



## **Plan Maestro de pruebas**

Versión 2.0



## **Tabla de contenido**

Introducción .....	3
Objetivo .....	3
Documentos relacionados.....	3
Alcance.....	4
Contexto de las pruebas .....	4
Objetivo de las pruebas.....	4
Estrategia de las pruebas .....	5
Diseño de casos de prueba.....	5
Técnica de caja blanca.....	5
Técnica de caja negra .....	5
Planificación de las pruebas.....	6
Entorno y configuración .....	6
Criterios de inicio.....	6
Criterios de aprobación o rechazo .....	6
Orden de ejecución de pruebas.....	7
Prueba de Caja Blanca .....	7
Pruebas de Caja negra.....	7
Prueba unitaria.....	7
Pruebas de integración .....	9
Pruebas del sistema .....	<b>¡Error! Marcador no definido.</b>
Ejecución de las pruebas .....	12
Ejecución de prueba de Caja Blanca .....	14
Ejecución de pruebas de Caja Negra .....	19
Pruebas unitarias .....	19
Prueba de integración .....	22

## Introducción

Proyecto(s)		Tipo de Proyecto
INFOTEC		Proyecto de Desarrollo de Software Académico.
Equipo de Proyecto		
Integrantes de Equipo	Ana Valeria Mendoza	
	Iván Gregorio Garzón	
	Johan Sebastián Zarate	
	Carlos Andrés Fraile González	

## Objetivo

Este documento, tiene como finalidad entregar las pautas y definir la estrategia que se seguirá para llevar a cabo la certificación del software INFOTEC desarrollado para la empresa Lubrimotos la 30.

El objetivo general del plan es establecer los tiempos y condiciones para la aplicación de las pruebas para así poder obtener, un sistema que pueda ser completado aceptación del cliente y pueda operar con la totalidad de las funciones requeridas para su funcionamiento.

## Documentos relacionados

Nombre	Descripción	URL
<b>Casos de Uso</b>	Expone los casos de uso que contiene el sistema de información INFOTEC	
<b>Manual de instalación</b>	Describe los pasos de instalación del sistema de información INFOTEC	
<b>Manual Técnico</b>	Proporciona una guía del desarrollo de la interfaz y de la instalación del sistema de información INFOTEC.	



### ***Alcance***

El presente documento contiene la guía para la planeación, ejecución y manejo de los resultados de las pruebas que serán aplicadas al sistema de información INFOTEC.

### **Contexto de las pruebas**

El sistema de información INFOTEC es un sistema de facturación e inventario diseñado para la industria automotriz. Cuenta con los siguientes módulos.

<b>Perfil</b>	<b>Módulos</b>
Administrador	<ul style="list-style-type: none"><li>• Gestionar Rol</li><li>• Gestionar Usuarios</li><li>• Gestionar Cliente</li><li>• Gestionar Vehículo</li><li>• Gestionar Categorías</li><li>• Gestionar Producto</li><li>• Gestionar Servicios</li><li>• Gestionar Proveedores</li><li>• Gestionar Facturas</li><li>• Gestionar inventarios</li></ul>
Vendedor	<ul style="list-style-type: none"><li>• Gestionar Cliente</li><li>• Gestionar Vehículo</li><li>• Gestionar Categorías</li><li>• Gestionar Producto</li><li>• Gestionar Servicios</li><li>• Gestionar Facturas</li><li>• Gestionar inventarios</li></ul>

### ***Objetivo de las pruebas***

Las pruebas para validar el correcto funcionamiento de los módulos anteriormente descritos son:

- Creación, modificación, eliminación y almacenamiento de los datos ingresados en cada uno de los módulos.
- La respuesta y realización de las transacciones de cada módulo.



- Respuesta del sistema en la consulta de inventarios sea lógica y verdadera.
- La visualización de los datos registrados con anterioridad.

### ***Estrategia de las pruebas***

Para la aplicación de las pruebas al sistema de información INFOTEC se incluirán los siguientes tipos de pruebas.

#### ***- Pruebas Unitarias***

Tiene como objetivo validar que el comportamiento de los objetos creados para el sistema de información y la lógica de estos sea correcta, además este nivel de prueba permite definir si se cumplen con los requisitos pedidos por el cliente. Esta prueba se realiza en cada módulo o componente del sistema.

#### ***- Pruebas de integración***

Tiene como objetivo comprobar la integración adecuada de los componentes del sistema, además, que cada componente cumpla con su tarea asignada y se adapten a los requisitos establecidos. También se probará la interacción del sistema de información con la base de datos.

### ***Diseño de casos de prueba***

Teniendo en cuenta la estrategia de pruebas, donde se definió el tipo de pruebas que se van a emplear, usaremos las siguientes técnicas

#### ***Técnica de caja blanca***

Esta técnica tiene como objetivo analizar el diseño, el código y la estructura interna del sistema de información. Permite encontrar errores en el código y garantizar que no exista código muerto.

#### ***Técnica de caja negra***

Esta técnica tiene como objetivo analizar la funcionalidad del sistema, en esta técnica no se tiene en cuenta la estructura interna del código. Este tipo de técnica se enfoca en validar que la interacción en cada módulo sea la esperada y cumpla con los requisitos especificados por el cliente. Esta técnica se llevará a cabo mediante los casos de uso y la partición de equivalencia.



## **Planificación de las pruebas**

A continuación, se exponen los lineamientos en los cuales se planifica como serán ejecutadas las pruebas del sistema INFOTEC.

### ***Entorno y configuración***

Para un óptimo desarrollo de las pruebas se requiere tener en disposición entornos mencionados a continuación:

- **Equipo INFOTEC:**
  - o Dell con procesador Core I7, 8 Gb de RAM, HDD 250 GB y Windows Profesional
- **Herramienta para pruebas:**
  - o Cypress versión 11.2.0
- **Base de Datos Maria:** Reside en equipo de del grupo de trabajo INFOTEC detallado en el punto a del entorno y configuración de las pruebas
  - o Base de Datos: INFOTEC
  - o Servidor BD: Maria DB
  - o Datos: localhost/phpmyadmin

### ***Criterios de inicio***

Aceptación del plan de pruebas. Revisión y aceptación del documento que contiene los casos de pruebas para la certificación del proyecto.

Aceptación de paquetes. Revisión y aceptación de los paquetes de desarrollo, y que este cumpla con las condiciones de aceptación.

Aceptación de ambiente. Revisión y aceptación del ambiente de certificación, y que este cumpla con las condiciones de aceptación.

