



# PROGRAMACIÓN ORIENTADA A PRUEBAS

PROYECTO INTEGRADOR

[HTTPS://GITHUB.COM/KoRnelius-87/OrientadoAPruebas\\_MoranL.git](https://github.com/KoRnelius-87/OrientadoAPruebas_MoranL.git)

Luis Armando Moran Macias  
091DSVA

Uscanga Domínguez Luis Josué  
Docente

## Contenido

Introducción .....	2
Planteamiento del problema .....	2
Historias de usuario.....	3
Ejecución desde consola .....	7
Comando=> <b>npm run test-report</b> .....	7
Repositorio GitHub.....	8
Herramienta Mochawesome-report.....	8
Resultados de test Mochawesome-report.....	8

## Introducción

A continuación se planteara el proyecto final “Integrador” de la materia Programación Orientada a Pruebas, con la cual se pretende desarrollar estrategias de desarrollo de software que ayuden a las buenas practicas, realizando un análisis profundo de las un caso donde un cliente plantea sus necesidades para un nuevo sistema de control de ventas, inventario y empleados.

## Planteamiento del problema

Una farmacia necesita desarrollar un sistema que les permita controlar su inventario, ventas y empleados; en cada uno de estas necesidades se deben considerar temas como la seguridad, el control por diversos roles que se consideran manejar, como lo son los líderes de sucursales, empleados de piso y los administradores del sistema.

Quieren que los administradores puedan ingresar, modificar o en su caso eliminar usuarios del sistema, así como poder tener acceso a información de las diversas sucursales y a sus datos de rendimiento, así como inventarios. También deberían poder dar de alta nuevos productos y sucursales y tener control completo sobre ellas.

De parte los líderes de sucursal, deben poder visualizar los diversos productos y stock de ellos en la sucursal y modificar la existencia de ellos, así como el poder realizar ventas en el sistema. También podrán agregar nuevos empleados a la sucursal que administran y visualizar los datos de rendimiento de la sucursal.

Los empleados por su lado deben poder ver los productos y stock de la sucursal y hacer ventas de ellos.

El sistema debe considerar el acceso de los roles, y los límites que tiene cada uno, también es importante considerar que no se debe permitir realizar ventas de productos que requieren receta y si se quiere hacer una venta así, retornar un error sobre el producto para que así soliciten la receta al cliente y una vez tengan la receta poder enviar la petición nuevamente confirmando que poseen la receta, podrán hacer la confirmación directa, más debe ser manual.

También a los administradores les interesa poder ver que sucursales cuentan con cierto producto y así poder mover existencia entre las diversas sucursales, en especial entre las que son cercanas unas de otras.

## Historias de usuario

1. Como Usuario, necesito acceder al sistema por medio de un usuario y contraseña.
  - a. Si está registrado
  - b. Error cuando el nombre de usuario o la contraseña están vacíos
  - c. Error cuando el usuario es incorrecto
  - d. Error con Contraseña incorrecta
  - e. Error por Usuario sin parámetros suficientes
2. Como Sistema, necesito un control de usuarios por roles, para limitar las opciones a utilizar.
  - a. El rol usuario es Administrador
  - b. El rol usuario es Líder
  - c. El rol usuario es Empleado
  - d. Error cuando el rol de usuario no existe
3. Como Administrador, necesito poder dar de alta usuarios.
  - a. Usuario creado correctamente
  - b. Error si usuario esta duplicado
  - c. Error al intentar crear sin tener Rol Administrador
  - d. Error parámetros incorrectos.
  - e. Error por Usuario sin parámetros suficientes
4. Como Administrador, necesito poder modificar usuarios.
  - a. Usuario modificado correctamente
  - b. Valida parámetros correctos
  - c. Actualizar solo Nombre
  - d. Actualizar solo Correo

