

# **INSTITUTO SUPERIOR DE PROFESORADO**

## **SEDES SAPIENTIAE**

**TECNICATURA SUPERIOR EN ANÁLISIS Y DESARROLLO DE SOFTWARE**



**TrashAPP**

**Un sistema de información en tiempo real sobre la recolección de residuos.**



**Integrantes del equipo**

**Martínez Bender Santiago, [santimartinez944@gmail.com](mailto:santimartinez944@gmail.com)**  
**Quiroga Kevin, [quirogakevingv@gmail.com](mailto:quirogakevingv@gmail.com)**

## **Introducción**

Este proyecto se enmarca dentro de la carrera de Tecnicatura superior en análisis y desarrollo de Software, específicamente en la cátedra de Práctica Profesionalizante II.

Los integrantes de este equipo somos quienes encontramos la problemática de “Sacar la basura” en la ciudad. Que, a pesar de ser una tarea fácil, no es raro ver bolsas de residuos rotas o sin recolectar debido a confusiones u olvidos. Se tiene entonces como objetivo el crear un sistema de información para gestionar la recolección de los residuos, que es la solución a este problema y el producto.

Se definen los requerimientos funcionales y no funcionales y la solución está diseñada mediante el modelo orientado a objetos, pudiendo adaptar así al ámbito real todo el diseño llevado a cabo mediante el lenguaje unificado de modelado (UML) en distintos tipos de diagramas de especificación.

Todo lo exployado en este documento puede ser desarrollado mediante las herramientas de: JavaScript, HTML, CSS, PHP y MySQL.

## Marco conceptual

Este proyecto ha sido descrito y desarrollado mediante el modelo de objetos y el lenguaje unificado de modelado (UML).

Mediante el modelo de objetos podemos identificar elementos y componentes tal cuales son en la vida real; su interacción con el sistema, con el usuario, y con ellos mismos, como en el ejemplo de: Tener un Cronograma de recolección de residuos. ¿Qué tiene un cronograma de recolección de residuos? Bueno, está compuesto por una Zona determinada, el total y la individualidad de los depósitos de residuos, un horario, un camión que va a recorrer esa zona... En una zona viven personas, las personas crean sus usuarios, los usuarios pueden realizar consultas... una clase de usuario se subdivide en cliente y administrador, etcétera

Es así todo un panorama de objetos y clases que son diferentes, pero que cumplen funciones específicas de ellos y que se interrelacionan.

Por su parte, UML nos permite describir casos de uso, acciones y flujo de acciones de objetos, nos permite plasmar las clases en un diagrama estándar que cualquier individuo con conocimiento en el área pueda leer y comprender, nos da herramientas para relacionar entidades e identificar los estados de un objeto a lo largo de la vida del mismo en el sistema.

## Gestación del proyecto

### Contexto

TrashAPP es una aplicación móvil y plataforma en línea capaz de solucionar la problemática de sacar los residuos domiciliarios. Se basó en la ciudad de Gualeguaychú, en la que la mayoría de viviendas no dispone de un cesto externo de residuos libre de las inclemencias de las calles (Como los animales sueltos). Lo que resulta en: residuos sueltos por las calles, veredas, animales comiendo de ellos.

- Y También, la ciudad es turística, con ello se quiere decir que no todas las personas que lleguen se quedarán poco tiempo, y es probable que se confundan con los cronogramas de la localidad. Esto puede generar que, al retirar los residuos de los hogares, ocurra lo siguiente: O no se saque el residuo correcto, o perderse el recorrido del camión.

### Objetivos generales y específicos

**Objetivo General:** Se insta en **Crear un sistema de información para gestionar la recolección de los residuos**. Es sabido que implica una dificultad para las personas en general y que, por diferentes razones, el simple hecho de sacar la basura puede ser algo problemático y más simple e interactivo.

#### Objetivos específicos:

- Administrar depósitos de residuos
- Informar sobre el estado del servicio
- Alertar usuarios con respecto al recorrido del camión
- Identificar zonas urbanas de recolección y gestionar sus recorridos

### Alcance del proyecto

Este producto está enfocado tanto a consumidores finales, como los ciudadanos de una ciudad o pueblo (de cualquier edad), así también como a los organismos estatales (municipios) o alguna organización privada que quiera contratar este servicio.

### Usuarios

Esta aplicación está dirigida a los estados municipales. La idea está basada en la ciudad de Gualeguaychú, pero es aplicable a todo municipio que esté dispuesto a su uso. Un estado municipal sugerirá el consumo de esta solución a los ciudadanos y ellos serán quienes aprovechen sus ventajas.

Resumiendo:

- Municipios: A quienes se dirige esta solución.
- Ciudadanos: Quienes consumirán la aplicación.

## **Relevamiento**

El relevamiento de este proyecto ha surgido de manera Informal, primero planteándonos un problema cualquiera, que podría ya haber sido solucionado en la actualidad, pero que requería de nuestro razonamiento y generar una solución para dicho problema.

No se realizaron análisis de mercado, pero se pensó en los competidores y el alcance de nuestro producto.

No se utilizaron técnicas de relevamiento formales, todo tipo de diagrama realizado y expresado en este proyecto ha surgido del conocimiento propio de los integrantes. Es posible que se haya indagado en la red para constatar algún flujo de una clase/objeto.

## Requerimientos del sistema

### Requerimientos funcionales

- El sistema debe permitir a los usuarios registrarse con un nombre de usuario y contraseña.
- El sistema debe permitir a los usuarios iniciar sesión con un nombre de usuario y contraseña.
- El sistema debe permitir a los usuarios buscar su cronograma de recolección de residuos
- El sistema debe ser capaz de generar una interfaz en tiempo real de la recolección de residuos
- El sistema debe almacenar la ubicación de contenedores y centros de reciclaje, junto con su tipo.
- El sistema debe permitir a un usuario denunciar la irregularidad del servicio
- Se debe tener una interfaz de usuario y administrador
- Debe ser posible para el administrador el registrar los contenedores, camiones y cronogramas.
- El sistema debe generar alertas hacia los usuarios sobre el estado del servicio (Cuando funciona, cuando comienzan los recorridos...).

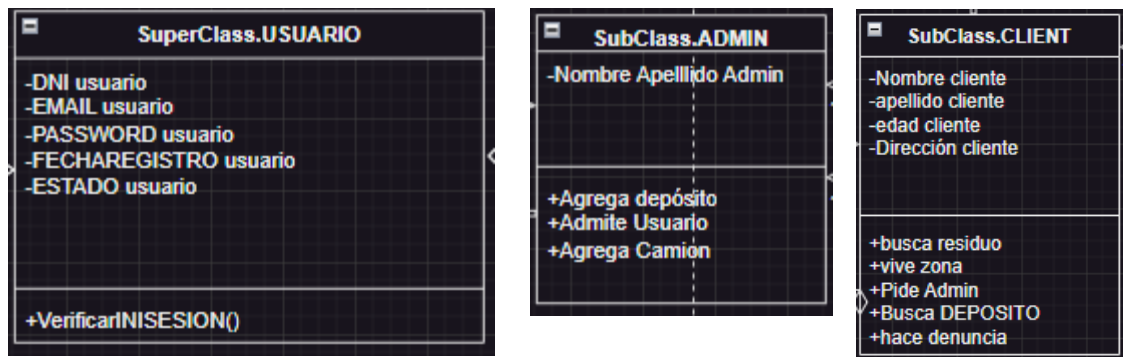
### Requerimientos no funcionales

- El sistema debe soportar un promedio de 13000 usuarios visitantes (al menos un 10% de la población de la ciudad)
- El sistema debe ser escalable
- El sistema debe ser compatible con modelos modernos y antiguos de celulares
- El sistema debe ser compatible con tecnologías de geolocalización, métricas y sensores
- El sistema debe ser capaz de verificar la veracidad de las denuncias emitidas por los usuarios
- El tiempo de respuesta debe ser no mayor a 5 segundos

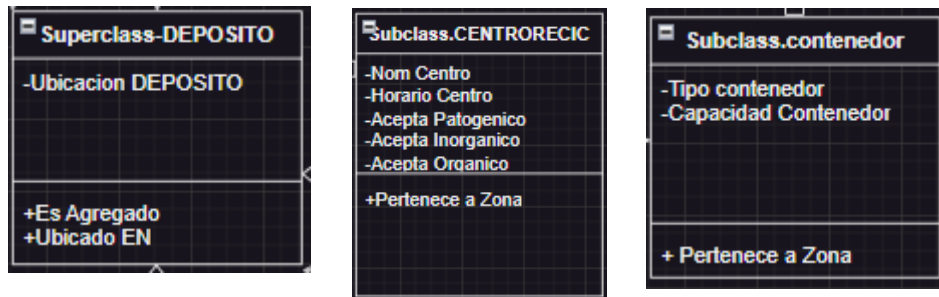
## Diseño orientado a objetos

Diagrama de clases: A continuación, se especifican los Diagramas de Clases; que son representaciones de la estructura (mediante atributos y firmas) y relaciones de cada una de las partes del sistema que sean identificadas como clases.

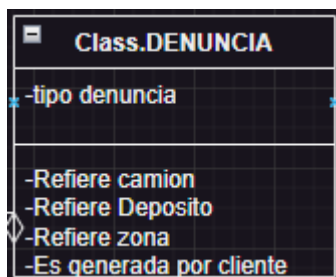
Clase de usuario:



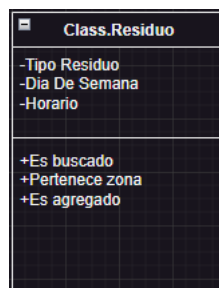
Clase de Depósito:



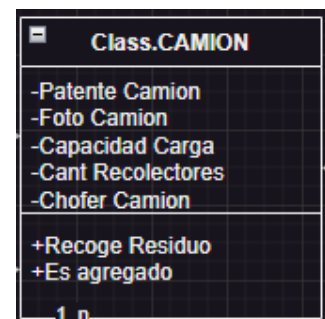
Denuncias:



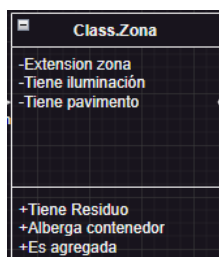
Residuos:



Camiones:



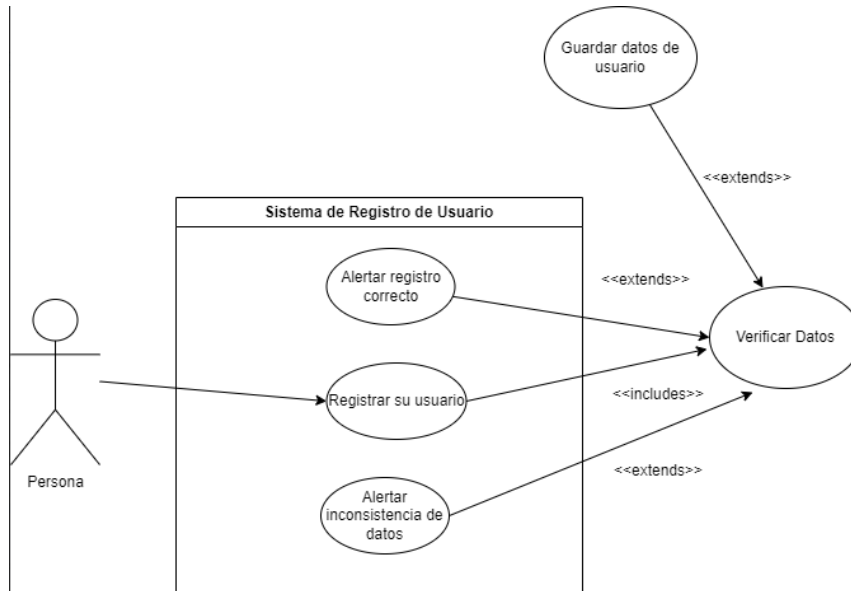
Zonas:



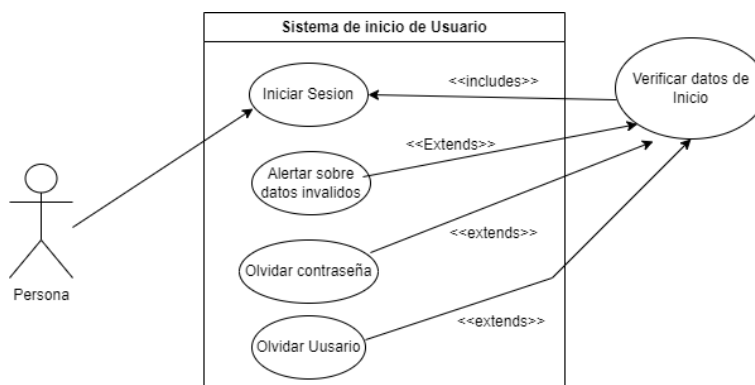
Diagramas de casos de uso (escenarios y subescenarios):

Un diagrama de casos de uso es una representación en donde se muestra la interacción de un sistema de software junto con un actor, mediante un caso de uso (o varios) unidos por flechas.

