**Documento de Arquitectura de Software**

**IEEE-1471-2000**

******

**Arquitectura**

**Del Software**

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre del proyecto: | *Smart Parking* | | |
| Alias del proyecto: | *Smart\_Parking\_App* | | |
| Versión: | *1.0* | Fecha última modificación: | *09/06/2017* |

Contenido

[1. Introducción 3](#_Toc484947473)

[1.1 Propósito 3](#_Toc484947474)

[1.2 Alcance 3](#_Toc484947475)

[2. Referencias 3](#_Toc484947476)

[3. Definiciones, acrónimos y abreviaciones. 3](#_Toc484947477)

[4. Framework Conceptual 4](#_Toc484947478)

[4.1 Descripción de la arquitectura en contexto 4](#_Toc484947479)

[4.2 Roles 4](#_Toc484947480)

[4.2 Identificación de los roles y sus responsabilidades 5](#_Toc484947481)

[4.3 Selección de puntos de vista de la arquitectura. 5](#_Toc484947482)

[4.4 Vistas de la arquitectura 6](#_Toc484947483)

[4.4.1 Vista Lógica 6](#_Toc484947484)

[4.4.2 Vista Despliegue 8](#_Toc484947485)

[4.4.2.2 Paquetes 9](#_Toc484947486)

[4.4.4Vista Físico 11](#_Toc484947487)

[4.5 DESCRIPCIÓN DE MÓDULOS 13](#_Toc484947488)

[4.5.1 Características 13](#_Toc484947489)

[4.5.2 Tickets 13](#_Toc484947490)

[4.5.3 Información 13](#_Toc484947491)

[4.5.4 Mapa 13](#_Toc484947492)

[4.5.5 Llamar 13](#_Toc484947493)

[4.5.6 Saldo 13](#_Toc484947494)

[4.5.7 Vigilante 14](#_Toc484947495)

[4.5.8 Administrador 14](#_Toc484947496)

[4.6 Lenguajes y plataformas 14](#_Toc484947497)

[4.7 Parquímetro 15](#_Toc484947498)

[4.7.1 Componentes del parquímetro. 15](#_Toc484947499)

[4.8 Comunicación de los elementos 16](#_Toc484947500)

# 1. Introducción

## 1.1 Propósito

Este documento es una descripción grafica de la arquitectura de software del proyecto “Smart Parking”, mostrando los diagramas que describen el software.

## Alcance

El presente documento describe el análisis realizado a los requisitos de la aplicación Smart Parking representándolos por medio de los diagramas que se incluyen en el modelo de cuatro vistas más uno.

# 2. Referencias

- *IEEE 830-1998 ST*

- *The 4+1 View. Kruchten - 1009*

# 3. Definiciones, acrónimos y abreviaciones.

DAS: Documento de Arquitectura de Software

HTTP: Protocolo de Trasferencia de Hipertexto.

TCP: Protocolo de control de transmisión.

ARQUITECTURA DE SOFTWARE: conjunto de elementos estáticos, propios del diseño intelectual del sistema, que definen y dan forma tanto al código fuente, como al comportamiento del software en tiempo de ejecución. Naturalmente este diseño arquitectónico ha de ajustarse a las necesidades y requisitos del proyecto.

DESCRIPCIÓN DE ARQUITECTURA: colección de productos de documentación. VISTAS: es una representación de un área de interés o perspectiva del sistema en alto nivel.

TIPOS DE VISTAS: especificación de una convención de cómo construir y usar una vista. Deben satisfacer la capacidad de creación y análisis de una vista.

ROLES: Individuo, equipo u organización con intereses relativos al sistema.

MODULO O COMPONENTE: cualquier elemento estructural abstracto, visible, externo, de alto nivel, analizable, que pueda constituir una funcionalidad de la solución del sistema.

ATRIBUTOS DE CALIDAD: un atributo de calidad, es una cualidad deseable de la solución, que pueda manifestarse en forma de requerimiento no funcional, que pueda ser medible, testeable y finalmente evaluable.

