|  |  |
| --- | --- |
| Geo Parking | Modelado del Sistema  2014 |

El documento describe información del modelado del sistema durante el desarrollo del producto Geo Parking.

# Control de la documentación

### Control de la Configuración.

|  |  |
| --- | --- |
| Título: | Modelado del Sistema |
| Referencia: | GeoP\_Proyecto\_ModeladoDelSistema.docx |
| Autores: | Lucas Toneatto |
| Fecha: | 30/09/2014 |

### Histórico de Versiones.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Versión | Fecha | Estado | Responsable | Cambios |
| 1.0\_DraftA | 230/09/2014 | Pendiente de Revisión | Lucas Toneatto[autor] |  |
|  |  |  |  |  |
|  |  |  |  |  |

Contenido

[Control de la documentación 2](#_Toc399880993)

[Control de la Configuración. 2](#_Toc399880994)

[Histórico de Versiones. 2](#_Toc399880995)

[Introducción 5](#_Toc399880996)

[MODELO DE NEGOCIO 5](#_Toc399880997)

[MODELO DE INTERACCION 5](#_Toc399880998)

[Búsqueda Playas por Ciudad 5](#_Toc399880999)

[Búsqueda de Playas por Ciudad por Filtro 6](#_Toc399881000)

[Login de Usuario 7](#_Toc399881001)

[Registro de Usuario 8](#_Toc399881002)

[Registro de Playa 9](#_Toc399881003)

[MODELO DE ARQUTECTURA/DEPLIEGUE 9](#_Toc399881004)

[Servidor 10](#_Toc399881005)

[ Presentación 10](#_Toc399881006)

[ Negocio 10](#_Toc399881007)

[ Entity Framework 10](#_Toc399881008)

[ SQL Server 10](#_Toc399881009)

[Cliente Web 10](#_Toc399881010)

[Cliente Móvil 10](#_Toc399881011)

[MODELO DE COMPONENTES 11](#_Toc399881012)

[Vista de Componentes 11](#_Toc399881013)

[Vista de Datos 11](#_Toc399881014)

[Vista de Negocio 12](#_Toc399881015)

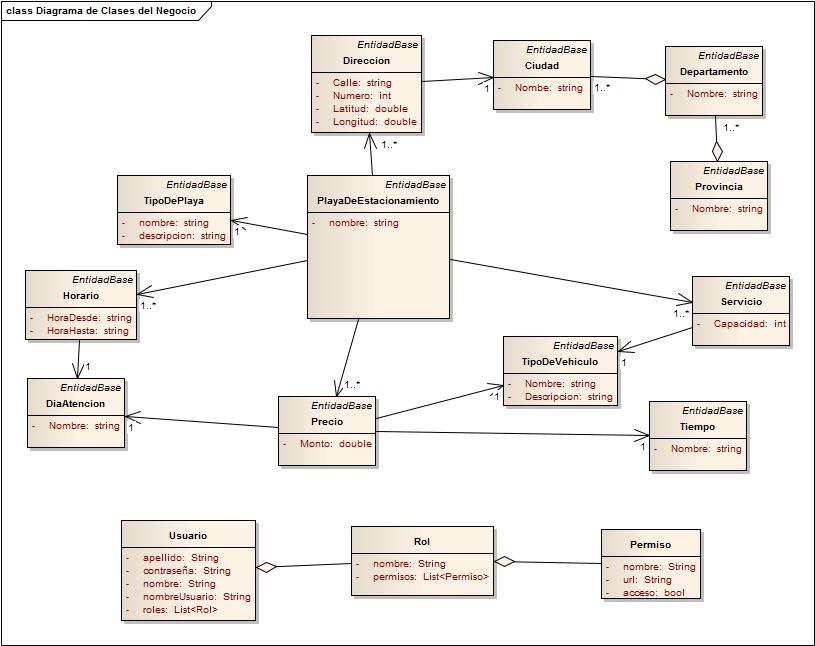
[Vista de WebService 13](#_Toc399881016)

[Vista de Presentación 13](#_Toc399881017)

