

*Diseño de Alto Nivel*

*Versión 1*

*Grupo 05*

*Jorge González Valencia*

*Beatriz Lacruz Alcaraz*

*Alfonso Gerardo Mateos Pérez-Iñigo*

*Alberto Ogbechie Condés*

*Diego Ortiz Moreno*

Historial de versiones

|  |  |  |
| --- | --- | --- |
| **Versión** | **Notas** | **Petición de cambio relacionada** |
| 1 | Versión inicial | PC\_1 |

Índice

[1. Diagrama de clases](#h.simkupyhjlpw)

[1.3 Relaciones](#h.gxjpo1kliiot)

[2. Modelo entidad-relación](#h.jof9h5v27b6c)

[3. Casos de uso](#h.wgibm5dbh182)

[3.1 Login](#h.wjf4lceavxst)

[3.2 Registrarse](#h.57xosp76xxoc)

[3.3 Consulta acciones](#h.hlq8y5ff9dox)

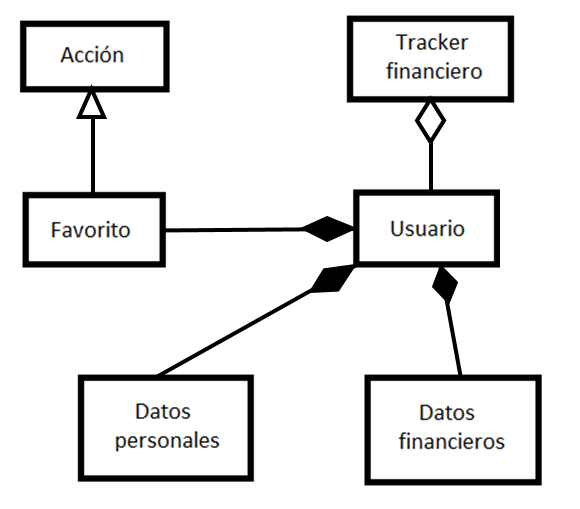
[3.4 Consultar favoritos](#h.peaecm84s52v)

[3.5 Agregar/Quitar favoritos](#h.sl1q2hz81u0i)

# 1. Diagrama de clases

**1.1 Introducción**

En este primer diagrama de alto nivel se pretende dar una visión de las clases existentes en el proyecto de tracker financiero y las relaciones que existen entre ellas. El diagrama de clases obtenido es el siguiente:



**1.2 Clases**

Se han obtenido en total ocho clases que representan cada uno de los objetos presentes en la aplicación. Son las siguientes:

* **Acción:** representa las acciones del servidor externo que se podrían consultar en el sistema.
* **Favorito:** acción que el usuario elige como favorita y se añade a su lista favoritos.
* **Tracker financiero:** objeto que representa el sistema o aplicación de tracker financiero, es quien principalmente realiza las funcionalidades.
* **Usuario:** persona que hace uso de la aplicación.
* **Datos personales:** son parte de los datos que el usuario introduce al sistema, nombre y apellidos, teléfono, etc.
* **Datos financieros:** estos datos también forman parte del usuario pero en este caso relevantes a su lugar de trabajo, profesión, cargo, salario, etc.

## 1.3 Relaciones

En el diagrama se dan únicamente tres relaciones que se explicarán a continuación:

* **De herencia:** la clase hereda los atributos y métodos de la clase de la que hereda, en este caso únicamente la clase favoritos hereda de acción, ya que tiene las mismas características añadiendo que ha sido elegida por el usuario como favorito.
* **De agregación:** una clase es parte de otra clase, en este caso un usuario forma parte de la aplicación pero hay una relación más débil.
* **De composición:** la mayoría de las relaciones en el diagrama son de este tipo, donde una clase forma parte de otra y hay una relación más fuerte donde, por ejemplo, si se borra un usuario también se borrarán sus datos personales, preferencias, etc.

# 2. Modelo entidad-relación

A continuación se muestra con más detalle las entidades y los atributos que hemos obtenido:

* USUARIO: Para cada usuario almacenaremos un identificador de usuario (id) y su nombre de usuario, también almacenaremos el email y la contraseña que va a utilizar el usuario para realizar el login en la aplicación.
* ACCIÓN: Las acciones que el usuario puede tanto consultar como marcar como favoritas se identificarán por su código (cod), también se almacenará información como su nombre, el valor actual (v\_actual), el volumen acumulado (v\_acumulado) y el valor de la jornada anterior (v\_jant).
* DATOS PERSONALES: Como datos personales del usuario se almacenará el nombre, los apellidos, el teléfono, la dirección, la fecha de nacimiento (f\_nac) y un identificador de los datos personales.
* DATOS FINANCIEROS: Al igual que en la entidades anteriores almacenaremos un identificador para los datos financieros, además guardaremos la profesión, la ocupación, lugar de trabajo, el cargo y el income anual.

