

Universidad Tecnológica Nacional

Facultad Regional Córdoba

Ingeniería en Sistemas de Información

Proyecto Final

Documentación del Producto

5° AÑO 5K2

Docentes: Zohil, Julio Cesar Nelson (Adjunto)
Liberatori, Marcelo Sadi (JTP)
Jaime, María Natalia (JTP)

Alumnos: Bär Coch, Edgardo Ezequiel
Barrera, Marcos
Frigerio, Ignacio
Romero, Leonel
Toneatto, Lucas Fernando

Contenido

Estudio Inicial	5
Introducción	5
Ámbito de aplicación.....	6
Encuestas y estadísticas	6
Definición del producto.....	7
Objetivo del sistema.....	7
Alcances del sistema	7
Requerimientos técnicos de hardware y software	8
Requerimientos mínimos de Hardware	8
Requerimientos mínimos de Software.....	8
Análisis de prefactibilidad	9
Desarrollo	9
Implementación	10
Identificación de riesgos	11
Beneficios y ventajas que se procuran en su implementación	11
Antecedentes de sistemas similares:	11
Metodología de trabajo y desarrollo a aplicar:.....	12
Integrantes del equipo de trabajo:	12
Proyectos de capacitación complementarios	13
Geolocalización y Navegación GPS.....	13
Utilización API Google Maps	13
Programación Móvil - Android	14
Estadísticas de entrevistas	14
Currícula del equipo de trabajo.....	15
Lucas Fernando Toneatto.....	15
Marcos Barrera.....	16
Leonel Romero	18
Ignacio Frigerio	18
Ezequiel Bär Coch	20
Modelado del sistema	22
Control de la documentación.....	22
Control de la Configuración.....	22
Histórico de Versiones.....	22
Acrónimos	23
Introducción	24

Modelo de negocio	25
Modelo de diseño	26
Diseño aplicación WEB	26
Diseño aplicación MÓVIL.....	30
Modelo de interacción	32
Aplicación web	32
Aplicación Móvil	36
Mostrar listado de playas con disponibilidad - Móvil	41
Modelo de arquitectura y despliegue	42
Servidor	42
• Presentación.....	42
• Negocio.....	42
• Entity Framework	43
• SQL Server	43
Cliente Web	43
Cliente Móvil	43
Seguridad.....	43
Conexión con API GeoParking	44
Modelo de arquitectura AJAX (Asynchronous JavaScript And XML).....	45
Modelo de componentes	46
Vista de Componentes	46
Vista de Datos.....	47
Vista de Negocio.....	47
Vista de Entidades	48
Vista de Reglas de Negocio	49
Vista de WebService.....	49
Vista de Presentación.....	50
Modelo de patrones.....	51
Singleton.....	51
Informe de ejecución de pruebas	53
Control de la documentación.....	53
Control de la Configuración.....	53
Histórico de Versiones.....	53
Introducción	54
Tipo de Casos de Prueba	55
Casos de prueba exploratorios.....	55

Casos de prueba manuales	55
Casos de prueba automáticos	55
Tipos de ciclos de Ejecución	55
Ciclo de Sistema	55
Ciclo de Regresión	55
Ejecuciones de casos de prueba.....	55
Historia 3: Administrar Playa de Estacionamiento.....	56
Historia 2: Consultar playas de estacionamiento en la web	59
Historia 7: Consultar disponibilidad de playa de estacionamiento.....	60
Historia 13: Consultar disponibilidad de playas de estacionamiento en tiempo real	60
Historia 25: Visualizar estadísticas históricas.....	61
Conclusión	62
Introducción	63
Matriz de trazabilidad	64
Control de la documentación.....	64
Control de la Configuración.....	64
Histórico de Versiones.....	64
MATRIZ DE TRAZABILIDAD	65

Estudio Inicial

Introducción

Con la realización de este proyecto vemos la oportunidad de concretar lo aprendido en nuestro trayecto universitario, integrando los conocimientos aprendidos de las materias cursadas, plasmando el resultado en un producto de software, el cual esperamos que sea beneficioso para los usuarios.

Desde un punto de vista académico, nuestro objetivo es culminar esta importante etapa de nuestras vidas, logrando obtener el título de Ingeniero en Sistemas de Información en el tiempo estimado, siendo la finalización del mismo a principios del mes de Marzo de 2015. Este logro nos brindará la posibilidad de crecer económica y profesionalmente, logrando mejores puestos de trabajo, o bien iniciando un emprendimiento propio.

Para lograr nuestro objetivo, deberemos trabajar como un equipo unido, resolviendo los problemas que se presenten, y mejorar aspectos personales para generar un ambiente de trabajo adecuado.

En el siguiente estudio inicial, trataremos de analizar los puntos más importantes a tener en cuenta para la realización del proyecto. En una primera parte se explicará el ámbito y contexto en que surge dicho proyecto, en el cual se desarrollara un producto y para ello se identificará el mercado y sector al cual pertenece. Además se hará una comparación con productos similares buscando distinguir y perfeccionar la definición del nuestro.

Luego haremos un detallado análisis de prefactibilidad en carácter técnico, operativo y económico tanto del desarrollo como la implementación del producto con el fin de conocer y cerciorarnos de los recursos que necesitaremos para alcanzar el objetivo propuesto.

En una segunda parte del estudio, describiremos en profundidad nuestro producto a desarrollar. Estableceremos de manera clara y concisa su objetivo y alcances para luego describir la metodología elegida para llevarlo a cabo. Se establecerán los tiempos estimados para su realización y por último describiremos la conformación del equipo de trabajo y detallaremos los roles de cada integrante.

Ámbito de aplicación

Una problemática que nos afecta desde hace algunos años y que se encuentra en continuo crecimiento es el tránsito en las grandes ciudades, que tiene su concepción en la creciente industria automotriz. Como consecuencia de la misma, observamos una gran dificultad por parte de los conductores para encontrar un lugar seguro donde dejar su vehículo mientras realiza sus actividades cotidianas o se encuentra haciendo turismo.

La oportunidad que identificamos observando esta problemática es la realización de una aplicación web y móvil que le brinde a los conductores la mejor forma de llegar a un espacio con ciertas características deseadas, asegurando disponibilidad de plaza y permitiéndole visualizar la ubicación donde dejó su vehículo para hacer más simple la tarea de encontrarlo.

Los principales usuarios del sistema, serán los conductores de vehículos particulares. Quienes al momento de querer estacionar realizarán una consulta a través de su dispositivo móvil, la cual les permitirá saber dónde tienen estacionamiento libre cerca de su destino final, con las características de su agrado y como llegar hasta la misma. Para lograrlo, las playas de estacionamiento, deberán registrarse en un módulo web, completando sus datos y características que deseen proveer.

Con la información recopilada, se podrá disponer de informes con datos como ubicación más requerida para estacionar, puntos críticos de la ciudad en materia de estacionamiento, horarios críticos. Estos informes le serán de utilidad a entidades como inversionistas de proyectos de estacionamientos y/o áreas del gobierno municipal como Tránsito.

A continuación se describen los posibles usuarios del sistema:

- Conductores de Automóviles: es aquella persona que conduce su vehículo particular en la ciudad y que tiene como objetivo encontrar un lugar de estacionamiento.
- Playas de Estacionamiento: es aquella entidad que tiene la infraestructura adecuada para brindar un servicio de estacionamiento, con lo cual cumple su objetivo de poder albergar vehículos por un tiempo determinado a cambio de un ingreso monetario. Utilizará el sistema para actualizar la disponibilidad de lugares.
- Entidad Reguladora de Tránsito: es aquella que tiene como actividad obtener información del flujo vial y establecer la disposición de los lugares de estacionamientos en la ciudad. Su objetivo es lograr obtener información útil para que los lugares de estacionamiento tengan una disposición estratégica lo cual permite satisfacer la necesidad de estacionamiento que tienen los conductores en la ciudad.

Encuestas y estadísticas

Más allá del análisis realizado sobre la problemática, se realizaron encuestas a las playas de estacionamiento para obtener datos más exactos sobre la viabilidad del producto que queremos desarrollar.

Las encuestas se hicieron entre los días 22 y 24 de Abril sobre un muestreo de 10 playas de estacionamiento, ubicadas en las zonas de Nueva Córdoba y Centro de la ciudad de Córdoba. Los tipos de las playas de estacionamientos que fueron abarcadas pertenecían a diferentes categorías.

Como resultados de las mismas, se obtuvo la información necesaria para poder realizar parte del análisis de factibilidad del sistema. Dichos resultados se ven con mayor precisión en el anexo.

Definición del producto

Objetivo del sistema

“Brindar información sobre disponibilidad y ubicación de cocheras cercanas al destino del usuario, así como de información relativa a las necesidades de estacionamiento de los ciudadanos, para realizar reportes, informes y estadísticas sobre el flujo del tránsito dentro de la ciudad”.

Alcances del sistema

Gestión de Playas de estacionamiento: Gestionar las playas de estacionamiento registradas en la ciudad. Cada playa deberá quedar registrada en el sistema brindando información sobre su ubicación y la disponibilidad de lugares con los que cuenta.

Gestión de Ubicación y Disponibilidad de Playas: Obtener la ubicación del vehículo, que permita detectar dentro de un radio determinado, las playas de estacionamiento con ubicaciones disponibles. El vehículo informará su punto de ubicación exacta, el cual permitirá realizar una búsqueda de las playas de estacionamientos cercanas dentro de un radio determinado, y mostrar por cada una de ellas su nivel de disponibilidad. Toda esta información será actualizada en tiempo real para dar una información más exacta al conductor del vehículo. Al seleccionar una playa como destino para estacionar, se le mostrará a modo de GPS el recorrido para llegar a ese destino.

Gestión de Ubicación de Vehículos: Registrar y conocer la ubicación de un vehículo estacionado. Al momento de estacionar el conductor registra la posición del vehículo, dándole la posibilidad de regresar al mismo en cualquier momento. Dicha búsqueda indicará a modo de GPS como llegar a la ubicación donde quedó estacionado el vehículo.

Gestión de informes de tránsito: Generar y emitir informes relativos al tránsito de la ciudad. Los reportes que el sistema generará darán distintos tipos de información útiles para los procesos de análisis de la organización urbana en cuanto al tránsito. Estos informes incluirán:

- Zonas donde la necesidad de estacionamiento no esté satisfecha.
- Zonas donde la necesidad de estacionamiento esté satisfecha y se disponga de disponibilidad para la recepción de más vehículos.
- Horas pico de solicitudes de estacionamiento, clasificados por zonas
- Tiempo de recorrido entre la solicitud y la registración del estacionamiento
- Cocheras más seleccionadas por zona
- Visualización de la ocupación de las plazas de estacionamiento en su totalidad
- Puntos de interés más solicitados
- Visualización en tiempo de real de la disponibilidad de playas de estacionamiento.

Gestión de puntos de Interés: Registrar puntos de interés de la ciudad, para obtener información sobre playas de estacionamiento cercanas al mismo. El conductor que desea ir a un punto de interés y busca donde estacionar, el sistema le permitirá visualizar previamente que playas de estacionamientos están ubicadas en los alrededores, como así también su disponibilidad y el correspondiente recorrido para llegar al mismo.

Requerimientos técnicos de hardware y software

El sistema desde el punto de vista técnico necesitara contar con la tecnología necesaria para las etapas desarrollo e implementación. Para tener una idea más completa de cuáles son estos requerimientos se ha realizado un listado de los mismos con su correspondiente descripción.

Requerimientos mínimos de Hardware

- *Servidor de Desarrollo:* el servidor deberá contar como mínimo con las siguientes características:
 - Intel® Celeron® & Pentium™ Dual
 - SATA - 80 GB a 2 TB
 - DDR3 - 1 a 2 GB
- *Servidor de Implementación:* a la hora de implementar el sistema necesitaremos que el servidor pueda contar con lo siguiente:
 - Intel® Core™ i7 & Xeon®
 - SATA / SAS / SSD - 60 GB a 2 TB
 - DDR3 / EC - 4 a 32 GB
- *Equipos de desarrollo:* los equipos involucrados de desarrollo deberán contar con:
 - RAM: 3GB
 - DISCO: 10gb de espacio libre

Requerimientos mínimos de Software

- *Sistemas Operativos:*
 - Windows 7,8: un sistema de los dos deberá estar instalado en los equipos de desarrollo y en el servidor de desarrollo e implementación.
 - Android: es el sistema que tendrán los dispositivos móviles que utilizaran el sistema.
- *Software de Desarrollo:* los programas necesarios para el desarrollo del sistema serán los siguientes:
 - Visual Studio 2012
 - Microsoft SQL Server 2012
 - Eclipse
 - IceScrum (Gestión de Scrum)

Análisis de prefactibilidad

A la hora de realizar un análisis de prefactibilidad, debemos tener en cuenta la diferenciación entre factibilidad técnica, operativa y económica. A su vez para abordar este análisis nos enfocamos en dos etapas en la realización del sistema, ellas son la etapa de desarrollo y la etapa de implementación, las cuales se tratarán a continuación.

Desarrollo

- **Factibilidad Técnica:** El análisis elaborado dio como resultado que la tecnología involucrada en el desarrollo del sistema está al alcance de nosotros, lo cual significa, que por lo menos en este aspecto no se nos deberían presentar contratiempos en la obtención de los recursos necesarios. Cuando nos referimos a “tecnología involucrada”, estamos hablando de hardware, software como el conocimiento para su utilización.

La tecnología que utilizaremos será la siguiente:

- ASP.NET: con ello será realizado la parte web del sistema.
- GOOGLE MAPS: será utilizado para el manejo de todo lo referido a geo posicionamiento.
- ANDROID: será la plataforma en que se desarrolle el sistema Móvil.
- **Factibilidad Económica:** En cuanto a este aspecto se analizaron ítems que son positivos a la hora de plantear el desarrollo del sistema, estos puntos son:
 - *Software:* refiriéndonos a los costos de utilización de software específico para la etapa de desarrollo, hemos llegado a la conclusión de que todo lo necesario para estas tareas, son accesibles de manera gratuita, lo cual es una ventaja al no tener que usar tecnología en donde se deban abonar licencias o cánones por su utilización. Tenemos un costo único que abonar de U\$S 25 para poseer la licencia de developer y subir aplicaciones a Google Play ©.
 - *Hardware:* los únicos componentes involucrados y analizados son los equipos de desarrollo y el servidor. En cuanto a los equipos para el desarrollo se cuenta con las PC de cada uno del equipo de desarrollo; y en cuanto al servidor, una posibilidad sería alquilar un Servidor que estaremos en primera etapa con un servidor que nos permita desarrollar y testear en el mismo: **<ver requerimientos técnicos>** Este servidor incurrirá en un gasto de aproximadamente \$430 al mes.
 - Personal: las personas involucradas en el desarrollo principalmente el equipo de desarrollo, seremos nosotros, lo cual no representa costos.
- **Factibilidad Operativa:** Aquí abordamos lo que sería la capacidad del equipo de desarrollo para llevar a cabo la creación del sistema. Según el análisis de nuestras capacidades, podemos llevar a cabo las tareas y sortear los inconvenientes que se nos puedan presentar. Si bien se requerirá trabajar con tecnologías que no manejamos, la capacitación será una herramienta para solucionar este requerimiento. Para ello se ha planificado distintas etapas del proyecto en donde se adquirirán los conocimientos

necesarios para el manejo dichas tecnologías. En una etapa temprana del proyecto dedicaremos 7 horas por integrante para obtener dicha capacitación. De ser necesario en otras etapas del proyecto se dedicaran más horas para dicha actividad.

Implementación

- **Factibilidad Técnica:** Este punto lanzó una cuestión importante en la implementación del sistema que está relacionado a las playas de estacionamiento. Lo que sucede es que las playas de estacionamiento no cuentan todas con el mismo sistema, lo cual es un inconveniente. Sin embargo con nuestros análisis realizados hemos descubierto que podemos solucionarlo de la siguiente manera. La mayoría de las playas de estacionamiento cuentan con sistemas pertenecientes a la misma empresa fabricante. Nuestra opción sería establecer una alianza estratégica con dicho fabricante para que implemente en sus sistemas la funcionalidad que necesitamos en nuestro sistema. Al realizar dichos cambios en su software, el mismo brindará mayor valor y oportunidades de negocio a las playas que lo usen, lo cual sería una ventaja competitiva para la empresa fabricante.
- **Factibilidad Económica:** A la hora de implementar el sistema tendremos que manejar distintos tipos de costos, tales como lo son el costo de publicar la aplicación móvil en Google Play Store como los costos del servidor que dará soporte a todo el sistema. En cuanto a este último, una de las posibilidades en la cual no se originarían gastos sería que la entidad reguladora de tránsito emplee sus instalaciones del área de sistema para albergar nuestro producto. En caso contrario, un servidor que tenga las prestaciones necesarias para un buen desenvolvimiento de la aplicación nos incurrirá en aproximadamente \$975 al mes *<ver detalles técnicos>*.
- **Factibilidad Operativa:** Prevemos que las funcionalidades del sistema asegurarán un alto grado de aceptación por parte de los usuarios. Esto ha sido comprobado realizando encuestas personales a allegados que realmente consideran el hecho de encontrar un lugar de estacionamiento como un “artilugio del conductor”, entendiéndose por eso como algo bastante complejo de conseguir hoy en día en las calles de la Ciudad de Córdoba.

También podemos asegurar que la información brindada por el sistema será de utilidad para la entidad reguladora de tránsito como así también para dueños o inversores de futuras cocheras. Debemos tener en cuenta la posibilidad de que las playas no logren actualizar la información debido a diversos factores, como puede ser falta de acceso a internet; lo cual repercutirá en información errónea que será transmitida por el sistema.

