GEO PARKING

Modelado del Sistema

2014

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	30/09/2014	Pendiente de Revisión	Lucas Toneatto[autor]	
1.1_DraftB	14/10/2014	Pendiente de Revisión	Lucas Toneatto	
1.1	25/10/2014	Aprobado	Leonel Romero	

Contenido

INTRODUCCIÓN

El presente documento técnico constituye parte del desarrollo e implementación del Sistema GeoParking. Lo que se podrá observar es la distribución de los principales componentes utilizados y los cuales componen el sistema GeoParking.

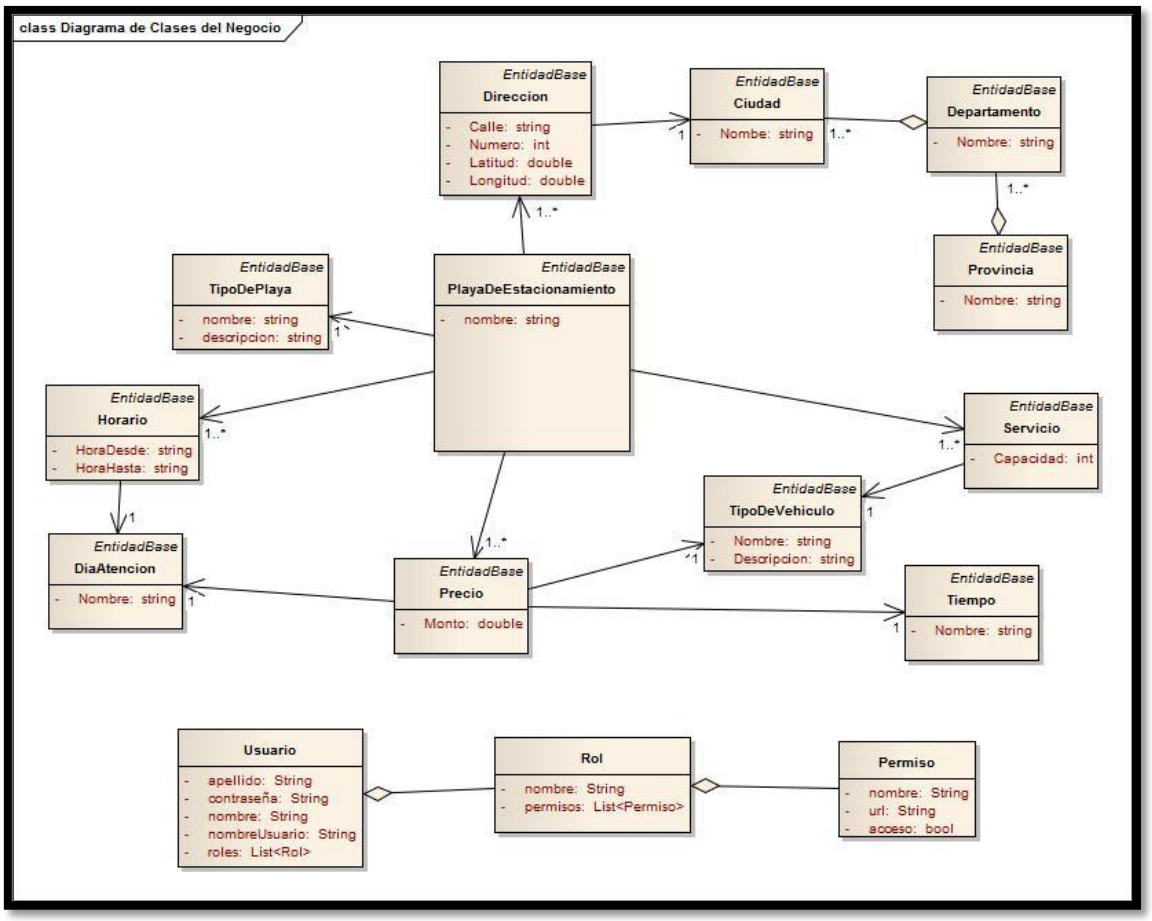
Describe el modelamiento del sistema en entorno Web como así también el entorno Móvil, como parte complementaria de la Arquitectura del Sistema. El modelamiento incluye tres perspectivas para describir el sistema, las cuales son:

- El Modelo de Negocio
- El Modelo de Arquitectura
- El Modelo de Comportamiento

En síntesis el presente documento, comprende a estructura del sistema GeoParking, cuya descripción y diseño considera el ingreso de datos e información, la manera en que estos estarán almacenados, la forma en que se procesaran y analizaran , y la forma en la información será presentada al usuario final

Las imágenes a continuación son diagramas de modelos que tratan de esquematizar la forma en que se estructura y se comporta el sistema GeoParking.

MODELO DE NEGOCIO



Como se puede observar el modelo negocio esta apuntado a abarcar todos aquellos elementos y características que se destacan en el negocio al que apunta el sistema GeoParking.

El elemento principal de este negocio es la “Playa de Estacionamiento” y de ella es que derivan los demás conformando sus características principales. El dominio del negocio que se plantea en el modelo es el del estacionamiento de vehículos particulares en playas de estacionamiento. Es hacia ese dominio, que trata de intervenir GeoParking para dar solución a problemas y mejoras en el manejo de información.

Y como todo sistema el modelo de negocio también abarca el usuario, el cual es quien interactúa con esos elementos y puede generar u obtener información a partir de los mismos.

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.

EL objetivo de estos diagramas es brindar una trazabilidad a nivel de código que permita a simple vista observar, captar y analizar el impacto el cambio de alguno de los componentes intervinientes en cada funcionalidad, permitiéndole todo esto, obtener una visión más precisa del impacto de los cambios a realizar.

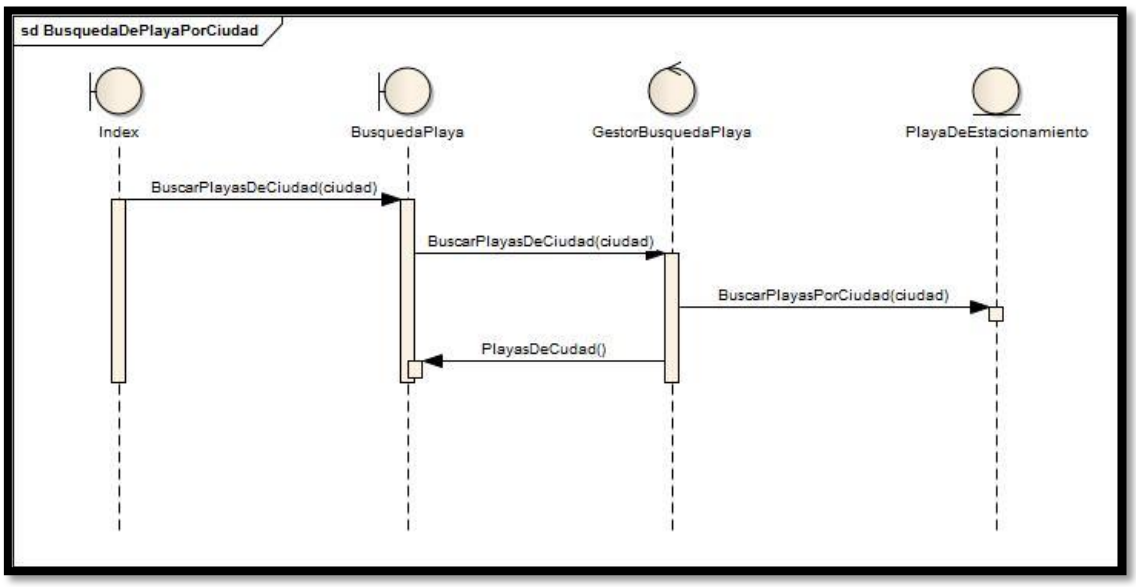
Búsqueda Playas por Ciudad – Web

Aquí el usuario web introduce una ciudad y se realiza una búsqueda de las playas de estacionamiento que existan en esa ciudad.