- e. Actualizar solo Edad
  - f. Error al querer actualizar Usuario inexistente
  - g. Error al utilizar usuario sin privilegios
5. Como Administrador, necesito poder eliminar usuarios del sistema.
- a. Usuario eliminado correctamente
  - b. Error al intentar eliminar sin tener rol Administrado
  - c. Error, Usuario no existe en el sistema
6. Como Administrador, necesito acceder a lista de Ventas por Sucursal.
- a. Recibe una lista en formato de arreglo
  - b. Error, al intentar obtener información sin ser Administrador
  - c. Error al colocar sucursal inexistente
  - d. Error al utilizar usuario sin privilegios
7. Como Administrador, necesito acceder a su Inventario por Sucursal.
- a. Obtener Inventario por Sucursal
  - b. Error al colocar sucursal inexistente
  - c. Error al utilizar usuario sin privilegios
8. Como Administrador, necesito poder dar de alta Productos.
- a. Producto creado correctamente
  - b. Error si producto esta duplicado
  - c. Error al intentar crear sin tener Rol Administrador
  - d. Error parámetros incorrectos (nombre, sucursal, receta)
  - e. Error por Producto sin parámetros suficientes
9. Como Administrador, necesito poder modificar Productos.
- a. Producto modificado correctamente
  - b. Valida parámetros correctos
  - c. Actualizar solo Nombre
  - d. Actualizar solo existencia
  - e. Actualizar solo sku
  - f. Actualizar necesidad de receta
  - g. Error al querer actualizar Producto inexistente
10. Como Como administrador, necesito eliminar productos del sistema

- a. Producto eliminado correctamente
  - b. Error al intentar eliminar sin tener rol administrador
  - c. Error, no existe el Producto en el sistema
11. Como Administrador, necesito poder dar de alta Sucursales.
- a. Sucursal creada correctamente
  - b. Error si Sucursal esta duplicada
  - c. Error al intentar crear sin tener Rol Administrador
  - d. Error parámetros incorrectos (nombre, dirección)
  - e. Error por Sucursal sin parámetros suficientes
12. Como Administrador, necesito poder modificar Sucursales del sistema.
- a. Sucursal modificada correctamente
  - b. Error al intentar modificar sin tener Rol de Administrador
  - c. Validación de parámetros
  - d. Valida que la sucursal exista en la lista
13. Como Administrador, necesito poder eliminar Sucursales.
- a. Sucursal eliminado correctamente
  - b. Error al intentar eliminar sin tener rol Administrado
  - c. Error, Sucursal no existe en el sistema
14. Como Líder, necesito acceder a lista de Producto de Sucursal asignada.
- a. Obtener lista de Producto por Sucursal asignada
  - b. Error al utilizar usuario sin privilegios
15. Como Líder, necesito poder modificar la Existencia de Productos en Sucursal asignada.
- a. Actualizar existencia de producto
  - b. Error al colocar Producto inexistente
  - c. Error al utilizar usuario sin privilegios
16. Como Líder, debería de poder realizar Venta de Productos.
- a. Venta realizada con éxito
  - b. Error por venta de producto inexistente
  - c. Error por venta con producto agotado
  - d. Error al utilizar usuario sin privilegios
17. Como Líder, necesita poder dar de alta Empleados a Sucursal asignada.

- a. Empleados creado correctamente
  - b. Error Creación de Empleados duplicado
  - c. Valida parámetros correctos
  - d. Error al utilizar usuario sin privilegios
  - e. Error en la creación de Empleados sin parámetro obligatorio\*
18. Como Líder, necesité acceder a lista de Ventas de Sucursal asignada.
- a. Obtener lista de Ventas por Sucursal
  - b. Error al utilizar usuario sin privilegios
19. Como Empleado, necesité acceder a lista de Producto de Sucursal asignada.
- a. Obtener lista de Producto por Sucursal asignada
  - b. Error al utilizar usuario sin privilegios
20. Como Empleado, debería de poder realizar Venta de Productos.
- a. Venta realizada con éxito
  - b. Error por venta de producto inexistente
  - c. Error por venta con producto agotado
  - d. Error al utilizar usuario sin privilegios
21. Como Empleado, necesité acceder a lista de Ventas de Sucursal asignada y realizadas por ellos.
- a. Obtener lista de Ventas por propias
  - b. Error al colocar sucursal inexistente
  - c. Error al utilizar usuario sin privilegios
22. Como vendedor, necesito que al querer vender un Producto que sea Solo con receta, muestre mensaje que no puede ser vendido sin receta para que sea solicitada.
23. como administrador, necesito listar stock de productos por sucursal más cercana.

