**Plan de pruebas de software Fase 1**

***[QA sistema de compras Online]***

***Fecha: [18/04/2020]***

### Integrantes:

* Karen Leyton.
* Christian Villarroel.

**Tabla de contenido**

Historial de Versiones 4

Información del Proyecto 4

Aprobaciones 4

Resumen Ejecutivo 5

Alcance de las Pruebas 5

Elementos de Pruebas 5

Nuevas Funcionalidades a Probar 6

Pruebas de Regresión 6

Funcionalidades a No Probar 7

Enfoque de Pruebas (Estrategia) 7

Criterios de Aceptación o Rechazo 8

Criterios de Aceptación o Rechazo 8

Criterios de Suspensión 8

Criterios de Reanudación 9

Entregables 9

Recursos 10

Requerimientos de Entornos – Hardware 10

Requerimientos de Entornos – Software 10

Herramientas de Pruebas Requeridas 11

Personal 11

Entrenamiento 12

Planificación y Organización 12

Procedimientos para las Pruebas 12

Matriz de Responsabilidades 13

Cronograma 13

Premisas 14

Dependencias y Riesgos 14

Referencias 15

Glosario 15

# Historial de versiones

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fecha** | **Versión** | **Autor** | **Organización** | **Descripción** |
| **18/04** | **0.0** | **Christian Villarroel** | **DuocUC** | **Revisión inicial.** |
| **19/04** | **1.0** | **Karen Leyton** | **DuocUc** | **Revisión punto 1.** |
|  |  |  |  |  |

# Información del proyecto

|  |  |
| --- | --- |
| Empresa / Organización | Duoc Software Solution |
| Proyecto | QA Sistema de compras online |
| Fecha de preparación | 17/04/2020 |
| Cliente | Computer Sail Online |
| Patrocinador principal | Duoc Software Solution |
| Gerente / Líder de proyecto | Christian Villarroel |
| Gerente / Líder de pruebas de software | Karen Layton |

# Aprobaciones

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre y Apellido** | **Cargo** | **Departamento u organización** | **Fecha** | **Firma** |
| **Karen Leyton** | **Lider de pruebas de software** | **QA** | **18/04** | **karen.leyton** |
| **Christian Villarroel** | **Lider de proyecto** | **Desarrollo** | **19/04** | **christian.villarroel** |
|  |  |  |  |  |
|  |  |  |  |  |

# Resumen ejecutivo

El propósito principal del presente documento es entregar las pautas y definir la estrategia para llevar a cabo el aseguramiento de calidad del software sistema de compras online de Computer sail Online, con la finalidad de de que se puedan identificar la mayor cantidad de errores y defectos que puedan existir dentro del software con el fin de corregirlos, verificar que el software cumpla con el comportamiento esperado y corroborar que el sistema de cumplimiento a los requerimientos establecidos por el cliente y que se tenga un rendimiento óptimo dentro del ambiente de prueba.

La ejecución de las pruebas se ha dividido por fases, para la primera se considera el testing en relación a las principales funcionalidades del sistema, definidas por los requerimientos funcionales del proyecto. Estas serán probadas mediante técnicas de caja gris abocadas a los requerimientos funcionales del sistema y test unitarios.

Para la segunda fase se realizarán pruebas de regresión sobre los componentes modificados en caso de que se hubiesen detectado defectos en el software, y se diseñarán las pruebas de los requerimientos no funcionales y test unitarios para los módulos faltantes. Si fuera necesario se volverá a iterar este proceso.

# Alcance de las pruebas

## Elementos de pruebas

Las pruebas a realizar tendrán por objetivo principal la revisión de las funcionalidades principales del software, agrupadas en los siguientes módulos:

1. LogIn de usuario.
2. Agregar producto
3. Modificar producto
4. Eliminar producto
5. Buscar producto
6. Agregar empleado
7. Modificar empleado
8. Eliminar empleado
9. Realizar compra

además se deben considerar pruebas de regresión de estos módulos, y de elementos visuales del software como la navegación y redirección entre componentes del sistema.

