Universidad Internacional de la Rioja

Maestría en Dirección e Ingeniería de Sitios Web

Desarrollo de la Internet de las Cosas

**Alfredo Jiménez Miguel**

**Elizabeth Susana Velázquez Zamora**

Actividad

Android Básico

Profesor: Israel Sandoval Grajeda

Ciudad de México, 13 de agosto de 2019.

Android

Introducción

Basado en el documento anterior “Herramientas de la IoT”, este trabajo muestra la implementación de la aplicación Android desarrollada para

"**Saa**" (***Space******always******available***), la cual funge como administrador de un estacionamiento privado. En este desarrollo se pretendió acercarse lo más posible a la definición planteada.

Adicional a este documento, se entrega código de desarrollo de **GitHub** y aplicación **APK**.

De acuerdo con la entrega anterior las funciones planeadas para el desarrollo fueron:

**Funciones de la aplicación.**

1. Iniciar sesión para acceder a la plataforma e identificar a un usuario previamente registrado.
2. En la página de inicio se mostrará el código de barras que identifica al usuario y le permite la entrada y salida del estacionamiento.
3. En la página de inicio también se ofrecen las opciones de:
   1. Encontrar un estacionamiento, con la opción de sugerir el más cercano si el usuario permite conocer su ubicación, esta opción es útil cuando el usuario aún no se encuentra en el estacionamiento.
   2. Administrar los métodos de pago.
      1. Guardar un nuevo método de pago.
      2. Eliminar algún método de pago.
      3. Revisar historial de pagos.
   3. Cerrar sesión.
4. Después de haber encontrado el estacionamiento más cercano con la opción de “Encuentra tu estacionamiento”, se podrá consultar su disponibilidad y/o realizar una **reserva anticipada**.
5. En caso de realizar una reserva anticipada\* o de haber ingresado físicamente\*:
   1. Se comienza una sesión de pago.
   2. Se realiza la asignación de la plaza más cercana, con la opción de acercar al usuario al lugar que visita o por alguna condición especial.
6. Cuando se tiene una sesión de pago activa, se podrá:
   1. Consultar el tiempo consumido y saldo estimado en cualquier momento.
   2. Pagar, con la opción de visualizar tu recibo.

Mapa de navegación

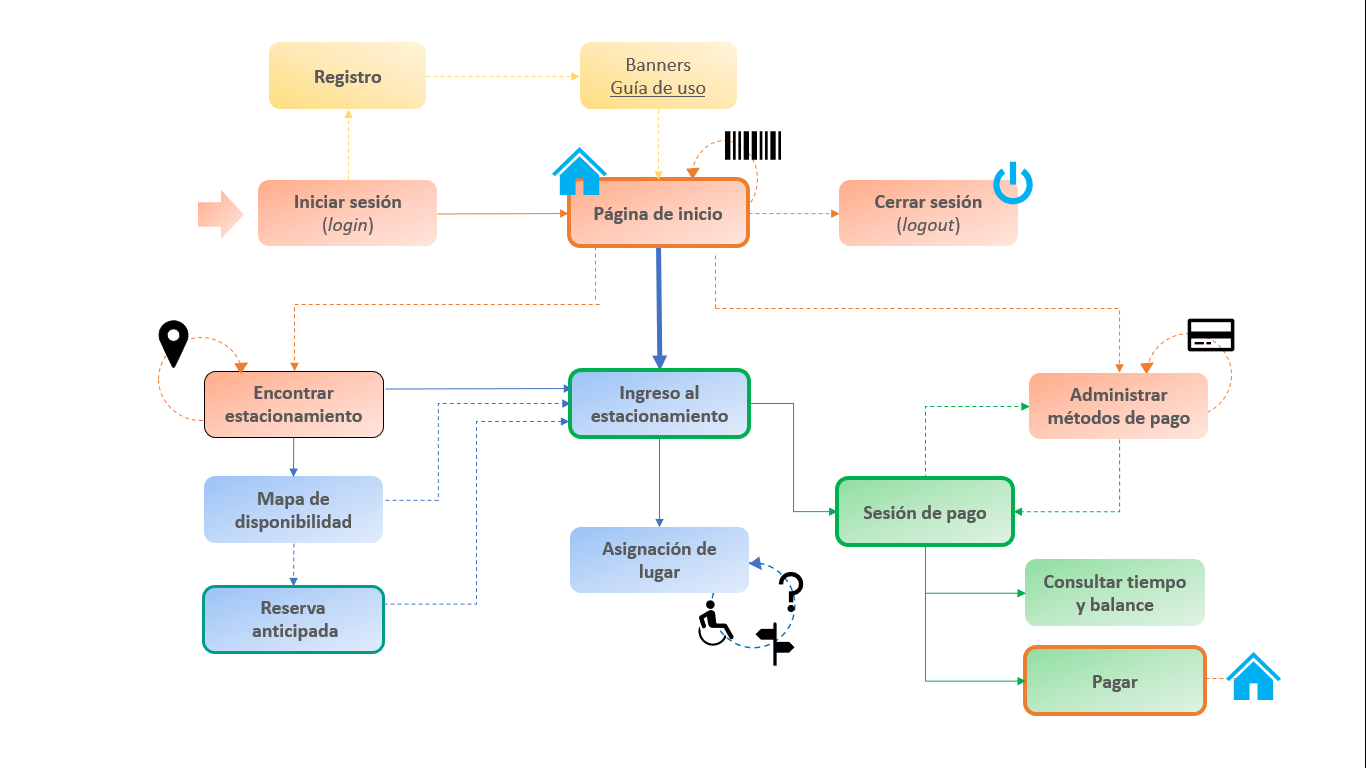
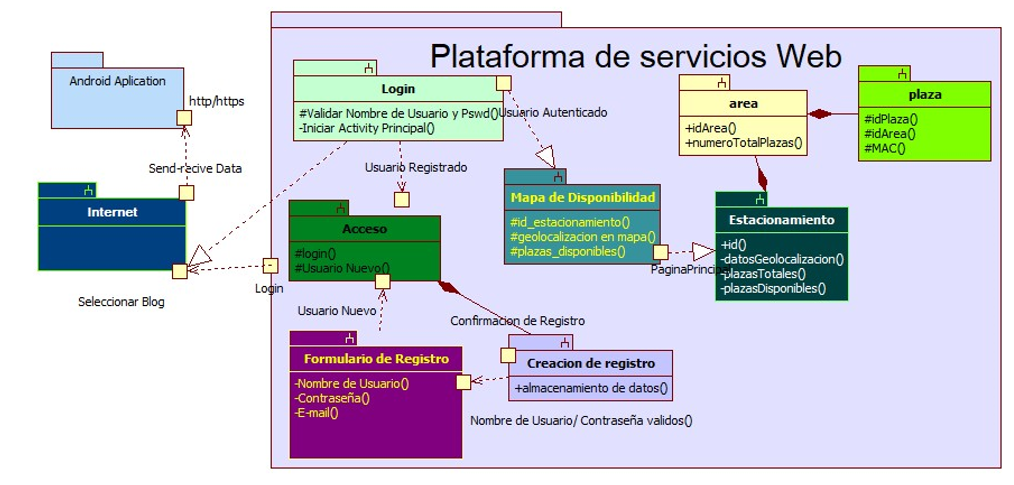