La petición se realiza en la página “Index” (página de inicio del sistema). Esta última envía a la página encargada de las búsquedas, la ciudad que ingreso el usuario, y la página “BusquedaPlaya” es la que se encarga de comenzar la búsqueda.

Se conecta con el gestor encargado de manejar las búsquedas de playas de estacionamiento y le envía la ciudad donde quiere recuperar las playas.

Como último paso el gestor retorna las playas resultantes de la búsqueda hacia la página de visualización para el usuario.

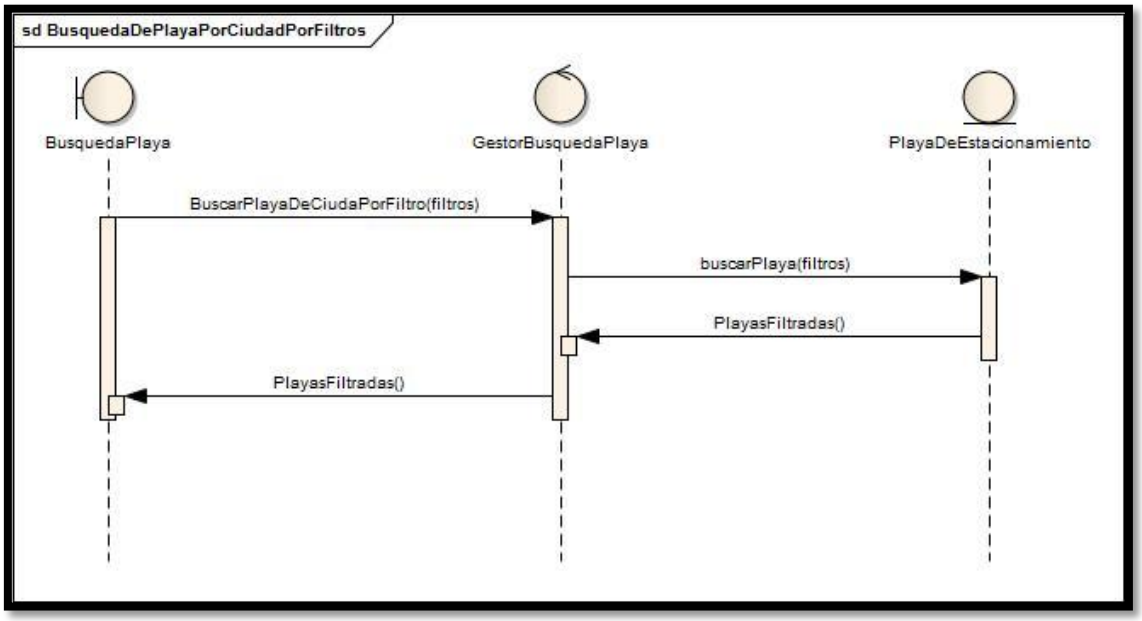


Búsqueda de Playas por Ciudad por Filtro – Web

Aquí el usuario web selecciona una serie de filtros y se realiza una búsqueda de las playas de estacionamiento que existan con esas características.

La petición se realiza en la página “BusquedaPlaya”. Esta última se conecta con el gestor encargado de manejar las búsquedas de playas de estacionamiento y le envía los parámetros (filtros) para realizar la búsqueda de las playas que concuerden con ese criterio.

Como último paso el gestor retorna las playas resultantes de la búsqueda hacia la página de visualización para el usuario.

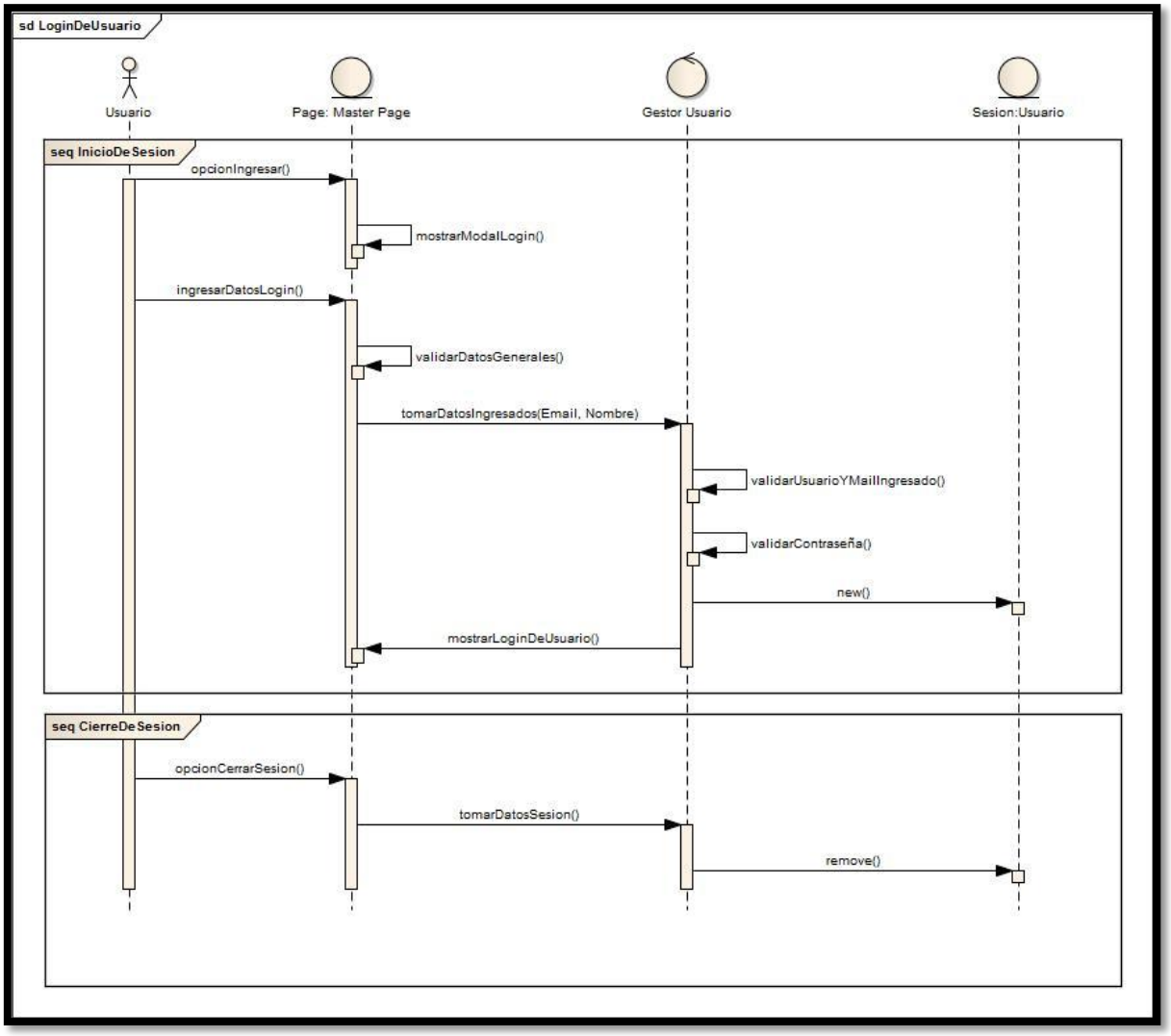


Login de Usuario – Web

Inicio de Sesión: Aquí el usuario web selecciona la opción de ingresar al sistema, y se le muestra un ventana donde deberá ingresar usuario y contraseña correspondientes. Una vez enviada la petición de logeo el sistema primero validara que los datos requeridos (usuario y contraseña) estén completos y con el formato correcto.

Una vez validados en la página, se envían dichos datos al servidor para que sean validados y se inicie la nueva sesión del usuario si es que los datos aportados fueron validados correctamente.

Cierre de Sesión: Aquí el usuario web selecciona la opción cerrar sesión y el sistema toma los datos de la sesión actual y elimina del sistema la misma.