En cuanto a la usabilidad del sistema, las ideas que se piensan implementar darán al usuario una experiencia de simplicidad al usarlo. Está pensado para que el usuario pueda obtener la mejor información, el mejor servicio de la manera más rápida y útil a sus necesidades. Se tiene pensado que el sistema este accesible en plataformas web y móviles, por lo cual el alcance del sistema es aún mayor, y el usuario tendrá un abanico mayor de posibilidades a la hora de su uso.

Con este análisis pudimos concluir que se dan las condiciones necesarias para llevar a cabo el proyecto, ya que los puntos importantes del mismo fueron abarcados en el estudio y los mismos pasaron el análisis de manera satisfactoria; dándonos una base importante para la decisión de continuar con el proyecto.

Un resultado a destacar de este estudio, a partir de los distintos enfoques utilizados, es que nos permitió obtener una visión a futuro del camino a seguir para la realización del proyecto y que procesos tendríamos que atravesar en el mismo.

Identificación de riesgos

Luego de haber realizado un análisis de factibilidad para el desarrollo e implementación de nuestro sistema, hemos logrado identificar que pueden llegar a existir ciertos riesgos, los cuales podrían intervenir en el normal desarrollo del producto. Sin embargo debido a esta temprana etapa en la que nos encontramos, no podemos asumir el nivel de impacto de cada uno de ellos sin realizar un análisis profundo de riesgos; el cual será desarrollado en una etapa posterior.

Beneficios y ventajas que se procuran en su implementación

Para poder analizar este punto debemos recordar cuál es el origen de la realización de esta idea, lo cual ha quedado bastante claro en todos los puntos que hemos desarrollado. Es por ello que creemos que la implementación del sistema de estacionamiento producirá un impacto positivo en la población que utilizará el mismo, el cual solucionará gran parte de la problemática planteada. La aceptación del sistema producirá importantes beneficios a los involucrados, ayudándolos y facilitándoles el cumplimiento de sus actividades y objetivos.

Antecedentes de sistemas similares:

Identificación: *Sistema Bs As*

Objetivo: Mostrar la ubicación de playas de estacionamiento cercanas en la Ciudad de Bs As.

Límites: Desde que el conductor solicita información de estacionamientos cercanos hasta que recibe la lista de estacionamientos.

Alcances: Búsqueda de estacionamientos cercanos a la posición actual.

Beneficios: Permite encontrar estacionamiento cerca de su posición a través de la geolocalización por medio del celular.

Desventajas: No se sabe a qué distancia exacta están ni la disponibilidad de lugares que existe en cada una de ellas.

Fuente de información: Google Play Store

Metodología de trabajo y desarrollo a aplicar:

Utilizaremos SCRUM como marco de trabajo de desarrollo, ya que los integrantes del equipo compartimos un enfoque ágil para el desarrollo y contamos con un entusiasmo importante para aprender a usar dicha metodología de trabajo. En cuanto a la aplicación a este producto en particular, consideramos que este marco de trabajo nos beneficiará al momento de comprometernos con el desarrollo del producto para alcanzar los objetivos propuestos, ya que los mismos son a corto plazo y de un tamaño muy manejable.

Específicamente, trabajaremos con Sprints de 2 semanas, en los cuales contemplamos 6 días de trabajo, cada integrante del equipo trabajara aproximadamente 24hs en cada Sprints en las diferentes tareas planificadas en el mismo. Nos hemos visto forzados a realizar una adaptación a Scrum, la misma fue realizar un Sprint 0 con diferente duración, esto lo realizamos para enfocarnos en el Sprint 0 en la documentación requerida por la cátedra para la presentación del proyecto. Cada Sprint conlleva su reunión de planificación, sus 6 reuniones diarias, la reunión de retrospectiva y una reunión de revisión.

En cuanto a los artefactos que se generaran para darle solución al proyecto, serán definidos en cada Sprint, podemos decir que realizaremos diagramas de clases del producto completo, y que, para las funcionalidades que se consideren necesarios, se realizaran diagramas de secuencia y diagramas de transición de estados, para modelar la gestión de datos realizaremos diagramas de entidad relación.

Actualmente el equipo carece de ciertos conocimientos necesarios para llevar a cabo este proyecto, es por eso que en el sprint 1 se realizaran capacitaciones en MVC, SCRUM y Android. Cada capacitación será realizada por los 5 integrantes del equipo, quienes luego compartirán sus aprendizajes, éxitos y fracasos, debiendo generar un documento de capacitación en donde queden registrados.

El rol de Product Owner será rotativo entre los integrantes del equipo durante el transcurso de los distintos sprints. Si bien poseemos un conocimiento básico sobre el negocio debido al estudio realizado hasta el momento, consideramos que necesitaremos un conocimiento más profundo con el transcurso del tiempo, por lo que efectuaremos un análisis más detallado sobre el mismo cuando consideremos sea necesario.

Integrantes del equipo de trabajo:

Los integrantes que conforman el equipo de trabajo son: Ezequiel Bär Coch, Marcos Barrera, Ignacio Frigerio, Leonel Romero, y Lucas Toneatto.

Para mayor información de cada uno de los mismos, revisar la sección Anexo en la cual podrá encontrar una breve currícula de cada uno.

• Los roles a desempeñar por cada uno de los integrantes del equipo serán los siguientes:

- Ezequiel Bär Coch: Team Member.
- Marcos Barrera: Team Member
- Ignacio Frigerio: Scrum Master.
- Leonel Romero: Team Member.
- Lucas Toneatto: Team Member.

El equipo de desarrollo se manejará dinámicamente de la siguiente manera:

- El Scrum Master ayudará al equipo para llegar a cumplir los objetivos de cada sprint además será el nexo entre el equipo y el Product Owner. En nuestro caso, utilizaremos un esquema rotativo de rol de Product Owner debido a que como nuestro proyecto concluirá en un producto que no está destinado a una empresa, seremos nosotros mismos.
- Los roles de Tester y Desarrolladores, serán rotativos entre los sprints, de forma tal que cada integrante desarrolle ambas actividades en distintos sprints.

Proyectos de capacitación complementarios

Debido a la falta de conocimiento en tecnologías que no nos son provistas por la universidad, y consideramos necesarias para el desarrollo del proyecto y del producto, dedicaremos gran parte del tiempo en los momentos iniciales a capacitarnos en estas tecnologías. A continuación exponemos los principales temas sobre los cuales nos capacitaremos.

Geolocalización y Navegación GPS

Área Temática: posicionamiento geográfico

Objetivos: lograr obtener los puntos de localización ya sea de los vehículos/conductores, como así también el de las playas de estacionamiento; y con ellos poder establecer una ruta de navegación a modo de GPS.

Fundamentos: debido a la gran utilización que tienen los Smartphone en la sociedad y en las prestaciones que los mismos nos brindan.

Resumen: investigar el manejo de los módulos de geolocalización y GPS de los dispositivos móviles a partir de la programación en Android.

Resultados Esperados: lograr la utilización de los recursos del celular (posicionamiento y GPS) de manera eficiente, para obtener una mayor performance del sistema.

Utilización API Google Maps

Área Temática: posicionamiento geográfico

Objetivos: lograr a través de un mapa brindar toda la información requerida para el procesamiento de ubicaciones y poder establecer rutas entre dos puntos cualesquiera del mapa.

Fundamentos: La facilidad para poder personalizar los mapas de Google y la facilidad de uso que tienen los mismos.

Resumen: investigar como cargar en un mapa toda la información de las ubicaciones de las playas de estacionamiento, como así también la de los puntos de interés.

Resultados: poder mostrar en un mapa la información de las playas de estacionamiento y los puntos de interés.

Programación Móvil - Android

Área Temática: programación de aplicaciones móviles

Objetivos: poder desarrollar una aplicación móvil que permita la utilización de los recursos de geolocalización.

Fundamentos: El masivo mercado de celulares Smart Phone y las prestaciones con que hoy en día contamos en el área de la geolocalización.

Resumen: investigar sobre programación móvil en la plataforma Android y lograr acceder a través de código al uso de los módulos de geolocalización que los dispositivos poseen.

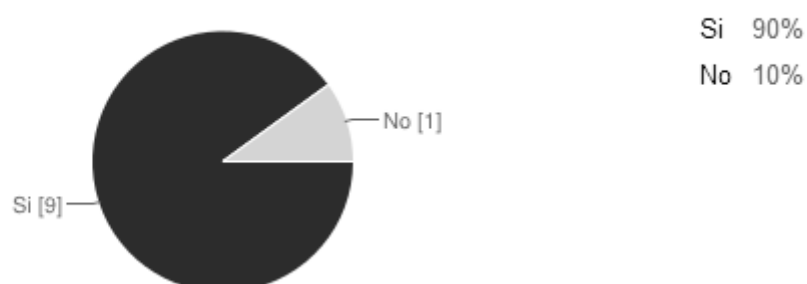
Resultados: poder obtener a partir del celular, la ubicación exacta.

Estadísticas de entrevistas

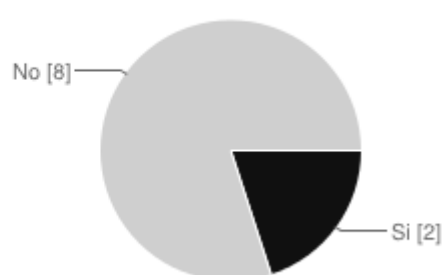
A continuación se agregan resultados parciales de las entrevistas hechas hasta el momento. Cabe aclarar que continuaremos realizando entrevistas a más playas de estacionamiento para tener una visión más global del mercado. Estos resultados parciales se basan en un total de 10 entrevistas, y los datos mostrados son solo aquellos que consideramos los más representativos al momento.

Consideramos que la cantidad actual no es una muestra lo suficientemente significativa como para sacar conclusiones. Las mismas se realizarán en un momento más avanzado del proyecto cuando contemos con una muestra lo suficientemente significativa y serán comentadas en el análisis de factibilidad y análisis de riesgos según corresponda.

Utilizan un sistema para registrar la E/S de vehículos?

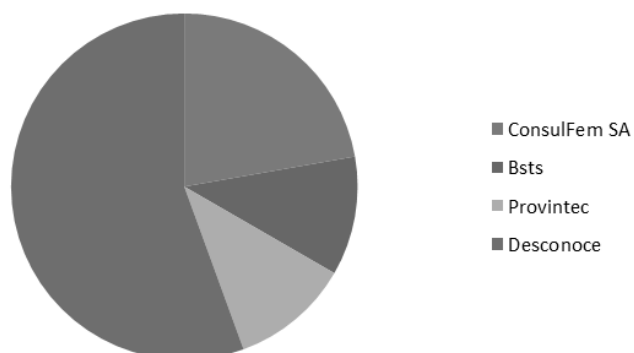


Posee acceso a Internet?



Si 20%
No 80%

Fabricante del Sistemas



Currícula del equipo de trabajo

Lucas Fernando Toneatto

Datos Personales

Nombres: Lucas Fernando

Apellido: Toneatto

DNI: 36.478.653

Fecha de Nacimiento: 17/10/1991

Nacionalidad: Argentina.

Estado Civil: Soltero.

Domicilio: Mtra. Olga Prosdócimo 58, Colonia Caroya, Córdoba.

Teléfono Fijo: 03525-463541

Teléfono Móvil: 03525-551787

E-Mail: lucastoneatto@gmail.com

Educación

- Ingeniería en Sistemas de Información
Universidad Tecnológica Nacional – Facultad Regional Córdoba
Año inicio: 2010
Año finalización (estimado): 2015 - Cursando 5° año de la carrera.
- IPEM N°65 Pbro. José Bonoris
Bachiller con orientación en Gestión de las Organizaciones
Año finalización: 2009
Distinción Académica: Alumno Distinguido (2009)

Experiencia

- ACTUALIDAD
Municipalidad de Córdoba - Dirección Catastro
Pasante: Desarrollador y soporte técnico.
Ingreso: Marzo 2014
- UTN-FRC
| Laboratorio de sistemas
Becario en el Laboratorio de Sistemas
Ingreso: Abril 2013 - Egreso: Diciembre 2013
Control de equipamiento, ayuda técnica a usuarios, documentación, etc.

Conocimientos Técnicos

- JAVA: Java SE, Java EE (Jsp, Servlets, Jpa).
- .NET con C#
- HTML, CSS.
- SQL
- UML
- Análisis y diseño de Sistemas de Información.

Aptitudes

- Identificación y análisis de problemas.
- Trabajo en equipo.
- Capacidad analítica.
- Responsable y comprometido.
- Capacidad de investigación.

Marcos Barrera

Datos Personales

Nombres: Marcos

Apellido: Barrera

DNI: 35.576.103

Fecha de Nacimiento: 19/11/1990

Nacionalidad: Argentina.

Estado Civil: Soltero.

Domicilio: Claudio Cuenca 1410 Escobar, Córdoba.

Teléfono Móvil: 0351-15388743

E-Mail: marcos.barrera90@gmail.com

Educación

- Ingeniería en Sistemas de Información
Universidad Tecnológica Nacional – Facultad Regional Córdoba
Año inicio: 2009
Año finalización (estimado): 2015 - Cursando 5° año de la carrera.
- Instituto León XIII
Título obtenido: Bachiller Humanista orientación en Ciencias Sociales.
Particularidad: Su plan de estudios incluía latín, griego, inglés y francés.
Promedio general: 7,81
- 1997-2002 Instituto Inmaculada Concepción
Colegio Bilingüe. Español – Inglés.

Experiencia

- Desde 2014 AVEIT Asociación Vocacional de Estudiantes Tecnológicos
Presidente Subcomisión Gestión Social y Ambiental
- Desde 2012 AVEIT Asociación Vocacional de Estudiantes Tecnológicos
Subcomisión Relaciones Institucionales
- 2008-2011 Armado y Reparación de PC
Independiente

Conocimientos Técnicos

- Programación de aplicaciones: Plataforma J2EE (JAVA, Servlets, JSP, EJB, JDBC), ASP .net, PHP, HTML, C# y Visual Basic.
- Frameworks: Struts 2, Hibernate (Java) ,Hibernate (.Net)
- Entornos de Programación: Visual Studio, Eclipse, NetBeans y Jboss.
- Servidores de aplicaciones: Glassfish, Jboss, Apache, Tomcat.
- Bases de datos: PL/SQL Oracle, SQL Server, MySql, Access.
- Diseño de aplicaciones y BD: Power Designer, Metodología UML.
- Sistemas Operativos: MS-DOS, Windows 2000-XP, Vista, 7 y 8. Linux Ubuntu.
- Ofimática: Word, Excel, Access y PowerPoint nivel avanzado.

Leonel Romero

Datos Personales

Nombres: Leonel

Apellido: Romero

DNI: 34.768.687

Fecha de Nacimiento: 26/09/1989

Nacionalidad: Argentina.

Estado Civil: Soltero.

Domicilio: Bolívar 41 4ºA Bº Centro, Córdoba.

Teléfono Móvil: 0351-152652796

E-Mail: leonel.romero89@gmail.com

Educación

- 2008- Actualidad
Ingeniería en Sistemas de la Información, actualmente cursando 5to año
- 2000-2007 Bachiller orientado en Humanidades con especialización en Ciencias Naturales. Secundario Saúl A. Taborda.

Experiencia

- 2014-Actualidad
Mantenimiento y Desarrollo de sistemas en lenguaje .Net, Centro de Investigación y Desarrollo de Software (CIDS)
- 2012-Actualidad
Servicio técnico Sistemas Mawy (Hardware y Software), empresa situada en Bs. As.
- 2010-2011 Socio-Gerente Restaurante Naturista "Carmen" (Rondeau 616), 9 personas a cargo.

Conocimientos Técnicos

- Lenguajes de programación: VB .Net, C# .Net, Java.
- Motores de Bases de Datos: SQL Server.
- Diagramación de Redes
- Herramientas de Diseño Gráfico: Adobe Illustrator, Adobe Photoshop, Adobe InDesign, Corel Draw.

Ignacio Frigerio

Datos Personales

Nombres: Ignacio

Apellido: Frigerio

DNI: 34.797.043

Fecha de Nacimiento: 17/03/1990

Nacionalidad: Argentina.

Estado Civil: Soltero.

Domicilio: B° Centro Córdoba, Argentina. CP 5000, Córdoba.

Teléfono Móvil: 0351-152652796

E-Mail: frigerio.ignacio@gmail.com

Educación

- 2008- Actualidad
Ingeniería en Sistemas de la Información, actualmente cursando 5to año
- 2002-2007 Bachiller en economía y gestión de las organizaciones.
Instituto privado Presbítero Juan Vicente Brizuela.

Experiencia

- 2014-Actualidad
Java Developer (Java, HTML, CSS, JavaScript)
Laboratorio de Investigación y Desarrollo en Ingeniería y Calidad del Software (LIDICALSO)
- 2011-2013
Colaborador en la subcomisión de Organización y Eventos en Asociación Vocacional de Estudiantes e Ingenieros Tecnológicos (AVEIT).

Conocimientos Técnicos

- Java: Desarrollo de aplicaciones web y de escritorio nivel Jr.
- C#: Desarrollo de aplicaciones de escritorio nivel Jr., Desarrollo web basado en .NET nivel Jr.
- SQL: Conocimientos en uso de sentencias SQL y normalización de Base de Datos nivel Jr. Motor: SQL Server 2008
- Ofimática: Conocimiento intermedio de herramientas de ofimática.
- Análisis y diseño: Conocimientos sobre Análisis y Diseño de sistemas, utilización de herramientas Enterprise Architect y StartUML.
- Agile (SCRUM): Conocimientos básicos de las prácticas de SCRUM en el desarrollo de software.