## Ejecución desde consola

Comando=> **npm run test-report**

```
PS C:\Users\korn1\OneDrive\Escritorio\CESUN\C09_01_PROGRAMACIÓN ORIENTADA A PRUEBAS\tdd_Integrador> npm run test-report
```

```
> tdd_integrador@1.0.0 test-report
> mocha 'test' --reporter mochawesome --require mochawesome/register
```

1. Como usuario, necesito iniciar sesión en el sistema con un nombre de usuario y contraseña
  - ✓ a. Si esta registrado
  - ✓ b. Error cuando el nombre de usuario o la contraseña están vacíos
  - ✓ c. Error cuando el usuario es incorrecto
  - ✓ d. Error con Contraseña incorrecta
  - ✓ e. Error por Usuario sin parametros suficientes
2. Como Sistema, necesito un control de usuarios por roles, para limitar las opciones a utilizar.
  - ✓ a. El rol usuario es Administrador
  - ✓ b. El rol usuario es Lider
  - ✓ c. El rol usuario es Empleado
  - ✓ d. Error cuando el rol de usuario no existe
3. Como Administrador, necesito poder dar de alta usuarios
  - ✓ a. Usuario creado correctamente
  - ✓ b. Error si usuario esta duplicado
  - ✓ c. Error al intentar crear sin tener Rol Administrador
  - ✓ d. Error parámetros incorrectos.
  - ✓ e. Error por Usuario sin parametros suficientes

5. Como administrador, necesito eliminar usuarios del sistema
  - ✓ a. Usuario eliminado correctamente
  - ✓ b. Error al intentar eliminar sin tener Rol Administrador
  - ✓ c. Error, Usuario no existe en el sistema
6. Como Administrador, necesito acceder a lista de Ventas por Sucursal.
  - ✓ a. Recibe una lista en formato de arreglo
  - ✓ b. Error si producto esta duplicado
  - ✓ c. Error al intentar crear sin tener Rol Administrador
  - ✓ d. Error parámetros incorrectos (nombre, sucursal, receta)
  - ✓ e. Error por Usuario sin parametros suficientes
10. Como administrador, necesito eliminar productos del sistema
  - ✓ a. Producto eliminado correctamente
  - ✓ b. Error al intentar eliminar sin tener Rol Administrador
  - ✓ c. Error, no existe el Producto en el sistema
11. Como Administrador, necesito poder dar de alta sucursales
  - ✓ a. Sucursal creada correctamente
  - ✓ b. Error si Sucursal esta duplicada
  - ✓ c. Error al intentar crear sin tener Rol
  - ✓ d. Error parámetros incorrectos (nombre, dirección)
  - ✓ e. Error por Sucursal sin parámetros suficientes
12. Como Administrador, necesito poder modificar Sucursales del sistema.
  - ✓ a. Sucursal fue modificada correctamente
  - ✓ b. Error al intentar crear sin tener Rol Administrador
  - ✓ c. Valida parámetros correctos
  - ✓ d. Valida que la sucursal exista en la lista
13. Como administrador, necesito eliminar sucursales del sistema
  - ✓ a. Sucursal eliminada correctamente
  - ✓ b. Error al intentar eliminar sin tener Rol Administrador
  - ✓ c. Error, Sucursal no existe en el sistema

41 passing (80ms)

