

# Nuestro equipo

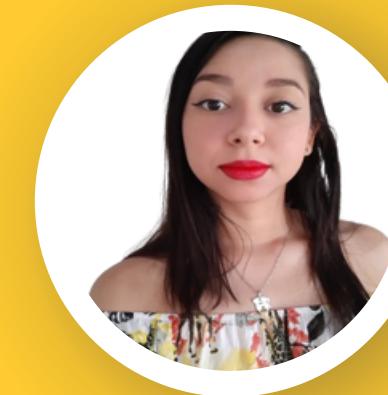
Equipo 3



**Michell Botello**



**Carlos bedoya**



**Nelbis Olivero**



**Esteban cuero**



**Leidy Tacuma**



**Dorian Tenorio**



**Cristian Mora**



**Andres SZ**



**Angelica Pineda**

# UBER

## Equipo 3

Uber es una plataforma tecnológica que revolucionó la forma en que nos movemos por las ciudades, ofreciendo una alternativa flexible y conveniente al transporte tradicional.





# Requerimiento del cliente

Estan necesitando una base de datos conformada por 3 tablas y nos sugieren algunos campos que podrian contener:

**Tabla Usuario**(identificador,nombre usuario, número de telefono)

**Tabla Conductor**(identificador,nombre conductor, matrícula del automovil)

**Tabla Viajes**(identificador, usuario,conductor,matricula,origen, destino)



# Épica

# HU

Historias de usuarios relacionadas con las tablas solicitadas por el cliente

The screenshot shows the Zephyr Scale application interface. On the left, there's a sidebar with sections for 'Epic' (containing 'Backlog (3 incidencias)' with items like SCRUM-5 HU-Usuarios, SCRUM-7 HU-Conductores, and SCRUM-6 HU-Viajes, all marked as 'FINALIZADA' and assigned to 'EPICA-UBERTO'), 'Incidencias sin epic', and '+ Crear epic'. Below this is a section for 'Epica-UberTo'. The main area is titled 'SMARTBEAR Zephyr Scale' and has tabs for 'Casos de Prueba', 'Ejecuciones de Prueba', 'Planes de Pruebas', and 'Informes'. The 'Casos de Prueba' tab is selected, showing a list of test cases with columns for 'P', 'Clave', 'V', 'Nombre', 'Status', and a gear icon. The test cases listed are:

P	Clave	V	Nombre	Status
<input type="checkbox"/>	SCRUM-T4	1.0	Editar campo numero de teléfono en la tabla usuario	APROBADO
<input type="checkbox"/>	SCRUM-T1	1.0	Registrar un usuario	APROBADO
<input type="checkbox"/>	SCRUM-T5	1.0	Seleccionar Datos de Tabla Viaje visualizando datos de Usuario y Conductor	APROBADO
<input type="checkbox"/>	SCRUM-T6	1.0	Seleccionar matriculas terminadas En	APROBADO
<input type="checkbox"/>	SCRUM-T3	1.0	Validar longitud matricula_vehiculo	APROBADO
<input type="checkbox"/>	SCRUM-T2	1.0	Validar tabla conductor	APROBADO

# TC

4 TC Sobre validación de campos  
2 TC validación de consultas

# 1er SPRINT

▾ Tablero Sprint 1 24 sep – 24 sep (6 incidencias)

Definición de las historias de usuario y consolidación de la información necesaria para la ejecución de las pruebas

<input checked="" type="checkbox"/> SCRUM-2 Epica		FINALIZADA	
<input checked="" type="checkbox"/> SCRUM-3 Diagrama		FINALIZADA	
<input checked="" type="checkbox"/> SCRUM-4 Matriz		FINALIZADA	
<input checked="" type="checkbox"/> SCRUM-5 HU-Usuarios	EPICA-UBERTO	FINALIZADA	
<input checked="" type="checkbox"/> SCRUM-6 HU-Viajes	EPICA-UBERTO	FINALIZADA	
<input checked="" type="checkbox"/> SCRUM-7 HU-Conductores	EPICA-UBERTO	FINALIZADA	

+ Crear incidencia

# 2do SPRINT

▾ Tablero Sprint 2 25 sep – 25 sep (2 incidencias)