Ezequiel Bär Coch

Datos Personales

Nombres: Edgardo Ezequiel

Apellido: Bär Coch

DNI: 33.414.420

Fecha de Nacimiento: 16/12/187

Nacionalidad: Argentina.

Estado Civil: Soltero.

Domicilio: Santa Fe 1181, Córdoba.

Teléfono Móvil: 0351-155467903

E-Mail: eebarcoch@gmail.com

Educación

- Universidad Tecnológica Nacional- Facultad Regional Córdoba
Ingeniera en Sistemas de Información (5º año). (Por cursar en 2014)
- Colegio Instituto Nuestra Señora del Huerto. (Córdoba, Argentina)
Título: "Especialización orientada en Ciencias Naturales, Especialidad Salud y Ambiente"

Experiencia

- ACTUALIDAD
Municipalidad de Córdoba - Dirección Catastro
Pasante: Desarrollador y soporte técnico.
Ingreso: Marzo 2014

Conocimientos Técnicos

- Conocimientos de Programación Web:
 - -HTML5
 - -CSS3
 - -jQuery/JavaScript
 - -CMS (Joomla,Drupal,WordPress)
- Conocimientos sobre Programación: - Visual Studio .NET 2012
 - (C#: Asp.Net,MVC)
 - Java
 - -PHP
 - Android
- Conocimientos sobre Bases de Datos: - Microsoft SQL Server 2008/2012

- Oracle 9i
 - IBM DB2
- Conocimientos de Herramientas de Diseño : - Corel Draw Graphics Suite X5
 - Adobe Photoshop CS3
 - Adobe Dreamweaver CS3
- Gestión de versiones y calendarización:
 - Trello
 - Git/Github
 - Ankhsvn
- Herramienta de Modelado de Sistemas:
 - Enterprise Architect

Modelado del sistema

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	
1.2_DraftA	01/02/2015	Pendiente de Revisión	Lucas Toneatto	Se actualizaron modelos de Negocio, reglas de negocio, Web Service y Presentación. Quedan pendientes cambios en modelos de comportamiento.
1.2_DrafB	05/02/2015	Pendiente de Revisión	Ignacio Frigerio	Se actualizo el diagrama de clases del modelo de negocio
1.0	21/02/2015		Ezequiel Bär Coch	Actualización y reestructuración de estilos en homogeneización con demás documentos. Se crea tabla de imágenes y sección acrónimos. Se agrega explicación de diagrama de clases móvil.

Acrónimos

El siguiente listado tiene como objetivo referenciar a los acrónimos utilizados durante el desarrollo del presente documento.

Acrónimo	Significado
BD	Base de datos
DC	Diagrama de clases
DS	Diagrama de secuencia
DD	Diagrama de despliegue

Introducción

El presente documento técnico constituye parte del desarrollo e implementación del Sistema GeoParking. En el mismo, se podrá observar los principales módulos que lo componen y cómo estos interactúan para brindar información. Describe el modelado del sistema del entorno Web y el entorno Móvil, como parte complementaria de la Arquitectura del Sistema. El modelado 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 la estructura del sistema GeoParking, donde el diseño de la solución considera desde el ingreso de datos e información, la manera en que estos estarán almacenados, así como la forma en que estos se procesarán, analizarán y presentarán al usuario final.

Las imágenes que se mostrarán 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

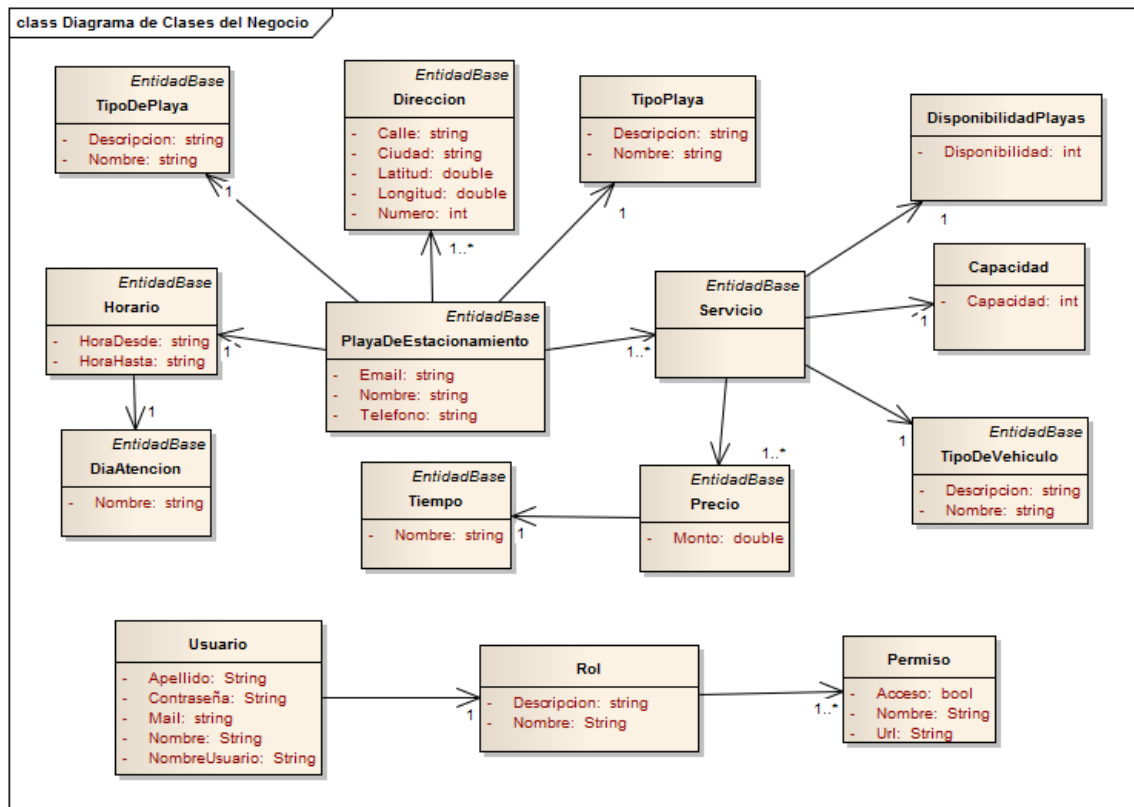


Imagen 1: DC - Del negocio

Como se puede observar el modelo de negocio está dirigido a abarcar todos aquellos elementos y características que se destacan en el negocio de playas de estacionamiento, como ser: playas, vehículos, tipos de vehículos, horarios, días de atención, precios, disponibilidades, etc.

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 de estacionamiento de vehículos particulares en playas de estacionamiento. Es hacia ese dominio, que se inserta GeoParking para dar solución a problemas y generar mejoras en el manejo de información correspondiente.

Y por lo general, en la mayoría de los sistemas, el modelo de negocio también abarca al usuario, el cual interactúa con esos elementos donde su rol es brindar datos u obtener información a partir de los mismos.

Modelo de diseño

A continuación se mostrará la estructura principal del diseño que da soporte a la solución implementada. Se podrá visualizar tanto el diseño para la aplicación web como para la aplicación móvil.

Diseño aplicación WEB

La aplicación web de GeoParking, está constituida por una arquitectura Cliente - Servidor, siendo un navegador web el cliente y un servidor dedicado a responder las solicitudes y enviar la información al cliente (navegador web). Este modelo posee un patrón N-tier de 4 capas:

- Datos: Almacenamiento físico
- Entidades: Clases que se vinculan con las tablas en la base de datos y permiten obtener datos y la interacción entre ellos para cumplir con las funciones.
- Reglas de negocio: Comprende toda la lógica y mecanismos empleados para el funcionamiento del sistema.
- Presentación: Interfaz web que permite la interacción con el usuario.

Este diseño se definió con el objetivo de tener una aplicación modular, que posee seguridad y brindar escalabilidad de su funcionalidad.

Los módulos principales de esta aplicación son:

- ✓ Búsqueda de playa: Permite a un usuario buscar y consultar información acerca de playas de estacionamiento de una determinada ciudad.
- ✓ Administración de usuarios: Alta, baja, modificación y consulta de usuarios del sistema.
- ✓ Administración de roles y permisos: Para seguridad, cada usuario tiene un rol y permisos sobre los distintos recursos del sistema.
- ✓ Gestión de solicitudes de conexión de playas: Permite que un dueño de playa, solicite que su playa esté en el sistema GeoParking para que, según se apruebe o no dicha solicitud por el administrador del sistema, pueda ser visualizado por los demás usuarios.
- ✓ Gestión de estadísticas y reportes: Módulo encargado de recolectar información, procesarla y mostrarla en formato gráficos de distintos tipos para poder que pueda ser utilizado por un administrador para la toma de decisiones.
- ✓ Gestión de zonas: Permite registrar zonas de interés para los administradores y en base a estas, utilizarlos como filtros en las estadísticas.

En la siguiente imagen se muestra el diagrama de clases de diseño de la aplicación web.

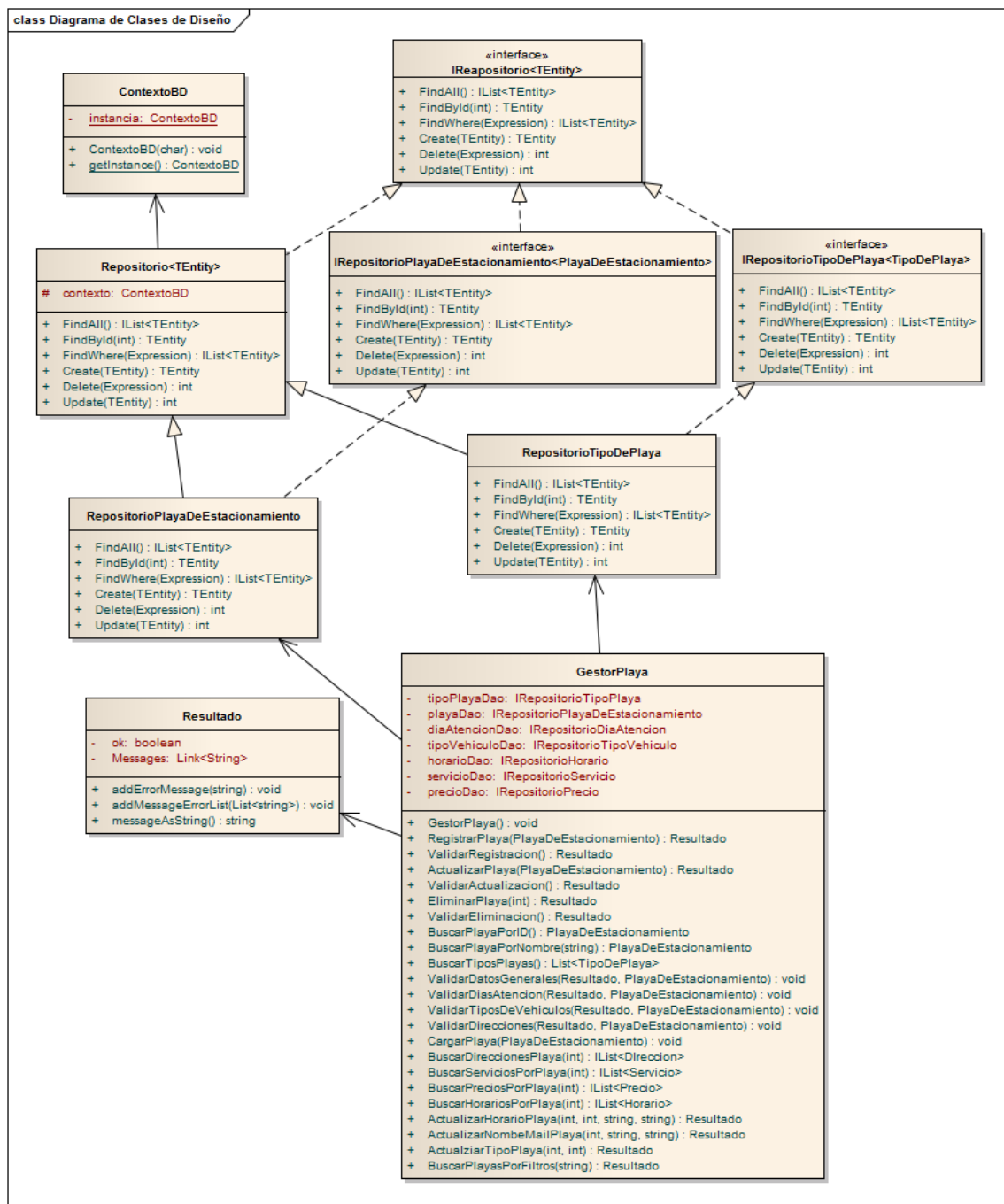


Imagen 2: DC - Diseño aplicación web

En la imagen se puede apreciar el modelo de solución implementado por GeoParking, en el cual se destaca la implementación de un patrón llamado **REPOSITORIO**, utilizado para dar soporte a la recuperación de información desde la BD y como esta es mapeada en entidades a nivel de dominio.

Todo esto se puede lograr través de la intervención de **GESTORES**, especializados en manejar un grupo de entidades específicas.

Estos **GESTORES** se dedican a manejar las entidades y realizar con ellas operaciones sobre la BD. Operaciones tales como búsqueda, inserción, eliminación y consultas.

A modo de simplificar el diagrama, solo se puede visualizar a modo de ejemplo el GestorPlaya, pero los demás gestores, se presentan en la solución con las mismas relaciones destacadas en este diagrama.

A continuación se ilustran los demás gestores que conforman el diseño de la solución GeoParking.

Nota: Para una mayor visualización del contenido de la imagen, remitirse al documento digital anexado.

En la siguiente imagen se muestra en un diagrama de clases, los distintos gestores que componen y manejan la lógica de la aplicación, con sus respectivos métodos:

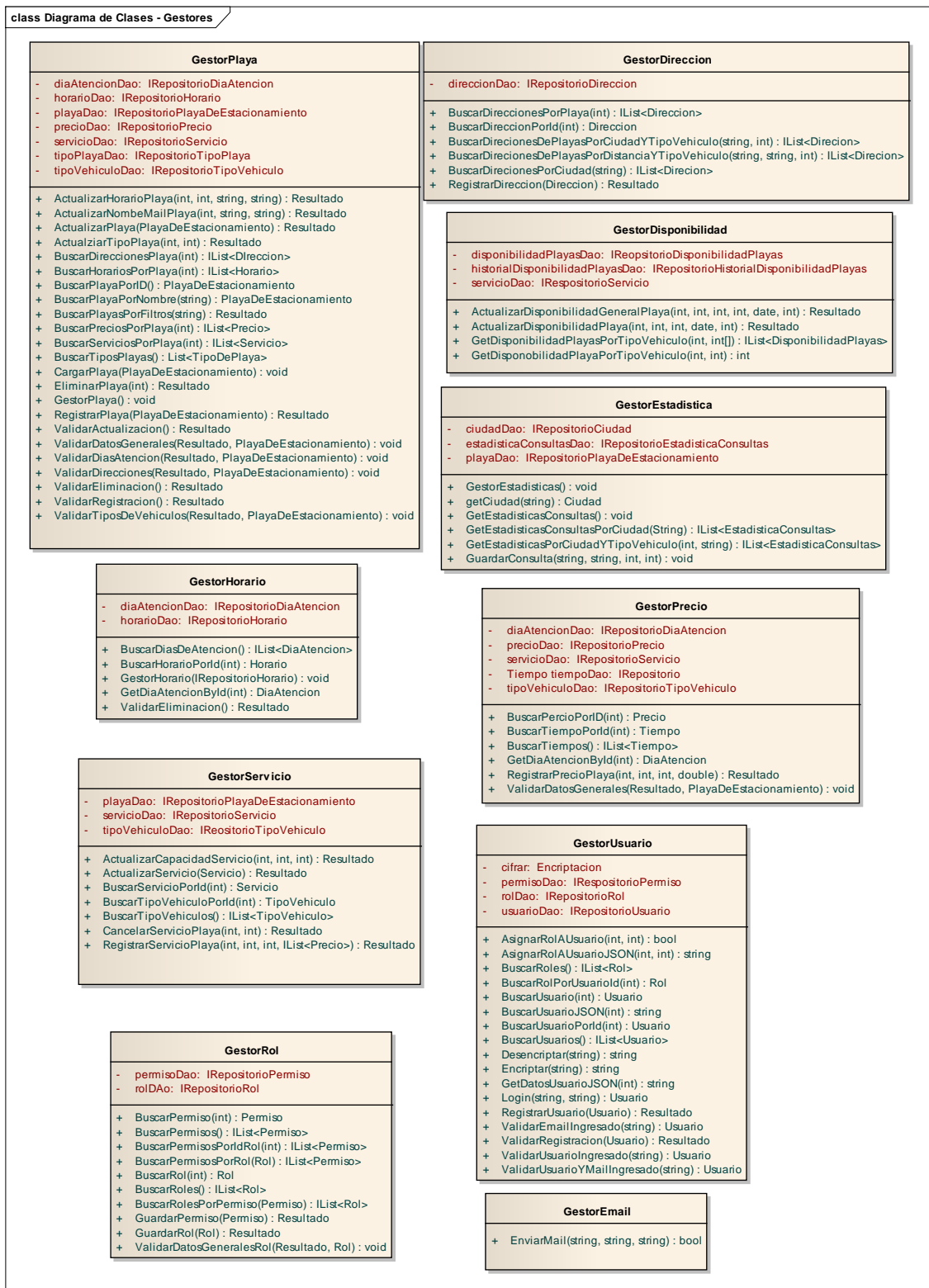


Imagen 3: DC - Gestores aplicación web

Diseño aplicación MÓVIL

La aplicación móvil, está diseñada para ser utilizada desde un dispositivo móvil con conexión a internet y un sistema operativo:

- **Android:** versiones 2.3.3 o superior.
- **iOS iPhone:** versiones 5.1.1 o superior.
- **Windows Phone:** versiones 7.5 o superior.