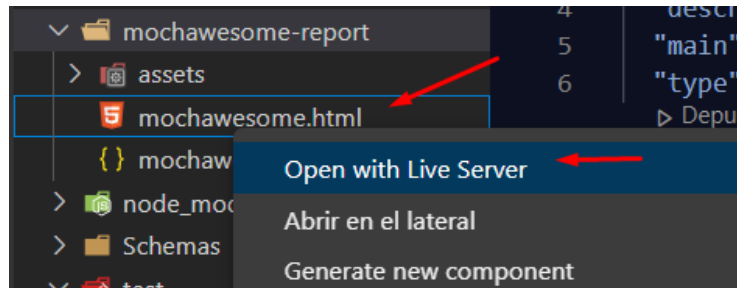


## Repositorio GitHub

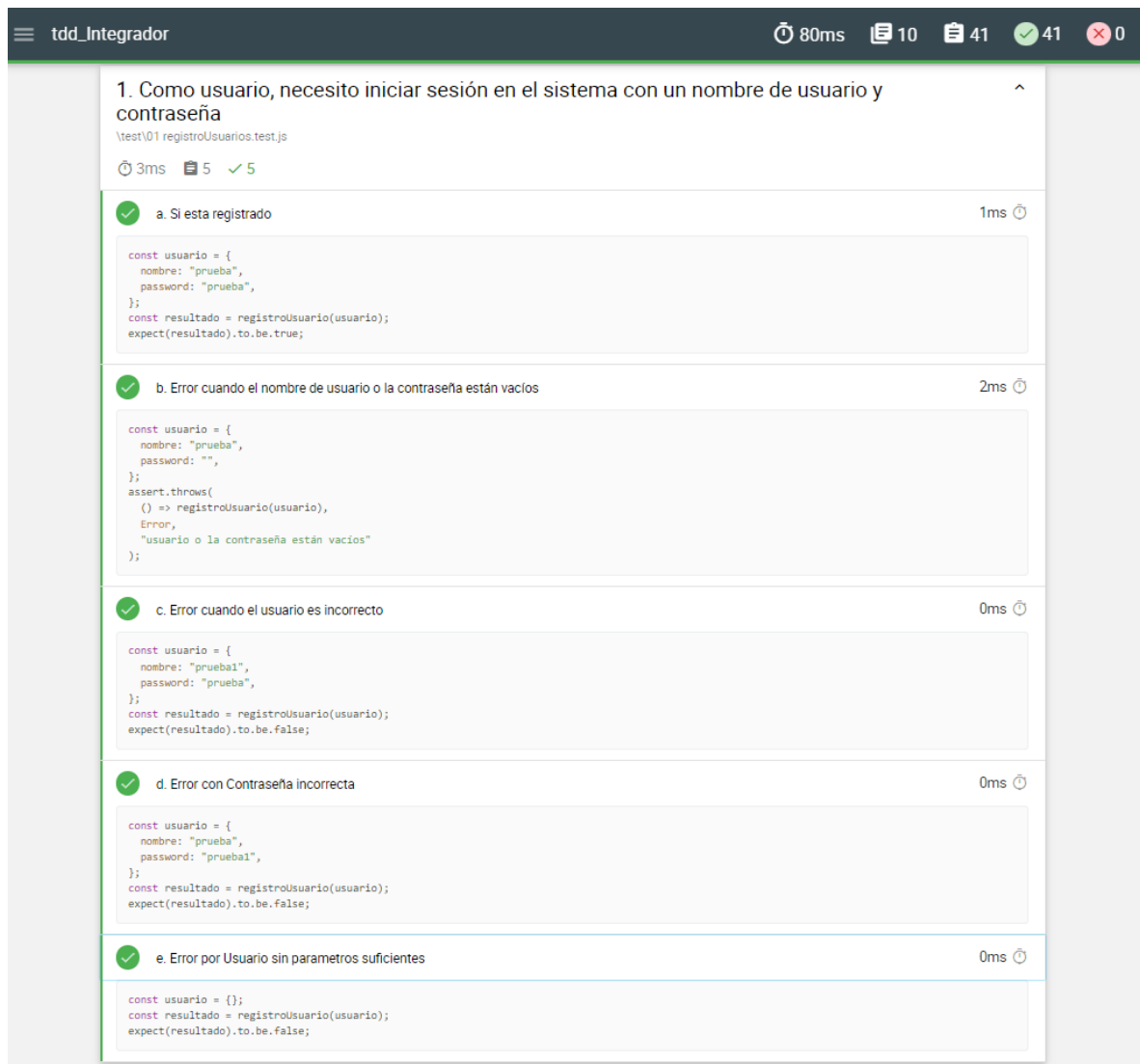
[HTTPS://GITHUB.COM/KoRnelius-87/OrientadoAPruebas\\_MORANL.GIT](https://github.com/KoRnelius-87/OrientadoAPruebas_MORANL.GIT)

## Herramienta Mochawesome-report

Se ejecuta archivo mochawesome-report/mochawesome.html con Live server.



