada origen" es un menú desplegable que permite seleccionar el punto de partida de su viaje, ofreciendo una lista de todas las paradas disponibles en el sistema. Esta selección define dónde inicia la consulta del recorrido del colectivo urbano.



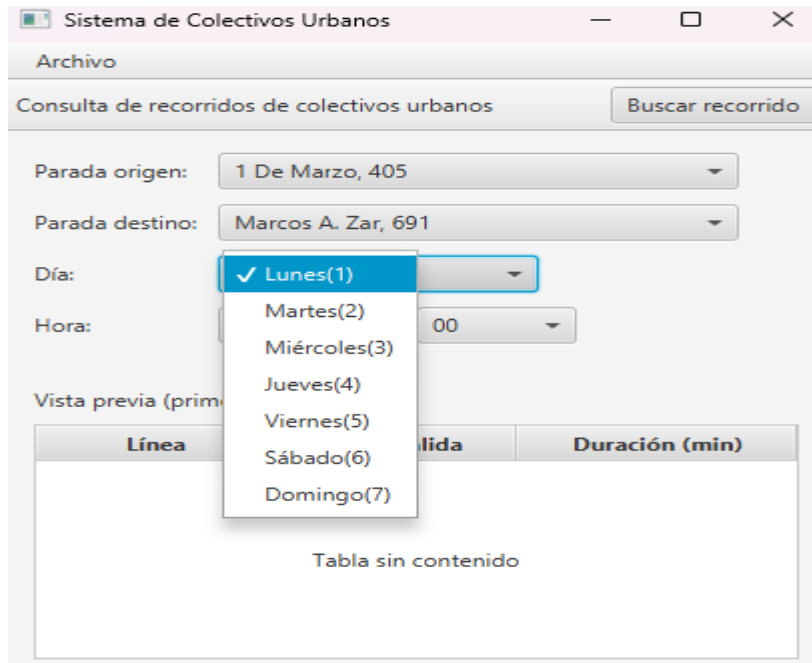
6.2 Parada de destino

El campo "Parada destino" es un menú desplegable que le permite seleccionar el punto final deseado para su viaje, eligiendo una dirección específica entre la lista de paradas disponibles.



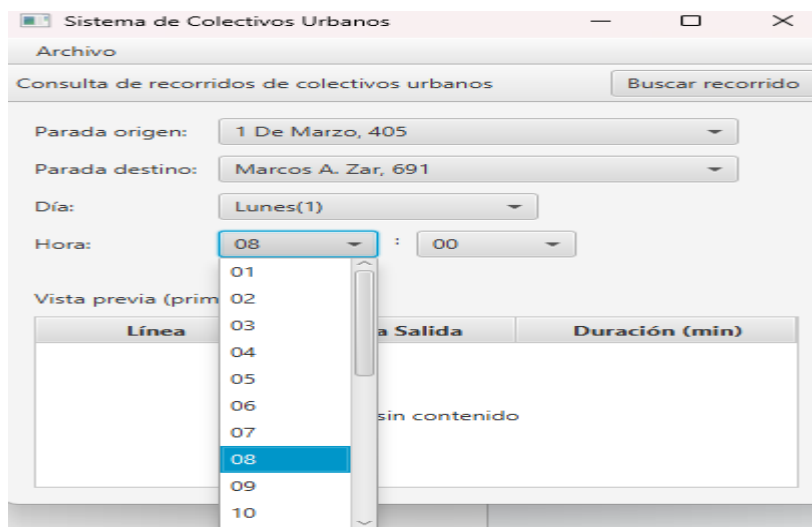
6.3 Día

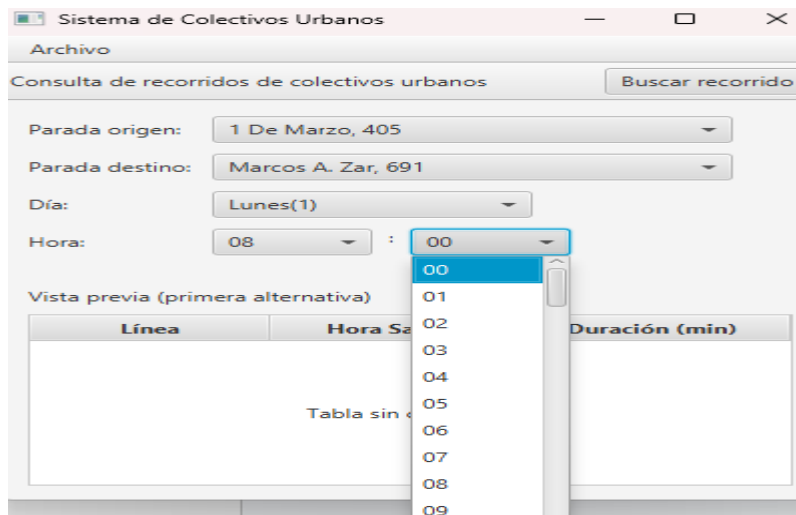
El campo "Día" es un desplegable que permite seleccionar el día de la semana específico para el que desea consultar el recorrido, eligiendo entre opciones como "Lunes(1)" hasta "Domingo(7)".



6.4 Hora

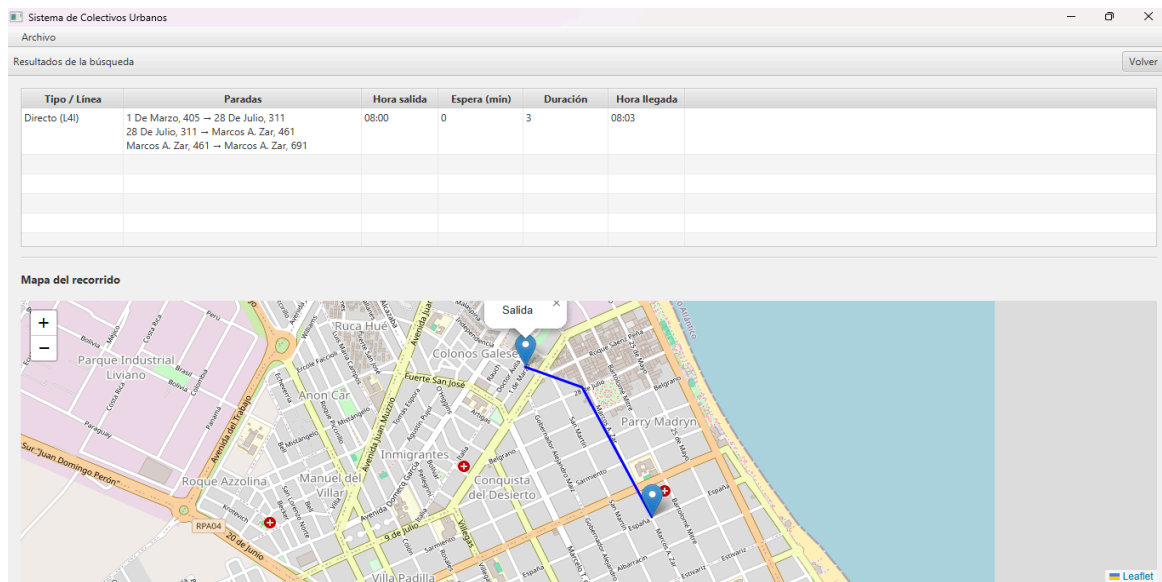
El campo "Hora" se compone de dos desplegables numéricos que le permiten al usuario seleccionar la hora y los minutos exactos en que desea iniciar su viaje.