Creación de la Base de datos y búsqueda mediante las consultas correspondientes

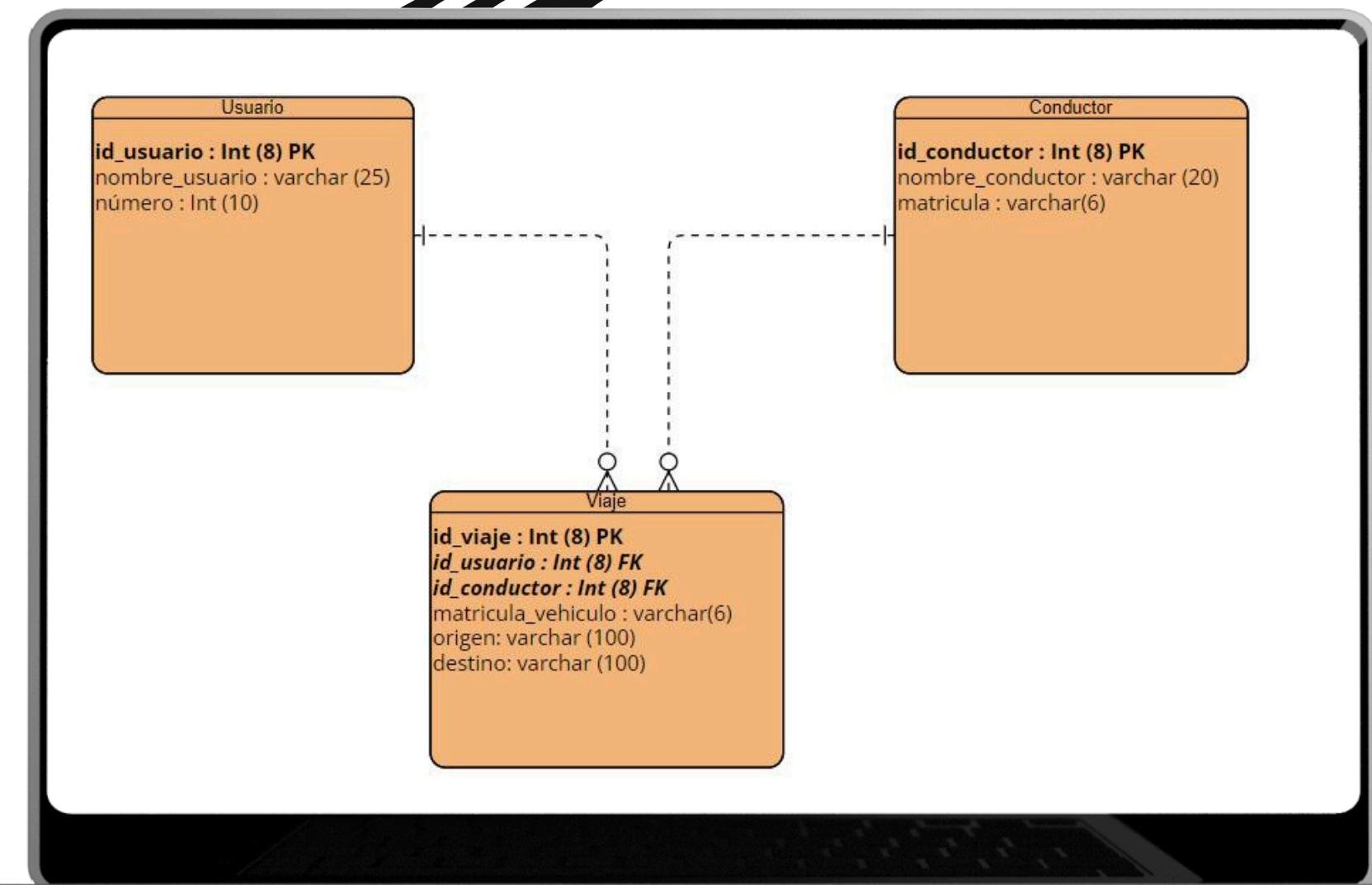
<input checked="" type="checkbox"/> SCRUM-9 BDD	FINALIZADA	
<input checked="" type="checkbox"/> SCRUM-10 Consultas	FINALIZADA	

+ Crear incidencia

# Diagrama

# Diagrama ERD

El diagrama representa un modelo donde un usuario puede realizar muchos viajes, un conductor puede conducir muchos viajes, y un viaje está asociado tanto con un usuario como con un conductor.



