

UNIVERSIDAD DE LAS FUERZAS ARMADAS “ESPE”

METODOLOGÍA DE DESARROLLO DE SOFTWARE

INTEGRANTES: ACALO CRISTIAN, CARRERA NAHIR, GARCIA JONATHAN,
OBANDO LEONARDO

GRUPO 1

PRUEBA DE CAJA BLANCA
ENTRE HILOS & ALGODÓN

VERSIÓN 4

FECHA: 03/03/2024

ING. JENNY RUIZ

LOGIN

PASO 1. CÓDIGO

Validación pin, acceso empleado/administrador al sistema

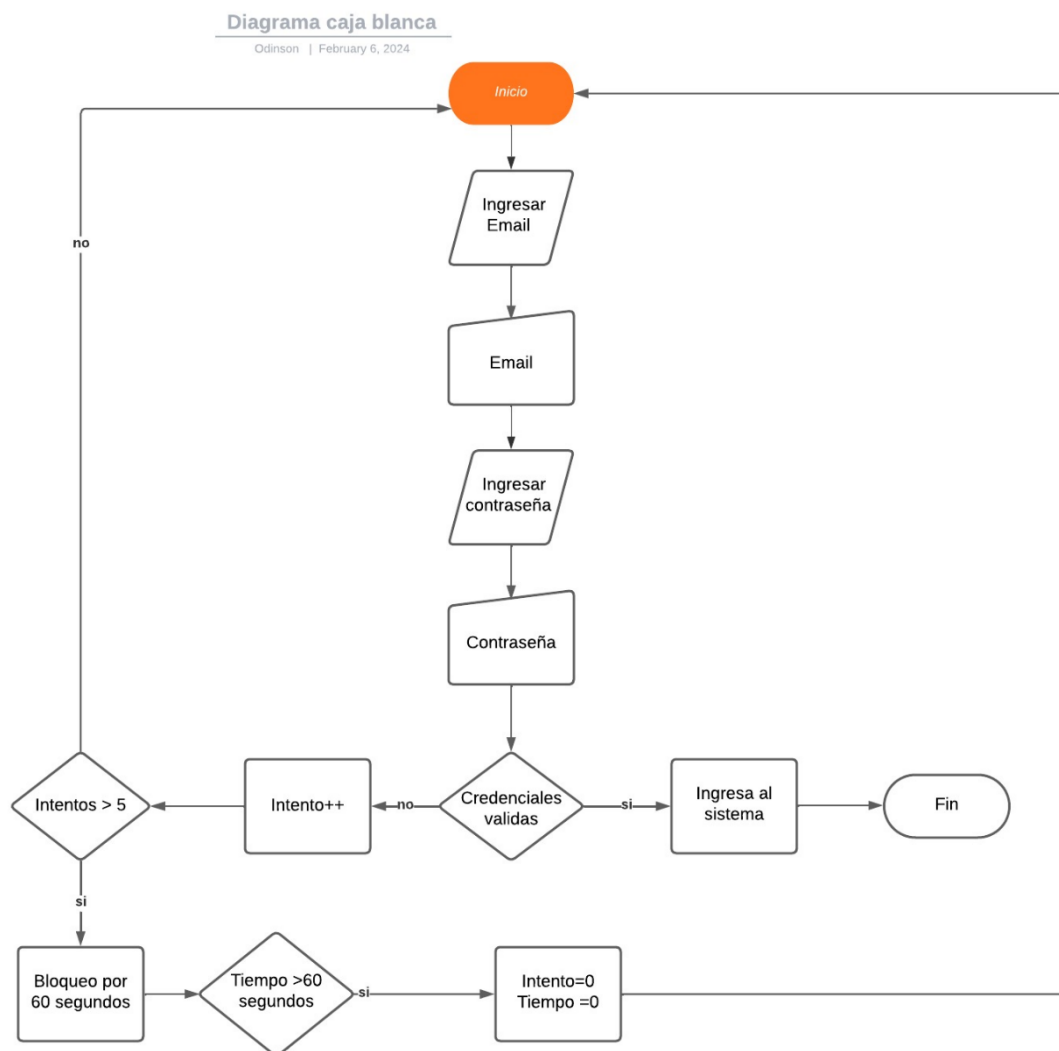
```
public function authenticate(): void
{
    $this->ensureIsNotRateLimited();

    if (!Auth::attempt($this->only('email', 'password'), $this->boolean('remember'))) {
        RateLimiter::hit($this->throttleKey());

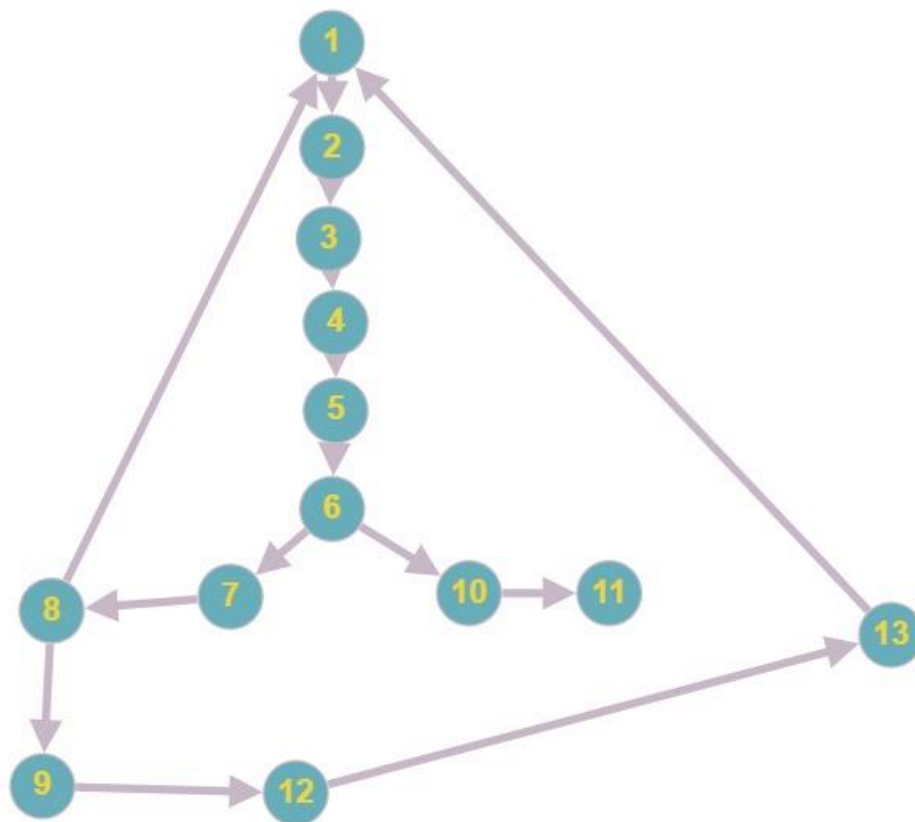
        throw ValidationException::withMessages([
            'email' => trans('auth.failed'),
        ]);
    }

    RateLimiter::clear($this->throttleKey());
}
```

PASO 2. DIAGRAMA DE FLUJO



PASO 3. GRAFO



PASO 4.