### ***Criterios de aprobación o rechazo***

**Errores Graves:** información crítica presentada erróneamente, información mal registrada en la base de datos, caídas de programas, incumplimiento de objetivos en funciones principales, etc.

**Errores Medios (comunes):** errores en documentos impresos que se entregan a personas ajenas a la organización, errores en presentación de datos, incumplimiento de objetivos en funciones secundarias, caídas de programas auxiliares, etc.



**Errores Leves:** errores en presentación de datos secundarios, no adecuación a estándares, comportamientos correctos pero diferentes en situaciones similares, dificultades de operación, etc.

### ***Orden de ejecución de pruebas***

Las pruebas se llevarán a cabo de la siguiente forma:

- Descarga del sistema de información almacenado en [GitHub](#)
- Configuración del equipo del cliente y del sistema de información (Ver manual de instalación)
- Configuración de Cypress
- Ejecución de prueba de Caja Blanca
- Ejecución de pruebas de Caja Negra
  - o Pruebas unitarias
  - o Pruebas de integración

### ***Prueba de Caja Blanca***

La prueba de Caja blanca que se llevará a cabo será la examinación del código interno que permite ingresar, eliminar y modificar los datos del módulo rol.

### ***Pruebas de Caja negra***

Las pruebas de Caja negra que se ejecutarán serán las siguientes según los tipos mencionados en la estrategia de las pruebas:

#### ***Prueba unitaria***

Por medio de la herramienta de Cypress se llevará a cabo la prueba de los casos de prueba mencionados a continuación:

ID REQUERIMIENTO		RF02		
DESCRIPCION REQUERIMIENTO		Crear Usuario		
ID CASO USO		CU07		
DESCRIPCION CASO DE USO		Crear Usuario		
REQUERIMIENTO DE AMBIENTE		Ambiente Pruebas		
ID	Caso de Prueba	Fecha	Área Funcional / SubProceso	Funcionalidad / característica
CP02	El sistema debe permitir al usuario con rol administrador crear usuario satisfactoriamente según rol correspondiente.		Modulo Usuarios	Creación de usuarios
Precondición / Datos de Entrada		Descripción		Resultado Esperado
1) Debe existir un usuario con rol administrador registrado en el sistema. 2) Debe existir el rol creado en la base de datos 3) Debe diligenciar el formulario de creación de usuarios en su totalidad y debe estar logeado como Administrador. ----- usuario: admin@correo.com pass: 12345		Realizar el registro correspondiente de datos personales del usuario para el ingreso al sistema.		Ver el usuario registrado en la tabla



ID REQUERIMIENTO		RF03		
DESCRIPCION REQUERIMIENTO		Crear Cliente		
ID CASO USO		CU031		
DESCRIPCION CASO DE USO		Crear cliente		
REQUERIMIENTO DE AMBIENTE		Ambiente Pruebas		
ID	Caso de Prueba	Fecha	Área Funcional / SubProceso	Funcionalidad / característica
CP03	El sistema debe permitir al usuario crear un cliente satisfactoriamente.		Modulo Cliente	Creación de Cliente
Precondición / Datos de Entrada		Descripción		Resultado Esperado
1) Debe existir un usuario con rol administrador/Vendedor registrado en el sistema. 2) Debe diligenciar el formulario de creación de clientes en su totalidad y debe estar logeado como Administrador/vendedor. ----- usuario: admin@correo.com pass: 12345		Realizar el registro correspondiente de datos personales del cliente para el ingreso al sistema.		Ver el cliente registrado en la tabla

### **Pruebas de integración**

Por medio de la herramienta Cypress se llevará a cabo la prueba integración entre los módulos de categoría y producto, donde se podrá evidenciar que los datos almacenados en el módulo de categoría pueden ser visibles en el módulo de producto según como se planteó el requerimiento del cliente. Además, se podrá verificar que estos módulos del sistema funcionan adecuadamente en conjunto. Estas pruebas tendrán un enfoque ascendente que se describe de la siguiente manera:

<b>Identificador</b>	Prueba de integración #1
<b>Nombre</b>	Crear una categoría y un producto
<b>Requerimiento de la prueba</b>	Creación de una categoría y un producto
<b>Precondición</b>	
<ul style="list-style-type: none"> <li>- El administrador y/o el vendedor pueden crear categorías y un producto ingresando al sistema con su usuario y contraseña.</li> <li>- No se puede crear un producto sin antes crear la categoría.</li> </ul>	
<b>Objetivo</b>	
<ul style="list-style-type: none"> <li>- Permitir que el administrador y/o el vendedor puedan ingresar los datos de una categoría.</li> <li>- Permitir que el administrador y/o el vendedor puedan crear productos enlazados a las diferentes categorías.</li> </ul>	
<b>Pasos</b>	
<ol style="list-style-type: none"> <li>1. El administrador y/o el vendedor deben iniciar sesión con sus credenciales anteriormente asignadas.</li> <li>2. En el menú de tarjetas "Productos" identificar la tarjeta que dice "categorías", dar click sobre ella.</li> <li>3. Allí podrá ver el formulario con los campos requeridos para agregar una nueva categoría y una tabla que contiene las categorías que ya están almacenados con las funciones de modificar y eliminar</li> <li>4. El administrador y/o el vendedor debe ingresar los datos del cliente y darle en el botón "Enviar"</li> <li>5. El nuevo registro creado se podrá ver en la tabla</li> <li>6. Para ingresar un producto asociado a la categoría que acaba de registrar, el administrador y/o el vendedor debe dar click en el botón "Crear Productos".</li> <li>7. Allí podrá ver el formulario con los campos requeridos para la creación de un nuevo producto y una tabla que contiene los productos que anteriormente fueron creados.</li> <li>8. En la casilla que dice "Elija la categoría" podrá encontrar la categoría que creo anteriormente y seleccionarla, y a continuación podrá completar los demás datos para la creación del producto.</li> <li>9. Cuando termine de diligenciar todos los campos del formulario debe dar click en el botón "Enviar" para guardar los datos</li> <li>10. En la tabla lateral podrá encontrar el producto creado asociado a la categoría que anteriormente había creado.</li> <li>11. Fin de la prueba de integración.</li> </ol>	



<b>Resultado esperado</b>
<ul style="list-style-type: none"><li>- La nueva categoría se puede visualizar en la vista de categorías y en el selector "elija una categoría" del módulo de productos.</li><li>- El producto queda almacenado con el nombre de la categoría.</li><li>- Los datos ingresados se guardan correctamente en la base de datos.</li></ul>



### Ejecución de las pruebas

Siguiendo lo descrito anteriormente en el orden de ejecución de las pruebas, para iniciar a ejecutar las pruebas se debe realizar la descarga del sistema de información almacenado en [GitHub](#) y realizar la configuración del equipo del cliente y del sistema de información (Ver manual de instalación)

Teniendo el sistema de información y el equipo configurados, continuamos con la configuración de [Cypress](#) y [Node.js](#), estos dos programas se descargan cada uno de su pagina oficial.