# 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.

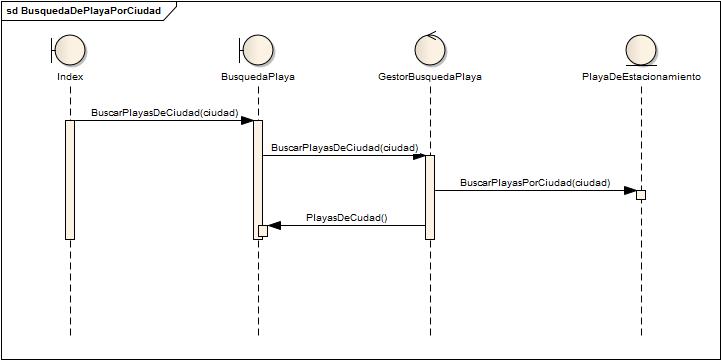
# MODELO DE NEGOCIO



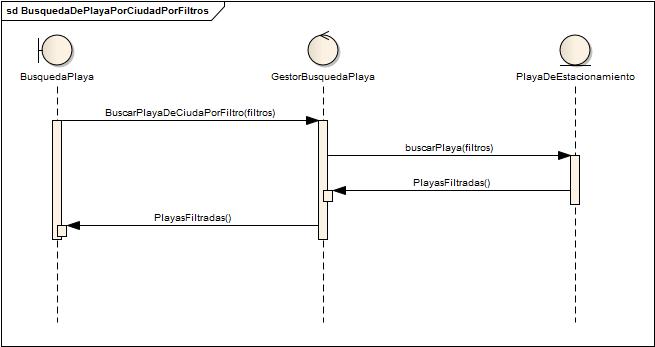
# MODELO DE INTERACCION

En los siguientes diagramas de secuencia lo que se trata de mostrar es la interacción general de como los componentes del sistema interactúan entre sí para poder brindar la funcionalidad que se tiene como objetivo.

