

**Medic Inventory**

Planes de Pruebas Funcionales

Versión: 0100

Fecha: 24/11/2023

[1.0.1]

**HOJA DE CONTROL**

|  |  |  |  |
| --- | --- | --- | --- |
| **Organismo** | <Nombre Consejería u Organismo Autónomo><Nombre Consejería u Organismo Autónomo> | | |
| **Proyecto** | Medic Inventory | | |
| **Entregable** | Planes de Pruebas Funcionales | | |
| **Autor** | Jose Riaño  Felipe Forero  Ronaldo Barragán  Sergio Gómez  Javier Cortés | | |
| **Versión/Edición** | 0100 | **Fecha Versión** | 24/11/2023 |
| **Aprobado por** |  | **Fecha Aprobación** | 25/11/2023 |
|  |  | **Nº Total de Páginas** |  |

REGISTRO DE CAMBIOS

|  |  |  |  |
| --- | --- | --- | --- |
| **Versión doc** | **Causa del Cambio** | **Responsable del Cambio** | **Fecha del Cambio** |
| 0100 | Versión inicial | Jose Riaño | 25/11/2023 |
|  |  |  |  |
|  |  |  |  |

CONTROL DE DISTRIBUCIÓN

|  |
| --- |
| **Nombre y Apellidos** |
| Jose Riaño |
| Felipe Forero |
| Ronaldo Barragán |
| Sergio Gómez |
| Javier Cortés |

**ÍNDICE**

[1 INTRODUCCIÓN 4](#_1fob9te)

[1.1 Objeto 4](#_3znysh7)

[1.2 Alcance 4](#_2et92p0)

[2 TRAZABILIDAD DE CASOS DE PRUEBAS – REQUISITOS 5](#_tyjcwt)

[3 DEFINICIÓN DE LOS CASOS DE PRUEBAS 6](#_3dy6vkm)

[4 ESTRATEGIA DE EJECUCIÓN DE PRUEBAS 7](#_1t3h5sf)

[5 ANEXOS 8](#_4d34og8)

[6 GLOSARIO 9](#_2s8eyo1)

[7 BIBLIOGRAFÍA Y REFERENCIAS 10](#_17dp8vu)

# INTRODUCCIÓN

## Objeto

Desarrollar las pruebas unitarias necesarias para comprobar el buen funcionamiento del sistema que fue establecido con los requisitos específicos. Se establecerán las pruebas unitarias bajo la necesidad de revisar el CRUD del sistema de los diferentes modelos que presenta el software establecido.

* Determinar el software el cual ayudará a realizar las pruebas unitarias con el modelo de caja negra
* Realizar las peticiones correspondientes con el bot Selenium para el CRUD de los módulos que presenta el software
* Emplear acciones de mejora con los resultados dados por el bot

## Alcance

El alcance que se establece para el desarrollo de las pruebas unitarias en el software Medic Inventory son los requisitos funcionales y los no funcionales estén en orden y funcionamiento continuo, por esto mismo se determina que los módulos como usuario, pedidos, medicamentos, devoluciones y reportes deben ser sometidos a exhaustivas pruebas unitarias. Cada uno de estos módulos es fundamental para el funcionamiento integral de Medic Inventory. El módulo de usuario garantiza la gestión adecuada de accesos y permisos, asegurando la seguridad del sistema. Los pedidos y el manejo de inventario son esenciales para el flujo de trabajo, por lo que su correcto funcionamiento es crucial para evitar inconvenientes en la cadena de suministro médico.

Los medicamentos, al ser el corazón de la aplicación, deben ser probados minuciosamente para garantizar la precisión en su registro, actualización y manejo. Las devoluciones, por su parte, requieren un tratamiento específico para asegurar que se gestionen adecuadamente y se actualice el inventario en consecuencia. Finalmente, los reportes, que proveen información vital para la toma de decisiones, deben ser rigurosamente evaluados para garantizar su exactitud y disponibilidad constante.

El alcance establecido prioriza la integridad y funcionalidad de estos módulos, asegurando que Medic Inventory opere con eficiencia y precisión, cumpliendo con los estándares de calidad requeridos en el ámbito médico.

# TRAZABILIDAD DE CASOS DE PRUEBAS – REQUISITOS

En este apartado se establecen los requisitos funcionales que serán puestos a prueba mediante las pruebas unitarias establecidas para los CRUD correspondientes de cada uno de los módulos que fueron generados con base en los requisitos funcionales.

RF-001: Iniciar Sesión  
RF-002: Registro de Usuarios

RF-003: Medicamentos

RF-004: Reportes

RF-005: Creación de ordenes

RF-006: Devoluciones

RF-007: Registro de Sucursales

RF-008: Subcategoría

RF-009: Registro de proveedores

RF-010: Categorías

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **RF-001** | **RF-002** | **RF-003** | **RF-004** | **RF-005** | **RF-006** | **RF-007** | **RF-008** | **RF-009** | **RF-010** |
| **<CP-1>** | **X** |  |  |  |  |  |  |  |  |  |
| **<CP-2>** | **X** | **X** | **X** |  |  |  |  |  |  |  |
| **<CP-3>** | **X** | **X** | **X** |  |  |  | **X** |  |  |  |
| **<CP-4>** |  |  |  |  |  |  |  |  |  |  |
| **<CP-5>** |  |  |  |  |  |  |  |  |  |  |
| **<CP-6>** |  |  |  | **x** |  |  |  |  |  |  |
| **<CP-7>** |  |  |  | **x** |  |  |  |  |  |  |
| **<CP-8>** |  |  |  | **x** |  |  |  |  |  |  |
| **<CP-9>** |  |  |  | **x** |  |  |  |  |  |  |
| **<CP-10>** |  |  |  | **x** |  |  |  |  |  |  |
| **<CP-11>** |  |  |  | **x** |  |  |  |  |  |  |
| **<CP-12>** |  |  |  | **x** |  |  |  |  |  |  |
| **<CP-13>** |  |  |  | **x** |  |  |  |  |  |  |

# DEFINICIÓN DE LOS CASOS DE PRUEBAS

Se definirán los casos de prueba para el software desarrollado como aquellos los cuales buscaran o pondrán a prueba los diferentes módulos que hacen que el sistema de inventarios funcione de manera correcta, esto se realizará con el programa Selenium que grabara los pasos a realizar en el sistema y los ejecuta para verificar el correcto funcionamiento.

|  |  |  |
| --- | --- | --- |
| **Inicio de Sesión y verificación de datos ingresados** | **<Código del CP>** | **CP-1** |
| **¿Prueba de despliegue?** | **Si/No** |
| **Descripción:**  Este caso de prueba evalúa el proceso de inicio de sesión del sistema y la verificación de los datos ingresados por el usuario. | | |
| **Prerrequisitos**  1 El sistema está desplegado y accesible.  2 El usuario tiene credenciales válidas (correo y contraseña). 3 Ingresar los datos de manera correcta. | | |
| **Pasos:**  1 Ingresar al sistema utilizando las credenciales del usuario. 2 Estar en el apartado de Login  3 Ingresar Correo y contraseña 4 hacer clic en el botón inicio de sesión 5 Verificar que la página de inicio se cargue correctamente 6 Comprobar que la información del usuario(Correo, Contraseña) sea la esperada | | |
| **Resultado esperado:**  Después de completar los pasos, se espera que el usuario sea redirigido a la página de inicio con la información correcta del usuario. | | |
| **Resultado obtenido:**  La página de inicio se cargó correctamente, se verificó que la información del usuario ingresado sean las correctas. | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Creación de Categoría, Subcategoría y Medicamento, además de la actualización de la información** | **Código del CP** | **CP-2** |
| **¿Prueba de despliegue?** | **Si** |
| **Descripción:**  Se entrará a la página web del sistema Medic Inventory como administrador y se creará además de editar la categoría, subcategoría y medicamento | | |
| **Prerrequisitos**  1 no tener registrado la categoría, la subcategoría y el medicamento que se va a crear  2 establecer una conexión como administrador en la página web  3tener la información correspondiente sobre la categoría, subcategoría y el medicamento a crear | | |
| **Pasos:**  1 Dar clic en el apartado Categoría.  2 Dar clic en el apartado agregar nueva Categoría  3 Ingresar los datos correspondientes  4 Dar clic en agregar subcategoría  5 Dar clic en el apartado subcategoría.  6 Dar clic en el apartado agregar nueva subcategoría  7 Ingresar los datos correspondientes  8 Dar clic en agregar subcategoría  9 Dar clic en el apartado Medicamentos.  10 Dar clic en la Categoría y subcategoría en la cual desea agregar un medicamento  11 Ingresar los datos correspondientes  12 Dar clic en agregar Categoría  13 Dar clic en editar  14 Ingresar los cambios de información que están erróneos  15 Dar clic en editar Categoría  16 Dar clic en editar  17 Ingresar los cambios de información que están erróneos  18 Dar clic en editar subcategoría  19 Dar clic en Medicamentos  20 Escoger la Categoría y subcategoría donde está el medicamento  21 Ingresar los cambios de información que están erróneos  22 Dar clic en editar Medicamento | | |
| **Resultado esperado:**  Crear y editar la Categoría, subcategoría y el medicamento correspondiente sin errores en el sistema que no establezcan esta información. | | |
| **Resultado obtenido:**  Se obtuvo la creación y los cambios sobre Categoría, subcategoría y el medicamento que se proporcionaron para la prueba unitaria | | |

|  |  |  |
| --- | --- | --- |
| **Eliminación de Categoría, Subcategoría y Medicamento, además de la actualización de la información** | **Código del CP** | **CP-3** |
| **¿Prueba de despliegue?** | **Si** |
| **Descripción:**  Se entrará a la página web del sistema Medic Inventory como administrador y se eliminará la categoría, subcategoría y medicamento | | |
| **Prerrequisitos**  1 tener registrado la Categoría, la subcategoría y el medicamento que se va a eliminar  2 establecer una conexión como administrador en la página web  **3** determinar la Categoría subcategoría y el medicamento que se eliminara | | |
| **Pasos:**  1 Dar clic en el apartado Categoría  2 Escoger la categoría a eliminar  3 Dar clic en eliminar categoría  4 Dar clic en agregar subcategoría  5 Escoger la subcategoría a eliminar  6 Dar clic en eliminar subcategoría  7 Dar clic en el apartado Medicamentos.  8 Dar clic en la Categoría y subcategoría en la cual desea eliminar un medicamento  9 Escoger el medicamento a eliminar y dar clic en el botón eliminar | | |
| **Resultado esperado:**  eliminar la categoría, subcategoría y el medicamento deseado | | |
| **Resultado obtenido:**  se logra eliminar la categoría, subcategoría y el medicamento deseado | | |

|  |  |  |
| --- | --- | --- |
| **Creación de Sucursales, además de la actualización de la información** | **<Código del CP>** | **CP-04** |
| **¿Prueba de despliegue?** | **Si/No** |
| **Descripción:**  Se entrará a la página web del sistema Medic Inventory como administrador | | |
| **Prerrequisitos**  1 no tener registrado la sucursal que se va a crear  2 establecer una conexión como administrador en la página web  3tener la información correspondiente sobre la sucursal a crear | | |
| **Pasos:**  1 Dar clic en el apartado Sucursales.  2 Dar clic en el apartado agregar nueva Sucursal  3 Ingresar los datos correspondientes 4 Dar clic en agregar Sucursal 5 Dar clic en editar  6 Ingresar los cambios de información que están erróneos  7 Dar clic en editar Sucursal | | |
| **Resultado esperado:**  Crear y editar las sucursales correspondientes sin errores en el sistema que no establezcan esta información. | | |
| **Resultado obtenido:**  Se obtuvo la creación y los cambios sobre la sucursal que se proporcionaron para la prueba unitaria | | |

|  |  |  |
| --- | --- | --- |
| **Eliminación de Sucursales** | **<Código del CP>** | **CP-05** |
| **¿Prueba de despliegue?** |  |
| **Descripción:**  Entraremos a medic inventory con un usuario con rol de administrador | | |
| **Prerrequisitos**  Debe estar logueado con un usuario con rol de administrador | | |
| **Pasos:**  1 Dar clic en el apartado Sucursales.  2 Dar clic en el apartado eliminar junto a la sucursal.  3 Dar clic en “si, borrar”. | | |
| **Resultado esperado:**  La sucursal debe ser eliminada y por ende no aparecer en el listado de las Sucursales. | | |
| **Resultado obtenido:**  Se ha eliminado la sucursal satisfactoriamente. | | |

|  |  |  |
| --- | --- | --- |
| **Eliminación de Proveedores** | **<Código del CP>** | **CP-06** |
| **¿Prueba de despliegue?** |  |
| **Descripción:**  Entraremos a medic inventory con un usuario con rol de administrador | | |
| **Prerrequisitos**  Debe estar logueado con un usuario con rol de administrador | | |
| **Pasos:**  1 Dar clic en el apartado Proveedores.  2 Dar clic en el apartado eliminar junto al respectivo proveedor.  3 Dar clic en “si, borrar”. | | |
| **Resultado esperado:**  El proveedor debe ser eliminado y por ende no aparecer en el listado de los proveedores. | | |
| **Resultado obtenido:**  Se ha eliminado el proveedor satisfactoriamente. | | |

|  |  |  |
| --- | --- | --- |
| **Creación de Proveedores, además de la actualización de la información** | **<Código del CP>** | **CP-07** |
| **¿Prueba de despliegue?** | **Si/No** |
| **Descripción:**  Se entrará a la página web del sistema Medic Inventory como administrador | | |
| **Prerrequisitos**  1 no tener registrado el proveedor que se va a crear  2 establecer una conexión como administrador en la página web  3tener la información correspondiente sobre la sucursal a crear | | |
| **Pasos:**  1 Dar clic en el apartado Proveedores.  2 Dar clic en el apartado agregar nuevo Proveedor.  3 Ingresar los datos correspondientes 4 Dar clic en agregar Proveedor 5 Dar clic en editar  6 Ingresar los cambios de información que están erróneos  7 Dar clic en editar Proveedor. | | |
| **Resultado esperado:**  Crear y editar los diferentes proveedores correspondientes sin errores en el sistema que no establezcan esta información. | | |
| **Resultado obtenido:**  Se obtuvo la creación y los cambios sobre los diferentes proveedores que se proporcionaron para la prueba unitaria | | |

|  |  |  |
| --- | --- | --- |
| **Creación de Pedido, además de la actualización de la información** | **<Código del CP>** | **CP-08** |
|  |  |
| **Descripción:**  Se entrará a la página web del sistema Medic Inventory como administrador | | |
| **Prerrequisitos**  1 no tener registrado el pedido que se va a crear  2 establecer una conexión como administrador en la página web  3tener la información correspondiente sobre el pedido a crear | | |
| **Pasos:**  1 Dar clic en el apartado Pedidos.  2 Dar clic en el apartado agregar nuevo Pedido  3 Ingresar los datos correspondientes | | |
| **Resultado esperado:**  Crear y editar el Pedido correspondiente sin errores en el sistema que no establezcan esta información. | | |
| **Resultado obtenido:**  Se obtuvo un error al momento de la creación del pedido cuando se ingresan los datos correspondientes | | |

|  |  |  |
| --- | --- | --- |
| **Creación de devolución, además de la actualización de la información** | **<Código del CP>** | **CP-09** |
|  |  |
| **Descripción:**  Se entrará a la página web del sistema Medic Inventory como administrador | | |
| **Prerrequisitos**  1 no tener registrado el pedido que se va a crear  2 establecer una conexión como administrador en la página web  3tener la información correspondiente sobre el pedido a crear | | |
| **Pasos:**  1 Dar clic en el apartado Devoluciones  2 Dar clic en el apartado agregar nueva devolución  3 Ingresar los datos correspondientes | | |
| **Resultado esperado:**  Crear y editar la devolución correspondiente sin errores en el sistema que no establezcan esta información. | | |
| **Resultado obtenido:**  Se obtuvo un error al momento de la creación del devolución cuando se ingresa el id del pedido | | |

|  |  |  |
| --- | --- | --- |
| **Creación de Pedido y Devolución, además de la actualización de la información** | **<Código del CP>** | **CP-10** |
| **¿Prueba de despliegue?** | **Si/No** |
| **Descripción:**  Se entrará a la página web del sistema Medic Inventory como administrador | | |
| **Prerrequisitos**  1 no tener registrado el pedido y la devolución que se va a crear  2 establecer una conexión como administrador en la página web  3tener la información correspondiente sobre el pedido y la devolución a crear | | |
| **Pasos:**  1 Dar clic en el apartado Pedidos.  2 Dar clic en el apartado agregar nuevo Pedido  3 Ingresar los datos correspondientes 4 Dar clic en agregar Pedido 5 Dar clic en el apartado de Devoluciones  6 Ingresar los datos correspondientes  7 Dar clic en agregar Devolución.  8 Dar clic en editar  9 Ingresar los cambios de información que están erróneos  10 Dar clic en editar Devolución  11 Dar clic en editar Pedido 12 Ingresar los cambios de información que están erróneos  12 Dar clic en editar Pedido | | |
| **Resultado esperado:**  Crear y editar el Pedido y la devolución correspondiente sin errores en el sistema que no establezcan esta información. | | |
| **Resultado obtenido:**  Se obtuvo la creación y los cambios sobre Pedido y Devolución que se proporcionaron para la prueba unitaria | | |

