

# TEST PLAN

*Universidad Autónoma de Zacatecas*  
*“Francisco García Salinas”*

Integrantes:

- Ana Cristina España García
- Andrea Isabel Castro Esquivel
- María de los Angeles Pérez Cardona



Contenido

Abstract	3
Historial de Versiones	3
INTRODUCCIÓN	3
1 ESTRATEGIA DE TEST	3
1.1 Alcance del Testing	3
1.1.1 Característica a ser testada	3
1.1.2 Características que no serán testeadas	4
1.2 Tipo de Test	4
1.3 Riesgos y Problemas	5
1.4 Logística de Pruebas	5
1.4.1 ¿Quiénes realizarán las pruebas?	5
1.4.2 ¿Cuándo ocurrirán las pruebas?	5
2 OBJETIVOS DEL TEST	6
3 CRITERIO DE PRUEBA	6
3.1 Criterio de Suspensión	6
3.2 Criterio de Salida	6
4. PLANEACIÓN DE RECURSOS	6
4.1 Recursos del sistema	6
4.2 Recursos Humanos	7
5. AMBIENTE DE PRUEBAS	7
6. CALENDARIO Y ESTIMACIÓN	8
6.1 Todas las tareas y estimaciones del proyecto.	8
7. ENTREGABLES DE PRUEBA	8
7.1 Antes de la fase de prueba.	8
7.2 Durante las pruebas.	8
7.3 Después de terminar ciclo de prueba.	8

ABSTRACT

Este documento provee una vista general del proyecto y la estrategia de pruebas del producto, una lista de los entregables de testing y el plan para el desarrollo.

HISTORIAL DE VERSIONES

Versión	Escrito por	Fecha de revisión	Aprobado por	Fecha de aprobación	Salidas
0.1	Cherry Hue Softwares	04/06/2024	Cherry Hue Softwares	04/06/2024	APROBADO

INTRODUCCIÓN

El Test Plan está diseñado para prescribir el alcance, enfoque, recursos y el calendario de todas las actividades de testing del proyecto Altas MFC

En el plan se identifican los elementos y características a ser testeados, las características a ser testeadas, los tipos de testing a ser ejecutados, el personal responsable del testing, los recursos y calendario requerido para completar el testing y los riesgos asociados con el plan.

1 ESTRATEGIA DE TEST

1.1 Alcance del Testing

1.1.1 Característica a ser testead

Todas las características del sitio web de Altas MFC que necesitan ser testeados.

Nombre del módulo	Roles Aplicables	Descripción
----------------------	------------------	-------------

<b>Registrar usuario</b>	Administrador.	<b>Administrador:</b> El administrador es el único que puede registrar usuarios en el sistema.
<b>Nuevo Equipo</b>	Administrador. Empleado. Técnico.	<b>Usuarios:</b> Todos los usuarios existentes en el sistema pueden agregar equipos, junto con su información correspondiente para que estos sean examinados o reparados.
<b>Nueva Orden de servicio</b>	Administrador. Empleado. Técnico.	<b>Usuarios:</b> Todos los usuarios existentes en el sistema pueden añadir notas sobre la revisión de un equipo.
<b>Poner cotizaciones</b>	Administrador. Empleado. Técnico.	<b>Usuarios:</b> Todos los usuarios existentes en el sistema pueden agregar una cotización sobre las reparaciones correspondientes al equipo.
<b>Indicar el servicio realizado</b>	Administrador. Empleado. Técnico.	<b>Usuarios:</b> Todos los usuarios existentes en el sistema pueden añadir notas indicando los detalles que se han llevado a cabo para la reparación del equipo.
<b>Permisos</b>	Administrador.	<b>Administrador:</b> Únicamente el administrador puede brindar permisos a los usuarios registrados.
<b>Listas</b>	Administrador. Empleado. Técnico.	<b>Usuarios:</b> Todos los usuarios pueden ver los detalles de las soluciones realizadas de los equipos.
<b>Login</b>	Administrador. Empleado. Técnico.	<b>Usuarios:</b> Todos los usuarios pueden iniciar sesión proporcionando su usuario y contraseña.
<b>Editar</b>	Administrador. Empleado. Técnico.	<b>Usuarios:</b> Todos los usuarios pueden editar un equipo o una orden.
<b>Eliminar</b>	Administrador. Empleado. Técnico.	<b>Usuarios:</b> Todos los usuarios pueden eliminar un equipo o una orden.

