**Plan de Pruebas**

Índice

[Propósito 2](#_Toc524096214)

[Alcance de las pruebas 2](#_Toc524096215)

[Descripción 2](#_Toc524096216)

[Herramientas Utilizadas 2](#_Toc524096217)

[Ejecución de los casos de pruebas 3](#_Toc524096218)

# 

# Propósito

En el presente documento se describe el plan para probar las funcionalidades desarrolladas de DeliveryEat, de una forma general. Se espera encontrar la mayor cantidad de defectos posibles con la menor cantidad de casos de prueba, a los fines de poder darles el tratamiento adecuado y resolver el problema que se manifieste.

# Alcance de las pruebas

Estará determinado por las funcionalidades desarrolladas en cada sprint. Para ello planificaremos el esfuerzo y los recursos que se llevarán a cabo en cada sprint; describiremos cómo se realizarán las pruebas y las herramientas a utilizar.

Dentro de cada sprint identificaremos qué componentes del software serán probados, y sus respectivos casos de prueba.

# Descripción

Todos los miembros del equipo serán los encargados de realizar las pruebas, ya que se trabaja con una metodología ágil.

El proceso de testing se llevará en forma manual. No se utilizarán herramientas de automatización debido a que la puesta en marcha y capacitación en testing requiere de dos meses aproximadamente para realizar un testing acorde a las herramientas.

Las Pruebas se realizarán en los niveles:

* Unitarias: Realizada por las personas encargadas de desarrollar la User Story.
* Integración: Realizada por uno de los miembros del equipo de trabajo.
* Aceptación: Realizada por Product Owner bajo supervisión de miembros del equipo de trabajo.
* Sistema.

# Herramientas Utilizadas

Como herramientas se utilizarán las planillas de casos de prueba, particiones de equivalencia y checklists para las pruebas de historias de usuario. Nos ayudaremos mediante la GUI, entorno de base de datos y scripts de Java para visualizar si los resultados obtenidos son los esperados.

Se utilizará la estrategia de caja negra, con las siguientes técnicas:

* Partición de equivalencias.
* Pruebas de historias de usuario.

Para definir las clases de equivalencia nos guiaremos por la siguiente tabla:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Clases de equivalencia de entrada | | | | |
| Condición externa | Clases de equivalencia válidas | | Clases de equivalencia inválidas | |
|  |  |  |  |  |
|  |  |  |  |  |
| Clases de equivalencia de salida | | | | |
| Condición externa | Clases de equivalencia válidas | | Clases de equivalencia inválidas | |
|  |  |  |  |  |
|  |  |  |  |  |

# Ejecución de los casos de pruebas

Al ejecutar los casos de prueba, se registran los resultados obtenidos en una planilla de Reporte de Defectos y se realizarán capturas de pantalla para documentar y demostrar los resultados

Cantidad de ciclos de prueba: Mínimo 2, máximo 4.

Al ejecutar los casos de prueba se utilizará una planilla de check list para registrar los resultados.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID Caso de Prueba | |  |  |  |
| Nombre | |  |  |  |
| N° Clase de equivalencia | Precondiciones | Pasos | Resultados Esperados | Resultados Obtenidos |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| Nota: | |  |  |  |
| Fecha ejecución: | |  |  |  |
| Ciclo: | |  |  |  |
| Estado (Pasa/Falla): | |  |  |  |
| Ejecutante: | |  |  |  |

Para el reporte de los defectos utilizaremos la siguiente planilla

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Reporte de defectos | | | | | | | |
| ID del bug | CP asociado | Fecha de creación | Descripción | Pasos para reproducirlo | Severidad | Prioridad | Estado |
| 1 |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |