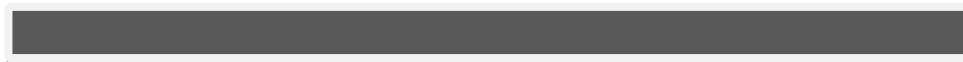


# Documento de requerimientos de software

Parkimaniacos Fecha: [12/14/2020]



## Tabla de contenido

Historial de Versiones.....	3
Información del Proyecto .....	3
Aprobaciones .....	3
1. Propósito .....	4
2. Alcance del producto / Software.....	4
3. Referencias.....	5
4. Funcionalidades del producto .....	7
5. Clases y características de usuarios .....	7
6. Entorno operativo.....	7
7. Requerimientos funcionales.....	8
7.1 Registro e inicio en la aplicación móvil.....	8
7.2 Funcionamiento con Temporizador y lector QR.....	9
7.2 Capacidad para poder interactuar con diferentes sistemas.....	10
8. Reglas de negocio .....	11
9. Requerimientos de interfaces externas.....	12
9.1. Interfaces de usuario .....	12
9.2. Interfaces de hardware .....	12
9.3. Interfaces de software .....	13
9.4. Interfaces de comunicación.....	13
10. Requerimientos no funcionales.....	14
11. Otros requerimientos.....	16
12. Glosario.....	16





## Historial de Versiones

Fecha	Versión	Autor	Organización	Descripción
10/08/2020	1	José Ángel	Maniacorp	Inicio del Documento
11/08/2020	2	José Baltazar	Maniacorp	Ajustes Generales
18/08/2020	3	José Baltazar	Maniacorp	Corrección de los errores
12/14/2020	4	José Ángel	Maniacorp	Actualizar logotipos

## Información del Proyecto

Empresa / Organización	Maniacorp
Proyecto	Parkimaniacos
Fecha de preparación	2/10/2020
Cliente	El
Patrocinador principal	
Gerente / Líder de Proyecto	José Ángel García Arce
Gerente / Líder de Análisis de negocio y requerimientos	José Baltazar de la Rosa Martínez

## Aprobaciones

Nombre y Apellido	Cargo	Departamento u Organización	Fecha	Firma
José Ángel García Arce	CEO	Presidente de Maniacorp	10/8/2020	
Luis Emmanuel Méndez Barrios	Gerente del Área de Desarrollo y Ventas	Desarrollo y Ventas	10/8/2020	



## 1. Propósito



El proyecto, como se puede apreciar en la figura 1.0, tiene como nombre Parkimaniacos, lo cual hace referencia al concepto del proyecto está desarrollando con el nombre de la empresa Figura 1.0 Maniacorp.

Actualmente estamos en la versión 1.1 del proyecto, contando con diversas mejoras al anterior diseño del mismo.

Este documento se compone de la totalidad del proyecto, incluyendo el hardware que será utilizado para la correcta administración de los parquímetros, esperamos que este documento cumpla todas las necesidades del proyecto en general.

## 2. Alcance del producto / Software

Con este proyecto se pretende que cualquier persona que hace uso de los parquímetros de manera regular logre tener acceso a una opción practica y sencilla de utilizar los parquímetros, se espera tener un alcance amplio.

El tipo de clientes a los que está dirigido son: Sector de gobierno o cualquier empresa que necesite la administración de su estacionamiento. El usuario final será a quien consideramos para lograr una aplicación amigable con el usuario, el proyecto tendrá un amplio alcance , de lo descrito anteriormente, también se busca que aparte del uso que le pudiera dar el sector poblacional a la aplicación Parkimaniacos, las autoridades correspondientes también tendrán un papel importante en el correcto uso de los parquímetros ya que ellos de la mano con la aplicación en cuestión van a poder mejorar el uso de los parquímetros, y a su vez monitorear las horas en las cuales haya más flujo vehicular en ellos.



### 3. Referencias

#### El alcance y estudio en la demanda

- Número y ubicación de cajones disponibles en la vía pública.

Se contabilizan todos aquellos lugares formalmente establecidos en la calle para ser ocupados como estacionamiento en el área de intervención.

- Número y ubicación de cajones disponibles fuera de la calle.

Se contabilizan los cajones ubicados en residencias particulares, edificios de oficinas, equipamiento y estacionamientos públicos con tarifa.

- Rotación media de los cajones.

Se refiere al número de veces que se utiliza un cajón en la vía pública a lo largo de un día y de una semana. El estudio debe enfatizar el promedio de rotación que se produce en horas pico.

- Estancia promedio o duración promedio de la rotación.

A lo largo de todo el día y la semana. Hay que poner especial atención en los promedios de rotación en horas pico.

- Tasa de ocupación o intensidad de uso.

Se debe identificar el porcentaje de cajones que permanece ocupado a distintas horas y a lo largo de toda la semana.

- Cajones reservados

Contabiliza y ubica aquellos espacios que son reservados de manera oficial (servicios públicos, embajadas, vehículos de emergencia, sitios de taxi) e informal (a través de la colocación de diversos objetos para impedir su uso).

- Estacionamiento ilegal.

Hay que identificar el porcentaje de automóviles estacionados en sitios no formalmente habilitados para ello (doble fila, banqueta, cruceros, frente a rampas para personas con discapacidad, etc.). La información debe ir acompañada de un plano en el que se señalen aquellos lugares donde estas prácticas se producen con mayor frecuencia.



Aquí podemos ver el muestreo de cómo se organizan las áreas el amarillo el azul y rosa aquí como podemos ver la mayor concentración es en rosa por lo tanto se aplicará un estudio más fondo para saber el promedio de la hora que se pueda cobrar Referencias

- [1] G. d. Guadalajara, «Parquímetros,» 8 septiembre 2017. [En línea]. Available: <https://guadalajara.gob.mx/tags/parquímetros> . [Último acceso: 13 septiembre 2020].
- [2] I. TORREON, «<http://www.trcimplan.gob.mx/>,» 13 Septiembre 2019. [En línea]. Available: <http://www.trcimplan.gob.mx/blog/menos-estacionamiento-masespaciopublico-decalidad.html> . [Último acceso: 13 septiembre 2020].
- [3] I. d. P. p. e. T. y. D. México., «POLITICAS PUBLICAS DESTINADAS A REDUCIR EL USO DEL AUTOMOVIL,» Griselda Ojeda, Mónica Peón, Ciudad de Mexico, 2012.



## 4. Funcionalidades del producto

1. Registros de Cuentas
2. Control de acceso mediante un login
3. Gestión de Cuentas de Usuario
4. Protección de la información
5. Uso de QR y Temporizador
6. Sensor para saber si el vehículo se encuentra en el cajón o no
7. Cámaras para identificar las placas del vehículo.
8. Servicio de Pago Online
9. Capacidad del software para interactuar con uno o más sistemas específicos.
10. Administración del estado del parquímetro

## 5. Clases y características de usuarios

La clasificación de los usuarios que van a utilizar nuestro software Parkimaniacos se divide en 2 categorías en general tendremos dos clientes que son el gobierno y las empresas que quieran utilizar nuestros servicios de parquímetros, por parte del gobierno ya que estos utilizarán la app para recolectar información que pueda ser utilizada para un estudio sobre la frecuencia de uso de zonas muy transitadas por otro lado por parte las empresas que tengan centros comerciales o zonas privadas que también quieran implementar nuestro servicio de parquímetro.

## 6. Entorno operativo

Sistema operativo

**Windows :** Windows 10 es el sistema operativo más utilizado en la actualidad por lo que se aplicará el uso de este sistema operativo más rentable además de ser de fácil interpretación y entendimiento al usuario

**Página web:** se divide en dos cosas el Front end que es la parte de los clientes y el back end de la parte del desarrollador y para esto hablaremos de esto

**HTML:** se utilizó para dar el esqueleto o estructura del desarrollo de la página web

**CSS:** se utilizó para dar estilos a la página como darle un toque único a la página **JavaScript:** se utilizó para las animaciones al css y html combinados



Aquí solo se menciona lo superficialmente el uso de las tres tecnologías html, css y JavaScript para el mejor desempeño y vista del usuario a continuación hablaremos de esto en back en

## Desarrollador Back-end:

Node.js: se utilizó para poder combinar lo que es el servidor y JavaScript y los formularios y no tiene que utilizar diferentes lenguajes en pocas palabras la unificación del lenguaje

Mysql: es un sistema de gestión de base de datos relacional (RDBMS) MySQL se ejecuta en prácticamente todas las plataformas, incluyendo Linux, UNIX y Windows y nosotros optaremos aparte de la portabilidad las prestaciones ya que sql server cobra y mysql es gratuito.