Las principales características de esta aplicación son:

- ✓ Administración de visualización de playas: Cargar el mapa con las playas registradas y mostrar aquellas playas comprendidas en el radio de búsqueda configurado, en un listado que muestra la disponibilidad e información principal de cada una.
- ✓ Administración de recorridos en mapa: Trazado de recorridos hacia una playa o hacia la ubicación del vehículo almacenado. También comprende la presentación de la información de la trayectoria (paso a paso).
- ✓ Administración de búsquedas por puntos de interés: Todo lo referido a la búsqueda por direcciones o puntos de interés (comercios, cines, hospitales, restaurants, etc.) y además permitir la visualización de aquellas playas que estén comprendidas en el radio de búsqueda configurado, en un listado que contenga la disponibilidad y principal información de las mismas.
- ✓ Gestión de configuraciones: Permitir al usuario la personalización de ajustes que son tomados como parámetros para los filtros de búsquedas realizados: Tipos de vehículos, radio de alcance de búsqueda y la opción de GPS para una mayor precisión en los resultados de geo posicionamiento.
- ✓ Gestión de historial de consultas: Permite contar con un histórico de las últimas playas consultadas por el usuario para que el mismo pueda recurrir a ellas y visualizar las playas en el mapa de una manera más directa.
- ✓ Gestión de almacenamiento interno: Referido a guardar y recuperar información en el dispositivo. Esto permite el almacenamiento de tipo de vehículo, radio de búsqueda, opción GPS configurado por el cliente; por otra parte, se almacena el historial de búsqueda y la posición del vehículo cuando el usuario lo registra.

A continuación, se muestra una imagen que contiene el diagrama de clases de diseño empleado para la realización de la aplicación móvil GeoParking.

Nota: Para una mayor visualización del contenido de la imagen, remitirse al documento digital anexo.

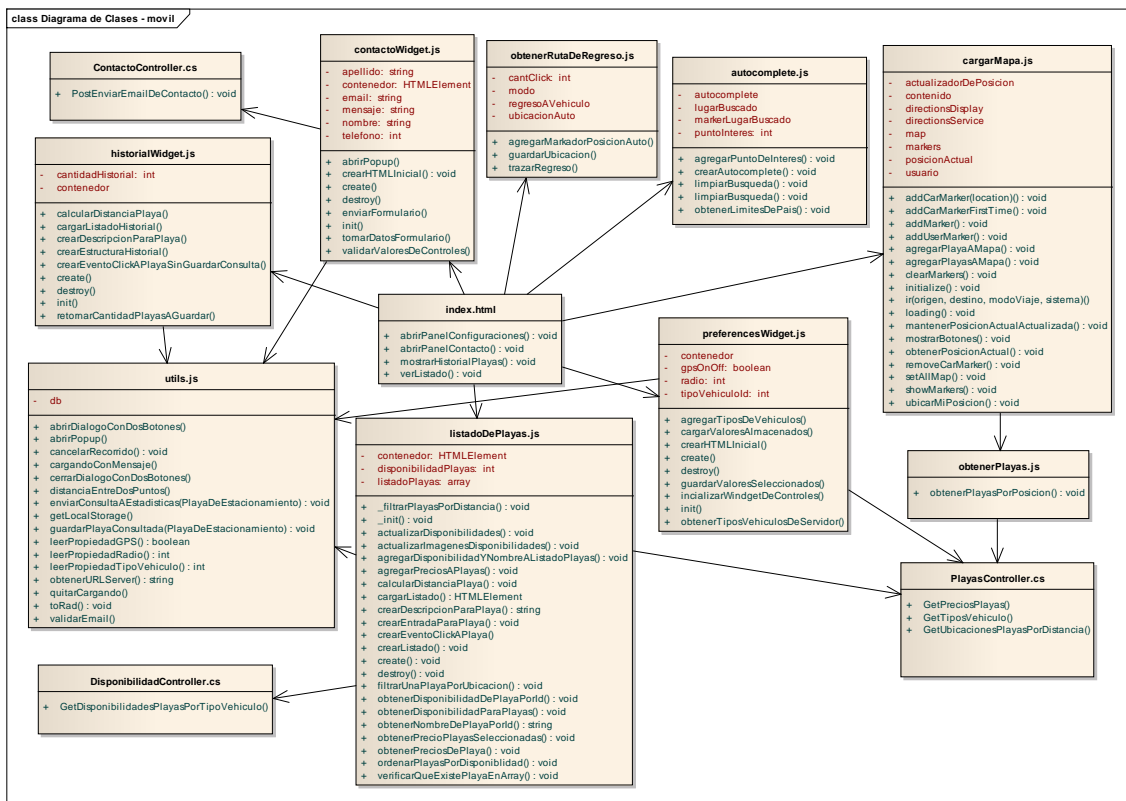


Imagen 4: DC - Aplicación móvil

Como se muestra en la imagen anterior, el funcionamiento se centra en un elemento HTML llamado **index.html** el cual representa la capa de presentación al usuario y por detrás existe un conjunto de operaciones (reglas de negocio) en distintos archivos confeccionados para tal fin:

El archivo **Utils.js** es de propósito general facilitando operaciones como inicializar mensajes, ejecutar conversiones de sistema decimal a radian para la manipulación del mapa, validar formularios, entre otros. Tiene como atributo el **db** que representa el objeto **localStorage** con el cual se hace uso de la base de datos interna del dispositivo.

Se cuenta con un conjunto de widgets:

- **cargarMapa.js**: este es el que inicia el mapa utilizando la API de **GoogleMaps v3**.
- **obtenerPlayas.js**: se encarga de conectarse con el servidor mediante peticiones AJAX al WebService, para obtener las playas registradas en la base de datos y mostrarlas en el mapa.
- **historialWidget.js**: el cual tiene como objetivo es almacenar y recuperar la información sobre las últimas playas consultadas por el usuario móvil, almacenándose con la tecnología **localStorage**, en el propio dispositivo móvil.
- **preferencesWidget.js**: su principal objetivo es almacenar y recuperar la información brindada por el usuario sobre el tipo de vehículo seleccionado, el radio de distancia de búsqueda y la activación/desactivación del GPS.
- **contactoWidget.js**: como su nombre indica, tiene como función principal, crear un formulario web y realizar petición AJAX al WebService para el envío de correos.
- **Autocomplete.js**: realiza la inicialización del objeto **places** de la librería **Places** de la API v3 de google maps, con el objetivo de buscar y obtener un punto de interés en el mapa.

Modelo de interacción

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 del 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.

Aplicación web

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 “**web.aspx**” (página de inicio de búsqueda del sistema). Esta última envía a la página encargada de las búsquedas, la ciudad que ingreso el usuario, y la página “**BusquedaPlaya.aspx**” 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.

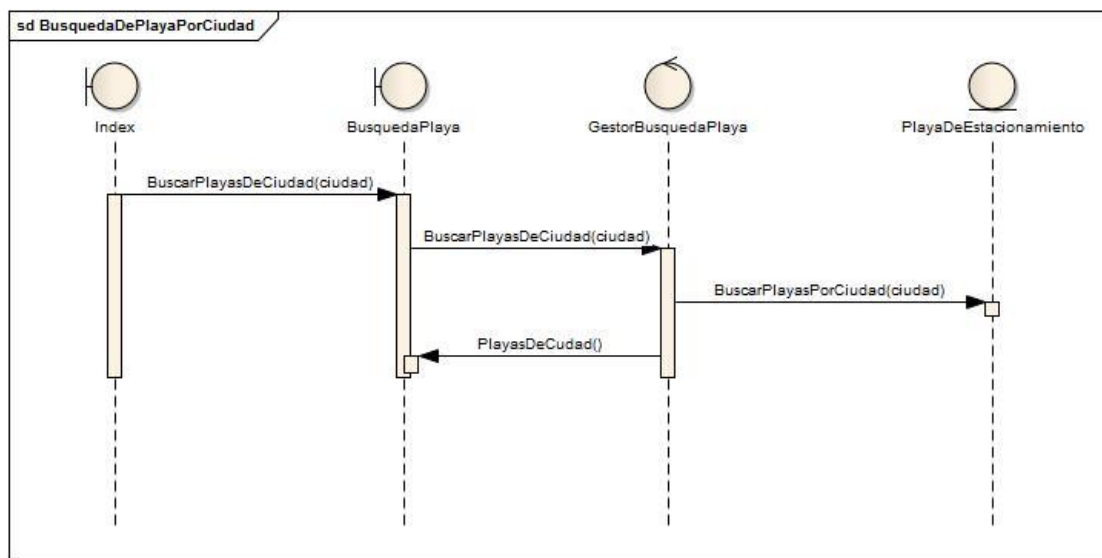


Imagen 5: DS - Búsqueda de playas por ciudad

Búsqueda de Playas por Ciudad por Filtro– Web

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

La petición se realiza en la página “**BusquedaPlaya.aspx**”. 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.

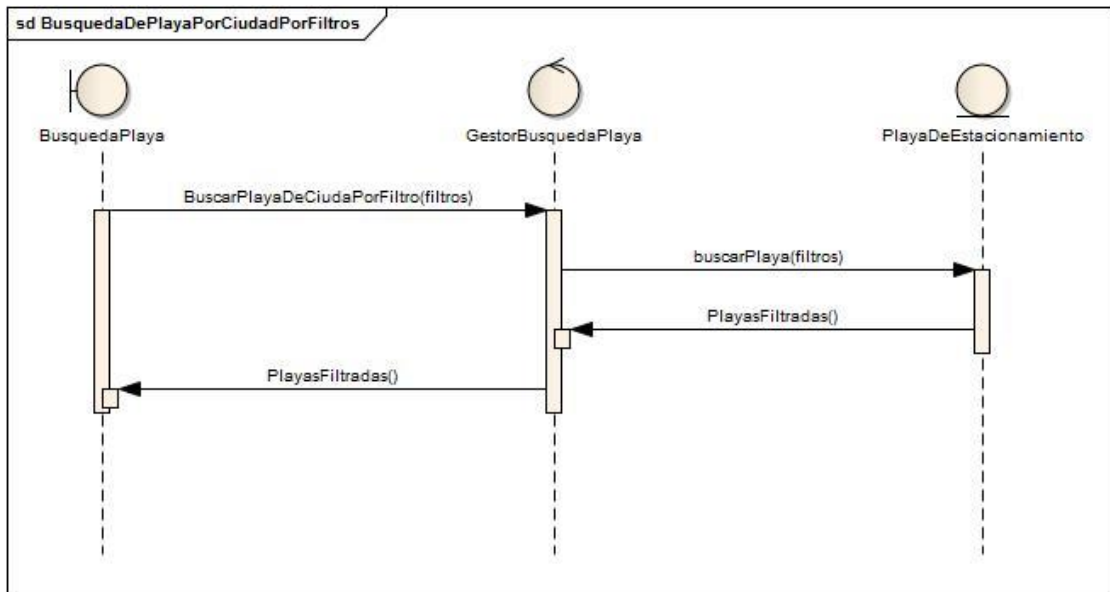


Imagen 6: DS - Búsqueda de playas en ciudad por filtros

Login de Usuario– Web

Inicio de Sesión: Aquí el usuario web selecciona la opción de ingresar al sistema, y se le muestra una 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. Caso contrario, el sistema emitirá un mensaje de error, notificando el motivo del mismo.

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.

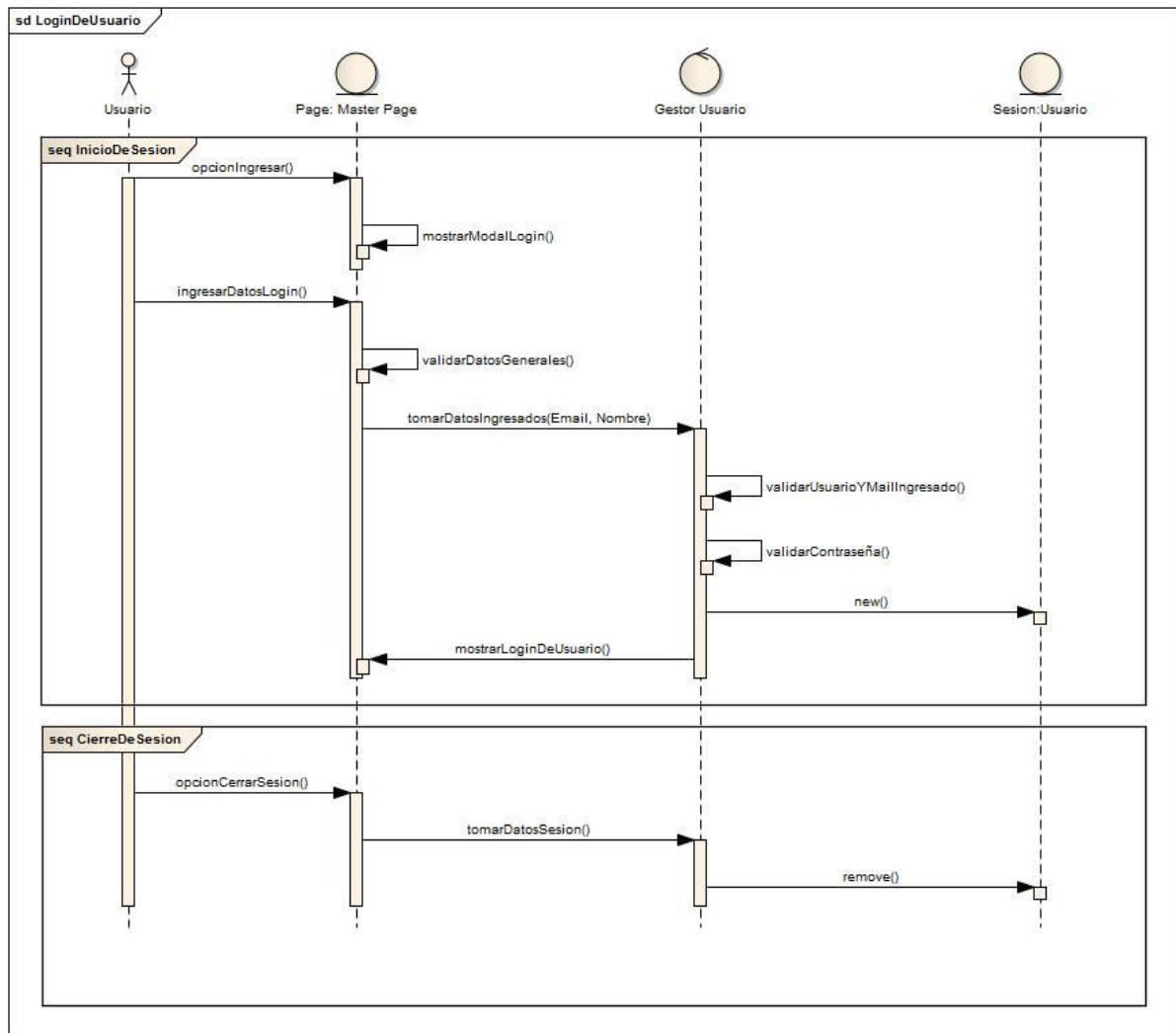


Imagen 7: DS - Inicio y cierre de sesión

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. Caso contrario, se emite un mensaje de error notificando el motivo del mismo.

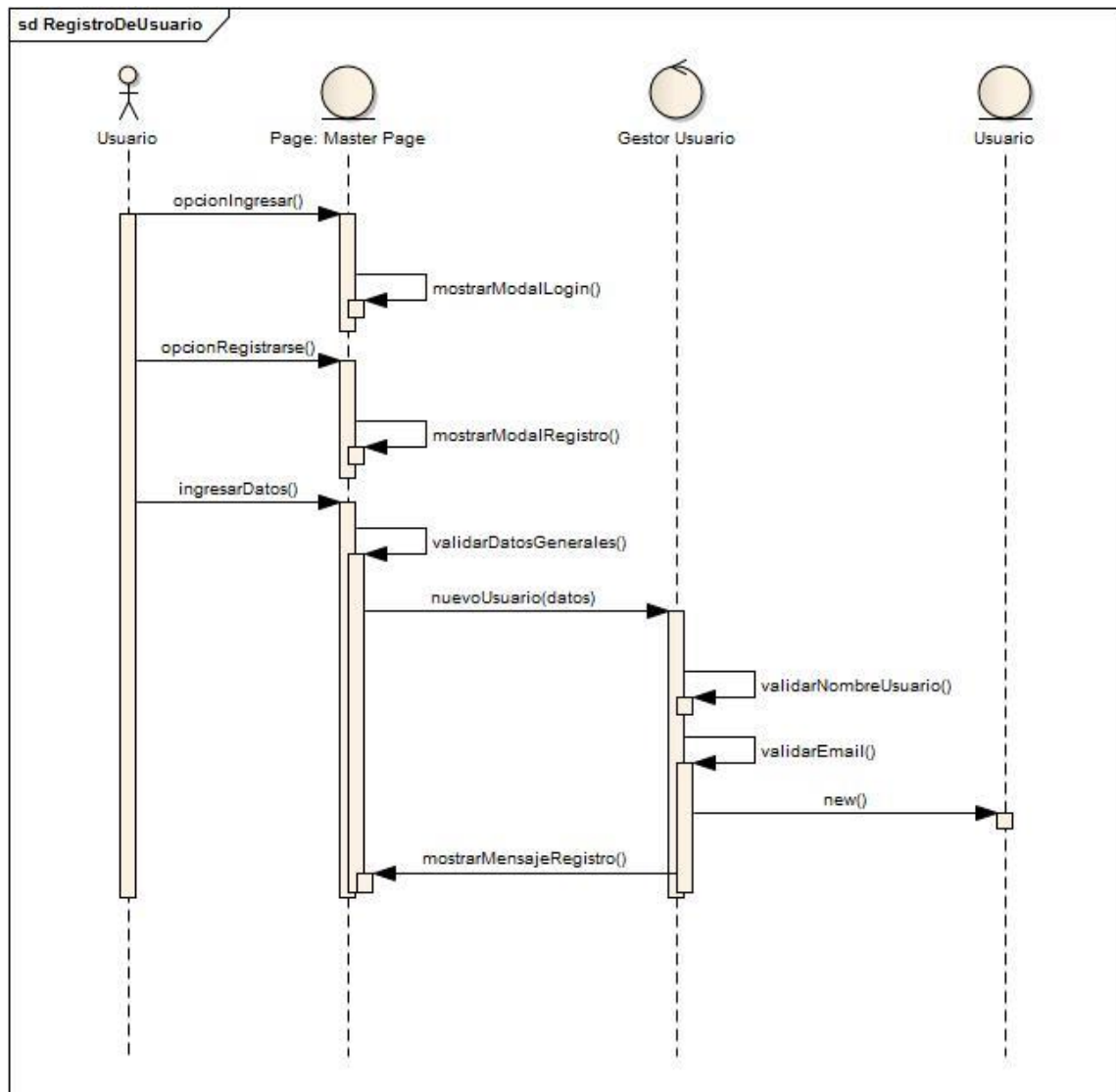


Imagen 8: DS - Registro de usuario

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. Caso contrario, el sistema emitirá un mensaje de error, notificando el motivo del mismo.

