



# **Especificación de Requisitos Software**

(Formato IEEE Std. 830-1998)

ANA BÉLEN DUARTE LEÓN  
LUCÍA LATORRE  
NOELIA PÉREZ  
ELENA PINUAGA  
ÁNGELES PLAZA  
ÁNGELA RUIZ RIBERA  
BELÉN SÁNCHEZ

**2018/2019**

# CONTROL DE CAMBIOS

## VERSIÓN 1

Autores	Descripción
<i>ANA BÉLEN DUARTE LEÓN</i>	Pantallas, casos de uso y diagrama actividad
<i>LUCÍA LATORRE</i>	Casos de uso
<i>NOELIA PÉREZ</i>	Casos de uso
<i>ELENA PINUAGA</i>	Casos de uso
<i>ÁNGELES PLAZA</i>	Pantallas, casos de uso y diagrama actividad
<i>ÁNGELA RUIZ RIBERA</i>	Redacción, supervisión y ensamblado
<i>BELÉN SÁNCHEZ</i>	Redacción, supervisión y ensamblado

## VERSIÓN 2

Autores	Descripción
<i>ANA BÉLEN DUARTE LEÓN</i>	
<i>LUCÍA LATORRE</i>	
<i>NOELIA PÉREZ</i>	
<i>ELENA PINUAGA</i>	
<i>ÁNGELES PLAZA</i>	
<i>ÁNGELA RUIZ RIBERA</i>	Estética y correcciones semánticas
<i>BELÉN SÁNCHEZ</i>	Hipervínculos

## CONTENIDO

Control de cambios .....	1
Versión 1 .....	1
Versión 2 .....	2
Introducción.....	4
Propósito.....	4
Alcance.....	4
Definiciones, acrónimos y abreviaturas .....	4
Referencias.....	5
Organización del documento .....	5
Descripción general .....	6
Perspectiva del producto .....	6
Funciones del producto .....	6
Características del usuario .....	7
Restricciones .....	7
Supuestos y dependencias.....	8
Requisitos futuros .....	8
Requisitos específicos .....	9
Interfaces externos .....	9
Requisitos funcionales .....	10
Requisitos de rendimiento.....	44
Requisitos lógicos de la base de datos.....	44
Restricciones de diseño .....	44
Atributos del sistema software .....	45
Bibliografía .....	46

# INTRODUCCIÓN

## PROPÓSITO

Esta SRS presenta el trabajo realizado para la creación y desarrollo de una aplicación móvil destinada a la realización de **reservas**. La interfaz permitirá reservar **restaurante, mesa y menú**, así como su pago por adelantado (si el cliente así lo desea).

El propósito del documento es mostrar de forma clara las características y especificaciones, así como recoger los requisitos necesarios para la creación del software.

## ALCANCE

Esta aplicación pretende llegar a un **gran número de personas** y facilitar e innovar el proceso de elección y reserva en un restaurante. Además, ayuda a estos comercios a darse a conocer y conseguir más clientes.

Será de gran utilidad para empresarios, estudiantes y todos aquellos que dispongan de poco tiempo para comer. El proyecto les dará la posibilidad de reservar sus menús por adelantado y llegar al local con la mesa puesta.

Además, permitirá a los más quisquillosos poder escoger la mesa que más se adecúe a sus gustos y cualquier ciudadano de calle podrá reservar fácilmente en su restaurante favorito.

## DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS

<b>Reserva:</b>	<i>información que se le pasa al restaurante con el número de clientes, la hora de llegada y, en algunos casos, la mesa y el menú que van a tomar.</i>
<b>Filtros:</b>	<i>características para acotar determinados restaurantes de un conjunto más amplio, como pueden ser la disponibilidad, la distancia o el tipo de comida que sirven.</i>
<b>Base de datos:</b>	<i>sistema formado por un conjunto de datos almacenados en discos que permiten el acceso directo a ellos y un conjunto de programas que manipulen ese conjunto de datos.</i>



## REFERENCIAS

Para la construcción y diseño de la aplicación hemos tomado como referencia aplicaciones ya existentes como “El Tenedor”, “Club Kviar”, “Glovo” o “UberEats”.

## ORGANIZACIÓN DEL DOCUMENTO

Podemos distinguir tres partes en el siguiente documento:

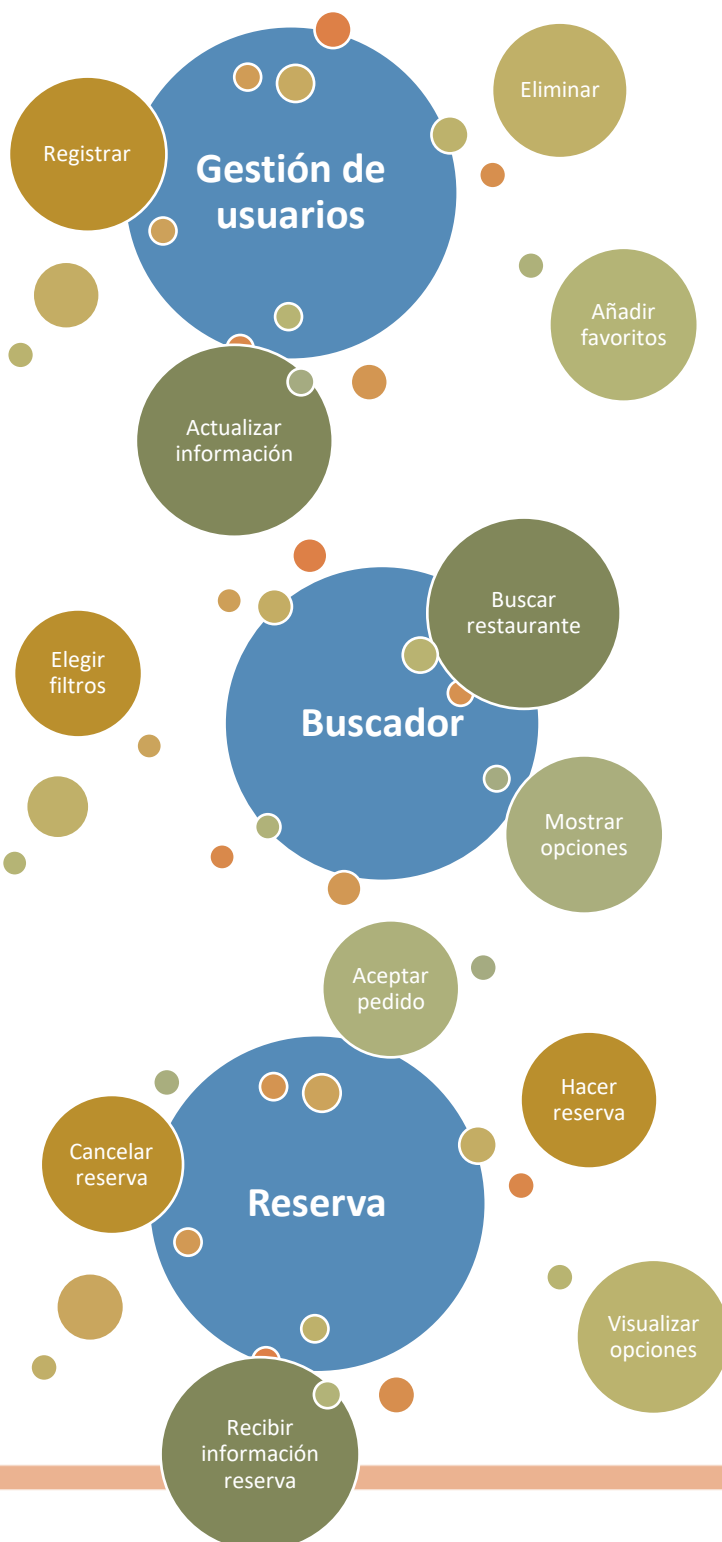
- **INTRODUCCIÓN:** Muestra una idea general del proyecto.
- **DESCRIPCIÓN GENERAL:** Describe de forma más detallada el producto sin detenerse en los aspectos más técnicos.
- **REQUISITOS ESPECÍFICOS:** Muestran las características necesarias para la implementación técnica del proyecto

## DESCRIPCIÓN GENERAL

### PERSPECTIVA DEL PRODUCTO

CómoComo es una aplicación destinada a ser usada en dispositivos móviles. Será utilizada tanto por restaurantes como por clientes, y requerirá de conexión a internet y a una base de datos.

### FUNCIONES DEL PRODUCTO



#### GESTIÓN DE USUARIOS

Este subsistema se encarga de **gestionar y actualizar** a los distintos **usuarios** que harán uso de la aplicación; las personas y los restaurantes.

