

*Diseño de Bajo Nivel Ciclo 1*

*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 específico

1.1 Introducción

1.2 Clases

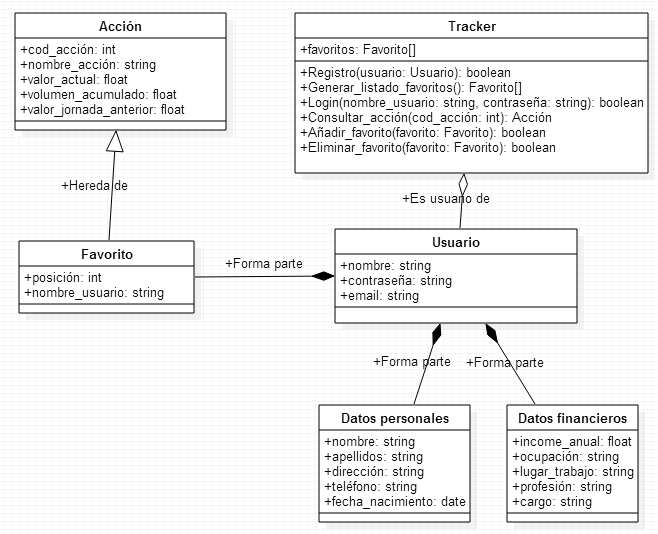
1.3 Relaciones

2. Paso a tablas

# 1. Diagrama de clases

**1.1 Introducción**

En el diagrama de clases específico se consigue una visión más detallada del modelo. Además de mantenerse las clases y relaciones que existen entre ellas, ya se pueden observar atributos referentes a cada clase y los métodos que contienen cada una. Más adelante se explicará en detalle los distintos componentes. En este diagrama solo aparecen las clases y métodos contenidos dentro del ciclo1.



**1.2 Clases**

Las clases presentes son las ocho clases que se obtuvieron en el diseño de alto nivel, pero detalladas. En cada una de las clases se han añadido los atributos correspondientes pero solo en *Tracker financiero*, que es la que realiza la mayor parte de la funcionalidad, y en *preferencias* se han añadido métodos. Aun así, por claridad, no se han especificado los métodos básicos en cada una de las demás clases, que serían los distintos **set** y **get** de cada uno de los atributos de las clases y solo se mantienen los que realizan alguna funcionalidad concreta de la aplicación.

* **Acción:** los atributos definidos son
  + cod\_acción: es un número (podría ser string) que identifica a la acción.
  + nombre\_acción: guarda el nombre de la acción.
  + valor\_actual: guarda el valor que tiene en ese momento concreto la acción.
  + volumen\_acumulado: representa el valor que ha acumulado la acción.
  + valor\_jornada\_anterior: guarda el valor que se obtuvo en la jornada anterior.
* **Favorito:** hereda de acción, heredando sus métodos y atributos y añadiendo los siguientes:
  + posición: número que indica su posición en la lista de favoritos.
  + nombre\_usuario: nombre del usuario al que pertenece dicha acción.
* **Tracker financiero:** objeto que representa la aplicación, añade un atributo:
  + favoritos: lista de favoritos del usuario que guarda las acciones que desea.

y los siguientes métodos:

* + Añadir\_favorito: añade el favorito pasado como parámetro a la lista de favoritos del usuario. Devuelve mensaje de éxito o error mediante booleano.
  + Eliminar\_favorito: elimina de la lista de favoritos del usuario el favorito pasado por parámetro. Devuelve mensaje de éxito o error mediante booleano.
  + Registro: registra el usuario pasado como parámetro en el sistema y devuelve un mensaje según se ha realizado o no con éxito mediante un booleano.
  + Generar\_listado\_favoritos: devuelve la lista de favoritos del usuario.
  + Login: permite al usuario iniciar sesión en el sistema mediante el usuario y contraseña.
  + Consultar\_acción: devuelve la acción cuyo código es el pasado por parámetro.
* **Usuario:** se le añaden los siguientes atributos:
  + nombre: cadena de caracteres con el nombre del usuario.
  + contraseña: clave del sistema del usuario.
  + email: email que registra el usuario que servirá para recuperar la contraseña.
* **Datos personales:** contiene los datos personales que el usuario registra en el sistema, los cuales son:
  + nombre y apellidos: nombre completo del usuario.
  + dirección: lugar de residencia del usuario.
  + teléfono: número de teléfono de contacto del usuario, se usa una cadena para llamadas internacionales.
  + fecha\_nacimiento: fecha de nacimiento del usuario.
* **Datos financieros:** mismo uso que los datos personales pero con datos financieros:
  + income\_anual: salario anual del usuario.
  + ocupación: ocupación del usuario.
  + lugar de trabajo: dirección de trabajo del usuario.
  + profesión: nombre de la profesión a la que se dedica el usuario.
  + cargo: cargo que desempeña el usuario en su trabajo.

## 1.3 Relaciones

Se mantienen las mismas relaciones que en el diseño de alto nivel, heredando favoritos de acción, relacionándose usuario con el tracker mediante agregación (relación débil) y los componentes de usuario con usuario mediante composición (relación fuerte).

# 2. Paso a tablas