Nodos (N): 13

Aristas (E): 14

Nodo Predicado/Condición /Decisión(P): 2

Rutas:

R1=1,2,3,4,5,6,7,8,1

R2=1,2,3,4,5,6,7,10,11

R3=1,2,3,4,5,6,7,8, 9, 12, 13,1

COMPLEJIDAD CICLOMÁTICA

$V(G)=DECISIONES+1$

$V(G)=2+1$

$V(G)=3$

$V(G)= E-N+2$

$V(G)=14-13+2$

$V(G)=3$

ACTUALIZACIÓN DE DATOS

PASO 1. CÓDIGO

Validación y actualización de datos en el sistema.

```
1 reference | 0 overrides
public function update(ProfileUpdateRequest $request): RedirectResponse
{
    $request->user()->fill($request->validated());

    if ($request->user()->isDirty('email')) {
        $request->user()->email_verified_at = null;
    }

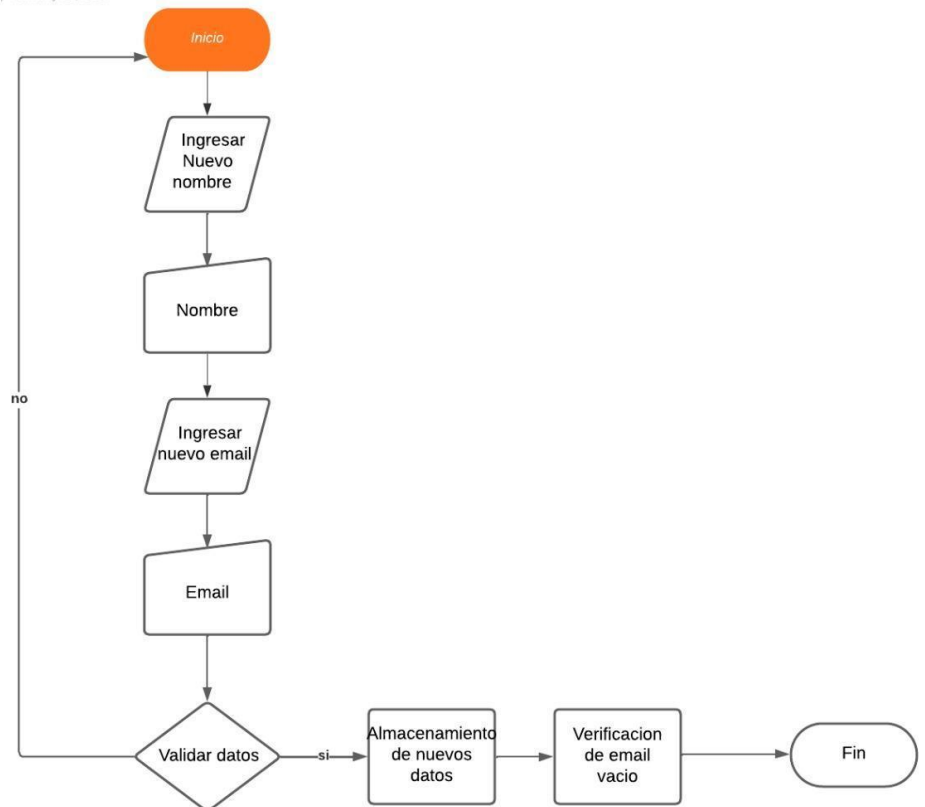
    $request->user()->save();

    return Redirect::route('profile.edit')->with('status', 'profile-updated');
}
```

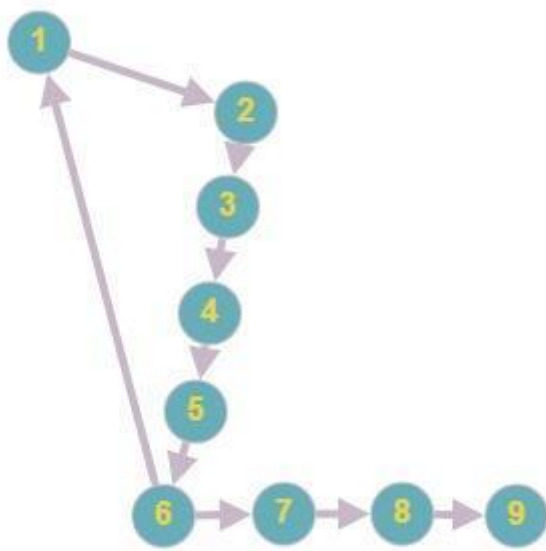
PASO 2. DIAGRAMA DE FLUJO

Diagrama caja blanca

Odinson | February 15, 2024



PASO 3. GRAFO



PASO 4.

Nodos (N): 9

Aristas (E): 9

Nodo Predicado/Condición /Decisión(P): 1

Rutas:

R1=1,2,3,4,5,6,7,8,9

R2=1,2,3,4,5,6,1

COMPLEJIDAD CICLOMÁTICA

$V(G) = \text{DECISIONES} + 1$

$V(G) = 1 + 1$

$V(G) = 2$

$V(G) = E - N + 2$

$V(G) = 9 - 9 + 2$

$V(G) = 2$

Actualización de Contraseña

PASO 1. CÓDIGO

Validación y actualización de la contraseña en el sistema.

```
1 reference | 0 overrides
public function update(Request $request): RedirectResponse
{
    $validated = $request->validateWithBag('updatePassword', [
        'current_password' => ['required', 'current_password'],
        'password' => ['required', Password::defaults(), 'confirmed'],
    ]);

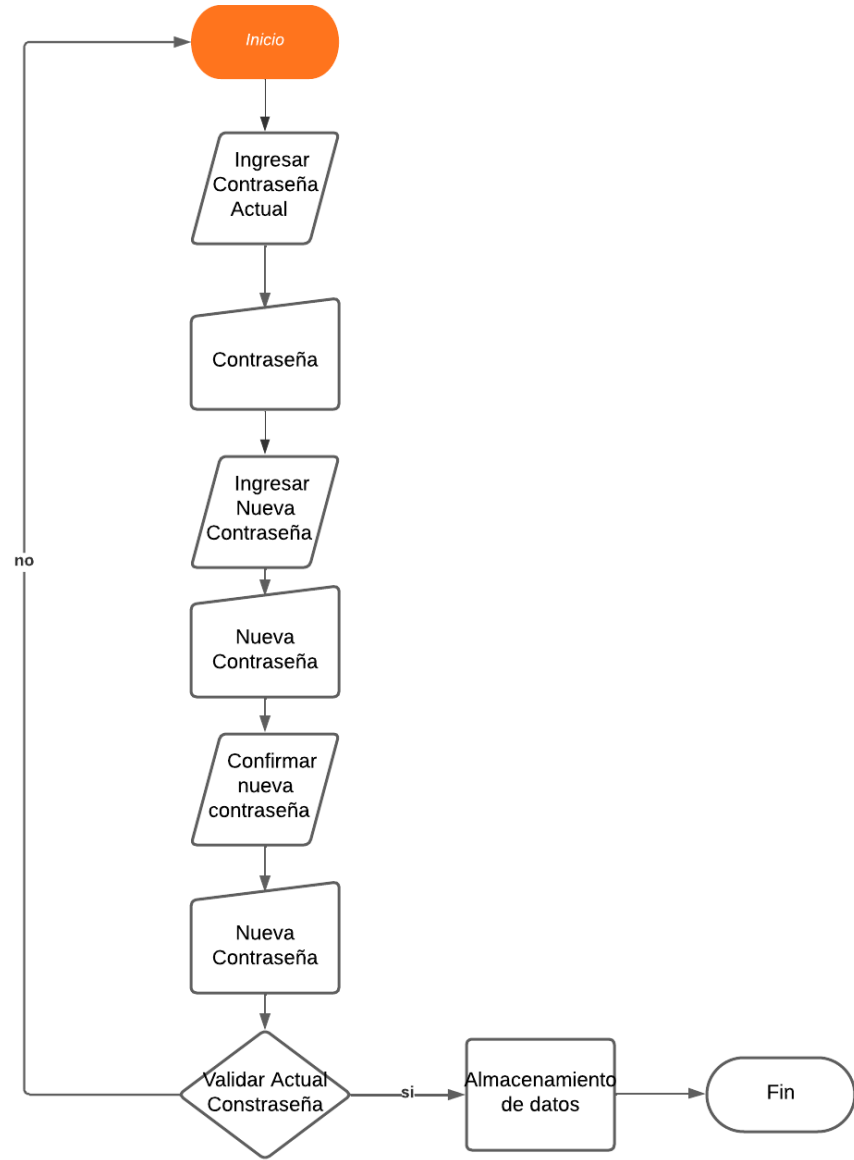
    $request->user()->update([
        'password' => Hash::make($validated['password']),
    ]);

    return back()->with('status', 'password-updated');
}
```

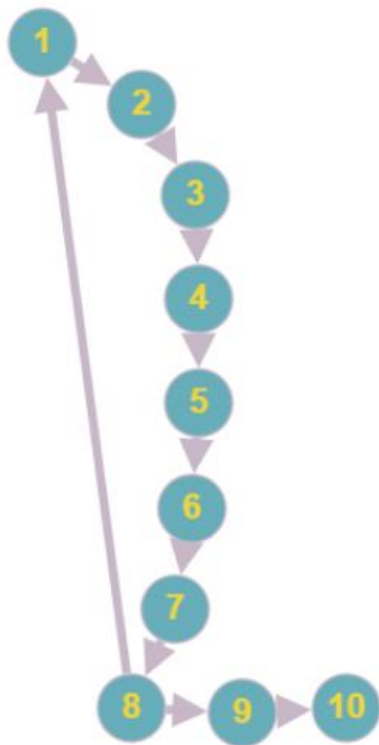
PASO 2. DIAGRAMA DE FLUJO.

Diagrama caja blanca

Odinson | February 16, 2024



PASO 3. GRAFO



PASO 4.

Nodos (N): 10

Aristas (E): 10

Nodo Predicado/Condición /Decisión(P): 1

Rutas:

R1=1,2,3,4,5,6,7,8,1

R2=1,2,3,4,5,6,7,8,9,10

COMPLEJIDAD CICLOMÁTICA

$V(G)=DECISIONES+1$

$V(G)=1+1$

$V(G)=2$

$V(G)= E-N+2$

$V(G)=10-10+2$

$V(G)=2$

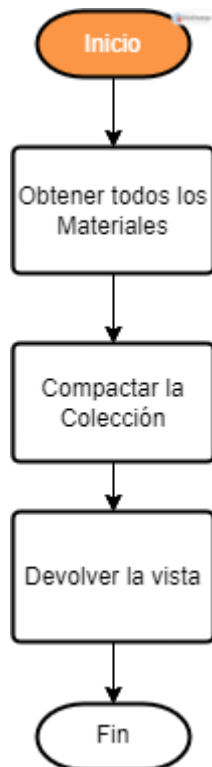
Ver lista de materiales

PASO 1. CÓDIGO

Ver Materiales

```
0 references | 0 overrides
public function index()
{
    $materiales = Materiales::all();
    return view('admin.materiales.materialesManager', compact('materiales'));
}
```

PASO 2. DIAGRAMA DE FLUJO.



PASO 3. GRAFO



PASO 4.