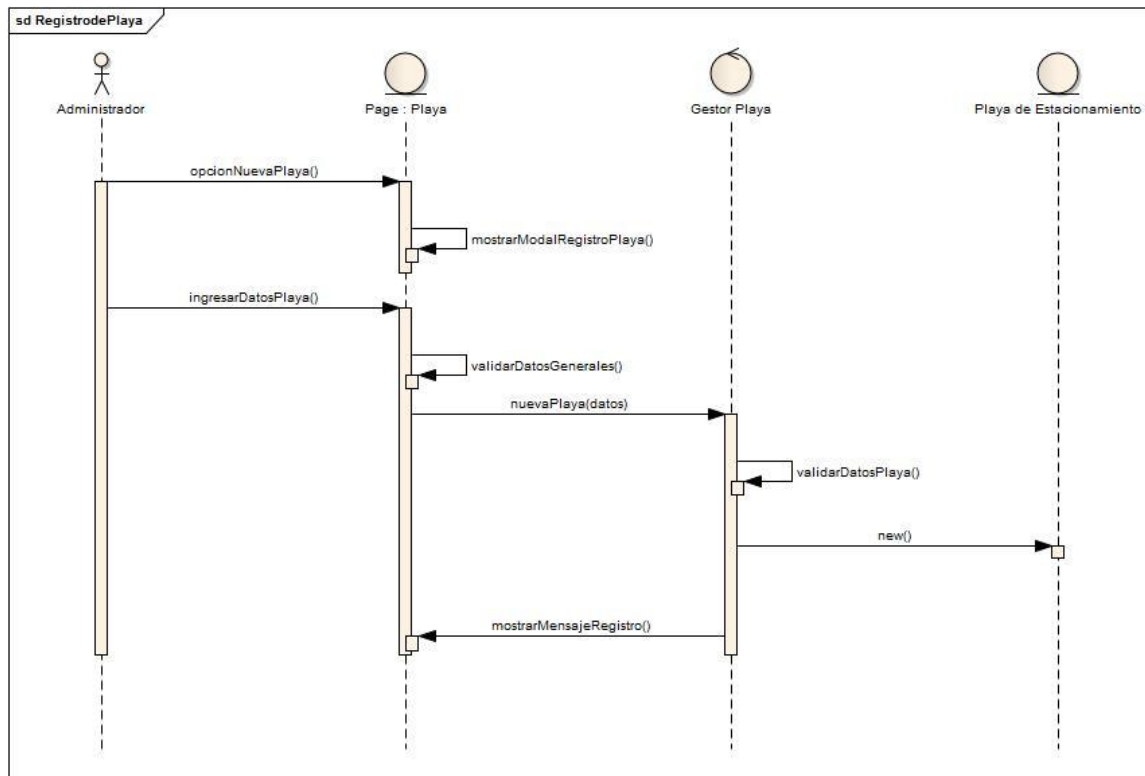


Imagen 9: DS - Registrar nueva playa

Aplicación Móvil

Consultarplayas – Móvil

Al inicializarse la aplicación se recurre a la consulta de las playas de estacionamiento. Para ello se obtiene la posición actual del usuario (su dispositivo) y se realiza la llamada al web service para buscar las playas de estacionamiento que existan en la posición pasada por parámetro y que cumpla con el filtro de tipo de vehículos, definido por el usuario durante la configuración de los ajustes. Luego estas playas son mostradas en el mapa mediante un marcador.

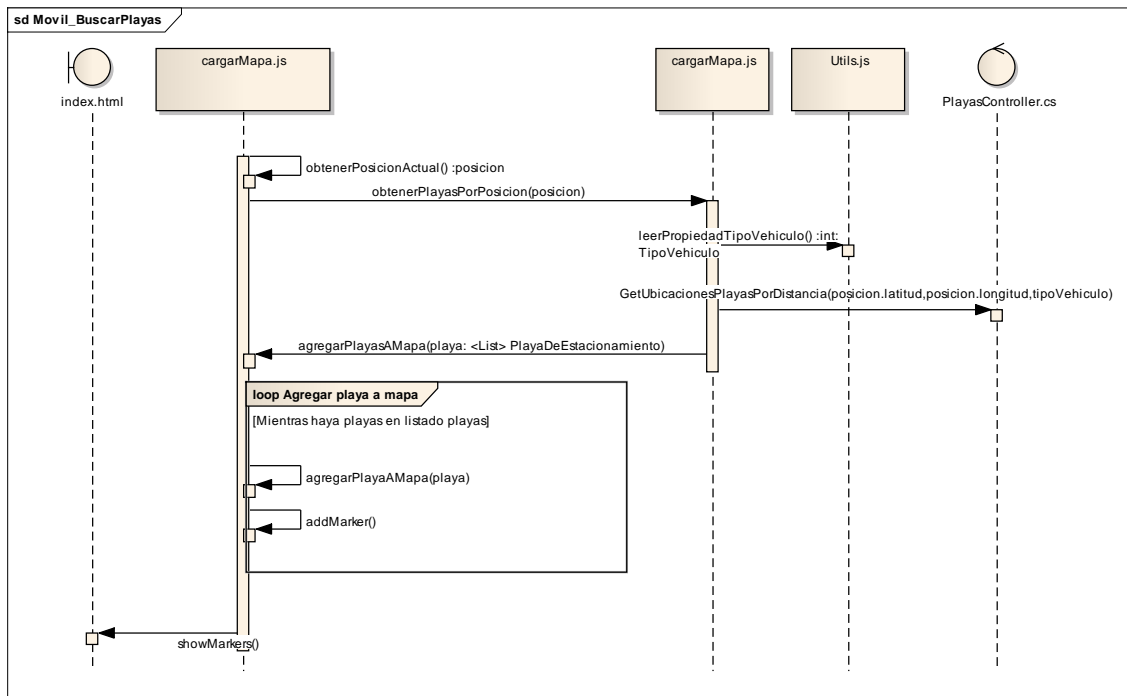


Imagen 10: DS - Consultar playas móvil

Ir a playa – Móvil

El usuario podrá visualizar el trayecto desde su posición actual hacia la playa seleccionada. Para ello, se toma la posición actual del usuario (dispositivo móvil), y el usuario podrá optar por presionar sobre un marcador de playa con su respectivo icono o desde el listado de playas generado en el apartado anterior una playa, lo que provocará que el sistema trace el recorrido por la ruta más corta posible hacia el objetivo final. Además, el usuario podrá visualizar la trayectoria presionando el botón “mostrarTrayectoria”. Ver **imagen 11**.

Enviar consultas para estadísticas – Móvil

Tiene como objetivo alimentar al módulo de estadísticas enviando datos acerca de la ubicación desde donde se realiza la consulta, el tipo de vehículo utilizado, las playas consultadas, etc.

Para ello, cada vez que el usuario consulta una playa ya sea desde el listado de playas disponibles explicadas en el inciso anterior, se envían los datos y se almacenan a través del web service. Luego estos, serán procesados y reflejados en los informes estadísticos explicados más adelante. Ver **imagen 11**.

Guardar playa en historial – Móvil

El usuario podrá tener almacenadas un conjunto de playas que son las que él consulto, ordenadas por antigüedad de consulta. Es decir, las playas más recientes aparecerán al inicio del listado mientras que las consultadas con más anterioridad serán las que ocupen la base de la lista. Ver **imagen 11**.

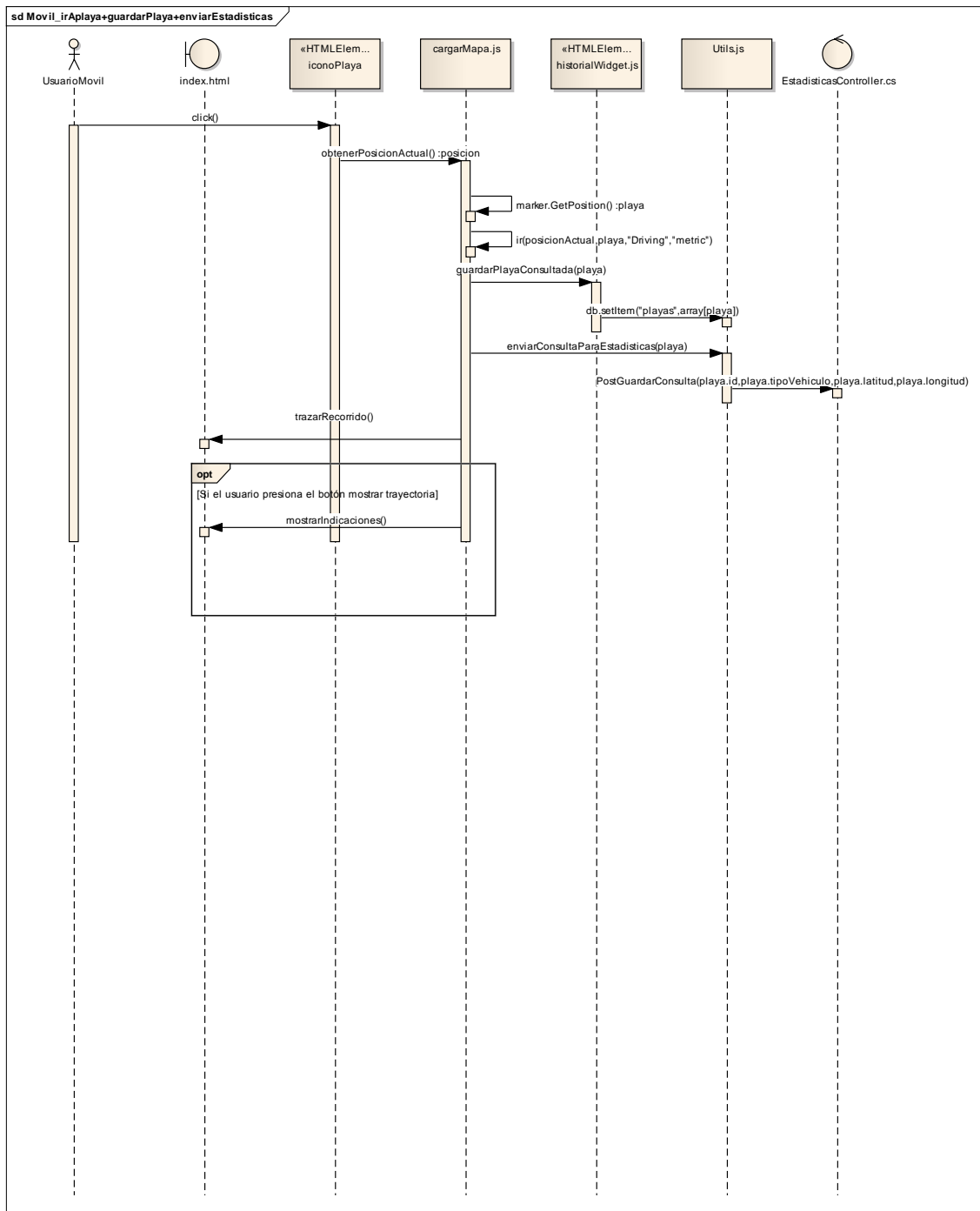


Imagen 11: DS - Ir a playa, guardar playa en historial y enviar estadísticas

Buscar playas según un punto de interés – Móvil

Con el fin de proporcionarle al usuario una forma de búsqueda a través de lugares donde pueda tener interés dirigirse o buscar lugar de estacionamiento. Estos podrían ser shoppings, parques, centros de salud, entes gubernamentales, universidades, etc.

Para ello el usuario ingresa un punto de interés o dirección, y el sistema responderá si encuentra o no dicha solicitud. En caso afirmativo le permitirá mostrar lo solicitado en el mapa y podrá buscar playas de estacionamiento cercanas en el radio comprendido entre el punto de interés y el radio configurado de búsqueda.

Guardar configuración – Móvil

El objetivo es almacenar las preferencias del usuario en cuanto al tipo de vehículo que utiliza, el radio de búsqueda de playas y la opción de habilitar /deshabilitar el GPS. Todo esto se almacena en el dispositivo en la **localStorage**. Esto es válido tanto en la pantalla de bienvenida (cuando es la primera vez que utiliza la aplicación GeoParking en el dispositivo) como en la pantalla de ajustes, donde puede configurar cuando el usuario lo requiera.

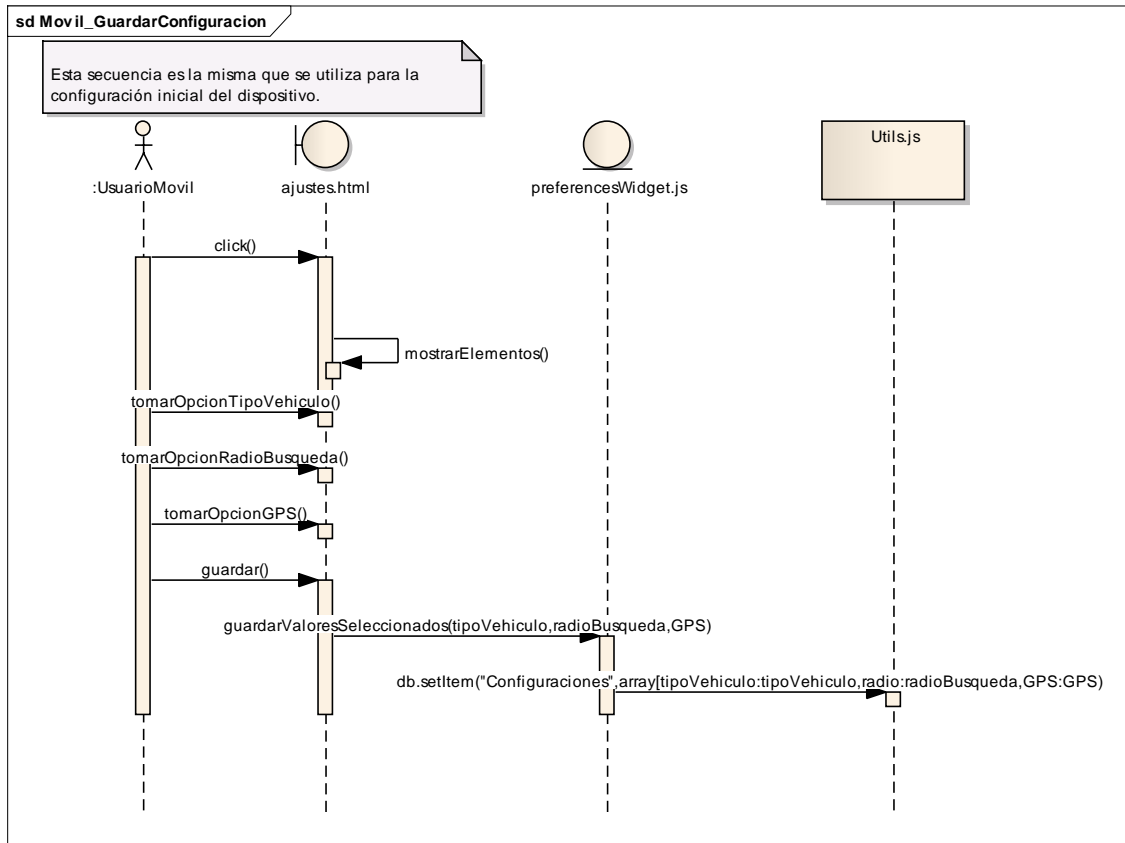


Imagen 12: DS - Guardar ajustes

Guardar posición de vehículo – Móvil

El usuario podrá almacenar la ubicación donde deje estacionado su vehículo, ya sea esta en una playa de estacionamiento o en un lugar disponible en la vía pública. Esto le permitirá luego, poder visualizar la trayectoria desde el lugar en donde se encuentra (mediante su dispositivo móvil) hasta el vehículo en cuestión. Este proceso se explica en el siguiente apartado: **Dirigirse a vehículo**.