Estando dentro de la carpeta del sistema de información, en la terminal de Visual Studio Code escribimos "npm init"

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS C:\xam  \htdocs\Infotec_definitivo-main>
PS C:\xampp\htdocs\Infotec_definitivo-main> npm init
```

Pedirá llenar varios campos con la siguiente información:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (infotec_definitivo-main)
version: (1.0.0)
description:
entry point: (cypress.config.js)
test command: test
git repository:
keywords: cypress
author: Infotec
license: (ISC)
About to write to C:\xampp\htdocs\Infotec_definitivo-main\package.json:
```

Verificara la información que se ingreso y luego escribiremos "npm install cypress -save-dev" para realizar la instalación de Cypress en el sistema.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

}
Is this OK? (yes)
PS C:\xampp\htdocs\Infotec_definitivo-main> npm install cypress --save-dev

added 172 packages, and audited 173 packages in 17s

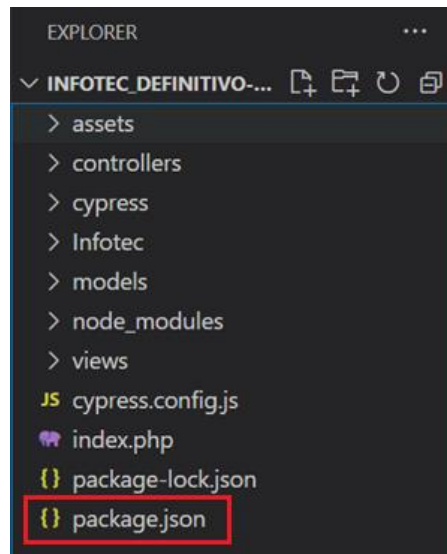
35 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

## Plan Maestro de pruebas



Este proceso nos creara un archivo que se llama "package.json"



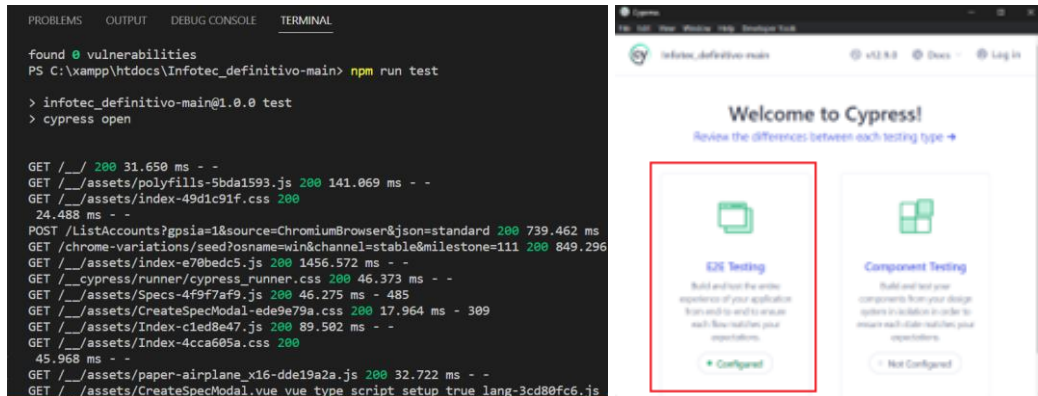
Abrimos el archivo y en donde dice "scripts" se modificará y se colocará "test": "cypress open"

```
{ package.json X
{} package.json > ...
1  {
2    "name": "infotec_definitivo-main",
3    "version": "1.0.0",
4    "description": "",
5    "main": "cypress.config.js",
6    "scripts": {
7      "test": "cypress open"
8    },
9    "keywords": [
10     "cypress"
11   ],
12   "author": "Infotec",
13   "license": "ISC",
14   "devDependencies": {
15     "cypress": "^12.9.0"
16   }
17 }
18
```

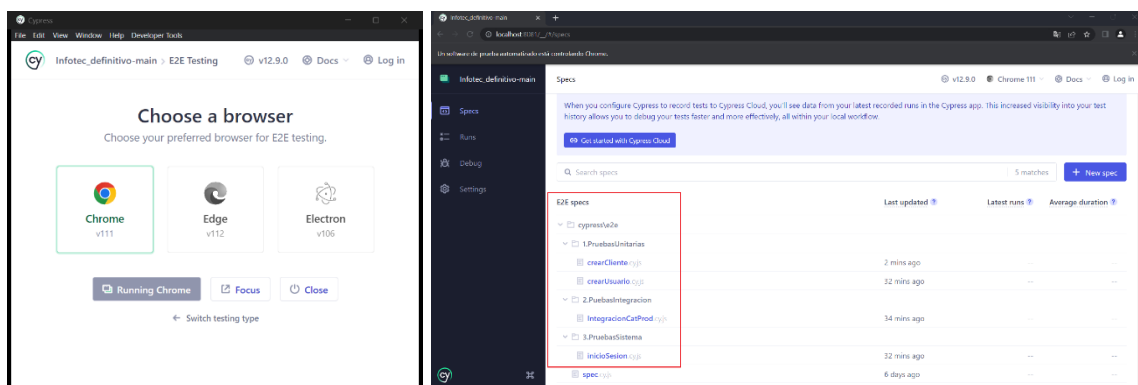
## Plan Maestro de pruebas



Luego volveremos a la terminal y escribiremos “npm run test” y esto iniciara una ventana de Cypress.



Se selecciona “E2E Testing” y se elige el navegador en el que se desea iniciar las pruebas y podrá ver las pruebas que están previamente diseñadas.

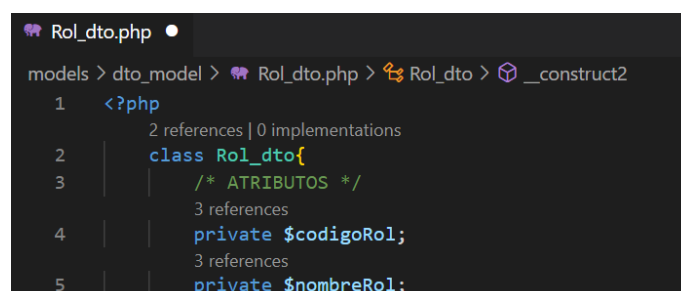


Al llegar a este punto de la instalación se realizará la ejecución de pruebas de Caja Negra.

