****

****

**PLAN DE PRUEBAS**

Resumen

**Capítulo I. Introducción**

1.1 Propósito

1.2 Alcance

1.3 Definiciones, Acrónimos y Abreviaturas

1.4 Referencias

1.5 Visión General del Documento

**Capítulo II. Misión de la evaluación y Motivaciones**

2.1 Misión de las pruebas

2.2 Motivaciones para la realización de las pruebas

**Capítulo III. Esquemas de pruebas planeadas**

3.1 Pruebas planeadas

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: | *Control prenatal mediante Aplicación móvil y wearable* | | |
| Alias del proyecto: | *Pregnancy Care* | | |
| Versión: | *1.0* | Fecha última modificación: | *13/02/2017* |

# Resumen.

Todo sistema que se crea tiene que ser verificado y corregido a tiempo para que los errores que lleguen a ocurrir se detecten a tiempo y no causen problemas mayores a largo plazo. Es por ello que se realizara una serie de pruebas para la mejorar la calidad de la aplicación que se está desarrollando.

En las siguientes páginas se especificaran cual es el propósito y el alcance del presente documento, donde en cada una se describirá las técnicas a utilizar para tener calidad en el software.

Así como se mencionarán los roles que cada integrante del equipo llevara a cabo para que el plan de pruebas cumpla con su misión.

# Capitulo I. Introducción.

En las siguientes páginas se especificaran las técnicas que se utilizaran para cumplir con el plan de pruebas en base al marco teórico especificado.

Para el plan de pruebas se evaluaran diferentes partes esenciales para la aplicación, con una breve explicación de la importancia de las mismas.

El documento está diseñado para ser como una guía para el proceso que se llevara a cabo, así como el entorno en donde se llevarán a cabo las pruebas.

También se especificaran las responsabilidades y necesidades de las personas involucradas, es decir, se especificaran los roles y tareas de cada persona entorno a la ejecución del plan de pruebas, así como la capacitación para cada uno de los miembros.

Cada prueba tiene como fin el buen funcionamiento de la aplicación y que los errores sean detectados y corregidos a tiempo antes de que suceda algo que provoque una alteración en la planeación de desarrollo del proyecto.

## 1.1 Propósito

En este documento se especificara los alcances, enfoque, recursos requeridos y los responsables en la ejecución del plan de pruebas.

Su objetivo es guiar al interesado en el proceso de pruebas en las distintas etapas del proyecto, esta será la herramienta en la que se encontrará con:

* La evaluación en busca de errores y/o defectos que puedan existir dentro de la aplicación con el fin de corregirlos.
* Recolectar todo la información necesaria, para así poder planear y controlar las pruebas.
* Se establecen las técnicas, herramientas y actividades relacionadas con la ejecución y validación del plan de pruebas.

## 1.2 Alcance

El finalizar las pruebas del sistemas, la aplicación se encuentra corregido de los errores que se hayan encontrada en los scripts, base de datos, etc. Todo descrito en este documento, este se enfocara dependiendo de en donde se encuentre el error.

Que puede llegar a ser en algunas de las siguientes categorías:

* Funcionalidad. Asegurar que la aplicación funcione de manera correcta.
* Recuperación. Verificar su adecuada recuperación a diversos tipos fallas.
* Conectividad. Chequearlos tiempos de respuesta de la aplicación.
* Seguridad. Haciendo uso del nombre de usuario y contraseña.
* Eficiencia. Verificar que use los recursos de manera eficiente.
* Confiabilidad. Para medir la cantidad de errores no sea alta.

## 1.3 Definiciones, Acrónimos y Abreviaturas

* Error

Se refiere a la diferencia entre la salida actual de un software y la correcta. También puede ser una acción que introduce en el software un defecto o falla.

* Defecto

Es una condición que causa que un software falle al realizar una función requerida

* Defecto

Es una condición que causa que un software falle al realizar una función requerida

## 1.4 Referencias

* PL\_NEGOCIOS\_V1.0.

