

Universidad Tecnológica Nacional

# **Soporte a la Gestión de Datos con Programación Visual**

4º Año Ingeniería en Sistemas de Información

Trabajo Práctico Integrador

Comisión nº: 401 - Grupo nº: 03

## **Integrantes:**

- Ferroni, Lisandro. Legajo: 44161. lisandro.ferroni@gmail.com
- Ballesterio, Martín. Legajo: 45463. martin.ballesterio@hotmail.com

<b>Introducción</b>	<b>3</b>
<b>Descripción del sistema</b>	<b>3</b>
<b>Características del sistema</b>	<b>4</b>
<b>Tecnologías utilizadas</b>	<b>4</b>
<b>Diagrama Entidad Relación</b>	<b>5</b>
<b>Esquema Arquitectura</b>	<b>5</b>
<b>User stories</b>	<b>6</b>
User Story 1 - Pestaña Paradas de Colectivo Parte 1	6
User Story 2 - Pestaña Paradas de Colectivo Parte 2	8
User Story 3 - Pestaña Cuadro de Horarios	9
User Story 4 - Store Procedure CuandoLlego	11
User Story 5 - Informes Pestaña Boletos	12
<b>Referencias</b>	<b>15</b>

# Introducción

Presentamos: **Cuando Llegó**, una herramienta orientada al estudio y análisis del funcionamiento del transporte público de la ciudad de Rosario, para poder tomar decisiones que permitan mejorar el rendimiento tanto a nivel económico como funcional de cara al usuario. La metodología implementada consiste en realizar una sucesión de informes y gráficos en base al sistema de transporte público de la ciudad de Rosario recabando información del mismo y simulando la utilización que tienen los pasajeros.

## Descripción del sistema

Parte de la información para el desarrollo del sistema es obtenida desde el sitio web del sistema "*¿Cuándo llega?*" de la Municipalidad de Rosario. A partir del mismo se pueden identificar todas las **líneas de colectivo** que se encuentran activas en la ciudad, junto con sus **recorridos**, **paradas** identificadas por intersecciones de **calles** y **horarios** que deben cumplir las unidades. Toda esta información es recopilada desde el código HTML que se inspecciona cuando se visita el sitio del ETR (*Ente de la Movilidad de Rosario*). Se monitorizan las peticiones HTTP que realiza el mismo al interactuar con los selectores del formulario y se comprende la estructura de la información disponible públicamente. Esta se almacenará la base de datos del sistema, previamente diseñada con tablas relacionadas que harán posible las diferentes consultas para generar reportes que se detallarán más adelante.

Para completar la información necesaria para cumplir con los objetivos del sistema a desarrollar, es necesario conocerla cantidad de pasajeros que ascienden en cada parada a cada unidad, y en qué momento. Esto nos permitirá estimar la densidad de personas en los colectivos, y a partir de allí continuar con el análisis establecido por el trabajo. Dichos datos pueden ser obtenidos mediante una fuente interna de la municipalidad, ya que han sido medidos en el pasado para determinadas unidades la cantidad de boletos que se marcan a lo largo del recorrido. En caso de no contar con esta fuente, se procederá a simular los datos.

# Características del sistema

Se confeccionarán reportes en forma de gráficos con el objetivo de visualizar diferentes características sobre el funcionamiento del sistema de transporte, y a partir de allí analizar posibles medidas para la optimización del servicio. Se le permitirá al usuario ingresar los parámetros mediante los cuales se crearán las consultas a la base de datos mediante GraphQL, tales como el área geográfica para conocer la densidad de pasajeros por barrio, franja horaria, días de la semana, líneas de colectivo, etc.