## Relaciones

### **Usuario - Viaje:**

Un usuario puede tener múltiples viajes, pero cada viaje está asociado con un solo usuario. Esta es una relación uno a muchos (1), donde Usuario se relaciona con Viaje a través de la clave foránea **id\_usuario**.

### **Conductor - Viaje:**

Un conductor puede participar en múltiples viajes, pero cada viaje tiene un solo conductor. Esta es una relación uno a muchos (1), donde Conductor se relaciona con Viaje a través de la clave foránea **id\_conductor**.

**BBDD**

---



# Tabla Usuario

The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with icons for zoom, refresh, and other database operations, followed by a blue "Run SQL" button and a "Available Tables" link. Below the toolbar is a table named "Usuario". The table has three columns: "id\_usuario", "nombre\_usuario", and "numero". The first row is labeled "empty". At the bottom left, a message says "set is empty.".

The screenshot shows the MySQL Workbench interface with the "Input" tab selected. It displays the SQL code for creating the "Usuario" table:

```
CREATE TABLE "Usuario" (
    "id_usuario" INT(8) NOT NULL,
    "nombre_usuario" VARCHAR(25) NOT NULL,
    "numero" INT(10) NOT NULL,
    PRIMARY KEY ("id_usuario")
)
```

Below the input area is an "Output" section which displays the message: "SQL query successfully executed. However, the result set is empty."

## Scripts:

```
CREATE TABLE "Usuario" (
    "id_usuario" INT(8) NOT NULL,
    "nombre_usuario" VARCHAR(25) NOT NULL,
    "numero" INT(10) NOT NULL,
    PRIMARY KEY ("id_usuario")
)
```

# Tabla Conductor

The screenshot shows the MySQL Workbench interface. On the left, there are two tables listed: 'Conductor' and 'Usuario'. The 'Conductor' table has columns: 'id\_conductor' (int(8)), 'nombre\_conductor' (varchar(20)), and 'matricula' (varchar(6)). The 'Usuario' table has columns: 'id\_usuario' (int(8)), 'nombre\_usuario' (varchar(25)), and 'numero' (int(10)). In the center, the 'Input' tab contains the SQL script for creating the 'Conductor' table:

```
CREATE TABLE "Conductor" (
    "id_conductor" INT(8) NOT NULL,
    "nombre_conductor" VARCHAR(20) NOT NULL,
    "matricula" VARCHAR(6) NOT NULL,
    PRIMARY KEY ("id_conductor")
)
```

The 'Output' tab shows the message: "SQL query successfully executed. However, the result set is empty."

## Scripts:

```
CREATE TABLE "Conductor" (
    "id_conductor" INT(8) NOT NULL,
    "nombre_conductor" VARCHAR(20) NOT NULL,
    "matricula" INT(6) NOT NULL,
    PRIMARY KEY ("id_conductor")
)
```

The screenshot shows the MySQL Workbench interface with the 'Run SQL' button highlighted. Below it, the 'Available Tables' section lists the 'Conductor' and 'Usuario' tables. Each table has three columns: 'id\_conductor' (empty), 'nombre\_conductor' (empty), and 'matricula' (empty) for 'Conductor', and 'id\_usuario' (empty), 'nombre\_usuario' (empty), and 'numero' (empty) for 'Usuario'. At the bottom, the message "The result set is empty." is displayed.

# Tabla Viaje

Available Tables

Conductor
<code>id_conductor</code>
<code>nombre_conductor</code>
<code>matricula</code>
empty

Usuario
<code>id_usuario</code>
<code>nombre_usuario</code>
<code>numero</code>
empty

Viaje
<code>id_viaje</code>
<code>id_usuario</code>
<code>id_conductor</code>
<code>matricula_vehiculo</code>
empty

ver, the result set is empty.