## 7. Requerimientos funcionales

En este punto se revisarán los requerimientos del proyecto en general los cuales serán cubiertos dadas la necesidad de los clientes.

### 7.1 Registro e inicio en la aplicación móvil

Descripción: El usuario debe de registrarse con su correo electrónico u otra alguna red social eso ira cambiando conforme con el desarrollo de la aplicación.

Prioridad: La prioridad de esta tarea es muy alta, se tiene que establecer una cuenta de usuario válida para poder acceder a la aplicación de los parquímetros de tal manera que este ligado el método de pago que el usuario desea usar

Acciones iniciadoras y comportamiento esperado: En este el usuario se registre en la app con una cuenta valida, de correo o con alguna red social como se ha mencionado previamente. Una vez hecho lo anterior se tendrá que loguear en la app para poder comenzar con el uso de la aplicación.

Requerimientos funcionales:

REQ 1 – Registro de Cuentas





El usuario para poder ingresar a la aplicación por primera vez tiene que registrarse previamente para poder interactuar con la app utilizando correo electrónico o alguna red social

#### REQ 2 - Control de acceso mediante un login

En este requerimiento hablamos acerca del usuario el cual se debe loguear en la aplicación para lograr interactuar con la aplicación, si existe un error es causa del registro o el inicio de sesión se debe a datos erróneo así que el usuario debe verificar bien la información.

#### REQ 3 - Gestión de Cuentas de Usuario

En este apartado aclaramos que la gestión de las cuentas de los usuarios es muy importante, por la seguridad que brindaremos será de mejor nivel, para lograr que el usuario se sienta seguro al momento de utilizar la aplicación.

#### REQ 4 - Protección de la información

La información tiene que estar protegida bajo los estándares determinados correspondientes.

### 7.2 Funcionamiento con Temporizador y lector QR

Descripción: el usuario a la hora de utilizar la aplicación después de registrarse tendrá que leer un código QR que se encontrara en el parquímetro, para después empezar con el contador de tiempo y cuando deje el parquímetro el sensor dará la señal a la aplicación y se detendrá el cronometro y se procederá a el cobro del tiempo que se estuvo en el parquímetro.

Prioridad: Este es un elemento que es de suma importancia por la razón que dependiendo del tiempo que el usuario este en el parquímetro será el monto para pagar y se generará un ingreso.

Acciones iniciadoras y comportamiento esperado: Lo esperado en esta parte es que el usuario empiece con la lectura del QR y después el cronometro empiece a contar, para que cuando el usuario regrese y desaloje el cajón el cronometro pare y se haga el cobro.

Requerimientos funcionales:

#### REQ 1 – Sensores



Se utilizarán sensores de proximidad los cuales vamos a definir mejor en la sección de requerimientos no funcionales.

#### REQ 2 – Cámaras lectoras de Matriculas

Se utilizarán cámaras en caso de que se incumplan las normas establecidas a la hora de la interacción del usuario con los parquímetros.

por ejemplo: no leer el QR correspondiente a su cajón o hacer cualquier uso indebido del parquímetro

#### REQ 3 – Pagos Online

El usuario deberá poder pagar de manera online con total seguridad

### 7.2 Capacidad para poder interactuar con diferentes sistemas.

Descripción: El Cliente podrá utilizar el software en el sistema que el dese y para cuando él lo dese.

Prioridad: No es demasiado alta, pero si es a tomar en cuenta y además que tenga seguridad

Requerimientos funcionales:

#### REQ 1 – Seguridad en el parquímetro

El Parquímetro debe ser seguro y para eso se implementarán cámaras aparte seguridad al momento del pago.