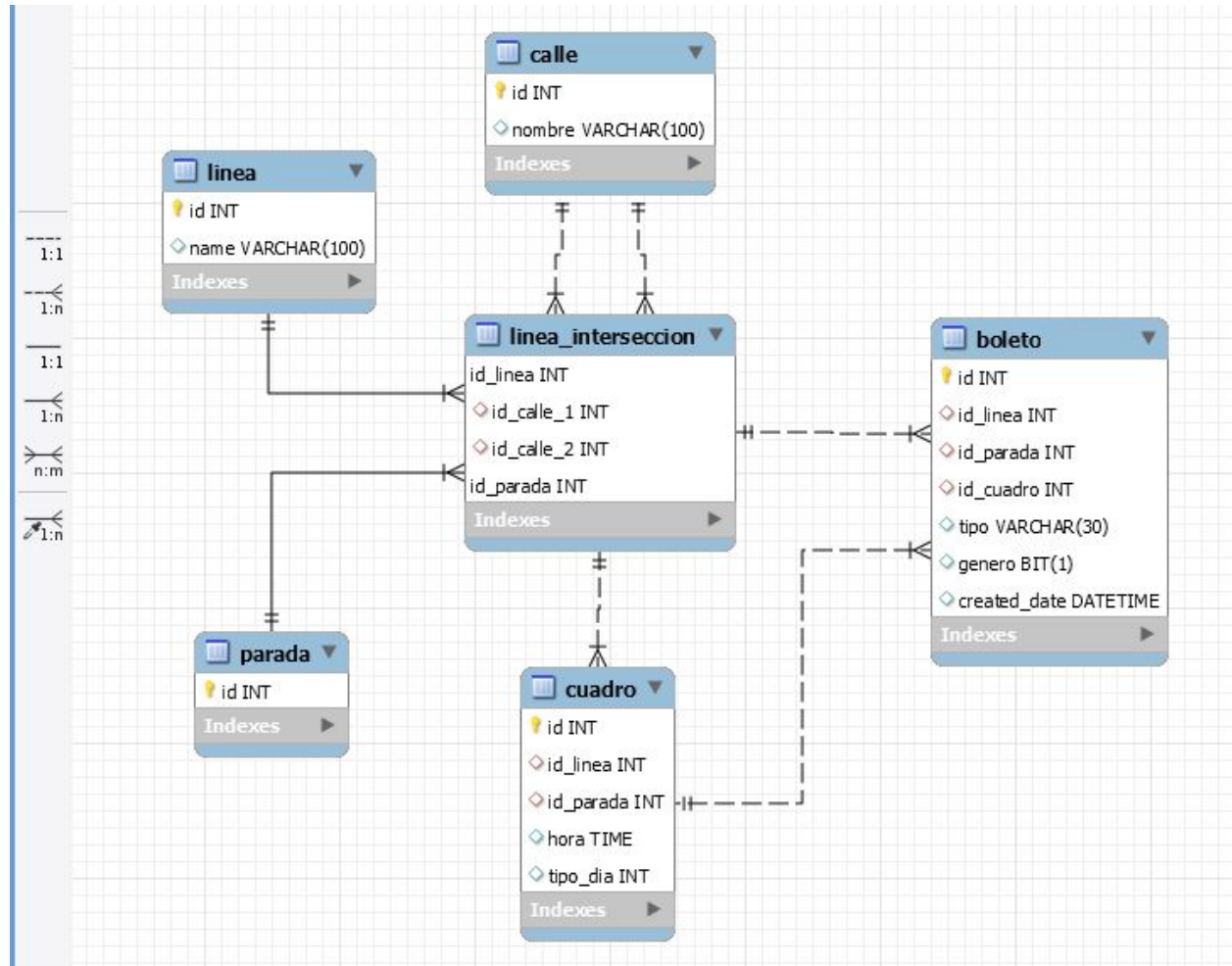
Los reportes podrán ser utilizados para:

- Auditar el servicio de transporte brindado
- Destacar qué zonas de la ciudad tienen mayor flujo de pasajeros
- Detectar cuándo la demanda excede la oferta
- Estimar el horario en el que pasa un colectivo por una parada sin servicio de GPS
- Uso del servicio durante el día y la noche según el género
- Tipos de boletos marcados.
- Uso del transporte en un período establecido
- Comparación empírica del servicio brindado ante el cuadro de horarios establecido.