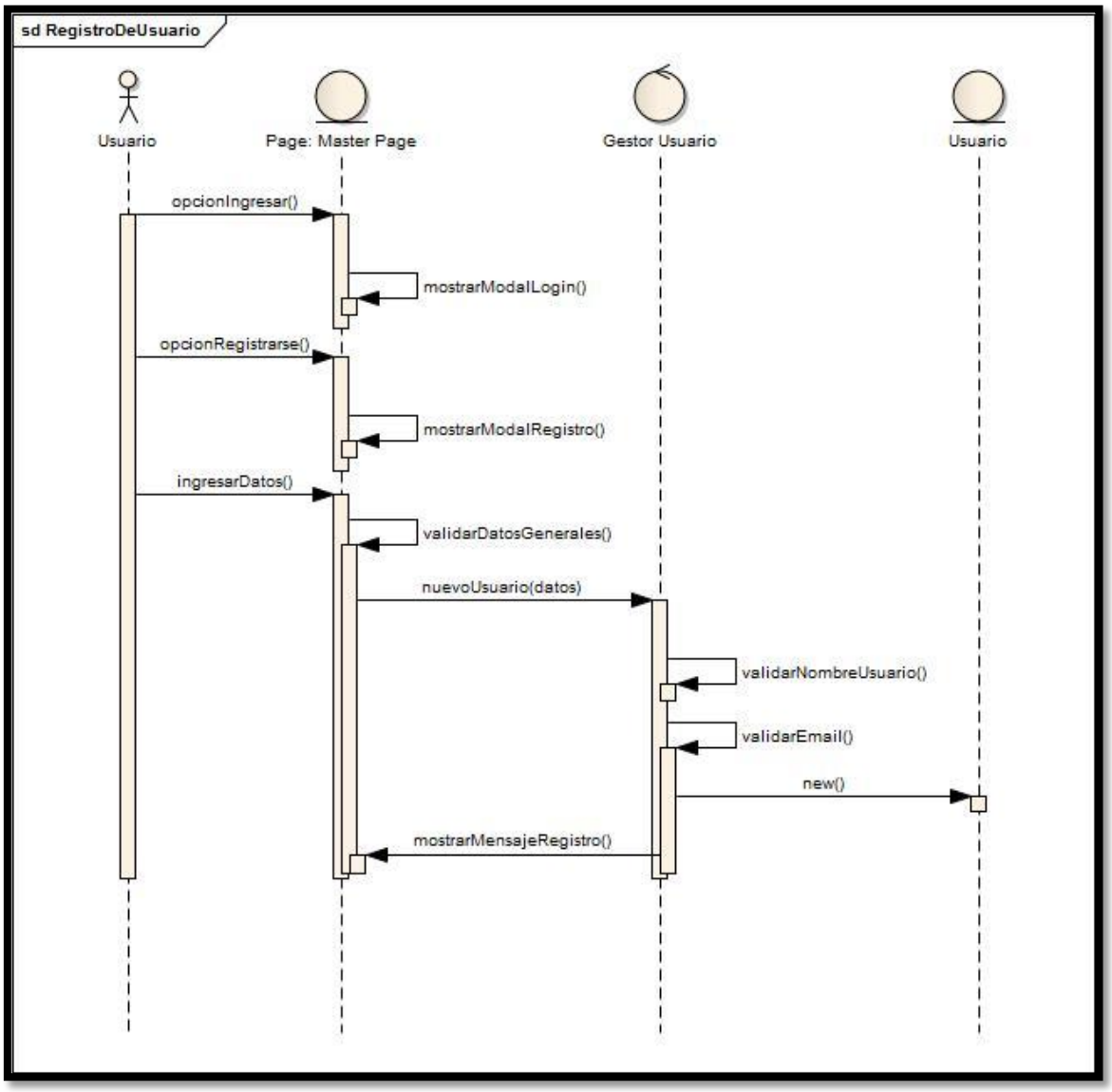


Registro de Usuario - Web

Aquí el usuario desea ingresar al sistema, y como no posee datos de acceso, selecciona la opción de registrarse. En ese momento se le muestra una ventana donde deberá ingresar los datos para la registración. Se realiza una validación de que los datos solicitados estén completos y en el formato correcto.

Una vez validados, se envía los datos al servidor donde se realiza una nueva validación, en este caso ya de consistencia con la información ya almacenada en el sistema. Un ejemplo de esos controles es de usuario existente en el sistema.

Con el control realizado y validado, se procede la creación del nuevo usuario en el sistema y se emite un mensaje de registración exitosa.