Nodos (N): 5

Aristas (E): 4

Nodo Predicado/Condición /Decisión(P): 0

Rutas:

R1=1,2,3,4,5

COMPLEJIDAD CICLOMÁTICA

$V(G) = \text{DECISIONES} + 1$

$V(G) = 0 + 1$

$V(G) = 1$

$V(G) = E - N + 2$

$V(G) = 4 - 5 + 2$

$V(G) = 1$

Creación del material

PASO 1. CÓDIGO

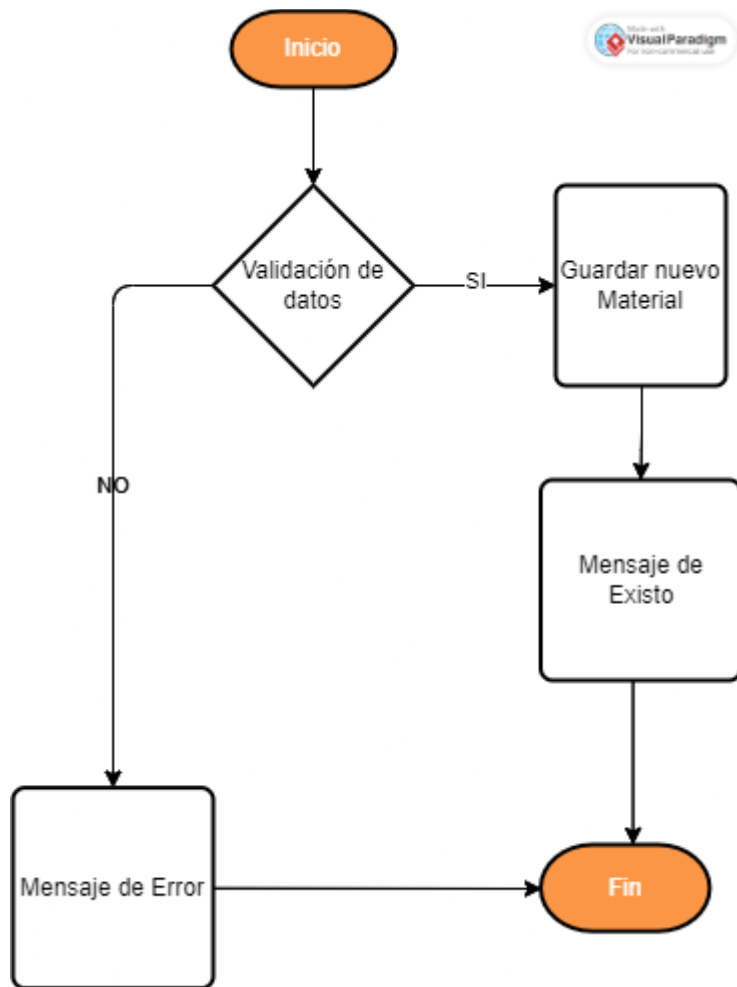
Creación de un nuevo material en el sistema.

```
0 references | 0 overrides
public function create(Request $newMaterialData)
{
    $validatedData = $newMaterialData->validate([
        'nombre' => 'required|unique:materiales|max:255',
        'ud_medida' => 'required',
        'cantidad' => 'required|numeric|min:1',
        'costo_total' => 'required|numeric'
    ]);

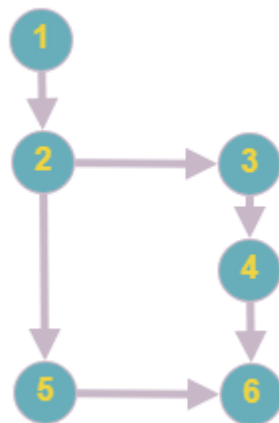
    $newMaterial = new Materiales();
    try {
        $newMaterial->nombre = $newMaterialData->nombre;
        $newMaterial->ud_medida = $newMaterialData->ud_medida;
        $newMaterial->cantidad = $newMaterialData->cantidad;
        $newMaterial->costo_total = $newMaterialData->costo_total;
        $newMaterial->costo_ud_medida = $newMaterialData->costo_total / $newMaterialData->cantidad;
        $newMaterial->save();

        return back()->with(['success' => 'El material se registró con éxito']);
    } catch (Exception $e) {
        return back()->with(['danger' => $e->getMessage()]);
        return $e->getMessage();
    }
}
```

PASO 2. DIAGRAMA DE FLUJO.



PASO 3. GRAFO



PASO 4.

Nodos (N): 6

Aristas (E): 6

Nodo Predicado/Condición /Decisión(P): 1

Rutas:

R1=1,2,3,4,6

R2=1,2,5,6

COMPLEJIDAD CICLOMÁTICA

V(G)=DECISIONES+1

V(G)=1+1

V(G)=2

V(G)= E-N+2

V(G)=6-6+2

V(G)=2

Edición del material

PASO 1. CÓDIGO

Edición del material con validaciones en el sistema.