#### BUSCADOR

Es el subsistema encargado de **filtrar y buscar** lo que quiere el usuario persona.

#### RESERVAS

Subsistema encargado de **efectuar, cancelar y aceptar** reservas.

*[diagrama interactivo]*

## CARACTERÍSTICAS DEL USUARIO

Encontramos dos tipos de usuario:

- **RESTAURANTE:** Son los encargados de proporcionar información sobre su local y lo que ofrecen, así como de utilizar la aplicación para servir a sus clientes adecuadamente.



- **CLIENTE:** Hará uso de la aplicación para reservar restaurante, mesa y menú en el local que más se adecúe a sus gustos.



## RESTRICCIONES

Hemos tomado en cuenta las siguientes restricciones:

- Aplicación destinada a dispositivos **móviles** Android y iOS.
- Debe ser **fácil de usar, ser atractiva e intuitiva**.
- Debe ser una **aplicación ligera** para que la gente se la descargue. Además, debe tener la capacidad de ser utilizada simultáneamente por diversos usuarios.



## SUPUESTOS Y DEPENDENCIAS

LOS FACTORES MÁS DESTACABLES QUE PUEDEN ALTERAR LOS REQUERIMIENTOS DEL SISTEMA SON:

**Cambios de requisitos.** En el caso de que durante el desarrollo de la aplicación el cliente no esté de acuerdo con alguna de las partes (estética, funcionalidades, usabilidad...), ésta se deberá rectificar para volver a hacerla de acuerdo con sus nuevas indicaciones.

Tras el desarrollo e implementación de la aplicación, dependiendo de cómo se comporte y si tiene éxito, podrán surgirnos **funcionalidades extras** que queramos añadir, por lo que es posible que haya que cambiar los requerimientos de software.

## CONDICIONES QUE SE DEBEN CUMPLIR (USUARIOS)

Tanto los restaurantes como los clientes deben tener una buena **conexión a internet**. Además, será necesaria la **descarga** de la **aplicación** compatible con su dispositivo móvil.

Además, la **información** debe estar **actualizada**. Es decir, los restaurantes deben tener una página creada y actualizada para que los clientes puedan efectuar las reservas. Para que esto sea posible, se tendrá que registrar, además, la actividad de los clientes.

## CONDICIONES DE LAS CUALES DEPENDE EL SISTEMA

La aplicación depende de una **base de datos ajena** a ella; la que contiene a los restaurantes. Si este servidor falla, los clientes no podrán acceder a dicha información.

El tratamiento de datos deberá respetar la **legislación** vigente.

## REQUISITOS FUTUROS

Algunas actualizaciones que podrían ser implementadas en un futuro son:

- Incluir el reparto a **domicilio**.
- Dar la opción de **catering**.
- Que la aplicación de **soporte** no solo a restaurantes sino también a otros servicios como cines, conciertos, teatros...

# REQUISITOS ESPECÍFICOS

## INTERFACES EXTERNOS

### INTERFACES HARDWARE



Dado que la aplicación está pensada para **móviles** (iOS y Android) **no necesitamos** aportar ningún tipo de hardware adicional, ya que estos ya cuentan con GPS y conexiones wifi y de datos móviles.

### INTERFACES SOFTWARE



La aplicación estará programada en **java**, por lo que necesita que todos los equipos lleven lo lleven instalado.

### INTERFACES DE COMUNICACIÓN



CómoComo se comunica con distintas **bases de datos**, en las cuales se encuentran la información relativa a los restaurantes y a los clientes. Para que esta comunicación sea posible, basta con habilitar el acceso remoto a un **servidor**, de forma de que los usuarios puedan actualizar su información.

Por otro lado, la aplicación también tendrá que estar conectada con Google Maps (Android), Maps (iOs) o cualquier alternativa que permita **acceder a mapas** y determinar una ubicación. Esto permitirá filtrar las búsquedas de restaurantes y localizar los establecimientos.

## REQUISITOS FUNCIONALES

En este apartado explicaremos los requisitos funcionales de la aplicación mediante diagramas de casos de uso, clasificados por subsistemas, y diagramas de actividad.

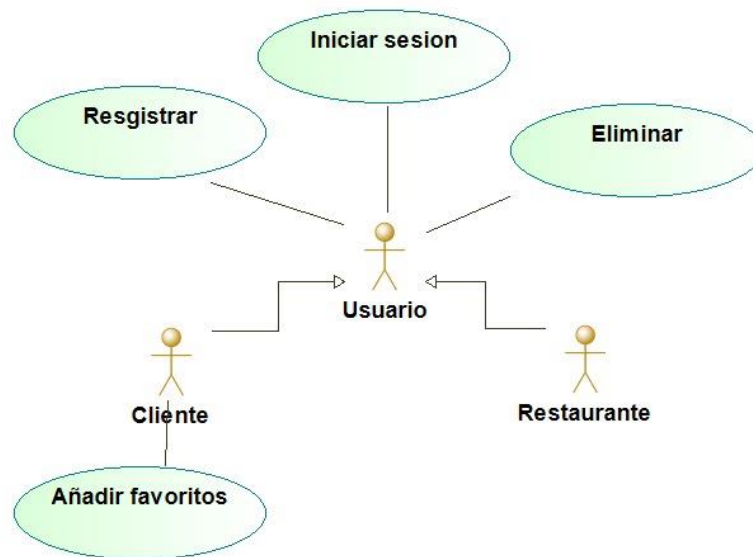
Cada caso de uso será explicado en detalle.

Subsistema por relevancia	Función por relevancia y subsistema	Función por relevancia
<u>RESERVA</u>	<ol style="list-style-type: none"><li>1. <u>Actualizar información</u></li><li>2. <u>Añadir favoritos</u></li><li>3. <u>Registrar</u></li><li>4. <u>Eliminar</u></li></ol>	<ol style="list-style-type: none"><li>1. <u>Buscar restaurante</u></li><li>2. <u>Hacer reserva</u></li><li>3. <u>Aceptar reserva</u></li><li>4. <u>Visualizar opciones</u></li><li>5. <u>Recibir información</u></li></ol>
<u>BÚSQUEDA</u>	<ol style="list-style-type: none"><li>1. <u>Buscar restaurante</u></li><li>2. <u>Mostrar opciones</u></li><li>3. <u>Elegir filtros</u></li></ol>	<ol style="list-style-type: none"><li>6. <u>Cancelar reserva</u></li><li>7. <u>Actualizar información</u></li><li>8. <u>Mostrar opciones</u></li></ol>
<u>GESTIÓN DE USUARIOS</u>	<ol style="list-style-type: none"><li>1. <u>Hacer reserva</u></li><li>2. <u>Aceptar reserva</u></li><li>3. <u>Visualizar opciones</u></li><li>4. <u>Recibir información</u></li><li>5. <u>Cancelar reserva</u></li></ol>	<ol style="list-style-type: none"><li>9. <u>Elegir filtros</u></li><li>10. <u>Añadir favoritos</u></li><li>11. <u>Registrar</u></li><li>12. <u>Eliminar</u></li></ol>

## DIAGRAMA CASOS DE USO

### SUBSISTEMA GESTIÓN DE USUARIOS

Este subsistema se encarga de **gestionar y actualizar** a los distintos **usuarios** que harán uso de la aplicación; las personas y los restaurantes.



A continuación, detallamos la información relativa a cada caso de uso.

## ACTUALIZAR INFORMACIÓN

Nombre del autor: Noelia Pérez

- **Identificador:** G1
- **Objetivo en Contexto:**  
Cuando un usuario se registra en la aplicación o sale de ella el gestor de usuarios actualiza tal información.
- **Actor principal:** Aplicación
- **Actores secundarios:** Usuario
- **Qué datos usa:**  
Los datos personales del usuario (nombre, teléfono, correo electrónico, tarjeta de crédito, etc.).
- **Precondiciones:**  
El usuario está en la Página de la Aplicación y se registra o elimina su cuenta.
- **Postcondiciones:**  
**Éxito:** La aplicación registra el cambio y se envía mensaje: “se ha registrado correctamente” o “su cuenta ha sido eliminada”.  
**Fallo:** Mensajes de error al usuario
- **Flujo principal**
  1. El gestor de usuarios recibe un mensaje de entrada o salida de un usuario de la aplicación.
  2. Si entra un usuario se almacenan todos sus datos y este ya puede comenzar a usar la aplicación. Si sale un usuario se eliminan todos sus datos de la base de datos.
  3. Se envía un mensaje al usuario de confirmación: activación o eliminación de cuenta.
- **Flujos secundarios**
  1. a) El gestor de usuarios no recibe ningún mensaje de entrada o salida, no se lleva a cabo ninguna acción.
  2. a) Los datos no se almacenan correctamente: se vuelve al paso 2.  
Los datos no se eliminan correctamente: se vuelve al paso 2.
  3. a) No se envía el mensaje correctamente. Se vuelve al paso 3.