## Resultados de test Mochawesome-report



## 2. Como Sistema, necesito un control de usuarios por roles, para limitar las opciones a utilizar.

\test\02 controlRoles.test.js

2ms 4 ✓ 4

### ✓ a. El rol usuario es Administrador

1ms

```
const usuario = administradorPrueba;
const resultado = controlRoles(usuario.rol);
expect(resultado).toEqual(0);
```

### ✓ b. El rol usuario es Lider

0ms

```
const usuario = liderPrueba;
const resultado = controlRoles(usuario.rol);
expect(resultado).toEqual(1);
```

### ✓ c. El rol usuario es Empleado

0ms

```
const usuario = empleadoPrueba;
const resultado = controlRoles(usuario.rol);
expect(resultado).toEqual(2);
```

### ✓ d. Error cuando el rol de usuario no existe

1ms

```
const usuario = undefined;
const resultado = controlRoles(usuario?.rol);
expect(resultado).toEqual(null);
```

## 3. Como Administrador, necesito poder dar de alta usuarios

\test\03 registroUsuariosAdministrador.test.js

3ms 5 ✓ 5

### ✓ a. Usuario creado correctamente

0ms

```
const usuario = administradorPrueba;
const nuevoUsuario = {
  nombre: "Nuevo Usuario",
  password: "new",
  email: "nuevousuario@gmail.com",
  rol: "empleado",
};
const resultado = adminCrearUsuario(usuario, nuevoUsuario);
expect(resultado).toBe(true);
```

### ✓ b. Error si usuario esta duplicado

1ms

```
const usuario = administradorPrueba;
const nuevoUsuario = {
  nombre: "prueba",
  password: "prueba",
};
assert.throws(
  () => adminCrearUsuario(usuario, nuevoUsuario),
  Error,
  "Error usuario duplicado"
);
```

### ✓ c. Error al intentar crear sin tener Rol Administrador

2ms

```
const usuario = empleadoPrueba;
const nuevoUsuario = {
  nombre: "Nuevo Usuario",
  password: "nuevousuario",
};
assert.throws(
  () => adminCrearUsuario(usuario, nuevoUsuario),
  Error,
  "Su rol no cuenta con el permiso necesario"
);
```

✓ d. Error parámetros incorrectos.

0ms ⓘ

```
const usuario = administradorPrueba;
const nuevoUsuario = {
  nombre: "",
  password: "",
  rol: "",
};
assert.throws(
  () => adminCrearUsuario(usuario, nuevoUsuario),
  Error,
  "Nombre de usuario o contraseña vacía"
);
```

✓ e. Error por Usuario sin parametros suficientes

0ms ⓘ

```
const usuario = administradorPrueba;
const nuevoUsuario = {};
assert.throws(
  () => adminCrearUsuario(usuario, nuevoUsuario),
  Error,
  "Faltan parametros"
);
```

## 5. Como administrador, necesito eliminar usuarios del sistema

\test\05 eliminarUsuariosAdministrador.test.js

⌚ 1ms 📄 3 ✓ 3

✓ a. Usuario eliminado correctamente

0ms ⓘ

```
const usuario = administradorPrueba;
const resultado = eliminarUsuarioAdministrador(usuario, usuarioEliminar);
expect(resultado).toEqual(true);
```

✓ b. Error al intentar eliminar sin tener Rol Administrador

0ms ⓘ

```
const usuario = liderPrueba;
assert.throws(
  () => eliminarUsuarioAdministrador(usuario, usuarioEliminar),
  Error,
  "Su rol no cuenta con el permiso necesario"
);
```

✓ c. Error, Usuario no existe en el sistema

1ms ⓘ

```
const usuario = administradorPrueba;
const usuarioAEliminar = {
  id: null,
  nombre: "prueba",
  password: "prueba",
  rol: "empleado",
};
assert.throws(
  () => eliminarUsuarioAdministrador(usuario, usuarioAEliminar),
  Error,
  "Usuario no existe"
);
```

## 6. Como Administrador, necesito acceder a lista de Ventas por Sucursal.

\test\06 listaVentasSucursalAdministrador.test.js

🕒 2ms 📄 4 ✓ 4

### ✓ a. Recibe una lista en formato de arreglo

1ms 🕒

```
const user = administradorPrueba;  
const sucursal = "Sucursal prueba3";  
const resultado = listaVentasporSucursalAdministrador(user, sucursal);  
expect(resultado).to.be.an("array");
```

### ✓ b. Error, al intentar obtener información sin ser Administrador

0ms 🕒

```
const usuario = liderPrueba;  
const sucursal = "Sucursal prueba3";  
assert.throws(  
  () => listaVentasporSucursalAdministrador(usuario, "Branch1"),  
  Error,  
  "Su rol no cuenta con el permiso necesarios"  
);
```