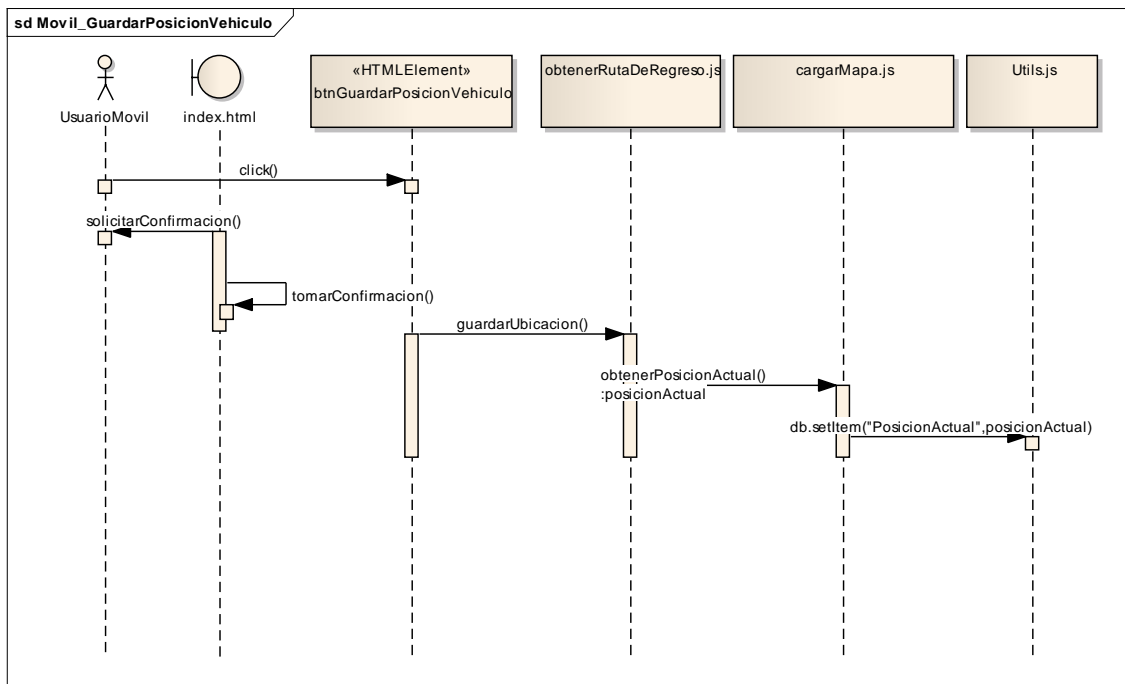


Imagen 13: DS - Guardar posición vehículo

Dirigirse a Vehículo – Móvil

Como se mencionó anteriormente, el usuario podrá visualizar el trayecto comprendido desde su ubicación actual (dispositivo móvil) hasta la posición donde registró el vehículo con anterioridad. Este trayecto está configurado para ser trazado desde el rol de un peatón.

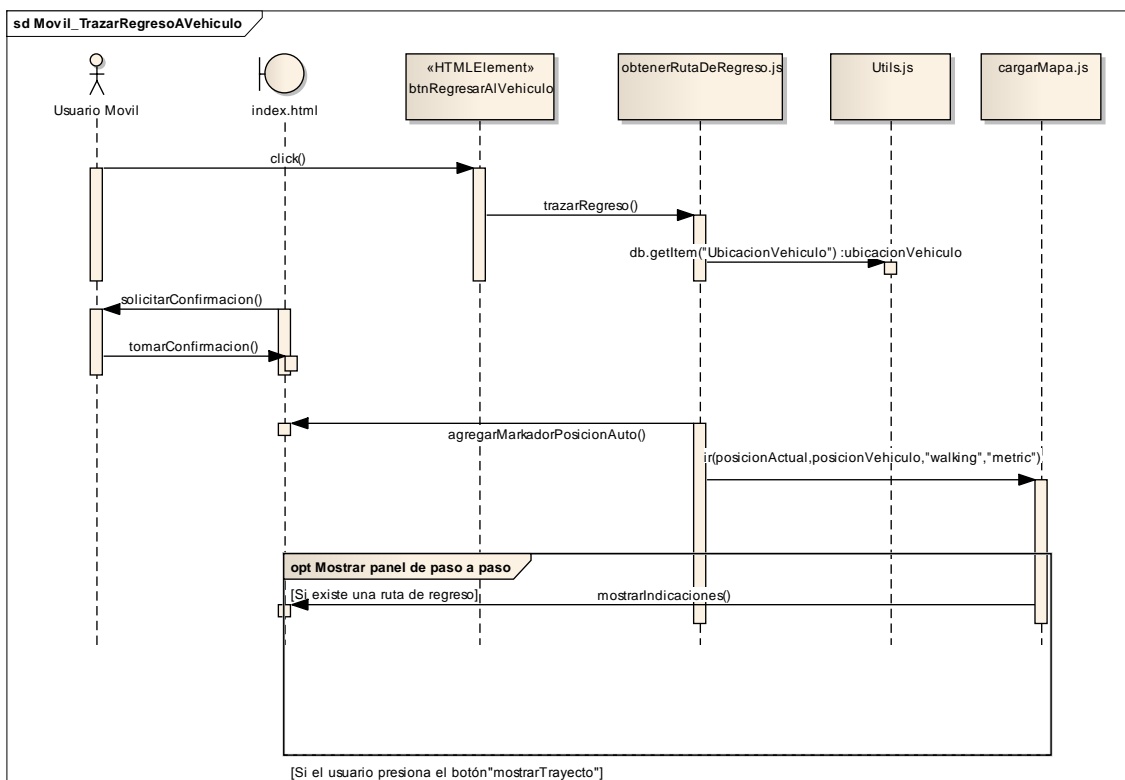


Imagen 14: DS - Regresar a la posición del vehículo

Mostrar listado de playas con disponibilidad - Móvil

Para la obtención de playas cercanas al usuario, se toma la ubicación actual del usuario (su dispositivo) a través de geo posicionamiento. Esto, junto a los datos configurados por el usuario en las preferencias (tipo de vehículo, radio de búsqueda y GPS) son enviados al web service cuyo método se encargará de retornar un listado de playas que cumplan con los filtros enviados como parámetros. Esto se presentará al usuario como un listado, mostrando en una escala de colores según su disponibilidad, nombre de las playas, dirección, precios por hora (si los tuviera) y distancia en metros desde la ubicación del usuario hacia la playa.

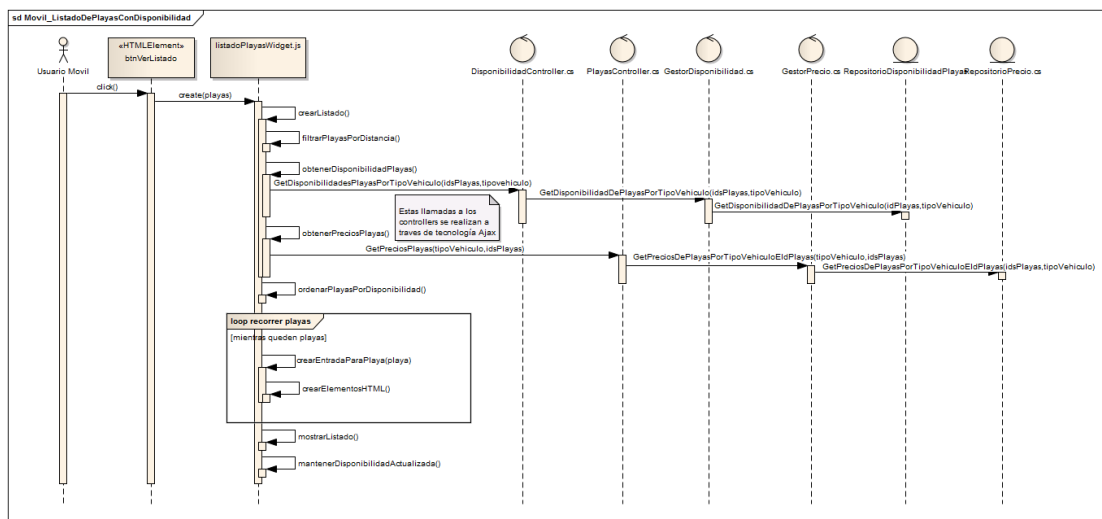


Imagen 15: DS - Listado de playas con disponibilidad

Nota: Para una mayor visualización del contenido de las imágenes, remitirse al documento digital anexo.

Modelo de arquitectura y despliegue

Nota: Para una mayor visualización del contenido de las imágenes, remitirse al documento digital anexo.

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

Servidor

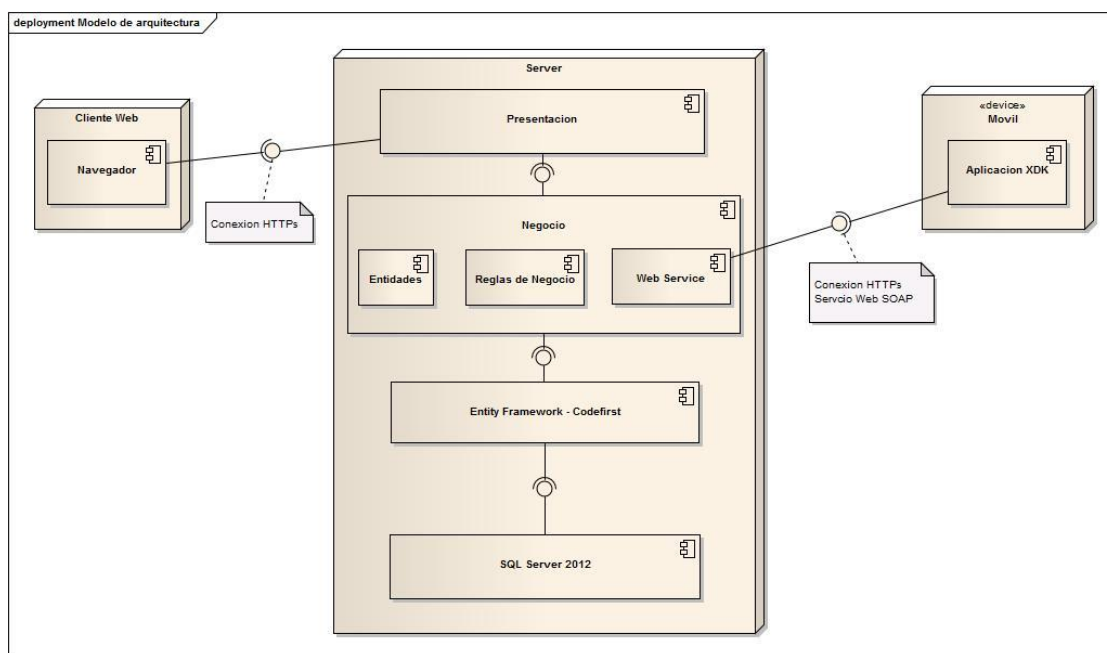


Imagen 16: DD - Modelo de arquitectura

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 web 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.

Conexión con API GeoParking

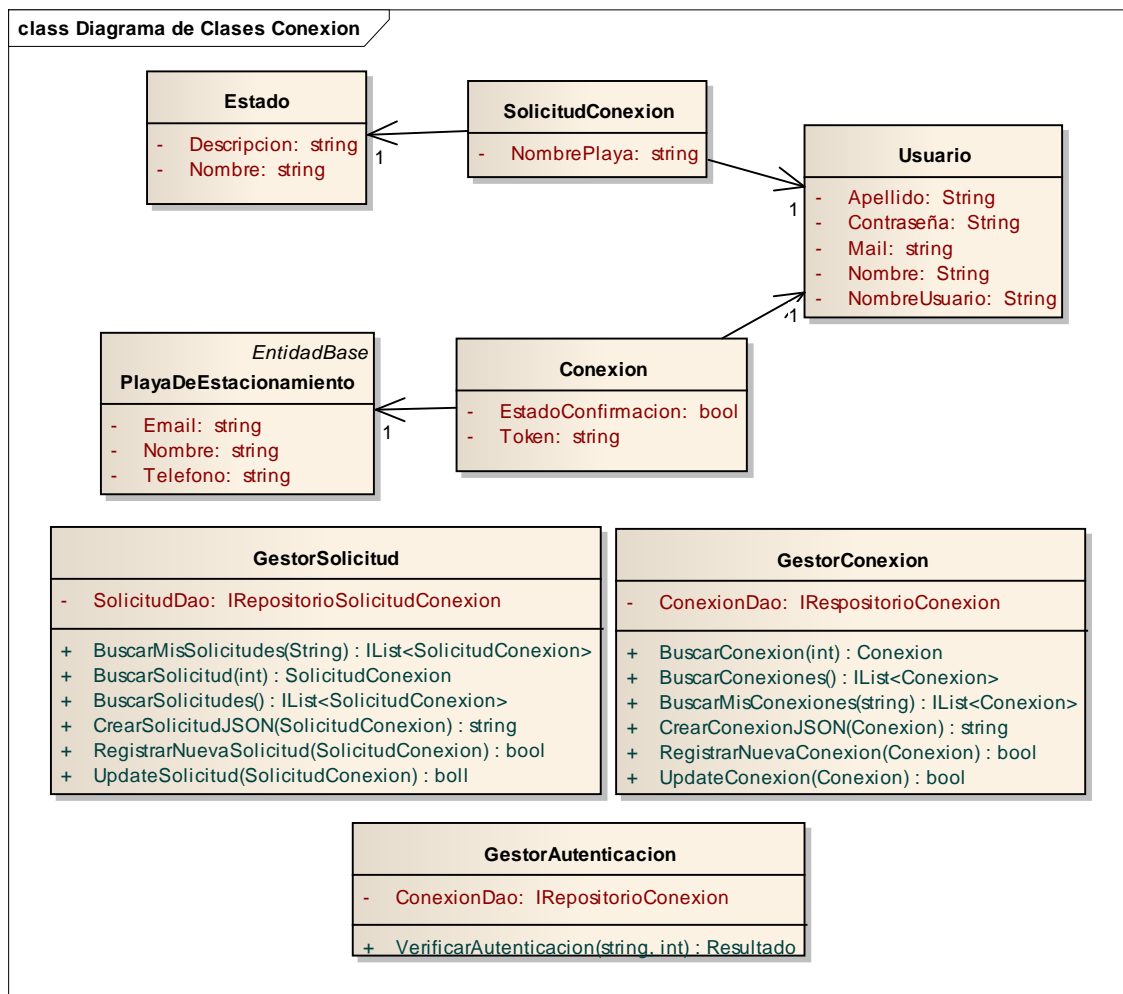


Imagen 17: DC- Conexión con API GeoParking

Como se puede apreciar los componentes presentes en el modelo, corresponden a los distintos pasos para el uso del API GeoParking por parte de los sistemas de las playas de estacionamiento.

El proceso consta de crear una solicitud, que la misma sea aprobada, y que paso seguido se genere la conexión para comunicar la playa con GeoParking.

Para ello lo destacable es la funcionalidad que proporciona cada uno de los gestores intervinientes en el proceso.

GestorSolicitud: es el encargado de crear las solicitudes para una nueva conexión y de actualizar el estado de las mismas.

GestorConexion: es el encargado de crear la conexión, que le permite a la playa obtener los datos de acceso al API de GeoParking.

GestorAutenticacion: es el encargado de validar la conexión de quien se trata de comunicar con el API de GeoParking y de darle acceso a la modificación de información de las playas.

Modelo de arquitectura AJAX (Asynchronous JavaScript And XML)

Nota: Para una mayor visualización del contenido de las imágenes, remitirse al documento digital anexo.

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.

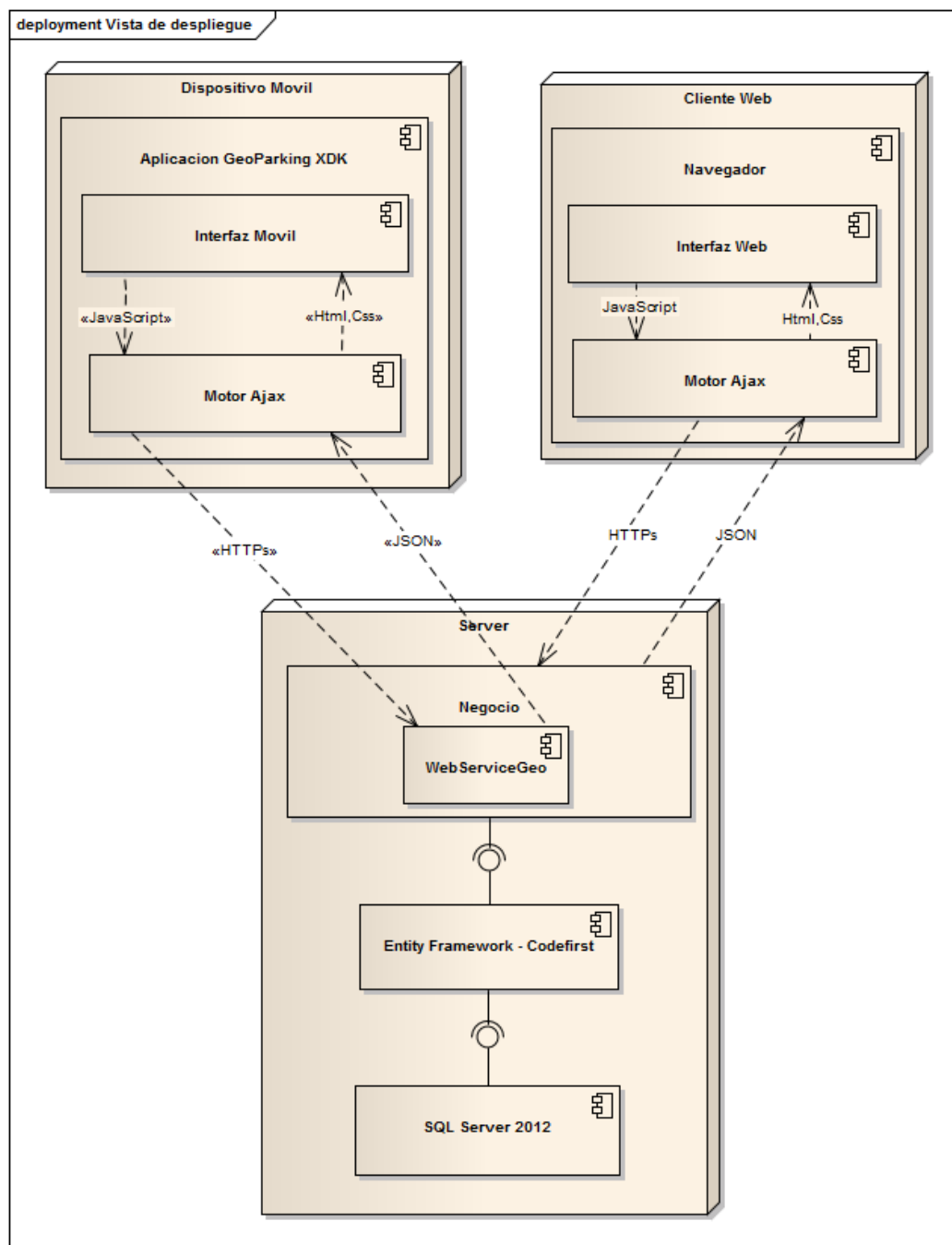


Imagen 18: DD - Vista funcional completa

Modelo de componentes

Nota: Para una mayor visualización del contenido de las imágenes, remitirse al documento digital anexo.