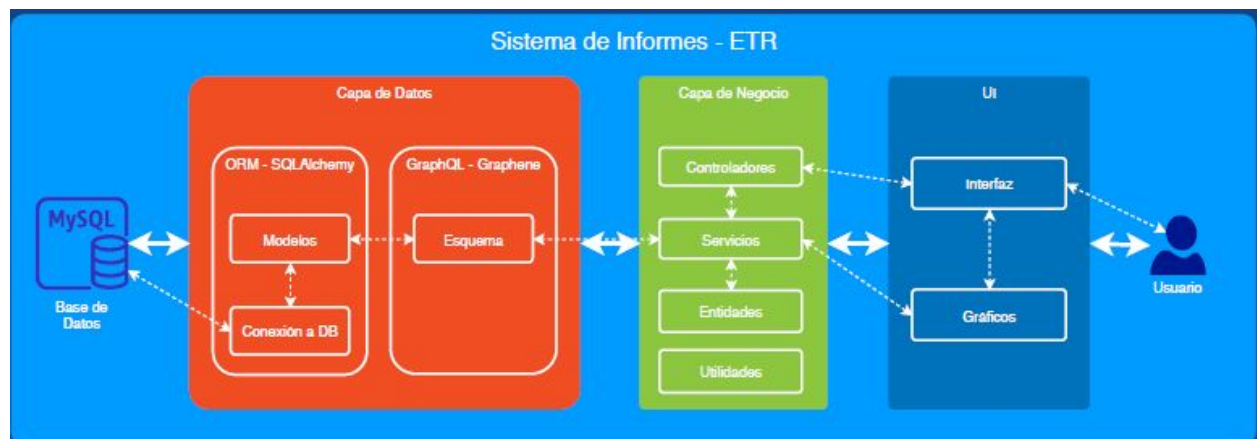
# Tecnologías utilizadas

El lenguaje de programación para construirlo es Python 3.x, respetando el modelo de capas. Se utilizará una base de datos diseñada en MySQL, mediante el ORM llamado SQLAlchemy. Se utilizará la librería GraphQL de Python llamada Graphene. Para el servidor web se seleccionó Flask como framework de aplicación web, junto a HTML y Javascript para el desarrollo de la interfaz. Los gráficos de los reportes se crean con una librería llamada ChartJS.

## Diagrama Entidad Relación



## Esquema Arquitectura



# User stories

## User Story 1 - Pestaña Paradas de Colectivo Parte 1

Como usuario

Quiero poder consultar cuándo llegó un colectivo a una parada, mediante línea de colectivo y número de parada

Para enterarme el tiempo aproximado de arribo

### **Criterios de Aceptación :**

1. En la pestaña de Paradas de Colectivo debe haber un desplegable 'Línea de Colectivo', y cuando se le hace click se deben mostrar todas las líneas existentes.
2. En la pestaña de Paradas de Colectivo, debe haber un cuadro de texto 'Parada' donde se pueda escribir el número de parada.
3. Cuando Línea de Colectivo y Parada estén completos y al hacer click en el botón 'Consultar' se debe mostrar en pantalla el tiempo estimado de llegada del colectivo consultado.
  - a. Si no se encuentra un colectivo llegando a la parada, se debe mostrar el mensaje 'Próximo servicio no encontrado.'

Snapshot:

 Cuando llegó

<

## Paradas de colectivo - Cuando llegó

[Home](#) [Paradas de colectivo](#) [Cuadro de horarios](#)

LINEA DE COLECTIVO

Seleccionar line 

Seleccionar linea de

101

102N

102R

103

106

107

110

112

PARADA

|



CONSULTAR

## User Story 2 - Pestaña Paradas de Colectivo Parte 2

Como usuario

Quiero obtener el número de parada seleccionando las calles de una parada de colectivo para una línea elegida

Para luego poder consultar cuando llega el colectivo a esa parada

### Criterios de Aceptación:

1. En la pestaña de Paradas de Colectivo debe haber un desplegable 'Calle'. Cuando se haya elegido una Línea de colectivo y al hacer click en el desplegable, se deben mostrar todas las calles recorridas por la Línea elegida.
2. En la pestaña de Paradas de Colectivo debe haber un desplegable 'Calle Intersección'. Cuando se haya elegido la calle principal en el desplegable del punto 1, y al hacer click en el desplegable de Calle Intersección se deben listar todas las calles que intersectan con la calle principal y disponen de parada de colectivo.
3. Al completar los desplegables 'Calle' y 'Calle Intersección', se debe auto completar el cuadro de texto 'Parada' con el número de parada correspondiente a dichas calles.

### Snapshots:

[Home](#) [Paradas de colectivo](#) [Cuadro de horarios](#)

LINEA DE COLECTIVO

101 NEGRO

CALLE

A. SABIN - ROSARIO

CALLE INTERSECCIÓN

Seleccionar parada prim

PARADA

1104

CONSULTAR

Próximo servicio en: 2 minutos.

[Home](#) [Paradas de colectivo](#) [Cuadro de horarios](#)

LINEA DE COLECTIVO

101 NEGRO

CALLE

PELLEGRINI - R

CALLE INTERSECCIÓN

BUENOS AIRES

PARADA

1916

CONSULTAR

Próximo servicio no encontrado.