### ✓ Error, si el parametro sucursal esta vacio

0ms 🕒

```
const usuario = administradorPrueba;  
assert.throws(  
  () => listaVentasporSucursalAdministrador(usuario, ""),  
  Error,  
  "Falta nombre de Sucursal"  
);
```

### ✓ Error, si sucursal no existe

1ms 🕒

```
const usuario = administradorPrueba;  
assert.throws(  
  () => listaVentasporSucursalAdministrador(usuario, "Branch5"),  
  Error,  
  "Sucursal no existe"  
);
```

## 8. Como Administrador, necesito poder dar de alta usuarios

\test\08 registroProductosAdministrador.test.js

🕒 1ms 📄 5 ✓ 5

### ✓ a. Producto creado correctamente

0ms 🕒

```
const usuario = administradorPrueba;  
const nuevoProducto = {  
  nombre: "Producto nuevo",  
  cantidad: 99,  
  receta: true,  
  sucursal: "sucursal nueva",  
};  
const resultado = adminCrearProducto(usuario, nuevoProducto);  
expect(resultado).to.be.true;
```

### ✓ b. Error si producto esta duplicado

0ms 🕒

```
const usuario = administradorPrueba;  
const nuevoProducto = {  
  nombre: "Producto prueba",  
  receta: true,  
  sucursal: "sucursalPrueba",  
};  
assert.throws(  
  () => adminCrearProducto(usuario, nuevoProducto),  
  Error,  
  "Error producto duplicado"  
);
```

### ✓ c. Error al intentar crear sin tener Rol Administrador

0ms 🕒

```
const usuario = empleadoPrueba;  
const nuevoProducto = {  
  nombre: "Producto nuevo",  
  cantidad: 99,  
  sucursal: "sucursal nueva",  
};  
assert.throws(  
  () => adminCrearProducto(usuario, nuevoProducto),  
  Error,  
  "Su rol no cuenta con el permiso necesario"  
);
```

✓ d. Error parámetros incorrectos (nombre, sucursal, receta)

0ms

```
const usuario = administradorPrueba;
const nuevoProducto = {
  nombre: "",
  cantidad: "",
  sucursal: "",
  receta: "",
};
assert.throws(
  () => adminCrearProducto(usuario, nuevoProducto),
  Error,
  "Nombre de producto, sucursal, estado de receta vacia"
);
```

✓ e. Error por Usuario sin parametros suficientes

1ms

```
const usuario = administradorPrueba;
const nuevoProducto = {};
assert.throws(
  () => adminCrearProducto(usuario, nuevoProducto),
  Error,
  "Faltan parametros"
);
```

## 10. Como administrador, necesito eliminar productos del sistema

\test\10 eliminarProductoAdministrador.test.js

🕒 1ms 📄 3 ✓ 3

✓ a. Producto eliminado correctamente

1ms

```
const usuario = administradorPrueba;
const resultado = eliminarProductoAdministrador(usuario, productoPrueba);
expect(resultado).toEqual(true);
```

✓ b. Error al intentar eliminar sin tener Rol Administrador

0ms

```
const usuario = liderPrueba;
assert.throws(
  () => eliminarProductoAdministrador(usuario, productoPrueba),
  Error,
  "Su rol no cuenta con el permiso necesario"
);
```

✓ c. Error, no existe el Producto en el sistema

0ms

```
const usuario = administradorPrueba;
const producto = {
  id: null,
  nombre: "prueba",
  precio: 100,
  cantidad: 10,
  sucursal: "sucursal nueva",
};
assert.throws(
  () => eliminarProductoAdministrador(usuario, producto),
  Error,
  "Producto no existe"
);
```

## 11. Como Administrador, necesito poder dar de alta sucursales

\test\11 registroSucursalAdministrador.test.js

🕒 5ms 📄 5 ✅ 5

### ✅ a. Sucursal creada correctamente

1ms 🕒

```
const usuario = administradorPrueba;
const nuevaSucursal = {
  nombre: "Sucursal nueva",
  direccion: "Direccion Sucursal nueva",
};
const resultado = adminCrearSucursal(usuario, nuevaSucursal);
expect(resultado).toBe(true);
```