|  |  |  |
| --- | --- | --- |
| **Eliminación de Pedido y Devolución** | **<Código del CP>** | **CP-11** |
| **¿Prueba de despliegue?** |  |
| **Descripción:**  Se entrará a la página web del sistema Medic Inventory como administrador y se eliminará el pedido y la devolución | | |
| **Prerrequisitos**  1 Debe existir al menos un pedido registrado en el sistema. 2 Debe existir al menos una devolución registrada en el sistema. | | |
| **Pasos:**  1 Navegar al apartado de Pedidos  2 Seleccionar un pedido existente que se va a eliminar  3 se inicia el proceso de eliminación del pedido 4 Confirmar la eliminación cuando se solicite la confirmación. 5 verificar que el pedido se elimine correctamente del sistema  6 Navegar a la sección de devoluciones. 7 Seleccionar una devolución existente que se va a eliminar  8 dar clic en eliminar  9 se inicia el proceso de eliminación de la Devolución 10 Confirmar la eliminación cuando se solicite la confirmación.  11 verificar que el pedido se elimine correctamente del sistema | | |
| **Resultado esperado:**  Después de completar los pasos, se espera que el pedido y la devolución seleccionadas se eliminen exitosamente del sistema | | |
| **Resultado obtenido:**  se logra eliminar el pedido y la devolución deseada | | |

|  |  |  |
| --- | --- | --- |
| **Creación y edición de usuarios** | **<Código del CP>** | **CP-12** |
| **¿Prueba de despliegue?** |  |
| **Descripción:**  Entraremos a medic inventory con un usuario con rol de administrador. | | |
| **Prerrequisitos**  Debe estar logueado con un usuario con rol de administrador | | |
| **Pasos:**  1 Dar clic en el apartado Usuarios.  2 Dar clic en el apartado agregar nuevo Usuario  3 Ingresar los datos correspondientes 4 Dar clic en agregar Usuario 5 Dar clic en editar  6 Ingresar los datos correspondientes a editar  7 Dar clic en editar usuario | | |
| **Resultado esperado:**  Crear y editar el usuario sin que se generen errores. | | |
| **Resultado obtenido:**  Se ha creado y editado el usuario sin errores. | | |

|  |  |  |
| --- | --- | --- |
| **Eliminación del Usuario** | **<Código del CP>** | **CP-13** |
| **¿Prueba de despliegue?** |  |
| **Descripción:**  Entraremos a medic inventory con un usuario con rol de administrador | | |
| **Prerrequisitos**  Debe estar logueado con un usuario con rol de administrador | | |
| **Pasos:**  1 Dar clic en el apartado Usuarios.  2 Dar clic en el apartado eliminar Usuario  3 Dar clic en “si, borrar” | | |
| **Resultado esperado:**  El usuario debe ser eliminado y por ende no aparecer en el listado de los usuarios | | |
| **Resultado obtenido:**  Se ha eliminado el usuario satisfactoriamente. | | |

# ESTRATEGIA DE EJECUCIÓN DE PRUEBAS

Para este apartado se establecerán los ciclos para algunas de las pruebas unitarias que se realizaron al software Medic Inventory uno de estos ciclos sería el intento de ingreso a la página mediante un usuario correcto con su contraseña correspondiente e ir variando entre usuario y contraseñas erróneas.

Ciclo 1: En el ciclo 1 se determina el ingreso de información correcta, para los diferentes módulos llámese medicamentos, categorías, subcategorías, usuarios, pedidos, devoluciones y entre otros.

Ciclo 2: Se hacen variaciones en la información correspondiente de los módulos, pero de manera errónea

Ciclo 3: Se ingresa información verdadera junto con información errónea para los diferentes módulos llámese medicamentos, categorías, subcategorías, usuarios, pedidos, devoluciones y entre otros.

Ciclo 4: Se valida la usabilidad y la consistencia de la interfaz de la página dependiendo el módulo en el que el usuario esté.  
  
Ciclo 5: Validar la funcionalidad completa del sistema ejecutando flujos de trabajo completos que involucran múltiples módulos  
  
Ciclo 6:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **<Ciclo 1>** | **<Ciclo 2>** | **<Ciclo 3>** | **<Ciclo 4>** | **<Ciclo 5>** | **….** | **<Ciclo n>** |
| **<CP-1>** | X | X |  |  |  |  |  |
| **<CP-2>** | x |  |  |  |  |  |  |
| **<CP-3>** | x |  |  |  |  |  |  |
| **<CP-4>** | x |  |  |  |  |  |  |
| **<CP-5>** | x |  |  |  |  |  |  |
| **<CP-6>** | x | x |  |  |  |  |  |
| **<CP-7>** | x | x |  |  |  |  |  |
| **<CP-8>** | x | x |  |  |  |  |  |
| **<CP-9>** | x | x |  |  |  |  |  |
| **<CP-10>** | x | x |  |  |  |  |  |
| **<CP-11>** | x | x |  |  |  |  |  |
| **<CP-12>** | x | x |  |  |  |  |  |
| **<CP-13>** | x | x |  |  |  |  |  |

# ANEXOS

Prueba 1: Inicio de sesión y autenticación del que inicia  
Código evaluado:  
<?php

session\_start ();

include('database.php');

$error = ""; // Inicializa la variable de error

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$correo = $\_POST['correo'];

$contrasena = $\_POST['contrasena'];

$stmt = $conexion->prepare("SELECT \* FROM persona WHERE correo = :correo");

$stmt->bindParam(':correo', $correo);

$stmt->execute();

$usuario = $stmt->fetch(PDO::FETCH\_ASSOC);

if ($usuario) {

// Verifica la contraseña sin usar password\_verify

if ($contrasena === $usuario['contrasena']) {

$\_SESSION['idPersona'] = $usuario['idPersona'];

$\_SESSION['nombreP'] = $usuario['nombreP']; //le agregue este no mas para ver y saber cual es la persona que se inicia si algo se elimina y ya

$\_SESSION['correo'] = $usuario['correo'];

$\_SESSION['rol'] = $usuario['ROL\_idRol'];

// Redirigir a la página principal después de iniciar sesión

header("Location: index.php");

exit();

} else {

$error = "Credenciales incorrectas";

}

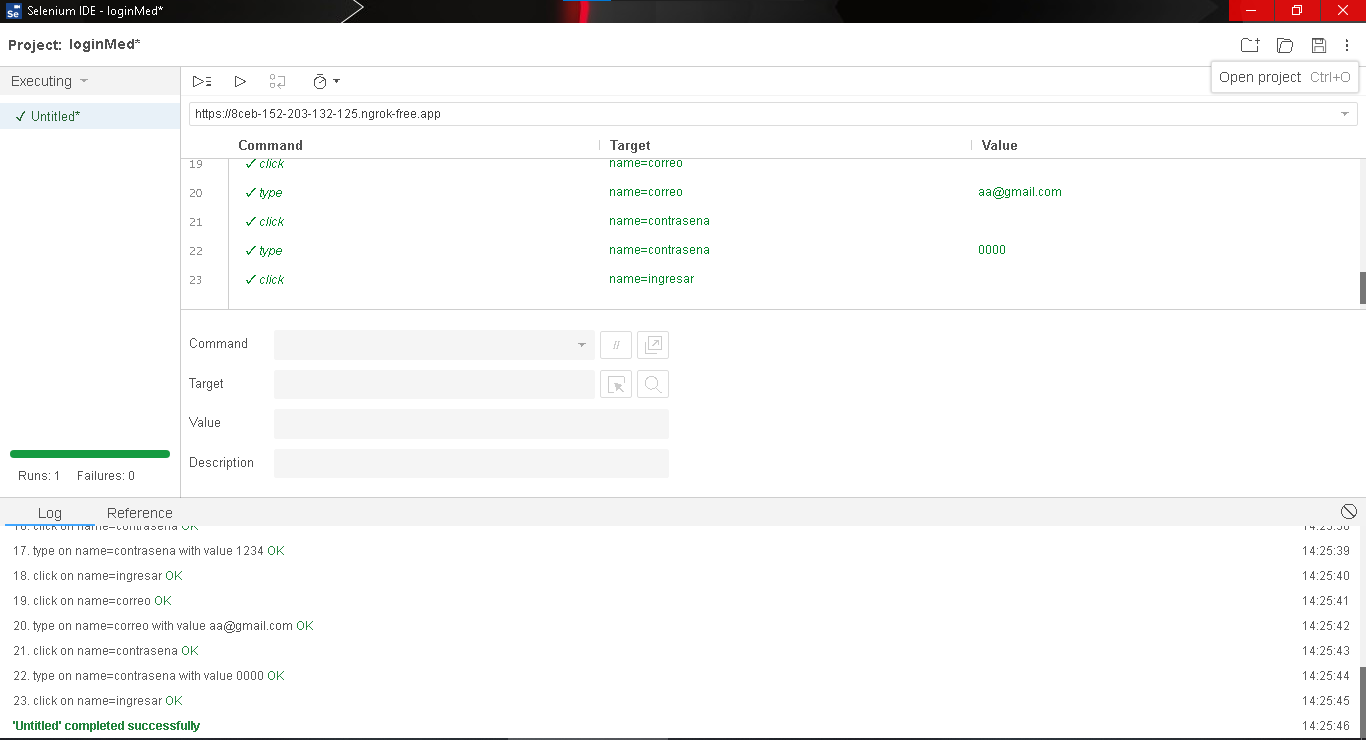
} else {

$error = "Usuario no encontrado";

}

}

?>



Prueba 2: creación de Categoría, editar Categoría, creación de la subcategoría, editar la subcategoría, creación del medicamento y editar el medicamento   
Código evaluado: (Categoria)  
<?php

include("../../database.php");

if($\_POST){

$idCATEGORIA = (isset($\_POST["idCATEGORIA"]) ? $\_POST["idCATEGORIA"] : "");

$nombreCat = (isset($\_POST["nombreCat"]) ? $\_POST["nombreCat"] : "");

$DescripcionCate = (isset($\_POST["DescripcionCate"]) ? $\_POST["DescripcionCate"] : "");

$sentencia = $conexion->prepare("INSERT INTO categoria (idCATEGORIA, nombreCat, DescripcionCate) VALUES (null, :nombreCat, :DescripcionCate)");

$sentencia->bindParam(":nombreCat", $nombreCat);

$sentencia->bindParam(":DescripcionCate", $DescripcionCate);

$sentencia->execute();

$mensaje="Registro agregado";

header("Location:index.php?mensaje=".$mensaje);

}

?>

(Editar):  
<?php

include("../../database.php");

if (isset($\_GET['txtID'])){

$txtID = (isset($\_GET['txtID'])) ? $\_GET['txtID'] : "";

$sentencia = $conexion->prepare("SELECT \* FROM categoria WHERE idCATEGORIA=:idCATEGORIA");

$sentencia->bindParam(":idCATEGORIA", $txtID);

$sentencia->execute();

$registro = $sentencia->fetch(PDO::FETCH\_ASSOC);

if ($registro) {

$nombreCat = $registro["nombreCat"];

$DescripcionCate = $registro["DescripcionCate"];

}

}

if ($\_POST){

$txtID = (isset($\_POST['txtID'])) ? $\_POST['txtID'] : "";

$nombreCat = (isset($\_POST["nombreCat"])) ? $\_POST["nombreCat"] : "";

$DescripcionCate = (isset($\_POST["DescripcionCate"])) ? $\_POST["DescripcionCate"] : "";

$sentencia = $conexion->prepare("UPDATE categoria SET nombreCat=:nombreCat, DescripcionCate=:DescripcionCate WHERE idCATEGORIA=:idCATEGORIA");

$sentencia->bindParam(":nombreCat", $nombreCat);

$sentencia->bindParam(":DescripcionCate", $DescripcionCate);

$sentencia->bindParam(":idCATEGORIA", $txtID);

$sentencia->execute();

$mensaje="Registro actualizado";

header("Location:index.php?mensaje=".$mensaje);

}

?>

Subcategoría (crear):  
<?php

include("../../database.php");

if ($\_POST) {

// Recolección de datos

$NombreSubcat = (isset($\_POST["nombreSubcat"]) ? $\_POST["nombreSubcat"] : "");

$DescripcionSubcat = (isset($\_POST["descripcionSubcat"]) ? $\_POST["descripcionSubcat"] : "");

$idCATEGORIA = (isset($\_POST["CATEGORIA\_idCATEGORIA"]) ? $\_POST["CATEGORIA\_idCATEGORIA"] : "");

// Insertar datos

$sentencia = $conexion->prepare("INSERT INTO subcategoria(idSUBCATEGORIA, nombreSubcat,descripcionSubcat, CATEGORIA\_idCATEGORIA)

VALUES (null, :nombreSubcat, :descripcionSubcat, :idCATEGORIA)");

// Asignación de valores

$sentencia->bindParam(":nombreSubcat", $NombreSubcat);

$sentencia->bindParam(":descripcionSubcat", $DescripcionSubcat);

$sentencia->bindParam(":idCATEGORIA", $idCATEGORIA);

$sentencia->execute();

$mensaje = "Registro agregado";

header("Location:index.php?mensaje=" . $mensaje);

}

$sentenciaCat = $conexion->prepare("SELECT idCATEGORIA, nombreCat FROM categoria");

$sentenciaCat->execute();

$CATEGORIA = $sentenciaCat->fetchAll(PDO::FETCH\_ASSOC);

?>

(editar):  
<?php

include("../../database.php");

$sentenciaCat = $conexion->prepare("SELECT idCATEGORIA, nombreCat FROM categoria");

$sentenciaCat->execute();

$Categorias = $sentenciaCat->fetchAll(PDO::FETCH\_ASSOC);

if (isset($\_GET['txtID'])) {

$txtID = (isset($\_GET['txtID'])) ? $\_GET['txtID'] : "";

$sentencia = $conexion->prepare("SELECT \* FROM subcategoria WHERE idSUBCATEGORIA=:idSUBCATEGORIA");

$sentencia->bindParam(":idSUBCATEGORIA", $txtID);

$sentencia->execute();

$registro = $sentencia->fetch(PDO::FETCH\_LAZY);

if ($registro) {

# code...

$NombreSubcat = $registro["nombreSubcat"];

$DescripcionSubcat = $registro["descripcionSubcat"];

$idCATEGORIA = $registro["CATEGORIA\_idCATEGORIA"];

}

}

if ($\_POST) {

// Recolección de datos

$NombreSubcat = (isset($\_POST["nombreSubcat"]) ? $\_POST["nombreSubcat"] : "");

$DescripcionSubcat = (isset($\_POST["descripcionSubcat"]) ? $\_POST["descripcionSubcat"] : "");

$CATEGORIA\_idCATEGORIA = (isset($\_POST["CATEGORIA\_idCATEGORIA"]) ? $\_POST["CATEGORIA\_idCATEGORIA"] : "");

// Insertar datos

$sentencia = $conexion->prepare("UPDATE subcategoria SET descripcionSubcat=:descripcionSubcat, nombreSubcat=:nombreSubcat, CATEGORIA\_idCATEGORIA = :CATEGORIA\_idCATEGORIA where idSUBCATEGORIA=:idSUBCATEGORIA");

// Asignación de valores

$sentencia->bindParam(":nombreSubcat", $NombreSubcat);

$sentencia->bindParam(":descripcionSubcat", $DescripcionSubcat);

$sentencia->bindParam(":CATEGORIA\_idCATEGORIA", $CATEGORIA\_idCATEGORIA);

$sentencia->bindParam("idSUBCATEGORIA", $txtID);

$sentencia->execute();

$mensaje = "Registro Actualizado";

header("Location:index.php?mensaje=" . $mensaje);

}

?>

Medicamento(crear)  
<?php

include("../../database.php");

if ($\_POST) {

// Recolectar datos (método post)

// Recolectar datos (método post)

$descripcionMedica = (isset($\_POST["descripcionMedica"]) ? $\_POST["descripcionMedica"] : "");

$fechaVencimientoMedica = (isset($\_POST["fechaVencimientoMedica"]) ? $\_POST["fechaVencimientoMedica"] : "");

$cantidadCajas = (isset($\_POST["cantidadCajas"]) ? $\_POST["cantidadCajas"] : "");

$noLoteMedica = (isset($\_POST["noLoteMedica"]) ? $\_POST["noLoteMedica"] : "");

$valorUnitMedica = (isset($\_POST["valorUnitMedica"]) ? $\_POST["valorUnitMedica"] : "");

$fechaFabricacionMedica = (isset($\_POST["fechaFabricacionMedica"]) ? $\_POST["fechaFabricacionMedica"] : "");

$nombreMedica = (isset($\_POST["nombreMedica"]) ? $\_POST["nombreMedica"] : "");

$cantidadUnidades = (isset($\_POST["cantidadUnidades"]) ? $\_POST["cantidadUnidades"] : "");

$idCATEGORIA = (isset($\_POST["SUBCATEGORIA\_CATEGORIA\_idCATEGORIA"]) ? $\_POST["SUBCATEGORIA\_CATEGORIA\_idCATEGORIA"] : "");

$idSUBCATEGORIA = (isset($\_POST["SUBCATEGORIA\_idSUBCATEGORIA"]) ? $\_POST["SUBCATEGORIA\_idSUBCATEGORIA"] : "");

$idRol = (isset($\_POST["Persona\_ROL\_idRol"]) ? $\_POST["Persona\_ROL\_idRol"] : "");

$idPersona = (isset($\_POST["Persona\_idPersona"]) ? $\_POST["Persona\_idPersona"] : "");

$sentencia = $conexion->prepare("INSERT INTO medicamento (idMEDICAMENTO, descripcionMedica, fechaVencimientoMedica, cantidadCajas, noLoteMedica, valorUnitMedica, fechaFabricacionMedica, nombreMedica, cantidadUnidades, SUBCATEGORIA\_CATEGORIA\_idCATEGORIA, SUBCATEGORIA\_idSUBCATEGORIA, Persona\_idPersona, Persona\_ROL\_idRol)

VALUES (null, :descripcionMedica, :fechaVencimientoMedica, :cantidadCajas, :noLoteMedica, :valorUnitMedica, :fechaFabricacionMedica, :nombreMedica, :cantidadUnidades, :idCATEGORIA, :idSUBCATEGORIA, :idPersona, :idRol)");

// Asignar valores que tienen un solo :variable

$sentencia->bindParam(":descripcionMedica", $descripcionMedica);

$sentencia->bindParam(":fechaVencimientoMedica", $fechaVencimientoMedica);

$sentencia->bindParam(":cantidadCajas", $cantidadCajas);

$sentencia->bindParam(":noLoteMedica", $noLoteMedica);

$sentencia->bindParam(":valorUnitMedica", $valorUnitMedica);

$sentencia->bindParam(":fechaFabricacionMedica", $fechaFabricacionMedica);

$sentencia->bindParam(":nombreMedica", $nombreMedica);

$sentencia->bindParam(":cantidadUnidades", $cantidadUnidades);

$sentencia->bindParam(":idCATEGORIA", $idCATEGORIA);

$sentencia->bindParam(":idSUBCATEGORIA", $idSUBCATEGORIA);

$sentencia->bindParam(":idRol", $idRol);

$sentencia->bindParam(":idPersona", $idPersona);

$sentencia->execute();

header("location:index.php");

}

$sentenciaRoles = $conexion->prepare("SELECT idRol, nombreRol FROM rol");

$sentenciaRoles->execute();

$roles = $sentenciaRoles->fetchAll(PDO::FETCH\_ASSOC);

$sentenciaPersona = $conexion->prepare("SELECT idPersona, nombreP FROM persona");

$sentenciaPersona->execute();

$Persona = $sentenciaPersona->fetchAll(PDO::FETCH\_ASSOC);

$sentenciaCat = $conexion->prepare("SELECT idCATEGORIA, nombreCat FROM categoria");

$sentenciaCat->execute();

$Categorias = $sentenciaCat->fetchAll(PDO::FETCH\_ASSOC);

$sentenciaSubcat = $conexion->prepare("SELECT idSUBCATEGORIA, nombreSubcat FROM subcategoria ");

$sentenciaSubcat->execute();

$Subcategorias = $sentenciaSubcat->fetchAll(PDO::FETCH\_ASSOC);

// // Obtener roles desde la base de datos

// $sentenciaRoles = $conexion->prepare("SELECT idRol, nombreRol FROM rol");

// $sentenciaRoles->execute();

// $roles = $sentenciaRoles->fetchAll(PDO::FETCH\_ASSOC);

?>

<?php

include("../../database.php");

if ($\_POST && isset($\_POST["idCategoria"])) {

$idCategoria = filter\_var($\_POST["idCategoria"], FILTER\_VALIDATE\_INT);

if ($idCategoria !== false) {

$sentenciaSubcat = $conexion->prepare("SELECT idSUBCATEGORIA, nombreSubcat FROM subcategoria WHERE CATEGORIA\_idCATEGORIA = :idCategoria");

$sentenciaSubcat->bindParam(":idCategoria", $idCategoria, PDO::PARAM\_INT);

$sentenciaSubcat->execute();

$subcategorias = $sentenciaSubcat->fetchAll(PDO::FETCH\_ASSOC);

if (!empty($subcategorias)) {

foreach ($subcategorias as $subcategoria) {

echo '<option value="' . $subcategoria['idSUBCATEGORIA'] . '">' . $subcategoria['nombreSubcat'] . '</option>';

}

} else {

echo '<option value="">No hay subcategorías disponibles</option>';

}

} else {

echo '<option value="">Error: Categoría no válida</option>';

}

}

?>

(Editar):  
<?php

include("../../database.php");

if (isset($\_GET['txtID'])) {

$txtID = (isset($\_GET['txtID'])) ? $\_GET['txtID'] : "";

$sentencia = $conexion->prepare("SELECT \* FROM medicamento WHERE idMEDICAMENTO = :idMEDICAMENTO");

$sentencia->bindParam(":idMEDICAMENTO", $txtID);

$sentencia->execute();

$registro = $sentencia->fetch(PDO::FETCH\_LAZY);

if ($registro) {

$descripcionMedica = $registro["descripcionMedica"];

$fechaVencimientoMedica = $registro["fechaVencimientoMedica"];

$cantidadCajas = $registro["cantidadCajas"];

$noLoteMedica = $registro["noLoteMedica"];

$valorUnitMedica = $registro["valorUnitMedica"];

$fechaFabricacionMedica = $registro["fechaFabricacionMedica"];

$nombreMedica = $registro["nombreMedica"];

$cantidadUnidades = $registro["cantidadUnidades"];

$idCATEGORIA = $registro["idCATEGORIA"];

$idSUBCATEGORIA = $registro["idSUBCATEGORIA"];

}

}

if ($\_POST) {

$descripcionMedica = (isset($\_POST["descripcionMedica"]) ? $\_POST["descripcionMedica"] : "");

$fechaVencimientoMedica = (isset($\_POST["fechaVencimientoMedica"]) ? $\_POST["fechaVencimientoMedica"] : "");

$cantidadCajas = (isset($\_POST["cantidadCajas"]) ? $\_POST["cantidadCajas"] : "");

$noLoteMedica = (isset($\_POST["noLoteMedica"]) ? $\_POST["noLoteMedica"] : "");

$valorUnitMedica = (isset($\_POST["valorUnitMedica"]) ? $\_POST["valorUnitMedica"] : "");

$fechaFabricacionMedica = (isset($\_POST["fechaFabricacionMedica"]) ? $\_POST["fechaFabricacionMedica"] : "");

$nombreMedica = (isset($\_POST["nombreMedica"]) ? $\_POST["nombreMedica"] : "");

$cantidadUnidades = (isset($\_POST["cantidadUnidades"]) ? $\_POST["cantidadUnidades"] : "");

$idCATEGORIA = (isset($\_POST["SUBCATEGORIA\_CATEGORIA\_idCATEGORIA"]) ? $\_POST["SUBCATEGORIA\_CATEGORIA\_idCATEGORIA"] : "");

$idSUBCATEGORIA = (isset($\_POST["SUBCATEGORIA\_idSUBCATEGORIA"]) ? $\_POST["SUBCATEGORIA\_idSUBCATEGORIA"] : "");

$idRol = (isset($\_POST["Persona\_ROL\_idRol"]) ? $\_POST["Persona\_ROL\_idRol"] : "");

$idPersona = (isset($\_POST["Persona\_idPersona"]) ? $\_POST["Persona\_idPersona"] : "");

$sentencia = $conexion->prepare("UPDATE medicamento SET descripcionMedica = :descripcionMedica, fechaVencimientoMedica = :fechaVencimientoMedica, cantidadCajas = :cantidadCajas, noLoteMedica = :noLoteMedica, valorUnitMedica = :valorUnitMedica, fechaFabricacionMedica = :fechaFabricacionMedica, nombreMedica = :nombreMedica, cantidadUnidades = :cantidadUnidades, SUBCATEGORIA\_CATEGORIA\_idCATEGORIA = :idCATEGORIA, SUBCATEGORIA\_idSUBCATEGORIA = :idSUBCATEGORIA WHERE idMEDICAMENTO = :idMEDICAMENTO");

$sentencia->bindParam(":descripcionMedica", $descripcionMedica);

$sentencia->bindParam(":fechaVencimientoMedica", $fechaVencimientoMedica);

$sentencia->bindParam(":cantidadCajas", $cantidadCajas);

$sentencia->bindParam(":noLoteMedica", $noLoteMedica);

$sentencia->bindParam(":valorUnitMedica", $valorUnitMedica);

$sentencia->bindParam(":fechaFabricacionMedica", $fechaFabricacionMedica);

$sentencia->bindParam(":nombreMedica", $nombreMedica);

$sentencia->bindParam(":cantidadUnidades", $cantidadUnidades);

$sentencia->bindParam(":idCATEGORIA", $idCATEGORIA);

$sentencia->bindParam(":idSUBCATEGORIA", $idSUBCATEGORIA);

$sentencia->bindParam(":idMEDICAMENTO", $txtID);

$sentencia->execute();

header("Location:index.php");

}

$sentenciaCat = $conexion->prepare("SELECT idCATEGORIA, nombreCat FROM categoria");

$sentenciaCat->execute();

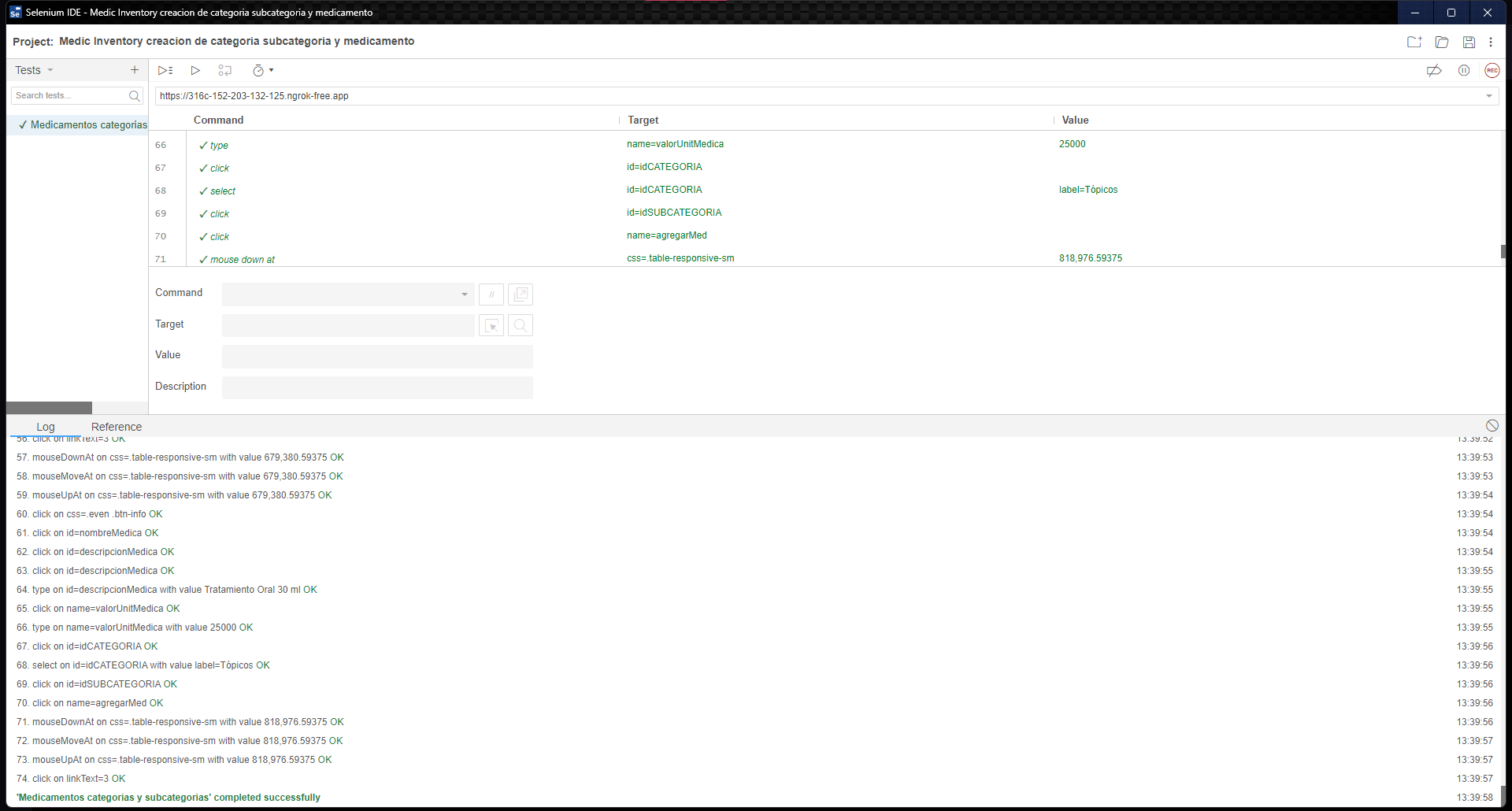
$Categorias = $sentenciaCat->fetchAll(PDO::FETCH\_ASSOC);

$sentenciaSubcat = $conexion->prepare("SELECT idSUBCATEGORIA, nombreSubcat FROM subcategoria");

$sentenciaSubcat->execute();

$Subcategorias = $sentenciaSubcat->fetchAll(PDO::FETCH\_ASSOC);

?>



Prueba 3: Eliminación de creada Categoría, eliminación de la creada subcategoría, eliminación del medicamento creado.  
Categoria (Eliminar):  
<?php

include("../../database.php");

if (isset($\_GET['txtID'])) {

$txtID = (isset($\_GET['txtID'])) ? $\_GET['txtID'] : "";

$sentencia = $conexion->prepare("DELETE FROM categoria WHERE idCATEGORIA=:idCATEGORIA");

$sentencia->bindParam(":idCATEGORIA", $txtID);

$sentencia->execute();

$mensaje = "Registro eliminado";

header("Location:index.php?mensaje=" . $mensaje);

}

Subcategoría (Eliminar):  
<?php

include("../../database.php");

if (isset($\_GET['txtID'])) {

$txtID = (isset($\_GET['txtID'])) ? $\_GET['txtID'] : "";

$sentencia = $conexion->prepare("DELETE FROM subcategoria WHERE idSUBCATEGORIA=:idSUBCATEGORIA");

$sentencia->bindParam(":idSUBCATEGORIA", $txtID);

$sentencia->execute();

header("location:index.php");

}

Medicamento (Eliminar):  
<?php

include('../../database.php');

if (isset($\_GET['txtID'])) {

$txtID = (isset($\_GET['txtID'])) ? $\_GET['txtID'] : "";

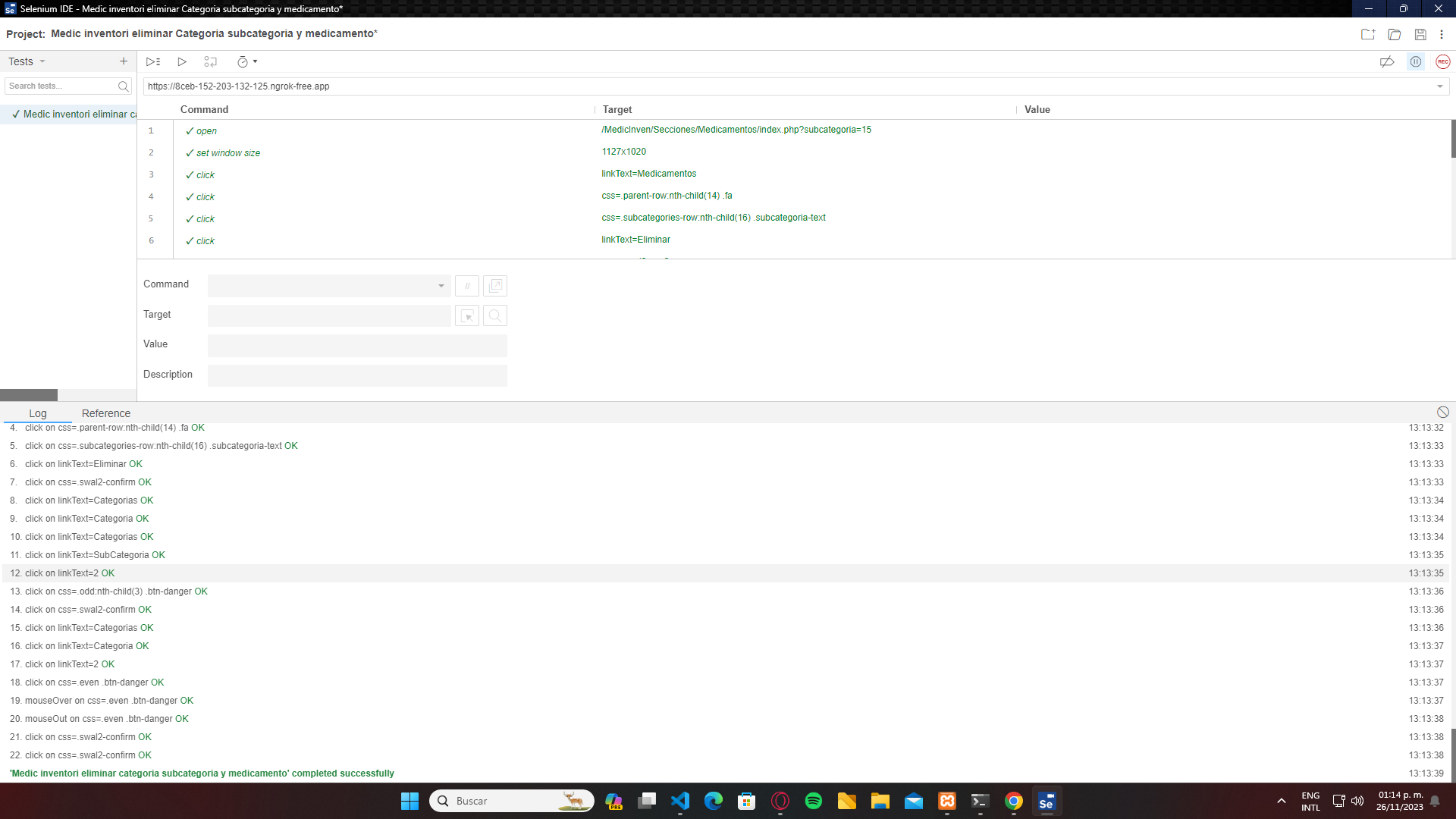
$sentencia = $conexion->prepare("DELETE FROM medicamento WHERE idMEDICAMENTO =:idMEDICAMENTO");

$sentencia->bindParam(":idMEDICAMENTO", $txtID);

$sentencia->execute();

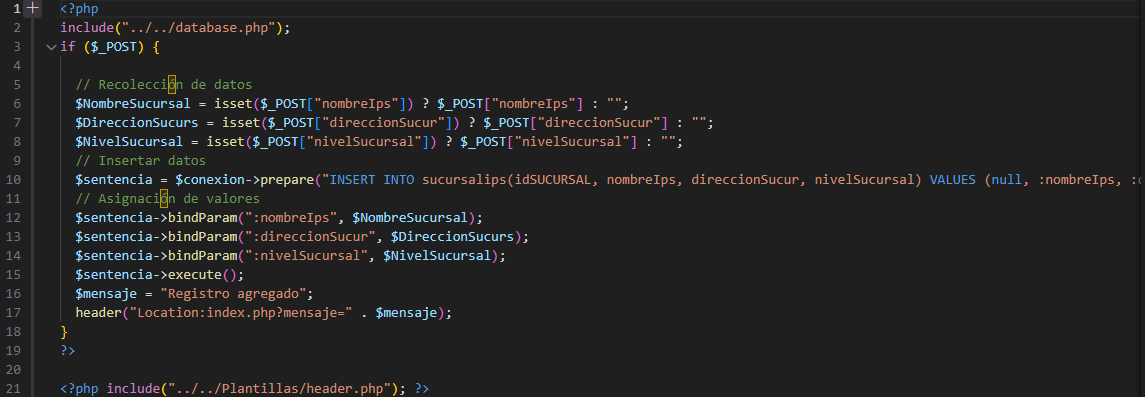
header("Location:index.php");

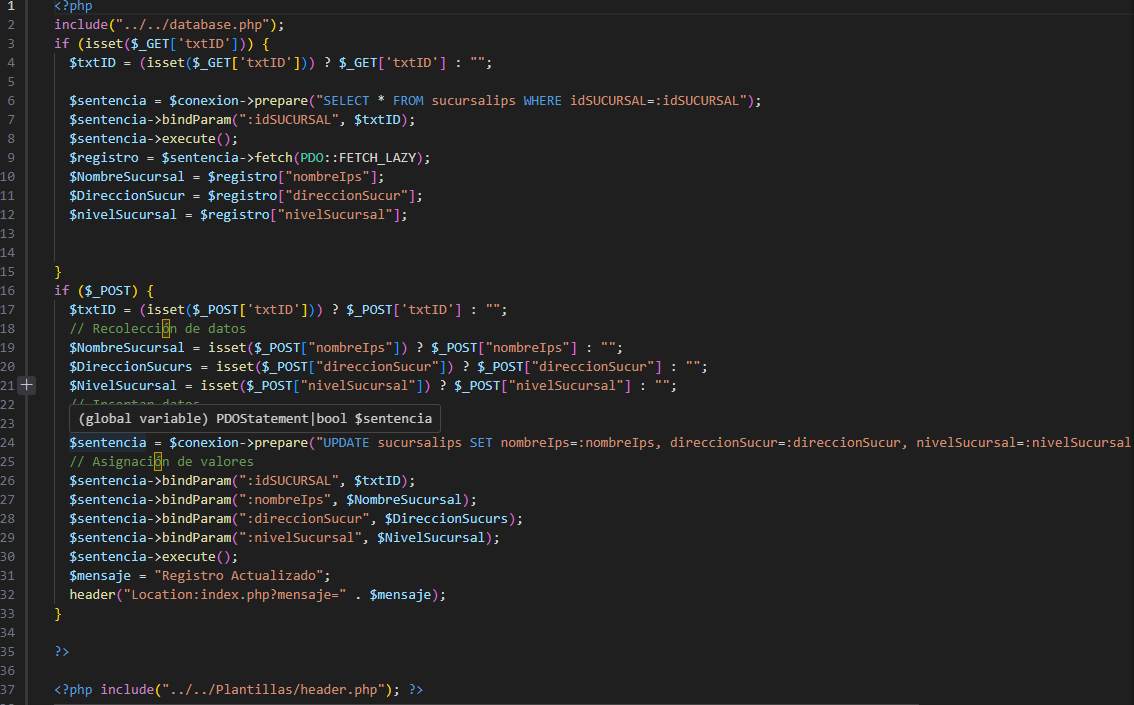
}



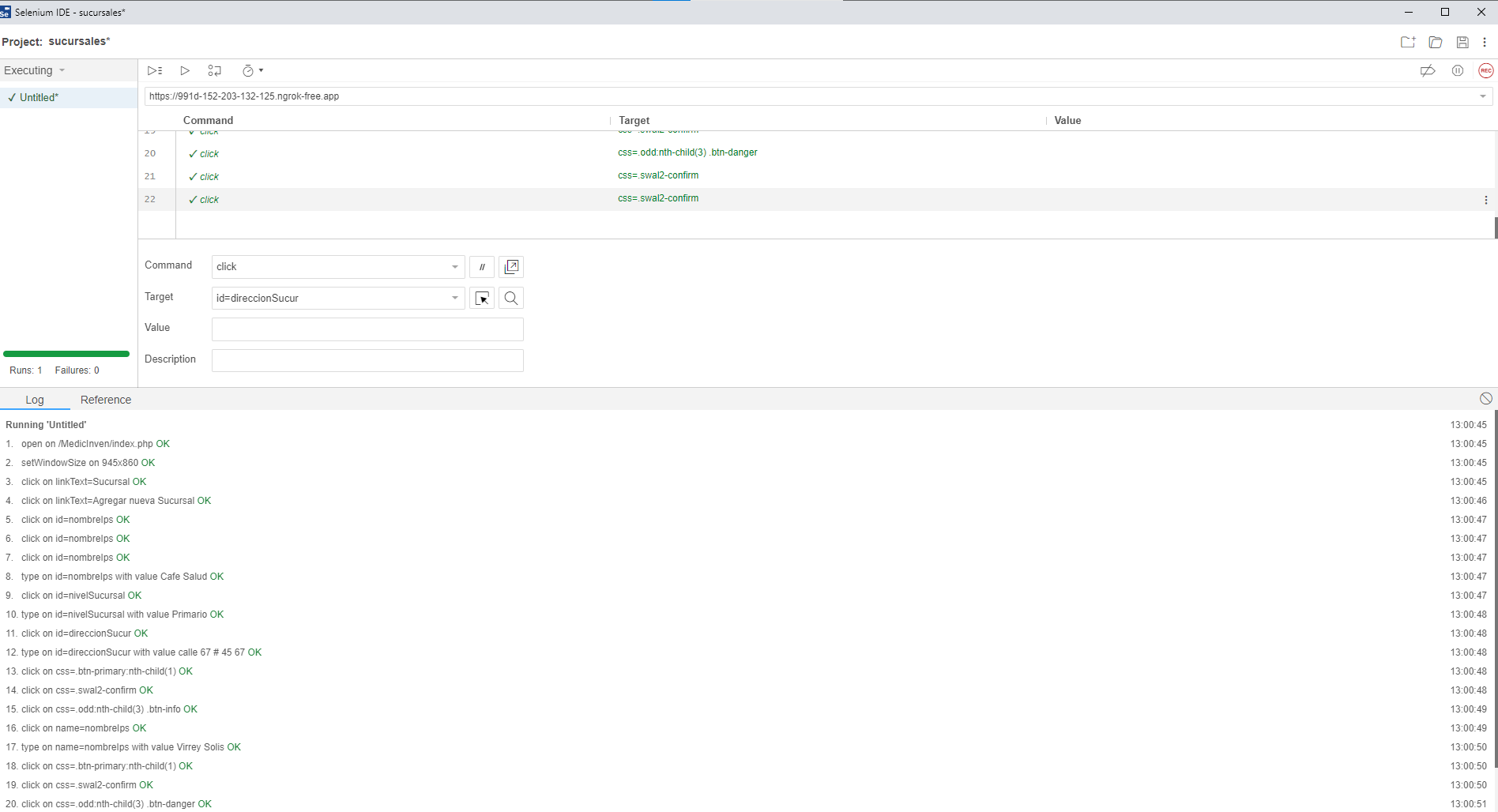
Prueba 4: Creación de Sucursales, editar Sucursales y eliminación de sucursales se evidencian cambios en la base de datos satisfactorios .

Código evaluado:



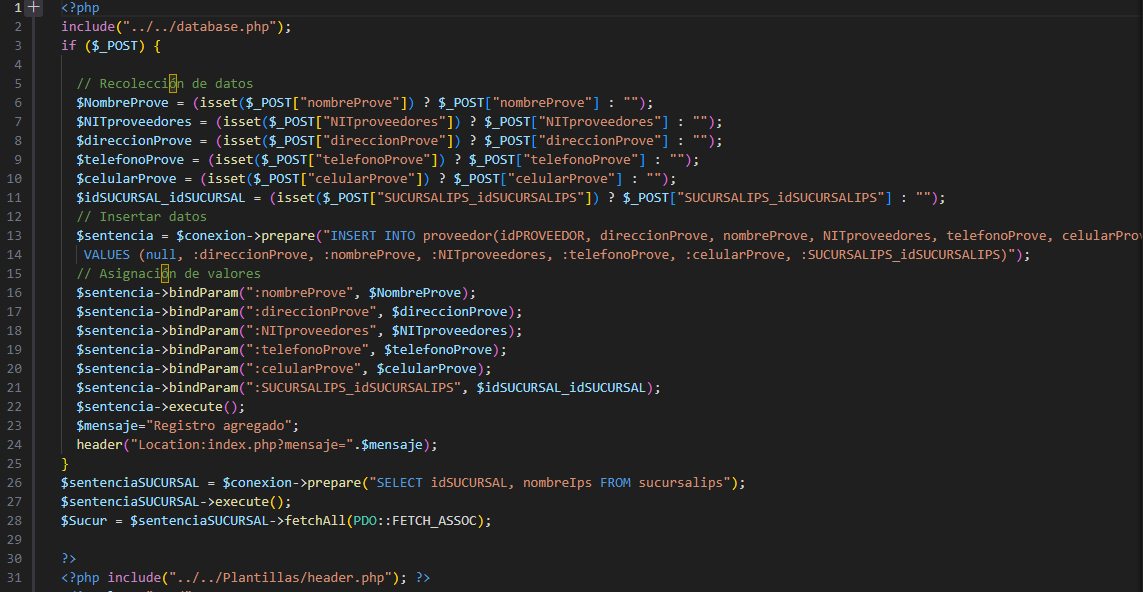


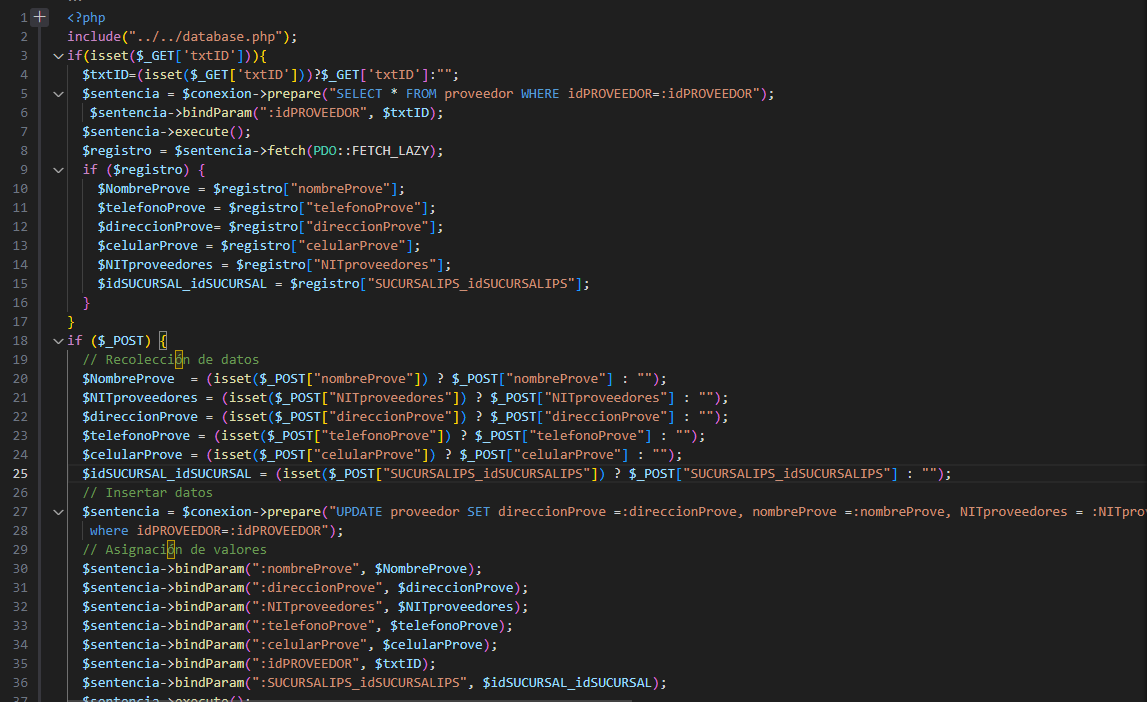


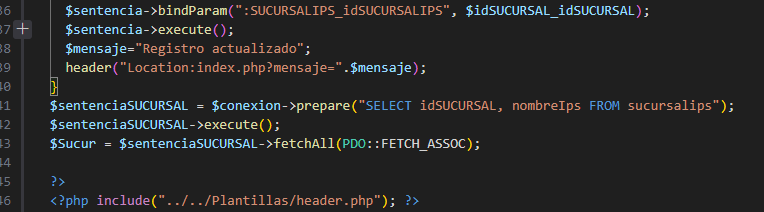


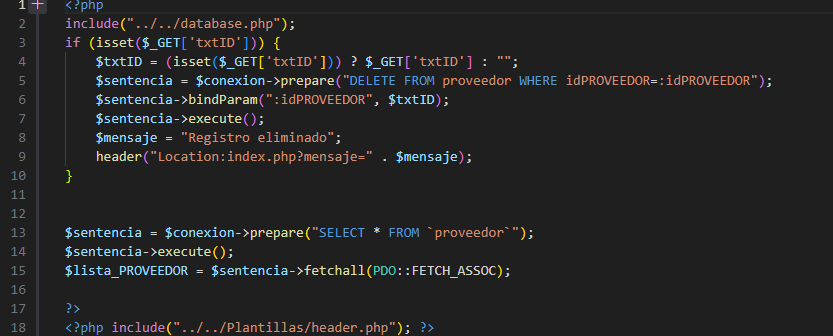
Prueba 5: Creación de proveedores, editar proveedores y eliminar proveedores se evidencian cambios satisfactorios en la base de datos.