# 4. Framework Conceptual

## 4.1 Descripción de la arquitectura en contexto

Este documento presenta la arquitectura como una serie de vistas basadas en la arquitectura de software del modelo 4+1 de Kruchten. Estas vistas son: la vista de escenarios, la vista lógica., la vista de desarrollos, la vista física, la vista de procesos. No hay ninguna vista separada de una misma implementación, descrita en este documento. Estas vistas están hechas sobre Lenguaje de modelo unificado (UML) en su versión 2.0 desarrolladas usando StarUml y Dia.

## 4.2 Roles

Este documento presenta los roles necesarios para la interacción de desarrollo del software del proyecto Smart Parking.

## 4.2 Identificación de los roles y sus responsabilidades

|  |  |
| --- | --- |
| Rol | Integrantes |
| Programador | Alejandro Martínez Valdés  Alan Hernán Ramírez García |
| Analista | Daniel Isaac Sosa Rincón |
| Diseñador | Ricardo Peralta Atlahua  Cristopher Mario Ruiz Cruz  Pedro Hernández Hernández |
| Tester | Daniel Isaac Sosa Rincón |

|  |  |
| --- | --- |
| Rol | Integrantes |
| Programador | Construcción de prototipos. Colaboración en la elaboración de las pruebas funcionales, modelo de datos y en las validaciones con el usuario. |
| Analista | Analista-Diseñador: Captura, especificación y validación de requisitos interactuando con el cliente mediante entrevistas. |
| Diseñador | Elabora el modelo de análisis y diseño. Colaboración en la elaboración de pruebas funcionales y el modelo de datos. |
| Tester | Evalúa el sistema ante las posibles situaciones que se pueden presentar en el ambiente real del sistema. |

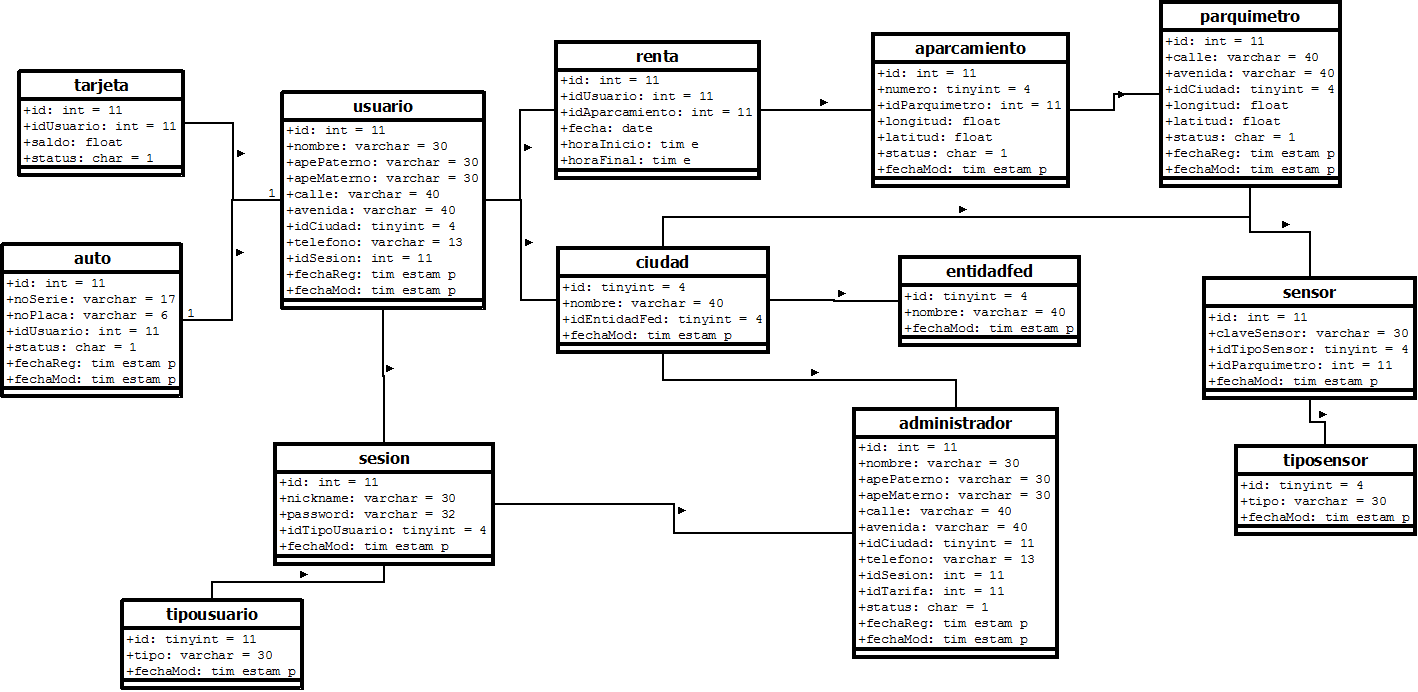
## 4.3 Selección de puntos de vista de la arquitectura.

