# Plantilla para Pruebas de Regresión

## Información General

Nombre del Proyecto: Sociedad-Comercial-Correa-Correa

Versión del Software: 1.0.0

Fecha: 14/12/2024

Responsable de las Pruebas: Victor Manuel Martinez Soto

Herramientas Utilizadas: Visual Studio 2022

## Objetivo de las Pruebas de Regresión

Validar que el sistema y sus funcionalidades críticas continúan funcionando correctamente después de modificaciones, actualizaciones o correcciones.

## Casos de Prueba

| ID Caso de Prueba | Descripción | Módulo/Funcionalidad | Pasos | Resultado Esperado | Resultado Obtenido |
| --- | --- | --- | --- | --- | --- |
| CP01 | Prueba del debounce en el inicio de sesión para evitar clics rápidos duplicados. | Iniciar Sesión | 1. Abre el software y accede a la ventana de inicio de sesión.  2. Ingresa tu nombre de usuario y clave.  3. Haz clic en el botón "Ingresar" varias veces de forma rápida. | Solo una solicitud al servidor debe ejecutarse, sin duplicación | Las solicitudes se procesan correctamente con el debounce implementado. |

## Configuración del Entorno

Sistema Operativo: Window 10

Base de Datos: Sql Server 2022

Versión del Software: 1.0.0

## Resultados Generales

| Número Total de Pruebas Ejecutadas | 20 |
| --- | --- |
| Número de Pruebas Exitosas | 20 |
| Número de Pruebas Fallidas | 0 |
| Errores encontrados | 0 |
| Tareas pendientes | 0 |

## Conclusión

Se validó que el sistema y sus funcionalidades críticas continúan funcionando correctamente después de implementar el debounce en el inicio de sesión. No se detectaron errores al realizar clics rápidos, ya que el debounce garantizó que solo una solicitud al servidor se ejecutará.

Los cambios implementados mejoraron la estabilidad y robustez de la autenticación, evitando problemas de duplicación y sobrecarga innecesaria. Se logró una comunicación eficiente con la base de datos, manteniendo un rendimiento consistente y confiable.

Recomendaciones:

Realizar pruebas periódicas de regresión cada vez que se realicen actualizaciones o correcciones.

Implementar pruebas automatizadas para verificar el comportamiento del debounce y otras funcionalidades críticas.

Optimizar otras interacciones del sistema que puedan verse afectadas por el flujo de usuario, garantizando la fluidez y rapidez del acceso entre módulos y ventanas.