## AÑADIR FAVORITOS

Nombre del AUTOR: Lucía Latorre

- **Identificador:** G2
- **Objetivo en Contexto:**  
El usuario cliente guarda en el apartado de favoritos de su cuenta de CómoComo los restaurantes que más le gusten.
- **Actor principal:** Usuario cliente
- **Actores secundarios:** Aplicación
- **Qué datos usa:**  
La información almacenada en la base de datos del restaurante que se quiere añadir como favorito.
- **Precondiciones:**  
El usuario está en la Página de la Aplicación. La búsqueda según filtros se ha realizado con éxito y el usuario se encuentra dentro de la descripción del restaurante
- **Postcondiciones:**  
**Éxito:** Se muestra por pantalla “Se ha añadido el restaurante x a sus favoritos”.  
**Fallo:** Mensajes de error y el usuario busca otro restaurante o termina
- **Flujo principal**
  1. El usuario selecciona el restaurante que quiere marcar como favorito.
  2. El sistema busca en su base de datos el restaurante y muestra su información general.
  3. El usuario selecciona la opción “Añadir a favoritos”.
  4. Se muestra por pantalla “Se ha añadido el restaurante x a sus favoritos”.
- **Flujos secundarios**
  2. a) El sistema no encuentra el restaurante en su base de datos o lo encuentra, pero se está indispuerto.  
Se saca un mensaje de error “El restaurante no está disponible”. Se vuelve al paso 1 o termina.

## REGISTRAR



**Nombre del AUTOR:** Lucía Latorre

- **Identificador:** G3
- **Objetivo en Contexto:**

Un usuario quiere empezar a utilizar la aplicación CómoComo. Tanto restaurantes como clientes se registran creándose una cuenta que pasará a formar parte de la gestión de usuarios de la aplicación.
- **Actor principal:** Usuario cliente o restaurante.
- **Actores secundarios:** aplicación.
- **Qué datos usa:**

Los datos personales del usuario: nombre, contraseña, número de teléfono, correo electrónico, tarjeta de crédito, etc. (nombre, correo electrónico y contraseña obligatorios). Inciso: en caso de ser el restaurante se usará información adicional como menú, dirección, fotos, etc.
- **Precondiciones:**

El usuario ha abierto la aplicación y ha elegido la opción de registrarse.
- **Postcondiciones:**

**Éxito:** Se crear el nuevo usuario correctamente con por lo menos los siguientes datos: nombre, correo electrónico y contraseña.  
**Fallo:** Mensajes de error y vuelta a pedir datos o termina.
- **Flujo principal**
  1. El usuario entra en la aplicación y como todavía no se ha registrado como usuario, pulsa el botón de registrarse.
  2. El sistema pregunta al usuario por su nombre.
  3. El usuario introduce su nombre.
  4. El sistema pregunta al usuario por su correo electrónico.
  5. El usuario introduce su correo electrónico.
  6. El sistema comprueba que el correo electrónico introducido es válido.
  7. El sistema pide al usuario que meta una contraseña con la que entrará la próxima vez que entre en la aplicación.
  8. El usuario introduce una contraseña de 8 caracteres como mínimo.
  9. El sistema comprueba que la contraseña introducida es válida.
  10. El sistema pregunta al usuario si es un restaurante o un cliente.

11. El usuario responde a la cuestión anterior: es un cliente.
12. El sistema da la opción al cliente de rellenar más campos con datos personales o finalizar (podrá rellenar estos datos en otro momento si lo desea)
13. El usuario decide meter los siguientes datos: número de teléfono, tarjeta de crédito, nombre completo y dirección.
14. El sistema va comprobando que estos datos son correctos.
15. El usuario presiona el botón de finalizar.
16. El sistema guarda los datos del cliente en la base de datos de gestión de usuarios y muestra por pantalla “Usuario creado”.

- **Flujos secundarios**

- 1.a) El usuario presiona el botón de iniciar sesión o bien por equivocación, que entonces podrá darle al botón de atrás volviendo así al paso 1, o porque ya es un usuario registrado. Si es así, el usuario introduce el correo electrónico y la contraseña. El sistema comprueba que pertenece a su base de datos y que la contraseña es correcta y da acceso al usuario a su cuenta. Sin embargo, si los datos no son correctos o no existe el usuario el sistema muestra por pantalla “Datos incorrecto” y pide que introduzca los datos de nuevo.
- 6.a) El correo electrónico introducido no es válido o no existe. Se espera nueva entrada, volviendo al paso 5.
- 9.a) La contraseña introducida no es válida. Se espera nueva entrada, volviendo al paso 8
- 11.a) El usuario es un restaurante. El sistema le da la opción de rellenar más campos con otros datos que son interesantes para el cliente y para darse a conocer o saltarse el proceso. Podrá saltarse el proceso, pero la aplicación se pondrá en contacto con el restaurante vía correo electrónico pidiendo muchos más datos para poder completar su perfil como restaurante y para poder ofrecer el mejor servicio posible a los clientes.
- 12.a) El usuario presiona el botón de finalizar avanzando al paso 16.
- 14.a) Alguno de los datos es inválido. Se espera que se vuelva a introducir hasta que sea correcto o que se deje en blanco ya que es opcional introducirlo. Se continua en el flujo normal.







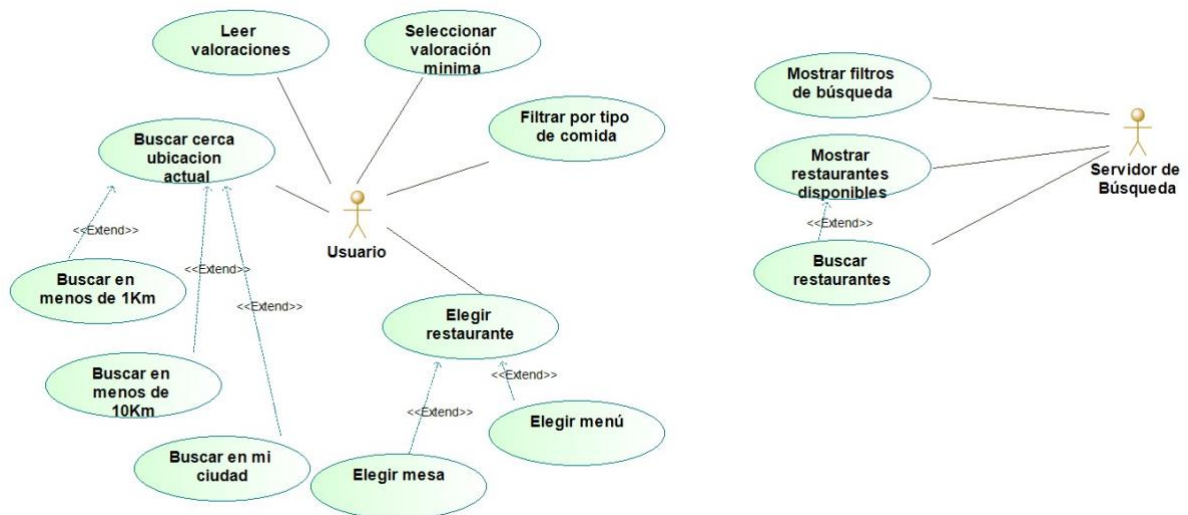
## ELIMINAR

**Nombre del AUTOR:** Lucía Latorre

- **Identificador:** G4
- **Objetivo en Contexto:**  
Un usuario quiere darse de baja de la aplicación CómoComo. Tanto restaurantes como clientes pueden borrar su cuenta que dejará de formar parte de la gestión de usuarios de la aplicación.
- **Actor principal:** Usuario cliente o restaurante.
- **Actores secundarios:** aplicación.
- **Qué datos usa:**  
Los datos personales del usuario incluidos en su cuenta.
- **Precondiciones:**  
El usuario está en la Página de la Aplicación dentro de la página de configuración.
- **Postcondiciones:**  
**Éxito:** Se muestra por pantalla “su cuenta ha sido eliminada con éxito” y los datos del usuario se eliminan correctamente de la base de datos.  
**Fallo:** El usuario no quiere eliminar su cuenta.
- **Flujo principal**
  1. El usuario presiona el botón de configuración seguido del botón de eliminar cuenta.
  2. El sistema pregunta al usuario si está seguro de que quiere eliminar la cuenta.
  3. El usuario dice que Sí.
  4. Los datos del usuario se eliminan de la base de datos y el sistema muestra por pantalla “su cuenta ha sido eliminada con éxito”.
  5. Se vuelve a la página de inicio donde se da la opción de iniciar sesión o registrarse.
- **Flujos secundarios**
  - 3.a) El usuario dice que No. Se cancela el proceso y se vuelve a la página de configuración.



