

Informe de Arquitectura y Diseño

Fernando Azcorreta Fernández y Oscar Herrero Casado

Introducción

En este documento se realizará la descripción tanto de la arquitectura como del diseño de dominio que se han llevado a cabo para el desarrollo de la aplicación. Para ellos se usarán como ayuda diferentes imágenes llevadas a cabo por medio de la herramienta *MagicDraw*.

Diseño de dominio

Para el diseño de la aplicación nos hemos basado en el diseño dado en Moodle, y nos hemos limitado a realizar leves cambios en el mismo. De esta forma hemos revisado las multiplicidades y le hemos añadido un campo *id* a cada una de las clases para facilitar el mapeado en la parte de persistencia. También hemos añadido un atributo booleano a la clase *Estacionamiento* para facilitar el tener controlado si un estacionamiento ha finalizado o no. Se muestra en la siguiente imagen (Figura 1).

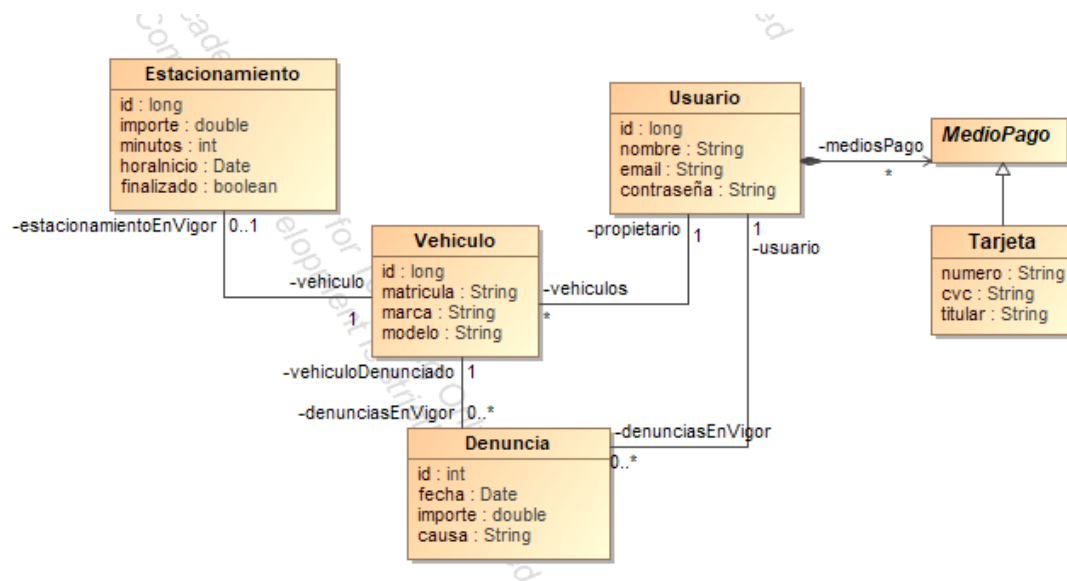


Figura 1: Diseño de Dominio

Arquitectura de la aplicación

Dado que se trata de una aplicación basada en componentes se dividió la misma en 3 capas y se diseñaron los componentes necesarios para cada capa (Figura 2).

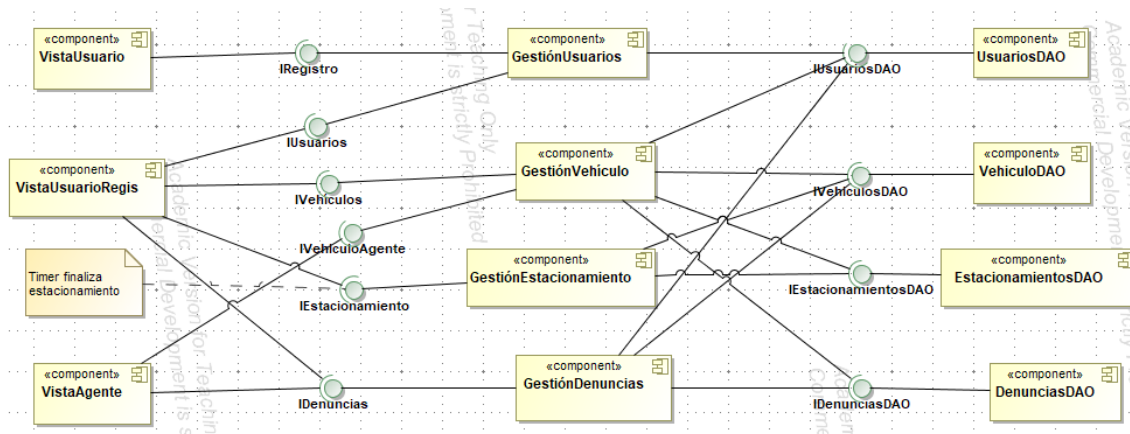


Figura 2: Diseño de la Arquitectura

Del mismo modo se ha especificado cada una de las interfaces (Figura 3).