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 interactúan con el sistema.

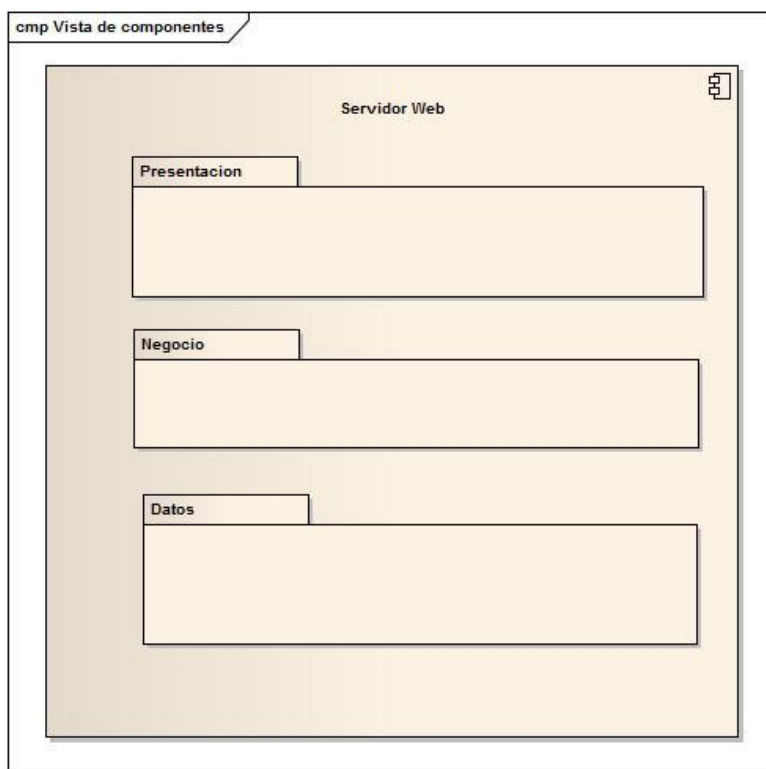


Imagen 19: Vista de componentes

Vista de Datos

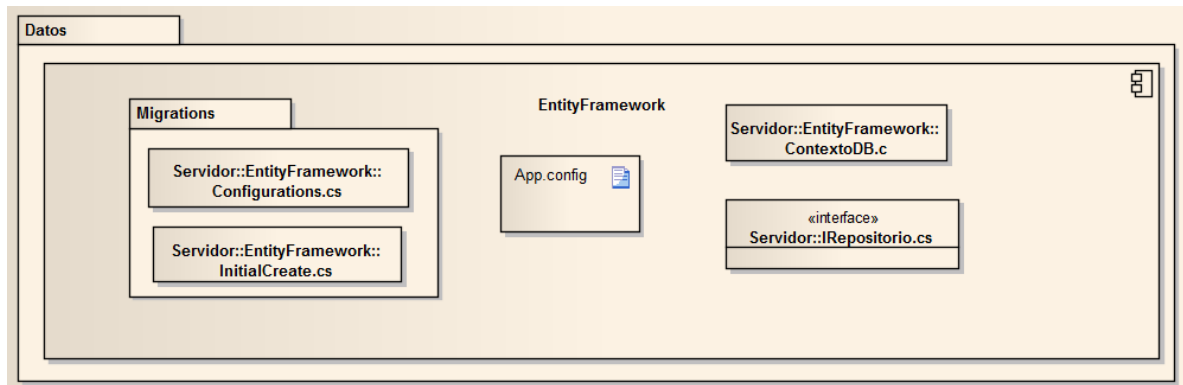


Imagen 20: Vista de datos

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



Imagen 21: Diagrama de paquetes 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

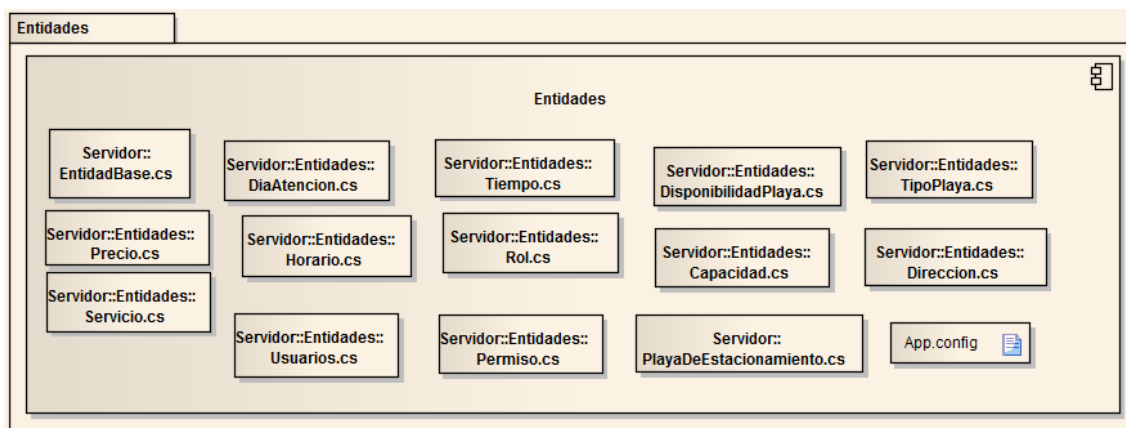


Imagen 22: Vista de entidades

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 Reglas de Negocio

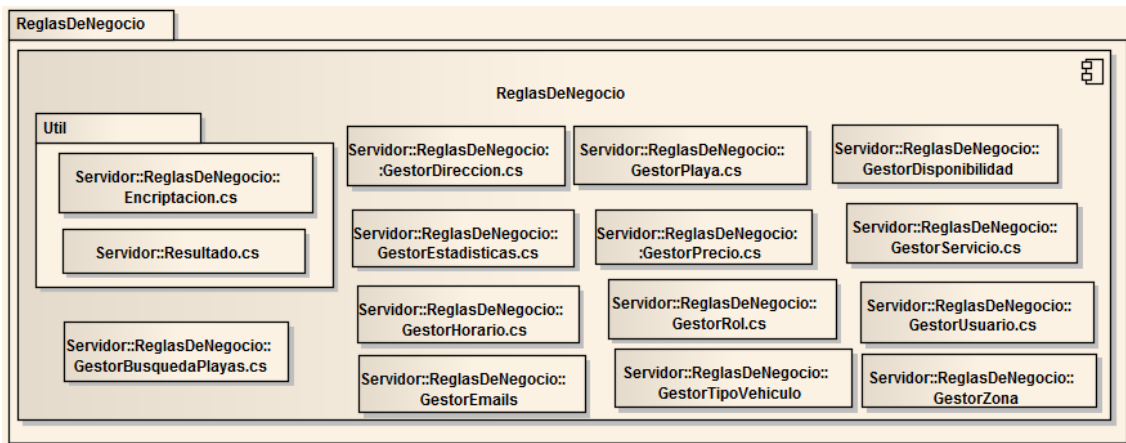


Imagen 23: Vista de reglas de negocio

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 `Encriptacion.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

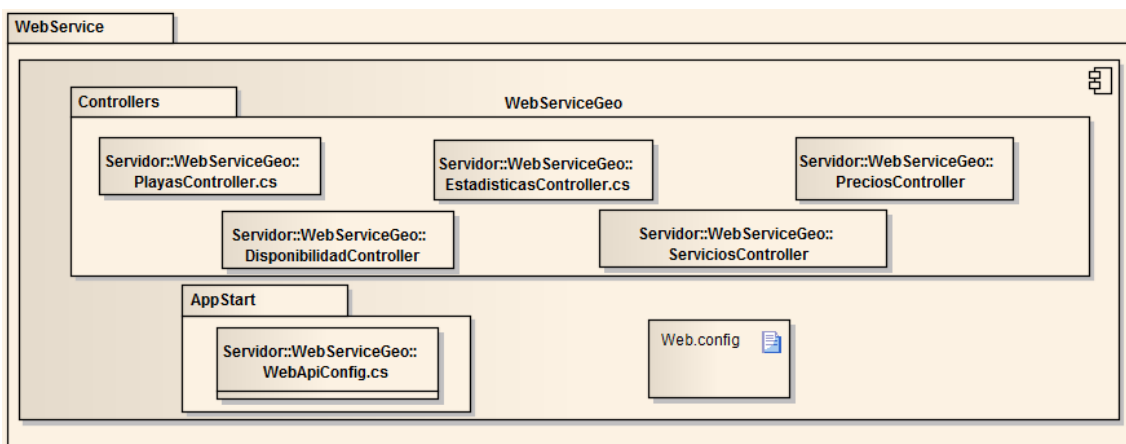


Imagen 24: Vista de Web Service

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 procesamiento 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

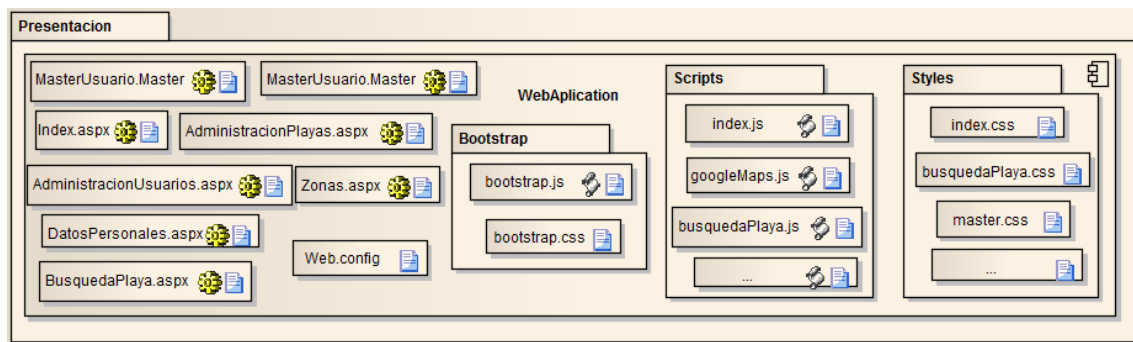


Imagen 25: Vista de presentación

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

- **Site.master:** plantilla 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

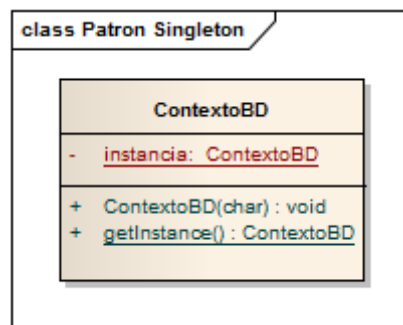


Imagen 26: Patrón Singleton

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

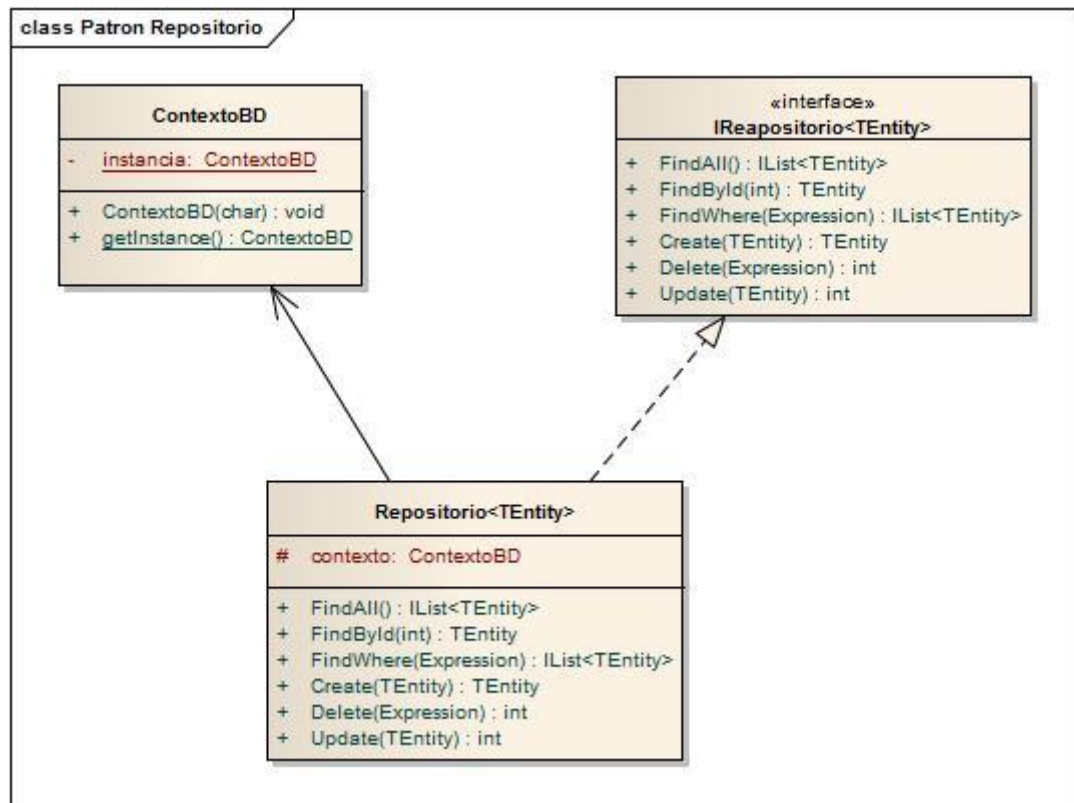


Imagen 27: Patrón Repositorio

JUSTIFICACIÓN

- 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 “CodeFirst”, se podrá utilizar el repositorio para cualquier entidad compleja que se defina.

Informe de ejecución de pruebas

Control de la documentación

Control de la Configuración.

Título:	Informe de ejecución de pruebas
Referencia:	GeoP_Producto_InformeTesting.docx
Autores:	Ignacio Frigerio
Fecha:	15/02/2015

Histórico de Versiones.

Versión	Fecha	Estado	Responsable	Cambios
1.0_DraftA	15/02/2015	Pendiente de Revisión	Ignacio Frigerio [autor]	Se crea el informe de testing
1.0	26/02/2015	Revisado	Leonel Romero	

Introducción

En el siguiente documento se presenta un informe sobre las actividades relacionadas con testing en el proyecto GeoParking, resumiendo las ejecuciones más importantes, los resultados obtenidos y las conclusiones a las cuales nos permitieron alcanzar.

Tipo de Casos de Prueba

Durante el desarrollo del proyecto se realizaron 3 tipos diferentes de casos de prueba los cuales se describen a continuación:

Casos de prueba exploratorios

Se ejecutaron casos de pruebas exploratorios basados en las pruebas de aceptación de las distintas historias, los cuales planteaban un escenario de ejecución de la historia sin especificar con exactitud los pasos a seguir.

Casos de prueba manuales

Se plantearon y modelaron casos de prueba con la intención de ser corridos manualmente, planteando los pasos a seguir y los resultados esperados en cada caso, en las ejecuciones se tuvieron en cuenta dichos resultados y se registraron los resultados obtenidos para comparar con los esperados y así facilitar información sobre el resultado final de caso de prueba

Casos de prueba automáticos

Se plantearon en las primeras etapas del desarrollo la realización de casos de pruebas automáticos, se realizaron ejecuciones automáticas de dichos casos de prueba, pero debido al alto costo de actualización de los mismos, debido a los abundantes cambios sobre las historias se tomó la decisión de no mantener los mismos.

Tipos de ciclos de Ejecución

Ciclo de Sistema

Se ejecutaron los casos de pruebas manuales y exploratorios pertenecientes a cada historia una vez terminada la etapa de desarrollo de la misma, para poder considerar a la historia como terminada.

Ciclo de Regresión

Debido a la realización de un solo reléase del producto, se realizó un solo ciclo de regresión, en el cual se corrieron todos los casos de pruebas que representaban el escenario principal de cada una de las historias.