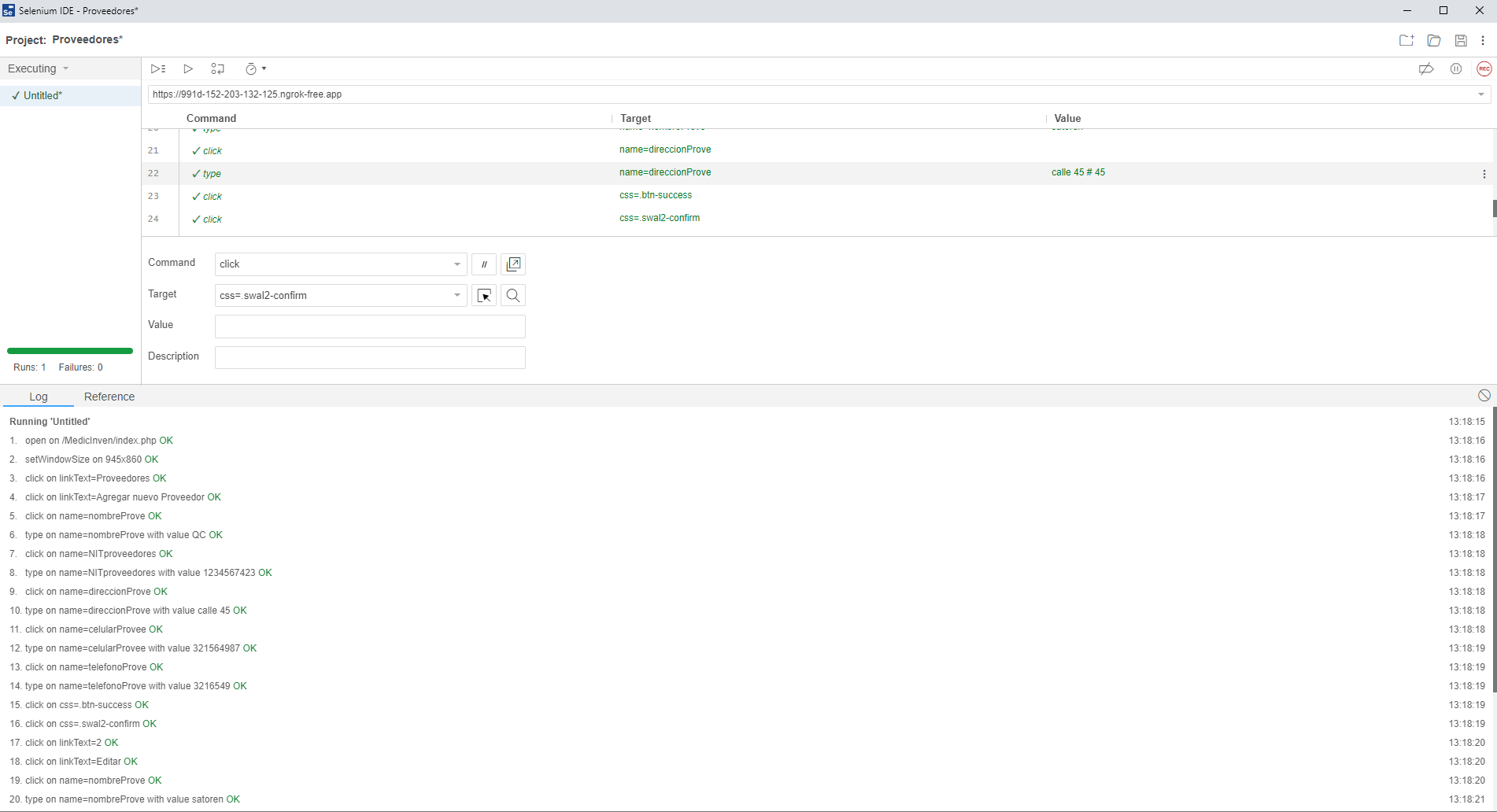
Código evaluado:



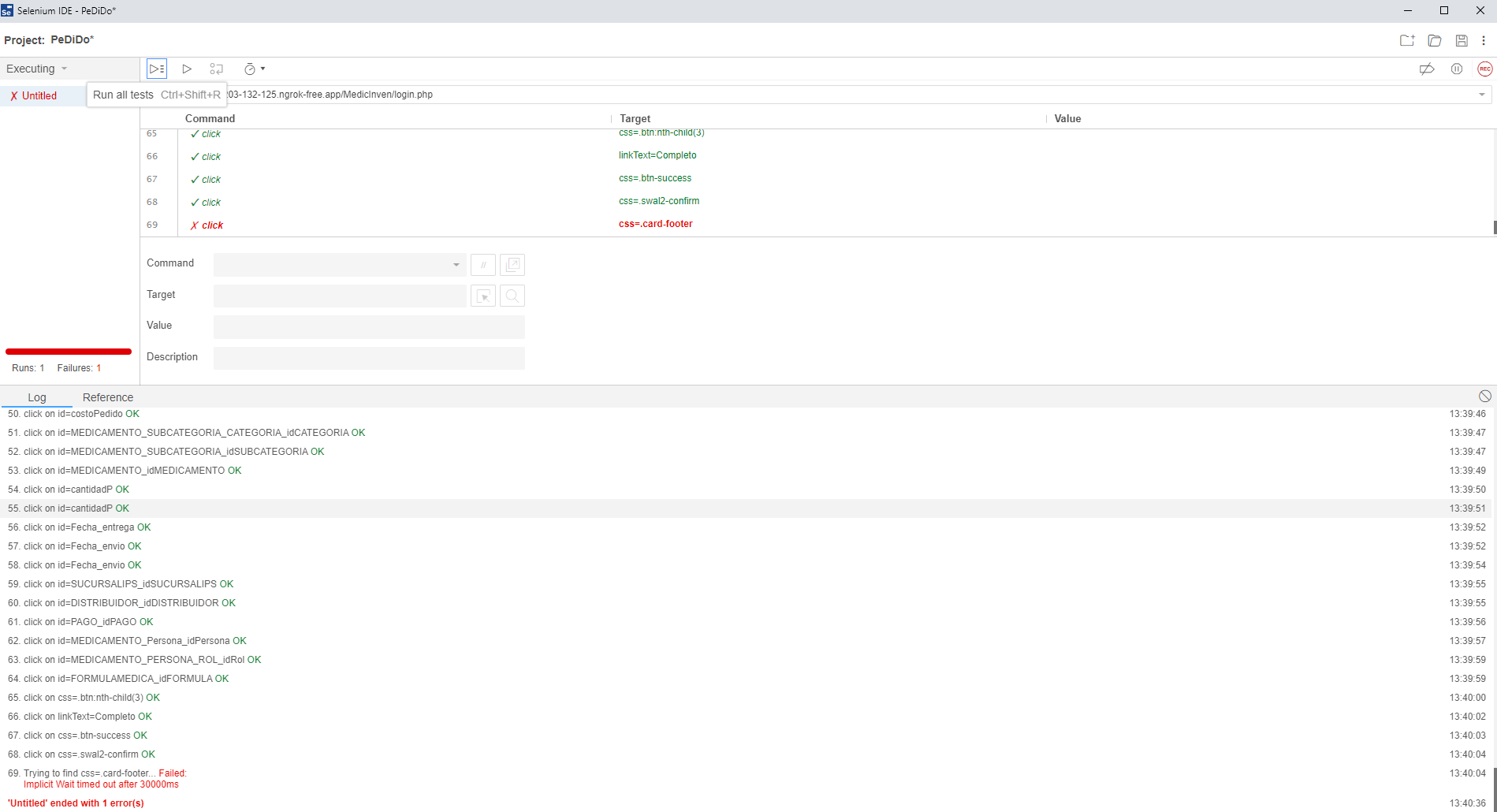








prueba 6: Creación de pedido y editar pedido se evidencia un error.

  
  
Prueba 7: creación de pedido y editar pedido se evidencian cambios exitosos en la base de datos  
Código evaluado: (Crear)  
  
<?php

include("../../database.php");

$sentenciaNombresMedicamentos = $conexion->prepare("SELECT idMEDICAMENTO, nombreMedica FROM medicamento");

$sentenciaNombresMedicamentos->execute();

$nombresMedicamentos = $sentenciaNombresMedicamentos->fetchAll(PDO::FETCH\_ASSOC);

$sentenciaidsucursal = $conexion->prepare("SELECT idSUCURSAL, nombreIps FROM sucursalips");

$sentenciaidsucursal->execute();

$idsucursal = $sentenciaidsucursal->fetchAll(PDO::FETCH\_ASSOC);

$sentenciaDistribuidores = $conexion->prepare("SELECT idDISTRIBUIDOR, nombreDistri FROM distribuidor");

$sentenciaDistribuidores->execute();

$distribuidores = $sentenciaDistribuidores->fetchAll(PDO::FETCH\_ASSOC);

$sentenciaPagos = $conexion->prepare("SELECT idPAGO, ReferenciaPago FROM pago");

$sentenciaPagos->execute();

$pagos = $sentenciaPagos->fetchAll(PDO::FETCH\_ASSOC);

$sentenciaPersonas = $conexion->prepare("SELECT idPersona, nombreP FROM persona");

$sentenciaPersonas->execute();

$personas = $sentenciaPersonas->fetchAll(PDO::FETCH\_ASSOC);

$sentenciaRoles = $conexion->prepare("SELECT idRol, nombreRol FROM rol");

$sentenciaRoles->execute();

$roles = $sentenciaRoles->fetchAll(PDO::FETCH\_ASSOC);

$sentenciaCategorias = $conexion->prepare("SELECT idCATEGORIA, nombreCat FROM categoria");

$sentenciaCategorias->execute();

$categorias = $sentenciaCategorias->fetchAll(PDO::FETCH\_ASSOC);

$sentenciaSubcategorias = $conexion->prepare("SELECT idSUBCATEGORIA, nombreSubcat FROM subcategoria");

$sentenciaSubcategorias->execute();

$subcategorias = $sentenciaSubcategorias->fetchAll(PDO::FETCH\_ASSOC);

$sentenciaformula = $conexion->prepare("SELECT idFORMULA, Referenciaformula FROM formulamedica");

$sentenciaformula->execute();

$formulaM = $sentenciaformula->fetchAll(PDO::FETCH\_ASSOC);

if ($\_POST) {

$idPEDIDO = (isset($\_POST["idPEDIDO"]) ? $\_POST["idPEDIDO"] : "");

$Tipo\_pedido = (isset($\_POST["Tipo\_pedido"]) ? $\_POST["Tipo\_pedido"] : "");

$fechaPedido = (isset($\_POST["fechaPedido"]) ? $\_POST["fechaPedido"] : "");

$costoPedido = (isset($\_POST["costoPedido"]) ? $\_POST["costoPedido"] : "");

$Nombre\_Producto = (isset($\_POST["Nombre\_Producto"]) ? $\_POST["Nombre\_Producto"] : "");

$cantidadP = (isset($\_POST["cantidadP"]) ? $\_POST["cantidadP"] : "");

$Fecha\_entrega = (isset($\_POST["Fecha\_entrega"]) ? $\_POST["Fecha\_entrega"] : "");

$Fecha\_envio = (isset($\_POST["Fecha\_envio"]) ? $\_POST["Fecha\_envio"] : "");

$EstadoP = (isset($\_POST["EstadoP"]) ? $\_POST["EstadoP"] : "");

$SUCURSALIPS\_idSUCURSALIPS = (isset($\_POST["SUCURSALIPS\_idSUCURSALIPS"]) ? $\_POST["SUCURSALIPS\_idSUCURSALIPS"] : "");

$DISTRIBUIDOR\_idDISTRIBUIDOR = (isset($\_POST["DISTRIBUIDOR\_idDISTRIBUIDOR"]) ? $\_POST["DISTRIBUIDOR\_idDISTRIBUIDOR"] : "");

$PAGO\_idPAGO = (isset($\_POST["PAGO\_idPAGO"]) ? $\_POST["PAGO\_idPAGO"] : "");

$MEDICAMENTO\_idMEDICAMENTO = (isset($\_POST["MEDICAMENTO\_idMEDICAMENTO"]) ? $\_POST["MEDICAMENTO\_idMEDICAMENTO"] : "");

$MEDICAMENTO\_Persona\_idPersona = (isset($\_POST["MEDICAMENTO\_Persona\_idPersona"]) ? $\_POST["MEDICAMENTO\_Persona\_idPersona"] : "");

$MEDICAMENTO\_PERSONA\_ROL\_idRol = (isset($\_POST["MEDICAMENTO\_PERSONA\_ROL\_idRol"]) ? $\_POST["MEDICAMENTO\_PERSONA\_ROL\_idRol"] : "");

$MEDICAMENTO\_SUBCATEGORIA\_idSUBCATEGORIA = (isset($\_POST["MEDICAMENTO\_SUBCATEGORIA\_idSUBCATEGORIA"]) ? $\_POST["MEDICAMENTO\_SUBCATEGORIA\_idSUBCATEGORIA"] : "");

$MEDICAMENTO\_SUBCATEGORIA\_CATEGORIA\_idCATEGORIA = (isset($\_POST["MEDICAMENTO\_SUBCATEGORIA\_CATEGORIA\_idCATEGORIA"]) ? $\_POST["MEDICAMENTO\_SUBCATEGORIA\_CATEGORIA\_idCATEGORIA"] : "");

$FORMULAMEDICA\_idFORMULA = (isset($\_POST["FORMULAMEDICA\_idFORMULA"]) ? $\_POST["FORMULAMEDICA\_idFORMULA"] : "");

$nombreProductoSeleccionado = (isset($\_POST["MEDICAMENTO\_idMEDICAMENTO"])) ? $\_POST["MEDICAMENTO\_idMEDICAMENTO"] : "";

$estadoPedidoSeleccionado = (isset($\_POST["EstadoP"])) ? $\_POST["EstadoP"] : "";

$sentencia = $conexion->prepare("INSERT INTO pedido (idPEDIDO, Tipo\_pedido, fechaPedido, costoPedido, Nombre\_Producto, cantidadP, Fecha\_entrega, Fecha\_envio, EstadoP, SUCURSALIPS\_idSUCURSALIPS, DISTRIBUIDOR\_idDISTRIBUIDOR, PAGO\_idPAGO, MEDICAMENTO\_idMEDICAMENTO, MEDICAMENTO\_Persona\_idPersona, MEDICAMENTO\_PERSONA\_ROL\_idRol, MEDICAMENTO\_SUBCATEGORIA\_idSUBCATEGORIA, MEDICAMENTO\_SUBCATEGORIA\_CATEGORIA\_idCATEGORIA, FORMULAMEDICA\_idFORMULA)

VALUES (null, :Tipo\_pedido,:fechaPedido, :costoPedido, :Nombre\_Producto, :cantidadP, :Fecha\_entrega, :Fecha\_envio, :EstadoP, :SUCURSALIPS\_idSUCURSALIPS, :DISTRIBUIDOR\_idDISTRIBUIDOR, :PAGO\_idPAGO, :MEDICAMENTO\_idMEDICAMENTO, :MEDICAMENTO\_Persona\_idPersona, :MEDICAMENTO\_PERSONA\_ROL\_idRol, :MEDICAMENTO\_SUBCATEGORIA\_idSUBCATEGORIA, :MEDICAMENTO\_SUBCATEGORIA\_CATEGORIA\_idCATEGORIA, :FORMULAMEDICA\_idFORMULA)");

$sentencia->bindParam(":Tipo\_pedido", $Tipo\_pedido);

$sentencia->bindParam(":fechaPedido", $fechaPedido);

$sentencia->bindParam(":costoPedido", $costoPedido);

$sentencia->bindParam(":cantidadP", $cantidadP);

$sentencia->bindParam(":Fecha\_entrega", $Fecha\_entrega);

$sentencia->bindParam(":Fecha\_envio", $Fecha\_envio);

$sentencia->bindParam(":SUCURSALIPS\_idSUCURSALIPS", $SUCURSALIPS\_idSUCURSALIPS);

$sentencia->bindParam(":DISTRIBUIDOR\_idDISTRIBUIDOR", $DISTRIBUIDOR\_idDISTRIBUIDOR);

$sentencia->bindParam(":PAGO\_idPAGO", $PAGO\_idPAGO);

$sentencia->bindParam(":MEDICAMENTO\_idMEDICAMENTO", $MEDICAMENTO\_idMEDICAMENTO);

$sentencia->bindParam(":MEDICAMENTO\_Persona\_idPersona", $MEDICAMENTO\_Persona\_idPersona);

$sentencia->bindParam(":MEDICAMENTO\_PERSONA\_ROL\_idRol", $MEDICAMENTO\_PERSONA\_ROL\_idRol);

$sentencia->bindParam(":MEDICAMENTO\_SUBCATEGORIA\_idSUBCATEGORIA", $MEDICAMENTO\_SUBCATEGORIA\_idSUBCATEGORIA);

$sentencia->bindParam(":MEDICAMENTO\_SUBCATEGORIA\_CATEGORIA\_idCATEGORIA", $MEDICAMENTO\_SUBCATEGORIA\_CATEGORIA\_idCATEGORIA);

$sentencia->bindParam(":FORMULAMEDICA\_idFORMULA", $FORMULAMEDICA\_idFORMULA);

$sentencia->bindParam(":Nombre\_Producto", $nombreProductoSeleccionado);

$sentencia->bindParam(":EstadoP", $estadoPedidoSeleccionado);

if ($Tipo\_pedido === "Entrada") {

// Lógica para una entrada de medicamento

$sentenciaUpdateMedicamento = $conexion->prepare("UPDATE medicamento SET cantidadUnidades = cantidadUnidades + :cantidadPedido WHERE idMEDICAMENTO = :medicamentoID");

$sentenciaUpdateMedicamento->bindParam(":cantidadPedido", $cantidadP);

$sentenciaUpdateMedicamento->bindParam(":medicamentoID", $MEDICAMENTO\_idMEDICAMENTO);

} elseif ($Tipo\_pedido === "Salida") {

// Lógica para una salida de medicamento

$sentenciaUpdateMedicamento = $conexion->prepare("UPDATE medicamento SET cantidadUnidades = cantidadUnidades - :cantidadPedido WHERE idMEDICAMENTO = :medicamentoID");

$sentenciaUpdateMedicamento->bindParam(":cantidadPedido", $cantidadP);

