



Apellido y Nombre: ..... Legajo: .....

Arquitectura (35)			Dominio (35)		Persistencia (30)	
1 (15)	2 (10)	3 (10)	1 (20)	2 (15)	1 (20)	2 (10)

**Condiciones de aprobación:** Para aprobar debe sumar como mínimo 60 puntos y no menos del 50 % en cada sección.

## VC - Viajes Compartidos

En la Ciudad se ha observado un creciente problema para los pasajeros de colectivos dado que por la baja de frecuencias y reducción de servicios se encuentran con problemas para cumplir con sus obligaciones laborales, familiares y estudiantiles.

Nuestro equipo se propuso realizar aportes a la búsqueda de una solución a través de herramientas de sistemas de información con una novedosa integración de las prácticas asociadas al *carpooling* y a las redes sociales.

El *carpooling* es el término en inglés para el uso de vehículo compartido, una práctica que consiste en compartir un automóvil con otras personas tanto para viajes periódicos como para trayectos puntuales.

### **Usuarios**

Toda persona que desee utilizar el sistema deberá registrarse con su nombre de usuario y registrar su nombre, apellido, número de DNI, correo electrónico, teléfono móvil. Además se sabe qué vehículo/s tiene registrado/s en el sistema con su marca (ejemplo: Chevrolet), modelo (ejemplo: Cruze), color y dominio (patente).

La autenticación de la persona en el sistema se realizará con servicio de SSO.

### **Viajes**

Un viaje puede ser creado por cualquier persona que posea vehículo. De cada viaje se conoce el destino previsto y algunos puntos intermedios que el creador del viaje crea conveniente definir y esté asociado al recorrido esperado. También se conoce la fecha en que se realizará y para cada punto intermedio y para el destino final un horario estimado de llegada.

Además se permite planificar viajes periódicos indicando la fecha del primer viaje, la fecha del último viaje previsto, el día de la semana en que se realiza (lunes, martes, etc.) y el destino final y puntos intermedios (posibles destinos parciales) con el horario estimado de llegada a cada destino.

Existirá un creador de viajes a partir de los viajes periódicos que se detalla más adelante.



## **Rutas**

Con el objetivo de facilitar el uso del sistema se prevé simplificar el diseño para los aspectos asociados a destino previsto y puntos intermedios incorporando el concepto de ruta. Una ruta puede incluir diversas rutas intermedias y constituyen trayectos con un punto de origen y punto de destino.

El creador del viaje además de asociar los puntos deberá asociar las rutas. Por el momento el sistema tendrá la complejidad de duplicar la tarea para el usuario, pero se espera optimizar este aspecto para la próxima versión.

## **Pasajeros**

Los pasajeros podrán ser asignados tanto a los viajes como a los viajes periódicos y corresponderá obligatoriamente a alguno de los usuarios registrados en el sistema.

### **Asignación de pasajero a viaje**

Cada usuario podrá visualizar los viajes existentes y que tuviera habilitados, filtrar por el destino deseado (considerando que su destino puede ser destino final de un viaje o alguno de los destinos intermedios definidos) y agregarse al viaje.

Dado que el sistema busca incrementar la seguridad de los viajes y la confiabilidad de los pasajeros e integrarse a las redes sociales, posee un sistema de identificación de posibles pasajeros para cada viaje, por lo que el usuario sólo podrá visualizar viajes en los que cumpla las condiciones de relación con otro usuario.

El objetivo es que debe existir entre los pasajeros una relación habitual en alguna de las redes sociales para quedar habilitados a viajar juntos. Un pasajero podrá viajar en un auto que incluya al menos 1 pasajero que pertenezca a su máximo círculo de relación en alguna de las redes (no necesita conocer a todos los pasajeros, sino al menos a 1)

Cada usuario tendrá asociado en su perfil sus nombres de usuarios de redes sociales. Por el momento se prevé trabajar con Instagram, X (exTwitter) y TikTok, pero se prevé incorporar nuevas redes sociales en el futuro.

Existe un proceso de identificación de relación entre usuarios que es explicado más adelante.

