



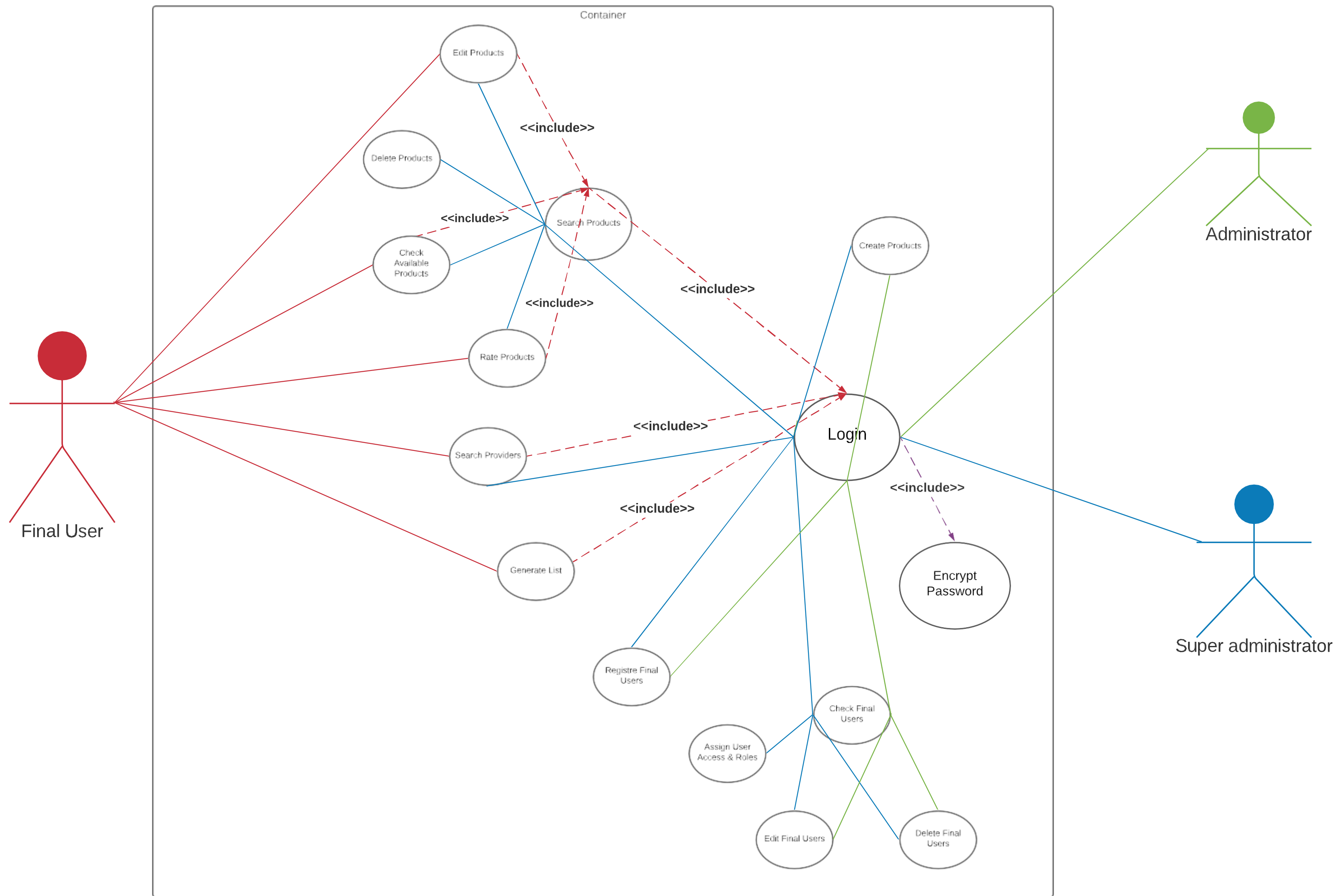
MISIONTIC 2022

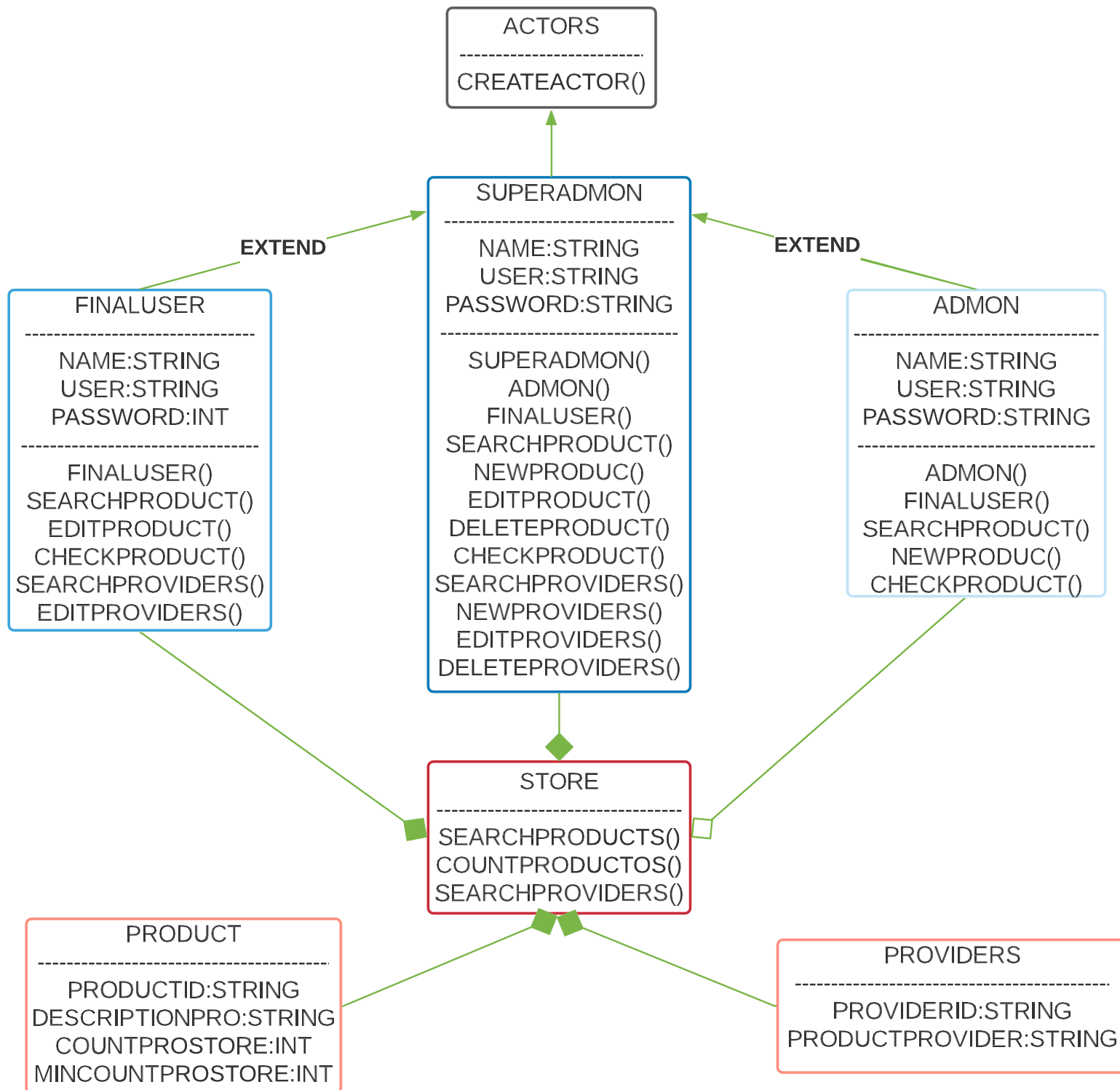
Sprint 3 grupo 7

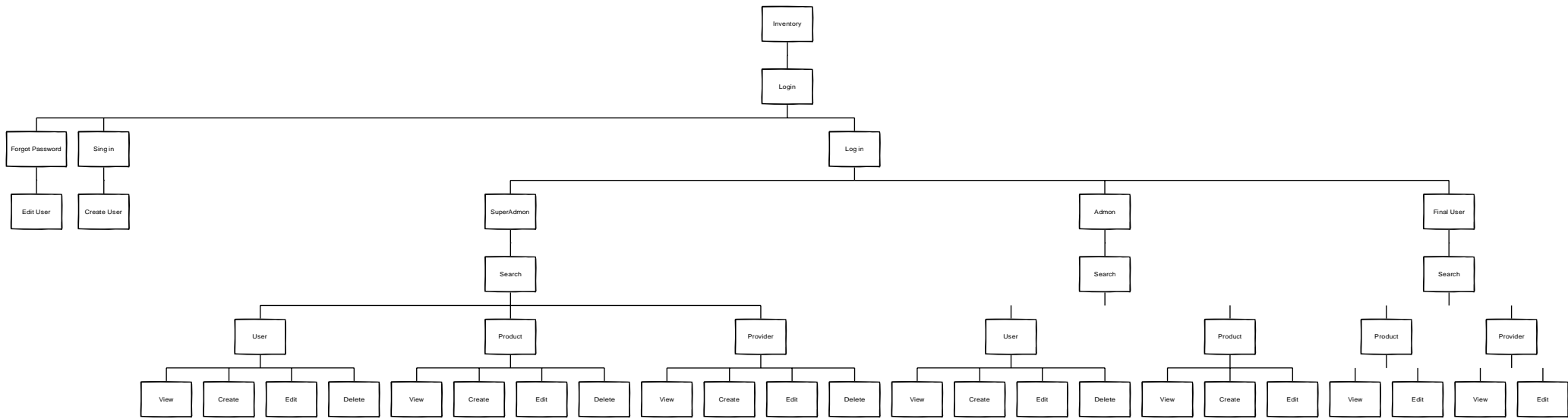
ASIGNATURA : DESARROLLO DE SOFTWARE

PRESENTADO POR : Diana Carolina Legarda López
Marco Tulio Niño Mogollón
Jonathan Cardozo Núñez
Erick Ricardo Mantilla Bolaño
Camilo Castillo Vásquez