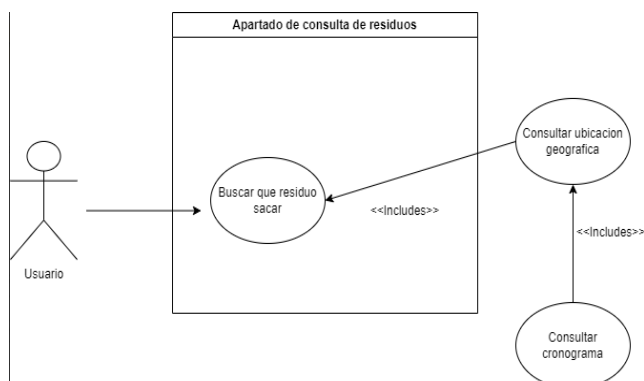
Registrar usuario:



Iniciar Sesión:

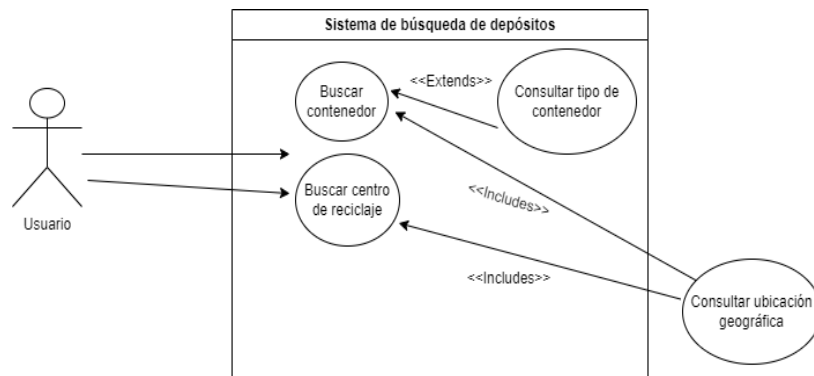


Buscar residuo a sacar:

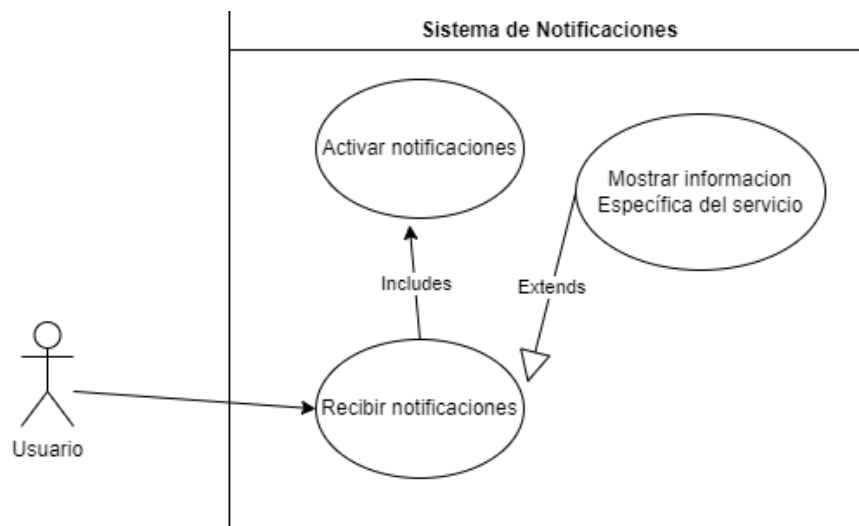




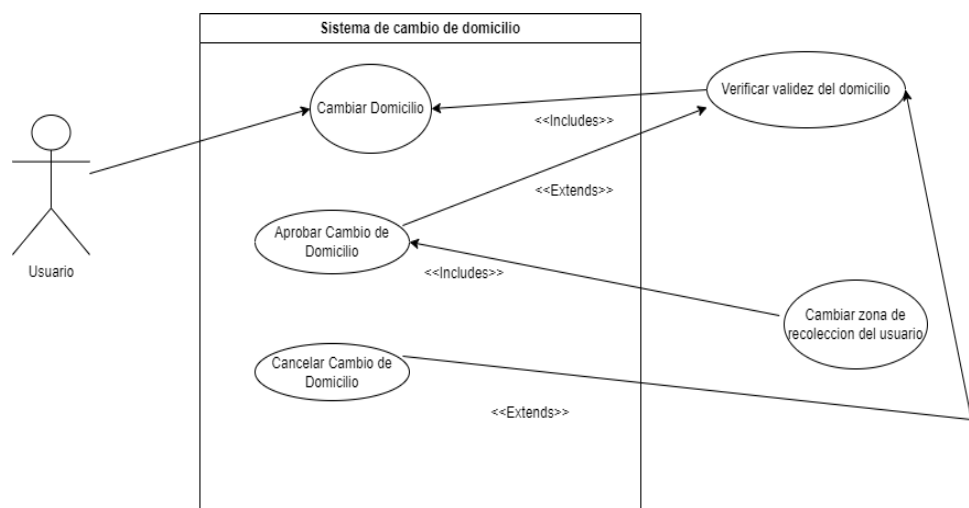
Buscar depósitos:



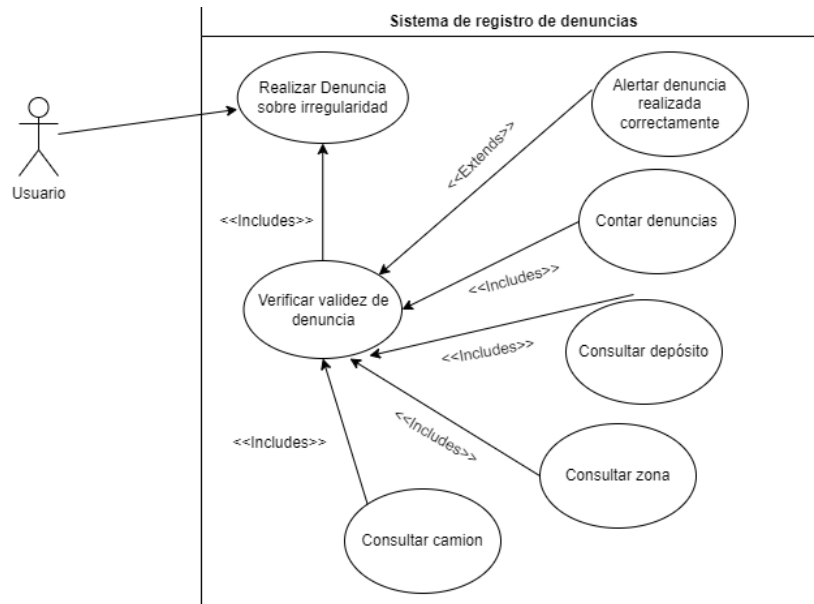
Notificar usuarios:



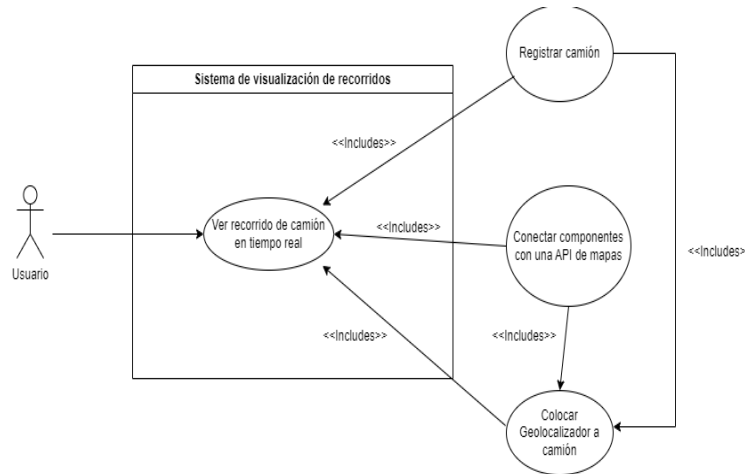
Cambiar domicilio:



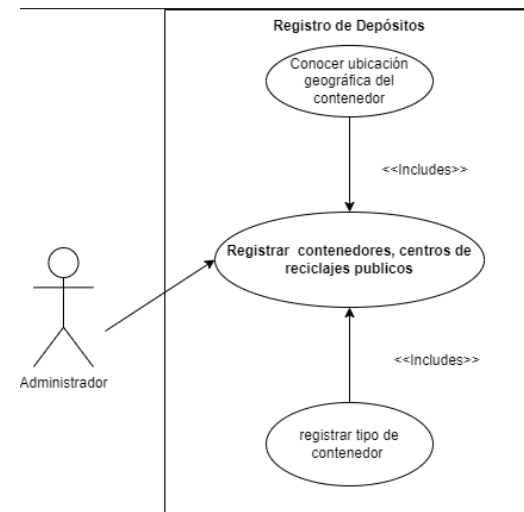
### Denunciar irregularidades del servicio:



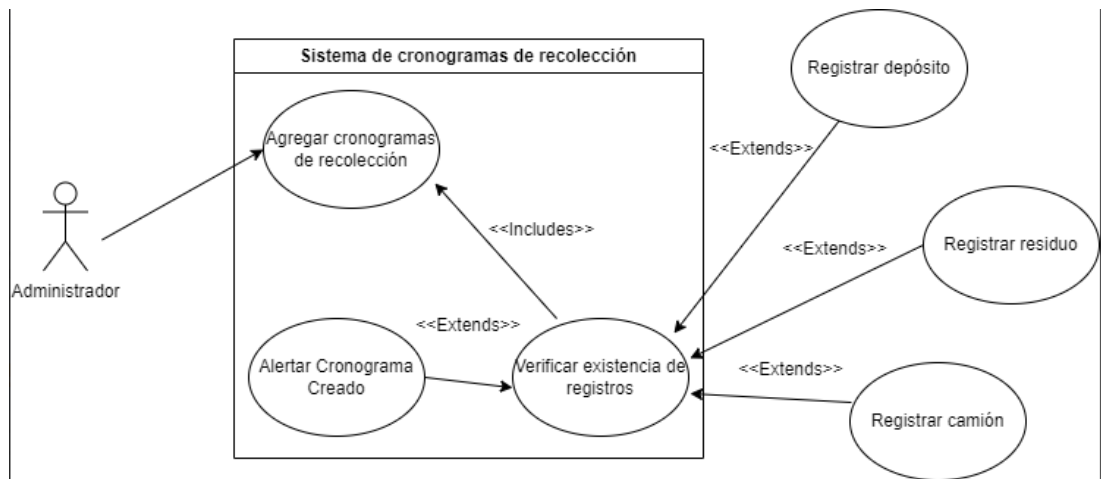
### Ver recorridos en tiempo real:



### Registrar depósitos:



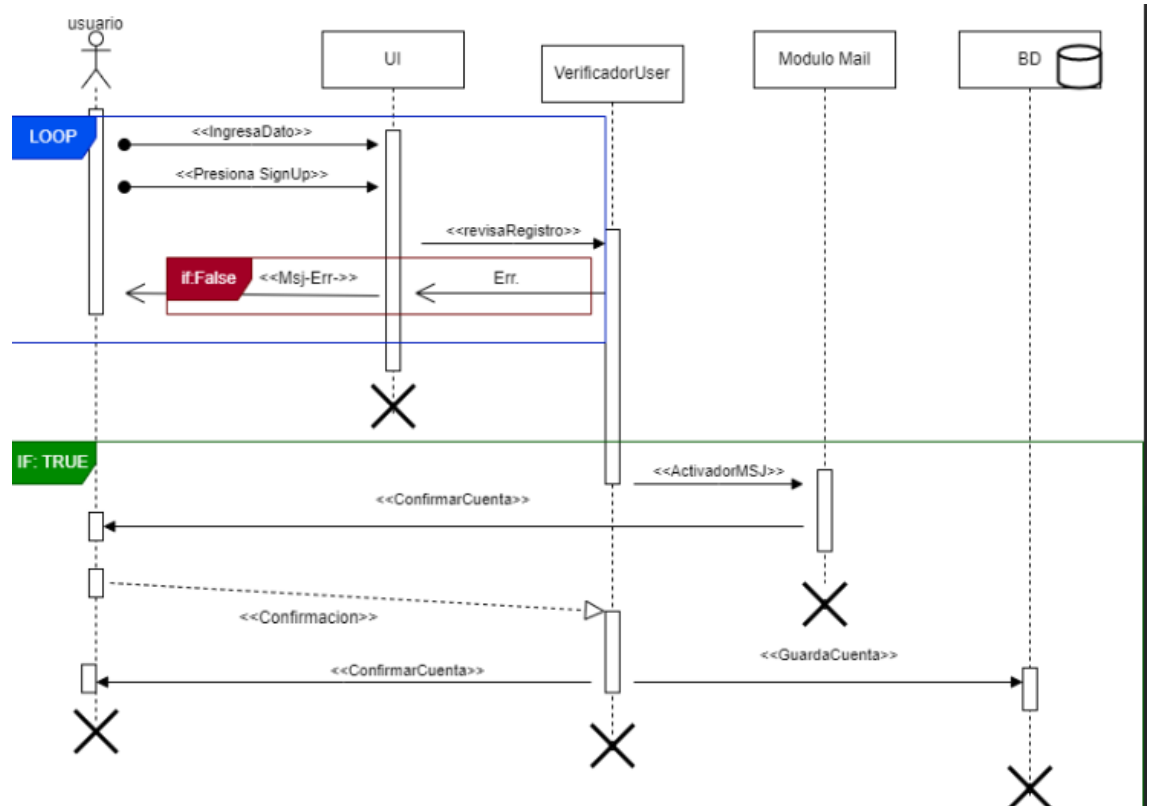
Registrar cronogramas:



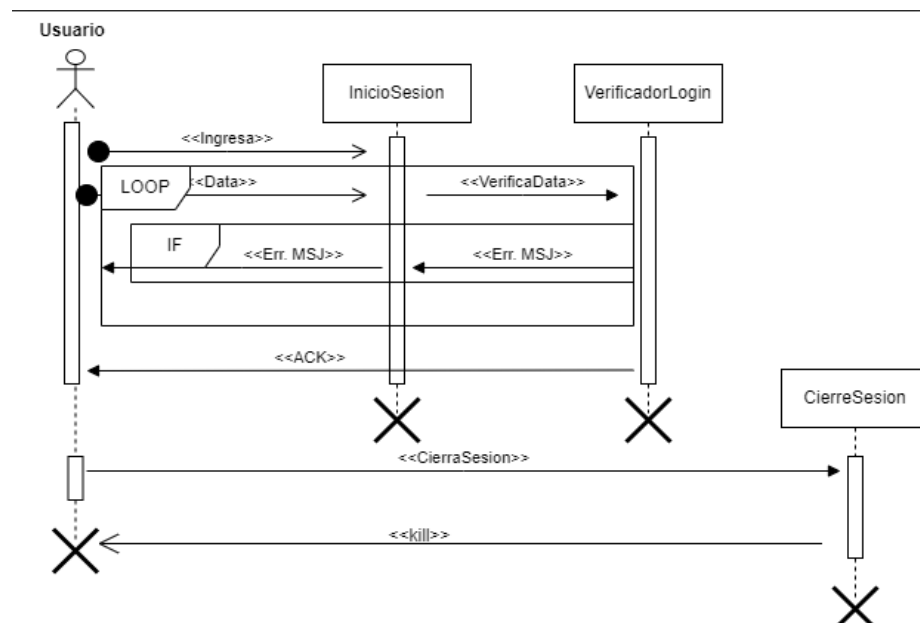
## Diagramas de secuencia

Los diagramas de secuencia muestran interacciones, mensajes, y flujos de acciones entre diferentes objetos del sistema de software, en un mismo escenario.

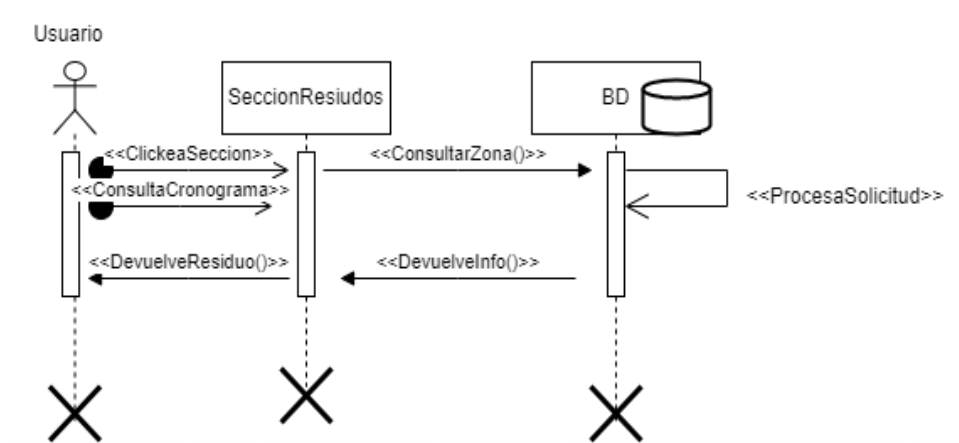
Registrar usuarios:



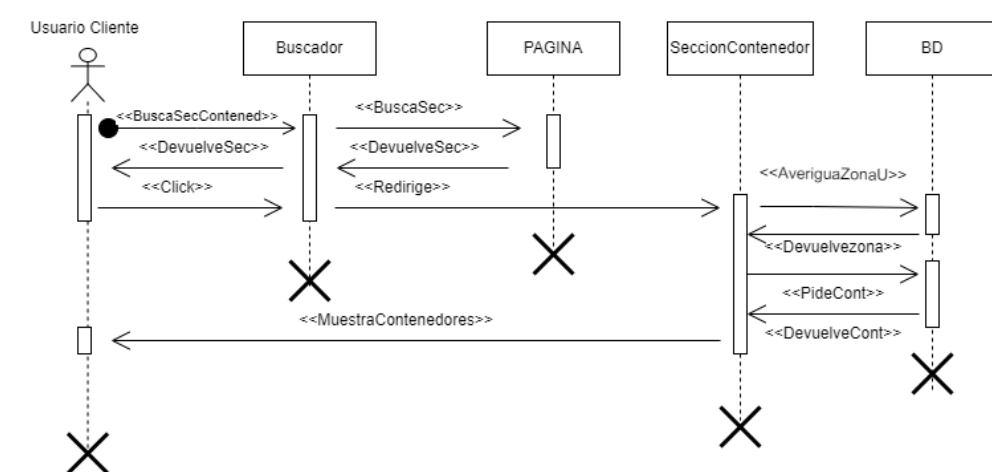
Iniciar sesión:



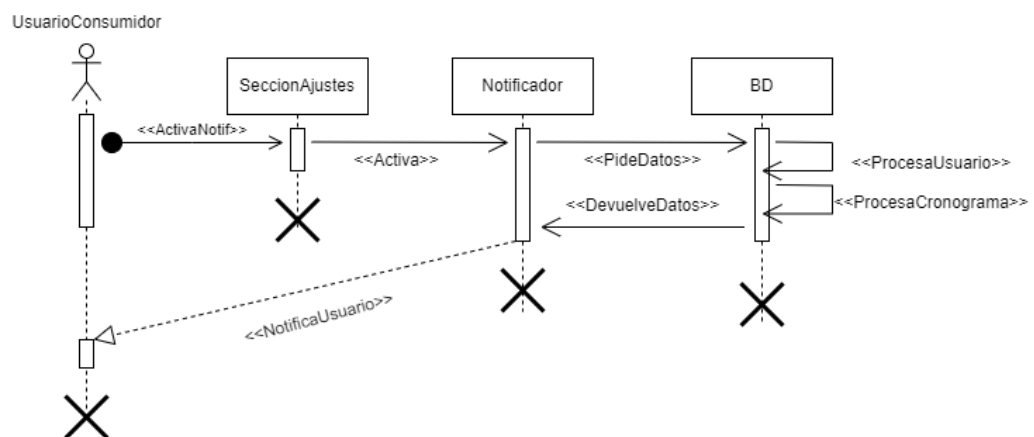
Buscar residuo a sacar:



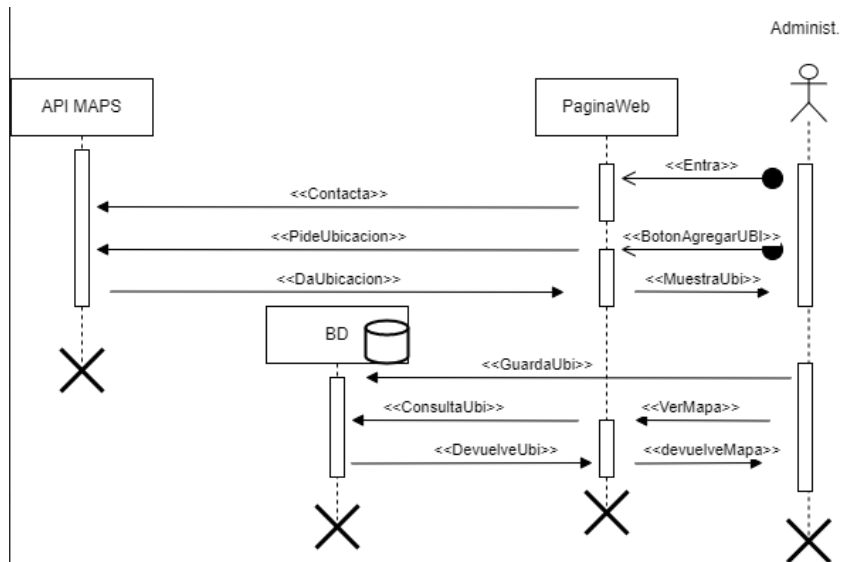
Buscar depósitos:



Notificar usuarios:



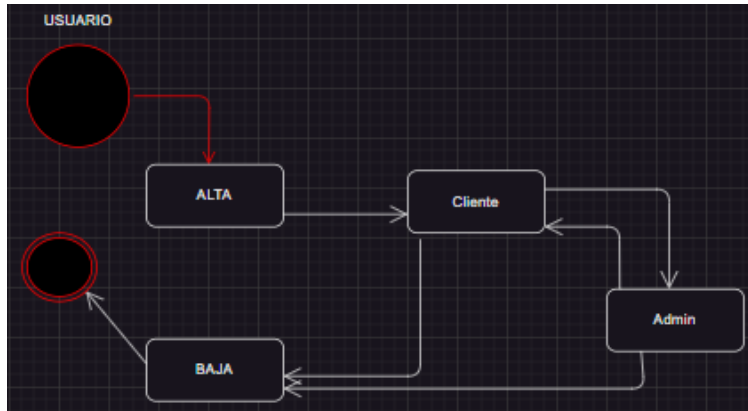
Registrar depósitos:



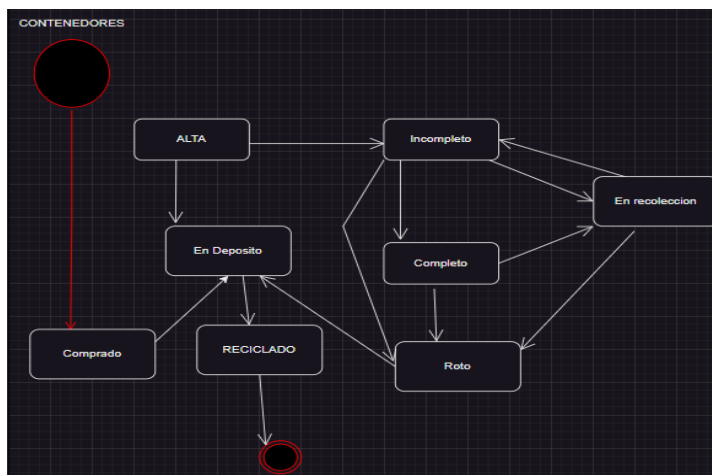
## Diagramas de estados

Este tipo de diagrama visualiza los diferentes estados y transiciones que experimenta un objeto a lo largo de su ciclo de vida en el sistema de software. Que van a variar según eventos o condiciones específicas.

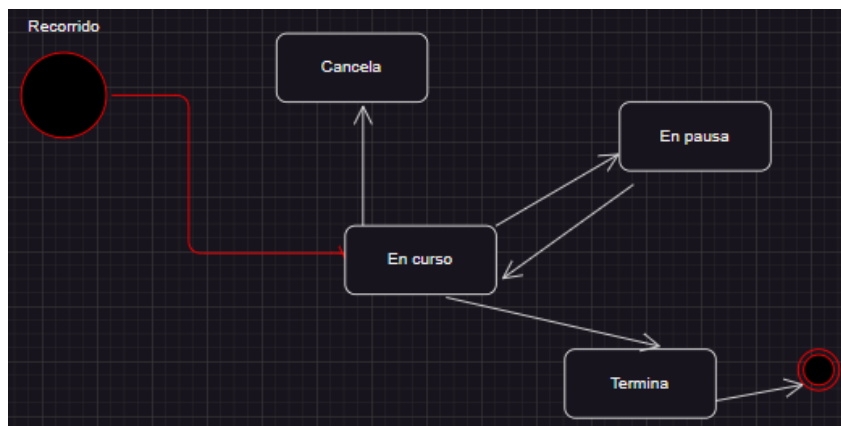
Estados de usuario:



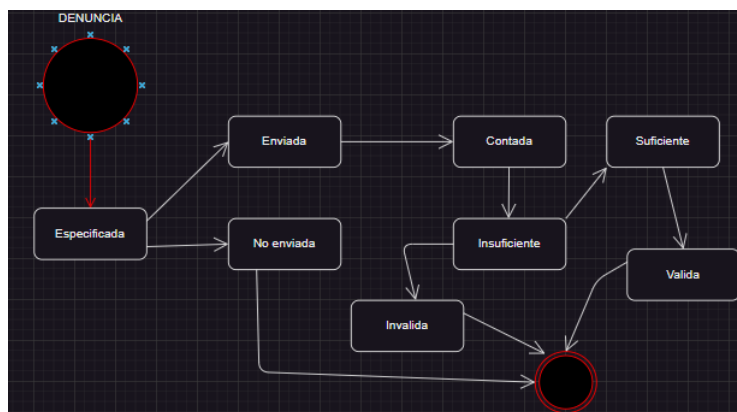
Estados de depósito contenedor:



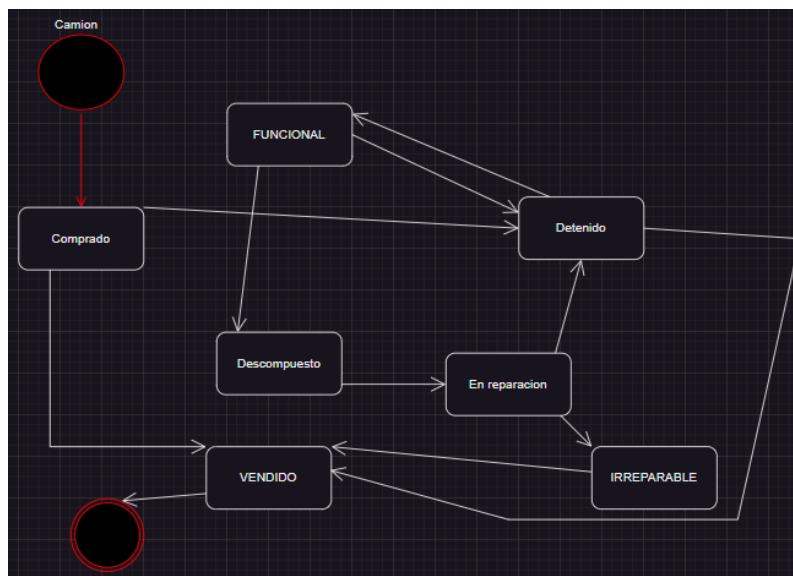
Estados de un recorrido:



Estados de una denuncia:



Estados de un camión:

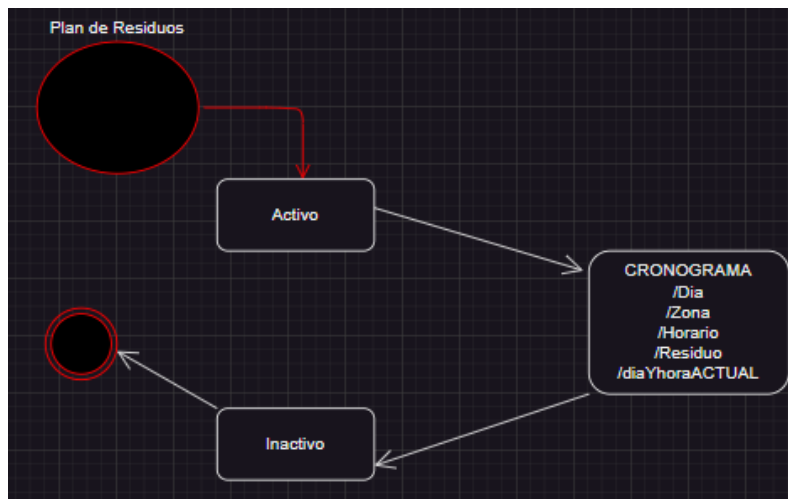




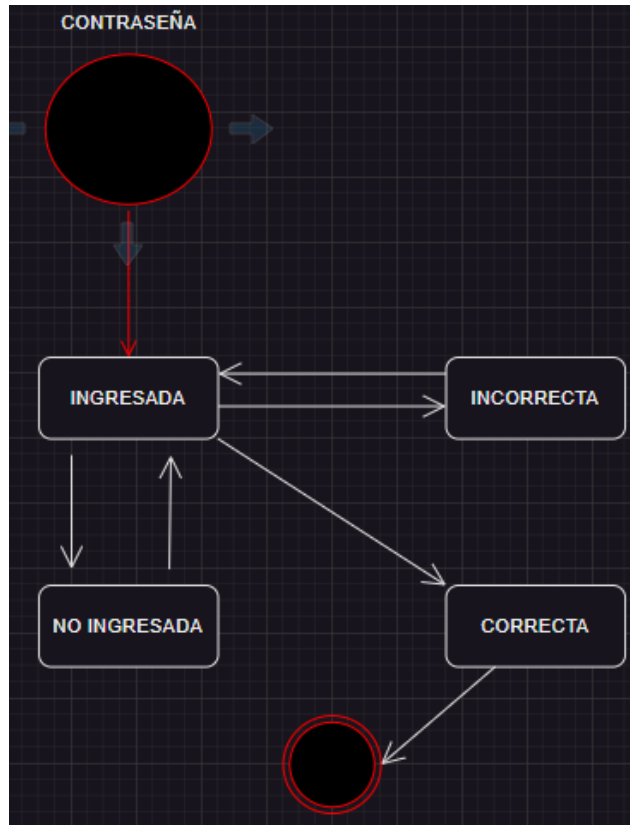
Estados de un Centro de reciclaje:



Estados de un Cronograma de residuos:

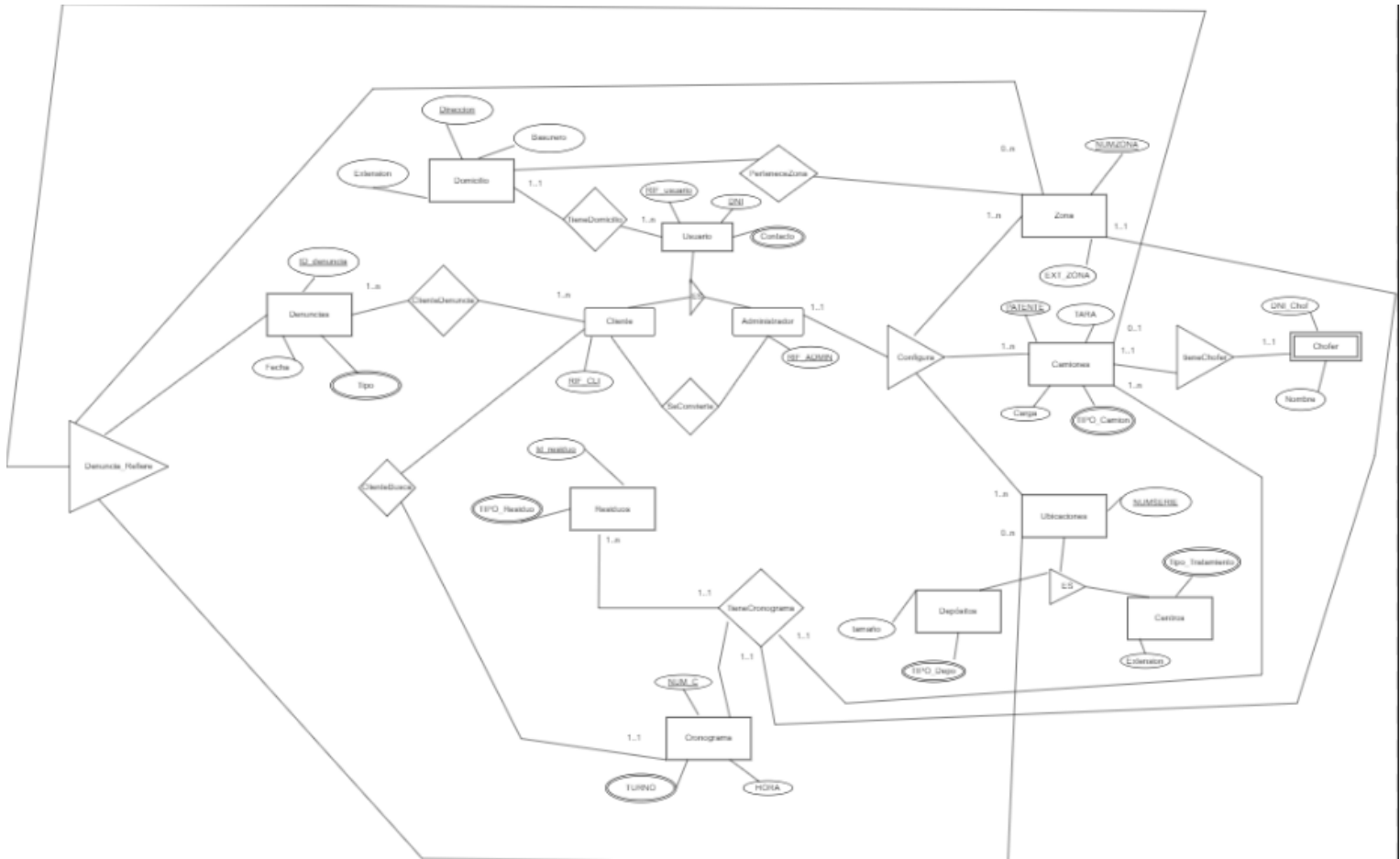


Estados de una contraseña de Usuario:



## Diagrama de entidad-relación:

El diagrama presentado a continuación, contiene las entidades, sus atributos y como se relacionan entre sí. Este diagrama (DER) es utilizado como punto de partida básico y general para determinar una base de datos.



## Desarrollo

### Definición de metodología de desarrollo:

Este proyecto podría ser desarrollado mediante la metodología del modelo de espiral, puesto que acepta que se realicen cambios mientras el desarrollo ya ha comenzado, ha ocurrido que muchas ideas se presentan a medida que se avanza en el desarrollo, tanto como para considerarlas importantes.

### Selección y justificación de herramientas:

Con las bases que se disponen al momento, se podría prácticamente diseñar un sitio web de prueba en el que se plasme nuestro producto.

Las tecnologías a utilizar serían:

- Una base de datos (MySQL):  
Se necesita almacenar la información de los usuarios y de los componentes (Camiones, depósitos, residuos, cronogramas, etc). Por lo que la base de datos es indispensable para ello.
- PHP (Como Backend): Al momento de la escritura de este documento, en Programación II se está incursionando en PHP, se sabe que PHP es utilizado para realizar las comprobaciones del lado del servidor y hacer de punto medio entre el FrontEnd y la base de datos.
- JavaScript, HTML, CSS, Bootstrap (Como Frontend): Lo que más conocemos ambos integrantes, se puede realizar una página plantilla con estos lenguajes, que se vea simple pero funcional y no tan anticuada.

Cabe recalcar que estas son las tecnologías optadas por conocimiento y por tiempo a la fecha.

¿Hay que poner un plazo? Por lo menos en 5 meses, algo sólido se puede desarrollar.

## Conclusiones finales

### Del trabajo:

El trabajo es interactivo y excitante, ha permitido a los integrantes el incursionar en el desarrollo de un producto desde cero partiendo desde una problemática cotidiana, y saber y experimentar también todo el proceso lógico detrás, como entender algo tan simple como “Qué va y qué no va”, qué cosas son correctas, si lo escrito es válido a nivel funcional o es más de bajo nivel, y anima a comunicarle a un público la solución desarrollada con sus bases bien fundadas.

Sería correcto recibir un poco más de ejemplificación respecto del tema de los diagramas UML, puesto que ver mucha ejemplificación ayuda a entender cómo aplicar el caso desarrollado.

### Del proceso grupal:

Se tuvo una dificultad por la pérdida de un integrante, Matevé Alejandro. Debido a una coincidencia en materias libres, hay que decir que esta cátedra se nutre más mediante el trabajo grupal; hay muchos aspectos que debatir con más de un punto de vista y eso enriquece el proceso al mismo tiempo que hace lazos más fuertes entre los que integran al proyecto.

Sin lugar a dudas es complicado entender el proceso de desarrollo de un proyecto como tal, aun así, con ejemplificación y práctica. Uno ya desarrolla un pensamiento crítico y funcional, que es lo más importante.

## **Bibliografía**

### **Básica:**

- Presman, R. (1997). Ingeniería de Software. Ed: Prentice Hall.
- Kendall K. y Kendall J (2005). Análisis y diseño de sistemas. Ed. Pearson.
- Booch, G; Rumbaugh, J; Jacobson, I (2000) UML : el lenguaje unificado de modelado. Ed. Addison - Wesley

### **Complementaria:**

- Apuntes y recursos digitales disponibles en el aula virtual.