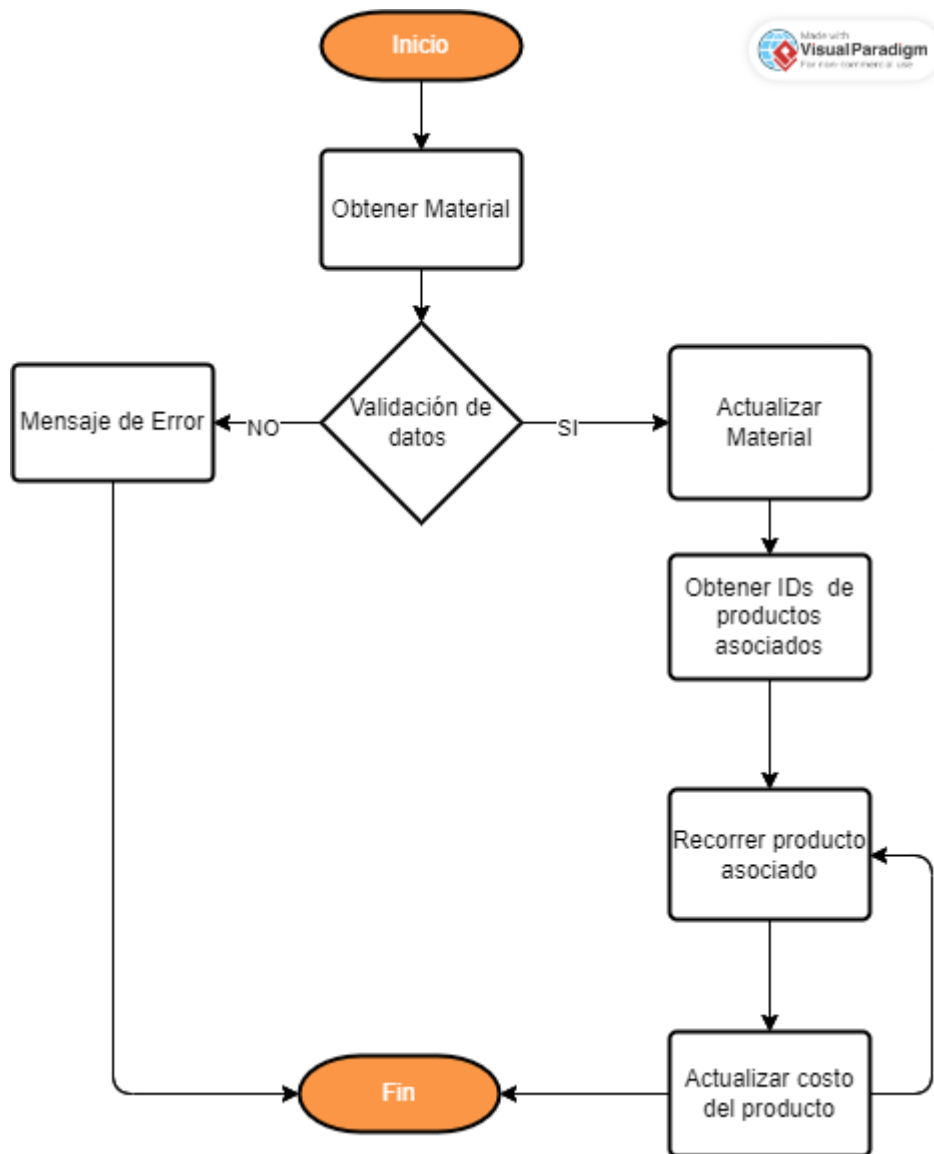
```
0 references | 0 overrides
public function edit(Request $newData, $materialID)
{
    $material = Materiales::where('id', decrypt($materialID))->first();
    if ($material->nombre != $newData->nombre) {
        $validatedData = $newData->validate([
            'nombre' => 'required|unique:materiales|max:255',
        ]);
    }
    $validatedData = $newData->validate([
        'ud_medida' => 'required',
        'cantidad' => 'required|numeric|min:1',
        'costo_total' => 'required|numeric'
    ]);

    try {
        $material->nombre = $newData->nombre;
        $material->ud_medida = $newData->ud_medida;
        $material->costo_ud_medida = $newData->costo_por_ud_medida;
        $material->save();

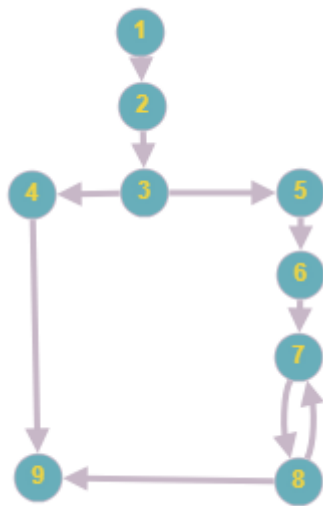
        $idsProductosAsignados = MaterialesPorProducto::where('material_id', decrypt($materialID))->pluck('producto_id')->unique()->toArray();
        foreach ($idsProductosAsignados as $productId) {
            $this->calculateCost($productId);
        }

        return back()->with(['success' => 'El material se actualizó con éxito']);
    } catch (Exception $e) {
        return back()->with(['danger' => $e->getMessage()]);
    }
    return $e->getMessage();
}
```

PASO 2. DIAGRAMA DE FLUJO.



PASO 3. GRAFO



PASO 4.

Nodos (N): 9

Aristas (E): 10

Nodo Predicado/Condición /Decisión(P): 2

Rutas:

R1=1,2,3,4,9

R2=1,2,3,5,6,7,8,9

R3=1,2,3,5,6,7,8,7,8,9

COMPLEJIDAD CICLOMÁTICA

$V(G)=DECISIONES+1$

$V(G)=2+1$

$V(G)=3$

$V(G)= E-N+2$

$V(G)=10-9+2$

$V(G)=3$

Eliminación del material

PASO 1. CÓDIGO

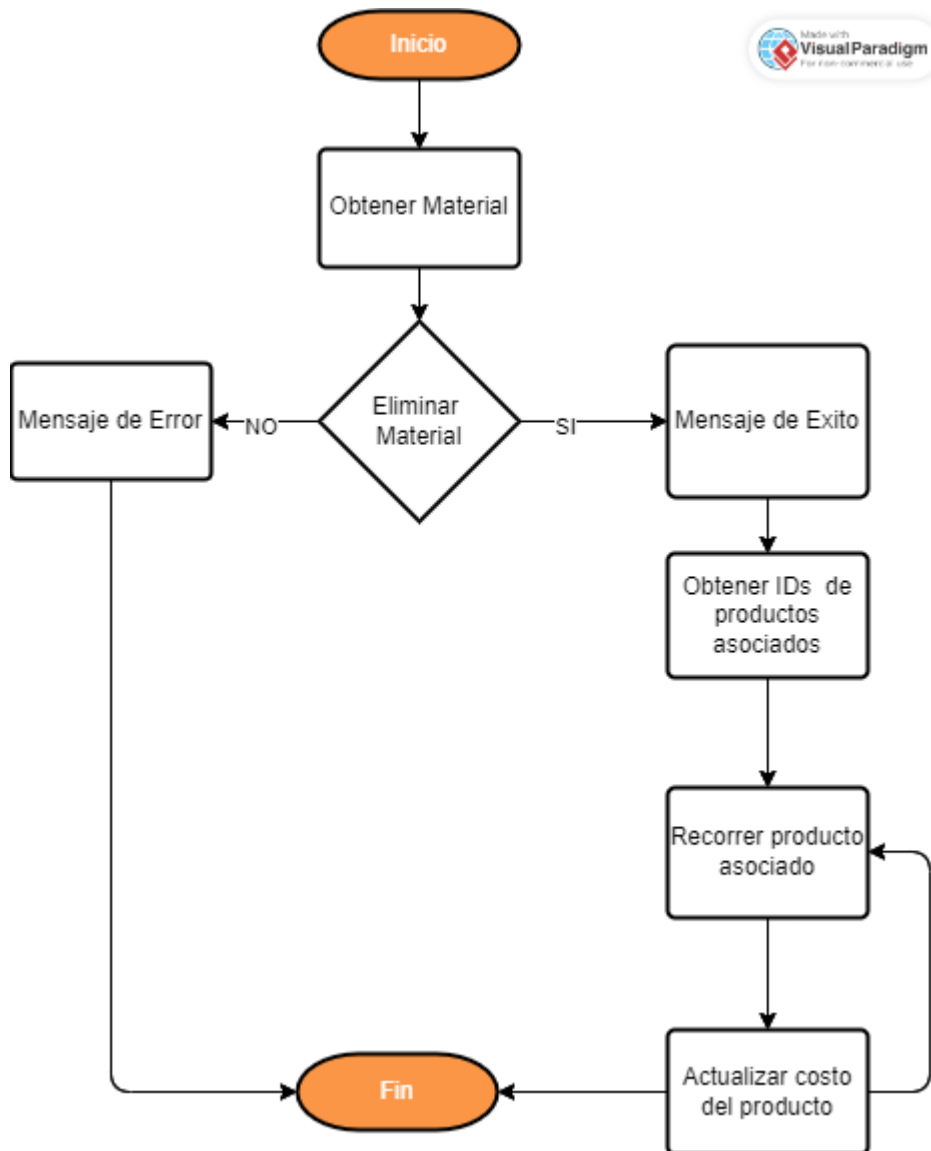
Eliminación del material en el sistema, validación de material.

```
0 references | 0 overrides
public function destroy($materialID)
{
    try {
        $material = Materiales::where('id', decrypt($materialID))>first();
        $material->delete();

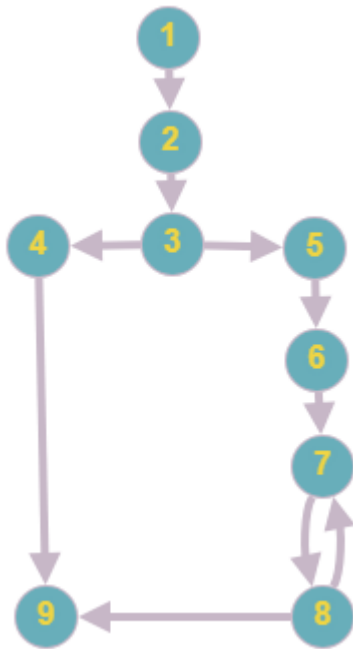
        $idsProductosAsignados = MaterialesPorProducto::where('material_id', decrypt($materialID))>pluck('producto_id')>unique()->toArray();
        foreach ($idsProductosAsignados as $productId) {
            $this->calculateCost($productId);
        }

        return back()->with(['success' => 'El material se eliminó con éxito']);
    } catch (Exception $e) {
        return $e->getMessage();
    }
}
```

