MODELO DE COMPONENTES

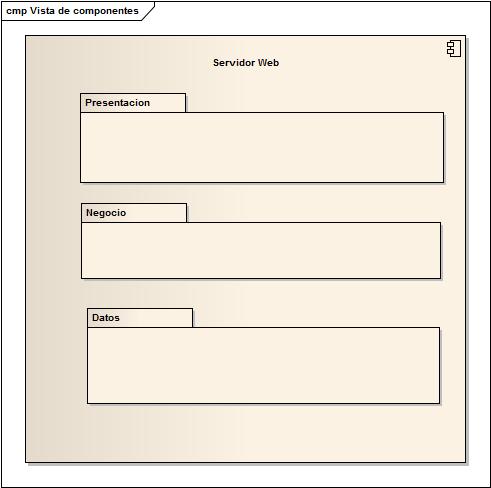
# Introducción

El siguiente documento lo que trata de mostrar es la distribución de los principales componentes utilizados y los cuales componen el sistema GeoParking. Las imágenes a continuación son pequeños modelos que tratan de esquematizar la forma en que se estructura los componentes del sistema.

## Vista de Componentes

Aquí se puede observar como nuestro sistema el componente principal está constituido o estructura en una división por capas, las cuales son:

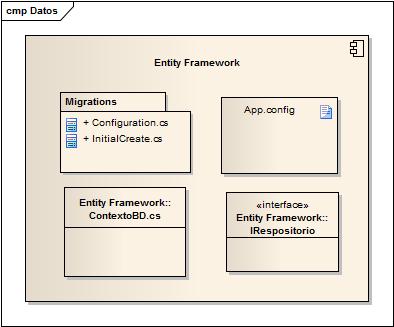
1. **Datos**: capa dedicada a la materialización y desmaterialización de los datos de la aplicación.
2. **Negocio**: capa dedicada a aplicar la lógica del negocio en el sistema y todas aquellas reglas que rigen su funcionamiento.
3. **Presentación**: capa dedicada a brindar una interfaz visual con la cual los usuarios interactúen con el sistema.



## Vista de Datos

Aquí se puede observar como esta capa está basada en la tecnología Entity Framework, empleando el modelo Codefirst. Para ello estos son los principales componentes que permiten en funcionamiento de esta capa del sistema:

* **Migrations**: este paquete contiene dos clases muy importantes en el funcionamiento de Entity Framework
  + **Configuration.cs**: tiene establecido el contexto de la BD donde se montara la conexión para el mapeo de los objetos.
  + **InitialCreate.cs**: tiene establecido la creación del modelo en memoria de los objetos para el posterior mapeo.
* **App.config**: tiene establecido la conexión a la instancia de la Base de Datos. Y además la versión de Framework con la que debe realizar el mapeo objeto relacional.
* **ContextoBD.cs**: crea el contexto en memoria en relación a los objetos presentes en la Base de Datos. Establece contexto particulares para cada objeto del sistema.
* **IRepositorio:** interfaz que establece las acciones que se pueden realizar sobre cada objeto del sistema en cada contexto.



## Vista de Negocio

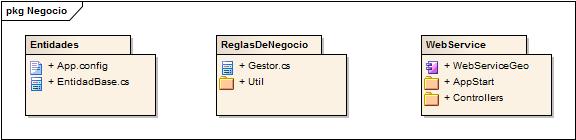
Aquí se puede observar como esta capa está estructurada y subdividida en tres paquetes, los cual tienen funcionalidades muy diferentes pero que así ves se relacionan para cumplir un objetivo en común, brindar la funcionalidad necesaria al usuario aplicando las distintas reglas del negocio. En esta capa se divide en:

1. Entidades
2. ReglasDeNegocio
3. WebService

**Entidades:** alberga todas las clases que representan los objetos manejados en la aplicación.

**ReglasDeNegocio:** alberga todos los controladores que aplican las distintas reglas de negocio sobre los objetos del sistema.

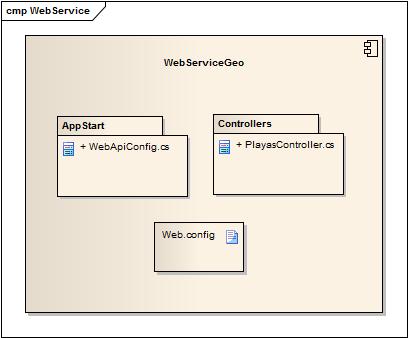
**WebService:** alberga la aplicación que se corresponde al webService para poder ofrecer la funcionalidad del sistema a la aplicación móvil.



## Vista de WebService

Aquí se puede observar que el componente principal es el servicio web WebServiceGeo, el cual interactúa con los demás componentes de la capa de negocio para que la funcionalidad que provee el sistema esté disponible a través de la aplicación móvil. Este servicio web se compone de:

* **WebApiConfig.cs:** establece el formato de ruta con la cual se puede acceder a través de un protocolo de internet a las distintas funcionalidades que ofrece el WebService.
* **Web.config:** establece la conexión con la BD.
* **PlayasController.cs:** ejemplo de unos de los controladores que brinda la funcionalidad del WebService, es a estos métodos a los cuales accede la aplicación móvil a través del formato de ruta establecida en WebApiConfig.



## Vista de Presentación

Aquí se puede observar como esta capa del sistema está conformada por una aplicación web donde se pueden observar componentes como:

* **Site.master:** platilla modelo de interfaz de donde heredan las demás páginas.
* **Index.aspx**: ejemplo de página que hereda de la master.
* **Web.config**: configuración de distintos aspectos de la aplicación web.