### 1.1.2 Características que no serán testeadas

Estas características no serán testeadas debido a que no son incluidas dentro de la especificación de requerimientos de software, además de que no se contemplan dentro del alcance o requerido dentro de la materia de Testing.

- Interfaces de Usuario.
- Interfaces de Hardware.
- Interfaces de Software.
- Lógica de la Base de Datos.

- Interfaces de Comunicaciones.
- Seguridad del Sitio Web.

## 1.2 Tipo de Test

En el proyecto Altas MFC , hay tres tipos de testing que serán realizados en el desarrollo del mismo.

- **Pruebas Unitarias:** Son las pruebas que se realizan de manera colaborativa (cada miembro del equipo), son los módulos de software individual que serán desarrollados y testeados por cada uno de los módulos.
- **Pruebas Funcionales:** Son las pruebas con las que también deberá cumplir el proyecto; en las cuales se simula el comportamiento del sistema ante el uso de sus usuarios. Es como si el usuario estuviera utilizando el sistema, y esta es una manera de evaluar la conformidad con los requerimientos especificados.
- **Pruebas de Performance con JMeter:** Estas son las pruebas que se aplicarán para evaluar el performance del sistema y ver cómo se comporta el sistema bajo ciertos estímulos de acceso y carga del mismo.
- **Pruebas de integración:** La finalidad de estas pruebas consiste en garantizar que cada uno de los componentes del sistema se comuniquen correctamente entre sí.

## 1.3 Riesgos y Problemas

Riesgo	Mitigación
Falta de habilidades requeridas para realizar testing más adecuado al sistema web.	Tomar cursos de entrenamiento, sobre testing de sistemas web, como apoyo para mejorar las habilidades de cada uno de los miembros del equipo.
El horario o calendarización del proyecto es muy reducido en cuestiones de tiempo, puede ser difícil completar en el tiempo designado de entrega.	Designar o definir una prioridad de test para cada una de las actividades de testing, dentro del proyecto.
Falta de entrenamiento con respecto a la metodología Test Driven Development (TDD) para el desarrollo y pruebas para el proyecto.	Realización del proceso poco a poco y en caso de presentar dudas apoyarnos entre los miembros del equipo o con el profesor.
Problemas al instalar las librerías y herramientas necesarias	Apoyarse con el profesor y en foros para resolver los problemas e incompatibilidades para su uso o instalación

## 1.4 Logística de Pruebas

### 1.4.1 ¿Quiénes realizarán las pruebas?

Para este proyecto las pruebas serán realizadas por los miembros del equipo y fungirán con el rol de testers con el objetivo de ahorrar costos del proyecto.

### 1.4.2 ¿Cuándo ocurrirán las pruebas?

Los testers del equipo podrán comenzar la ejecución de las pruebas cuando las siguientes entradas se encuentren listas:

- Se cuente con los suficientes recursos humanos para poder realizar el trabajo.
- De acuerdo a la metodología TDD el desarrollo del proyecto (software) se hará a la par que se realizan las pruebas del mismo.
- Contar con un ambiente de prueba adecuado para realizar el desarrollo y pruebas del proyecto; para este caso se hará uso del Framework Django.
- Se cuente con la definición de los requerimientos técnicos, criterios de aceptación de las historias de usuario para el proyecto.

### 1.5 Ubicación del proyecto:

[https://github.com/EspaSandia24/Proyecto\\_Testing.git](https://github.com/EspaSandia24/Proyecto_Testing.git)

## 2 OBJETIVOS DEL TEST

Los objetivos del test son verificar la funcionalidad del sistema web Altas MFC y por medio de la implementación de la metodología TDD desarrollar dicha funcionalidad a partir de la creación de diferentes pruebas: pruebas de aceptación y pruebas unitarias. Este proyecto debería enfocarse en testear la funcionalidad del proyecto en general (usando las historias de usuario descritas en el SRS); teniendo en cuenta manejo de operaciones como: alta de equipos, órdenes, asignación de permisos, y modificación de órdenes y equipos. Esto con el objetivo de garantizar que estas operaciones se implementen y puedan funcionar de manera normal en un entorno real de trabajo.