Input

```
CREATE TABLE "Viaje"(
    "id_viaje" INT(8) NOT NULL,
    "id_usuario" INT(8) NOT NULL,
    "id_conductor" INT(8) NOT NULL,
    "matricula_vehiculo" VARCHAR(6) NOT NULL,
    "origen" VARCHAR (100) NOT NULL,
    "destino" VARCHAR (100) NOT NULL,
    PRIMARY KEY ("id_viaje"),
    FOREIGN KEY ("id_usuario") REFERENCES "Usuario"("id_usuario"),
    FOREIGN KEY ("id_conductor") REFERENCES "Conductor"("id_conductor")
)
```

## Output

SQL query successfully executed. However, the result set is empty.

## Scripts:

```
CREATE TABLE "Viaje" (
    "id_viaje" INT(8) NOT NULL,
    "id_usuario" INT(8) NOT NULL,
    "id_conductor" INT(8) NOT NULL,
    "matricula_vehiculo"VARCHAR(6) NOT NULL,
    "origen" VARCHAR (100) NOT NULL,
    "destino" VARCHAR (100) NOT NULL,
    PRIMARY KEY ("id_viaje"),
    FOREIGN KEY ("id_usuario") REFERENCES "Usuario"("id_usuario"),
    FOREIGN KEY ("id_conductor") REFERENCES "Conductor"("id_conductor")
)
```

# INSERT USUARIOS

The screenshot shows a database interface with three main sections: Input, Available Tables, and Output.

**Input:** Displays an SQL query for inserting data into the 'Usuario' table:

```
INSERT INTO Usuario VALUES
(1, "Juan Pérez", 3001234567),
(2, "María Gómez", 3019876543),
(3, "Carlos Ramírez", 3024567890),
(4, "Laura Fernández", 3031239876),
(5, "José Martínez", 3046543210),
(6, "Andrés Rodríguez", 3058765432),
(7, "Paula Salazar", 3062345678),
(8, "David Ospina", 3073456789),
(9, "Luisa Herrera", 3089876543),
(10, "Alejandro Montoya", 3091234567)
```

**Available Tables:** Shows the 'Usuario' table with 10 rows of data:

id_usuario	nombre_usuario	numero
1	Juan Pérez	3001234567
2	María Gómez	3019876543
3	Carlos Ramírez	3024567890
4	Laura Fernández	3031239876
5	José Martínez	3046543210
6	Andrés Rodríguez	3058765432
7	Paula Salazar	3062345678
8	David Ospina	3073456789
9	Luisa Herrera	3089876543
10	Alejandro Montoya	3091234567

**Output:** Displays the message: "SQL query successfully executed. However, the result set is empty."

# INSERT VIAJES

The screenshot shows a database interface with three main sections: Input, Available Tables, and Output.

**Input:** Displays an SQL query for inserting data into the 'Viaje' table:

```
INSERT INTO Viaje VALUES
(1, 1, 1, "ABC123", "Av. Siempre Viva 742, Springfield", "Calle 123, 1º piso, Bogotá"),
(2, 1, 2, "XYZ789", "Carrera 7, No. 45-78, Cali", "Calle 10, No. 25-60, Medellín"),
(3, 3, 4, "GHI789", "Av. Libertador 123, Barranquilla", "Carrera 15, No. 78-45, Bucaramanga"),
(4, 4, 3, "DEF456", "Calle 34 No. 12-45, Bogotá", "Carrera 12, No. 23-56, Cartagena"),
(5, 5, 5, "JKL012", "Carrera 50 No. 15-30, Medellín", "Calle 8, No. 12-50, Bogotá"),
(6, 7, 6, "PQR678", "Calle 65 No. 45-23, Cali", "Carrera 20, No. 34-60, Pereira"),
(7, 6, 7, "STU901", "Av. Circunvalar 230, Bucaramanga", "Calle 45, No. 67-12, Manizales"),
(8, 9, 8, "VWX234", "Calle 50 No. 12-78, Bogotá", "Carrera 30, No. 56-89, Neiva"),
(9, 8, 9, "YZA567", "Carrera 5 No. 34-45, Barranquilla", "Calle 14, No. 45-12, Cartagena"),
(10, 10, 10, "BCD890", "Calle 78 No. 56-34, Medellín", "Av. Roosevelt 45, No. 67-23, Cali")
```

**Available Tables:** Shows the 'Viaje' table with 10 rows of data:

id_viaje	id_usuario	id_conductor	matricula_vehiculo
1	2	1	ABC123
2	1	2	XYZ789
3	3	4	GHI789
4	4	3	DEF456
5	5	5	JKL012
6	7	6	PQR678

**Output:** Displays the message: "SQL query successfully executed. However, the result set is empty."

# INSERT CONDUCTOR

The screenshot shows a database interface with three main sections: Input, Available Tables, and Output.