PASO 2. DIAGRAMA DE FLUJO.



PASO 3. GRAFO



PASO 4.

Nodos (N): 9

Aristas (E): 10

Nodo Predicado/Condición /Decisión(P): 2

Rutas:

R1=1,2,3,4,9

R2=1,2,3,5,6,7,8,9

R3=1,2,3,5,6,7,8,7,8,9

COMPLEJIDAD CICLOMÁTICA

$V(G) = \text{DECISIONES} + 1$

$V(G) = 2 + 1$

$V(G) = 3$

$V(G) = E - N + 2$

$V(G) = 10 - 9 + 2$

$V(G) = 3$

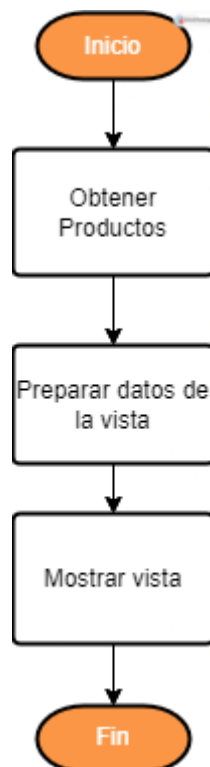
Ver la lista de productos

PASO 1. CÓDIGO

Ver productos

```
0 references | 0 overrides  
public function index()  
{  
    $productos = Productos::all();  
    return view('admin.productos.ProductosManager', compact('productos'));  
}
```

PASO 2. DIAGRAMA DE FLUJO.



PASO 3. GRAFO



PASO 4.

Nodos (N): 5

Aristas (E): 4

Nodo Predicado/Condición /Decisión(P): 0

Rutas:

R1=1,2,3,4,5

COMPLEJIDAD CICLOMÁTICA

$V(G) = \text{DECISIONES} + 1$

$V(G) = 0 + 1$

$V(G) = 1$

$V(G) = E - N + 2$

$V(G) = 4 - 5 + 2$

$V(G) = 1$

Creación de los productos

PASO 1. CÓDIGO

Creación con validaciones del producto.

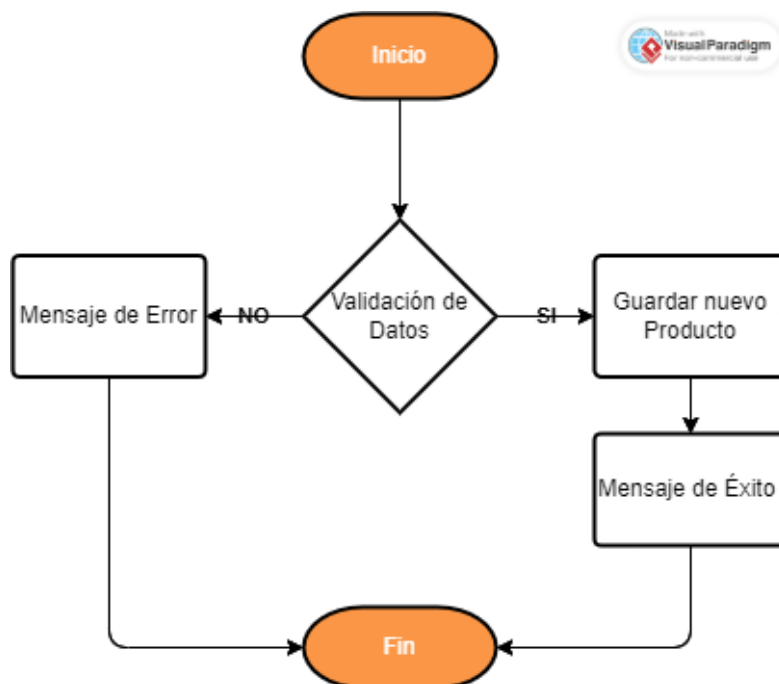
```
0 references | 0 overrides
public function create(Request $newProductoData)
{
    $validatedData = $newProductoData->validate([
        'nombre' => ['required', 'string', 'max:255', 'unique:productos'],
        'descripcion' => ['required', 'string'],
        'costo_unitario' => ['numeric'],
        'cantidad' => ['required', 'integer', 'min:1'],
        'horas_trabajo' => ['required', 'numeric', 'min:0'],
    ]);

    $newProducto = new Productos();

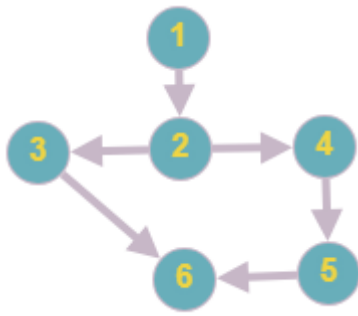
    try {
        $newProducto->nombre = $newProductoData->nombre;
        $newProducto->descripcion = $newProductoData->descripcion;
        $newProducto->costo_unitario = $newProductoData->costo_unitario;
        $newProducto->cantidad = $newProductoData->cantidad;
        $newProducto->horas_trabajo = $newProductoData->horas_trabajo;
        $newProducto->save();

        return back()->with(['success' => 'El Producto se registró con éxito']);
    } catch (Exception $e) {
        return $e->getMessage();
    }
}
```

PASO 2. DIAGRAMA DE FLUJO.