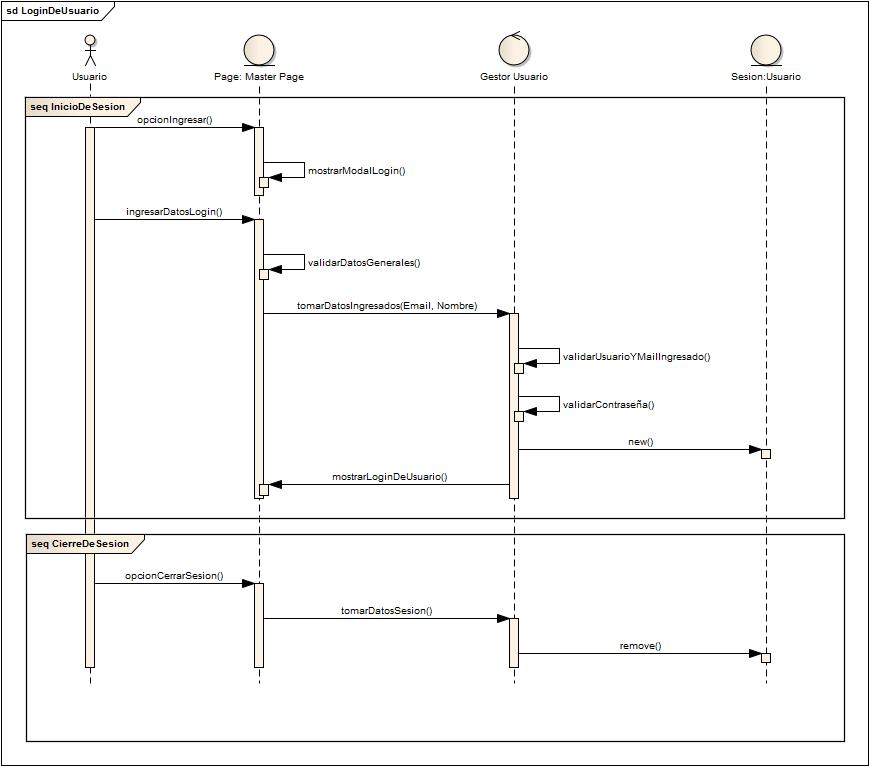
### Búsqueda Playas por Ciudad



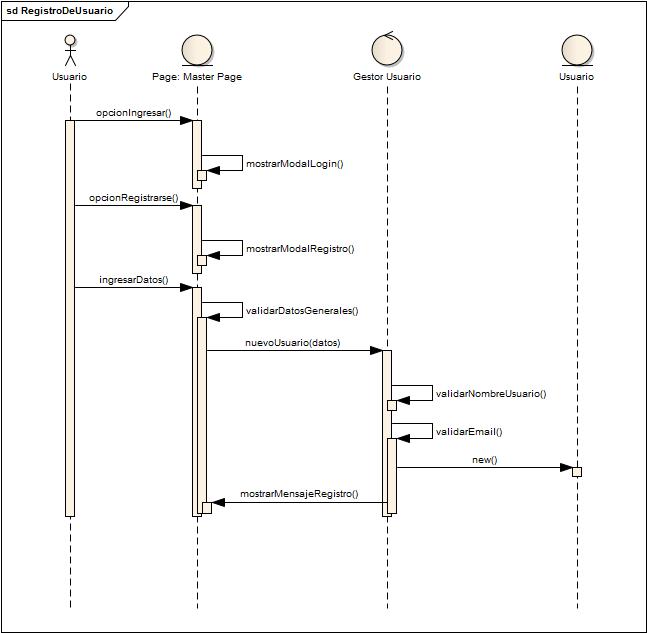
### Búsqueda de Playas por Ciudad por Filtro



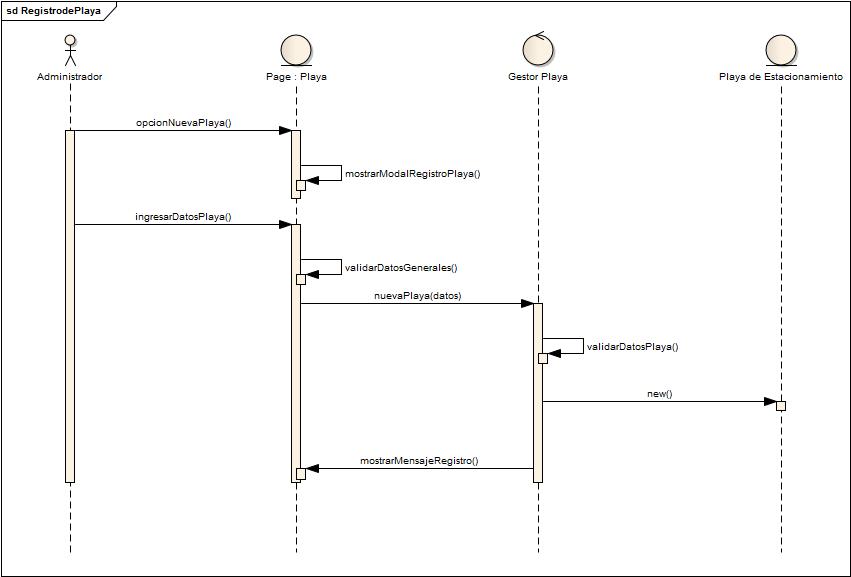
### Login de Usuario



### Registro de Usuario



### Registro de Playa



# MODELO DE ARQUTECTURA/DEPLIEGUE

La arquitectura que presenta el sistema GeoParking es como el diagrama que se muestra a continuación. La arquitectura se organiza en 3 bloques principales que están en contante interacción: El servidor de la aplicación, el cliente Web y el Cliente Móvil.