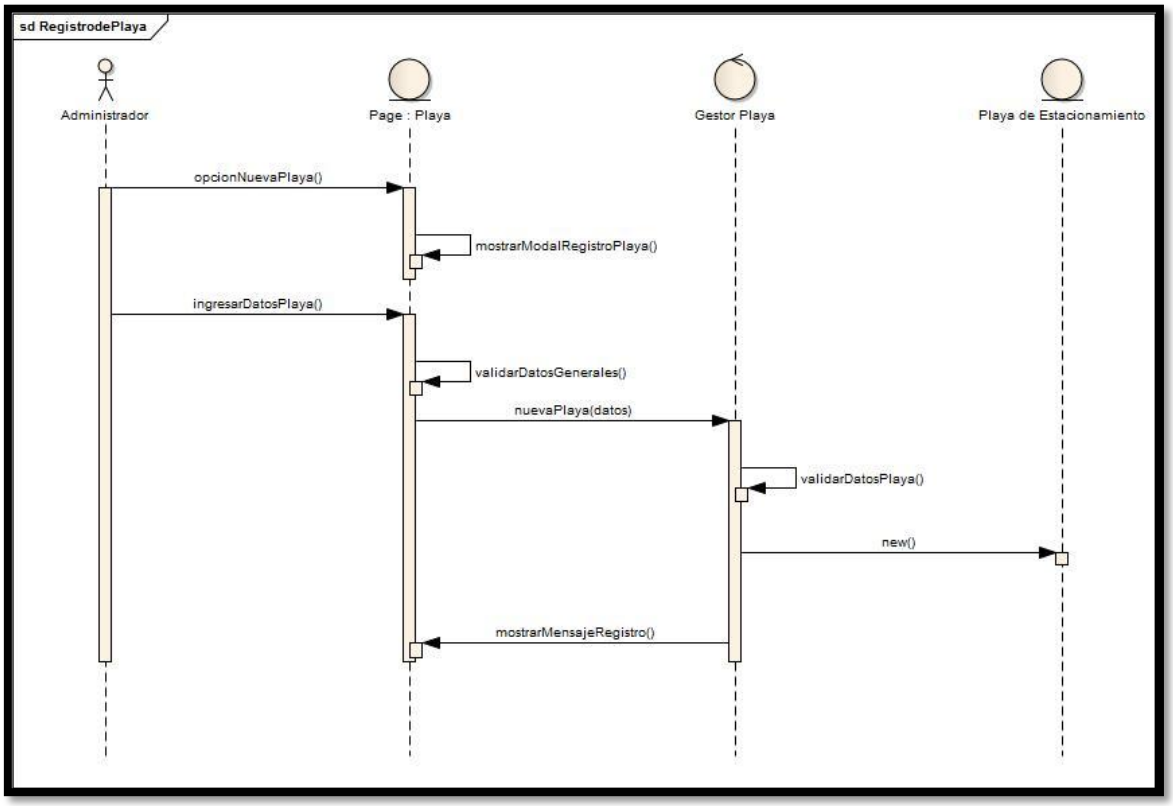


Registro de Playa – Web

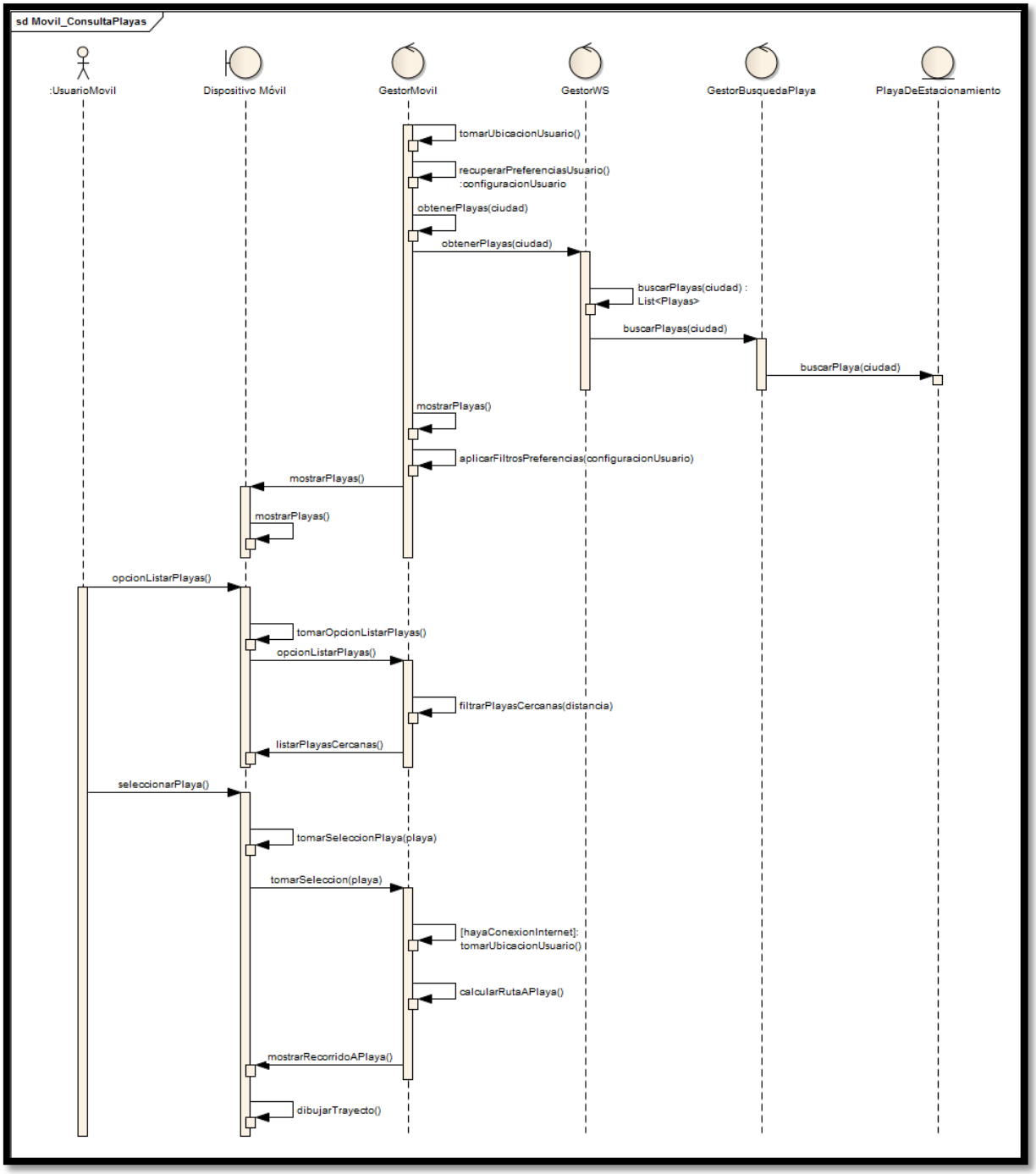
Aquí el usuario desea ingresar una nueva playa al sistema y por ello selecciona la opción "Nueva Playa". En ese momento se le mostrara una ventana donde se le solicitan todos los datos necesarios de la playa de estacionamiento para su registración.

Antes de enviar los datos al servidor, se realiza una validación en el cliente donde se evalúa que los datos estén completos, en el formato correcto y consistente entre ellos.

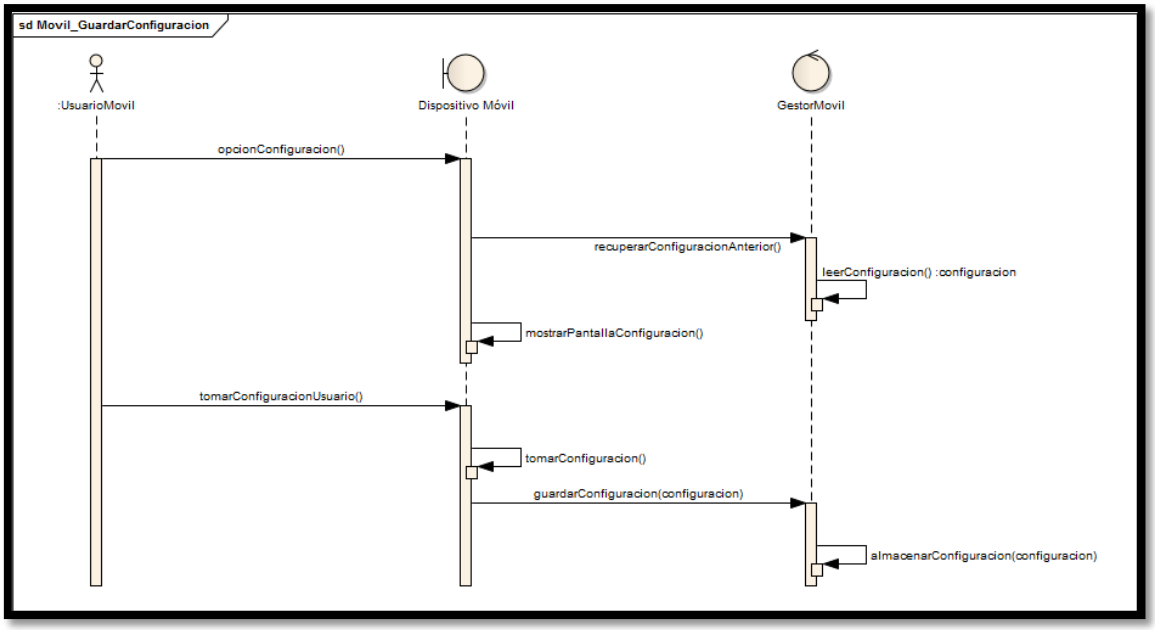
Una vez validados en el cliente se envían los datos al servidor para un nuevo control de los mismos a nivel de sistema y si estos son correctos, se registra la playa de estacionamiento y se muestra el mensaje de registración exitosa.