|  |  |
| --- | --- |
| **Vistas** | **UML** |
| Lógica | Clases, comunicación y secuencia |
| Despliegue | Components y paquetes |
| Proceso | Actividad |
| Físico | Despliegue |
| Escenarios | Casos de uso |

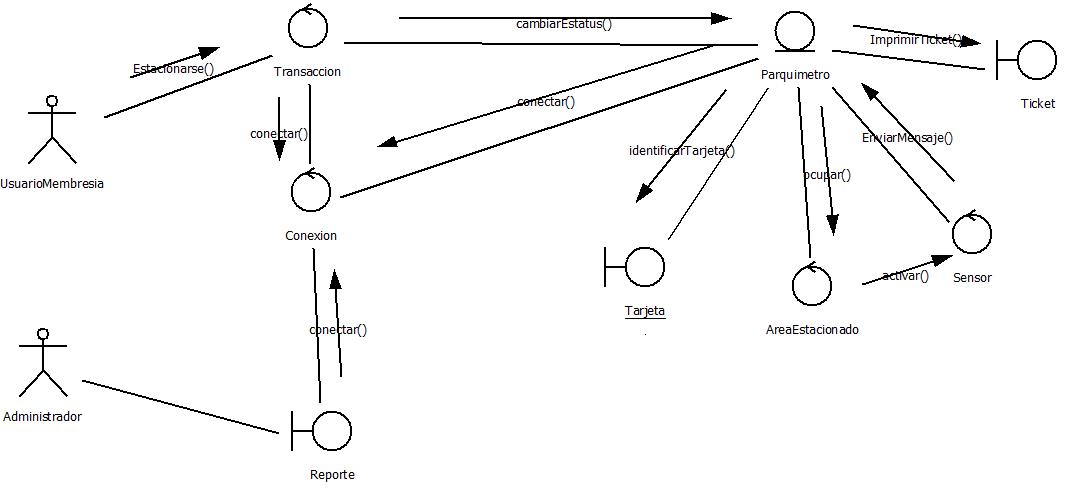
## Vistas de la arquitectura

## 4.4.1 Vista Lógica

#### Clases

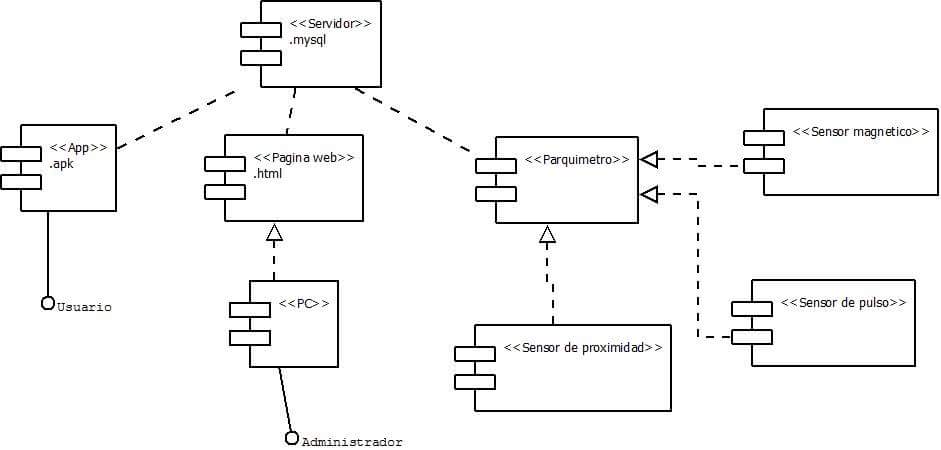
****

#### 4.4.1.2 Comunicación

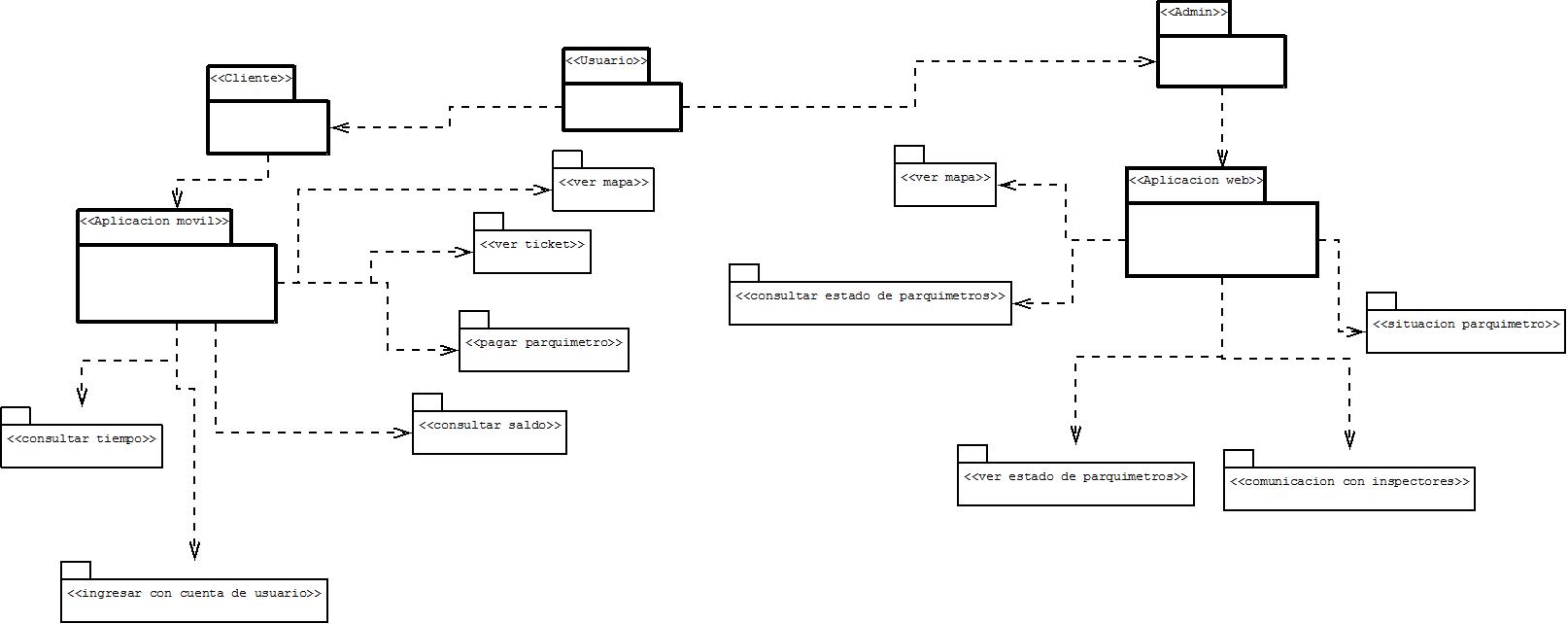


### 4.4.2 Vista Despliegue

#### 4.4.2.1 Componentes

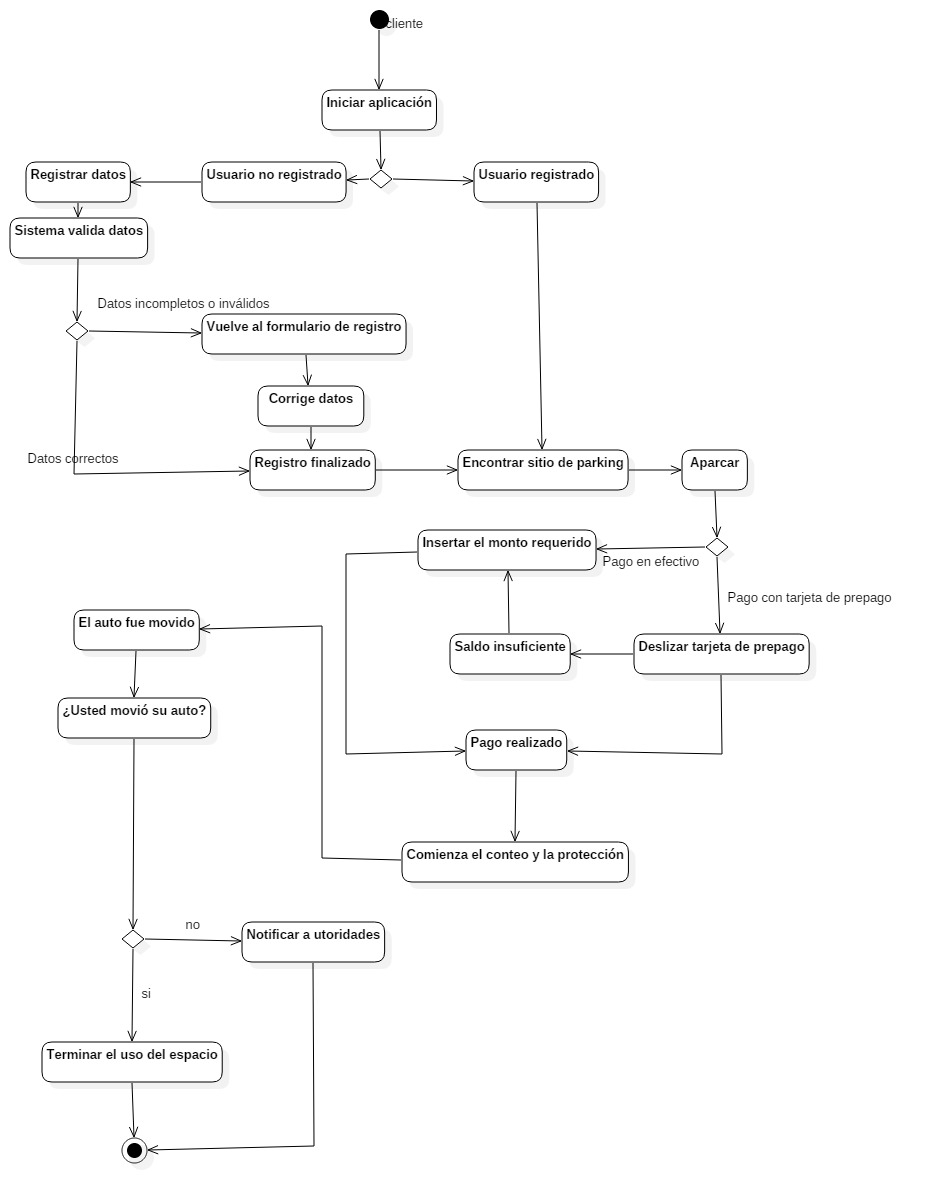


### 4.4.2.2 Paquetes



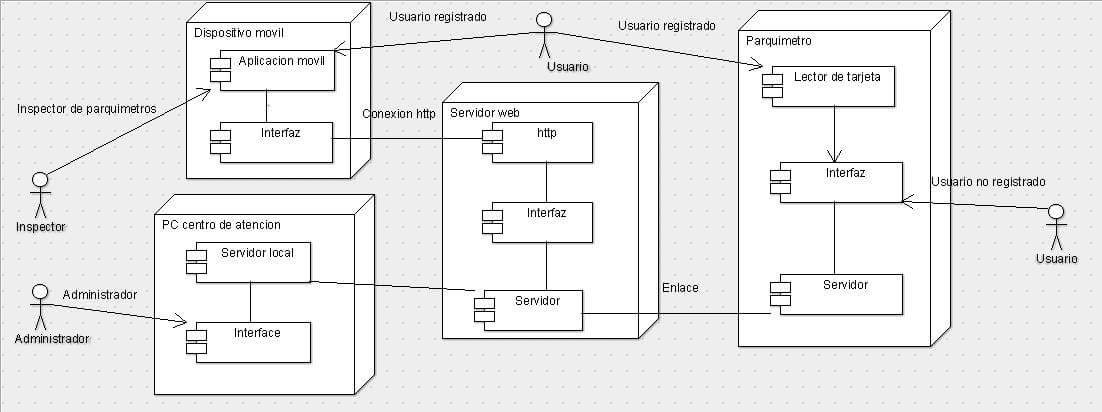