## SUBSISTEMA BUSCADOR



Es el subsistema encargado de **filtrar y buscar** lo que quiere el usuario persona.  
A continuación, detallamos la información relativa a cada caso de uso.

## BUSCAR RESTAURANTE


Nombre del autor: Ana Belén Duarte y Ángeles Plaza

- **Identificador:** B1
- **Objetivo en Contexto:**

La aplicación, una vez recibidos los filtros de búsqueda introducidos por el usuario tales como número de personas o tipo de comida buscará los restaurantes disponibles que cumplan con estos requisitos previamente impuestos por el cliente.
- **Actor principal:** Aplicación
- **Actores secundarios:** Gestor de restaurantes, gestor de ubicación
- **Precondiciones:** Usuario introduce unos filtros para definir sus preferencias
- **Postcondiciones:**

**Éxito:** La aplicación carga los restaurantes que cumplen los requisitos que el usuario ha impuesto.

**Fallo:** Mensaje de error y vuelve a pedir los filtros
- **Flujo principal**
  1. La aplicación recibe los filtros previamente introducidos por el usuario
  2. El sistema comprueba estos filtros.
  3. El sistema carga los restaurantes que existen en su base de datos.
  4. El sistema compara cada filtro con la información sobre los restaurantes ya cargado.
  5. El sistema carga los restaurantes compatibles a los filtros introducidos.
  6. La aplicación comprueba la disponibilidad de mesas que hay en estos restaurantes.
  7. El sistema carga los planos de los restaurantes indicando las mesas disponibles para el usuario (se mostrarán posteriormente en el proceso de reserva).
- **Flujos secundarios**
  - 2 a) El sistema no es capaz de reconocer los filtros
  - 3 a) El sistema no carga correctamente los restaurantes que existen registrados en su base de datos.



Se vuelven a reintentar la carga de esta información ya que este paso es crucial.

4 a) El sistema no carga correctamente la información acerca de los restaurantes de su base de datos.

Se volverá al paso 3.

5 a) Ningún restaurante cumple con todos los requisitos facilitados por el usuario.

Se volverá al paso dónde el usuario introduce los filtros o se le dará la opción de salir.

6 a) Si ninguno de los restaurantes que cumplen los requisitos tiene mesas disponibles se volverá al paso dónde el usuario introdujo los filtros o se le dará la opción de salir.



- Se espera nueva entrada, volviendo al paso 1.

## ELEGIR FILTROS

Nombre del AUTOR: Lucía Latorre

- **Identificador:** B3
- **Objetivo en Contexto:**  
El cliente, con el objetivo de encontrar un restaurante que se ajuste a sus requisitos, elige uno a uno filtros que se aplicarán en la búsqueda del restaurante más idóneo para llevar a cabo una reserva.
- **Actor principal:** Usuario cliente.
- **Actores secundarios:** aplicación.
- **Qué datos usa:**  
Las elecciones del usuario con respecto a los filtros.
- **Precondiciones:**  
El usuario está en la Página de la Aplicación en la pestaña de añadir filtros de búsqueda.
- **Postcondiciones:**  
**Éxito:** Se introducen los filtros correctamente y la aplicación los aplica resultando útil a la hora de elegir un restaurante.  
**Fallo:** Mensajes de error y vuelta a pedir datos o termina.
- **Flujo principal**
  1. El cliente se encuentra en la página de búsqueda de restaurantes donde podrá buscar un restaurante usando el mapa o aplicando unos filtros que harán que la búsqueda sea más sencilla.
  2. El cliente selecciona la pestaña de añadir filtros de búsqueda.
  3. El sistema pide al cliente el nombre del restaurante en el que quiere hacer la reserva.
  4. El cliente escribe el nombre o decide saltarse esta acción.
  5. El sistema comprueba que el restaurante se encuentra en la base de datos.
  6. El sistema pide al cliente el tipo de comida que desea tomar para así, mostrar restaurantes que ofrezcan entre otros el tipo de comida elegido.
  7. El cliente elige uno de entre todos los de la lista de los predeterminados: italiano, oriental, comida española, comida rápida, etc. Puede decidir saltarse esta acción.
  8. El sistema pide al cliente una ubicación dando la opción de utilizar la ubicación actual.

9. El cliente introduce una ubicación que puede incluir: calle, número, zona, región, código postal, etc. Puede decidir saltarse esta acción.
10. El sistema comprueba que la ubicación es válida.
11. El sistema pide al cliente la fecha y hora de la reserva.
12. El cliente selecciona la fecha a través de un calendario y luego la hora a través de una lista con las predeterminadas o, en su defecto, introduce ambas a mano. Puede decidir saltarse esta opción.
13. El sistema comprueba que la fecha y hora son válidas.
14. El sistema pide al cliente el número de comensales de la posible reserva para buscar restaurantes con esa disponibilidad.
15. El cliente lo introduce o decide saltarse esta acción.
16. El sistema comprueba que el número introducido es válido.
17. El sistema pide al cliente el rango de precio en el que quiere que se encuentre el restaurante.
18. El usuario elige entre las siguientes opciones: € (barato), €€(precio medio), €€€(precio relativamente alto), €€€€ (caro). Puede decidir saltarse esta opción.
19. El usuario presiona el botón de aplicar filtros.
20. Con los datos recopilados el sistema buscará los restaurantes que se ajusten a los filtros utilizados.

- **Flujos secundarios**

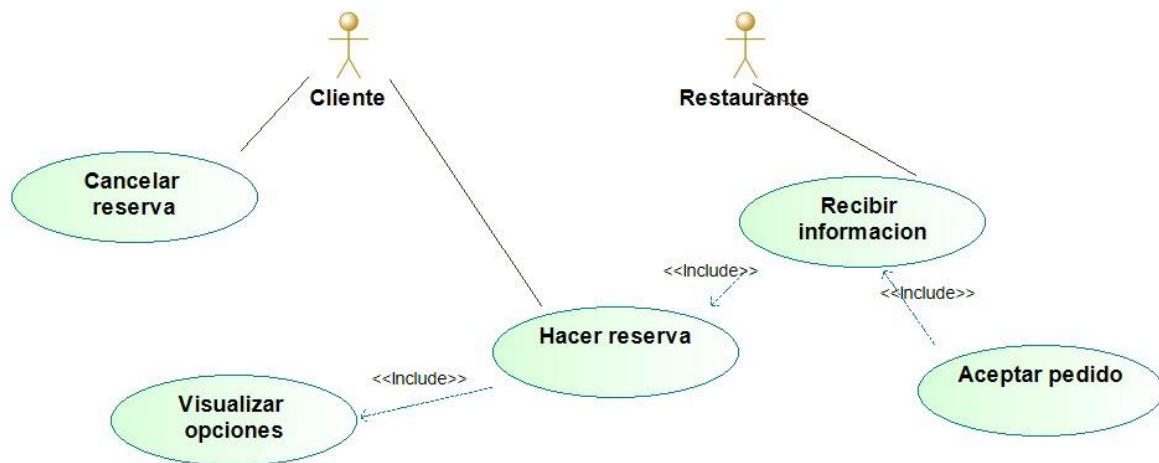
- 4.a) El cliente decide saltarse esta opción y se va al paso 6.
- 5.a) El restaurante no existe o no se encuentra en la base de datos. Se espera nueva entrada, volviendo al paso 4.
- 7.a) El cliente decide saltarse esta opción y se va al paso 8.
- 9.a) El cliente decide saltarse esta opción y se va al paso 11.
- 10.a) La ubicación introducida no existe o no es válida. Se espera nueva entrada, volviendo al paso 9.
- 12.a) El cliente decide saltarse esta opción y se va al paso 14.
- 13.a) La fecha o la hora no son válidas. Se espera nueva entrada, volviendo al paso 12.
- 15.a) El cliente decide saltarse esta opción y se va al paso 17.
- 16.a) El número introducido no es válido. Se espera nueva entrada, volviendo al paso 15.
- 18.a) El cliente decide saltarse esta opción y se va al paso 19.



## SUBSISTEMA RESERVA

Subsistema encargado de **efectuar, cancelar y aceptar** reservas.

A continuación, detallamos la información relativa a cada caso de uso.