NRC: 1201









https://inventario-DashboardAd



SUBARU

INVENTORY

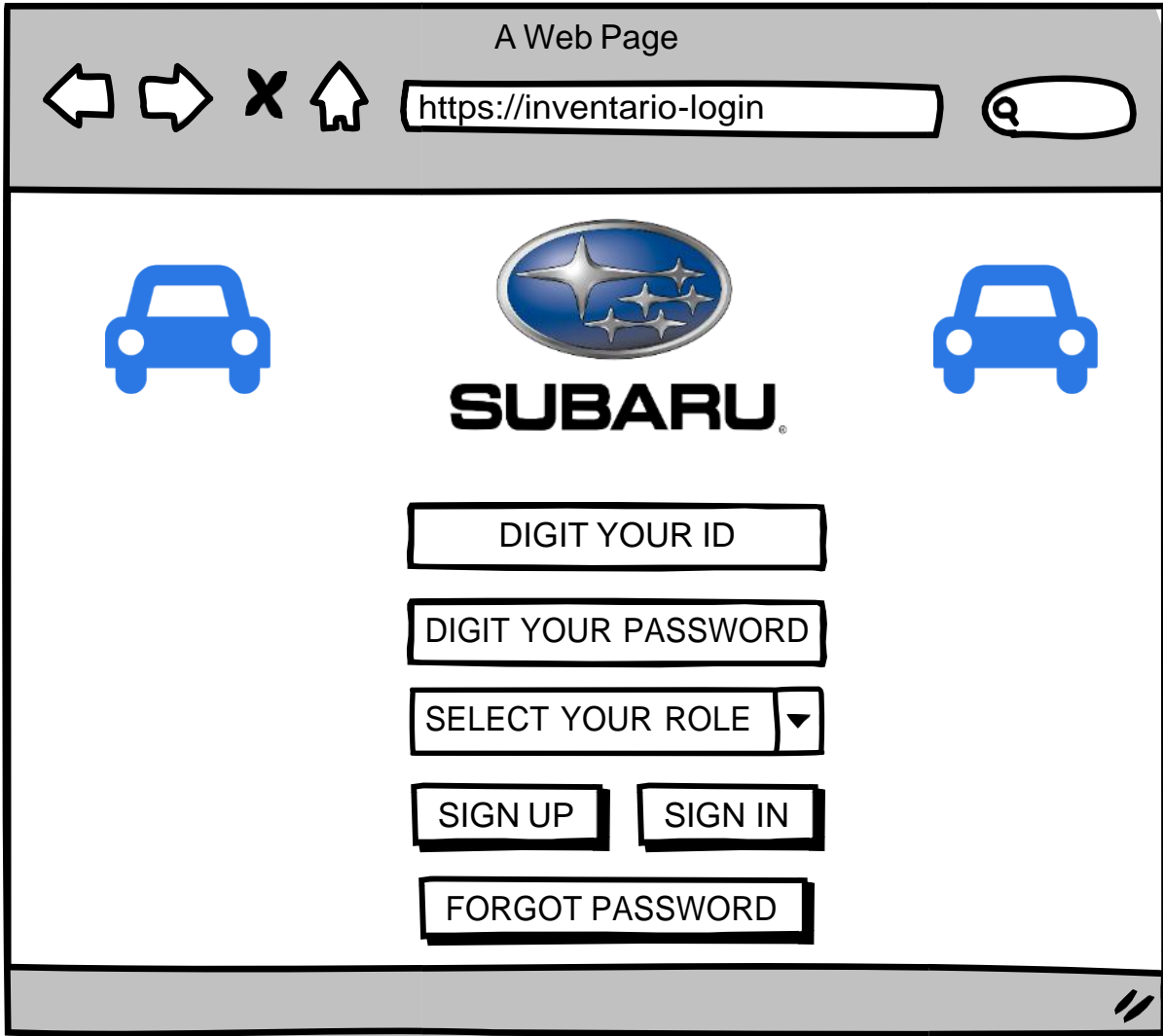
WHAT DO YOU WANT TO DO?

CREATE

EDIT

DELETE

SEARCH



Vista LOGIN



https://inventario_crearusuario.com



CREATE USER

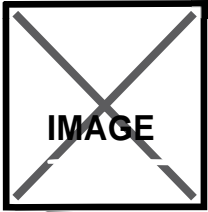
SING UP



https://inventario-editarusuario



EDIT USER



EDIT

FIRST NAME

EDIT

LAST NAME

EDIT

ROLL

EDIT

EMAIL

EDIT

ID

EDIT

TO CONTINUE, ENTER YOUR PASSWORD

PASSWORD

DEFINITELY EDIT USER





DELETE USER

WHY DO YOU WANT TO DELETE THIS USER?



TO CONTINUR, ENTER YOUR PASSWORD
AGAIN

[HAVE YOU FORGOTTEN THE PASSWORD?](#)

DEFINITELY DELETE USER

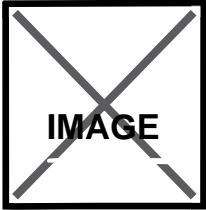




https://inventario-createproduct



CREATE PRODUCT



UPLOAD

PRODUCT NAME

PRODUCT DESCRIPTION

MINIMUM AMOUNT

QUANTITY AVAILABLE

PROVIDER NAME

TO CONTINUE, ENTER YOUR PASSWORD

PASSWORD

DEFINITELY CREATE PRODUCT



https://inventario-buscar



SEARCH

PRODUCT



PROVIDER



SEARCH



https://inventario-buscar



PRODUCTS

Product Name	Select
Product One	
Product Two	
Product Three	
Product Four	<input type="radio"/>

Description

Edit

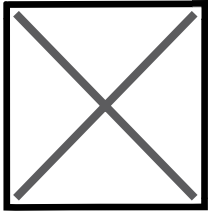
Delete



https://inventario-buscar



PRODUCT DESCRIPTION



PRODUCT NAME

PRODUCT DESCRIPTION

PROVIDERS

MINIMUM AMOUNT

QUANTITY AVAILABLE

SELECT PROVIDER



CHECK



https://inventario-buscar



PROVIDERS

Provider Name	Select
Provider One	
Provider Two	
Provider Three	
Provider Four	<input type="radio"/>