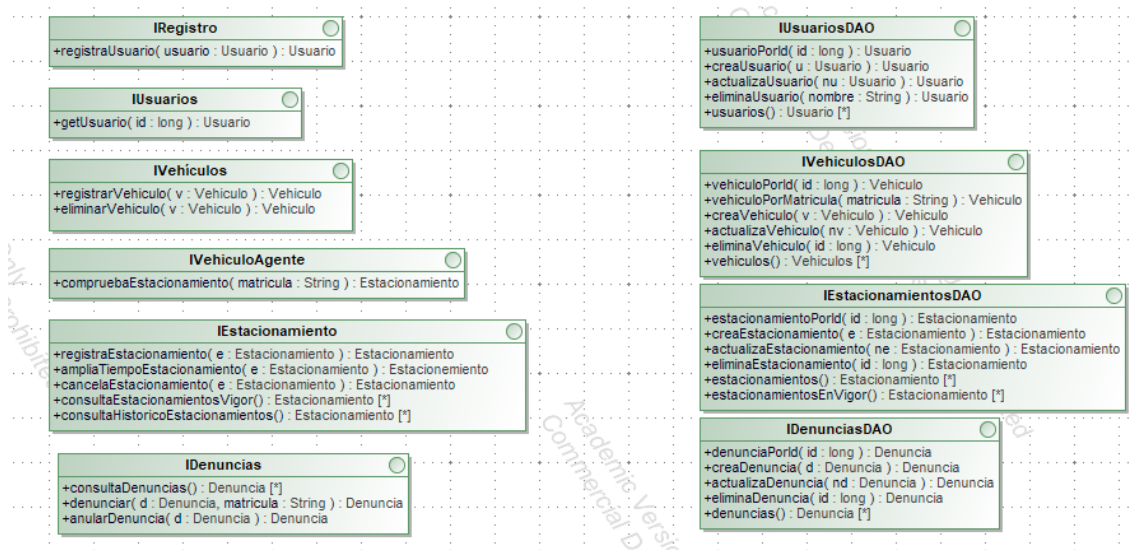


Figura 3: Especificación de las Interfaces

● Capa DAO

En esta capa se encuentran los componentes relacionados con la base de datos. Se ha creado un componente por cada uno de los elementos que se quiere gestionar, resultando en la creación de 4 componentes: *UsuariosDAO*, *VehiculosDAO*, *EstacionamientosDAO* y *DenunciasDAO*. Se ha llevado a cabo la implementación de todos y cada uno de ellos.

● Capa de Negocio

En esta capa se encuentran los componentes relacionados con las operaciones de negocio del sistema. De nuevo se han creado e implementado 4 componentes, los cuales están directamente relacionados con los anteriores: *GestiónUsuarios*, *GestiónVehiculos*, *GestiónEstacionamientos* y *GestiónDenuncias*.

Tanto *GestiónUsuarios* como *GestiónVehiculos* cuentan con dos interfaces diferentes para diferenciar las operaciones que pueden llevar a cabo diferentes roles. En el caso de *GestiónDenuncias* también se podría haber hecho, pero hemos decidido que no merecía la pena

diferenciar las operaciones solo por una de ellas teniendo en cuenta que la operación podría ser válida para ambos roles perfectamente.

Para la *GestiónEstacionamientos* también se le da acceso al timer al ser necesario para una de sus operaciones (en la práctica no se ha implementado).

- **Vistas**

Para la vista se ha creado 3 componentes diferenciados, uno por cada uno de los roles posibles en la aplicación. En la práctica se implementa la *VistaUsuario*, a partir de la cual se accede a la *VistaUsuarioRegis* por medio de la interfaz *IRegistro*, y posteriormente esta vista hace uso de *IEstacionamientos*.