## HACER RESERVA

**Nombre del AUTOR:** Lucía Latorre y Elena Pinuaga

- **Identificador:** R1

- **Objetivo en Contexto:**

El cliente, una vez elegido el restaurante, la comida y la localización de la mesa, introduce sus datos y hace la reserva dándole al botón de finalizar Reserva.

- **Actor principal:** Usuario cliente.
- **Actores secundarios:** Restaurante elegido y aplicación.
- **Qué datos usa:**

Los datos personales del usuario (nombre, teléfono, correo electrónico, tarjeta de crédito, etc.), las elecciones del usuario, y la información almacenada en la base de datos del restaurante elegido.

- **Precondiciones:**

El usuario está en la Página de la Aplicación y la búsqueda según filtros se ha realizado con éxito.

- **Postcondiciones:**

**Éxito:** Se muestra por pantalla “su reserva se ha efectuado correctamente”, se realizan los pagos correspondientes y el restaurante recibe la reserva.

**Fallo:** Mensajes de error y vuelta a pedir datos o termina.

- **Flujo principal**

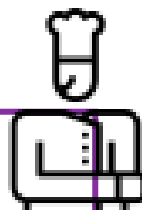
2. El usuario selecciona el restaurante en el que quiere realizar la reserva.
3. El sistema busca en su base de datos el restaurante y muestra su información general.
4. El usuario selecciona la opción “Reservar”.
5. El sistema pregunta al usuario por la fecha y la hora de la reserva.
6. El usuario introduce los datos anteriores.
7. El sistema comprueba que las fechas son compatibles con los horarios de disponibilidad del restaurante.

8. El sistema muestra los planos del restaurante junto con sus mesas disponibles.
9. El usuario elige la mesa que desea reservar o decide saltar este proceso.
10. En el caso de que se haya elegido una mesa, el sistema comprueba que está disponible ésta y se selecciona.
11. El sistema muestra la carta del restaurante dándole al usuario la opción de elegir lo que desea comer.
12. El usuario elige el o los platos que desea o, se salta este proceso.
13. En el caso de que se haya escogido el o los platos, el sistema comprueba que están disponibles y los selecciona.
14. En el caso de que el usuario haya escogido comida, el sistema solicita al usuario los datos necesarios para el pago.
15. El usuario proporciona los datos del pago.
16. El sistema comprueba que los datos de pago introducidos son válidos.
17. El sistema comprueba los datos de pago con el banco.
18. El sistema confirma la reserva con el restaurante en cuestión.
19. El sistema ejecuta el pago.
20. El usuario selecciona la opción "Finalizar Reserva".

- **Flujos secundarios**

- 6.a) El sistema no reconoce la fecha indicada o esta no está disponible.  
Se espera nueva entrada, volviendo al paso 5.
- 8.a) En el caso de que se salte este proceso, se va al paso 10.
- 9.a) La mesa seleccionada no está disponible.  
Se espera nueva entrada, volviendo al paso 8.
- 11.a) En el caso de que se salte este proceso, el sistema confirma la reserva con el restaurante en cuestión y el usuario selecciona la opción "Finalizar Reserva".
- 12.a) Alguno de los platos no está disponibles.  
Esto se comunica al usuario y le pide que vuelva a elegir la comanda respecto de los platos que ya había elegido anteriormente.
- 15.a) Los datos son inválidos. Se vuelve al paso 14.
- 16.a) Los datos del pago no concuerdan con los datos del banco. Se vuelve al paso 14.





## ACEPTAR RESERVA

Nombre del AUTOR: Noelia Pérez

- **Identificador:** R2
- **Objetivo en Contexto:**

Una vez que el cliente realiza la reserva, el restaurante recibe la información de esta y a continuación se encarga de aceptarla o no.
- **Actor principal:** Usuario restaurante.
- **Actores secundarios:** Usuario cliente.
- **Qué datos usa:**

Los datos de la reserva introducidos por el usuario (hora de llegada, pedido, mesa, etc.), los datos personales del usuario (nombre, teléfono, correo electrónico, tarjeta de crédito, etc.) y la información almacenada en la base de datos del restaurante elegido.
- **Precondiciones:**

El usuario ha realizado la reserva después de visualizar opciones habiendo indicado la hora de llegada y la forma de reserva.
- **Postcondiciones:**

**Éxito:** El sistema envía al usuario el mensaje “su reserva ha sido aceptada”, se confirman los pagos correspondientes y el cliente puede ir al restaurante.

**Fallo:** El restaurante no ha aceptado la reserva, debe realizarse otra reserva distinta o salir.
- **Flujo principal**
  1. La aplicación notifica al restaurante de la realización de una reserva.
  2. El sistema reconoce al usuario.
  3. El sistema recibe el pedido del usuario.
  4. El restaurante analiza el pedido por si hubiese algún inconveniente.
  5. En caso de que no hubiese ningún problema el restaurante acepta el pedido.
  6. El sistema envía al usuario un mensaje de aceptación de pedido.
  7. Se confirma el pago.
  8. Se cobra el dinero retenido.
  9. Finaliza el pedido.
- **Flujos secundarios**
  4. a) El restaurante recibe un pedido inapropiado (por ejemplo, pedido disponible pero cuyo precio no supera el euro como un suplemento de pan o de sazón). El restaurante rechaza el pedido y se envía al usuario un mensaje de rechazo de pedido. Se anula la retención.



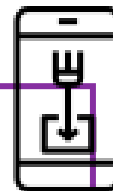


## VISUALIZAR OPCIONES

**Nombre del AUTOR:** Noelia Pérez

- **Identificador:** R3
- **Objetivo en Contexto:**  
El cliente, cuando selecciona hacer la reserva visualiza las opciones ofrecidas por la aplicación.
- **Actor principal:** Usuario cliente.
- **Actores secundarios:** Restaurante elegido y aplicación.
- **Qué datos usa:** información que almacena la aplicación.
- **Precondiciones:**  
El usuario ha seleccionado hacer reserva.
- **Postcondiciones:**  
**Éxito:** Se muestran las opciones de reserva.  
**Fallo:** Mensajes de error y vuelta a seleccionar reservar o termina.
- **Flujo principal**
  1. El usuario selecciona hacer reserva.
  2. El sistema muestra opciones de hacer reserva:
    - ▶ Reservar mesa concreta o elegirla al llegar (escoger la que queda si solo queda una)
    - ▶ y/o reservar pedido y pagarla o elegirla al llegar y pagar allí.
    - ▶ Siempre hay que establecer hora de llegada disponible.
  3. El usuario ya puede continuar y **Hacer Reserva**
- **Flujo secundario**
  - 2.a) Se produce un error al mostrar opciones de reserva. Se vuelve al punto 1.





## RECIBIR INFORMACIÓN

Nombre del AUTOR: Noelia Pérez


- **Identificador:** R4
- **Objetivo en Contexto:**  
Una vez que se ha producido **Hacer Reserva**, el restaurante recibe la información de la reserva.
- **Actor principal:** Aplicación
- **Actores secundarios:** Cliente y restaurante
- **Qué datos usa:** Los datos personales del usuario (nombre, teléfono, correo electrónico, tarjeta de crédito, etc.), las elecciones del usuario.
- **Precondiciones:**  
El usuario ha realizado la reserva según sus preferencias.
- **Postcondiciones:**  
**Éxito:** Llega correctamente la información de la reserva.  
**Fallo:** Mensajes de error al usuario.
- **Flujo principal**
  1. El restaurante recibe la reserva del cliente para proceder a aceptarla.
- **Flujos secundarios**
  - 1.a) El restaurante no recibe la información de la reserva y se le notifica al usuario. Se anula la retención del pago por parte del usuario y se le permite realizar otra reserva.



## CANCELAR RESERVA

Nombre del AUTOR: Noelia Pérez

- **Identificador:** R5
- **Objetivo en Contexto:**  
Una vez que se ha realizado la reserva y aceptado, el usuario decide cancelarla.
- **Actor principal:** Usuario cliente.
- **Actores secundarios:** Restaurante elegido y aplicación.
- **Qué datos usa:** Los datos personales del usuario (nombre, teléfono, correo electrónico, tarjeta de crédito, etc.), las elecciones del usuario, y la información almacenada en la base de datos del restaurante elegido.
- **Precondiciones:**  
El usuario está en la Página de la Aplicación y la reserva se ha realizado con éxito.
- **Postcondiciones:**  
**Éxito:** Se muestra por pantalla “su reserva se ha cancelado correctamente” y se devuelve el pago correspondiente.  
**Fallo:** Mensajes de error y vuelta a pedir cancelar o termina.
- **Flujo principal**
  1. El usuario selecciona la reserva que quiere cancelar.
  2. El sistema comprueba que queda más de 1 hora hasta la hora establecida en la reserva.
  3. El sistema busca en su base de datos el restaurante y notifica la intención de cancelación.
  4. Si el restaurante lo acepta el sistema pregunta al usuario si quiere confirmar la cancelación.
  5. El usuario la confirma.
  6. Se notifica la confirmación de cancelación al restaurante
  7. Se envía al usuario un mensaje de cancelación de reserva.
  8. El sistema ejecuta el ingreso el dinero correspondiente en la cuenta con la que había pagado el usuario si es necesario.
  9. El sistema actualiza sus horarios de reserva disponibles (se deja libre la hora de la reserva cancelada).

- 
10. El sistema actualiza las mesas disponibles del restaurante: se deja libre la mesa concreta elegida o la que ha establecido el restaurante.
  11. En caso de que se hubiese elegido un pedido el sistema actualiza su disponibilidad si es necesario (pedido con escasa disponibilidad).

- **Flujos secundarios**

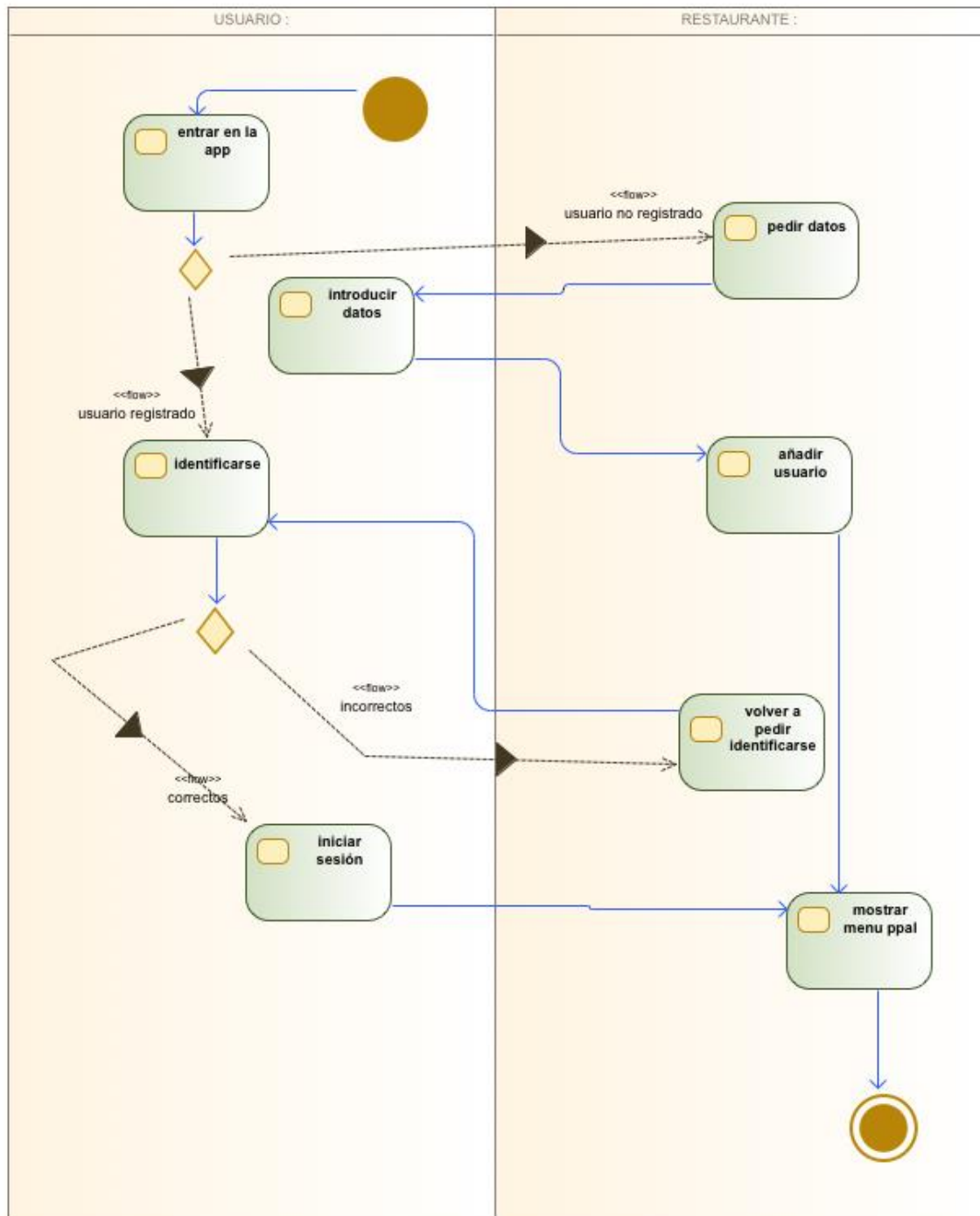
- 6.a) Si se cancela la reserva después de que quede una hora para la fecha establecida, se interrumpe el proceso. El sistema envía al usuario que no es posible cancelar la reserva.
4. a) El restaurante no permite la cancelación. El sistema envía al usuario que no es posible cancelar la reserva.
- 5.a) El usuario rechaza la cancelación. Se vuelve al paso 1 o salir.
8. a) El usuario no ha realizado ningún pedido. No es necesario ingresar dinero.
11. a) La disponibilidad del pedido siempre es alta; no es necesario actualizarla.



## DIAGRAMAS DE ACTIVIDAD

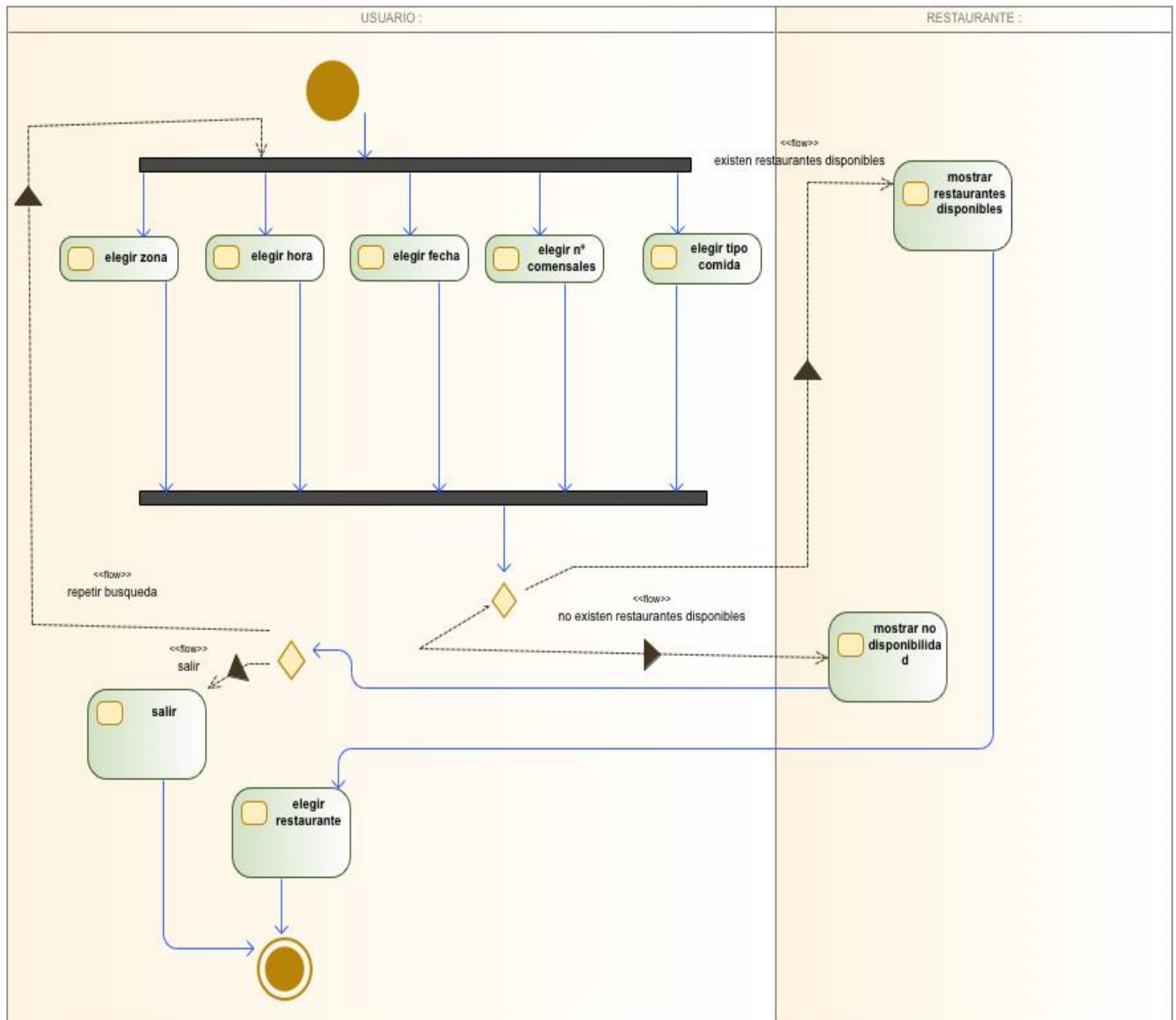
Los siguientes diagramas de actividad muestran el **flujo de actividades** que seguirá la aplicación.