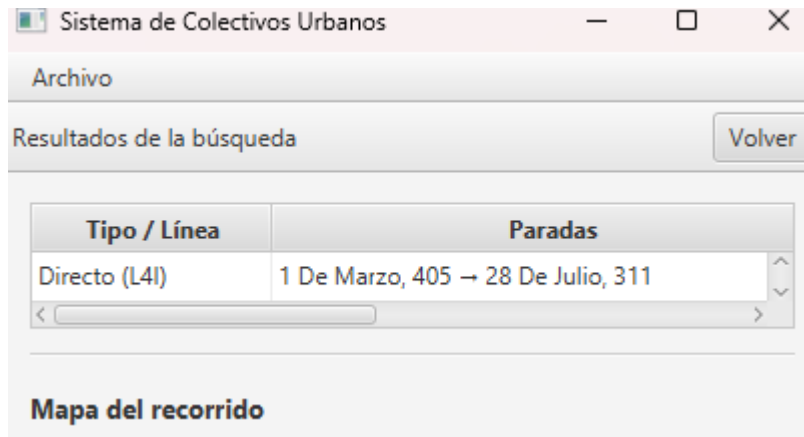
6.5 Recorrido

Al hacer clic en el botón "Buscar recorrido", el sistema presenta la interfaz de Resultados de la búsqueda, compuesta por una tabla y un mapa visual. La Tabla de Resultados muestra una o más opciones de viaje, detallando en cada fila el Tipo de Recorrido, las paradas intermedias, la Hora de Salida, el tiempo de Espera, la Duración total y la Hora de Llegada. El componente Mapa del recorrido ofrece una representación gráfica de la ruta seleccionada. En el mapa se visualiza la trayectoria del colectivo y los puntos específicos de Origen y Destino del viaje. Finalmente, el usuario puede utilizar los controles de zoom del mapa para una mejor inspección visual de la ruta seleccionada.



6.6 Volver

El botón "Volver" se utiliza para retornar a la pantalla de Consulta de recorridos.



7. Persistencia de datos

Los datos se obtienen de acuerdo con la configuración:

- En modo Secuencial, los DAO leen los archivos de texto solo una vez (uso de bandera interna).
- En modo Base de Datos, los DAO realizan consultas SQL a las tablas de PostgreSQL.

7.1 Registro de eventos

El sistema utiliza Log4j2 para registrar eventos en distintos niveles (INFO, WARN, ERROR). Los archivos de log se almacenan en la carpeta logs/.

8. Ejemplos de ejecución

Ejemplo 1: Recorrido con conexión vehicular (L3I – L3R)

Paradas consultadas:

Origen → 1 De Marzo, 405

Destino → Agustín Pujol, 284

Configuración:

- Día de servicio: Lunes
- Hora de llegada deseada: 09:00

Resultado mostrado por el sistema:

Tipo Recorrido	Paradas	Hora Salida	Espera (min)	Duración (min)	Hora Llegada
Conexión (L3I)	1 De Marzo, 405 → 28 De Julio, 311 → Marcelo T. de Alvear, 500	08:25	25	6	08:31
Conexión (L3R)	Dorrego, 700 → 28 De Julio, 200 → 1 De Mayo, 405	09:00	–	7	09:07

Ejemplo 2: Recorrido con conexión vehicular (L2R – L2I)

Paradas consultadas:

Origen → Aarón Jenkins, 219

Destino → Bv, Almirante Brown, 2925

Configuración:

- Día de servicio: Martes
- Hora de llegada deseada: 09:00

Resultado mostrado por el sistema:

Tipo Recorrido	Paradas	Hora Salida	Espera (min)	Duración (min)	Hora Llegada
Conexión (L2R)	Aarón Jenkins, 219 → Chiquichan, 22 → Avenida Roca, 300	08:19	19	20	08:39
Conexión (L2I)	Esquel, 1352 → Alto Río Senguer, 1311 → Paraguay, 1200	08:40	–	23	09:03

9. Enlaces de referencia

- [Documentación JavaFX](#)
- [Apache Log4j 2](#)
- [PostgreSQL Documentation](#)
- [JUnit 5](#)
- [Eclipse IDE](#)