



Software de ingeniería INFO248
Profesor: Raimundo Vega/ Cristian Olivares/ Mirko Geregat
Ingeniería Civil en Informática

Plan de pruebas de software

Integrantes:

- Andrés Gutierrez
- Miguel Nahuelpán
- Esteban Tejeda
- Felipe Zabala

Profesores:

- Mirko Gueregat
- Raimundo Vega

Curso: Taller de Ingeniería de Software

Tabla de contenido

Tabla de contenido	2
1. Historial de versiones	3
2. Información del proyecto	4
3. Aprobaciones	4
4. Resumen ejecutivo	4
5. Alcance de las pruebas	5
5.1 Elementos de pruebas	5
5.2 Nuevas funcionalidades a probar	5
5.3 Pruebas de regresión	5
5.4 Funcionalidades a no probar	5
5.5 Enfoque de pruebas (estrategias)	5
6. Criterios de aceptación o rechazo	6
6.1 Criterios de aceptación o rechazo	6
6.2 Criterios de suspensión	6
6.3 Criterios de reanudación	6
7. Entregables	7
8. Recursos	8
8.1 Requerimientos de entornos - Hardware	8
8.2 Requerimientos de entornos - Software	8
8.3 Herramientas de pruebas requeridas	8
8.4 Personal	8
8.5 Entrenamiento	8
9. Planificación y organización	9
9.1 Procedimientos para las pruebas	9
9.2 Matriz de responsabilidades	9
9.3 Cronograma	9
9.3.1 Especificación de casos de pruebas	9
9.4 Premisas	12
9.5 Dependencias y riesgos	12
10. Referencias	13
11. Glosario	14

1. Historial de versiones

Versión	Fecha	Comentario
v0.1	24-11-2020	Creación de documento.
v1.0	05-01-2021	Entregable 1
v1.1	14-01-2021	Feedback añadido

2. Información del proyecto

Empresa / Organización:	Neuss Producciones
Proyecto:	Neuss App
Fecha de preparación:	16-09-2020
Cliente:	Neuss Producciones
Patrocinador principal:	Camila Gutiérrez Salas
Gerente / Lider de proyecto.	Todos
Gerente / Lider de pruebas de software:	Todos

3. Aprobaciones

Nombre y Apellido	Cargo	Departamento u organización	Fecha	Firma
Esteban Tejeda	Desarrollador			
Felipe Zabala	Desarrollador			
Andrés Gutiérrez	Desarrollador			
Miguel Nahuelpán	Desarrollador			

4. Resumen ejecutivo

Este plan de pruebas tiene como base el desarrollo de las distintas funcionalidades de la aplicación, por medio del desarrollo individual (y secuencial en caso de verse oportuno), de los casos de uso definidos en la documentación anexa.

- **Propósito:** La naturaleza del proyecto es el desarrollo e implementación de una aplicación capaz de realizar compras basadas en los elementos que se marquen como disponibles en la base de datos de la empresa.
- **Alcance:** Se espera que un administrador sea capaz de agregar, modificar o eliminar platillos disponibles, mientras que el usuario pueda realizar una transacción con aquellos que estén disponibles, por lo que esta será la orientación de las pruebas a realizar. En resumen, realizar un pedido.
- **Restricciones:** Tiempo mayormente.
- **Alcance de pruebas:** Lograr efectuar el alcance visto antes.

5. Alcance de las pruebas

5.1 Elementos de pruebas

- Gestión de Administrador: Operaciones básicas del administrador con el programa.
- Gestión de Clientes: Operaciones básicas del usuario con el programa.

5.2 Nuevas funcionalidades a probar

- Agregar platillo.
- Realizar pedido.

5.3 Pruebas de regresión

-

5.4 Funcionalidades a no probar

- Historial: Su funcionamiento no es parte del alcance, aunque se espera que se generen los datos para su futuro uso.

5.5 Enfoque de pruebas (estrategias)

Pruebas funcionales:

- Crear usuario tipo administrador (PF1): Desde la primera ventana crearlo y validarlo con la base de datos.
- El administrador crea un platillo (PF2).
- Administrador ve la lista de platillos disponibles (PF3): Esto implica lo ya registrado en la base de datos, como posibles nuevos platillos.
- Crear usuario tipo cliente (PF4): Desde la primera ventana crearlo y validarlo con la base de datos.
- Usuario ve la lista de platillos disponibles (PF5): Esto implica lo ya registrado en la base de datos.
- Usuario realiza pedido (PF6): Se selecciona un platillo, hasta que se manda el mensaje de que se está atendiendo el pedido.
- Atender pedido (PF7): El administrador acepta o rechaza un pedido, devolviendo un mensaje que pueda ver el usuario que realizó el pedido.

Pruebas no funcionales:

- Tiempo de las tareas anteriores óptimo.

Pruebas de interfaces:

- Interfaces capaces que de manera clara y simplificada permitan realizar las pruebas funcionales. Se busca que su diseño sea funcional más que estético.

6. Criterios de aceptación o rechazo

Las pruebas se realizan con la idea de que las mismas arrojen feedback inmediato por la naturaleza de las mismas. Se considerarán las pruebas funcionales como el mayor enfoque.

6.1 Criterios de aceptación o rechazo

En el caso de aceptación, sólo se considerará como aceptado si todos los elementos del recuadro se cumplen. Para el rechazo basta que uno no se cumpla. El poder realizar una prueba, considera que las pruebas anteriores se hayan efectuado con éxito.

Prueba Funcional	Aceptación	Rechazo
1	- Se genera en la base de datos el administrador con los datos ingresados. - Se logra validar el ingreso con los datos ya usados.	- No se logra generar en la base de datos el administrador. - Se logra ingresar al programa pese a no usar las credenciales correctas.
2	- Se ingresa cualquier cantidad de platillos, validando nombre, descripción, imagen y precio. - El conjunto de datos se ingresa a la base de datos.	- Se acepta un platillo sin estar todos los datos que se deben validar. - No se logra aceptar el platillo. - No se logra ingresar el platillo a la base de datos.
3	- El administrador ve la lista de los platillos en la base datos.	- No se logra ver la lista completa. - No se logra ver el platillo que se ha ingresado recientemente.
4	- Se genera en la base de datos el usuario con los datos ingresados. - Se logra validar el ingreso con los datos ya usados.	- No se logra generar en la base de datos el usuario. - Se logra ingresar al programa pese a no usar las credenciales correctas.
5	- El usuario ve la lista de los platillos en la base datos.	- No se logra ver la lista completa.
6	- El usuario puede seleccionar un platillo y mandar la solicitud de compra.	- No se logra visualizar el pedido del usuario por parte de algún administrador.
7	- El administrador ve la lista de pedidos activos y lo atiende (rechazar o aceptar).	- No se logra enviar el mensaje adecuado al usuario.

6.2 Criterios de suspensión

- **Tiempo:** Si no existe el tiempo disponible para continuar con las pruebas, se suspenden las mismas hasta un nuevo horario.

6.3 Criterios de reanudación

- **Tiempo:** Se encuentra un horario adecuado para poder continuar.

7. Entregables

Versión 1.0: Primera versión funcional del programa. Cumple con el alcance definido anteriormente.

8. Recursos

Los recursos considerados son los requeridos para probar el programa, como para editarlo en caso de requerir según las pruebas.