## C:\Users\Lukas\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Modelo de arquitectura.jpgServidor

En este nodo se despliegan 4 componentes, los cuales están encargados de realizar o brindar distintas funciones, es por ello que la arquitectura a desplegar en el servidor está organizada en capas. Las capas son las siguientes:

* Presentación: es el componente encargado de albergar todo aquello que genere una forma de interacción con el usuario, será aquella interfaz de comunicación a través de la cual se ofrezca las funcionalidades del sistema.
* Negocio: es el componente encargado de albergar todo aquello que establezca la manera en que funcionara el sistema, y las reglas que rigen ese comportamiento. Este componente se subdivide a su vez en:
  + **Entidades**: contendrá todos aquellos elementos que formen parte del problema que trata el sistema como a su vez también elementos para la solución de los mismos. Dichos elementos forman parte del modelo del negocio como del modelo de solución.
  + **Reglas de Negocio**: contendrá todo aquellos agentes que regulan el funcionamiento del sistema, y maneja el mundo de entidades que lo conforman.
  + **Web Service**: será el componente que brinde los servicios del sistema a la aplicación móvil, la cual está desarrollada de forma externa al sistema.
* Entity Framework: es aquel componente encargado del mapeo Objeto-Relacional de las entidades del sistema.
* SQL Server: es la instancia de la Base de Datos que alberga toda la información del sistema.

## Cliente Web

Este nodo contendrá el **navegador** a través del cual se realiza la conexión con nuestro sistema mediante de peticiones Http (Internet).

## Cliente Móvil

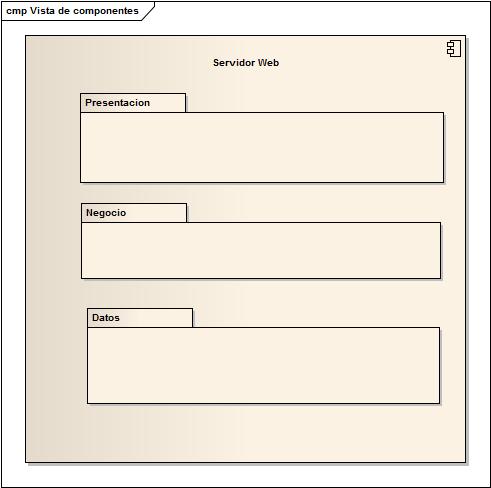
Este nodo contendrá la **aplicación móvil** desarrollada para que consuma los servicios del Web Service deployado en el Servidor y así podrán brindar las funcionalidades del sistema.

# MODELO DE COMPONENTES

## 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:

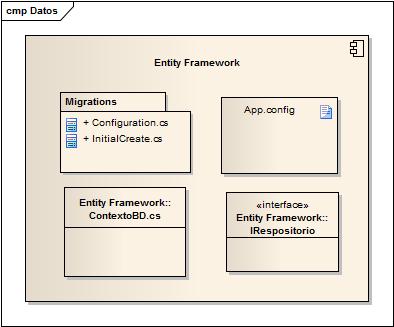
1. **Datos**: capa dedicada a la materialización y desmaterialización de los datos de la aplicación.
2. **Negocio**: capa dedicada a aplicar la lógica del negocio en el sistema y todas aquellas reglas que rigen su funcionamiento.
3. **Presentación**: capa dedicada a brindar una interfaz visual con la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

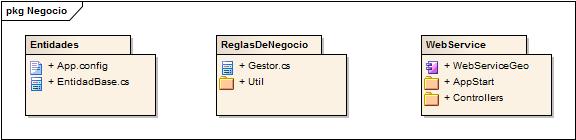
Aquí se puede observar como esta capa está estructurada y subdividida en tres paquetes, los cual tienen funcionalidades muy diferentes pero que así ves se relacionan para cumplir un objetivo en común, brindar la funcionalidad necesaria al usuario aplicando las distintas reglas del negocio. En esta capa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

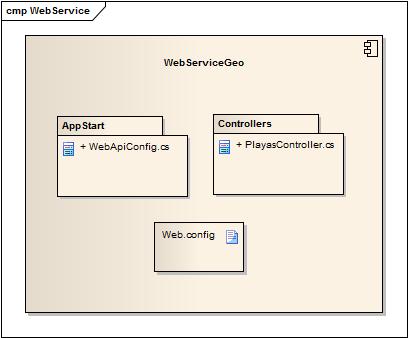
**WebService:** alberga la aplicación que se corresponde al webService para poder ofrecer la funcionalidad del sistema a la aplicación móvil.



## 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.