## User Story 3 - Pestaña Cuadro de Horarios

Como usuario

Quiero obtener el cuadro de horarios de una línea de colectivo

Para poder comparar los tiempos de arribo planeados de los empíricos

### Criterios de Aceptación:

1. En una nueva pestaña Cuadro de Horarios, se deben disponer los mismos desplegables y cuadros de texto que en la pestaña de Paradas de colectivo: Línea de colectivo, Calle, Calle intersección, Parada, teniendo estos el mismo comportamiento que en dicha pestaña.
2. Una vez completados Línea de colectivo y Parada, al hacer click en el botón consultar se debe mostrar una tabla con los siguientes encabezados:

Encabezado	Tipo de dato	Descripción	Ejemplo
Línea	string	Numero de línea	101 NEGRO
Parada	Int	Numero de parada	1104
Horario cuadro de horario	HH:MM	Horario en el cual arriba el colectivo según el cuadro de horarios	05:18
Horario empírico	HH:MM	Horario en el cual arriba el colectivo determinado empíricamente con arribos anteriores	05:26
Delta tiempo	string	Diferencia de tiempo entre Horario de cuadro de horario y Horario empírico	8 minutos

Snapshot:

# Cuadro de Horarios

LINEA DE COLECTIVO

101 NEGRO

CALLE

A. SABIN - ROSARIO

CALLE INTERSECCIÓN

Seleccionar parada principal

PARADA

1104

CONSULTAR

Linea	Parada	Horario cuadro de horario	Horario empírico	Delta tiempo
101 NEGRO	1104	05:18	05:26	8 minutos
101 NEGRO	1104	06:35	06:41	7 minutos
101 NEGRO	1104	07:23	07:28	6 minutos
101 NEGRO	1104	08:10	08:15	6 minutos
101 NEGRO	1104	08:55	09:02	7 minutos
101 NEGRO	1104	09:40	09:47	8 minutos
101 NEGRO	1104	10:26	10:32	6 minutos

## User Story 4 - Store Procedure CuandoLlego

Como usuario

Quiero disponer un store procedure que permita estimar el horario de llegada de un colectivo  
Para calcular cuánto falta para el próximo arribo

### Criterios de Aceptación:

1. Debe haber en la base de datos 'cuandollego' un store procedure que cumpla con los siguientes puntos:
  - a. Debe devolver en formato INT, la cantidad de segundos faltantes para dos *próximos arribos* de colectivos de una línea determinada para una parada determinada.
    - i. En caso de que un arribo haya ocurrido hace no más de 10 minutos, contabilizará como *próximo arribo*
  - b. Debe tomar como parámetros:
    - i. **deltaDias**: INT, cantidad de días hacia atrás desde dónde contabilizar boletos hasta **fecha**
    - ii. **fecha**: DATETIME, fecha y hora en la cual se realiza la consulta
    - iii. **linea**: INT, Id de la línea
    - iv. **parada**: INT, Id de la parada
  - c. Debe diferenciar días de semana, sábados y domingos. Una consulta hecha un sábado sólo analizará boletos de los días sábados, y así respectivamente con cada tipo de día
    - i. En caso de una consulta hecha, por ejemplo un sábado a las 23:59, debe buscar el próximo arribo sin importar que cambie el tipo de día; en este ejemplo encontraría el próximo servicio un del domingo.

## User Story 5 - Informes Pestaña Boletos

Como usuario

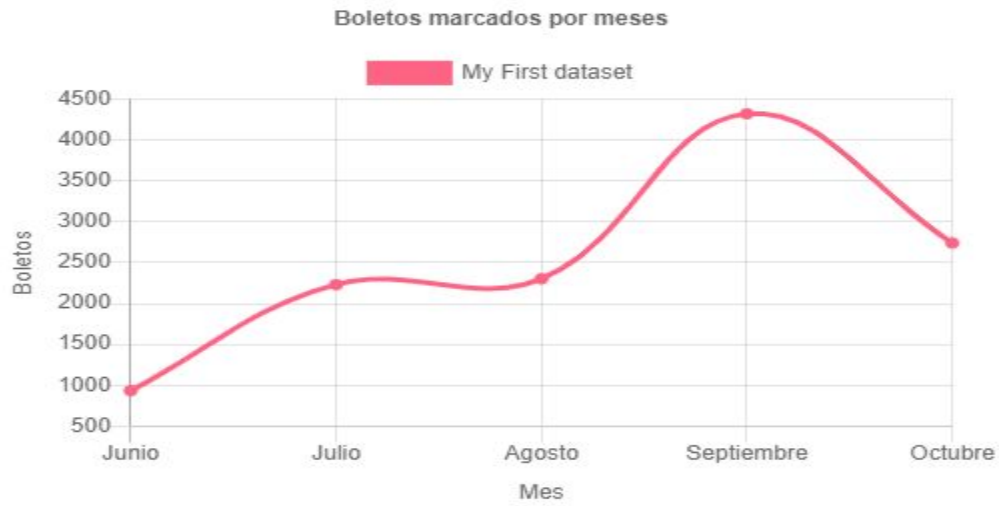
Quiero disponer de gráficos con información sobre los boletos marcados.

Para poder sacar conclusiones rápidamente a partir de visuales

### Criterios de Aceptación:

1. En una nueva pestaña Boletos, se deben disponer los siguientes gráficos:
  - a. **Boletos marcados por Género:** gráfico de torta, con dos variables: Masculino y Femenino, cada una con el porcentaje de boletos marcados por su respectivo género.
  - b. **Boletos marcados por Tipo:** gráfico de torta, con cuatro variables: Medio boleto, Diferencial o Monedas, Normal y Jubilado, cada una con el porcentaje de boletos pertenecientes a su respectivo tipo.
  - c. **Boletos marcados por Momento del día:** gráfico de torta con dos variables: Día y Noche, cada una con el porcentaje de boletos marcados en el Día (7AM - 8PM) y la Noche (8PM - 7AM).
  - d. **Boletos marcados por meses:** gráfico lineal, donde el eje horizontal son los meses, y el eje vertical la cantidad de boletos en total marcados para ese mes.
  - e. **Boletos marcados por género en el día:** gráfico de torta, con dos variables: Hombres y Mujeres, cada uno con el porcentaje de boletos marcados por su respectivo género en los horarios de 7AM a 8PM
  - f. **Boletos marcados por género en la noche:** gráfico de torta, con dos variables: Hombres y Mujeres, cada uno con el porcentaje de boletos marcados por su respectivo género en los horarios de 8PM a 7AM
2. Todos los gráficos del Criterio de Aceptación 1 deben tomar como parámetro los boletos marcados desde hace X cantidad de días atrás, donde X es un parámetro ingresado por el usuario en un cuadro de texto 'Cantidad de días'

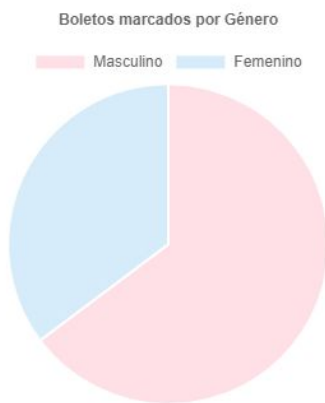
## Snapshots:



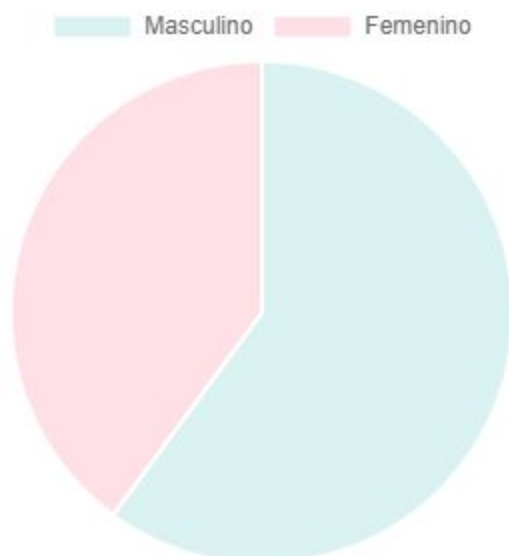
## Informes por boletos

CANTIDAD DE DÍAS

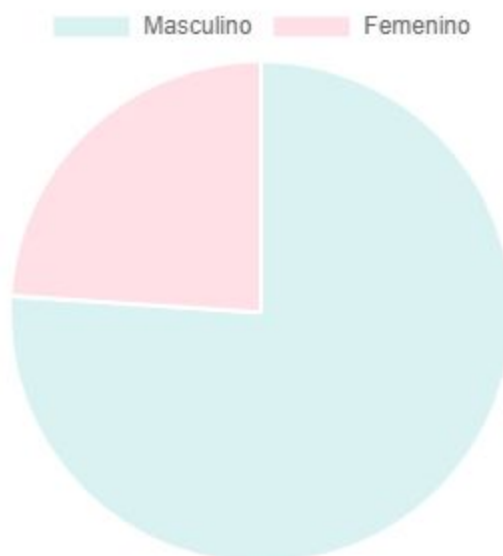
125



Boletos marcados por género durante el día



Boletos marcados por género durante la noche



# Referencias

Cameron Moss. *GraphQL server up and running with 50 lines of python*. Para Medium.com, Agosto 25, 2017.

Jonatas Baldin. *graphql-python Tutorial - Introduction* Para Howtographql.com.