Description

Edit

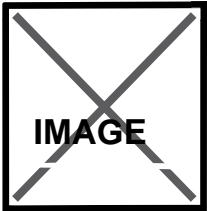
Delete



https://inventario-buscar



PROVIDER DESCRIPTION



PROVIDER NAME

PROVIDER ID

ASSOCIATED PRODUCTS

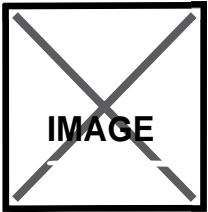
CONTACT INFO



https://inventario-editarproducto



EDIT PRODUCT



EDIT

PRODUCT NAME

EDIT

PRODUCT DESCRIPTION

EDIT

MINIMUM AMOUNT

EDIT

QUANTITY AVAILABLE

EDIT

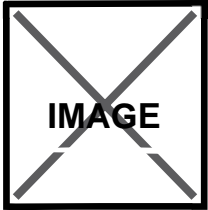
SAVE



https://inventario-eliminarproduc



DELETE PRODUCT



PRODUCT NAME

PRODUCT DESCRIPTION

WHY DO YOU WANT TO DELETE THIS PRODUCT?

REASON ▼

TO CONTINUE, ENTER YOUR PASSWORD

PASSWORD

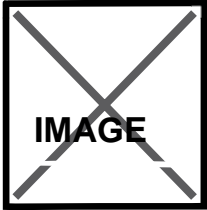
DEFINITELY DELETE PRODUCT



https://inventario-crearproveedo



CREATE PROVIDER



PROVIDER NAME

PROVIDER ID

ASSOCIATED PRODUCTS

TO CONTINUE, ENTER YOUR PASSWORD

PASSWORD

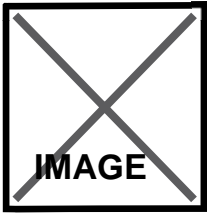
CREATE



https://inventario-editarproveedo



EDIT PROVIDER



PROVIDER NAME

EDIT

PROVIDER ID

EDIT

ASSOCIATED PRODUCTS

EDIT

WHY DO YOU WANT TO DELETE THIS PROVIDER?

REASON



TO CONTINUE, ENTER YOUR PASSWORD

PASSWORD

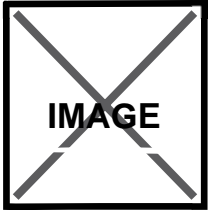
DEFINITELY EDIT PROVIDER



https://inventario-eliminarprovee



DELETE PROVIDER



PROVIDER NAME

ASSOCIATED PRODUCTS

WHY DO YOU WANT TO DELETE THIS PROVIDER?

REASON ▼

TO CONTINUE, ENTER YOUR PASSWORD

PASSWORD

DEFINITELY DELETE PROVIDER



https://login/signin/create



SUBARU

INVENTORY

WHAT DO YOU WANT TO CREATE?

CREATE USER

CREATE PROVIDER

CREATE PRODUCT

A Web Page



https://login/signin/edit



SUBARU

INVENTORY

WHAT DO YOU WANT TO EDIT?

EDIT USER

EDIT PROVIDER

EDIT PRODUCT



https://login/signin/delete



SUBARU

INVENTORY

WHAT DO YOU WANT TO DELETE?

DELETE USER

DELETE PROVIDER

DELETE PRODUCT

Métodos Manejadores del Backend

- Método Manejador 1: login()
- Ruta:
 - `@app.route('/login')`
 - `def login():`
 - `return render_template('signIn.html')`
- Métodos HTTP permitidos: POST
- Lógica algorítmica: *vista inicial donde el usuario puede iniciar sesión según su rol*

- Método Manejador 2: signin():
- Ruta:
 - `@app.route('/login/signin')`
 - `def signin():`
 - `return render_template('mainWeb.html')`
- Métodos HTTP permitidos: GET
- Lógica algorítmica: *vista página principal donde están todas las opciones principales según usuario*

- Método Manejador 3: createPage()
- Ruta:
 - `@app.route('/login /signin/create')`
 - `def createPage():`
 - `return render_template('createPage.html')`
- Métodos HTTP permitidos: GET
- Lógica algorítmica: *vista página donde están las opciones de seleccionar lo que se desea crear (usuario, producto y proveedores)*

- Método Manejador 4: createUser()
- Ruta:
 - `@app.route('/login/signin/create/createuser')`
 - `def createUser():`
 - `return render_template('createUser.html')`
- Métodos HTTP permitidos: POST
- Lógica algorítmica: *vista página para ingresar la información necesaria para crear usuarios*

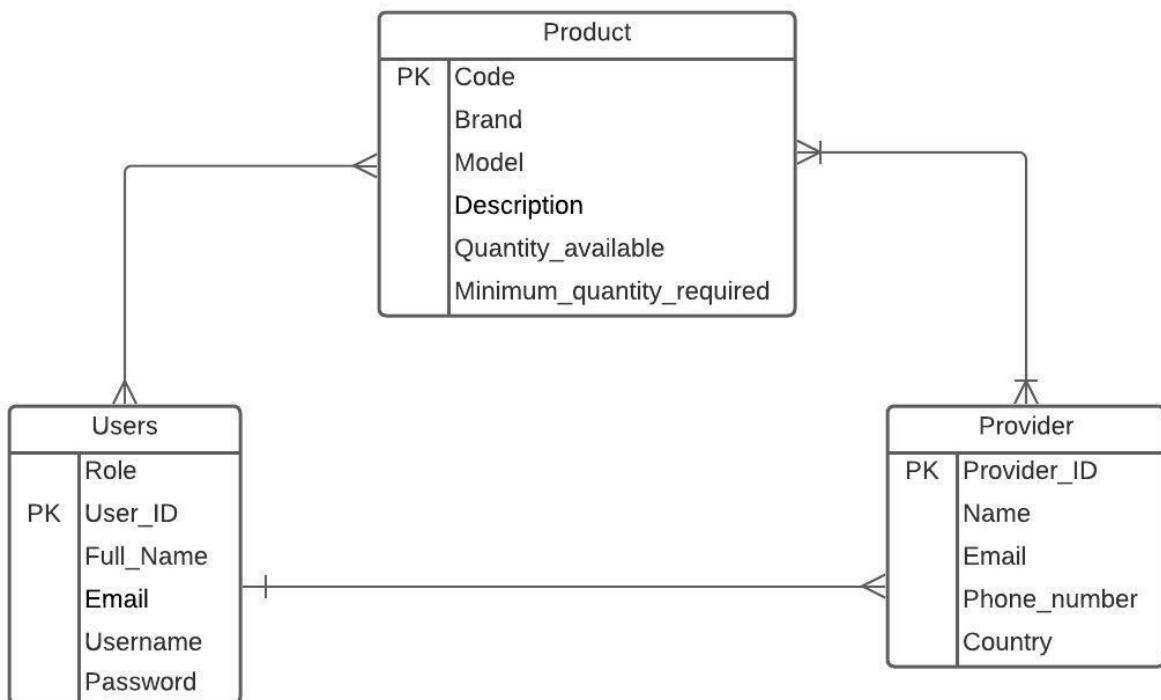
- Método Manejador 5: createProduct()
- Ruta:
 - @app.route('/login/signin/create/createproduct')
 - def createProduct():
 - return render_template('createProduct.html')
- Métodos HTTP permitidos: POST
- Lógica algorítmica: *vista página para ingresar la información necesaria para crear productos*
- Método Manejador 6: createProvider ()
- Ruta:
 - @app.route('/login/signin/create/createprovider')
 - def createProvider():
 - return render_template('createProvider.html')
- Métodos HTTP permitidos: POST
- Lógica algorítmica: vista página para ingresar la información necesaria para crear proveedor
- Método Manejador 7: editPage()
- Ruta:
 - @app.route('/login/signin/edit')
 - def editPage():
 - return render_template('editPage.html')
- Métodos HTTP permitidos: GET
- Lógica algorítmica: vista página donde están las opciones de seleccionar lo que se desea editar (usuario, producto y proveedores)
- Método Manejador 8: editUser()
- Ruta:
 - @app.route('/login/signin/edit/edituser')
 - def editUser():
 - return render_template('editUser.html')
- Métodos HTTP permitidos: POST
- Lógica algorítmica: vista página para ingresar la información necesaria para editar usuario