Figura 1. Mapa de navegación

Diagrama de componentes



*Figura 2. Diagrama de componentes*

**Alcance y acotamiento técnico**

1. Se desarrolló con *Android Studio* una aplicación móvil lo más apegado posible con las funciones descritas en la entrega anterior.
2. Se implementó *login*, *registro, geolocalización ,actividades gestión de pagos y menú principal.*
3. Debido al tiempo limitado de desarrollo:
   1. Se simularon los lugares ocupados, ya que no se contará con los dispositivos físicos.
   2. No se implementaron los métodos de pago.
4. Se creó el servicio web y es funcional bajo la dirección:

[http://www.diot.esy.es](http://www.diot.esy.es/)

A esta dirección las aplicaciones realizarán las peticiones necesarias para su funcionamiento.

Los API’s integrantes funcionan bajo el paradigma REST, realizando intercambio de datos en JSON, la interacción con base de datos, se realiza a través del Gestor My Sql, en el servidor la gestión se realiza por medio de PHP My Admin.

La comunicación, así como la serialización y deserialización en la App Android se realiza a través de la biblioteca Volley creada por colaboradores de Google, bastante sencilla de implementar y usar.

Para el listado de estacionamientos se utiliza el componente ListView-Entity-Adapter, que permite agregar Arrays y ArrayList.

Se implementó el uso de fragments para como una mejor opción de la metodología de componentes, que nos permitirá realizar entregas de prototipos.

El repositorio del Web service:

<https://github.com/ajbeem/iotserver>

Por otro lado, el repositorio para el proyecto SAA de Android:

<https://github.com/ajbeem/saa>

Pantallasde la aplicación

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |

Conclusiones

Como se menciona en la entrega anterior, este proyecto ha servido a nivel didáctico para dimensionar el esfuerzo que se tiene que realizar para la implementación de este tipo de propuestas, a nivel técnico, en esta entrega, como equipo nos pudimos percatar que, por la curva de aprendizaje y tiempos limitados de desarrollo, la implementación se limitó a solo una parte de lo esperado, sin embargo, se cumplió con los requisitos de la actividad y se planteó el escenario ideal para un proyecto de este tipo.

La revista digital Software Gurú (2017), indicaba que sólo el 35% de proyectos satisfacían las expectativas de los clientes que ordenaron el sistema desarrollado, en tiempo, forma y calidad, a mi me parecía una exageración, sin embargo, reflexionando un poco, determiné que uno de los puntos más sensibles y tal vez el más común son los tiempos de desarrollo, recuerdo que una de las métricas más utilizadas en la planeación de proyectos, era calcular el tiempo de LOC (Lines of code), suponiendo a este como una constante, lo que es inviable, escribir código como escribir una carta ni siquiera es posible a nivel Programador Senior, con años de experiencia, todo desarrollo significa una infraestructura totalmente nueva, si bien la reutilización de código es una realidad, la implementación en ocasiones no es inmediata, lo que lleva a la revisión en pruebas de caja blanca con los consecuentes retrasos, pienso que todo proyecto debería planearse con suficientes recursos tanto de tiempo como humanos, materiales, técnicos y monetarios, debido a que cualquiera de ellos, insuficiente, podría causar consecuencias de leves a graves.

Un jefe de proyecto experimentado tendrá una idea bastante cercana a los recursos necesarios y mencionados previamente, pero no debería aventurarse a proyectar tiempos, prototipos, fases o hitos casi irreales, desafortunadamente es más común de lo que desearíamos, ocasionando encono con los clientes y fatiga excesiva del equipo de trabajo.