

Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Análisis y Diseño de Sistemas 2
Sergio Arnaldo Méndez

Plan de pruebas

Julio César Osorio Boror 201020177
Marlen Paola Arias Salguero 201020381
Ariel Francisco Montejo Dominguez 201020716

Estrategias de pruebas

Elementos de prueba

Pruebas de Base de Datos

- Prueba de conexión.

Pruebas funcionales

- Verificar la inserción de productos al carrito de compra.
- Verificar la carga del listado de productos.
- Verificar el logueo de usuario.
- Verificar el registro de usuario.

Pruebas de la interfaz de usuario

- Navegación a través de todos los casos de uso, verificando que cada panel de interfaz de usuario puede ser fácil de entender.

Pruebas de carga

- Ninguno.

Seguridad y pruebas de control de acceso

- Verificar el logueo de un usuario registrado.

Failover / Recuperación de Pruebas

- Ninguno.

Pruebas de configuración

- Ninguno.

Prueba de la instalación

- Ninguno.

Pruebas de Compatibilidad con navegadores.

- Verificación del funcionamiento en distintos navegadores y sus distintas versiones.

Pruebas de compatibilidad con sistemas operativos

- Verificación de funcionamiento en distintos sistemas operativos.

Herramientas para las pruebas

Pruebas unitarias

JUnit, debido a que se adapta perfectamente con las aplicaciones Java, con una sintaxis muy sencilla de utilizar, por lo que es la herramienta seleccionada.

Pruebas de integración

Easyb es una herramienta de pruebas de integración o módulos basada en el lenguaje Groovy, que se utiliza para realizar tests en Java, un lenguaje sencillo de utilizar, permite realizar escenarios en los que se incluyen distintos componentes del sistema y se especifican una serie de pasos que el programa debe cumplir hasta llegar a un resultado final, por su fácil aprendizaje es la herramienta seleccionada.

Pruebas de compatibilidad

Java es un lenguaje multiplataforma, solo se necesita tener instalado el core de Java en los sistemas operativos y navegadores para asegurar su funcionamiento, pero debido a la gran variedad de estos en algunos casos podría fallar, por lo que se utilizará BrowserShots, una herramienta online para pruebas de compatibilidad, permite escoger una gran gama de combinaciones entre SO y navegador, además su uso es muy sencillo.

Criterios de evaluación

Pruebas de Base de Datos

- Se realizará una prueba de conexión con las credenciales de la base de datos, si no se encuentra ningún error, se aceptará la prueba.

Pruebas funcionales

- Se realizarán pruebas unitarias bajo ciertas condiciones especiales, como ingresar los mismos datos, garantizando que la prueba sea la misma en todas las ejecuciones y si el resultado es satisfactorio, se aceptará la prueba.

Pruebas de la interfaz de usuario

- Se consultará con los beta tester y con los clientes, si ellos aceptan la interfaz y la forma en que funciona, entonces se aceptará la prueba.

Seguridad y pruebas de control de acceso

- Se ingresará valores incorrectos, o valores que pueden inducir al error. Si el resultado de la prueba es satisfactorio, se acepta la prueba.

Pruebas de integración

- Obtener los resultados esperados en cada escenario de pruebas.

Pruebas de compatibilidad.

- Obtener un resultado positivo en la mayor cantidad de sistemas operativos y navegadores posibles, en su defecto en los más actuales y utilizados.

Plantillas de pruebas

Plantilla de resultado

Objetivo:	Se escribe el objetivo de la prueba a realizar.
Técnica:	Se realiza una breve descripción acerca de la técnica a aplicar, que se evaluará y cómo se ejecutará.
Herramientas:	Se realiza un listado de todas las herramientas involucradas en la prueba.
Criterios de aceptación:	Se realiza un listado de cada criterio de aceptación de la prueba realizada.
Consideraciones especiales:	Se detalla algún detalle relevante de la prueba para la ejecución despliegue de resultados.

Plantilla de prueba unitaria

```
public void testEjemplo() {  
    Prueba p = Prueba.getValor("1");  
    assertEquals(1, p);  
}
```

Plantilla de prueba de integración

```
scenario "Escenario de prueba", {  
    Usuario d;  
  
    given "Un nuevo usuario, se utiliza el módulo usuarios", {  
        d = new Usuario();  
    }  
  
    when "Se utiliza el módulo compras para registrar una compra", {  
        Compras c = new Compras();  
        c.Comprar(d,10,1);  
    }  
  
    then "El usuario debe tener registrada una compra", {  
        d.compras().shouldBe 1  
    }  
}
```

}

Plantilla de prueba de compatibilidad

Interfaz de usuario de <http://browsershots.org/>, herramienta seleccionada para estos tests.