## 1.5 Visión General del Documento

Encontrar y corregir de manera organizada los errores que surgen dentro del software que se está desarrollando.

Teniendo como principal objetivo tener una buena calidad en el producto que se está por ofrecer.

Para ello los integrantes del equipo estarán presentes en todo momento dependiendo del rol que se le asigne a cada uno y a sus tareas.

Por lo que será necesario que todos los involucrados conozcan de que se trata este plan de pruebas, para que así se cumpla de manera eficiente.

Este documento será como una guía a lo largo de las pruebas que se realizaran a la aplicación.

# Capitulo II. Visión en las evaluaciones y motivaciones.

## 2.1 Misión de las pruebas

Encontrar y corregir de manera correcta lo que se presente en el desarrollo y ejecución de la aplicación a crear, ya sean errores y/o riesgos previstos o no, con el fin de mejorar el producto y servicio de calidad.

Para ello se seguirá una metodología en el proceso del plan de pruebas, en cuyas fases se describirá la forma en que se llevaran a cabo las pruebas en el transcurso del desarrollo del producto.

Estas pruebas se realizaran para verificar que se están por satisfacer todos los requerimientos planteados. Así como ayudar a que el presupuesto no aumente en caso de que el proyecto no vaya de la manera en que se planeó en un principio, ya que puede llegar el caso de que si un error no es corregido este puede causar que el tiempo de desarrollo se alargue y con el ello el presupuesto.

Con el fin de no correr ningún riesgo como estos, la misión es prevenir todas las consecuencias que puede acarrear no prevenir un error y corregirlo.

## 2.2 Motivaciones para la realización de pruebas.

La realización de las pruebas tiene como motivación la mejora de la aplicación en cuanto las funciones, calidad, seguridad, conectividad entre los scripts, base de datos. Para prevenir riesgos mayores como el aumento de presupuestos, así como el tiempo para la realización del proyecto, aun si se termina a tiempo la calidad del producto se verá disminuida.

Por lo tanto la prevención en la corrección de errores y/o fallos evitara llegar a problemas mayores.

# Capítulo III. Esquema de pruebas planeadas

Entre los principales recursos a evaluar son:

* El código de la aplicación, ya que es necesario que esta se pueda configurar y conectar entre todas las partes de la aplicación, es necesario que funcione de forma correcta.
* Entre todos los integrantes del equipo deben conocer cuál es el principal objetivo de la aplicación, para que así saber hacia dónde va encaminado el plan de pruebas y que es lo que se necesita verificar primero.

## 3.1 Pruebas planeadas

* Prueba de funcionalidad.
* Prueba de conectividad.
* Prueba de base de datos.
* Prueba de interface.
* Prueba de recuperación.
* Prueba de seguridad.
* Prueba de configuración.

## 3.2 Pruebas no planeadas

# Capítulo IV. Contexto

Los sistemas que existen y se van creando día a día son más complejos con estándares de calidad altos.

Para este plan se utilizara la metodología codificar y corregir que consiste en dos pasos:

* Escribir partes de código de la aplicación.
* Resolver los problemas del código que se vayan presentando.

A pesar de ser un modelo poco útil, se utiliza con frecuencia. Cuando se inicia con este modelo se debe de tener la idea de lo que se empieza a construir, es decir cual es el objetivo final del producto.

Entre sus ventajas se encuentran:

* Que no se pierde tiempo en la documentación, planificación, documentación, etc.
* Se puede mostrar los progresos de la aplicación.
* No se necesita mucha experiencia.

## 4.1 Tipos de pruebas

* Prueba de funcionalidad.

Se verificara que todos sus componente funciones según los requerimientos y objetivos planteados.

* Prueba de conectividad.

Verificara que tanto la base de datos se conecte al script de la aplicación, así como que la pulsera se conecte con el script de la aplicación.

* Prueba de base de datos.

Comprobar que se pueda acceder a la base de datos y que la información se quede guardada en ella.