#### 4.4.3 Vista Proceso

#### 4.4.3.1 Actividad



### 4.4.4Vista Físico

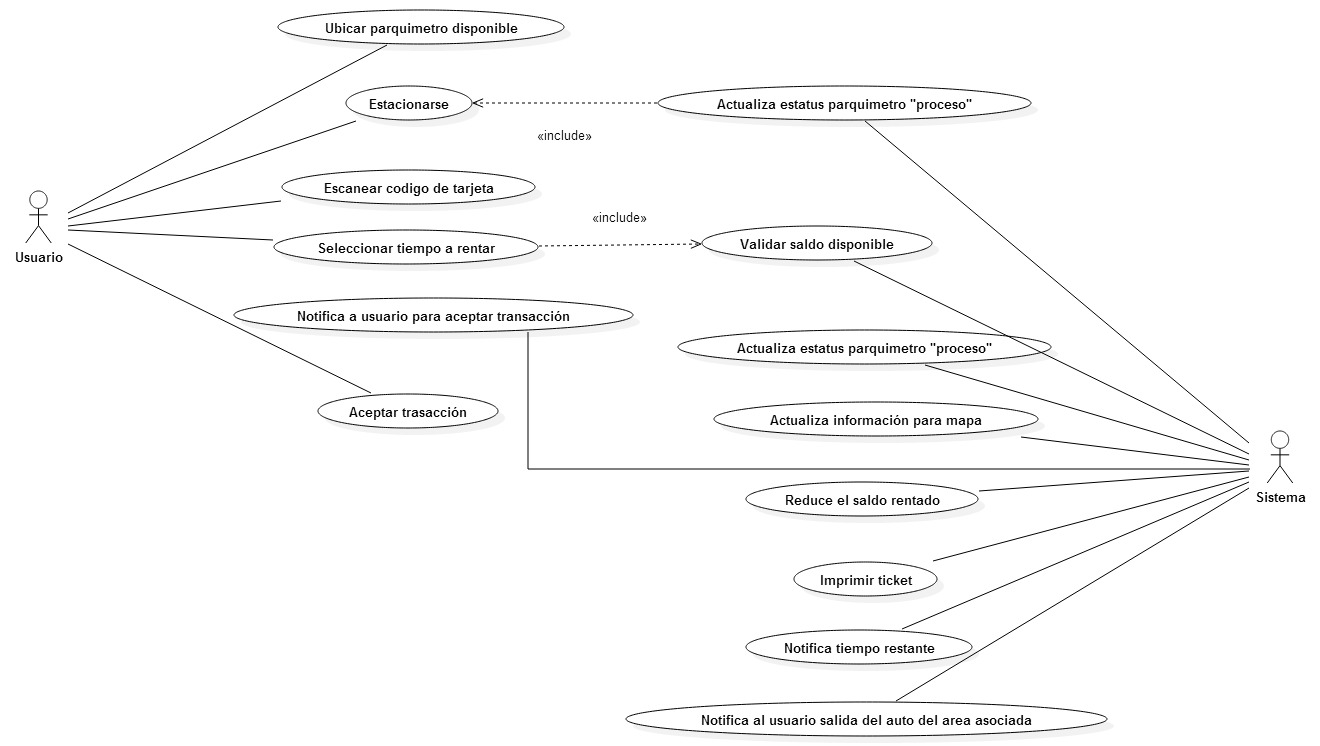
#### 4.4.4.1 Despliegue



#### 4.4.5 Vista Escenario

#### 4.4.5.1 Casos de uso renta de tiempo sin aplicación móvil

#### 4.4.5.2 Caso uso renta de tiempo con aplicación móvil



#### C:\Users\alejandro\AppData\Local\Microsoft\Windows\INetCacheContent.Word\14881390_1223787947681696_233269567_o (1).jpg4.4.5.3 Caso de uso visualización del vigilante

## 4.5 DESCRIPCIÓN DE MÓDULOS

### 4.5.1 Características

Este módulo contiene la información de usuario, sus datos generales (nombre, apellido materno, apellido paterno, edad, RFC), datos del auto (marca, modelo, placas), así como también contiene las opciones para modificar dicha información.

### 4.5.2 Tickets

Este módulo contiene la información del ticket generados, pero gráficamente, con la finalidad de tener una prueba de la renta realizada, así como también presentar un historial de uso del parquímetro.

### 4.5.3 Información

Este módulo contiene la información de la renta que se ha solicitado, los datos correspondientes, son la ubicación del espacio reservado y tiempo restante.

### 4.5.4 Mapa

Este módulo contiene un mapa con la ubicación de todos los parquímetros, se identificarán por colores, verde denota que el parquímetro está disponible, rojo denota que el parquímetro está ocupado, naranja indica que ese parquímetro está en proceso de ser ocupado o desalojado y azul que es una de las características que permiten a usuarios discapacitados ubicar un espacio de estacionamiento más rápidamente.

### 4.5.5 Llamar

Este módulo contiene los números de seguridad para realizar algún tipo de reporte a alguna autoridad según sea el caso, permite acceder directamente a realizar la llamada.

### 4.5.6 Saldo

Este módulo contiene la información del saldo disponible, y los lugares disponibles para recargas del mismo.

### 4.5.7 Vigilante

Este módulo presenta un mapa que contiene los parquímetros que están en uso desocupados o los que están por desocupar, se mostraran por colores, y se ubicara a aquellos usuarios que han excedido el tiempo rentado y necesitan realizar alguna multa.

### 4.5.8 Administrador

Este módulo contiene un formulario para agregar a todos los usuarios que interactúan con el sistema, además de un mapa que contendrá la ubicación y estatus de los parquímetros de la zona.