## 8. Reglas de negocio

1. 35% del costo total del producto va a pago de proveedores
2. La ganancia mínima será de 38% del costo total del proyecto
3. 5% de las ganancias se dirigirá a adolescentes interesados en Desarrollo de software (becados)
4. 3% de las ganancias se dirigirá a mejorar herramientas las de desarrollo
5. Hacer un respaldo Diario de todos los documentos y versiones de los softwares en el servidor de respaldo
6. Se hará junta informativa del personal cada semana los viernes 11AM a 1PM
7. El proyecto será auditable
8. El grupo de desarrollo tendrá 1 semana para pedir licencias de herramientas
9. Se proporcionar estadísticas quincenalmente sobre el avance total
10. Los requerimientos no se podrán cambiar después de 1 mes iniciado el proyecto
11. El cliente tendrá derecho a 3 modificaciones de requisitos.
12. El cliente proporcionara horarios para hacer pruebas en las zonas.
13. El gobierno proporcionara las reglas que debemos seguir para la instalación del proyecto

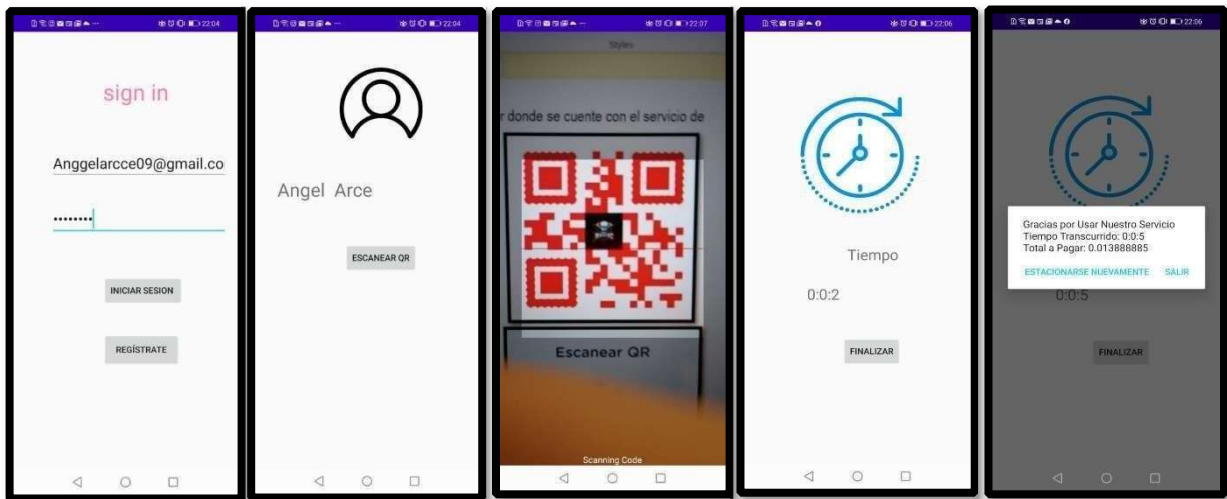


## 9. Requerimientos de interfaces externas

### 9.1. Interfaces de usuario

#### GUI para el usuario final (Fase Beta)

En esta interface se estandarizo un diseño minimalista para la fase de pruebas donde se pusieron iconos sencillos, en la ventana de inicio de sesión se incrusto botones predeterminados por el entorno gráfico , al igual que los campos de textos Las capturas de la Aplicación fueron tomadas cronológicamente:



### 9.2. Interfaces de hardware

Computadores: Pagina web para mostrar estadísticas y publicidad dispositivos móviles:

Aplicación Principal del proyecto con mayor interacción con el cliente

otros dispositivos: Arduino para poder controlar los sensores y actuadores de nuestro sistema y poder interactuar con la base datos

Protocolos de comunicación que soporta

1. TCP/IP
2. HTTP
3. HTTPS
4. SSH



### 9.3. Interfaces de software

Se creo un triángulo de interacción entre hardware - DB – móvil

Donde el hardware le mandara los espacios disponibles a la base de datos para cuando la aplicación móvil consulte los espacios aparezcan en tiempo real, por medio de triggers en la base de datos y por medio de sensores ultrasónicos en el hardware, con la herramienta de ethernet shield se conecta el Arduino a la base de datos y por medio de web service en el apartado de móvil

Y es en el mismo caso cuando el espacio está lleno , manda la información y en la aplicación móvil empieza a contar el tiempo para después que el hardware le comunique que está disponible nuevamente , termina de conteo y cobra la cantidad.

### 9.4. Interfaces de comunicación

ETHERNET: es la de un estándar de comunicación de red capaz de manejar grandes cantidades de datos a velocidades de 10 Mbps o 100 Mbps. La especificación utiliza un protocolo abierto en la capa de aplicación. En la industria es especialmente popular para aplicaciones de control.