### Ejecución de prueba de Caja Blanca

La prueba de técnica de Caja Blanca realizada en el modulo de rol tenía como objetivo validar que la consulta, ingreso, modificación y eliminación de los registros se realiza de forma correcta.

Para poder capturar los datos se crearon atributos



Se generan las funciones para poder traer los registros desde la base de datos.

```
models > dto_model > Rol_dto.php > Rol_dto
14 }
15 // Constructor
    0 references | 0 overrides
16 public function __construct2($codigoRol,$nombreRol){
17     $this->codigoRol = $codigoRol;
18     $this->nombreRol = $nombreRol;
19 }
20 /* MÉTODOS DE ACCESO: SETTER Y GETTERS*/
21 // Código Rol
    0 references | 0 overrides
22 public function setCodigoRol($codigoRol){
23     $this->codigoRol = $codigoRol;
24 }
    0 references | 0 overrides
25 public function getCodigoRol(){
26     return $this->codigoRol;
27 }
28 // Nombre Rol
    0 references | 0 overrides
29 public function setNombreRol($nombreRol){
30     $this->nombreRol = $nombreRol;
31 }
    0 references | 0 overrides
32 public function getNombreRol(){
33     return $this->nombreRol;
34 }
```

Se crean las funciones para realizar las consultas en la base de datos según sea el requerimiento, para crear un nuevo rol.

```
models > dao_model > Rol_dao.php > Rol_dao > createRol
0 references | 0 overrides
13 public function createRol($rol_dto){
14     try {
15         // Crear la Consulta
16         $sql = 'INSERT INTO roles VALUES (
17             '.$rol_dto->getCodigoRol().',
18             "'.$rol_dto->getNombreRol().'"
19         )';
20         // Preparar la BBDD para la consulta
21         mysqli_query($this->pdo,$sql);
22     } catch (Exception $e) {
23         die("....".$e->getMessage());
24     }
25 }
26 }
```

Función para leer los roles registrados en la base de datos.

```
Rol_dao.php X
models > dao_model > Rol_dao.php > Rol_dao > readRolDao

28 // # Consultar Roles
29 // 0 references | 0 overrides
29 public function readRolDao(){
30     try {
31         // Asignar una consulta al atributo $sql
32         $sql = 'SELECT * FROM ROLES';
33         // Creamos la variable $dbh y le asignamos la conexión y la consulta $sql
34         $dbh = mysqli_query($this->pdo,$sql);
35         return $dbh;
36     } catch (Exception $e) {
37         die($e->getMessage());
38     }
39 }
```

Función para eliminar el registro de la base de datos.

```
Rol_dao.php X
models > dao_model > Rol_dao.php > Rol_dao > eliminarRolDao

0 references | 0 overrides
51 public function eliminarRolDao($rolid){
52     try {
53         $sql= 'DELETE FROM ROLES WHERE id_rol='.$strval($rolid);
54         $dbh = mysqli_query($this->pdo,$sql);
55         return $dbh;
56     } catch (Exception $e) {
57         die($e->getMessage());
58     }
59 }
```

Función para modificar y almacenar los datos modificados.

```
Rol_dao.php X
models > dao_model > Rol_dao.php > Rol_dao > modificarRolDao

60 public function actualizarRolDao($idmod){
61     try {
62         $sql = 'SELECT * FROM ROLES where id_rol='.$idmod.'';
63         $dbh = mysqli_query($this->pdo,$sql);
64         return mysqli_fetch_row($dbh);
65     } catch (Exception $e) {
66         die($e->getMessage());
67     }
68 }
69 }
70 }

0 references | 0 overrides
71 public function modificarRolDao($codigoRol,$nombreRol){
72     try {
73         $sql = "UPDATE ROLES SET nombre_rol='".$nombreRol" where id_rol=$codigoRol";
74         $dbh = mysqli_query($this->pdo,$sql);
75         return $dbh;
76     } catch (Exception $e) {
77         die($e->getMessage());
78     }
79 }
80 }
```



Enlace de las funciones en el controlador, para la pagina principal donde se muestra los datos almacenados en la base de datos

```
Roles.php X
controllers > Roles.php > Roles > index
1  <?php
2  require_once "models/dto_model/Rol_dto.php";
3  require_once "models/dao_model/Rol_dao.php";
4
5  class Roles{
6      8 references
7      private $rolDao;
8      0 references | 0 overrides
9      public function __construct(){
10         $this->rolDao = new Rol_dao;
11     }
12     0 references | 0 overrides
13     public function index(){
14         $roles = $this->rolDao->readRolDao();
15         if ($_SERVER['REQUEST_METHOD'] == 'GET' && isset($_GET['id_rol'])) {
16             $result = $this->rolDao->consultarRolDao($_GET['id_rol']);
17             $rol_dto = new Rol_dto($result[0], $result[1]);
18             $rol_dto->setCodigoRol($result[0]);
19         }
20         elseif($_SERVER['REQUEST_METHOD'] == 'POST'){
21             // Capturar Datos
22             $rol_dto = new Rol_dto($_POST['id_rol'], $_POST['nombre_rol']);
23             $this->rolDao->createRol($rol_dto);
24             header("Location: ?c=Roles");
25         }
26         require_once "views/roles/admin/header_dash.php";
27         require_once "views/modules/0_rol/rol.view.php";
28         require_once "views/roles/admin/footer.php";
29     }
30 }
```

Para eliminar el registro.

```
Roles.php X
controllers > Roles.php > Roles > eliminar_rol
27 public function eliminar_rol(){
28     if ($_SERVER['REQUEST_METHOD'] == 'GET'){
29         // Capturar Datos
30         $this->rolDao->eliminarRolDao($_GET['id_rol']);
31     }
32     $roles = $this->rolDao->readRolDao();
33     require_once "views/roles/admin/header_dash.php";
34     require_once "views/modules/0_rol/rol.view.php";
35     require_once "views/roles/admin/footer.php";
36 }
```

Para modificar y guardar los datos modificados.

```

Roles.php X
controllers > Roles.php > Roles > modificar_rol
0 references | 0 overrides
37 public function editar_rol(){
38     if ($_SERVER['REQUEST_METHOD'] == 'GET') {
39         $editrol = $this->rolDao->actualizarRolDao($_GET['id_rol']);
40     }
41     require_once "views/roles/admin/header_dash.php";
42     require_once "views/modules/0_rol/rol.editar.php";
43     require_once "views/roles/admin/footer.php";
44 }
0 references | 0 overrides
45 public function modificar_rol(){
46     if ($_SERVER['REQUEST_METHOD'] == 'POST') {
47         $this->rolDao->modificarRolDao($_POST['id_rol'],$_POST['nombre_rol']);
48         header("Location: ?c=Roles");
49     }
50 }
51 }
52 ?>
    
```