**Input:** Displays an SQL query for inserting data into the 'Conductor' table:

```
INSERT INTO Conductor VALUES
(1, "Juan Pérez", "ABC123"),
(2, "María García", "XYZ789"),
(3, "Carlos Rodríguez", "DEF456"),
(4, "Ana Martínez", "GHI789"),
(5, "Luis Sánchez", "JKL012"),
(6, "Diana Castro", "PQR678"),
(7, "Rafael Cárdenas", "STU901"),
(8, "Andrea Mejía", "VWX234"),
(9, "Fernando Giraldo", "YZA567"),
(10, "Carolina Gutiérrez", "BCD890")
```

**Available Tables:** Shows the 'Conductor' table with 10 rows of data:

id_conductor	nombre_conductor	matricula
1	Juan Pérez	ABC123
2	María García	XYZ789
3	Carlos Rodríguez	DEF456
4	Ana Martínez	GHI789
5	Luis Sánchez	JKL012
6	Diana Castro	PQR678
7	Rafael Cárdenas	STU901
8	Andrea Mejía	VWX234
9	Fernando Giraldo	YZA567
10	Carolina Gutiérrez	BCD890

**Output:** Displays the message: "SQL query successfully executed. However, the result set is empty."

# Insertar datos

# Consultas

Input

```
SELECT id_viaje, Usuario.nombre_usuario, Conductor.nombre_conductor, origen, destino FROM Viaje JOIN Usuario on Viaje.id_usuario = Usuario.id_usuario JOIN Conductor on Viaje.id_conductor = Conductor.id_conductor
```

Output

id_viaje	nombre_usuario	nombre_conductor	origen	destino
1	Maria Gómez	Juan Pérez	Av. Siempre Viva 742, Springfield	Calle 123, 1º piso, Bogotá
2	Juan Pérez	Maria García	Carrera 7, No. 45-78, Cali	Calle 10, No. 25-60, Medellín
3	Carlos Ramírez	Ana Martínez	Av. Libertador 123, Barranquilla	Carrera 15, No. 78-45, Bucaramanga
4	Laura Fernández	Andrés Rodríguez	Carrera 12, No. 23-56,	Carrera 10, No. 34-22, Bogotá
5	José Martínez	Paula Salazar	Carrera 9, No. 12-34,	Calle 5, No. 20-10, Bogotá
6	Andrés Rodríguez	David Ospina	Carrera 8, No. 21-45,	Calle 15, No. 30-20, Bogotá
7	Paula Salazar	Luisa Herrera	Carrera 7, No. 18-30,	Calle 20, No. 25-15, Bogotá
8	David Ospina	Alejandro Montoya	Carrera 6, No. 15-25,	Calle 10, No. 10-20, Bogotá
9	Luisa Herrera		Carrera 5, No. 12-20,	
10	Alejandro Montoya		Carrera 4, No. 10-18,	

Available Tables

id_usuario	nombre_usuario	numero
1	Juan Pérez	3001234567
2	Maria Gómez	3019876543
3	Carlos Ramírez	3024567890
4	Laura Fernández	3031239876
5	José Martínez	3046543210
6	Andrés Rodríguez	3058765432
7	Paula Salazar	3062345678
8	David Ospina	3073456789
9	Luisa Herrera	3089876543
10	Alejandro Montoya	3091234567

Viaje

id_viaje	id_usuario	id_conductor	matricula_vehiculo
1	2	1	ABC123
2	1	2	XYZ789

Input

```
SELECT * FROM Conductor WHERE matricula LIKE "%9"
```

Output

id_conductor	nombre_conductor	matricula
2	Maria García	XYZ789
4	Ana Martínez	GHI789