$sentenciaUpdateMedicamento->bindParam(":medicamentoID", $MEDICAMENTO\_idMEDICAMENTO);

}

if ($sentencia->execute()) {

if ($sentenciaUpdateMedicamento->execute()) {

// Cantidad de unidades actualizada correctamente

// Redireccionar o mostrar mensaje de éxito

header("Location: index.php");

exit();

} else {

// Mensaje de error al actualizar la cantidad de unidades

echo "Error al actualizar la cantidad de unidades del medicamento.";

}

} else {

// Mensaje de error al insertar el pedido

echo "Error al insertar el pedido.";

}

header("Location:index.php?mensaje=" . $mensaje);

}

?>

(Editar):  
  
<?php

include("../../database.php");

if (isset($\_GET['txtID'])) {

$txtID = (isset($\_GET['txtID'])) ? $\_GET['txtID'] : '';

$sentencia = $conexion->prepare("SELECT \* from pedido where idPEDIDO = :idPEDIDO");

$sentencia->bindParam(":idPEDIDO", $txtID);

$sentencia->execute();

$registro = $sentencia->fetch(PDO::FETCH\_LAZY);

if ($registro) {

$cantidadPedidoOriginal = $registro["cantidadP"];

$idPEDIDO = $registro["idPEDIDO"];

$Tipo\_pedido = $registro["Tipo\_pedido"];

$fechaPedido = $registro["fechaPedido"];

$costoPedido = $registro["costoPedido"];

$Nombre\_Producto = $registro["Nombre\_Producto"];

$cantidadP = $registro["cantidadP"];

$Fecha\_entrega = $registro["Fecha\_entrega"];

$Fecha\_envio = $registro["Fecha\_envio"];

$EstadoP = $registro["EstadoP"];

$SUCURSALIPS\_idSUCURSALIPS = $registro["SUCURSALIPS\_idSUCURSALIPS"];

$DISTRIBUIDOR\_idDISTRIBUIDOR = $registro["DISTRIBUIDOR\_idDISTRIBUIDOR"];

$PAGO\_idPAGO = $registro["PAGO\_idPAGO"];

$MEDICAMENTO\_idMEDICAMENTO = $registro["MEDICAMENTO\_idMEDICAMENTO"];

$MEDICAMENTO\_Persona\_idPersona = $registro["MEDICAMENTO\_Persona\_idPersona"];

$MEDICAMENTO\_PERSONA\_ROL\_idRol = $registro["MEDICAMENTO\_PERSONA\_ROL\_idRol"];

$MEDICAMENTO\_SUBCATEGORIA\_idSUBCATEGORIA = $registro["MEDICAMENTO\_SUBCATEGORIA\_idSUBCATEGORIA"];

$MEDICAMENTO\_SUBCATEGORIA\_CATEGORIA\_idCATEGORIA = $registro["MEDICAMENTO\_SUBCATEGORIA\_CATEGORIA\_idCATEGORIA"];

$FORMULAMEDICA\_idFORMULA = $registro["FORMULAMEDICA\_idFORMULA"];

$nombreProductoSeleccionado = $registro["MEDICAMENTO\_idMEDICAMENTO"];

$estadoPedidoSeleccionado = $registro["EstadoP"];

}

}

$sentenciaNombresMedicamentos = $conexion->prepare("SELECT \* FROM medicamento");

$sentenciaNombresMedicamentos->execute();

$nombresMedicamentos = $sentenciaNombresMedicamentos->fetchAll(PDO::FETCH\_ASSOC);

$sentenciaidsucursal = $conexion->prepare("SELECT \* FROM sucursalips");

$sentenciaidsucursal->execute();

$idsucursal = $sentenciaidsucursal->fetchAll(PDO::FETCH\_ASSOC);

$sentenciaDistribuidores = $conexion->prepare("SELECT \* FROM distribuidor");

$sentenciaDistribuidores->execute();

$distribuidores = $sentenciaDistribuidores->fetchAll(PDO::FETCH\_ASSOC);

$sentenciaPagos = $conexion->prepare("SELECT \* FROM pago");

$sentenciaPagos->execute();

$pagos = $sentenciaPagos->fetchAll(PDO::FETCH\_ASSOC);

$sentenciaPersonas = $conexion->prepare("SELECT \* FROM persona");

$sentenciaPersonas->execute();

$personas = $sentenciaPersonas->fetchAll(PDO::FETCH\_ASSOC);

$sentenciaRoles = $conexion->prepare("SELECT \* FROM rol");

$sentenciaRoles->execute();

$roles = $sentenciaRoles->fetchAll(PDO::FETCH\_ASSOC);

$sentenciaCategorias = $conexion->prepare("SELECT \* FROM categoria");

$sentenciaCategorias->execute();

$categorias = $sentenciaCategorias->fetchAll(PDO::FETCH\_ASSOC);

$sentenciaSubcategorias = $conexion->prepare("SELECT \* FROM subcategoria");

$sentenciaSubcategorias->execute();

$subcategorias = $sentenciaSubcategorias->fetchAll(PDO::FETCH\_ASSOC);

$sentenciaformula = $conexion->prepare("SELECT \* FROM formulamedica");

$sentenciaformula->execute();

$formulaM = $sentenciaformula->fetchAll(PDO::FETCH\_ASSOC);

if ($\_POST) {

var\_dump($\_POST);

$idPEDIDO = (isset($\_POST["idPEDIDO"]) ? $\_POST["idPEDIDO"] : "");

$Tipo\_pedido = (isset($\_POST["Tipo\_pedido"]) ? $\_POST["Tipo\_pedido"] : "");

$fechaPedido = (isset($\_POST["fechaPedido"]) ? $\_POST["fechaPedido"] : "");

$costoPedido = (isset($\_POST["costoPedido"]) ? $\_POST["costoPedido"] : "");

$Nombre\_Producto = (isset($\_POST["Nombre\_Producto"]) ? $\_POST["Nombre\_Producto"] : "");

$cantidadP = (isset($\_POST["cantidadP"]) ? $\_POST["cantidadP"] : "");

$Fecha\_entrega = (isset($\_POST["Fecha\_entrega"]) ? $\_POST["Fecha\_entrega"] : "");

$Fecha\_envio = (isset($\_POST["Fecha\_envio"]) ? $\_POST["Fecha\_envio"] : "");

$EstadoP = (isset($\_POST["EstadoP"]) ? $\_POST["EstadoP"] : "");

$SUCURSALIPS\_idSUCURSALIPS = (isset($\_POST["SUCURSALIPS\_idSUCURSALIPS"]) ? $\_POST["SUCURSALIPS\_idSUCURSALIPS"] : "");

$DISTRIBUIDOR\_idDISTRIBUIDOR = (isset($\_POST["DISTRIBUIDOR\_idDISTRIBUIDOR"]) ? $\_POST["DISTRIBUIDOR\_idDISTRIBUIDOR"] : "");

$PAGO\_idPAGO = (isset($\_POST["PAGO\_idPAGO"]) ? $\_POST["PAGO\_idPAGO"] : "");

$MEDICAMENTO\_idMEDICAMENTO = (isset($\_POST["MEDICAMENTO\_idMEDICAMENTO"]) ? $\_POST["MEDICAMENTO\_idMEDICAMENTO"] : "");

$MEDICAMENTO\_Persona\_idPersona = (isset($\_POST["MEDICAMENTO\_Persona\_idPersona"]) ? $\_POST["MEDICAMENTO\_Persona\_idPersona"] : "");

$MEDICAMENTO\_PERSONA\_ROL\_idRol = (isset($\_POST["MEDICAMENTO\_PERSONA\_ROL\_idRol"]) ? $\_POST["MEDICAMENTO\_PERSONA\_ROL\_idRol"] : "");

$MEDICAMENTO\_SUBCATEGORIA\_idSUBCATEGORIA = (isset($\_POST["MEDICAMENTO\_SUBCATEGORIA\_idSUBCATEGORIA"]) ? $\_POST["MEDICAMENTO\_SUBCATEGORIA\_idSUBCATEGORIA"] : "");

$MEDICAMENTO\_SUBCATEGORIA\_CATEGORIA\_idCATEGORIA = (isset($\_POST["MEDICAMENTO\_SUBCATEGORIA\_CATEGORIA\_idCATEGORIA"]) ? $\_POST["MEDICAMENTO\_SUBCATEGORIA\_CATEGORIA\_idCATEGORIA"] : "");

$FORMULAMEDICA\_idFORMULA = (isset($\_POST["FORMULAMEDICA\_idFORMULA"]) ? $\_POST["FORMULAMEDICA\_idFORMULA"] : "");

//$nombreProductoSeleccionado = (isset($\_POST["MEDICAMENTO\_idMEDICAMENTO"])) ? $\_POST["MEDICAMENTO\_idMEDICAMENTO"] : "";

//$estadoPedidoSeleccionado = (isset($\_POST["EstadoP"])) ? $\_POST["EstadoP"] : "";

$sentenciaActualizarPedido = $conexion->prepare("UPDATE pedido SET Tipo\_pedido = :Tipo\_pedido, fechaPedido = :fechaPedido,

costoPedido = :costoPedido, Nombre\_Producto = :Nombre\_Producto, cantidadP = :cantidadP, Fecha\_entrega = :Fecha\_entrega,

Fecha\_envio = :Fecha\_envio, EstadoP = :EstadoP, SUCURSALIPS\_idSUCURSALIPS = :SUCURSALIPS\_idSUCURSALIPS, DISTRIBUIDOR\_idDISTRIBUIDOR = :DISTRIBUIDOR\_idDISTRIBUIDOR

, PAGO\_idPAGO = :PAGO\_idPAGO, MEDICAMENTO\_Persona\_idPersona = :MEDICAMENTO\_Persona\_idPersona, MEDICAMENTO\_idMEDICAMENTO = :MEDICAMENTO\_idMEDICAMENTO,

MEDICAMENTO\_PERSONA\_ROL\_idRol = :MEDICAMENTO\_PERSONA\_ROL\_idRol, MEDICAMENTO\_SUBCATEGORIA\_idSUBCATEGORIA = :MEDICAMENTO\_SUBCATEGORIA\_idSUBCATEGORIA

, MEDICAMENTO\_SUBCATEGORIA\_CATEGORIA\_idCATEGORIA = :MEDICAMENTO\_SUBCATEGORIA\_CATEGORIA\_idCATEGORIA, FORMULAMEDICA\_idFORMULA = :FORMULAMEDICA\_idFORMULA WHERE idPEDIDO = :idPEDIDO");

$sentenciaActualizarPedido->bindParam(":Tipo\_pedido", $Tipo\_pedido);

$sentenciaActualizarPedido->bindParam(":fechaPedido", $fechaPedido);

$sentenciaActualizarPedido->bindParam(":costoPedido", $costoPedido);

$sentenciaActualizarPedido->bindParam(":Nombre\_Producto", $Nombre\_Producto);

$sentenciaActualizarPedido->bindParam(":cantidadP", $cantidadP);

$sentenciaActualizarPedido->bindParam(":Fecha\_entrega", $Fecha\_entrega);

$sentenciaActualizarPedido->bindParam(":Fecha\_envio", $Fecha\_envio);

$sentenciaActualizarPedido->bindParam(":EstadoP", $EstadoP);

$sentenciaActualizarPedido->bindParam(":idPEDIDO", $idPEDIDO);

$sentenciaActualizarPedido->bindParam(":SUCURSALIPS\_idSUCURSALIPS", $SUCURSALIPS\_idSUCURSALIPS);

$sentenciaActualizarPedido->bindParam(":DISTRIBUIDOR\_idDISTRIBUIDOR", $DISTRIBUIDOR\_idDISTRIBUIDOR);

$sentenciaActualizarPedido->bindParam(":PAGO\_idPAGO", $PAGO\_idPAGO);

$sentenciaActualizarPedido->bindParam(":MEDICAMENTO\_idMEDICAMENTO", $MEDICAMENTO\_idMEDICAMENTO);

$sentenciaActualizarPedido->bindParam(":MEDICAMENTO\_Persona\_idPersona", $MEDICAMENTO\_Persona\_idPersona);

$sentenciaActualizarPedido->bindParam(":MEDICAMENTO\_PERSONA\_ROL\_idRol", $MEDICAMENTO\_PERSONA\_ROL\_idRol);

$sentenciaActualizarPedido->bindParam(":MEDICAMENTO\_SUBCATEGORIA\_idSUBCATEGORIA", $MEDICAMENTO\_SUBCATEGORIA\_idSUBCATEGORIA);

$sentenciaActualizarPedido->bindParam(":MEDICAMENTO\_SUBCATEGORIA\_CATEGORIA\_idCATEGORIA", $MEDICAMENTO\_SUBCATEGORIA\_CATEGORIA\_idCATEGORIA);

$sentenciaActualizarPedido->bindParam(":FORMULAMEDICA\_idFORMULA", $FORMULAMEDICA\_idFORMULA);

echo "Error al actualizar el pedido.";

if ($sentenciaActualizarPedido->execute()) {

// Lógica para actualizar la cantidad de unidades en la tabla 'medicamento' según el tipo de pedido

if ($Tipo\_pedido === "Entrada") {

$diferenciaCantidad = $cantidadP - $cantidadPedidoOriginal;

// Lógica para una entrada de medicamento

$sentenciaUpdateMedicamento = $conexion->prepare("UPDATE medicamento SET cantidadUnidades = cantidadUnidades + :diferenciaCantidad WHERE idMEDICAMENTO = :medicamentoID");

} elseif ($Tipo\_pedido === "Salida") {

$diferenciaCantidad = $cantidadPedidoOriginal - $cantidadP;

// Lógica para una salida de medicamento

$sentenciaUpdateMedicamento = $conexion->prepare("UPDATE medicamento SET cantidadUnidades = cantidadUnidades - :diferenciaCantidad WHERE idMEDICAMENTO = :medicamentoID");

}

if (isset($sentenciaUpdateMedicamento)) {

// Redireccionar a la página principal después de editar el pedido

$sentenciaUpdateMedicamento->bindParam(":diferenciaCantidad", $diferenciaCantidad);

$sentenciaUpdateMedicamento->bindParam(":medicamentoID", $MEDICAMENTO\_idMEDICAMENTO);

if ($sentenciaUpdateMedicamento->execute()) {

$mensaje = "Registro Actualizado";

header("Location: index.php?mensaje=" . $mensaje);

exit();

} else {

// Mensaje de error al actualizar la cantidad de unidades

echo "Error al actualizar la cantidad de unidades del medicamento.";

print\_r($sentenciaUpdateMedicamento->errorInfo());

}

} else {

echo "Error: La sentencia para actualizar el medicamento no está definida.";