Lectura de datos en la vista rol.view.php

```

rol.view.php X
views > modules > 0_rol > rol.view.php
41 </tr>/.text-center
42 </thead>
43 <tbody>
44 <?php
45 while($row=mysqli_fetch_array($roles)){
46     ?>
47     <tr>
48         <td class="text-center">
49             <?php echo $row['id_rol']?>
50         </td>/.text-center
51         <td class="text-center">
52             <?php echo $row['nombre_rol']?>
53         </td>/.text-center
54         <td class="text-center">
55             <a class="btn btn-warning" href="?c=Roles&a=editar_rol&id_rol=<?php echo $row['id_rol']?>">
56                 <i class="bi bi-pencil-square"></i>
57             </a>/.btn.btn-warning
58         </td>/.text-center
59         <td class="text-center">
60             <a class="btn btn-danger" href="?c=roles&a=eliminar_rol&id_rol=<?php echo $row['id_rol']?>">
61                 <i class="fa fa-trash"></i>
62             </a>/.btn.btn-danger
63         </td>/.text-center
64     </tr>
65 <?php
66 }
    
```

## Lectura de datos en la vista rol.editar.php

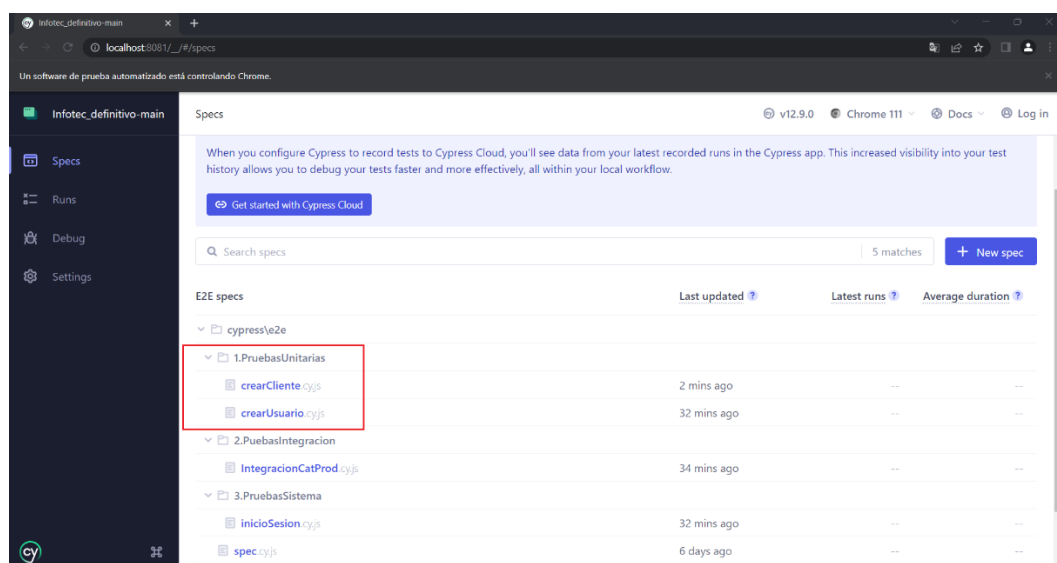
```
rol.editar.php X
views > modules > 0_rol > rol.editar.php
1 <div class="container">
2 <div class="row">
3 <div class="col-md-10">
4 <h1 class="titulos mt-1">Editar Rol</h1>
5 <hr>
6 <form class="row g-3 needs-validation" method="post" action="?c=Roles&
a=modificar_rol"novalidate>
7 <div class="col-md-6">
8 <label for="id_rol" class="form-label">Código Rol</label>
9 <input type="int" class="form-control mb-3" name="id_rol" placeholder="Codigo
Rol" value="<?php echo $editrol[0]?>" readonly >
10 </div>/.col-md-6
11 <div class="col-md-6">
12 <label for="nombre_rol" class="form-label">Nombre Rol</label>
13 <input type="text" class="form-control" id="nombre_rol" required
name="nombre_rol" placeholder="Nombre rol" value="<?php echo $editrol[1]?>">
14 </div>/.col-md-6
15 <input type="submit" class="btn btn-enviar mt-2">
16 </form>/.row.g-3.needs-validation
17 </div>/.col-md-10
18 </div>/.row
19 </div>/.container
20 </body>
```

## Ejecución de pruebas de Caja Negra

Para la ejecución de las siguientes pruebas es necesario ingresar a Cypress como se describió anteriormente. A continuación, se explicará cómo se ingresa a realizar cada una de las pruebas mencionadas

### Pruebas unitarias

En la ejecución de estas pruebas se realizó los dos casos de uso que se encontraban planificados anteriormente. En Cypress podrá encontrar la carpeta "1.PruebasUnitarias" donde se encuentran almacenadas las pruebas que se realizaron.



## Creación de Cliente

Se desarrollo un código donde la herramienta de Cypress completara los datos del cliente y el mismo diera click en el botón enviar. Así mismo, se validó que el registro quedara guardado.

**JS** crearCliente.cy.js X

```

cypress > e2e > 1.PruebasUnitarias > JS crearCliente.cy.js > ...
1 describe('Prueba Unitaria', () => {
2   it('creacion de cliente', () => {
3     cy.on("uncaught:exception", (e, runnable) => {
4       console.log("error", e);
5       console.log("runnable", runnable);
6       return false;
7     });
8     cy.viewport(1536, 656)
9     cy.visit('http://localhost:8081/Infotec_definitivo-main/?c=Cliente')
10    cy.get('[type="int"]').type('10')
11    cy.get('[name="identificacion_cliente"]').type('1254876954')
12    cy.get('[name="nombre_cliente"]').type('Daniel')
13    cy.get('[name="apellido_cliente"]').type('Hernandez')
14    cy.get('[name="telefono_cliente"]').type('3145984561')
15    cy.get('[name="correo_cliente"]').type('danielhez@gmail.com')
16    cy.get('.row > .btn').click()
17    cy.get('.div').should('include.text', '10')
18  })
19 })

```

Infotec\_definitivo-main

localhost:8081/\_/#/specs/runner?file=cypress/e2e/1.PruebasUnitarias/crearCliente.cy.js

Un software de prueba automatizado está controlando Chrome.