PASO 3. GRAFO



PASO 4.

Nodos (N): 6

Aristas (E): 6

Nodo Predicado/Condición /Decisión(P): 1

Rutas:

R1=1,2,3,6

R2=1,2,4,5,6

COMPLEJIDAD CICLOMÁTICA

$V(G) = \text{DECISIONES} + 1$

$V(G) = 1 + 1$

$V(G) = 2$

$V(G) = E - N + 2$

$V(G) = 6 - 6 + 2$

$V(G) = 2$

Edición del producto

PASO 1. CÓDIGO

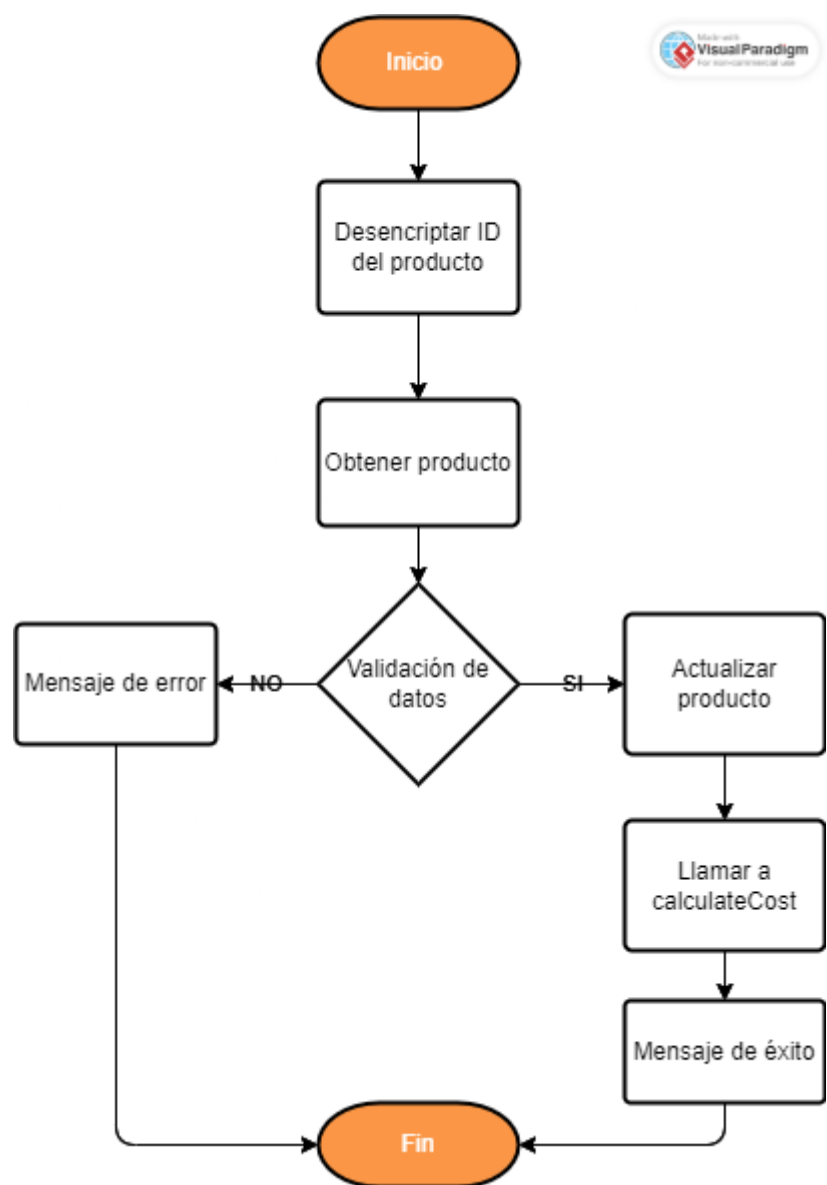
Edición con validaciones del producto.

```
0 references | 0 overrides
public function edit(Request $newData, $productID)
{
    $productID = decrypt($productID);
    $producto = Productos::where('id', $productID)->first();
    if ($producto->nombre != $newData->nombre) {
        $validatedData = $newData->validate([
            'nombre' => ['required', 'string', 'max:255', 'unique:productos'],
        ]);
    }
    $validatedData = $newData->validate([
        'descripcion' => ['required', 'string'],
        'costo_unitario' => ['numeric'],
        'cantidad' => ['required', 'integer', 'min:1'],
        'horas_trabajo' => ['required', 'numeric', 'min:0'],
    ]);

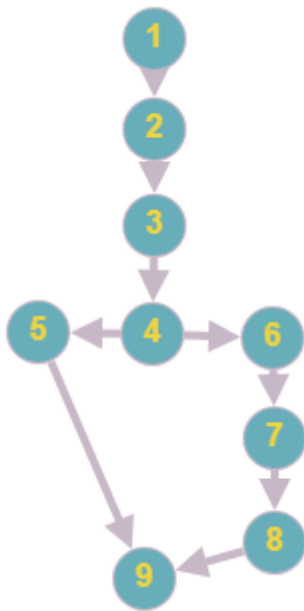
    try {
        $producto->nombre = $newData->nombre;
        $producto->descripcion = $newData->descripcion;
        $producto->costo_unitario = $newData->costo_unitario;
        $producto->cantidad = $newData->cantidad;
        $producto->horas_trabajo = $newData->horas_trabajo;
        $producto->save();
        $this->calculateCost($productID);

        return back()->with(['success' => 'El Producto se actualizó con éxito']);
    } catch (Exception $e) {
        return $e->getMessage();
    }
}
```

PASO 2. DIAGRAMA DE FLUJO.