}

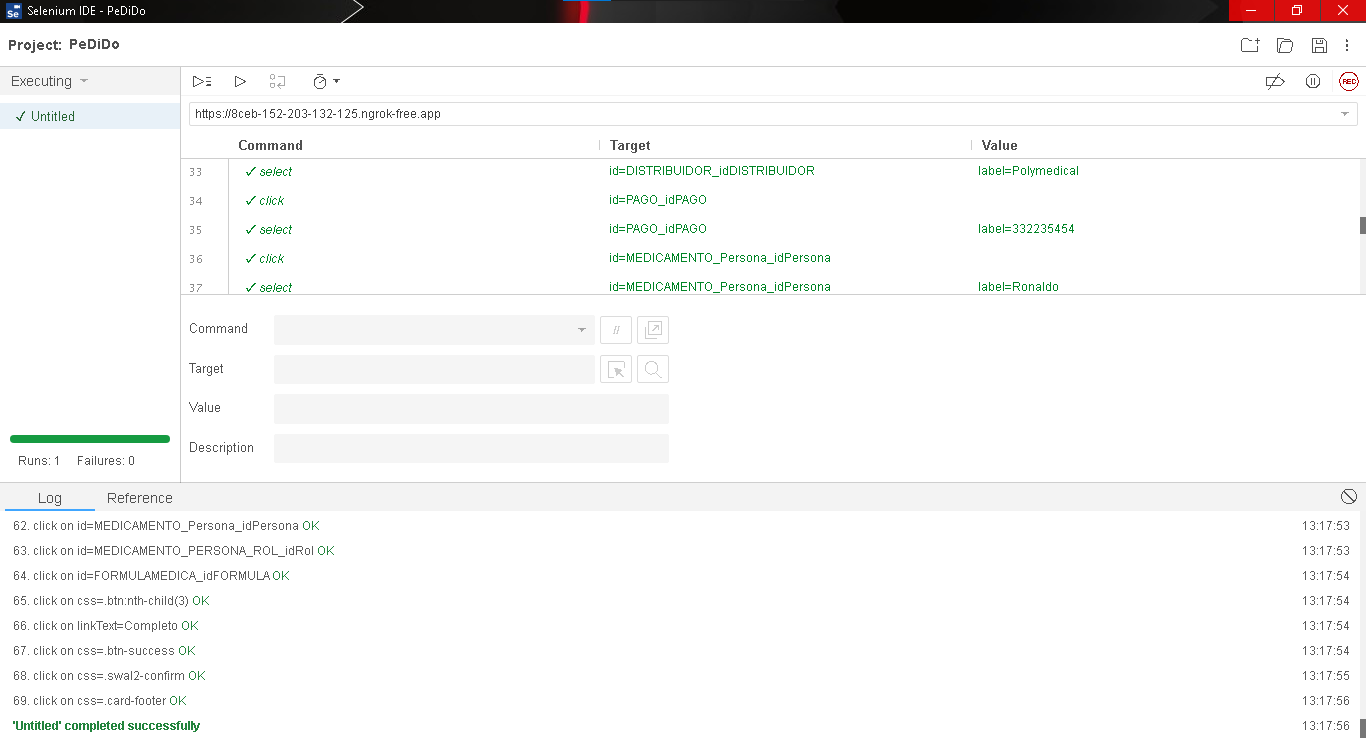
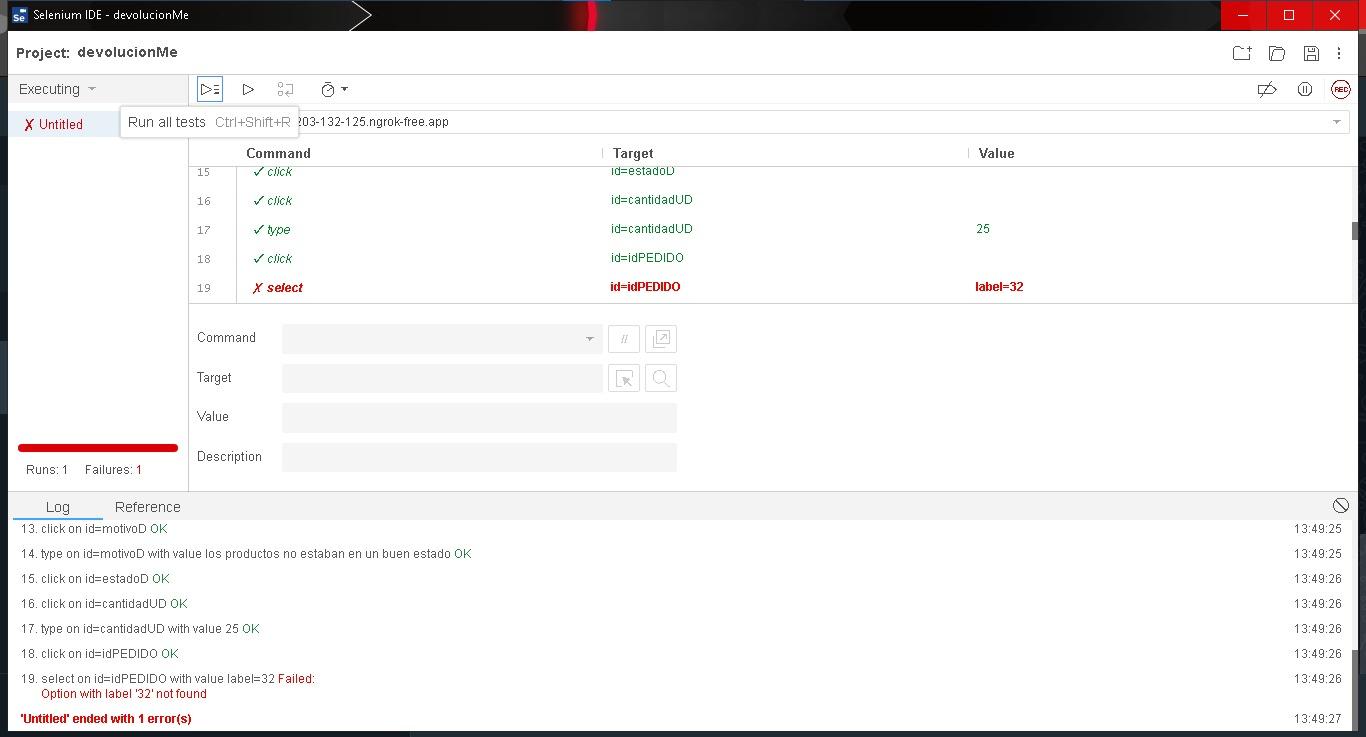
} else {

// Mensaje de error al actualizar el pedido

}

}

?>

  
  
  
  
  
Prueba 8: creación de devolución del pedido y editar devolución del pedido muestra un error al momento de crearlo.  
  
  
Prueba 9: creación de devolución del pedido y editar devolución del pedido se evidencian cambios en la base de datos exitosamente  
Código evaluado: (Crear)  
<?php

include("../../database.php");

if ($\_POST) {

// Recolectar datos (método post)

$cantidadD = (isset($\_POST["cantidadD"]) ? $\_POST["cantidadD"] : "");

$nombreProducto = (isset($\_POST["nombreProducto"]) ? $\_POST["nombreProducto"] : "");

$estadoD = (isset($\_POST["estadoD"]) ? $\_POST["estadoD"] : "");

$motivoD = (isset($\_POST["motivoD"]) ? $\_POST["motivoD"] : "");

$cantidadUD = (isset($\_POST["cantidadUD"]) ? $\_POST["cantidadUD"] : "");

$idPEDIDO = (isset($\_POST["PEDIDO\_idPEDIDO"]) ? $\_POST["PEDIDO\_idPEDIDO"] : "");

// Insertar la devolución en la tabla devoluciones

$sentencia = $conexion->prepare("INSERT INTO devoluciones (idDevoluciones, cantidadD, nombreProducto, estadoD, motivoD, cantidadUD, PEDIDO\_idPEDIDO)

VALUES (null, :cantidadD, :nombreProducto, :estadoD, :motivoD, :cantidadUD, :PEDIDO\_idPEDIDO)");

// Asignar valores

$sentencia->bindParam(":cantidadD", $cantidadD);

$sentencia->bindParam(":nombreProducto", $nombreProducto);

$sentencia->bindParam(":estadoD", $estadoD);

$sentencia->bindParam(":motivoD", $motivoD);

$sentencia->bindParam(":cantidadUD", $cantidadUD);

$sentencia->bindParam(":PEDIDO\_idPEDIDO", $idPEDIDO);

// Ejecutar la inserción de la devolución

if ($sentencia->execute()) {

// Obtener el ID del medicamento asociado al pedido

$sentenciaIdMedicamento = $conexion->prepare("SELECT MEDICAMENTO\_idMEDICAMENTO FROM pedido WHERE idPEDIDO = :idPedido");

$sentenciaIdMedicamento->bindParam(":idPedido", $idPEDIDO);

$sentenciaIdMedicamento->execute();

$resultadoIdMedicamento = $sentenciaIdMedicamento->fetch(PDO::FETCH\_ASSOC);

if ($resultadoIdMedicamento) {

$MEDICAMENTO\_idMEDICAMENTO = $resultadoIdMedicamento['MEDICAMENTO\_idMEDICAMENTO'];

// Actualizar la cantidad de unidades del medicamento

$sentenciaUpdateMedicamento = $conexion->prepare("UPDATE medicamento SET cantidadUnidades = cantidadUnidades + :cantidadPedido WHERE idMEDICAMENTO = :medicamentoID");

$sentenciaUpdateMedicamento->bindParam(":cantidadPedido", $cantidadUD, PDO::PARAM\_INT);

$sentenciaUpdateMedicamento->bindParam(":medicamentoID", $MEDICAMENTO\_idMEDICAMENTO);

// Ejecutar la actualización del medicamento

if ($sentenciaUpdateMedicamento->execute()) {

// Cantidad de unidades actualizada correctamente

// Redireccionar o mostrar mensaje de éxito

header("Location: index.php");

exit();

} else {

// Mensaje de error al actualizar la cantidad de unidades

echo "Error al actualizar la cantidad de unidades del medicamento.";

}

} else {

// No se pudo obtener el ID del medicamento asociado al pedido

echo "Error al obtener el ID del medicamento asociado al pedido.";

}

} else {

// Mensaje de error al insertar la devolución

echo "Error al insertar la devolución.";

}

}

$sentenciaidPEDIDO = $conexion->prepare("SELECT idPEDIDO, Nombre\_Producto FROM pedido");

$sentenciaidPEDIDO->execute();

$IdPEDIDO = $sentenciaidPEDIDO->fetchAll(PDO::FETCH\_ASSOC);

// Obtener nombres de medicamentos para el formulario

$sentenciaNombresMedicamentos = $conexion->prepare("SELECT idMEDICAMENTO, nombreMedica FROM medicamento");

$sentenciaNombresMedicamentos->execute();

$nombresMedicamentos = $sentenciaNombresMedicamentos->fetchAll(PDO::FETCH\_ASSOC);

?>

(Editar):  
<?php

include("../../database.php");

if (isset($\_GET['txtID'])) {

$txtID = (isset($\_GET['txtID'])) ? $\_GET['txtID'] : "";

$sentencia = $conexion->prepare("SELECT \* FROM devoluciones WHERE idDevoluciones = :idDevoluciones");

$sentencia->bindParam(":idDevoluciones", $txtID);

$sentencia->execute();

$registro = $sentencia->fetch(PDO::FETCH\_LAZY);

if ($registro) {

$cantidadori= $registro["cantidadUD"];

$cantidadD = $registro["cantidadD"];

$nombreProducto = $registro["nombreProducto"];

$estadoD = $registro["estadoD"];

$motivoD = $registro["motivoD"];

$cantidadUD = $registro["cantidadUD"];

$idPEDIDO = $registro["PEDIDO\_idPEDIDO"];

}

}

if ($\_POST) {

// Recolectar datos (método post)

$cantidadD = (isset($\_POST["cantidadD"]) ? $\_POST["cantidadD"] : "");

$nombreProducto = (isset($\_POST["nombreProducto"]) ? $\_POST["nombreProducto"] : "");

$estadoD = (isset($\_POST["estadoD"]) ? $\_POST["estadoD"] : "");

$motivoD = (isset($\_POST["motivoD"]) ? $\_POST["motivoD"] : "");

$cantidadUD = (isset($\_POST["cantidadUD"]) ? $\_POST["cantidadUD"] : "");

$idPEDIDO = (isset($\_POST["PEDIDO\_idPEDIDO"]) ? $\_POST["PEDIDO\_idPEDIDO"] : "");

// Obtener la cantidad original del medicamento

$sentenciaGetMedicamentoCantidad = $conexion->prepare("SELECT cantidadUnidades FROM medicamento WHERE idMEDICAMENTO IN (SELECT MEDICAMENTO\_idMEDICAMENTO FROM pedido WHERE idPEDIDO = :idPedido)");

$sentenciaGetMedicamentoCantidad->bindParam(":idPedido", $idPEDIDO);

$sentenciaGetMedicamentoCantidad->execute();

$resultado = $sentenciaGetMedicamentoCantidad->fetch(PDO::FETCH\_ASSOC);

if ($resultado) {

// Convertir a números enteros

$cantidadUD = intval($cantidadUD);

$cantidadD = intval($cantidadD);

// Calcular la diferencia entre la cantidad original y la cantidad devuelta

$diferenciaCantidad = $cantidadUD - $cantidadori;

// Calcular la nueva cantidad del medicamento sumando la diferencia

$nuevaCantidad = $diferenciaCantidad;

// Actualizar la cantidad del medicamento

$sentenciaUpdateMedicamento = $conexion->prepare("UPDATE medicamento SET cantidadUnidades = cantidadUnidades + :nuevaCantidad WHERE idMEDICAMENTO IN (SELECT MEDICAMENTO\_idMEDICAMENTO FROM pedido WHERE idPEDIDO = :idPedido)");

$sentenciaUpdateMedicamento->bindParam(":nuevaCantidad", $nuevaCantidad);

$sentenciaUpdateMedicamento->bindParam(":idPedido", $idPEDIDO);

if ($sentenciaUpdateMedicamento->execute()) {

// Actualizar la devolución

$sentenciaUpdateDevolucion = $conexion->prepare("UPDATE devoluciones SET cantidadD = :cantidadD, nombreProducto = :nombreProducto, estadoD = :estadoD, motivoD = :motivoD, cantidadUD = :cantidadUD, PEDIDO\_idPEDIDO = :PEDIDO\_idPEDIDO WHERE idDevoluciones = :idDevoluciones");

$sentenciaUpdateDevolucion->bindParam(":cantidadD", $cantidadD);

$sentenciaUpdateDevolucion->bindParam(":nombreProducto", $nombreProducto);

$sentenciaUpdateDevolucion->bindParam(":estadoD", $estadoD);

$sentenciaUpdateDevolucion->bindParam(":motivoD", $motivoD);

$sentenciaUpdateDevolucion->bindParam(":cantidadUD", $cantidadUD);

$sentenciaUpdateDevolucion->bindParam(":PEDIDO\_idPEDIDO", $idPEDIDO);

$sentenciaUpdateDevolucion->bindParam(":idDevoluciones", $txtID);

if ($sentenciaUpdateDevolucion->execute()) {

$mensaje = "Registro Actualizado";

header("Location:index.php?mensaje=" . $mensaje);

exit();

} else {

echo "Error al actualizar la devolución.";

}

} else {

echo "Error al actualizar la cantidad del medicamento.";

}

} else {

echo "Error al obtener la cantidad original del medicamento.";

}

}

$sentenciaidPEDIDO = $conexion->prepare("SELECT idPEDIDO, Nombre\_Producto FROM pedido");

$sentenciaidPEDIDO->execute();

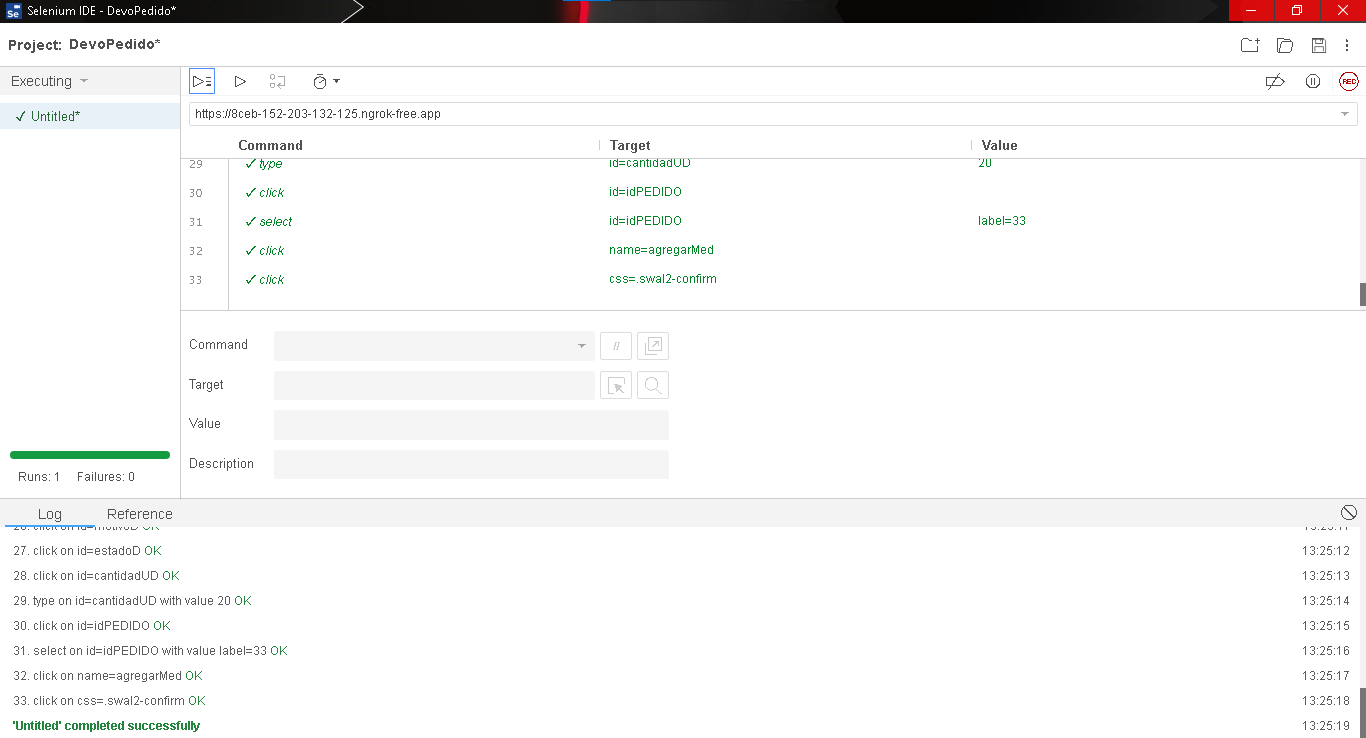
$IdPEDIDO = $sentenciaidPEDIDO->fetchAll(PDO::FETCH\_ASSOC);

$sentenciaNombresMedicamentos = $conexion->prepare("SELECT idMEDICAMENTO, nombreMedica FROM medicamento");

$sentenciaNombresMedicamentos->execute();

$nombresMedicamentos = $sentenciaNombresMedicamentos->fetchAll(PDO::FETCH\_ASSOC);

?>



Prueba 10: Eliminación de la devolución hecha y eliminación del pedido devuelto  
Código evaluado: (Devolucion)  
if (isset($\_GET['txtID'])) {

$txtID = (isset($\_GET['txtID'])) ? $\_GET['txtID'] : "";

$sentencia = $conexion->prepare("DELETE FROM devoluciones WHERE idDevoluciones =:idDevoluciones");