Specs

crearCliente.cy.js (100%)

```

12 -type 3145984561
13 get [name="correo_cliente"]
14 -type danielhez@gmail.com
15 get .row > .btn
16 -click
17 (fore sub) --submitting form--
18 (page load) --page loaded--
19 (fetch) GET 200 https://ha-
fontawesome.com/releases/v4.4.0/css/free.min.
css?tohamag712951db
20 (fetch) GET 200 https://ha-
fontawesome.com/releases/v4.4.0/css/free-va-
shins.min.css?tohamag712951db
21 (fetch) GET 200 https://ha-
fontawesome.com/releases/v4.4.0/css/free-va-
shins.min.css?tohamag712951db
22 (fetch) GET 200 https://ha-
fontawesome.com/releases/v4.4.0/css/free-va-
shins.min.css?tohamag712951db
23 get .div
24 -expect expected <div.div.col-md-9> to
contain text '10'

```

http://localhost:8081/Infotec\_definitivo-main/?c=Cliente

Chrome 111

1536x656 (54%)

**Cientes**

Clientes

# Cliente	Identificación	Nombre	Apellido	Telefono	Correo	Modificar	Eliminar
10	612548769	Daniel	Hernandez	3145984561	danielhez@gmail.com		
3	2213214	Carla	Gomez	3146108761	carla@gmail.com		
4	123894561	Juan	Carro	3178453681	juan@gmail.com		
5	98546512	Andria	Martinez	3125988594	andria2@gmail.com		
6	1234567	ana	acordada	1232412	"@kacaj@gmail.com"		
7	123233871	Alex	Cardona	871985	alexcardona@gmail.com		

## - Creación de Usuario

Se desarrollo un código donde la herramienta de Cypress completara los datos del usuario y el mismo diera click en el botón enviar. Así mismo, se validó que el registro quedara guardado.

JS crearUsuario.cy.js

```

1 describe('Prueba Unitaria', () => {
2   it('creacion de cliente', () => {
3     cy.on("uncaught:exception", (e, runnable) => {
4       console.log("error", e);
5       console.log("runnable", runnable);
6       return false;
7     });
8     cy.viewport(1536, 656)
9     cy.visit('http://localhost:8081/Infotec_definitivo-main/?c=Usuario')
10    cy.get('select.form-control').select('Administrador')
11    cy.get('[type="int"]').type('78541269')
12    cy.get('[name="nombres_usuario"]').type('Juan')
13    cy.get('[name="apellidos_usuario"]').type('Martinez')
14    cy.get('[name="correo_usuario"]').type('juan.mar@hotmail.com')
15    cy.get('[name="telefono_usuario"]').type('3112785495')
16    cy.get('.row > [type="password"]').type('juan.321')
17    cy.get('.row > .btn').click()
18    cy.get('.div').should('include.text', 'Juan')
19  })
20 })

```

Infotec\_definitivo-main

localhost:8081/\_/#/specs/runner?file=cypress/e2e/1.PruebasUnitarias/crearUsuario.cy.js

Un software de prueba automatizado está controlando Chrome.

Specs

crearUsuario.cy.js

```

14 -type 3112785495
15 get .row > [type="password"]
16 -type Juan.321
17 get .row > .btn
18 -click
19 (form sub) --submitting form--
20 (page load) --page loaded--
(fetch) GET 200 https://ha-
f.fontawesome.com/releases/v6.4.0/css/free.min.
css?token=ae713951db
(fetch) GET 200 https://ha-
f.fontawesome.com/releases/v6.4.0/css/free-v4-
shims.min.css?token=ae713951db
(fetch) GET 200 https://ha-
f.fontawesome.com/releases/v6.4.0/css/free-v5-
font-face.min.css?token=ae713951db
(fetch) GET 200 https://ha-
f.fontawesome.com/releases/v6.4.0/css/free-v6-
font-face.min.css?token=ae713951db
19 get .div
20 Cypress expected <div>div.col-md-9 to
contain text Juan

```

http://localhost:8081/Infotec\_definitivo-main/?c=Usuario

Chrome 111

1536x656 (64%)

Usuario

Elige el rol

# Identificación

Nombre

Apellidos

Correo

Telefono

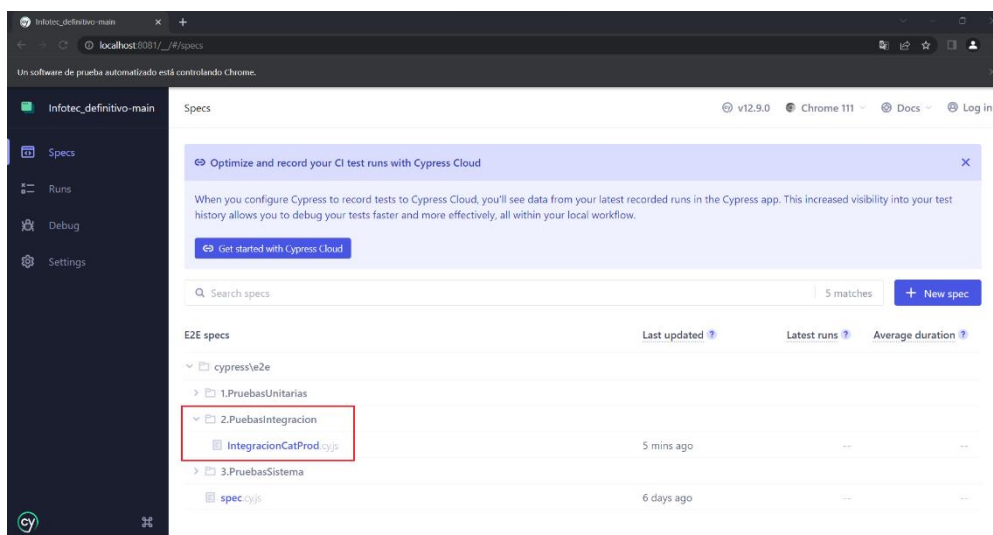
Contraseña

Enviar

Id	Identificación	Nombre	Apellidos	Correo	Telefono	Contraseña	Modificar	Eliminar
Administrador	1138749182	Carlos	Prados	carlupr@gmail.com	3138749182	salv	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Usuario	123307940	Ana	Morales	anapm@gmail.com	3128791873	aridazac	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Administrador	78541269	Juan	Martinez	juan.mar@hotmail.com	3112785495	juan.321	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Administrador	85421458	Patricia	Lara	patricigmail.com	3123639956	2541	<input checked="" type="checkbox"/>	<input type="checkbox"/>