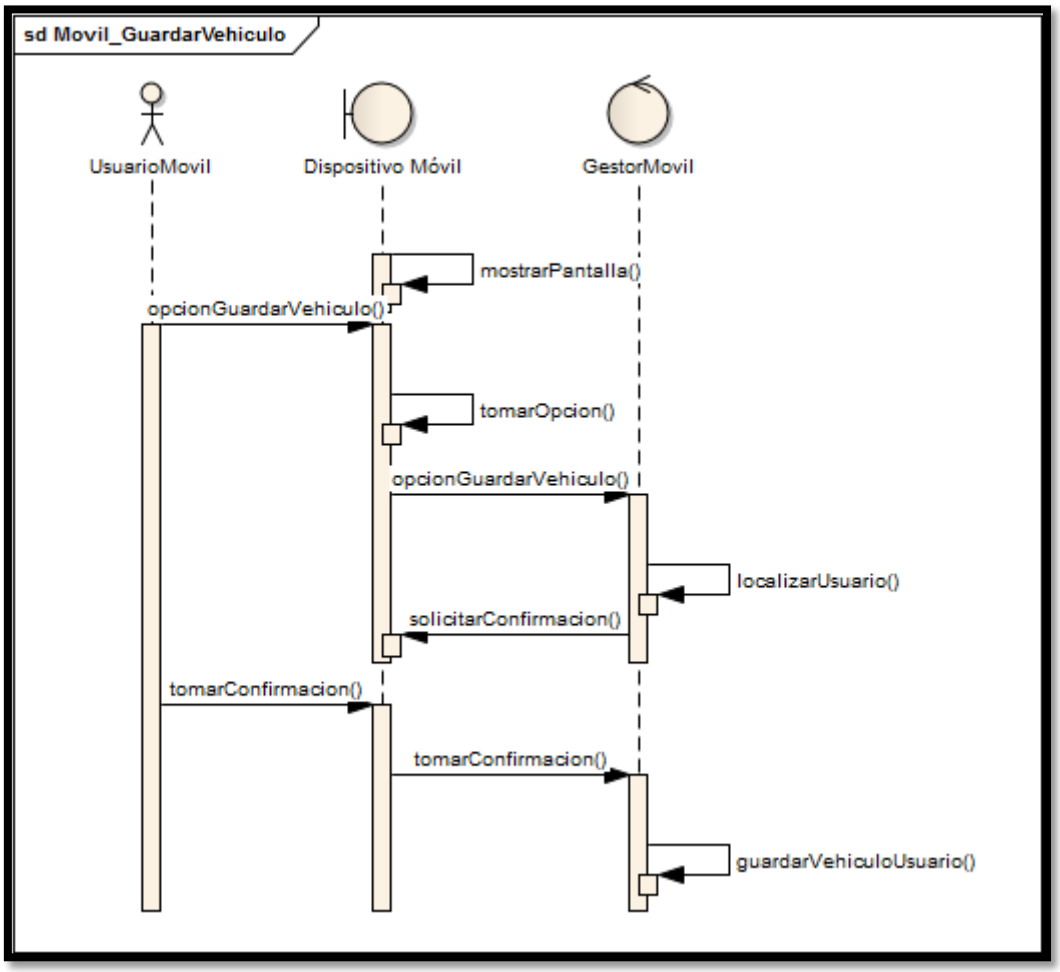
Consulta Playas – Móvil



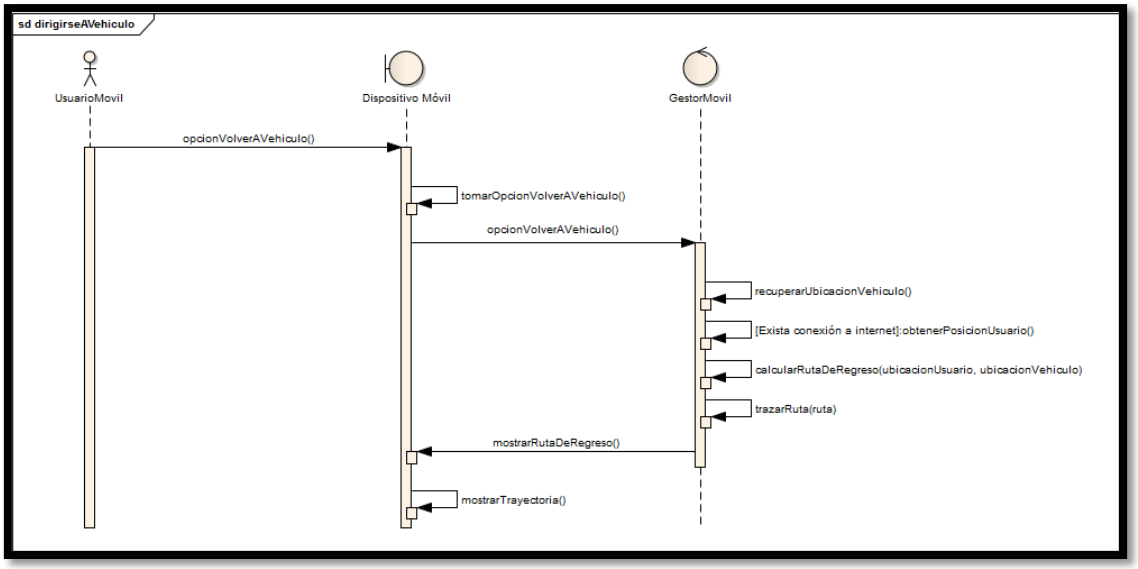
Guardar Configuración – Móvil



Guardar Vehículo – Móvil



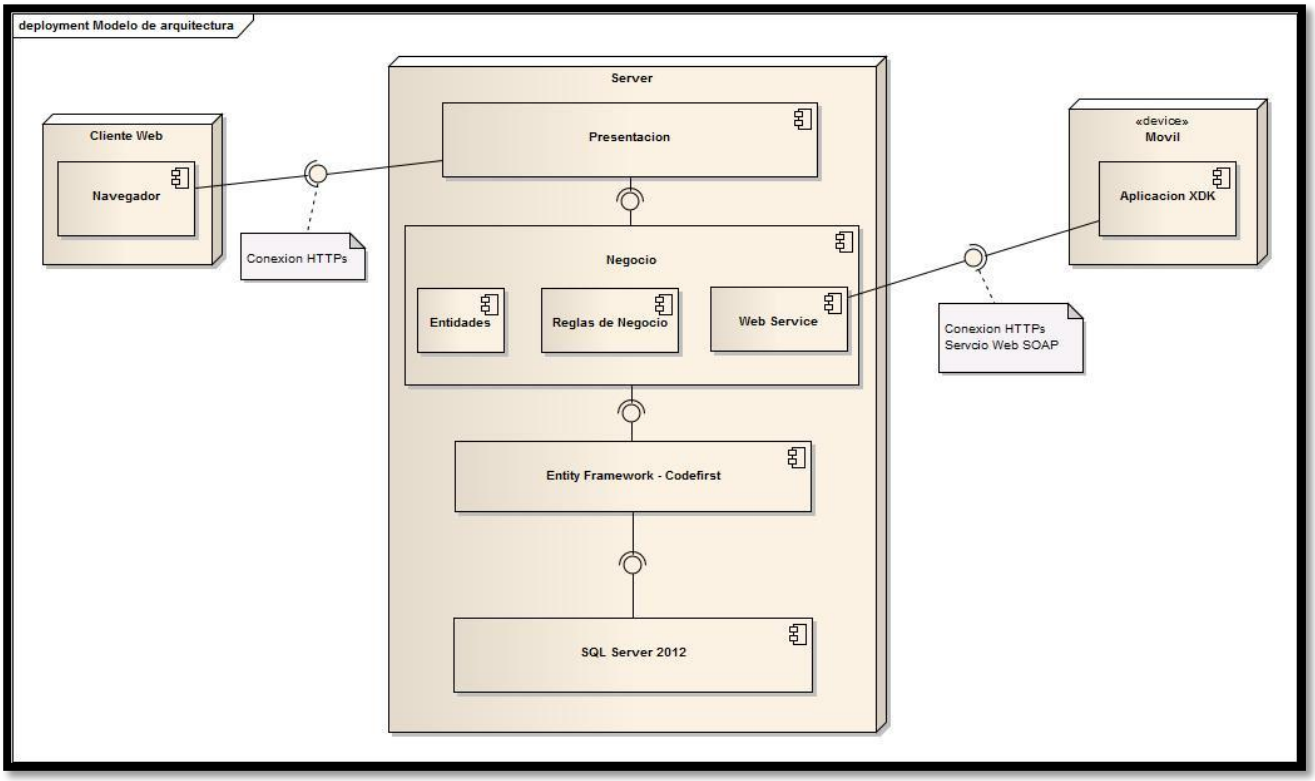
Dirigirse a Vehículo – Móvil



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.

Servidor



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.

SEGURIDAD

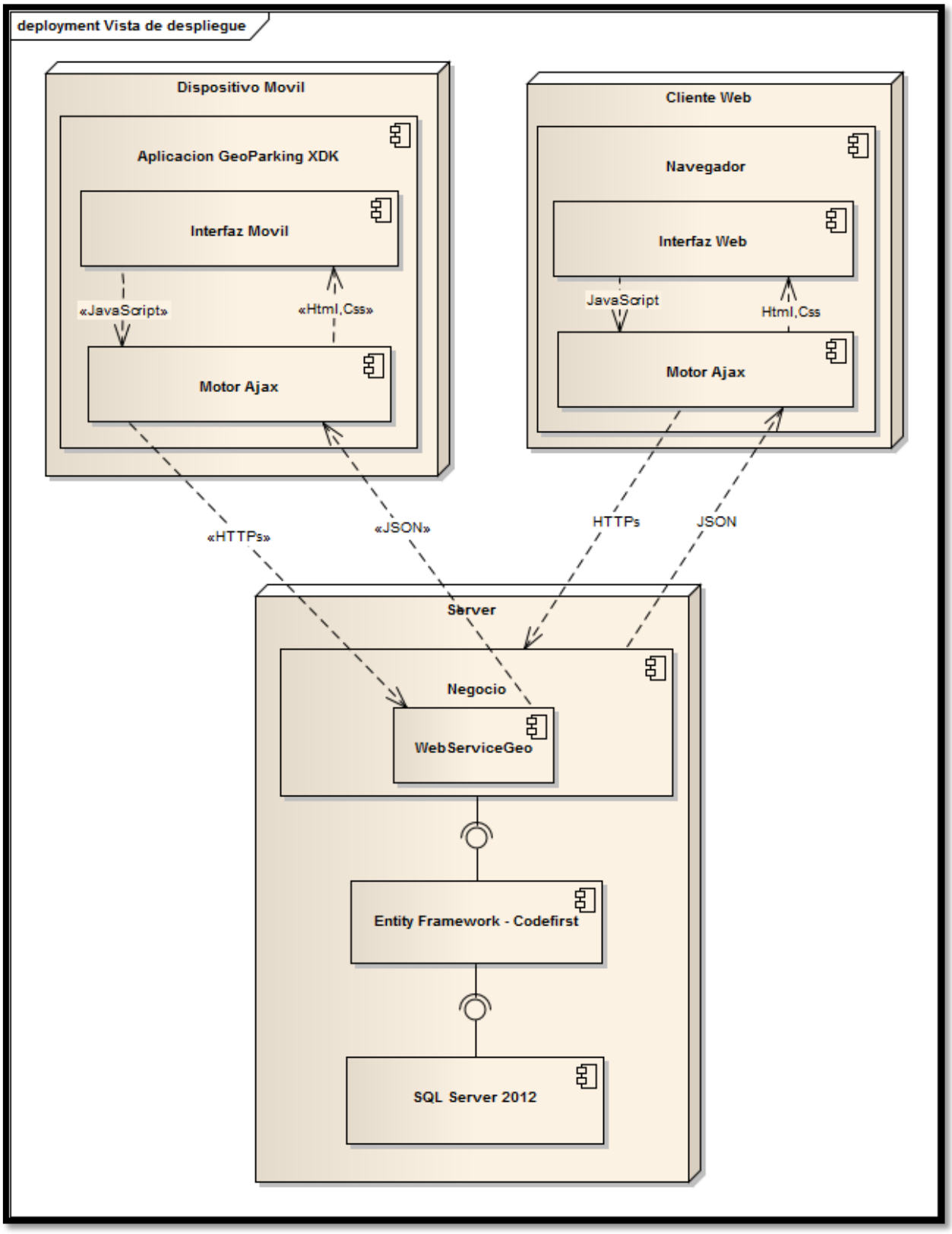
El manejo de la seguridad de la aplicación en cuanto al acceso de la información que el sistema obtiene, procesa y muestra, está basado en la autenticación de quien se comunica con el sistema ya sea web o desde la aplicación móvil.

En cuanto a la aplicación móvil la seguridad de acceso a la información esta manejada por el manejo de roles y permisos para los diferentes usuarios que interactúan con el sistema. Cada funcionalidad está restringida a roles específicos.

En la aplicación móvil, la obtención de la información que se muestra en la aplicación es obtenida a través de los servicios del componente WebService del sistema. Todas las peticiones que realiza nuestra aplicación móvil envían un clave al servicio web para que valide que es nuestra aplicación la que está realizando esas peticiones y no otra ajena al sistema.

MODELO DE ARQUTECTURA AJAX

Aquí se trata de mostrar de qué manera se hace uso de la tecnología Ajax en la implementación de nuestro proyecto GeoParking. En el diagrama se puede visualizar los componentes intervinientes en el flujo de datos.

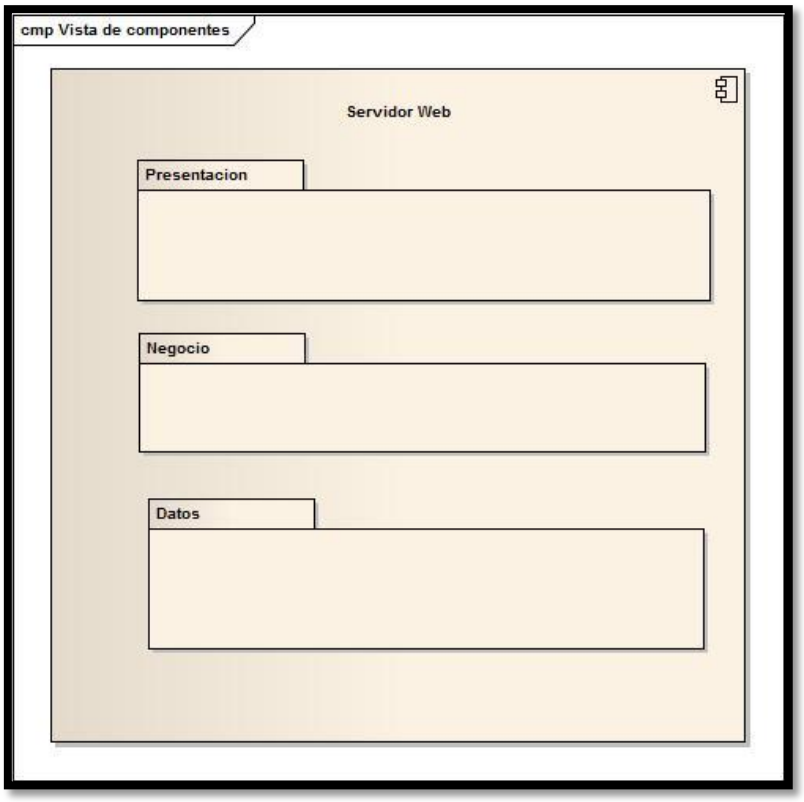


MODELO DE COMPONENTES

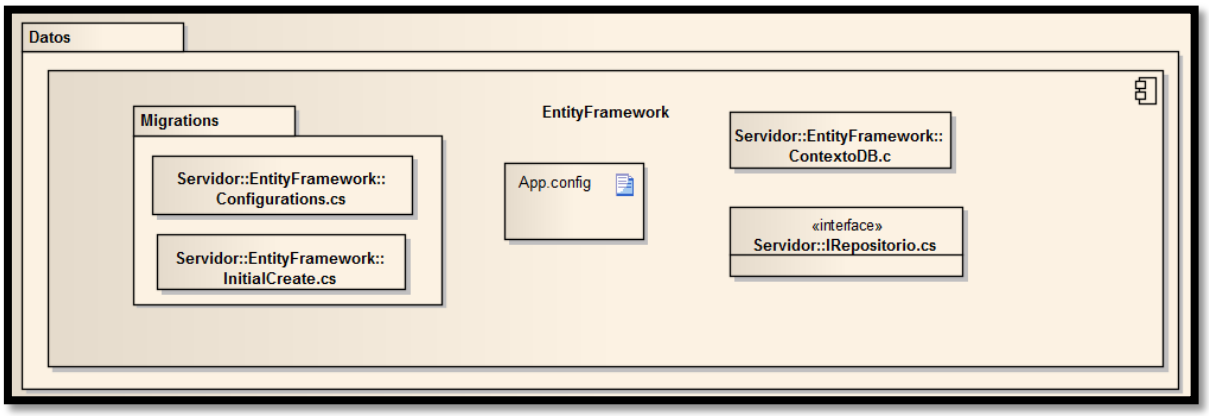
Vista de Componentes

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

1. **Datos**: capa dedicada a la materialización y desmaterialización de los datos de la aplicación, es lo que normalmente se denomina mapeo objeto relacional.
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 a su vez 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:

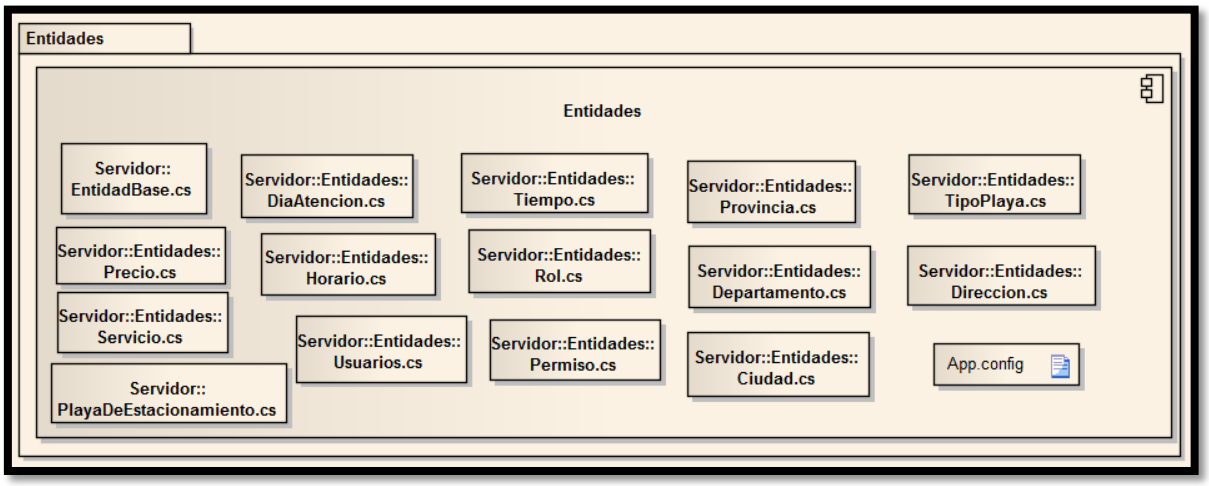
1. Entidades
2. ReglasDeNegocio
3. WebService

Entidades: alberga todas las clases que representan los objetos manejados en el modelo de negocio de la aplicación.

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

WebService: alberga la aplicación que se corresponde al webService, el cual permite ofrecer la funcionalidad del sistema a la aplicación móvil.

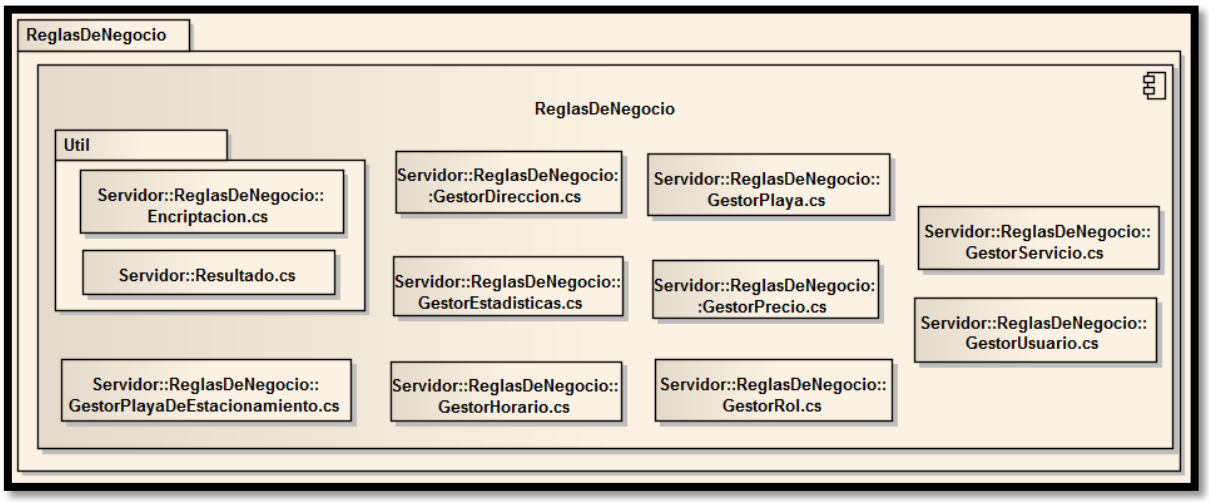
Vista de Entidades



Como se aprecia este componente alberga todas las entidades que dan soporte al modelo de negocio sobre el que trabaja el sistema GeoParking. Contiene los siguientes ítems:

- **Clase Base**: la clase EntidadBase.cs es la clase madre de todas las demás clases del negocio, de ella heredan las clases hijas.
- **Clases Hijas**: son todas aquellas clases que heredan de EntidadBase.css y ellas son:
  - **PlayaDeEstacionamiento.cs**
  - **TipoPlaya.cs**
  - **Serivicio.cs**
  - **Precio.cs**
  - **DiaAtencion.cs**
  - **Tiempo.cs**
  - **Horario.cs**
  - **Direccion.cs**
  - **Provincia.cs**
  - **Departamento.cs**
  - **Ciudad.cs**
  - **Usuario.cs**
  - **Rol.cs**
  - **Permiso.cs**
- **App.config:** archivo de configuración del componente.

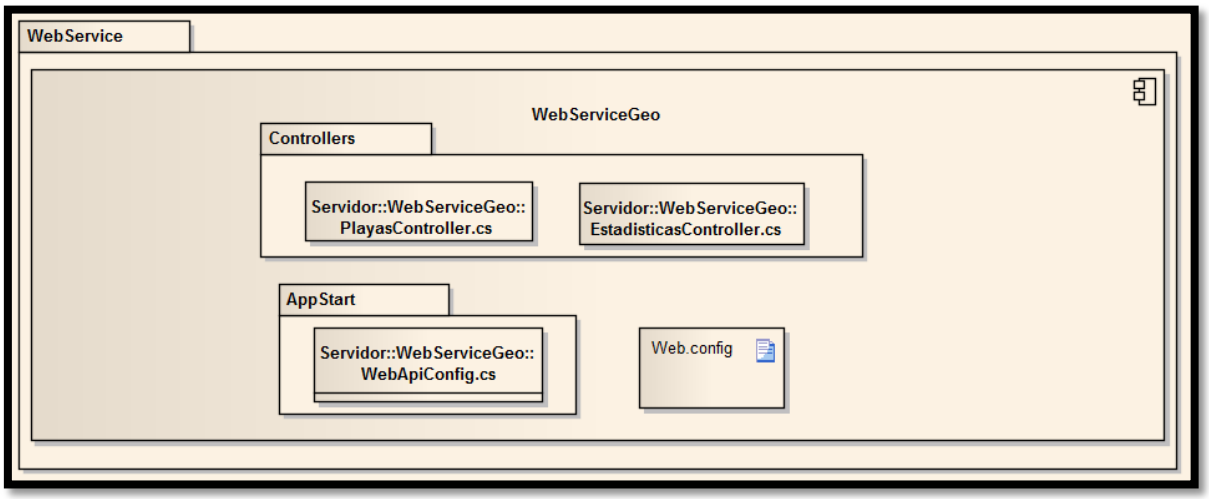
Vista de ReglasdeNegocio



Esta vista lo que muestra es como están soportadas las reglas de negocio del sistema (comportamiento), a través de componentes de software. En el componente principal que hace referencia al módulo de negocio se puede distinguir dos tipos de componentes importantes:

- **Paquete Útil**: aquí se albergan dos clases importantes en el manejo del sistema. Una de ellas es la clase Encriptación.cs, la cual maneja la seguridad de la información del usuario con su sesión activa. Y la otra clase es Resultado.cs, la cual permite retornar de manera más estructurada mensajes alusivos a los resultados del procesamiento que la información sufre en esta capa. Toda acción que se realice en esta capa y que deba retornar un resultado de la misma se hace a través de esta clase.
- **Gestores**: son las clases encargadas del procesamiento de las peticiones que se realizan al sistema. Cada gestor está encargado de una funcionalidad específica del sistema.