## Nuevas funcionalidades a probar

No se contempla el testing de nuevas funcionalidades dentro de las pruebas.

## Pruebas de regresión

En lo que respecta a las pruebas de regresión, se realizarán pruebas a los módulos principales del software según los casos de prueba asociados. Estas serán ejecutadas por el equipo de QA al comienzo del siguiente ciclo de pruebas, cuando se realicen correcciones a errores o defectos.

## Funcionalidades a no probar

No se probara que los cálculos se realizan correctamente ni que los productos disponibles se encuentren correctamente registrados en la base de datos previo a la entrega del software, debido a que se considera como parte de la responsabilidad del equipo de desarrollo y la toma de requerimientos.

se asume que de existir errores de este tipo, estos son menores y no afectarán gravemente la aplicación, ya que podrán ser subsanados con las funcionalidades del software.

## Enfoque de pruebas (estrategia)

En cuanto al enfoque de las pruebas, se realizarán 2 niveles de pruebas para el aseguramiento de calidad del sistema, consistente en pruebas unitarias y pruebas funcionales. Para estas últimas se utilizara la técnica de caja gris y pruebas de carácter manual.

1. Pruebas funcionales.

Se requiere que para la ejecución de las pruebas funcionales, estas sean realizadas en el ambiente de testing y que se genere evidencia de la ejecución de los casos de prueba por cada punto de instrucción de la prueba.

* Pruebas de caja negra.

Se ejecutarán con el objetivo de medir la funcionalidad del sistema, velando por el cumplimiento de los requisitos de comportamiento del software.

* + Pruebas de casos de uso.

Para las pruebas de casos de uso se debe establecer el módulo a probar, la descripción con todo lo que se debe hacer para ejecutar la prueba, el dato con el que se realizará la prueba y el resultado esperado de esta.

Además se realizarán pruebas de navegación para verificar que las redirecciones dentro del software y la integración entre los distintos componentes se esté realizando correctamente.

* + Pruebas de partición de equivalencia.

Se probará el ingreso de los datos no permitidos de entrada de los módulos a probar, cuando sea aplicable.

* + Análisis de valor inválido.

Se probará el ingreso de valores sobre los bordes permitidos, definidos por los tamaños del dato de entrada del campo.

* + Pruebas de transición de estados.

Se realizarán pruebas de navegación dentro del flujo del software.

* Pruebas de caja blanca.

Se analizará las estructura del código mediante pruebas de decisión de los componentes principales del sistema.

* + Pruebas de decisión.
  + Pruebas de instrucción.

1. Pruebas unitarias.

En cuanto a las pruebas unitarias, estas serán diseñadas e implementadas con Junit 5, en los módulos producto y empleado mediante Test Suites:

1. ProductoDAO.java

* Buscar producto.
* Eliminar producto.
* Agregar producto.

1. EmpleadoDAO.java

* Agregar empleado.
* Actualizar empleado.
* Eliminar empleado.

Además se contempla la creación de test unitarios por parte de los desarrolladores dentro de los elementos que afectan sus desarrollos.

# Criterios de aceptación o rechazo

## Criterios de aceptación o rechazo

Para poder aceptar el proyecto como un MVP, se deben garantizar los siguientes puntos:

1. Ejecutar un 100% de las pruebas funcionales y unitarias.
2. Contar con un 100% éxito en las pruebas unitarias y funcionales.
3. Tener al menos un 20% de éxito en cuanto a las pruebas no funcionales.

## Criterios de suspensión

La ejecución de las pruebas deberá ser suspendido cuando se presente algún error o defecto que afecte el comportamiento de un módulo principal asociado a un requerimiento funcional o que impida la ejecución de otros casos de prueba.

* Defecto o error crítico que detiene el funcionamiento de la aplicación.
* Defecto o error grave que afecte la funcionalidad completa de un módulo.
* Defecto o error grave en grabado de datos en la base de datos.