- Método Manejador 9: editProduct()
 - Ruta:
 - `@app.route('/login/signin/edit/editproduct')`
 - `def editProduct():`
 - `return render_template('editProduct.html')`
 - Métodos HTTP permitidos: POST
 - Lógica algorítmica: vista página para ingresar la información necesaria para editar producto
-
- Método Manejador 10: editProvider()
 - Ruta:
 - `@app.route('/login/signin/edit/editprovider')`
 - `def editProvider():`
 - `return render_template('editProvider.html')`
 - Métodos HTTP permitidos: POST
 - Lógica algorítmica: vista página para ingresar la información necesaria para editar proveedor
-
- Método Manejador 11: deletePage()
 - Ruta:
 - `@app.route('/login/signin/delete')`
 - `def deletePage():`
 - `return render_template('deletePage.html')`
 - Métodos HTTP permitidos: GET
 - Lógica algorítmica: vista página donde están las opciones de seleccionar lo que se desea eliminar (usuario, producto y proveedores)
-
- Método Manejador 12: deleteUser()
 - Ruta:
 - `@app.route('/login/signin/delete/deleteuser')`
 - `def deleteUser():`
 - `return render_template('deleteUser.html')`
 - Métodos HTTP permitidos: POST
 - Lógica algorítmica: vista página para ingresar la información necesaria para eliminar usuario

- Método Manejador 13: deleteProduct ()
 - Ruta:
 - @app.route('/login/signin/delete/deleteproduct')
 - def deleteProduct():
 - return render_template('deleteProduct.html')
 - Métodos HTTP permitidos: POST
 - Lógica algorítmica: vista página para ingresar la información necesaria para eliminar producto
-
- Método Manejador 14: deleteProvider()
 - Ruta:
 - @app.route('/login/signin/delete/deleteprovider')
 - def deleteProvider():
 - return render_template('deleteProvider.html')
 - Métodos HTTP permitidos: POST
 - Lógica algorítmica: vista página para ingresar la información necesaria para editar proveedor
-
- Método Manejador 15: search()
 - Ruta:
 - @app.route('/login/signin/search')
 - def search():
 - return render_template('search.html')
 - Métodos HTTP permitidos: POST
 - Lógica algorítmica: vista página donde están las opciones de seleccionar lo que se desea buscar (producto y proveedores)
-
- Método Manejador 16: searchProduct()
 - Ruta:
 - @app.route('/login/signin/search/product')
 - def searchProduct():
 - return render_template('productDescription.html')
 - Métodos HTTP permitidos: POST
 - Lógica algorítmica: vista página donde se presenta toda la información relacionada con el producto

- Método Manejador 17: searchProvider()
- Ruta:
 - @app.route('/login/signin/search/provider)
 - def searchProvider():
 - return render_template('providerDescription.html')
- Métodos HTTP permitidos: GET
- Lógica algorítmica: vista página donde se presenta toda la información relacionada con el proveedor

Diagrama base de datos



Descripción de tabla: conta de 3 tablas principales USER, PRODCUT y PROVIDER, inicialmente la tabla Users en cual contiene todo los atributos requeridos donde User_ID es nuestra primary key debido a que es la que identifica el tipo de usuario y es único e irrepetible, este se conecta con Producto y Provider.

También se encuentra la relación producto provider donde se visualiza que un proveedor puede tener varios productos y a su vez un producto puede provenir de varios proveedores

LINK GITHUB: https://github.com/dclegarda/Grupo7_Sprints.git