PASO 3. GRAFO



PASO 4.

Nodos (N): 9

Aristas (E): 9

Nodo Predicado/Condición /Decisión(P): 1

Rutas:

R1=1,2,3,4,5,9

R2=1,2,3,4,6,7,8,9

COMPLEJIDAD CICLOMÁTICA

$V(G) = \text{DECISIONES} + 1$

$V(G) = 1 + 1$

$V(G) = 2$

$V(G) = E - N + 2$

$V(G) = 9 - 9 + 2$

$V(G) = 2$

Eliminación del producto

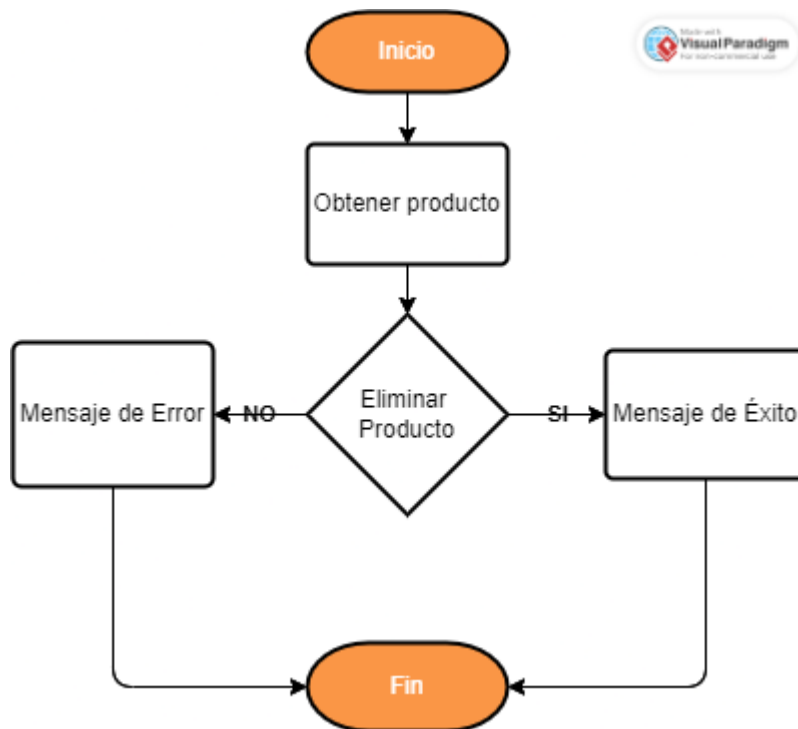
PASO 1. CÓDIGO

Eliminación del producto con validación de dato.

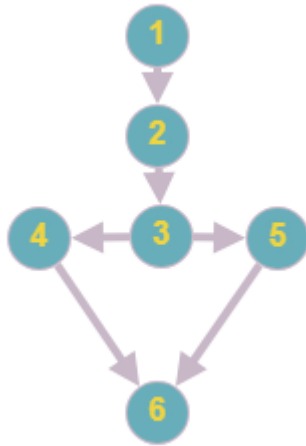
```
public function destroy($ProductoID)
{
    try {
        $Producto = Productos::where('id', decrypt($ProductoID))->first();
        $Producto->delete();

        return back()->with(['success' => 'El Producto se eliminó con éxito']);
    } catch (Exception $e) {
        return $e->getMessage();
    }
}
```

PASO 2. DIAGRAMA DE FLUJO.



PASO 3. GRAFO



PASO 4.

Nodos (N): 6

Aristas (E): 6

Nodo Predicado/Condición /Decisión(P): 1

Rutas:

R1=1,2,3,5,6

R2=1,2,3,4,6

COMPLEJIDAD CICLOMÁTICA

$V(G) = \text{DECISIONES} + 1$

$V(G) = 1 + 1$

$V(G) = 2$

$V(G) = E - N + 2$

$V(G) = 6 - 6 + 2$

$V(G) = 2$

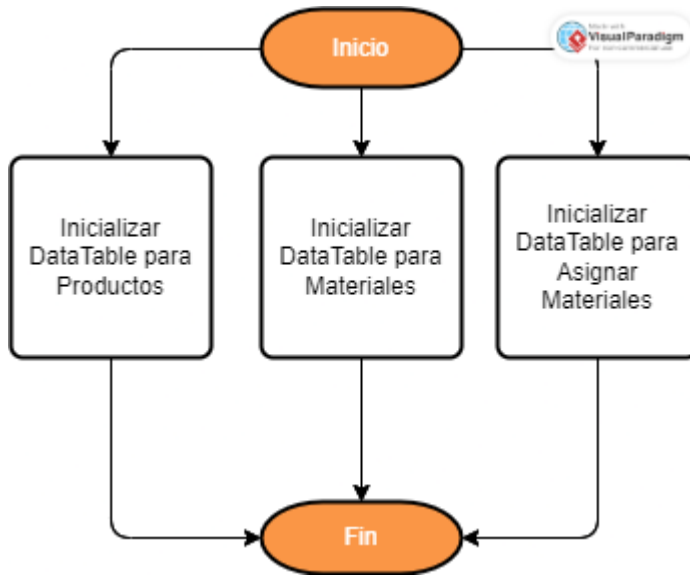
Habilitar búsqueda

PASO 1. CÓDIGO

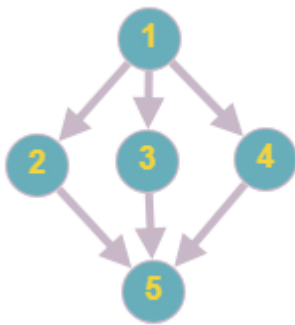
Búsqueda de productos, materiales.

```
<script>
  let tableProductos = new DataTable('#productos');
  let tableMateriales = new DataTable('#materiales');
  let tableAsignarMateriales = new DataTable('#asignarMateriales');
</script>
```

PASO 2. DIAGRAMA DE FLUJO.



PASO 3. GRAFO



PASO 4.

Nodos (N): 5

Aristas (E): 6

Nodo Predicado/Condición /Decisión(P): 0

Rutas:

R1=1,2,5

$$R2=1,3,5$$

$$R2=1,4,5$$

COMPLEJIDAD CICLOMÁTICA

$$V(G)=DECISIONES+1$$

$$V(G)=0+1$$

$$V(G)=1$$

$$V(G)=E-N+2$$

$$V(G)=6-5+2$$

$$V(G)=3$$

AGENDA