En una entrega anterior se especificó que sus requerimientos mínimos eran:

- Windows 7.
- Navegador Google Chrome 83.x

8.1 Requerimientos de entornos - Hardware

Se evalúa el rendimiento según se los rangos “malo”, “óptimo” y “adecuado”. Hasta el momento se ha probado en computadores con las especificaciones:

- Modelo: ASUS TUF GAMING FX505DT
- 8 GB RAM.
- AMD Ryzen 5 3550H (8 núcleos 2.1 Ghz)
- NVIDIA GeForce GTX 1650.
 - Rendimiento: Adecuado.
- Modelo: HP OMEN 15
- 8 GB RAM.
- Intel Core i5-8300H (8 núcleos 2.3 Ghz)
- NVIDIA GeForce GTX 1050TI.
 - Rendimiento: Adecuado.

8.2 Requerimientos de entornos - Software

- SO: Probado con Windows 10.
- Navegador: Probado con Mozilla firefox 84.0.1
- Angular Framework
- NodeJS
- PrimeNG (Angular)
- Discord
- Github
- Jenkins

8.3 Herramientas de pruebas requeridas

Ninguna.

8.4 Personal

Equipo de desarrollo:

- Andrés Gutiérrez
- Miguel Nahuelpan
- Esteban Tejeda
- Felipe Zabala

8.5 Entrenamiento

Conocimiento en las herramientas de Angular, NodeJS, PrimeNG y Github.

9. Planificación y organización

9.1 Procedimientos para las pruebas

Las pruebas se realizan a través de la plataforma discord, donde se puede proyectar la pantalla a modo que todo el equipo de desarrollo la puede ver en tiempo real. Estas pruebas pueden o no tener la presencia del equipo completo.

9.2 Matriz de responsabilidades

La matriz no se rellena, pues para la realización de las pruebas todos los que se encuentren presentes analizan la ejecución de las pruebas, como intentan replicarlas buscando diferencias en las mismas.

	Andres Gutierrez	Miguel Nahuelpán	Esteban Tejeda	Felipe Zabala
Ejecución de pruebas funcionales	-	-	-	-
Ejecución de pruebas no funcionales	-	-	-	-
Ejecución de pruebas de interfaz	-	-	-	-

9.3 Cronograma

9.3.1 Especificación de casos de pruebas

N° Caso	Especificación pruebas
1	Nombre: PF1 Ingreso sesión Administrador Objetivos: Un administrador ingresa al sistema. Pruebas a realizar: <ul style="list-style-type: none">- T1 crear un usuario administrador.- T2 ingresar con el administrador creado antes.- T3 ingresar con un administrador no existente.
2	Nombre: PF2 Agregar platillo Objetivos: Un administrador agrega un platillo al sistema. Pruebas a realizar: <ul style="list-style-type: none">- T1 ingresar un platillo, poniendo toda la información pedida.- T2 ingresar un platino, no poniendo toda la información pedida.- T3 ingresar un platillo, poniendo información errónea.
3	Nombre: PF3 Ver lista de platillos (Administrador) Objetivos: Un administrador visualiza la lista de platillos completa. Pruebas a realizar: <ul style="list-style-type: none">- T1 ver la lista de platillos, con el contenido actual de la lista en el menú principal (se espera ver los platillos propios de PF2).
4	Nombre: PF4 Ingreso sesión usuario Objetivos: Un usuario ingresa al sistema. Pruebas a realizar: <ul style="list-style-type: none">- T1 crear un usuario administrador.

	<ul style="list-style-type: none"> - T2 ingresar con el usuario anterior. - T3 ingresar con un usuario no existente.
5	Nombre: PF5 Ver lista de platillos (cliente) Objetivos: Un usuario visualiza la lista de platillos completa. Pruebas a realizar: <ul style="list-style-type: none"> - T1 ver la lista de platillos, con el contenido actual de la lista en el menú principal.
6	Nombre: PF6 Realizar pedido Objetivos: Un usuario realiza un pedido en el programa Pruebas a realizar: <ul style="list-style-type: none"> - T1 el usuario selecciona un platillo de la lista disponible, e intenta realizar la compra.
7	Nombre: PF7 Atender pedido Objetivos: Un administrador atiende un pedido realizado previamente. Pruebas a realizar: <ul style="list-style-type: none"> - T1 el administrador despliega la lista de pedidos activos. - T2 el administrador acepta un pedido. - T3 el administrador rechaza un pedido.