### Prueba de integración

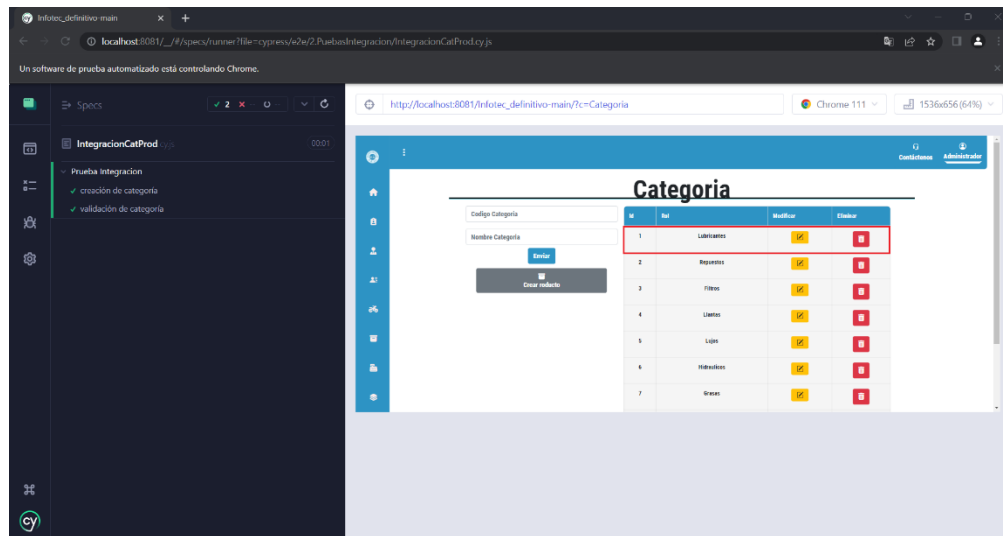
En la ejecución de estas pruebas se realizó un caso que se encontraban planificados anteriormente. En Cypress podrá encontrar la carpeta "2.PruebasIntegracion" donde se encuentran almacenada las prueba que se realizó.



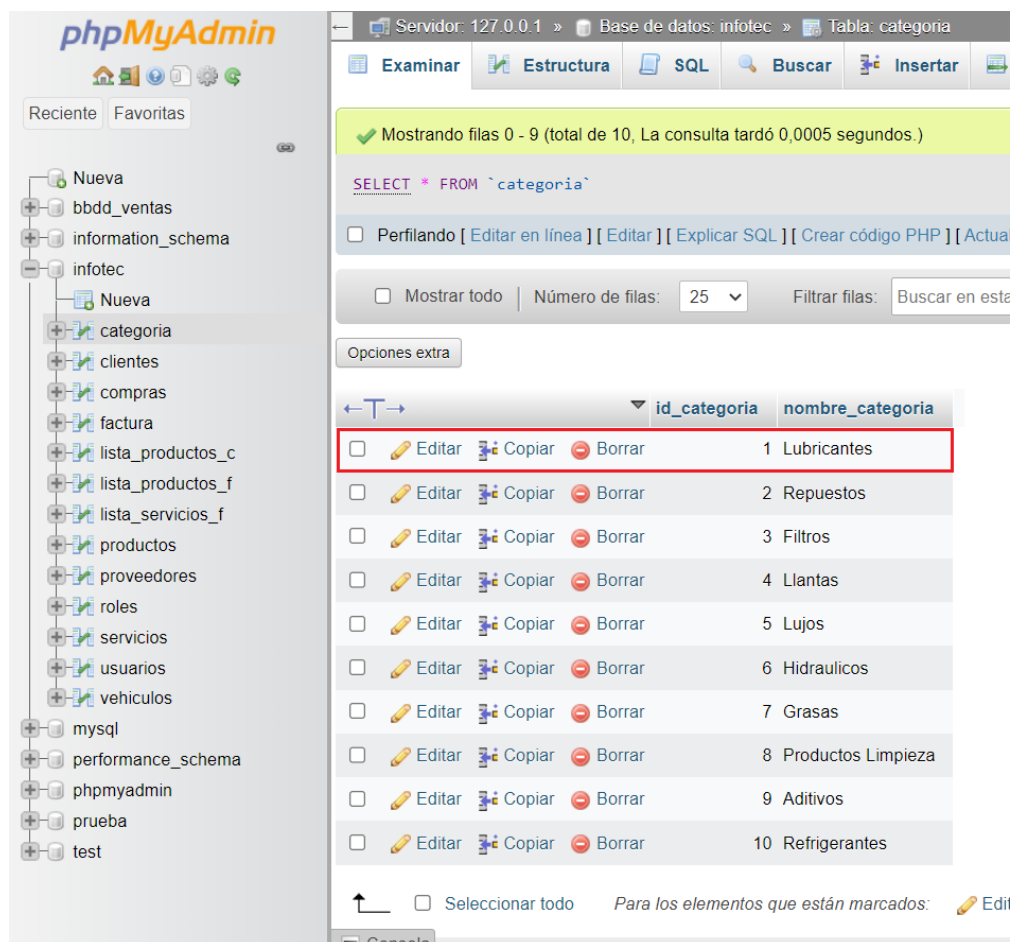
Para la ejecución de esta prueba se desarrolló un código donde la herramienta de Cypress completara los datos de la categoría y el mismo diera click en el botón enviar. Así mismo, se validó que el registro quedara guardado.

```
JS IntegracionCatProd.cy.js X
cypress > e2e > 2.PruebasIntegracion > JS IntegracionCatProd.cy.js > describe('Prueba Integracion') callback
1  describe('Prueba Integracion', () => {
2    it('creación de categoría', () => {
3      cy.on("uncaught:exception", (e, runnable) => {
4        console.log("error", e);
5        console.log("runnable", runnable);
6        return false;
7      });
8      cy.viewport(1536, 656)
9      cy.visit('http://localhost:8081/Infotec_definitivo-main/?c=Categoria')
10     cy.get('.mb-3').type('1')
11     cy.get('#validationCustom02').type('Lubricantes')
12     cy.get('.row > .btn').click()
13   })
14   it('validación de categoría', () => {
15     cy.on("uncaught:exception", (e, runnable) => {
16       console.log("error", e);
17       console.log("runnable", runnable);
18       return false;
19     });
20     cy.viewport(1536, 656)
21     cy.visit('http://localhost:8081/Infotec_definitivo-main/?c=Categoria')
22     cy.get('.div').should('include.text', 'Lubricantes')
23   })
}
```

## Plan Maestro de pruebas



Se puede evidenciar que el registro se almacena en la base de datos



Luego de la validar que la categoría fuera creada se procede a la creación del producto también por medio de Cypress y la validación de la creación de este.