El equipo de Q/A generará un reporte de error o defecto previo a la suspensión de pruebas, donde se detallaran los pasos para replicarlo, los componentes que afecta, logs y en qué módulo se encuentra para que este sea solucionado.

En cuanto a los motivos externos a la ejecución de las pruebas, estas se suspenderán si existen demoras significativas en la entrega del software para que sea probado, tanto en el inicio de las pruebas como entre ciclos de pruebas. Ya que esto genera conflictos y acorta el tiempo para el ciclo de pruebas.

## Criterios de reanudación

Las pruebas se reanudarán una vez sea corregido el defecto o error reportado por el equipo de QA y serán ejecutadas para nuevamente para todos los módulos que sean afectados en la solución.

# Entregables

Como parte de la ejecución del plan de pruebas para esta primera fase se entregarán los siguientes documentos.

* Documentación de plan de pruebas según fase.
* Plantilla de casos de prueba.
* Evidencia de ejecucion de caso de prueba.
* Evidencia de implementación de JUnit test.
* Reportes de bug o defectos.
* Logs de errores

# Recursos

## Requerimientos de entornos – Hardware

Para poder llevar a cabo las actividades relacionadas con las pruebas, los integrantes del equipo de QA deben contar con equipos que cumplan las siguientes características:

* Procesador: Intel i3 o superior
* Memoria RAM: 4GB o superior
* Almacenamiento: 15 GB o superior
* Conexión a internet alámbrica o inalámbrica de al menos 512 kbps.
* Periféricos (mouse y teclado).

## Requerimientos de entornos – Software

Para poder llevar a cabo las actividades relacionadas con las pruebas, los ambientes de QA deberán contar con los siguientes programas:

* Netbeans JEE 8 o superior con los siguientes plugins:
  + servidor GlassFish.
  + JUnit 5.
* Xampp versión 7.3 o superior.
* Java 8 JRE/JDK
* Microsoft office.
* Visual Studio Code.

## Herramientas de pruebas requeridas

Las herramientas de pruebas que se emplearán serán:

* Automatización de pruebas.
  + Carina.
* Server de automatización.
  + Jenkins.
* Controlador de versiones.
  + Git, GitHub.
* Editor de código.
  + Visual Studio Code.

## Personal

Para completar las actividades de pruebas se requiere del siguiente personal:

|  |  |
| --- | --- |
| Cantidad. | Rol. |
| 1 | Lider de pruebas. |
| 1 | Q/A |

## Entrenamiento

Se debe instruir al personal de Q/A sobre la creación de evidencia y ejecución de casos de prueba, así como el reporte de errores y otra documentación. Además se debe capacitar a estos en el uso de las siguientes herramientas:

* Selenium, <https://www.selenium.dev/>
* Jenkins, <https://jenkins.io/>
* GitHub, <https://github.com/>
* Git, <https://git-scm.com/>
* Junit, <https://junit.org/junit5/>

# Planificación y organización

## Procedimientos para las pruebas

Las pruebas se desarrollarán con el objetivo de lograr un modelo ágil de testing, para esto se deberá integrar la creación de test unitarios para el equipo de desarrollo y la automatización de las pruebas. Para poder lograr esto se deberán realizar las siguientes actividades ligadas a cada prueba.

* Analizar el módulo sobre el cual se va a probar.
* Identificar el flujo de la aplicación.
* Determinar la complejidad ciclomática.
* Determinar si es posible automatizar la prueba.
* Reporte de evidencia de prueba.

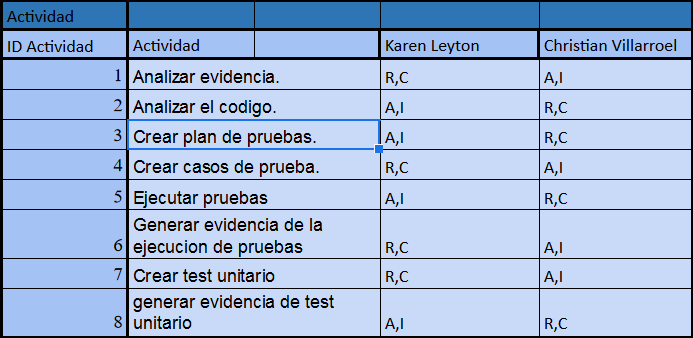
Debe contar con los siguientes elementos:

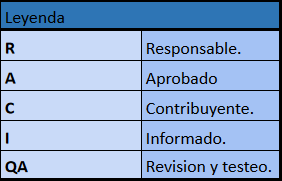
* + datos de entrada.
  + objetivo de la prueba.
  + descripción de los pasos.
  + módulo que se va a probar.
  + capturas de entrada
  + salida de puntos de ejecución.
* Reporte de evidencia de error o defecto.

En caso de que se encuentre algún error o defecto, este debe ser reportado al equipo de desarrollo para que sea subsanado. El reporte debe contar con:

* + Descripción del error o defecto.
  + Resultado esperado.
  + Pasos para reproducción.
  + Logs relevantes.
  + Configuraciones en base de datos.
  + Evidencia de error.

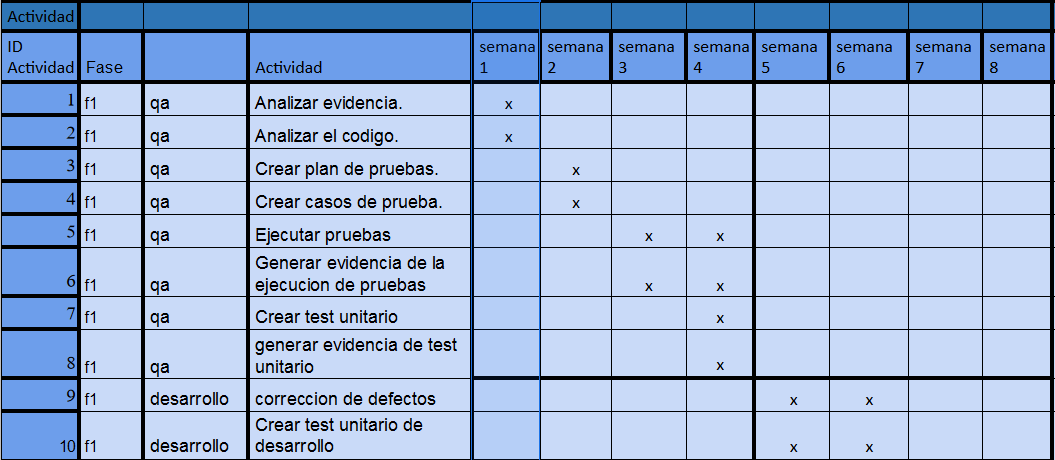
## Matriz de responsabilidades





## Cronograma

Se adjunta imagen con Fase1 del cronograma y archivo excel con el cronograma completo.



## Premisas

Para que el plan de prueba se lleve a cabo, se deben cumplir los siguientes supuestos:

1. Disponibilidad de ambientes de Q/A.
2. Capacidad técnica de los miembros del equipo de Q/A.

## Dependencias y Riesgos



# Referencias

1. Evidencia de producto por encargo.
2. Planilla de casos de prueba.
3. Matriz R.A.C.I.
4. Planilla Cronograma.

# Glosario

|  |  |
| --- | --- |
| Término | Definición |
| QA | Quality Assurance, ingeniero de calidad. |
| Desarrollo | Codificación de la aplicación o equipo desarrollador de sistema informático. |
| Software | Sistema informático. |
| Testing | Pruebas realizadas al software. |
| Caja gris | Método de prueba que integra caja negra y caja blanca. |
| Test unitario | Pruebas de módulos o funciones realizadas por unidades de código. |
| LogIn | Ventana de ingreso de credenciales. |
| MVP | Mínimo producto Viable. |
| Módulo | Se refiere a una parte del software que realiza una o varias tareas de las que realiza el sistema según sus requisitos. |
| Log / Logs | Registro de texto de los acontecimientos cronológicos en la ejecución del software. |