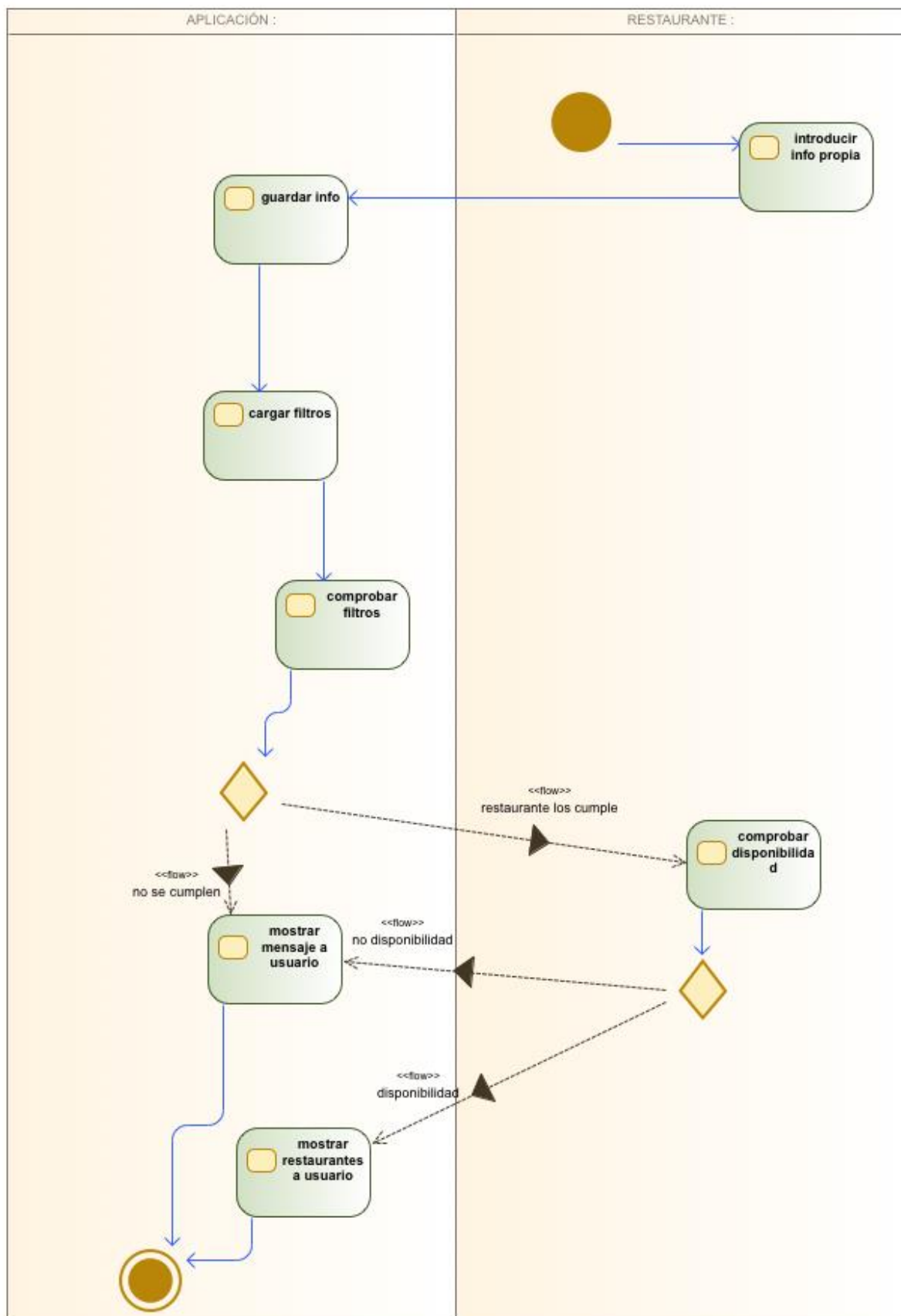
*Comenzamos con el diagrama que muestra los pasos a seguir desde que **se entra en la aplicación hasta el menú principal**.*



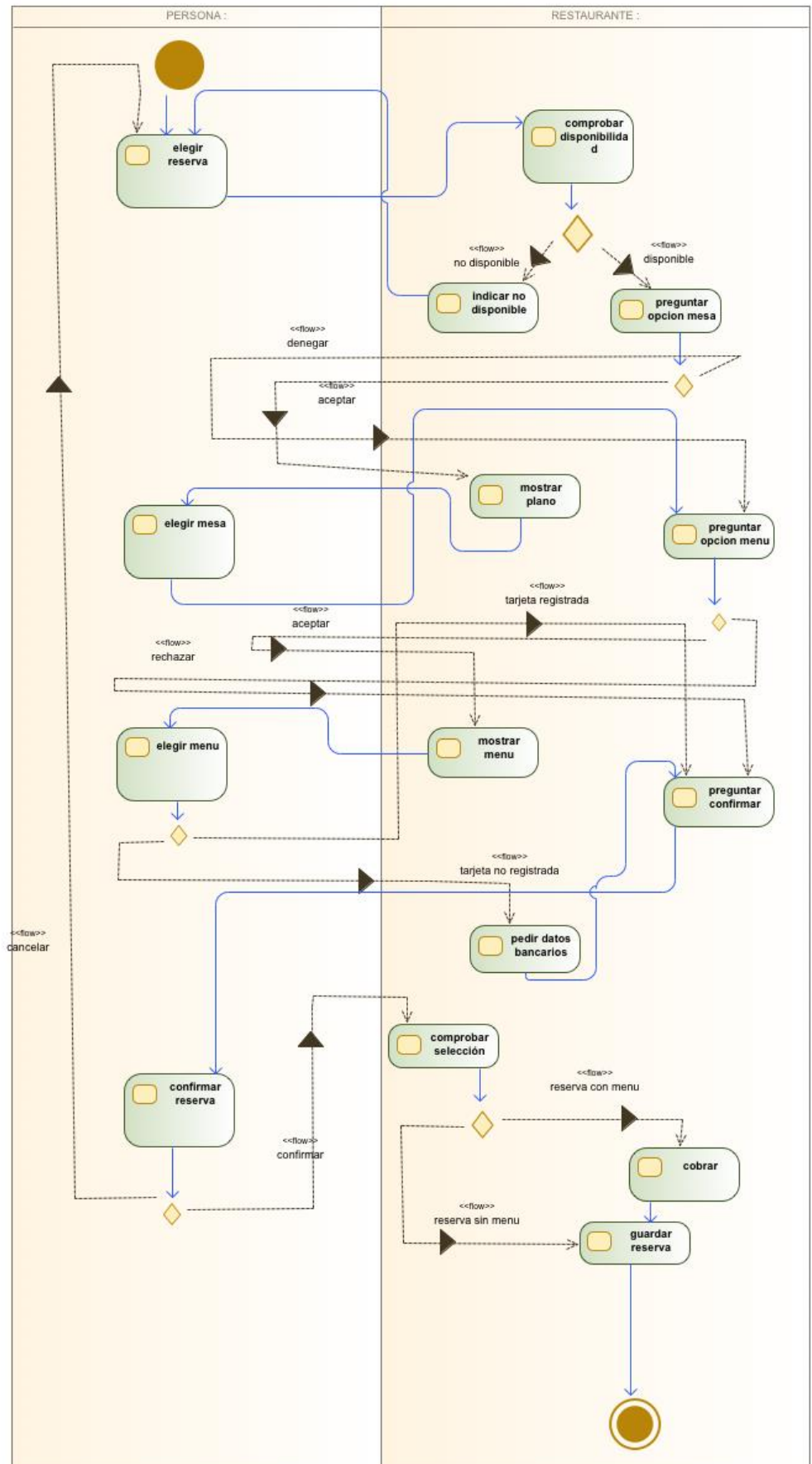


A continuación, tenemos el diagrama de actividad del proceso de **filtrar, buscar y visualizar opciones**.