JS IntegracionCatProd.cy.js X

```

cypress > e2e > 2.PuebasIntegración > JS IntegracionCatProd.cy.js > ...
22   cy.get('.div').should('include.text', 'Lubricantes')
23   })
24   it('creación de producto', () => {
25     cy.on("uncaught:exception", (e, runnable) => {
26       console.log("error", e);
27       console.log("runnable", runnable);
28       return false;
29     });
30     cy.viewport(1536, 656)
31     cy.visit('http://localhost:8081/Infotec_definitivo-main/?c=Categoria')
32     cy.get('.centarboton > .btn').click()
33     cy.get('select.form-control').select('Lubricantes')
34     cy.get('[name="id_producto"]').type('prod_3')
35     cy.get('[name="nombre_producto"]').type('Kit De Arrastre Casarella')
36     cy.get('[type="number"]').type('104500.00')
37     cy.get('[name="exist_producto"]').type('2')
38     cy.get('.row > .btn').click()
39   })
40   it('vañidación de producto', () => {
41     cy.on("uncaught:exception", (e, runnable) => {
42       console.log("error", e);
43       console.log("runnable", runnable);
44       return false;
45     });
46     cy.viewport(1536, 656)
47     cy.visit('http://localhost:8081/Infotec_definitivo-main/?c=Producto')
48     cy.get('.div').should('include.text', 'prod_3')
49   })
50 })

```

Infotec\_definitivo-main

localhost:8081/\_/#/specs?file=cypress/e2e/2.PuebasIntegracion/IntegracionCatProd.cy.js

Un software de prueba automatizado está controlando Chrome.

Specs

IntegracionCatProd

00:04

Prueba Integración

✓ creación de categoría

✓ validación de categoría

✓ creación de producto

✓ validación de producto

http://localhost:8081/Infotec\_definitivo-main/?c=Producto

Chrome 111

1536x656 (54%)

Producto

Elija la categoría

Producto

Nombre Producto

Precio

Existencia Producto

Botón

Crear

Cancelar

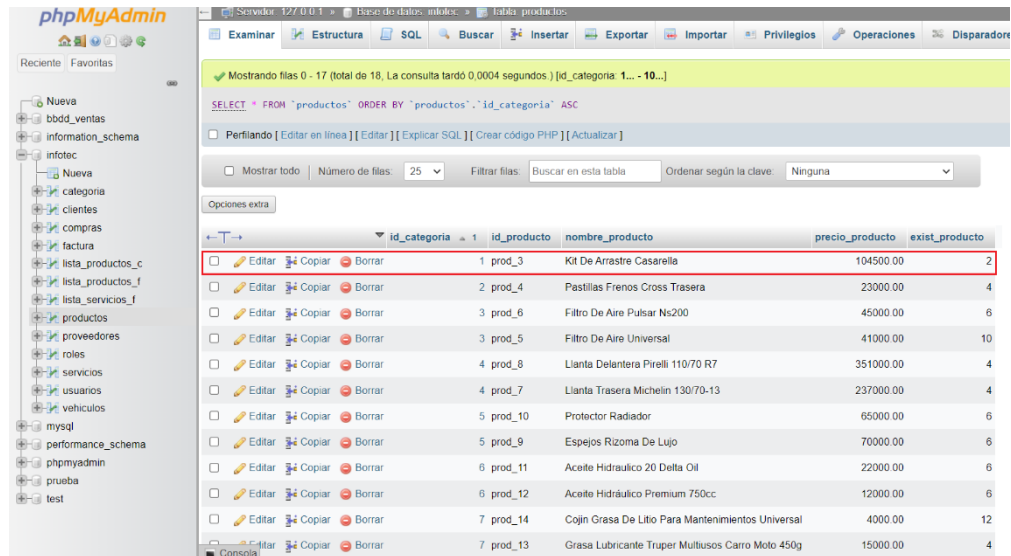
Proceder

Concepto	ID	Nombre	Precio	Existencia	Modificar	Eliminar
Lubricantes	prod_3	Kit De Arrastre Casarella	104500.00	2	✓	✗
Repuestos	prod_4	Fanalia Presso Duro Tronca	23000.00	4	✓	✗
Filtros	prod_5	Filtro De Aire Universal	40000.00	10	✓	✗
Filtros	prod_6	Filtro De Aire Pulver 8x200	40000.00	6	✓	✗
Llaves	prod_7	Llave Tronca Midecila 13x170-13	237000.00	4	✓	✗
Llaves	prod_8	Llave Delantera Pines 110x170-87	261000.00	4	✓	✗
Llaves	prod_10	Protector Radiador	60000.00	6	✓	✗

24



Se puede evidenciar que el registro se almacena en la base de datos



Mostrando filas 0 - 17 (total de 18, La consulta tardó 0.0004 segundos.) [id\_categoria: 1... - 10...]

```
SELECT * FROM `productos` ORDER BY `productos`.`id_categoria` ASC
```

Perfilando [ Editar en línea ] [ Editar ] [ Explicar SQL ] [ Crear código PHP ] [ Actualizar ]

Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla | Ordenar según la clave: Ninguna

		id_categoria	id_producto	nombre_producto	precio_producto	exist_producto
<input type="checkbox"/>	Editar	1	prod_3	Kit De Arrastre Casarella	104500.00	2
<input type="checkbox"/>	Editar	2	prod_4	Pastillas Frenos Cross Trasera	23000.00	4
<input type="checkbox"/>	Editar	3	prod_6	Filtro De Aire Pulsar Ns200	45000.00	6
<input type="checkbox"/>	Editar	3	prod_5	Filtro De Aire Universal	41000.00	10
<input type="checkbox"/>	Editar	4	prod_8	Llanta Delantera Pirelli 110/70 R7	351000.00	4
<input type="checkbox"/>	Editar	4	prod_7	Llanta Trasera Michelin 130/70-13	237000.00	4
<input type="checkbox"/>	Editar	5	prod_10	Protector Radiador	65000.00	6
<input type="checkbox"/>	Editar	5	prod_9	Espejos Rizoma De Lujo	70000.00	6
<input type="checkbox"/>	Editar	6	prod_11	Acetite Hidraulico 20 Delta Oil	22000.00	6
<input type="checkbox"/>	Editar	6	prod_12	Acetite Hidráulico Premium 750cc	12000.00	6
<input type="checkbox"/>	Editar	7	prod_14	Cojin Grasa De Lito Para Mantenimientos Universal	4000.00	12
<input type="checkbox"/>	Editar	7	prod_13	Grasa Lubricante Truper Multusos Carro Molo 450g	15000.00	4