Available Tables

id_viaje	id_usuario	id_conductor	matricula_vehiculo
1	2	1	ABC123
2	1	2	XYZ789
3	3	4	GHI789
4	4	3	DEF456
5	5	5	JKL012
6	7	6	PQR678

SELECT  
JOIN ON

SELECT,  
FROM,  
WHERE,  
LIKE

## Cycle time

Understand your project's Cycle Time to see how quickly you ship value. Your cycle time is the time it takes an issue to go from first commit to last production deployment based on development and CI/CD data. [Learn more](#)

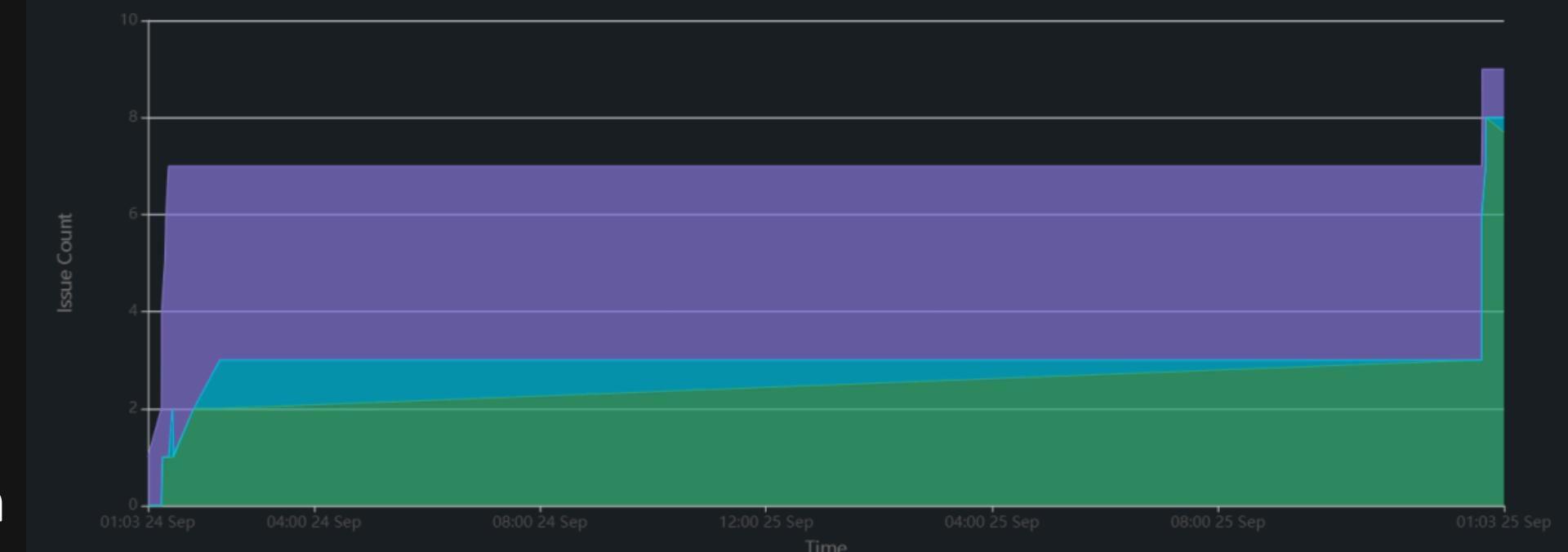


## Cycle time

### Cumulative flow diagram

Date filter  
From date: 9/24/2024  
To date: 9/25/2024

To Do   In Progress   Done



# Reporte

### Cumulative flow diagram

**Gracias  
por su atención**



# Links

[https://mitchellbotello-172721846666.atlassian.net/jira/software/projects/SCRUM/boards/1?  
atlOrigin=eyJpljoiMzkzOTdmODUyYzg0NGVIZmJhYTYxYmVjNzAwYjhiYTQiLCJwljoiaiJ9](https://mitchellbotello-172721846666.atlassian.net/jira/software/projects/SCRUM/boards/1?atlOrigin=eyJpljoiMzkzOTdmODUyYzg0NGVIZmJhYTYxYmVjNzAwYjhiYTQiLCJwljoiaiJ9)