* Prueba de recuperación.

La aplicación debe tener la capacidad de recuperarse a cualquier contingencia. En caso de que la verificación sea de forma automática debe de tener una recuperación de datos, así como el proceso de reinicio. En caso de que la recuperación siga en fallos este deberá tener intervención de humana para evaluar los tiempos de corrección.

* Prueba de seguridad.

Asegurar que cumpla con los objetivos de confidencialidad, disponibilidad, integridad y exactitud de la información. Se debe verificar que las contraselas no sean visibles al ser ingresadas y que se encuentren encriptadas.

* Prueba de configuración.

Asegurar que en un futuro se le puedan agregar y/o modificar algunas funciones a la aplicación.

## 4.2 Técnicas de pruebas

Principalmente se comenzara con las partes pequeñas de la aplicación para continuar con las más grandes.

Existen dos tipos de métodos de prueba:

* Los de caja blanca.
* Los de caja negra.

Las de caja blanca tienen como objetivo comprobar los estatutos condicionales, bucles y ciclos. Tiene varios puntos importantes como conocer el desarrollo del programa, comparar el desarrollo del programa con la documentación.

En ella también existen varios criterios de cobertura como:

* Estructurales.
* Funcionales.
* Estadísticos.

En cuanto a los de caja negra son las pruebas funcionales dedicadas en el exterior de la prueba. Se centra principalmente en un módulo o sección específica del software.

# Capítulo V. Entregables

## 5.1 Productos adicionales

En base al plan de prueba se generara un documento en donde se muestren los errores que se encontraron y como se solucionaron.

Especificando de manera exacta cada uno. En caso de que se encuentren errores que no estaban planeados, estos también se especificaran.

## 5.2 Entorno

Para probar cada uno de las partes del software se requiere de una computadora, asi como también de teléfono en que se instalara la aplicación para ir verificando las partes del código que se llevan.

Para ello también es necesario tener un espacio que esté tranquilo o por lo menos que no existan muchas distracciones para que así se pueda concentrar mejor para realizar las pruebas pertinentes.

# Capítulo VI. Responsabilidades y necesidades del personal

## 6.1 Personas y roles

|  |  |
| --- | --- |
| Persona | Rol |
| Gloria Guadalupe Rosas Zamudio | Líder de pruebas |
| Rosalía Contreras Ávila | Consultor |
| Jesús Vergara Cortes | Consultor |
| Gustavo Orbezo Hernández | Tester |
| Rene Gómez Hernández | Tester |
| Jessica Monfil Ramírez | Consultor |
| Elizabeth Ramírez Medina | Tester |

## 6.2 Tareas

* Líder de pruebas.

Este tiene experiencia en el diseño, ejecución y soporte de pruebas sobre un producto de software, también es capaz de estimar, planificar y dar seguimiento.

Estar al pendiente de las últimas técnicas para resolver pruebas.

Planificar y organizar las pruebas.

Analizar los requerimientos y requisitos del proyecto

* Tester de software.

Es quien diseña, ejecuta e informa el resultado de las pruebas sobre un producto de software. Los testers integran el equipo de pruebas e interactúan con el equipo de desarrollo y la gerencia de proyectos.

* + - Consultor.

Es un rol adecuado para aquellas personas capaces de dirige un equipo de pruebas y orientar sobre mejores prácticas.

# Capítulo VII. Apéndices

# Bibliografía

Megamaxi S.A. . (2011). Plan de pruebas. 13/02/2017, de SlideShare Sitio web: <http://es.slideshare.net/iguamba666/plan-de-pruebas>

Colegio Universitario de Caracas. (2012). Plan de pruebas. 13/02/2017, de Worpress Sitio web: <https://sistemagestionseccional.files.wordpress.com/2012/07/plan_de_pruebas_rev.pdf>

UNAM. (---). Marco teorico. 13/02/2017, de ptolomeo Sitio web: <http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/237/A4.pdf?sequence=4>