$sentencia->bindParam(":idDevoluciones", $txtID);

$sentencia->execute();

$mensaje = "Registro eliminado";

header("Location:index.php?mensaje=" . $mensaje);

}

(Pedido):  
<?php

include("../../database.php");

if (isset($\_GET['txtID'])) {

$txtID = (isset($\_GET['txtID'])) ? $\_GET['txtID'] : "";

$sentencia = $conexion->prepare("DELETE FROM pedido WHERE idPEDIDO =:idPEDIDO");

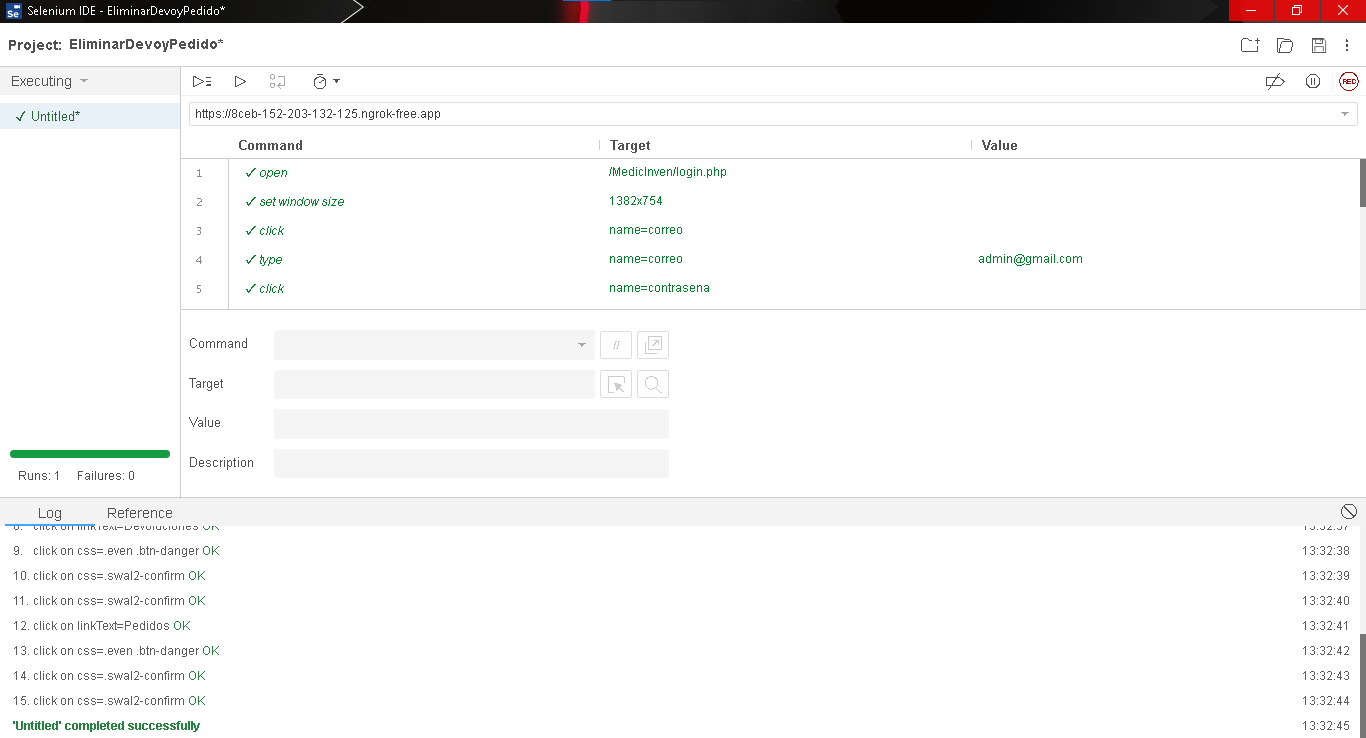
$sentencia->bindParam(":idPEDIDO", $txtID);

$sentencia->execute();

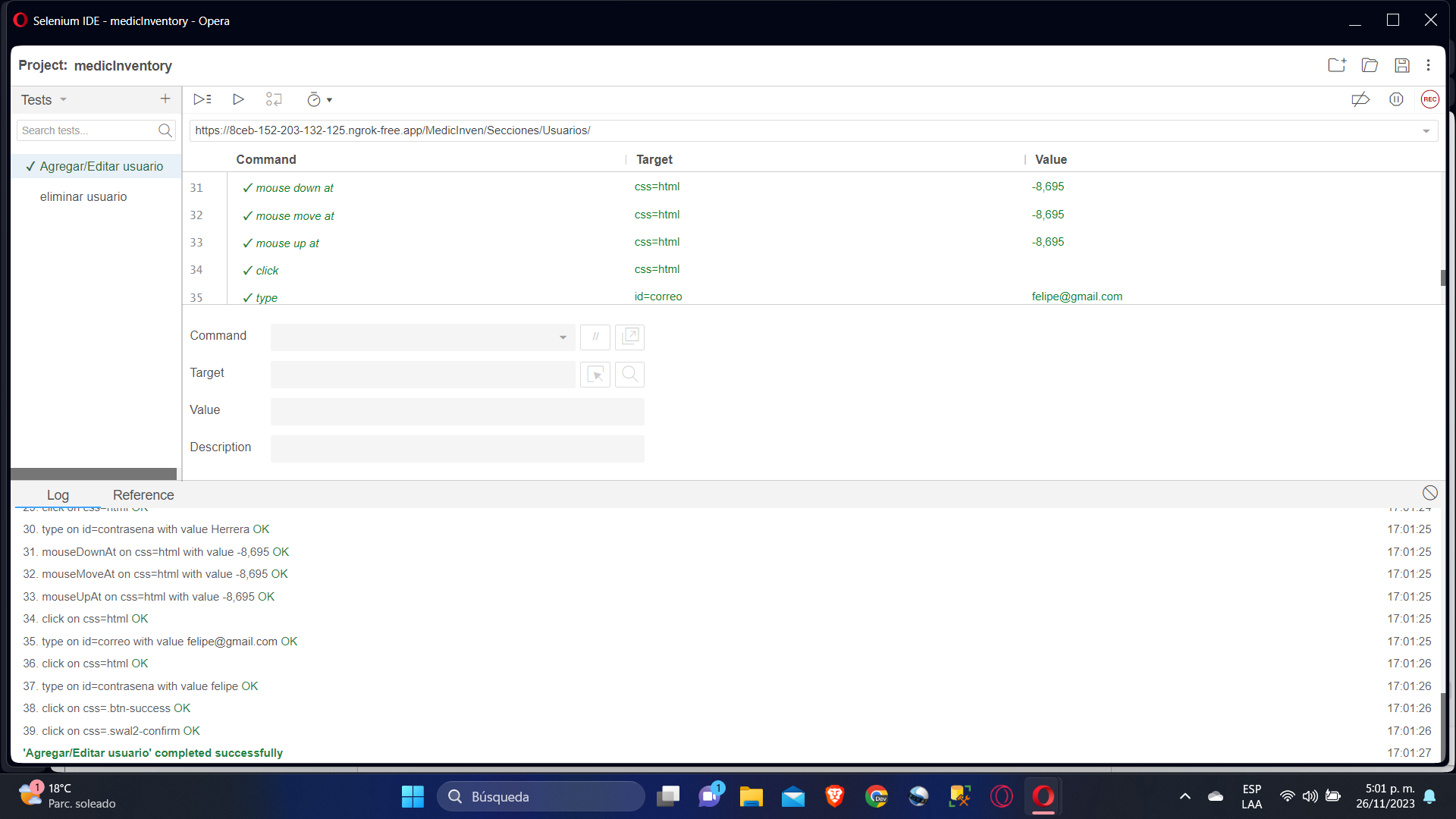
$mensaje = "Registro eliminado";

header("Location:index.php?mensaje=" . $mensaje);

}



Prueba 11: Creación de un nuevo usuario y la edición del mismo



Código Agregar usuario:

<?php

include("../../database.php");

if ($\_POST) {

print\_r($\_POST);

// Recolectar datos (método post)

// Recolectar datos (método post)

$persona = (isset($\_POST["idPersona"]) ? $\_POST["idPersona"] : "");

$nombreP = (isset($\_POST["nombreP"]) ? $\_POST["nombreP"] : "");

$apellidosP = (isset($\_POST["apellidosP"]) ? $\_POST["apellidosP"] : "");

$telefonoP = (isset($\_POST["telefonoP"]) ? $\_POST["telefonoP"] : "");

$numeroP = (isset($\_POST["numeroP"]) ? $\_POST["numeroP"] : "");

$correo = (isset($\_POST["correo"]) ? $\_POST["correo"] : "");

$cedulaP = (isset($\_POST["cedulaP"]) ? $\_POST["cedulaP"] : "");

$idRol = (isset($\_POST["idRol"]) ? $\_POST["idRol"] : "");

$contrasena = (isset($\_POST["contrasena"]) ? $\_POST["contrasena"] : "");

// Inserción de los datos

$sentencia = $conexion->prepare("INSERT INTO persona (idPersona,nombreP,apellidosP,telefonoP,numeroP,correo,cedulaP,contrasena,ROL\_idRol)

VALUES (null, :nombreP, :apellidosP, :telefonoP, :numeroP, :correo, :cedulaP, :contrasena, :idRol) ");

// Asignar valores que tienen un solo :variable

$sentencia->bindParam(":nombreP", $nombreP);

$sentencia->bindParam(":apellidosP", $apellidosP);

$sentencia->bindParam(":telefonoP", $telefonoP);

$sentencia->bindParam(":numeroP", $numeroP);

$sentencia->bindParam(":correo", $correo);

$sentencia->bindParam(":cedulaP", $cedulaP);

$sentencia->bindParam(":idRol", $idRol);

$sentencia->bindParam(":contrasena", $contrasena);

$sentencia->execute();

$mensaje = "Registro agregado";

header("Location:index.php?mensaje=" . $mensaje);

}

// Obtener roles desde la base de datos

$sentenciaRoles = $conexion->prepare("SELECT idRol, nombreRol FROM rol");

$sentenciaRoles->execute();

$roles = $sentenciaRoles->fetchAll(PDO::FETCH\_ASSOC);

?>

Código Editar:

<?php

include("../../database.php");

if (isset($\_GET['txtID'])) {

$txtID = (isset($\_GET['txtID'])) ? $\_GET['txtID'] : "";

$sentencia = $conexion->prepare("SELECT \* FROM persona WHERE idPersona=:idPersona");

$sentencia->bindParam(":idPersona", $txtID);

$sentencia->execute();

$registro = $sentencia->fetch(PDO::FETCH\_LAZY); //ejecuta la consulta preparada y recupera la siguiente fila de resultados de la base de datos

if ($registro) {

$nombre = $registro["nombre"];

$apellidos = $registro["apellidos P"];

$teléfono = $registro["teléfono"];

$número = $registro["número"];

$correo = $registró["correo"];

$cedula = $registro["cedula"];

$contraseña = $registro["contraseña"];

}

}

if ($\_POST) {

$txtID = (isset($\_POST['txtID'])) ? $\_POST['txtID'] : "";

$nombre = (isset($\_POST["nombre"]) ? $\_POST["nombre"] : "");

$apellidos = (isset($\_POST["apellidos"]) ? $\_POST["apellidos P"] : "");

$teléfono = (isset($\_POST["telefono"]) ? $\_POST["telefono"] : "");

$número = (isset($\_POST["numero"]) ? $\_POST["numero"] : "");

$correo = (isset($\_POST["correo"]) ? $\_POST["correo"] : "");

$cédula = (isset($\_POST["cedula"]) ? $\_POST["cedula"] : "");

$contraseña = (isset($\_POST["contrasena"]) ? $\_POST["contrasena"] : "");

$sentencia = $conexion->prepare("UPDATE persona SET nombreP=:nombre, apellidos=:apellidos, telefonoP=:telefonoP, numeroP=:numeroP, correo=:correo, cedula=:cédula, contrasena=:contrasena WHERE idPersona=:idPersona");

$sentencia->bindParam(":nombre", $nombre);

$sentencia->bindParam(":apellidos", $apellidos P);

$sentencia->bindParam(":teléfono", $teléfono);

$sentencia->bindParam(":número", $número);

$sentencia->bindParam(":correo", $correo);

$sentencia->bindParam(":cedula", $cedula);

$sentencia->bindParam(":idPersona", $txtID);

$sentencia->bindParam(":contrasena", $contrasena);

$sentencia->execute();

$mensaje = "Registro Actualizado";

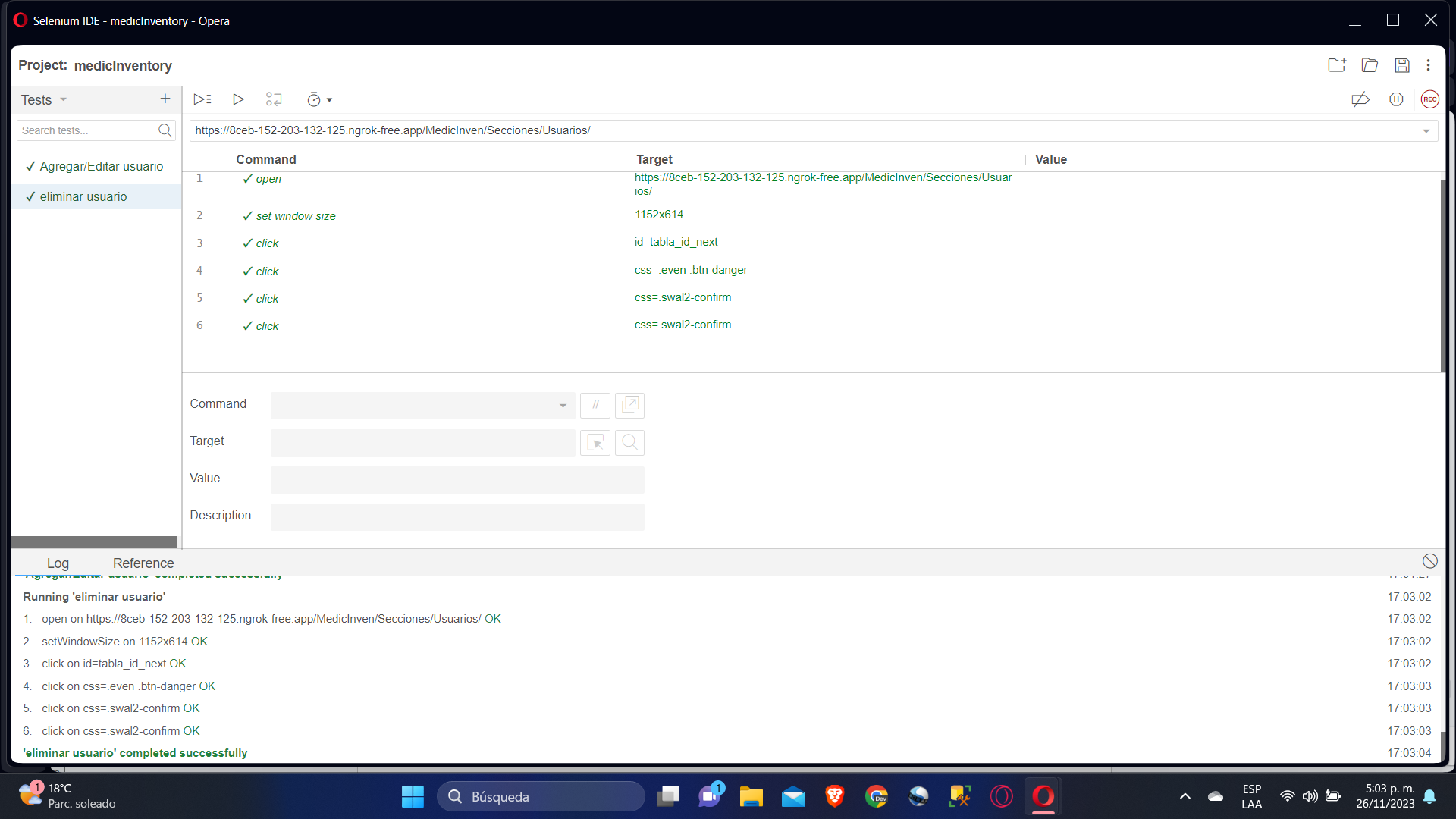
Header("Location: index.php?mensaje=" . $mensaje);

exit();

}

?>

Prueba 12: Eliminación del usuario creado y editado previamente



Código Delete:

if (isset($\_GET['txtID'])) {

$txtID = (isset($\_GET['txtID'])) ? $\_GET['txtID'] : "";

$sentencia = $conexion->prepare("DELETE FROM persona WHERE idPersona=:idPersona");

$sentencia->bindParam(":idPersona", $txtID);

$sentencia->execute();

$mensaje = "Registro eliminado";

header("Location:index.php?mensaje=" . $mensaje);

}

# GLOSARIO

|  |  |
| --- | --- |
| **Término** | **Descripción** |
|  |  |
|  |  |
|  |  |

# BIBLIOGRAFÍA Y REFERENCIAS

|  |  |
| --- | --- |
| **Referencia** | **Título** |
| Ref. 1 |  |
| Ref. 2 |  |
|  |  |
|  |  |