## 3 CRITERIO DE PRUEBA

### 3.1 Criterio de Suspensión

En dado caso que los miembros del equipo detecten que las pruebas no están afectando positivamente al desarrollo del proyecto conforme a la metodología TDD, se deberá revisar los apartados descritos por TDD, hacer las correcciones necesarias en las pruebas y retomar el ciclo descrito para continuar con el desarrollo del proyecto.

### 3.2 Criterio de Salida

Definen los criterios que denotan una terminación exitosa del desarrollo del proyecto con sus respectivas pruebas.

- De acuerdo a la metodología TDD las pruebas definidas tendrán que completar el 100% de las funcionalidades requeridas del proyecto.
- Definición de pruebas de aceptación (behave y selenium) y cubran las historias de usuario descritas y la definición de pruebas unitarias (por medio de TestCase).

## 4. PLANEACIÓN DE RECURSOS

### 4.1 Recursos del sistema

No.	Recursos	Descripciones
1.	Servidor	Con respecto al servidor es necesario tener instalados en los equipos donde se realizarán las pruebas: Servidor de Base de Datos MySQL.
2.	Herramienta de Testing	Utilizar una herramienta de Testing que permita realizar las pruebas del sistema a la par de su desarrollo; para este caso utilizar el framework Django, ya que facilita el desarrollo del sistema y la implementación de sus pruebas correspondientes.
3.	Red	Se debe contar con una red de internet de al menos una velocidad de 5 Mb/s para la consulta de material (en caso de ser necesario) y si se ofreciera para las pruebas del sistema.
4.	Computadoras	Al menos 3 computadoras con Windows 10/11, RAM de 4 GB y CPU de 2.5GHZ. Una para cada integrante del equipo; para realizar el desarrollo de las pruebas y construcción del sistema.
5.	Herramientas de contenedores.	Contar con Docker para poder desplegar la aplicación y tener un entorno de desarrollo independiente.

### 4.2 Recursos Humanos

No.	Miembro	Tareas
1	Test Manager	Administrar el proyecto completo. Definir y asignar tareas a cada uno de los miembros del equipo.
2	Test	Identificar, describir, realizar las pruebas necesarias utilizando las técnicas, herramientas y metodologías apropiadas. Ejecutar las pruebas, corregir e implementar el desarrollo del sistema, según la metodología TDD.
3	Desarrollador en Test	Implementar los casos de prueba, las pruebas del programa, entre otras.

## 5. AMBIENTE DE PRUEBAS

El ambiente de pruebas debería estar configurado de la siguiente manera: Cada uno de los integrantes del equipo de desarrollo deberá contar con un equipo de cómputo con las características mínimas requeridas; la cual deberá tener el framework Django para poder realizar las pruebas necesarias para la implementación del sistema siguiendo la metodología de TDD y teniendo a disposición las herramientas mencionadas.

Por el momento las pruebas y el desarrollo del sistema se están realizando de manera local en cada uno de los equipos de cómputo de las personas involucradas en el proyecto. Se espera que después se pueda montar el sistema web en un servidor de pruebas.

## 6. CALENDARIO Y ESTIMACIÓN

### 6.1 Todas las tareas y estimaciones del proyecto.

Tarea	Miembro	Esfuerzo estimado
Crear la especificación de prueba.	Todos	180 minutos
Pruebas unitarias.	Todos	420 minutos
Pruebas de aceptación.	Todos	360 minutos
Historias de usuario y criterios implementados con behave y selenium.	Todos	120 minutos
Realizar ejecución de prueba.	Todos	60 minutos
Informe de prueba	Todos	90 minutos
Pruebas de performance con JMeter.	Todos	210 minutos
Reportes de cobertura de código.	Todos	80 minutos
Código cumpliendo el estándar de PEP8.	Todos	240 minutos
Total		29.33 horas

## 7. ENTREGABLES DE PRUEBA

Las entregas de prueba se proporcionan a continuación

### 7.1 Antes de la fase de prueba.

- Documento del plan de prueba.
- Documentos de casos de prueba.



## 7.2 Durante las pruebas.

- Herramienta de prueba.
- Simuladores.
- Datos de prueba.
- Prueba de matriz de trazabilidad

## 7.3 Después de terminar el ciclo de prueba.

- Resultados de la prueba / informes.