HTTP: es el protocolo de comunicación que permite las transferencias de información en la World Wide Web. HTTP fue desarrollado por el World Wide Web Consortium y la Internet Engineering Task Force, El desarrollo de aplicaciones web necesita frecuentemente mantener estado. Para esto se usan las cookies, que es información que un servidor puede almacenar en el sistema cliente.

HTTPS: es un protocolo de comunicación de Internet que protege la integridad y la confidencialidad de los datos de los usuarios entre sus ordenadores y el sitio web. Dado que los usuarios esperan que su experiencia online sea segura y privada, te recomendamos que adoptes HTTPS para proteger sus conexiones con tu sitio web, independientemente de lo que este contenga.

WAP: es un estándar abierto internacional para aplicaciones que utilizan las comunicaciones inalámbricas, por ejemplo: acceso a servicios de Internet desde un teléfono móvil.

SSH: es un protocolo que facilita las comunicaciones seguras entre dos sistemas usando una arquitectura cliente/servidor y que permite a los usuarios conectarse a un host remotamente. A diferencia de otros protocolos de comunicación remota tales como FTP o Telnet, SSH encripta la sesión de conexión, haciendo imposible que alguien pueda obtener contraseñas no encriptadas.



## 10. Requerimientos no funcionales

Aquí se definen los requerimientos implícitos en las peticiones del usuario los cuales son los siguientes:

Web services: todo lo que tiene que ver con el manejo del servidor y servicios en la nube para poder acceder al servidor y a la información de este.

Eficiencia:

El sistema debe ser capaz de procesar N transacciones por segundo. Esto se medirá por medio de la herramienta SoapUI aplicada al Software Testing de servicios web.

Toda funcionalidad del sistema y transacción de negocio debe responder al usuario en menos de 5 segundos.

El sistema será capaz de operar adecuadamente con hasta 100.000 usuarios con sesiones concurrentes.

Los datos modificados en la base de datos deben ser actualizados para todos los usuarios que acceden en menos de 2 segundos.

Seguridad lógica y de datos:

Los permisos de acceso al sistema podrán ser cambiados solamente por el administrador de acceso a datos.

El nuevo sistema debe desarrollarse aplicando patrones y recomendaciones de programación que incrementen la seguridad de datos.

Todos los sistemas deben respaldarse cada 24 horas. Los respaldos deben ser almacenados en una localidad segura ubicada en un edificio distinto al que reside el sistema.

Todas las comunicaciones externas entre servidores de datos, aplicación y cliente del sistema deben estar encriptadas utilizando el algoritmo RSA.

Si se identifican ataques de seguridad o brecha del sistema, el mismo no continuará operando hasta ser desbloqueado por un administrador de seguridad.



#### Usabilidad:

El tiempo de aprendizaje del sistema por un usuario deberá ser menor a 4 horas.

La tasa de errores cometidos por el usuario deberá ser menor del 1% de las transacciones totales ejecutadas en el sistema.

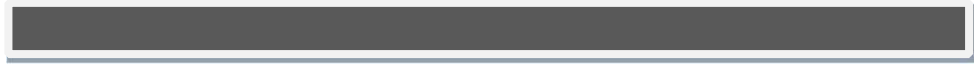
El sistema debe contar con manuales de usuario estructurados adecuadamente.

El sistema debe proporcionar mensajes de error que sean informativos y orientados a usuario final.

El sistema debe contar con un módulo de ayuda en línea.

La aplicación web debe poseer un diseño “Responsivo” a fin de garantizar la adecuada visualización en múltiples computadores personales, dispositivos tableta y teléfonos inteligentes.

El sistema debe poseer interfaces gráficas bien formadas y su mejora continua.



## 11. Otros requerimientos

## 12. Glosario

DB: Data Base o base de datos en español.

GUI: Graphic User Interface o interfaz gráfico de usuario en español

Triggers: Disparadores de acciones en la base de datos.

Ethernet Shield: Módulo que se le adecua al Arduino en la parte superior para darle conexión a internet

Entorno gráfico: IDE o herramienta que se va a utilizar para desarrollar o programar el código del proyecto

Responsive (responsivo): La aplicación web sea adaptable a distintos dispositivos que accedan

Mejora continua: Hacer que nuestra página web sea visible para la mayoría de los dispositivos que quieran acceder a ella, haciendo diversas mejoras.