Finalmente, el diagrama que muestra el flujo de actividades desde que se **elige la reserva hasta que esta se efectúa.**



## CASOS DE USO PANTALLAS

Para **ilustrar** los diagramas mostrados con anterioridad, incluimos unas pantallas de la aplicación.

**Nombre del autor: Ana Belén Duarte y Ángeles Plaza**



●●●●○ PROTO.IO 12:32 90%

Sign Up


User Name

Email


Password


Sign Up


Already have an account? [Sign In](#)


●●●●○ PROTO.IO 11:38 90% 


# Sign In



 User Name


 Password




[Don't have an account? Sign up](#) 


●●●●○ PROTO.IO

11:38

90% 



Cómo como

 FILTERS

Número de comensales

Nº

Fecha y Hora

Saturday 8<sup>th</sup> December 2018 11:38

Cerca de mi ubicación actual

☐

Me apetece comer:



Hamburguesa

---

Restaurantes disponibles:

XXX

---





hoy. 15:00. 4 pers.



09 dic



15:00



4 pers



## Goiko Grill



Hamburguesas

Precio medio: 12 €



## Lateral



Comida Mediterránea

Precio medio: 15€

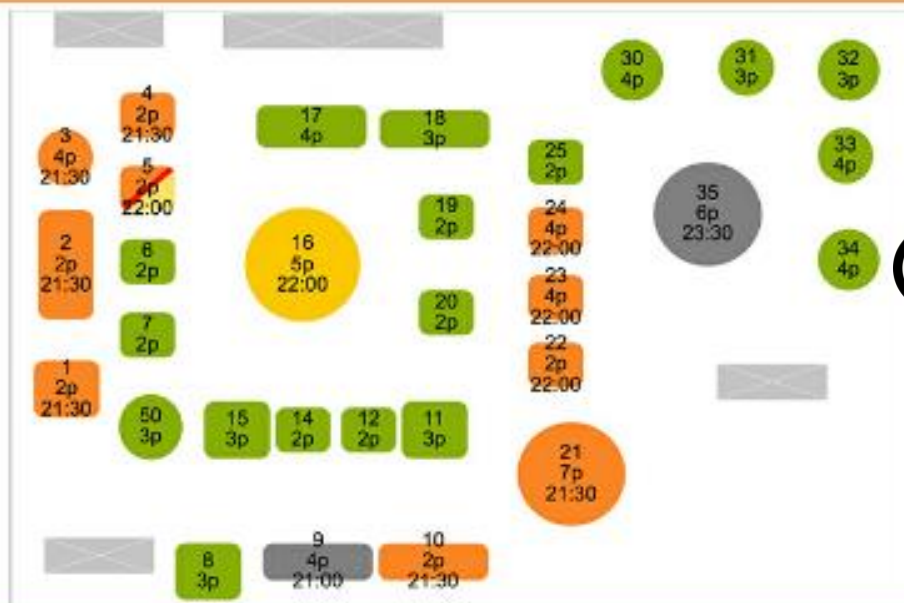








Seleccione la mesa que desee:



libre

reservada

no disponible  
para la  
aplicación

reservada  
cerca  
de tu hora  
de llegada



## Confirmación de Pedido

**Menú**

Goiko Grill Menú

**Order items:**

Chicken Waffle

Kevin Bacon Burguer

Nestea Limón (33cl)

Carrot Cake

TOTAL:

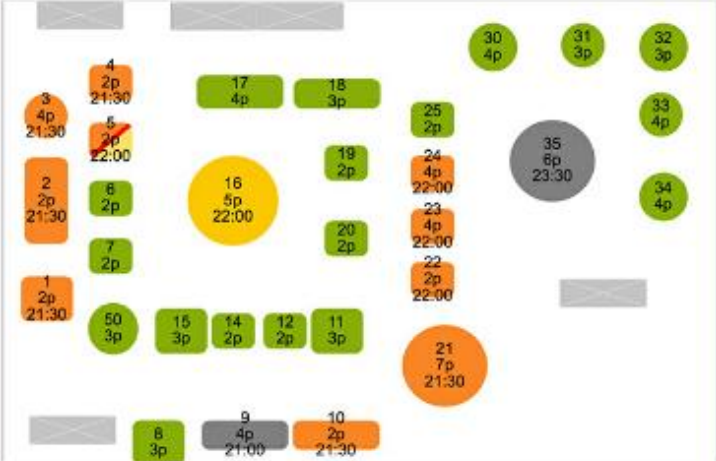
**29,99 EU**

... 1234

**Confirmar Pedido**

I have read and accept the [Terms of Service](#)  
and [Privacy Policy](#)

Cuando el cliente confirma su pedido, este aparece en la **pantalla del restaurante**.



The interface displays a floor plan of a restaurant with various tables represented by colored circles and rectangles. Each table contains a number, a time, and a status (e.g., 4p, 2p, 3p). The tables are arranged in a grid-like pattern. Below the floor plan, there are four tabs: 'All (67)', 'To be processed (52)', 'Being processed (0)', and 'Waiting (10)'. The 'To be processed' tab is currently selected. Below the tabs, there is a table with the following columns: 'Nb people', 'Time', 'Status', 'Name', 'Note', and 'Table n°'.

Nb people	Time	Status	Name	Note	Table n°
2	19:00	Confirmed	Steve Jobs	Es el cumpleaños de mi mujer. Me gustaría que la mesa tuviese	24

Below the table, there is a logo for 'cómo como' and a chef's hat icon.

## REQUISITOS DE RENDIMIENTO

Al ser una aplicación pensada para el público en general, debe soportar cuánto mayor **número de usuarios** mejor. Suponiendo un número de potenciales clientes de una ciudad como Madrid más el número de restaurantes activos, cuya mayor actividad además se concentrará en unas horas concretas del día (12h a 15h y 20h a 23h), resultan 1000 usuarios conectados simultáneamente en el caso mejor y 200 en el peor (la capacidad de soporte de usuarios se podrá ampliar en posteriores versiones).

## REQUISITOS LÓGICOS DE LA BASE DE DATOS

Todos los datos serán almacenados en la **Base de Datos**; registros de usuarios y restaurantes, información de las reservas, datos sobre los restaurantes etc. Así que es muy importante su correcto funcionamiento y que soporte muchas entradas y salidas de datos al mismo tiempo. Además, será conveniente contar con **copias de seguridad** en caso de que ocurra algún fallo.

La información contenida en la **Base de Datos** es la siguiente:

### RESTAURANTES

- Ubicación del restaurante
- Menús
- Distribución de las mesas y capacidad

### PERSONAS

- Datos de registro

## RESTRICCIONES DE DISEÑO

Como vamos a programar la aplicación en **java**, es necesario que la implementación sea **orientada a objetos**, organizada en paquetes y modular.

## ATRIBUTOS DEL SISTEMA SOFTWARE

### DISPONIBILIDAD

La aplicación estará **disponible en todo momento** para cualquier usuario. En caso de querer realizar cambios siempre se hará mediante una nueva versión; y en caso de fallo en la base de datos se recurrirá a las copias de seguridad.

### SEGURIDAD

La información de la base de datos debe seguir la normativa recogida por la **Agencia Española de Protección de Datos** y el sistema debe realizar todas las acciones de encriptación y protección red para que estos no se vean comprometidos.

Las **contraseñas** de los usuarios deberán ser **encriptadas** y se exigirán unos **requisitos mínimos** para garantizar su seguridad.

La **información**, además, deberá estar debidamente **protegida** para evitar que ordenadores se conecten a la red del servidor y accedan a información clasificada.

### MANTENIBILIDAD

En caso de fallo o saturación, los encargados de arreglarlo y mantener todo en orden serán los **administradores**.

### ACCESIBILIDAD

La aplicación permite que **varios terminales** se conecten **a la vez** y gestionen los datos simultáneamente. Además, como ya hemos mencionado en apartados anteriores, la **conexión** con el servidor es **inalámbrica**, permitiendo que cualquier dispositivo móvil con conexión a internet y la aplicación descargada pueda conectarse a ella.

## BIBLIOGRAFÍA

- **FLATICON:** añadidos gráficos de la documentación  
<https://www.flaticon.com/>
- **PROTO.IO:** diseño de las pantallas  
<https://proto.io/>
- **MODELIO:** diagramas de casos de uso y de actividad  
<https://www.modelio.org/>
- **Apuntes** de la asignatura Ingeniería del Software (UCM Fac. Infor. DG, 18/19)