N° Caso	N° Prueba	Especificación pruebas
1		Descripción: Ingresar administrador Requisitos probados: - Objetivo: Validar únicamente ingresos registrados en la base de datos. Precondiciones: - Procedimiento de prueba: T1 - T2 - T3
	T1	Prueba a realizar: Se ponen datos propios de un administrador a utilizar en adelante. Estímulo (acciones a realizar): <ul style="list-style-type: none"> - Se presiona el botón para poder acceder. - Se ingresan los datos para guardar. Respuesta esperada: <ul style="list-style-type: none"> - Se regresa a la interfaz inicial.
	T2	Prueba a realizar: Validar el ingreso correcto. Estímulo (acciones a realizar): <ul style="list-style-type: none"> - Se pone la información escrita previamente. Respuesta esperada: <ul style="list-style-type: none"> - Se accede a la ventana principal de la aplicación con la lista de platillos.
	T3	Prueba a realizar: Validar el ingreso correcto. Estímulo (acciones a realizar): <ul style="list-style-type: none"> - Se pone la información escrita previamente, de forma errónea. Respuesta esperada:

		<ul style="list-style-type: none"> - Se niega el acceso, hasta que las credenciales coinciden con algún administrador en la base de datos.
--	--	---

N° Caso	N° Prueba	Especificación pruebas
2		Descripción: Agregar platillo Requisitos probados: Objetivo: Agregar información de platillos a la base de datos. Precondiciones: PF1 Procedimiento de prueba:
	T1	Prueba a realizar: Ingresar platillo correctamente Estímulo (acciones a realizar): <ul style="list-style-type: none"> - Se presiona el botón para agregar platillo. - Se rellenan los campos requeridos del platillo. - Se presiona el botón para finalizar. Respuesta esperada: <ul style="list-style-type: none"> - El platillo se visualiza en la lista de platillos disponibles en la ventana principal.
	T2	Prueba a realizar: Ingresar platillo incorrectamente (datos faltantes) Estímulo (acciones a realizar): <ul style="list-style-type: none"> - Se presiona el botón para agregar platillo. - Se rellenan los campos requeridos del platillo, faltando al menos uno. - Se presiona el botón para finalizar. Respuesta esperada: <ul style="list-style-type: none"> - Aparece un mensaje indicando que el ingreso no es válido, que falta información.
	T3	Prueba a realizar: Ingresar platillo erróneamente (Datos incorrectos) Estímulo (acciones a realizar): <ul style="list-style-type: none"> - Se presiona el botón para agregar platillo. - Se rellenan los campos requeridos del platillo, poniendo un ingreso erróneo como letras en el precio. - Se presiona el botón para finalizar. Respuesta esperada: <ul style="list-style-type: none"> - Aparece un mensaje indicando el ingreso erróneo y qué dato no es correcto.

N° Caso	N° Prueba	Especificación pruebas
		Descripción: Requisitos probados: Objetivo:

		Precondiciones: Procedimiento de prueba:
	T1	Prueba a realizar: Estímulo (acciones a realizar): Respuesta esperada:

9.5 Dependencias y riesgos

En estos momentos, se requiere de la plataforma Jenkins para poder compartir públicamente el programa. Una vez se dé por finalizado el desarrollo (en su totalidad), se espera que esta dependencia sea eliminada.

En efectos prácticos, la aplicación no presenta ningún sistema que resguarde los datos de algún externo que logre acceder a los mismos, por lo que en estos momentos estos son vulnerables. Esto se remediara en futuras iteraciones del desarrollo del software, antes de su entrega a la empresa referenciada.

10. Referencias

Información relacionada a las herramientas utilizadas:

- <https://angular.io/>
- <https://www.primefaces.org/primeng/showcase/#/>