### ✅ b. Error si Sucursal esta duplicada

1ms 🕒

```
const usuario = administradorPrueba;
const nuevaSucursal = {
  nombre: "Sucursal prueba",
  direccion: "Direccion Sucursal prueba",
};
assert.throws(
  () => adminCrearSucursal(usuario, nuevaSucursal),
  Error,
  "Error sucursal duplicada"
);
```

### ✅ c. Error al intentar crear sin tener Rol

1ms 🕒

```
const usuario = empleadoPrueba;
const nuevaSucursal = {
  nombre: "Sucursal nueva",
  direccion: "Direccion Sucursal nueva",
};
assert.throws(
  () => adminCrearSucursal(usuario, nuevaSucursal),
  Error,
  "Su rol no cuenta con el permiso necesario"
);
```

### ✅ d. Error parámetros incorrectos (nombre, dirección)

1ms 🕒

```
const usuario = administradorPrueba;
const nuevaSucursal = {
  nombre: "",
  direccion: "",
};
assert.throws(
  () => adminCrearSucursal(usuario, nuevaSucursal),
  Error,
  "Nombre de sucursal o direccion estan vacia"
);
```

### ✅ e. Error por Sucursal sin parámetros suficientes

1ms 🕒

```
const usuario = administradorPrueba;
const nuevaSucursal = {};
assert.throws(
  () => adminCrearSucursal(usuario, nuevaSucursal),
  Error,
  "Faltan parametros"
);
```

## 12. Como Administrador, necesito poder modificar Sucursales del sistema.

\test\12 actualizaSucursalesAdministrador.test.js

🕒 4ms 📄 4 ✓ 4

### ✓ a. Sucursal fue modificada correctamente

1ms 🕒

```
const usuario = administradorPrueba;
const resultado = actualizarSucursalAdministrador(usuario, sucursalPrueba);
return expect(resultado).toBe(true);
```

### ✓ b. Error al intentar crear sin tener Rol Administrador

1ms 🕒

```
const usuario = liderPrueba;
assert.throws(
  () => actualizarSucursalAdministrador(usuario, sucursalPrueba),
  Error,
  "Su rol no cuenta con el permiso necesario"
);
```

### ✓ c. Valida parámetros correctos

1ms 🕒

```
const usuario = administradorPrueba;
const sucursal = {
  nombre: "",
  direccion: "",
};
assert.throws(
  () => actualizarSucursalAdministrador(usuario, sucursal),
  Error,
  "Nombre de Sucursal o direccion vacios"
);
```

### ✓ d. Valida que la sucursal exista en la lista

1ms 🕒

```
const usuario = administradorPrueba;
const sucursal = {
  nombre: "Sucursal prueba1",
  direccion: "Direccion Sucursal prueba1",
};
assert.throws(
  () => actualizarSucursalAdministrador(usuario, sucursal),
  Error,
  "Sucursal ya existe"
);
```

## 13. Como administrador, necesito eliminar sucursales del sistema

\test\13 eliminarSucursalAdministrador.test.js

🕒 1ms 📄 3 ✓ 3

### ✓ a. Sucursal eliminada correctamente

0ms 🕒

```
const usuario = administradorPrueba;
const resultado = eliminarSucursalAdministrador(usuario, sucursalPrueba);
expect(resultado).toEqual(true);
```

### ✓ b. Error al intentar eliminar sin tener Rol Administrador

0ms 🕒

```
const usuario = liderPrueba;
assert.throws(
  () => eliminarSucursalAdministrador(usuario, sucursalPrueba),
  Error,
  "Su rol no cuenta con el permiso necesario"
);
```

### ✓ c. Error, Sucursal no existe en el sistema

1ms 🕒

```
const usuario = administradorPrueba;
const sucursal = {
  id: null,
  nombre: "Sucursal prueba",
  direccion: "Direccion Sucursal prueba",
};
assert.throws(
  () => eliminarSucursalAdministrador(usuario, sucursal),
  Error,
  "Sucursal no existe"
);
```

## Conclusión

Se comprendió una manera diferente pero muy efectiva para el desarrollo de un sistema, como es con el método TDD, en el cual se piensa más en la lógica de los problemas, desarrollando las soluciones sin iniciar en el producto final, de esta manera se logran aclarar y comprender mejor el sistema, y de esta manera lograr un producto de calidad.