****

**PLAN DE**

**PRUEBAS**

Resumen

**Capitulo I. Introducción**

1.1 Propósito

1.2 Alcance

1.3 Definiciones, Acrónimos y Abreviaturas

1.4 Referencias

1.5 Visión General del Documento

**Capitulo II. Misión de la evaluación**

2.1 Misión de las pruebas

2.2 Motivaciones para la realización de las pruebas

**Capitulo III. Esquemas de pruebas planeadas**

3.1 Pruebas

3.2 Pruebas no planeadas

**Capítulo IV. Contexto**

4.1 Tipos de pruebas.

4.2 Técnicas de pruebas

**Capítulo V. Entregables**

5.1 Productos adicionales

5.2 Entorno

**Capítulo VI. Responsabilidades y necesidades del personal**

6.1 Personas y roles

6.2 Tareas

**Capítulo VII. Apéndices**

Bibliografía

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre del proyecto: | *SmartParking* | | |
| Alias del proyecto: | *Smart\_Parking\_App* | | |
| Versión: | *1.0* | Fecha última modificación: | *15/02/2017* |

**Resumen**

En el presente documento, “*Plan de Pruebas*”, se definirá y detallarán las pruebas que se realizarán durante el desarrollo del proyecto *SmartParking,* describiendo el alcance que debe tener el documento, su propósito, misión de las pruebas, entre otras.

De igual forma se especificarán los entregables con los que deberá cumplir el equipo de trabajo a lo largo del proyecto.

En el capítulo seis se hablará de los roles de cada uno de los integrantes del equipo, y las tareas correspondientes de cada uno de ellos.

**Capitulo I. Introducción**

* 1. **Propósito**

El propósito de este documento es explicar el alcance, enfoque, recursos requeridos, calendario, responsables y manejo de riesgos del proceso de pruebas que se llevará a cabo para el proyecto SmartParking.

* 1. **Alcance**

El equipo de trabajo pretende realizar pruebas unitarias, con las cuales pretende asegurar el buen funcionamiento de la aplicación, así como la detección de posibles fallos o errores de manera más eficaz.

* 1. **Definiciones, Acrónimos y Abreviaturas**
  2. **Referencias**

PL\_DESARROLLO\_SOFTWARE.docx

* 1. **Visión General del Documento**

Este documento consta de seis capítulos: introducción, misión de la evaluación y motivaciones, esquemas de pruebas planeadas, contexto, entregables, responsabilidades y necesidades del personal y apéndices.

**CAPITULO II. Misión de la evaluación y motivaciones.**

**2.1 Misión de las pruebas**

Encontrar y corregir errores o fallos no previstos durante la planeación del proyecto, con el objetivo de que la aplicación funcione de manera adecuada, corrigiendo de manera adecuada los errores que puedan surgir.

**2.2 Motivaciones para la realización de pruebas**

La motivación que tiene el equipo de trabajo para la realización de pruebas es la de asegurarse de corregir todos los posibles fallos que pueda presentar la aplicación para poder asegurar el buen funcionamiento de la misma y sobre todo un producto de calidad para satisfacer las necesidades del usuario.

**CAPITULO III. Esquema de pruebas planeadas**

**3.1 Pruebas planeadas**

* Pruebas de la base de datos
* Pruebas de la interfaz
* Pruebas de funcionalidad

**3.2 Pruebas no planeadas**

* Falla en la interfaz
* Falla de conexión
* Fallas en la aplicación

**CAPITULO IV. Contexto**

Un plan de pruebas es una herramienta prescindible pero muy útil dentro de un proyecto, es un documento que ayuda a dar estructura tanto al proceso de pruebas como a la documentación de las mismas, el propósito del plan de pruebas es explicitar el alcance, enfoque, recursos requeridos, calendario, responsables y manejo de riesgos de un proceso de prueba, en buen plan de pruebas ayuda a la organización del proyecto completo, a la estimación de los recursos necesarios y a la gestión de la calidad en general

En este capítulo se detallaran los tipos de pruebas de software que se realizaran para el sistema SmartParking, dando una breve descripción de los tipos de prueba a realizar y las técnicas de prueba que serán aplicadas en el software, se probarán las funciones de la interfaz móvil, módulos de ingresos de datos, la conexión con la base de datos y a los módulos de la interfaz.

**4.1 Tipos de pruebas**

**Pruebas Unitarias:**

Se focaliza en ejecutar cada módulo, busca asegurar que el código funciona de acuerdo con las especificaciones y que el módulo lógico es válido, este tipo de prueba divide el código en unidades lógicas fáciles de probar, por cada unidad hay que definir los casos de prueba que serán realizados en cada fragmento de código.

**Pruebas del Sistema**

Asegurar la apropiada navegación dentro del sistema, ingreso de datos, procesamiento y recuperación, las pruebas del sistema deben enfocarse en requisitos que puedan ser tomados directamente de casos de uso y reglas y funciones de negocios. El objetivo de estas pruebas es verificar el ingreso, procesamiento y recuperación apropiado de datos, y la implementación apropiada de las reglas de negocios.

**Pruebas de Desempeño**

Estas validan el tiempo de respuesta para las transacciones o funciones de negocios bajo dos condiciones, las pruebas de desempeño miden tiempos de respuesta, índices de procesamiento de transacciones y otros requisitos sensibles al tiempo, su objetivo es verificar y validar los requisitos de desempeño que se han especificado siendo en este caso el software o aplicación móvil.

**Pruebas de Carga**

Verificar el tiempo de respuesta del sistema para transacciones, las pruebas de carga miden la capacidad del sistema para continuar funcionando apropiadamente bajo diferentes condiciones de carga, el fin de las pruebas de carga es determinar y asegurar que el sistema funciona apropiadamente aún más allá de la carga de trabajo máxima esperada.

**4.2 Técnicas de pruebas**

Técnicas de caja negra o funcional: que realizan pruebas sobre la interfaz del programa a probar, entendiendo por interfaz las entradas y salidas de dicho programa. No es necesario conocer la lógica del programa, únicamente la funcionalidad que debe realizar. Esta técnica, consiste entonces en la construcción de casos prueba por cada entrada posible sin importar su valides. Sin embargo sería una tarea fuera de los límites de un proyecto de común, para lo cual existen entonces algunos métodos que se encargan de simplificar esta labor y que cubren los aspectos esperados de una prueba de certificación de software.

Técnica de caja blanca o estructurales: se centran en los detalles procedimentales del software, por lo que su diseño está fuertemente ligado al código fuente, se basa en el diseño de casos de prueba que usa la estructura de control del diseño procedimental para derivarlos. Mediante la prueba de la caja blanca el ingeniero del software puede obtener casos de prueba que garanticen que se ejerciten todos los caminos independientes de cada módulo, ejerciten todas las decisiones lógicas en las vertientes verdadera y falsa, se ejecuten todos los bucles en sus límites operacionales. Por esto se considera a la prueba de Caja Blanca uno de los tipos de pruebas más importantes que se le aplican al software, logrando como resultado que disminuya en un gran porciento el número de errores existentes en los sistemas y por ende una mayor calidad y confiabilidad.

**Capítulo V. Entregables**

**5.1 Productos adicionales**

Para el presente plan de pruebas

**5.2 Entorno**

Un caso de prueba (test case) en ingeniería del software es un conjunto de condiciones o variables bajo las cuáles un analista determinará si una aplicación, un sistema software, o una característica de éstos es parcial o completamente satisfactoria Responsabilidades y necesidades del persona.

|  |  |
| --- | --- |
| Servidor | Raspberry p 3 |
| Procesador | Procesador a 1,2 GHz de 64 bits con cuatro núcleos armv8 |
| Memoria RAM | 1GB RAM |
| Almacenamiento | Memoria micro sd 16 GB |
| Tipo de red | 802.11n Wireless LAN |
| Sistema operativo | UBUNTU |

|  |  |
| --- | --- |
| Servidor | Celular móvil |
| Procesador | ARMv5 o superior |
| Memoria RAM | 1GB RAM |
| Almacenamiento | Memoria interna 50 Mb |
| Tipo de red | 802.11n Wireless LAN y GSM 1700 MHz |
| Sistema operativo | Android 4.0 igual o superior |

**6.1 Personas y roles**

|  |  |
| --- | --- |
| 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 | Descripción |
| 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. |

**6.2 Tareas**

|  |  |
| --- | --- |
| Rol | Tarea |
| Programador | Desfragmentación del código para las pruebas de caja blanca y corregirá los errores que sean reportados por el diseñador. |
| Analista | Evaluará la función de cada prueba, que los módulos cumplan su función en las pruebas de caja negra. |
| Diseñador | Hará un reporte con todas las fallas que se generen en los módulos y entregará al tester los códigos y módulos que funcionen correctamente. |
| Tester | Evalúa el sistema final al conjuntar todos los módulos, verificará que todas las funciones sean correctas y en caso de que presenten alguna falla lo reportará directamente con el programador. |

**Capítulo VII. Apéndices**

Pueden contener todo tipo de información relevante para el Plan de pruebas pero que, propiamente, no forme parte del Plan de Pruebas.

**Bibliografía**

<http://ing-sw.blogspot.mx/2005/04/tipos-de-pruebas-de-software.html>