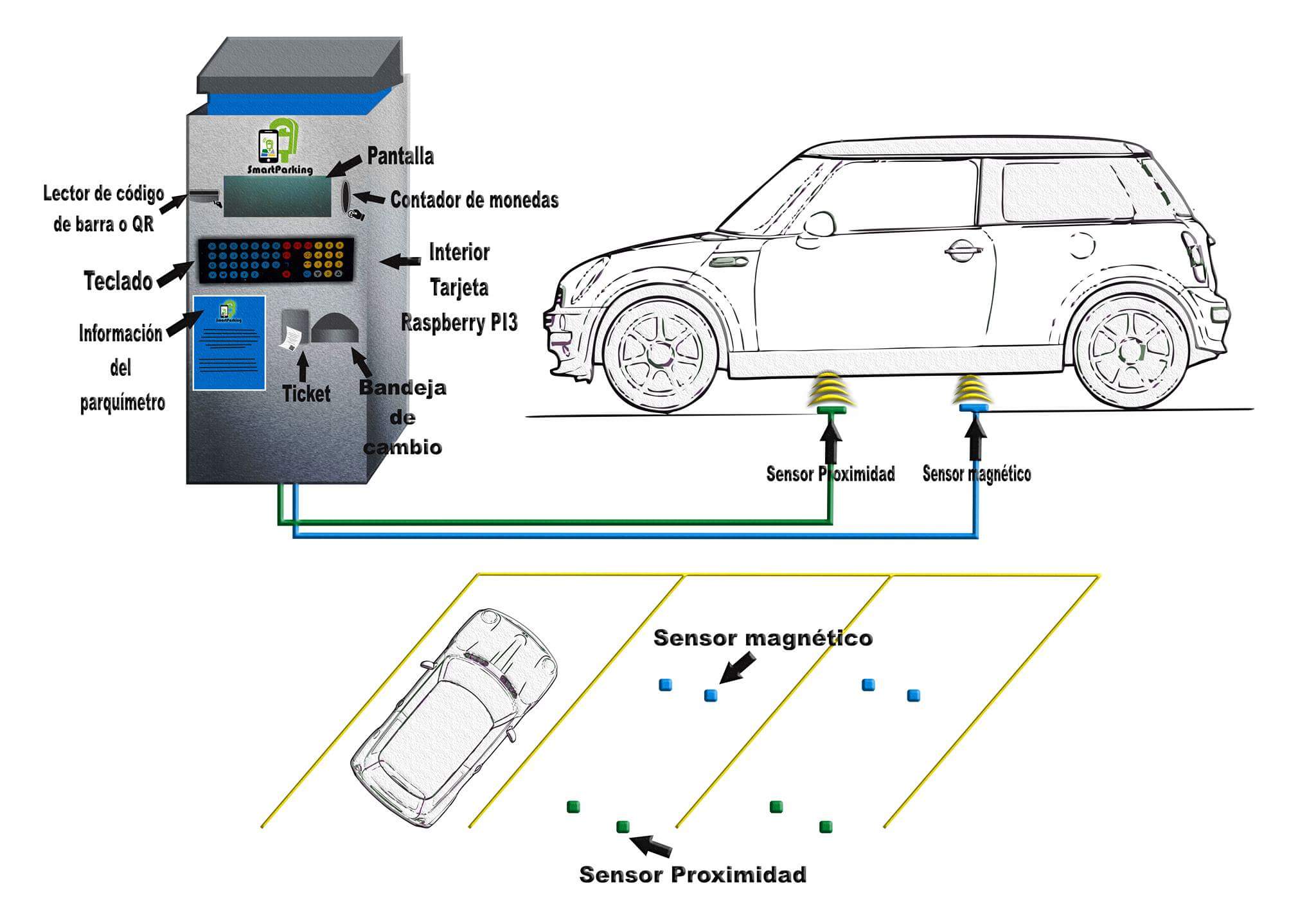
## 4.6 Lenguajes y plataformas

La estructura del diseño del software esta propuesto para utilizar lenguajes, PHP, HTML5, CSS3, JAVASCRIPT, los cuales son compatibles con la funcionalidad de la app para Android y la página web de administración de la información, para la gestión de la información se utilizar un gestor de base de datos, MYSQL el cual almacenara la estructura relacional de la información en una base de datos.

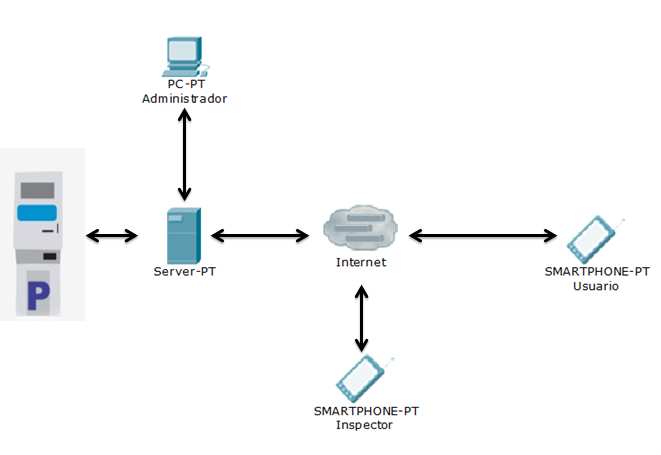
# 4.7 Parquímetro

## 4.7.1 Componentes del parquímetro.





# 4.8 Comunicación de los elementos

******