# 3. Casos de uso

## 3.1 Login

El usuario utiliza el sistema para registrarse con su nombre de usuario y contraseña. El sistema consulta la base de datos para comprobar si el nombre y la contraseña son correctos. Si es así, el sistema le muestra en pantalla sus listas de tareas.

**Objetivo**

Acceder al sistema

**Actores**

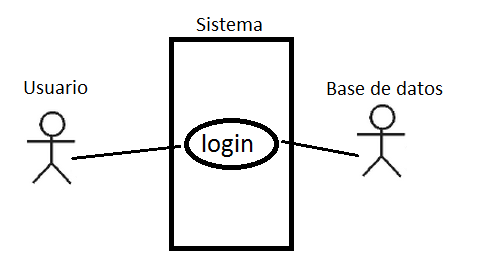
En este caso intervienen dos actores: el actor principal o **usuario** y la **base de datos** donde se contrastará el usuario y contraseña introducidos.

**Sistemas externos**

No encontraremos sistemas externos.

**Diagrama**

El diagrama que representa el caso de uso es el siguiente:



# 

## 3.2 Registrarse

El usuario envía una petición de crear usuario y el sistema le muestra la pantalla con los campos requeridos. El usuario rellena correctamente los datos: nombre, apellidos, dirección, teléfono, email, fecha de nacimiento, contraseña, income anual, ocupación, lugar de trabajo, profesión y cargo

**Objetivo**

Crear una cuenta en el sistema

**Actores**

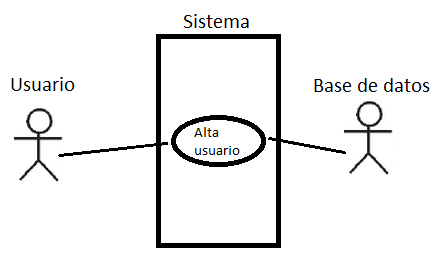
En este caso intervienen dos actores: el actor principal o **usuario** y la **base de datos** donde se almacenarán los datos introducidos.

**Sistemas externos**

No encontraremos sistemas externos.

**Diagrama**

El diagrama que representa el caso de uso es el siguiente:



## 3.3 Consulta acciones

El usuario selecciona la pestaña de visualizar acciones y el sistema interactúa con un sistema externo que le devuelve el valor actual de las acciones. Finalmente el sistema muestra por pantalla los datos recibidos.

**Objetivo**

Consultar el valor de las acciones actualmente.

**Actores**

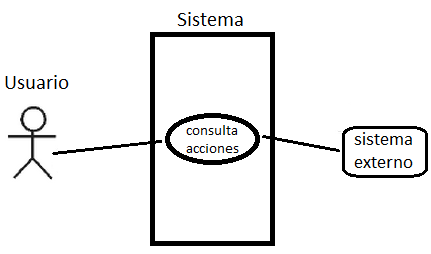
En este caso sólo interviene el actor principal o **usuario**.

**Sistemas externos**

En este caso de uso participará un sistema externo que proporciona los valores actuales de las acciones.

**Diagrama**

El diagrama que representa el caso de uso es el siguiente:



## 3.4 Consultar favoritos

El usuario selecciona la pestaña de visualizar favoritos. Acto seguido el sistema consulta con la base de datos cuales son las acciones que componen los “favoritos” del usuario y después se hace una consulta al sistema externo para obtener sus valores. finalmente se despliega el resultado por pantalla.

**Objetivo**

Consultar los valores de los “favoritos” del usuario

**Actores**

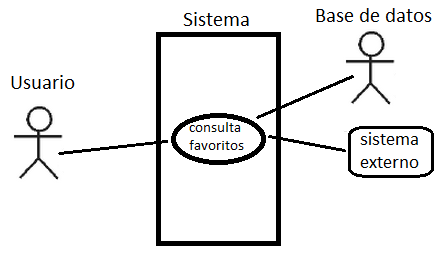
En este caso intervienen dos actores: el actor principal o **usuario** y la **base de datos** donde se almacenan las acciones “favoritas” del usuario.

**Sistemas externos**

En este caso de uso participará un sistema externo que proporciona los valores actuales de las acciones.

**Diagrama**

El diagrama que representa el caso de uso es el siguiente:



## 3.5 Agregar/Quitar favoritos

El usuario hará click en el botón de favorito correspondiente a una acción. Posteriormente se agregará/eliminará dicha acción de la base de datos.

**Objetivo**

Agregar/eliminar una acción de la base de datos de forma que se añada/elimine de sus favoritos

**Actores**

En este caso intervienen dos actores: el actor principal o **usuario** y la **base de datos** donde se almacenarán/eliminarán las acciones seleccionadas.

**Sistemas externos**

No encontraremos sistemas externos.

**Diagrama**

El diagrama que representa el caso de uso es el siguiente:

