
Departamento de Ciencias de la Computación (DCCO)

Ingeniería en Software

Análisis y Diseño de Software NRC 8311

Sistema de inventario para la empresa Blozcell

Caja Blanca Versión 1.0

Presentado por: Jonathan Maigua

Director: Ing. Jenny Ruíz



PRUEBA CAJA BLANCA CONEXIÓN A LA BASE DE DATOS

```
const databaseConnection = () => {  
  mongoose.set('strictQuery', false);  
  const DB_URI = process.env.DB_URI;  
  
  mongoose.connect(  
    DB_URI,  
    {  
      useNewUrlParser: true,  
      useUnifiedTopology: true,  
    },  
    (err, res) => {  
      if (!err) {  
        console.log("CONEXION CORRECTA");  
      } else {  
        console.log("ERROR");  
      }  
    }  
  );  
};
```

Diagrama de flujo:

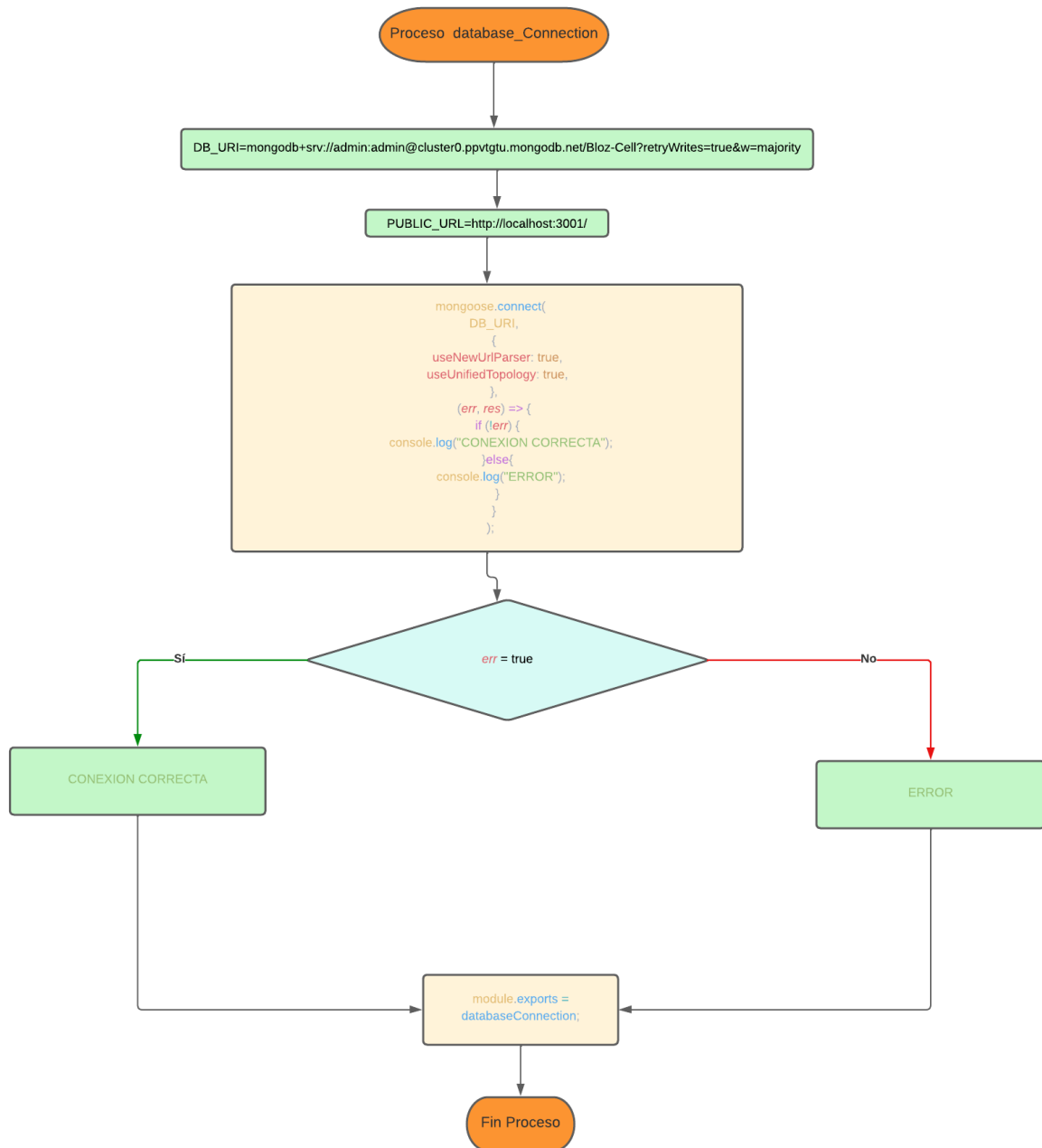
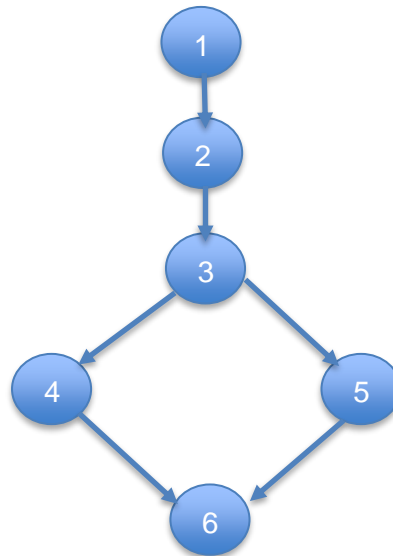


Diagrama de grafos:



RUTAS:

R1: 1, 2, 3, 4, 6

R2: 1, 2, 3, 5, 6

Complejidad Ciclomática

$$Cc = E - N + 2P$$

$$Cc = 6 - 6 + 2(1)$$

$$Cc = 2$$

E: Número de aristas

- E= 6

N: Número de nodos

- N=6

P: Número de nodos predicado

- P=1



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

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

$$V(G) = 2$$

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

$$V(G) = 1 \text{ nodo predicado} + 1 = 2$$

PRUEBA CAJA BLANCA LISTAR SALIDA DE PRODUCTO

```
getQuery = () => {  
  axios  
    .get(url)  
    .then((response) => {  
      this.setState({ data: response.data.data });  
    })  
    .catch((error) => {  
      console.log(error.message);  
    });  
};
```

Diagrama de flujo:

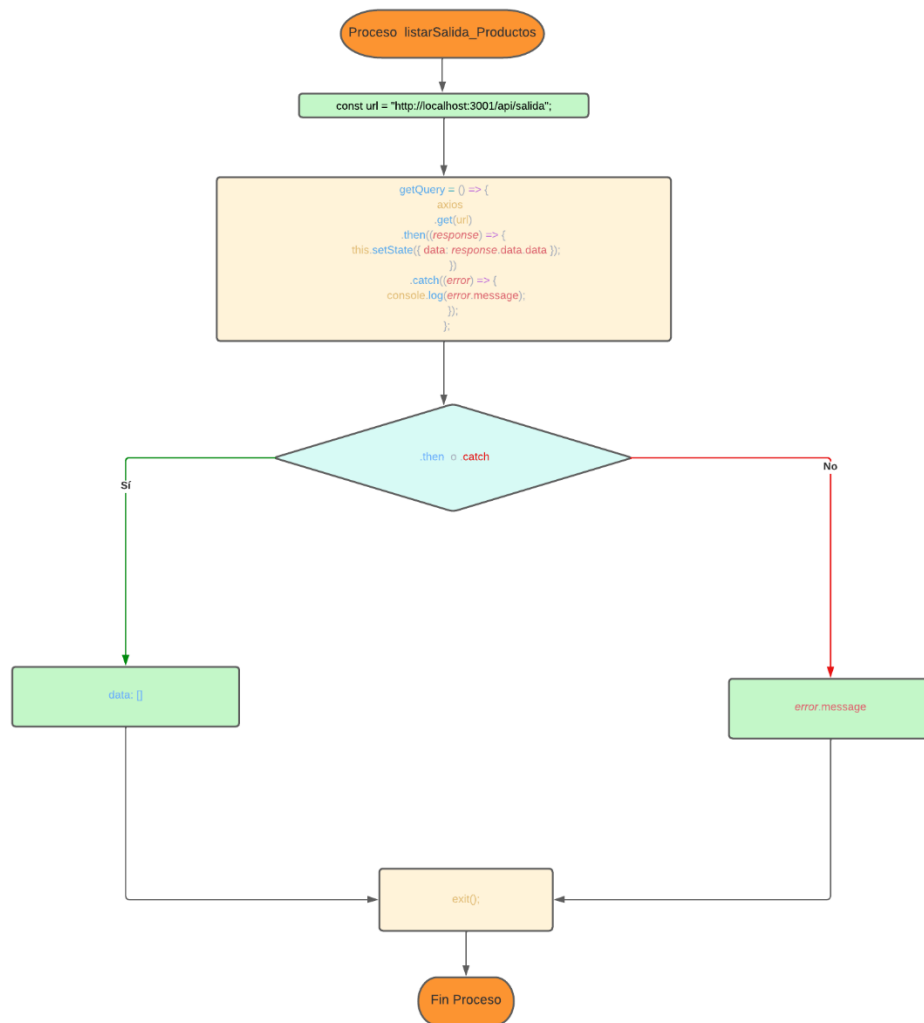
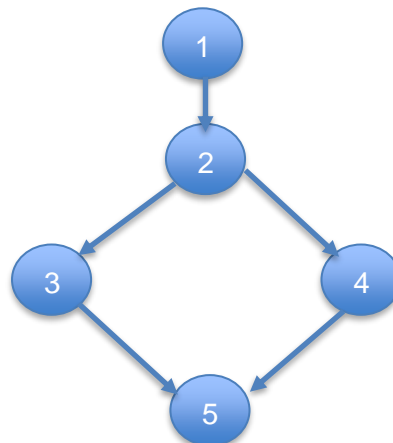


Diagrama de grafos:



Complejidad Ciclomática

$$Cc = E - N + 2P$$

$$Cc = 5 - 5 + 2(1)$$

$$Cc = 2$$

E: Número de aristas

- **E= 5**

N: Número de nodos

- **N=5**

P: Número de nodos predicado

- **P=1**

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

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

$$V(G) = 2$$

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

$$V(G) = 1 \text{ nodo predicado} + 1 = 2$$



PRUEBA CAJA BLANCA INSERTAR SALIDA DE PRODUCTO

```
//Modal insert
modalInsert = () => {
  this.setState({
    modalInsert: !this.state.modalInsert,
    form: { ...this.state.form, idProducto: this.state.data.length + 1 },
  });
};

//Funcion get Query
getQuery = () => {
  axios
    .get(url)
    .then((response) => {
      this.setState({ data: response.data.data });
    })
    .catch((error) => {
      console.log(error.message);
    });
};

postQuery = async () => {
  await axios
    .post(url, this.state.form)
    .then((response) => {
      this.modalInsert();
      this.getQuery();
    })
    .catch((error) => {
      console.log(error.message);
    });
};
```

Diagrama de flujo:

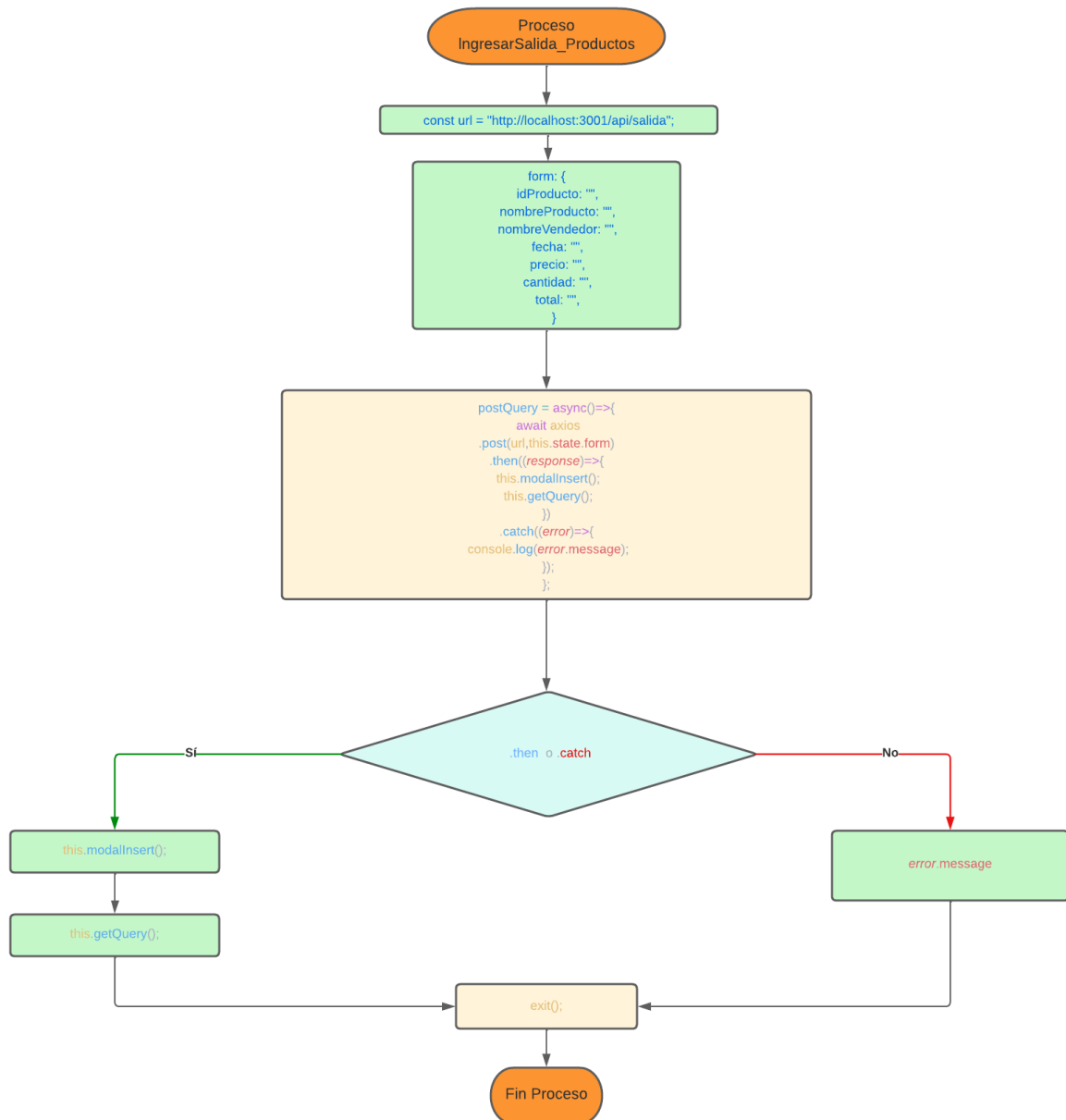
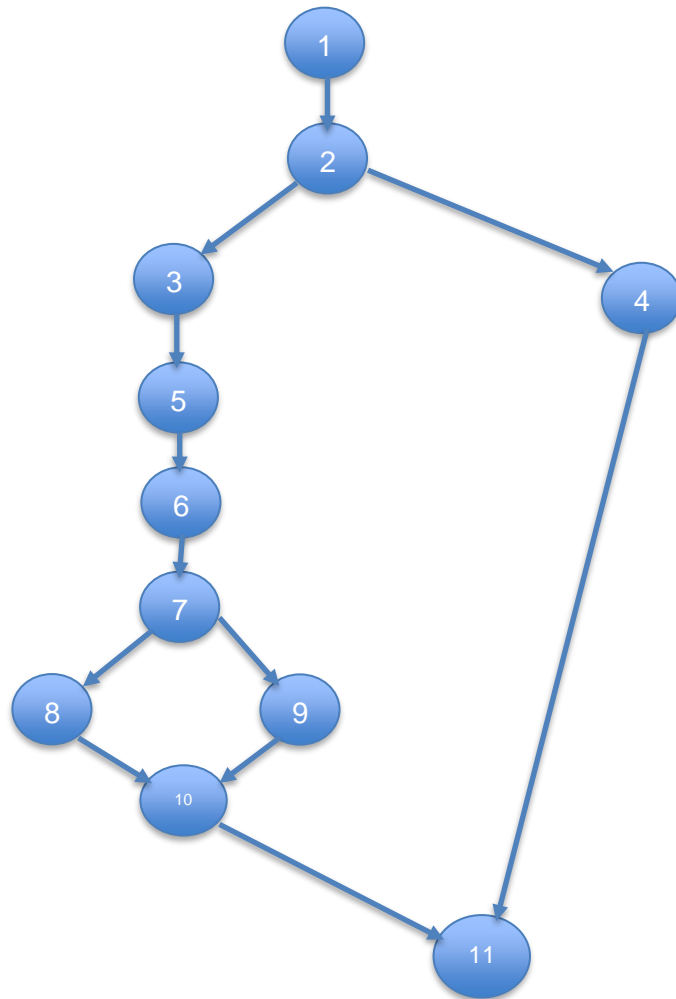


Diagrama de grafos:



RUTAS:

R1: 1, 2,3,5,6,7,8,10,11

R2: 1, 2,3,5,6,7,9,10,11

R3: 1,2,4,11

Complejidad Ciclomática

$$Cc = E - N + 2P$$

$$Cc = 12 - 11 + 2(2)$$

$$Cc = 5$$

E: Número de aristas

- **E = 12**

N: Número de nodos

- **N = 11**

P: Número de nodos predicado

- **P = 1**

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

$$V(G) = 12 - 11 + 2$$

$$V(G) = 3$$

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

$$V(G) = 2 \text{ nodo predicado} + 1 = 3$$



PRUEBA CAJA BLANCA EDITAR SALIDA DE PRODUCTO

```
putQuery = () => {  
  axios  
    .put(url + "/" + this.state.form.idProducto, this.state.form)  
    .then((response) => {  
      this.setState({ modalEdit: false })  
      this.getQuery();  
    });  
};
```

Diagrama de flujo:

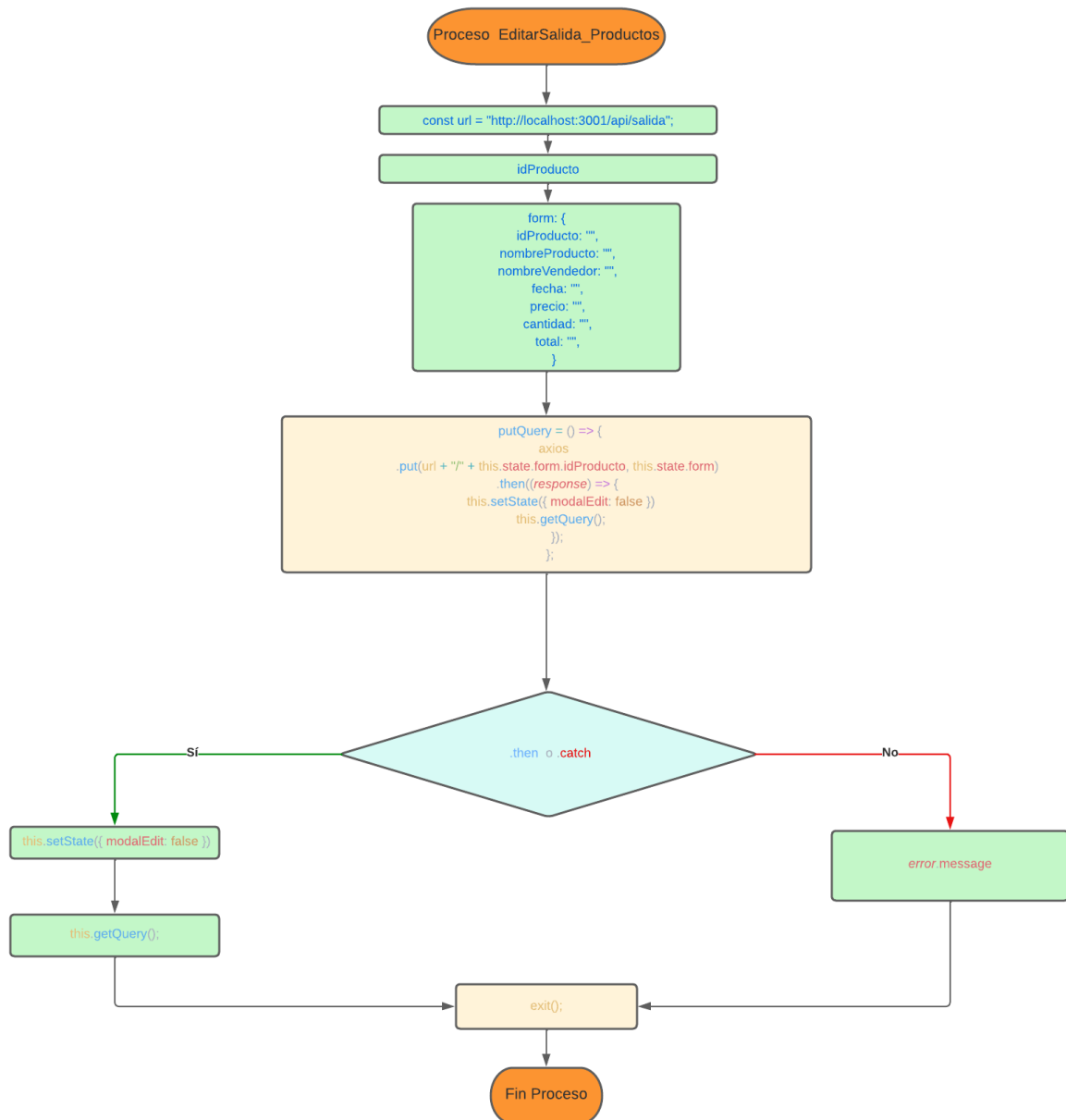
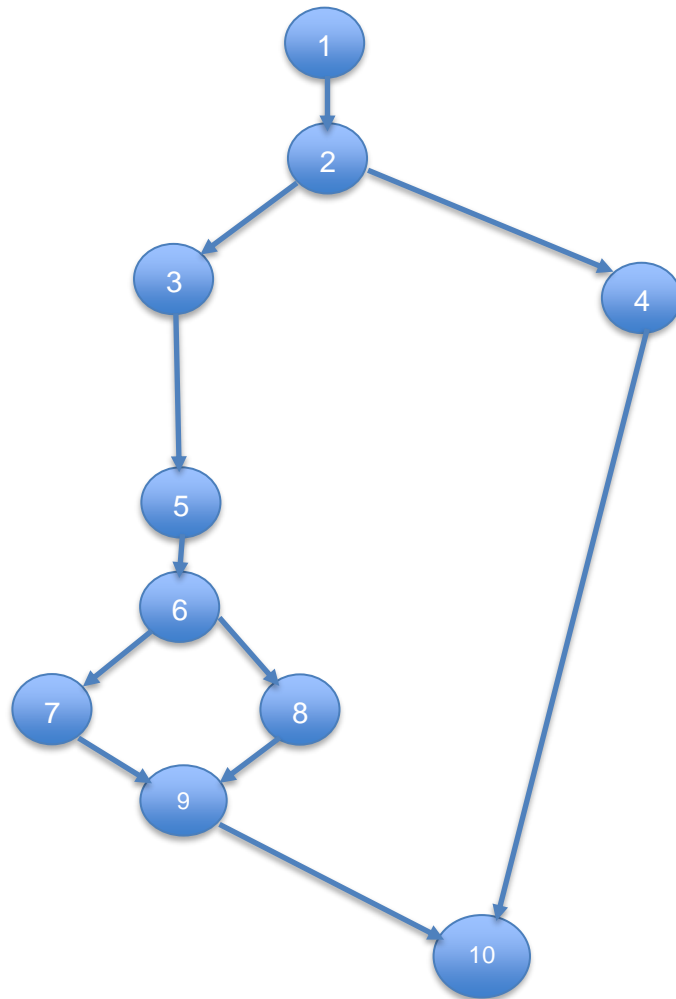


Diagrama de grafos:



RUTAS:

R1: 1, 2, 3, 5, 6,7,9,10

R2: 1, 2, 3,5,6,8,9,10

R3: 1, 2, 4,10

Complejidad Ciclomática

$$C_c = E - N + 2P$$

$$C_c = 11 - 10 + 2(2)$$

$$C_c = 5$$

E: Número de aristas

- **E = 11**

N: Número de nodos

- **N = 10**

P: Número de nodos predicado

- **P = 1**

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

$$V(G) = 11 - 10 + 2$$

$$V(G) = 3$$

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

$$V(G) = 2 \text{ nodo predicado} + 1 = 3$$

PRUEBA CAJA BLANCA ELIMINAR SALIDA DE PRODUCTO

```
deleteQuery = () => {  
  axios.delete(url + "/" + this.state.form.idProducto).then((response) => {  
    this.setState({ modalDelete: false });  
    this.getQuery();  
  });  
};
```

Diagrama de flujo:

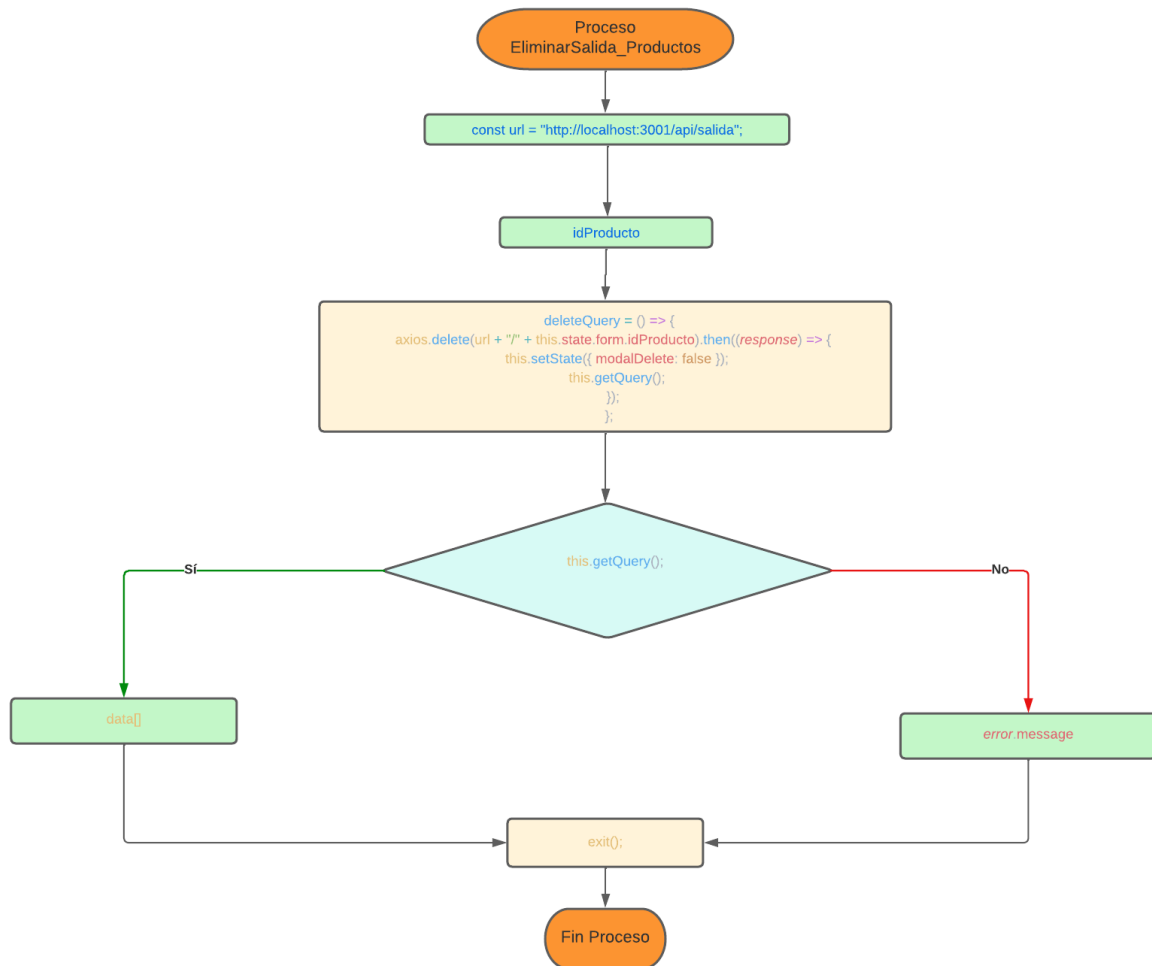
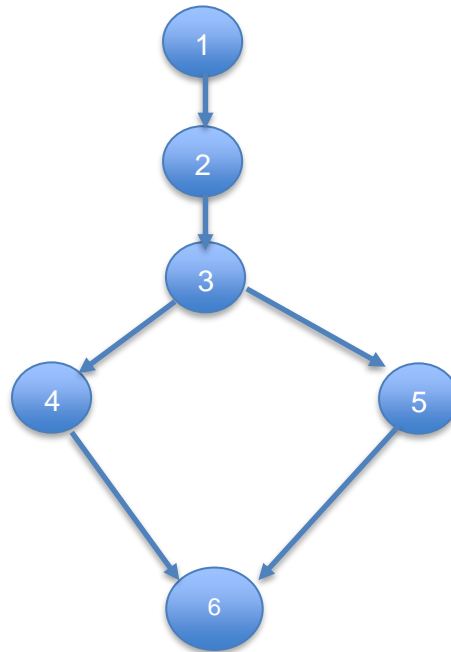


Diagrama de grafos:

Diagrama de grafos:



RUTAS:

R1: 1, 2, 3, 4, 6

R2: 1, 2, 3, 5, 6

Complejidad Ciclomática

$$Cc = E - N + 2P$$

$$Cc = 6 - 6 + 2 (1)$$

$$Cc = 2$$

E: Número de aristas

- E= 6

N: Número de nodos

- N=6

P: Número de nodos predicado

- P=1

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

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

$$V(G) = 2$$

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

$$V(G) = 1 \text{ nodo predicado} + 1 = 2$$

PRUEBA CAJA BLANCA REPORTE DE SALIDA DE PRODUCTO

```
const filtrar=(terminoBusqueda)=>{  
  var resultadosBusqueda=tablaProductos.filter((elemento)=>{  
    if(elemento.nombreProducto.toString().toLowerCase().includes(terminoBusqueda.toLowerCase())){  
      return elemento;  
    }  
  });  
  setProducto(resultadosBusqueda);  
}
```

Diagrama de flujo:

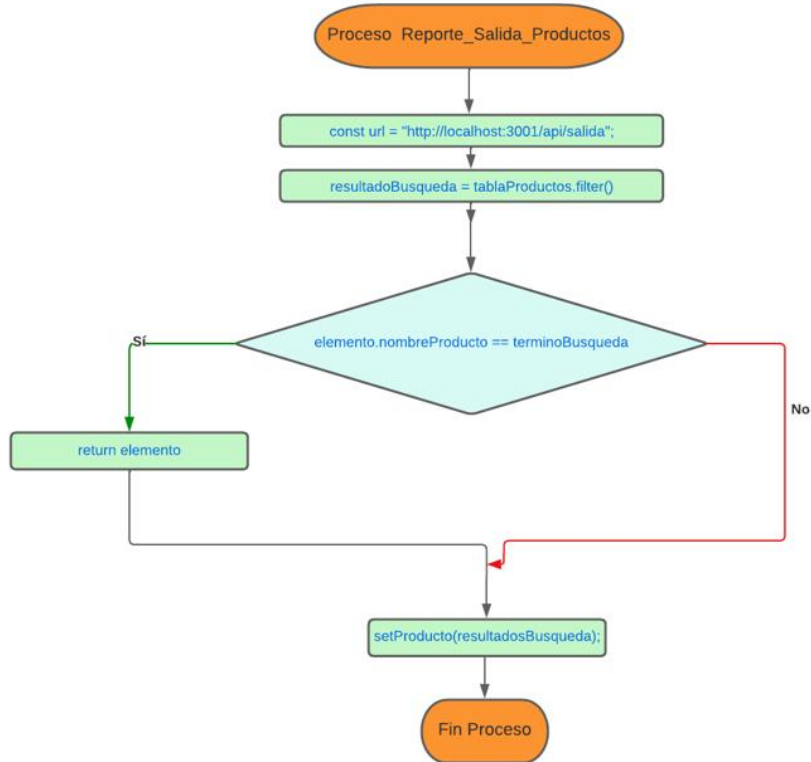
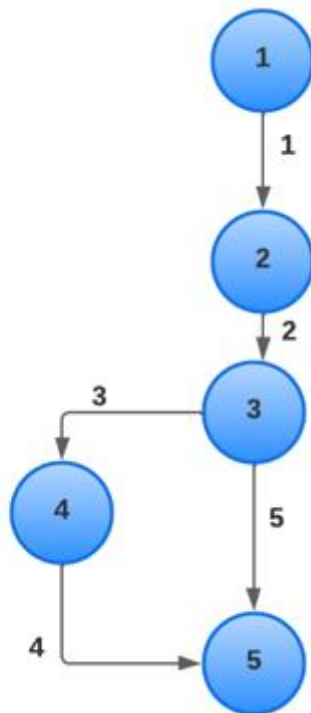


Diagrama de grafos:





RUTAS:

R1: 1, 2, 3, 4, 5

R2: 1, 2, 3, 5

Complejidad Ciclomática

$$Cc = E - N + 2P$$

$$Cc = 5 - 5 + 2 (1)$$

$$Cc = 2$$

E: Número de aristas

- E = 5

N: Número de nodos

- N = 5

P: Número de nodos predicado

- P = 1

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

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

$$V(G) = 2$$

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

$$V(G) = 1 \text{ nodo predicado} + 1 = 2$$

PRUEBA CAJA BLANCA REPORTE DE GANANCIA

```
const filtrar=(terminoBusqueda)=>{
  var resultadosBusqueda=tablaProductos.filter((elemento)=>{
    if(elemento.tipoOferta.toString().toLowerCase().includes(terminoBusqueda.toLowerCase())){
      return elemento;
    }
  })
}
```

```
    }  
  });  
  setProducto(resultadosBusqueda);  
}
```

Diagrama de flujo:

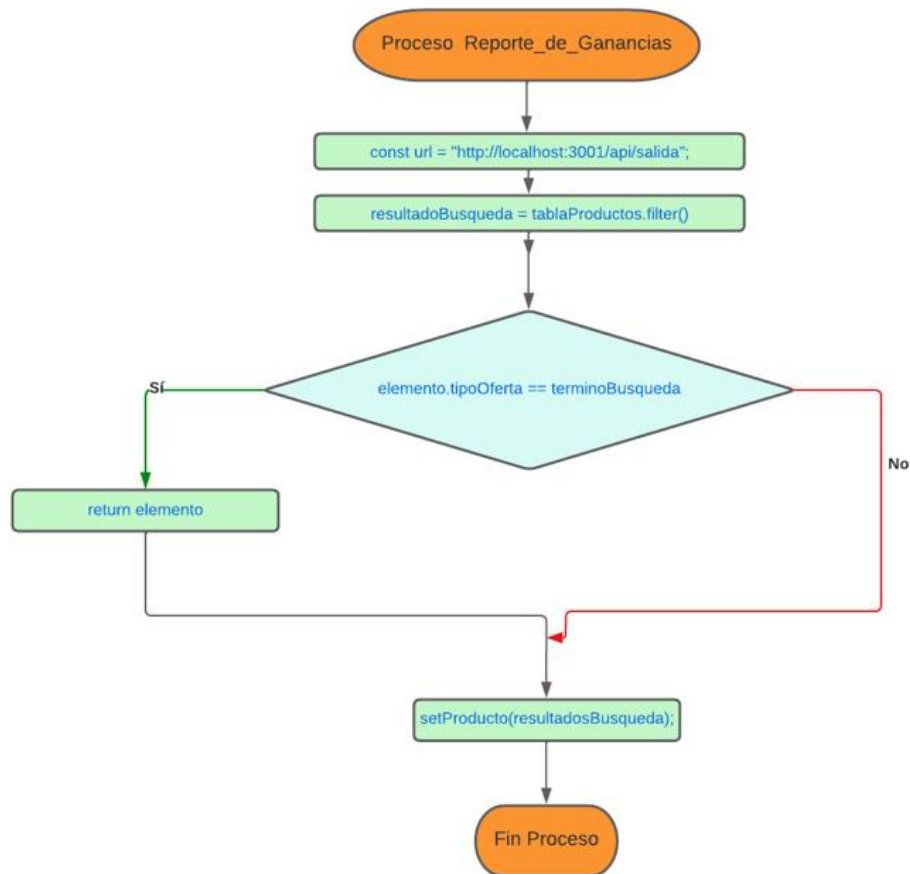
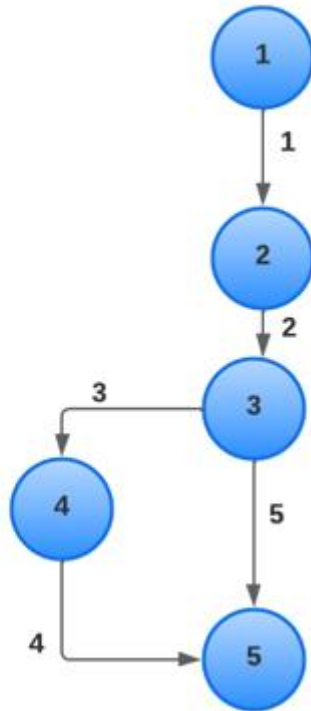


Diagrama de grafos:



RUTAS:

R1: 1, 2, 3, 4, 5

R2: 1, 2, 3, 5

Complejidad Ciclomática

$$Cc = E - N + 2P$$

$$Cc = 5 - 5 + 2 (1)$$

$$Cc = 2$$

E: Número de aristas

- E= 5

N: Número de nodos

- N=5

P: Número de nodos predicado



- $P=1$

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

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

$$V(G) = 2$$

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

$$V(G) = 1 \text{ nodo predicado} + 1 = 2$$