Ejecuciones de casos de prueba

A continuación se muestran las ejecuciones más significativas de los distintos casos de prueba correspondientes a distintos ciclos de sistema

Historia 3: Administrar Playa de Estacionamiento

Caso de prueba 1:

Nro.	Paso	Valor esperado	14/01/2015	22/01/2015	16/02/2015
			Valor Obtenido	Valor Obtenido	Valor Obtenido
1	Ingresar al menú administración de playas	se carga la sección de administrar playas de estacionamiento	OK	OK	OK
2	Presionar el botón "Agregar"	se cargan los componentes necesarios para registrar una playa de estacionamiento	OK	OK	OK
3	Ingresar <Playa1> en el campo Nombre		OK	OK	OK
4	Ingresar <NumeroTelefono1> en el campo teléfono		OK	OK	OK
5	Ingresar <email1> en el campo email		OK	OK	OK
6	Seleccionar <TipoPlaya1> en la lista desplegable		OK	OK	OK
7	Seleccionar los días de atención <DiaAtencion1>		OK	OK	OK
8	Seleccionar horario de atención <Horario1>		OK	OK	OK
9	Seleccionar <TipoVehiculo1> en la lista desplegable		OK	OK	OK
10	Ingresar <Capacidad1> en el campo capacidad		OK	OK	OK
11	Presionar el botón "Agregar"	Se crea una fila en la tabla precios con el tipo de vehículo <TipoVehiculo1> y la capacidad <Capacidad1>	OK	OK	OK
12	Ingresar <Precio1> en el tipo de tiempo <Tiempo1> y presionar el botón de confirmación		OK	OK	OK
13	Seleccionar la pestaña Dirección	Se muestran los componentes de dirección de playas de estacionamiento	OK	OK	OK
14	Ingresar la ciudad <Ciudad1> en el campo ciudad, y seleccionar la misma del listado de ciudades		OK	OK	OK
15	Ingresar la calle <Calle1> en el campo calle		OK	OK	OK

Nro.	Paso	Valor esperado	14/01/2015	22/01/2015	16/02/2015
			Valor Obtenido	Valor Obtenido	Valor Obtenido
16	Ingresar el numero <Numero1> en el campo numero>		OK	OK	OK
17	Presionar el botón "Ubicar"	Se muestra en el mapa un marcador en la ubicación ingresada	Se muestran múltiples marcadores en el mapa q no corresponden	OK	OK
18	Presionar el botón "Agregar"	Se agrega a la tabla de direcciones una nueva fila con los datos de la dirección ingresada	No ejecutado	OK	OK
19	Presionar el botón Guardar	Se muestra el siguiente mensaje: "La playa de estacionamiento ha sido guardada con éxito"	No ejecutado	OK	OK
20	Verificar que <Playa1> existe en la base de datos y los datos ingresados son válidos.	<Playa1> existe en la base de datos	No ejecutado	OK	OK

Caso de prueba 2:

Nro.	Paso	Valor esperado	14/01/2015	22/01/2015	16/02/2015
			Valor Obtenido	Valor Obtenido	Valor Obtenido
1	Ingresar al menú administrar playas	se carga la sección de administrar playas de estacionamiento	OK	OK	OK
2	Ingreso <Playa1> en el campo nombre de playa		OK	OK	OK
3	Presionar el botón buscar	Aparece una tabla que contiene la playa de estacionamiento <Playa1>	No se encuentra <Playa1> en la tabla	OK	OK
4	Hacer click en el botón modificar de la fila donde se encuentra la playa <Playa1>	se muestra el modal con los datos de la playa <Playa1> y los campos son editables	No ejecutado	OK	OK
5	Ingresar <NuevoNombrePlaya1>" en el campo nombre de playa		No ejecutado	OK	OK
6	Presionar el botón guardar	Se muestra el mensaje "Está seguro que desea guardar los cambios en la playa de estacionamiento <Playa1>?"	No ejecutado	OK	OK
7	Seleccionar la opción "Si"	Se muestra el mensaje "La playa se modificó exitosamente"	No ejecutado	OK	OK

Caso de prueba 5:

Nro.	Paso	Valor esperado	14/01/2015	22/01/2015	16/02/2015
			Valor Obtenido	Valor Obtenido	Valor Obtenido
1	Ingresar al menú administrar playa de estacionamiento	se carga la sección de administrar playas de estacionamiento	OK	OK	OK
2	Ingresar la ciudad <ciudad1> en el campo ciudad	se muestra un listado de ciudades mientras ingreso la ciudad	OK	OK	OK
3	Seleccionar <ciudad1> del listado		OK	OK	OK
4	Presionar el botón buscar	se muestra una tabla con las playas de esa ciudad	No se mostró la tabla con playas de estacionamiento	OK	OK
5	Ingresar "Playa1" en el campo de búsqueda	Aparece una tabla que contiene la playa de estacionamiento <Playa1>	no ejecutado	No se encontró la playa de estacionamiento en la tabla	OK
6	Presionar el botón ver	Se carga un modal con todos los datos de la playa <Playa1> y no me permite modificarlos	no ejecutado	no ejecutado	OK

Historia 2: Consultar playas de estacionamiento en la web

Caso de prueba 7:

Nro.	Paso	Valor esperado	11/10/2014	26/10/2014	16/02/2015
			Valor Obtenido	Valor Obtenido	Valor Obtenido
1	Ingresar a la página principal.		OK	OK	OK
2	Ingresar <Ciudad1> en el campo nombre de ciudad	Se muestra un listado de ciudades que coinciden con <Ciudad1>	OK	OK	OK
3	Seleccionar del listado la ciudad <Ciudad1>		OK	OK	OK
4	Presionar el botón buscar	Se carga la página BusquedaPlayas, con todas las playas de que pertenecen a la ciudad <Ciudad1> disponibles en un mapa. Se carga la información de las playas en la grilla de playas.	No aparecen las playas de la ciudad <Ciudad1>	OK	OK
5	Selecciono <TipoPlaya1> en el campo Tipo de Playa		No Ejecutado	OK	OK
6	Selecciono <TipoVehiculo1> en el campo Tipo de Vehículo		No Ejecutado	OK	OK
7	Ingreso <Precio1> en el campo precio desde		No Ejecutado	OK	OK
8	Ingreso <PrecioHasta1> en el campo precio hasta		No Ejecutado	OK	OK
9	Ingreso <Horario1> en el campo hora desde		No Ejecutado	OK	OK
10	Ingreso <Horario2> en el campo hora hasta		No Ejecutado	OK	OK
11	Selecciono <DíasDeAtencion1> en el campo Días de atención		No Ejecutado	OK	OK
12	Presiono el botón filtrar	Se muestran en el mapa solo las playas que cumplen con los filtros seleccionados	No Ejecutado	No se muestra playas que si cumple los filtros (no se muestra ninguna)	OK

Historia 7: Consultar disponibilidad de playa de estacionamiento

Caso de prueba 30:

Nro.	Paso	Valor esperado	11/10/2014	26/10/2015	16/02/2015
			Valor Obtenido	Valor Obtenido	Valor Obtenido
1	Ingresar a la aplicación GeoParking Móvil	Se carga la aplicación mostrando la ubicación actual del usuario (dispositivo móvil)	OK	OK	OK
2	Presionar el botón "Listar playas"	Se abre un panel en forma de lista de las playas que cumplen con la condición de que la distancia entre la posición actual del vehículo y la playas es menor o igual a <Radio1> y <Playa1> está dentro del listado Y se muestra la disponibilidad <Disponibilidad1> para la playa <Playa1>	No se muestra la disponibilidad de <Playa1>	OK	OK

Historia 13: Consultar disponibilidad de playas de estacionamiento en tiempo real

Caso de prueba 33:

Nro.	Paso	Valor esperado	07/01/2015	15/01/2015	16/02/2015
			Valor Obtenido	Valor Obtenido	Valor Obtenido
1	Ingresar a la aplicación GeoParking Móvil	Se carga la aplicación mostrando la ubicación actual del usuario (dispositivo móvil)	OK	OK	OK
2	Presionar el botón "Listar playas"	Se muestra un listado de playas en el cual se muestra <Playa1> y se muestra la disponibilidad <Disponibilidad1> esa playa	OK	OK	OK
3	Actualizar la disponibilidad de la playa <Playa1> a <Disponibilidad2> y esperar 2 segundos	En el listado de playas la disponibilidad de la playa <Playa1> cambia a <Disponibilidad2>	No cambia la disponibilidad de <Playa1>	OK	OK

Historia 25: Visualizar estadísticas históricas

Caso de prueba 17:

Nro.	Paso	Valor esperado	17/01/2015	05/02/2015	16/02/2015
			Valor Obtenido	Valor Obtenido	Valor Obtenido
1	Ingresar al sitio web.		OK	OK	OK
2	Ingresar a la opción Estadísticas		OK	OK	OK
3	Escribir <Ciudad1> en donde se debe ingresar la ciudad a buscar.	Se muestra el autocomplete con las opciones de las ciudad que se corresponden con <Ciudad1>	OK	OK	OK
4	Seleccionar <Ciudad1> en la lista de ciudad del autocomplete y presiono BUSCAR		OK	OK	OK
5	Presionar el botón Buscar	Se abre un panel para configurar las estadísticas	OK		
6	Presionar el botón "Nueva"	Se despliegan 2 opciones, "Histórica" y "Tiempo Real"	OK	OK	OK
7	Presionar el botón "Histórica"	Se muestra un panel para buscar estadísticas históricas	OK	OK	OK
8	Ingresar Estadística: "Consulta", Por: "Playa", Desde <Fecha1>, Hasta <Fecha2>		OK	OK	OK
9	Presionar el botón "Buscar"	Se despliega el panel de estadísticas, con los filtros y las opciones del grafico	No se muestra el grafico de estadísticas	OK	OK

Conclusión

Estas ejecuciones de casos de prueba nos permitieron analizar las funcionalidades desarrolladas para poder asegurar el nivel pretendido de calidad para los productos desarrollados. Podemos concluir que las actividades de testing realizadas representaron para el equipo un nuevo desafío ya que el mismo nunca había realizado este tipo de actividades con anterioridad. A lo largo del proyecto surgieron complicaciones que nos obligaron a realizar modificaciones a lo establecido en el plan, como por ejemplo la utilización de una nueva plantilla que nos resultó más cómoda para registrar las ejecuciones, la eliminación de los tests automáticos ya que nos consumían mucho tiempo y no veíamos una gran utilidad en los módulos de nuestro sistema que no son de gran complejidad. A pesar de estas dificultades presentadas, el equipo pudo llevar a cabo las actividades de testing dentro de los tiempos previstos para las mismas.

Introducción

En este documento, queremos presentar las historias de usuario al día de la fecha identificadas por el equipo en el rol de Product Owner colectivo que definimos en la presentación del proyecto.

Cada historia contendrá información acerca de que componentes fueron creados y utilizados para cumplimentarla, refiriéndonos a componentes a software y a modelados.

Dicha trazabilidad a través del diseño y de la arquitectura del sistema, nos permite saber en qué componentes tendrá impacto un determinado cambio a implementar.

Atte. El Equipo de GeoParking

Matriz de trazabilidad

Control de la documentación

Control de la Configuración.

Título:	Matriz de trazabilidad
Referencia:	GeoP_Proyecto_Matriz_Trazabilidad.docx
Autores:	Lucas Toneatto
Fecha:	10/02/2015

Histórico de Versiones.

Versión	Fecha	Estado	Responsable	Cambios
1.0_DraftA	10/02/2015	Pendiente de Revisión	Lucas Toneatto [autor]	
1.0	15/02/2015	Aprobado	Ignacio Frigerio	

MATRIZ DE TRAZABILIDAD

ID	HISTORIA	ENTIDADES/WIDGETS	GESTORES/WIDGETS	CONTROLADORES	PAGINAS/PAGE	MODELADO
1	Consultar playas de estacionamiento en el móvil (MAPA)	Mapa	CargarMapa.js ObtenerPlayas.js		mainPage	<u>Modelo de clases:</u> DiagramaDeClases-movil <u>Vista dinámica:</u> BuscarPlayas
2	Consultar playas de estacionamiento en la web	PlayaDeEstacionamiento TipoVehiculo TipoPlaya DiasAtencion	GestorBusquedaPlayas		Index.aspx BusquedaPlaya.aspx	<u>Vista Dinámica:</u> BusquedaDePlayaPorCiudad BusquedaDePlayaPorCiudadPorFiltros <u>Modelo De Clases:</u> Clases de Negocio Gestores
3	Administrar playas de estacionamiento	PlayaDeEstacionamiento TipoPlaya TipoVehiculo DiasAtencion Tiempo	GestorPlaya		AdministracionPlayas.aspx	<u>Vista Dinámica:</u> RegistroDePlaya <u>Modelo De Clases:</u> Clases de Negocio Gestores
4	Almacenar ubicación de vehículo		ObtenerRutaDeRegresos cargarMapa.js Utils.js		mainPage	<u>Modelo de clases:</u> Diagrama De Clases - movil <u>Vista dinámica:</u> GuardarPosicionVehiculo
5	Consultar ruta a playa de estacionamiento	Playa Ruta	cargarMapa.js Utils.js	EstadisticasControlleur.cs	mainPage	<u>Modelo de clases:</u> Diagrama De Clases - movil <u>Vista dinámica:</u> Movil_irAplaya+guardarPlaya+enviarEstadisticas
6	Consultar playas de estacionamiento cercanas a un punto de interés	Playa Ruta Place	cargarMapa.js Utils.js autocomplete.js		mainPage	<u>Modelo de clases:</u> Diagrama De Clases - movil <u>Vista dinámica:</u> Movil_irAplaya+guardarPlaya+enviarEstadisticas
7	Consultar disponibilidad de playa de estacionamiento					
8	Actualizar la disponibilidad de una playa de estacionamiento	PlayaDeEstacionamiento DisponibilidadPlayas HistorialDisponibilidadPlayas TipoVehiculo	GestorDisponibilidad	DisponibilidadController		<u>Modelo De Clases:</u> Clases de Negocio Gestores
9	Consultar ruta a posición almacenada de vehículo	localStorage[vehiculo]	obtenerRutaDeRegreso.js Utils.js		mainPage	<u>Modelo de clases:</u> Diagrama De Clases - movil <u>Vista dinámica:</u> Movil_TrazarRegresoAVehiculo
10	Almacenar playas de estacionamiento consultadas	Playa localStorage[playas]	Utils.js historialWidget.js		mainPage	<u>Modelo de clases:</u> DiagramaDeClases-movil <u>Vista dinámica:</u> Movil_irAplaya+guardarPlaya+enviarEstadisticas
11	Administrar configuración de dispositivo	localStorage[configuraciones]	preferencesWidget.js		preferencesPage	<u>Modelo de clases:</u> DiagramaDeClases-movil <u>Vista dinámica:</u> Movil_GuardarConfiguracion
12	Utilizar comando por voz para localización					
13	Consultar disponibilidad de playas de estacionamiento en tiempo real	DisponibilidadPlayas	GestorDisponibilidad GestorEstadisticas			
14	Crear reportes estadísticos (EPICA)					

15	Gestionar permisos de usuarios	Usuario TipoUsuario Rol Permiso	GestorRol GestorUsuario		AdministracionDeRolesyPermisos.aspx	<u>Vista Dinámica:</u> AdministracionDeUsuarios <u>Modelo De Clases:</u> Clases de Negocio Gestores
16	Registrar usuario	Usuario	GestorUsuario		MasterAdmin.aspx MasterUsuario.aspx	<u>Vista Dinámica:</u> RegistroDeUsuario <u>Modelo De Clases:</u> Clases de Negocio Gestores
17	Panel de Usuario	Usuario	GestorUsuario		DatosPersonales.aspx	
18	Consultar playas de estacionamiento en el móvil (Listado)	Playa Ruta	GestorDisponibilidad GestorPrecio RepositorioDisponibilidad Playas RepositorioPrecios	PlayasControler DisponibilidadControler	mainPage/listadoPlayas	<u>Modelo de clases:</u> DiagramaDeClases-movil <u>Vista dinámica:</u> Movil_ListadoDePlayasConDisponibilidad
19	Configurar Servidor					
20	Consultar historial de playas				mainPage/pnlPrincipal	<u>Vista dinámica:</u> Movil_irAplaya+guardarPlaya+enviarEstadísticas
21	Gestionar permisos de roles	Rol Permiso	GestorRol		AdministracionDeRolesyPermisos.aspx	<u>Modelo De Clases:</u> Clases de Negocio Gestores GestionarPermisosDeUSuarios
22	Creación de roles	Rol	GestorRol		AdministracionDeRolesyPermisos.aspx	<u>Modelo De Clases:</u> Clases de Negocio Gestores GestionarPermisosDeUSuarios
23	Administración de zonas	PalyaDeEstacionamiento TipoVehiculo TipoPlaya Zona DisponibilidadPlayas	GestorEstadísticas GestorPlaya GestorTiposVehiculo GestorZonas		Estadisticas.aspx	<u>Modelo De Clases:</u> Gestores
24	Visualizar estadísticas en tiempo real	DisponibilidadPlayas	GestorEstadísticas		Estadisticas.aspx	<u>Modelo De Clases:</u> Gestores
25	Visualizar estadísticas históricas	HistorialDisponibilidadPlayas	GestorEstadísticas		Estadisticas.aspx	<u>Modelo De Clases:</u> Gestores
26	Cambiar tipos de gráficos				Estadisticas.aspx	<u>Modelo De Clases:</u> Gestores
27	Filtrar estadísticas	TipoVehiculo TipoPlaya Zona	GestorTiposVehiculo GestorZonas GestorTipoPlaya		Estadisticas.aspx	<u>Modelo De Clases:</u> Gestores
28	Comparar estadísticas				Estadisticas.aspx	<u>Modelo De Clases:</u> Gestores