### **Creación de viajes**

El sistema contendrá un generador de viajes que toma diariamente los viajes periódicos programados para 1 semana después (por ejemplo: el 04/03 a las 0 h. se buscarán todos los viajes periódicos programados para el 11/03 durante todo el día) y se generarán los viajes. Para ello se debe generar el viaje e incluir a todos los usuarios previstos observando que cumplan las condiciones para participar del viaje; todo usuario que no cumpla las condiciones no será incluido en el listado de pasajeros de ese viaje para ese día (no del viaje periódico) y notificado por correo electrónico.



## Identificación de relación entre usuarios

Todas las semanas se ejecuta un proceso de identificación de relación entre usuarios. Este proceso busca para cada una de las redes sociales el nivel de relación entre usuarios utilizando datos de las API que dichas redes sociales proporcionan.

Como primer paso se buscan las relaciones entre usuarios para cada red social. Por el momento se desconoce el algoritmo concreto del proceso de identificación de usuarios y lo realiza un sistema externo accesible vía API REST. Sí se conoce que a partir de informar para cuál red social se desea ejecutar el proceso, el mismo retorna todas las relaciones entre usuarios. Ese sistema externo identificará con su propio algoritmo el criterio por el cual selecciona las mejores relaciones, entendiendo además que cada red social tiene su propia lógica.

- Solicitud: GET /api/red/ig
- Respuesta:

```
[
  {
    "user_id": "12",
    "user_id": "33"
  },
  {
    "user_id": "91",
    "user_id": "12"
  },
  {
    "user_id": "15",
    "user_id": "33"
  }
]
```

El siguiente paso es, con la información obtenida en el paso anterior, identificar la existencia de relación en al menos 1 red social entre las personas para que queden habilitadas en el propio sistema como usuarios relacionados. Debe tenerse en cuenta que la red social retorna su nombre de usuario, el cual puede ser distinto en cada red y por lo tanto debe ejecutarse correctamente el proceso que asegure verificar la relación correcta entre usuarios-pasajeros.

Un usuario puede solicitar ser notificado cuando se produce alguna modificación en los usuarios que lo relacionan.

Finalmente, todas las relaciones entre personas serán almacenadas en una base de datos no relacional de tipo documental para ser accedidas cuando fuera necesario en la búsqueda de viajes.



### **Alcance y requerimientos**

El Sistema deberá permitir:

- Gestionar usuarios
- Gestionar viajes puntuales
- Crear y eliminar viajes periódicos
- Gestionar rutas
- Buscar viajes habilitados
- Asociar un usuario a un viaje constituyéndolo en pasajero
- Ejecutar el proceso de creación de viajes puntuales a partir de los viajes periódicos
- Ejecutar el proceso de identificación de relación entre usuarios
- Notificar a usuarios por modificaciones en las relaciones

### **Punto 1 – Arquitectura (35 puntos)**

1. **(15 puntos)** Comunique mediante un Diagrama de Componentes y un Diagrama de Despliegue de UML la arquitectura del Sistema, indicando componentes, nodos, relaciones, protocolos y todo lo que considere necesario.
2. **(10 puntos)** Explicar cómo sería el proceso (y la relación con los sistemas externos) de “Identificación de relación entre usuarios” incluyendo los componentes y las llamadas entre componentes.
3. **(10 puntos)** Seleccionar 2 atributos de calidad que considera conveniente para el sistema, describir sus requerimientos no funcionales asociados y explicar las decisiones de diseño para cumplirlas.

### **Punto 2 - Modelo de Dominio (35 puntos)**

- A. **(20 Puntos)** Documentar la solución utilizando diagramas UML (diagrama de clases obligatorio).
- B. **(15 Puntos)** Justificar las decisiones de diseño que se tomen, por ejemplo, haciendo referencia a los principios que guían al diseño o las consecuencias de aplicar un determinado patrón. También puede optar por justificar mediante código, pseudocódigo o algún otro diagrama complementario.

### **Punto 3 – Persistencia (30 puntos)**

- A. **(20 Puntos)** Diseñar el modelo de datos del punto anterior para poder persistir en una base de datos relacional, indicando las entidades con sus respectivos campos, claves primarias, las foráneas, cardinalidad, modalidad y las restricciones según corresponda.
- B. **(10 Puntos)** Justificar:
  - Qué elementos del modelo es necesario persistir.
  - Cómo resolvió los *impedance mismatches*.
  - Las estructuras de datos que deban ser desnormalizadas, si corresponde.