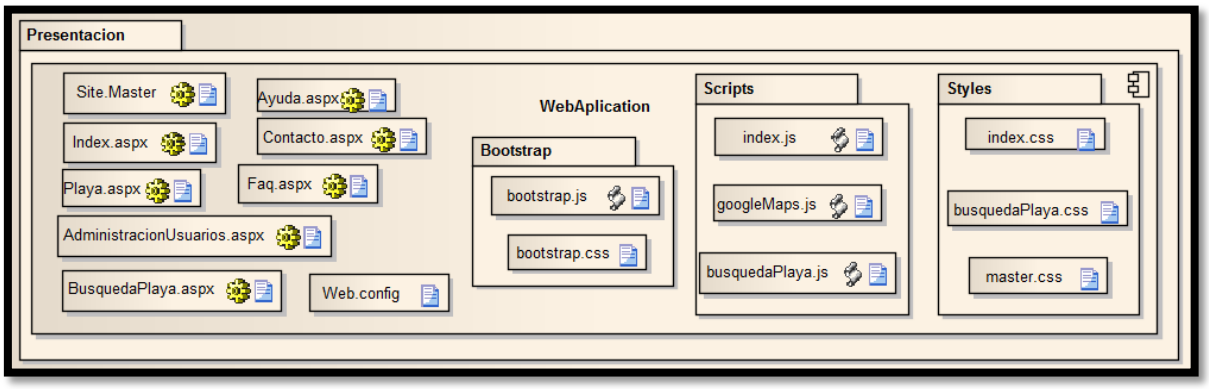
Vista de WebService



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.
- **Paquete Controllers:** este alberga todos los controladores que realizan procesamientos de información por cada una de las consultas que le llegan al web Service. Los controladores definidos en el web Service son:
    - **PlayasController.cs**: controlador que brinda la funcionalidad de manejar todas aquellas peticiones relacionadas al manejo de información de las playas de estacionamiento del sistema. Se encarga de buscar playas y retornarlas a la aplicación que las requiera, de acuerdo a los criterios recibidos en la petición.
    - **EstadisticasController.cs**: controlador que brinda la funcionalidad de manejar todas aquellas peticiones relacionadas al manejo de información estadística del sistema relacionada a la aplicación Móvil. Se encarga de tomar todos los datos que la aplicación móvil pueda brindarnos para luego realizar las estadísticas e informes correspondientes.

Vista de Presentación



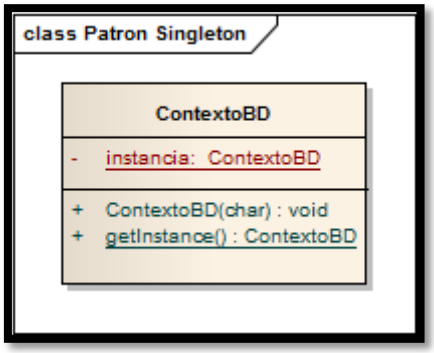
Aquí se puede observar como esta capa del sistema está conformada por una aplicación web donde se pueden observar componentes como:

- **Site.master:** platilla modelo de interfaz de donde heredan las demás páginas.
- **Paginas.aspx**: páginas que heredan de la master. Las paginas desarrolladas hasta el momento son:
    - **Index**.aspx
    - **Ayuda**.aspx
    - **Contacto**.aspx
    - **Playa**.aspx
    - **AdministracionUsuarios**.aspx
    - **BusquedaPlaya**.aspx
    - **Faq**.aspx
- **Web.config**: configuración de distintos aspectos de la aplicación web.
- **Paquete Bootstrap**: contiene todos los archivos JavaScript y Css (estilos) del framework Bootstrap (librería de terceros) para el maquetado de interfaces web.
- **Paquete Scripts**: contiene todos los archivos JavaScript de creación propia aplicados a la interactividad del usuario con componentes de la interfaz web del sistema.
- **Paquete Styles**: contiene todos los archivos de estilos (.Css) de creación propia, que actúan en el diseño de la interfaz visual del sistema, modificando su apariencia.

MODELO DE PATRONES

Aquí se muestra el diagrama de implementación de los patrones utilizados en el desarrollo del producto GeoParking.

Singleton

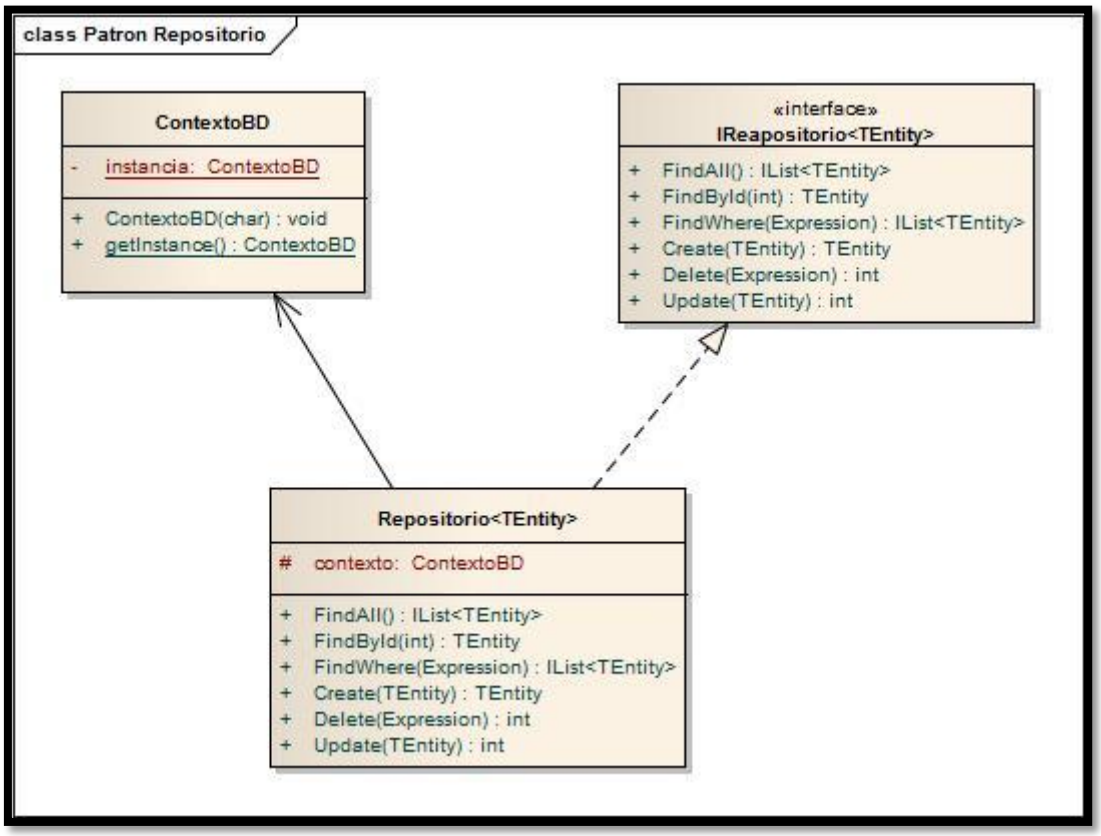


JUSTIFICACION

La aplicación del patrón Singleton sobre la clase ContextoBD surge por:

- Necesidad de que exista una instancia única en todo el sistema; esto es porque a la hora de instanciarse un objeto del tipo ContextoBD requiere de un tiempo para levantar toda la estructura para relacionarse con la base de datos, y poder mapear todos los objetos. Si se instanciara cada vez que se lo requiere la performance de la aplicación decaería precipitosamente.
- Acceso desde todas las clases DAO: por cada clase entidad, necesitaremos una clase específica que realice todas las tareas relacionadas a la base de datos a través de las funcionalidades que le preste el contexto.

Repositorio



JUSTIFICACION

- En pocas palabras, un repositorio es un mediador entre el dominio de la aplicación y los datos que le dan persistencia. Con este planteamiento podemos pensar que el usuario de este repositorio no necesitaría conocer la tecnología utilizada para acceder a los datos, sino que le bastaría con saber las operaciones que nos facilita este "mediador", el repositorio.
- Utilizando como tecnología de acceso a datos el ORM Entity Framework, con una aproximación "code-first", se podrá utilizar el repositorio para cualquier entidad compleja que se defina.