

Droguería Promedic El Gran C.H

- Natalia Ruiz Parra
- Sara Valentina Garcia Garcia
- Erick Hazann Zarto Gil
- Darwin Stiven Mora Martinez
- Jhonny Santiago Espitia Mendoza



Índice

1. Presentación.
 - 1.1 Objetivo general.
 - 1.2 Planteamiento problema.
2. Base de datos.
 - 2.1 Construcción de la base de datos.
 - 2.2 Uso de la bases de datos.
 - 2.3 Encriptación de datos.
3. FrontEnd Funcional.
4. Control de versiones
5. Prototipo navegable.

1. Presentación

Objetivo General

Desarrollar el sistema de información orientado a la web para la gestión de inventario de la DROGUERÍA PROMEDIC EL GRAN C.H, mejorando la eficiencia operativa, la precisión en inventarios y la satisfacción del cliente.





2. Base de datos



```
1  -- Crear base de datos
2  • create database ProyecPromedic;
3  • use ProyecPromedic;
4  -- Creacion de las tablas
5  • create table Categoria(
6      Id_Categoria int auto_increment not null,
7      Nombre_Categoria varchar (400) not null,
8      Descripcion_Categoria varchar(200) not null,
9      primary key (Id_categoria)
10 );
1
2  • create table Clasificacion (
3      Id_Clasificacion int auto_increment not null,
4      Nombre_Clasificacion varchar(200) not null,
5      Descripcion_Clasificacion varchar(200) not null,
6      Id_Categoria int,
7      primary key (Id_clasificacion),
8      foreign key (Id_categoria) references Categoria(Id_categoria)
9  );
10
1  • create table Estado_Producto(
2      Id_Estado_Producto int auto_increment not null,
3      Tipo_Estado_Producto varchar(200) not null,
```

2.1 Construcción de la base de datos

Esta parte del proyecto demuestra cómo se diseñan las tablas y estructuras de la base de datos. Usando DDL (SQL), se definen las entidades, columnas y restricciones necesarias para almacenar los datos.



2.2 Uso de la base de datos

En esta parte del proyecto se evalúa cómo se manipulan los datos. Se insertan, actualizan, eliminan y consultan registros. Se incluyen operaciones de consultas complejas con JOINS, consultas y subconsultas en SQL.

```
117 -- Insertar datos en la tabla Categoria
118 • INSERT INTO Categoria (Id_Categoria, Nombre_Categoria, Descripcion_Categoria) VALUES
119 (1, 'Medicamentos', 'Medicamentos generales'),
120 (2, 'Suplementos', 'Suplementos alimenticios'),
121 (3, 'Cuidado Personal', 'Productos de cuidado personal'),
122 (4, 'Instrumental', 'Instrumentos médicos'),
123 (5, 'Ortopedia', 'Equipos ortopédicos'),
124 (6, 'Higiene', 'Productos de higiene personal'),
125 (7, 'Bebidas', 'Bebidas medicinales'),
126 (8, 'Accesorios', 'Accesorios médicos'),
127 (9, 'Vitaminas', 'Vitaminas y minerales'),
128 (10, 'Belleza', 'Productos de belleza'),
129 (11, 'Calzado', 'Calzado ortopédico'),
130 (12, 'Cosméticos', 'Cosméticos medicinales'),
131 (13, 'Ropa Médica', 'Uniformes y ropa para médicos'),
132 (14, 'Protección', 'Equipos de protección personal'),
133 (15, 'Electrónica', 'Dispositivos electrónicos médicos'),
134 (16, 'Herbolarios', 'Productos a base de hierbas'),
135 (17, 'Diabetes', 'Productos para diabéticos'),
136 (18, 'Inmunología', 'Medicamentos para el sistema inmunológico'),
137 (19, 'Geriatría', 'Productos para personas mayores'),
138 (20, 'Pediatria', 'Productos para niños');
139
-- 1 Obtiene detalles completos de los productos junto con su estado, marca y cantidad en stock.
SELECT
    p.Nombre_Producto AS Nombre_Producto, -- Nombre del producto
    p.Descripcion_Producto AS Descripcion, -- Descripción del producto
    e.Tipo_Estado_Producto AS Estado, -- Estado del producto (e.g., disponible, agotado)
    m.Marca_Producto AS Marca, -- Marca del producto
    p.Cantidad_Stock AS Cantidad_Stock -- Cantidad disponible en stock
FROM
    Producto p -- Desde la tabla Producto
INNER JOIN
    Estado_Producto e ON p.Id_Estado_Producto = e.Id_Estado_Producto -- Unir con Estado_Producto para obtener el estado
INNER JOIN
    Marca m ON p.Id_Marca = m.Id_Marca; -- Unir con Marca para obtener la marca del producto
```

4. Encriptación de datos

Se evidencia la implementación de seguridad en la base de datos. Incluye la encriptación de contraseñas.

```
322 • ALTER TABLE Regente
323 DROP COLUMN Contraseña;
324
325 -- 2. Agregar columna de clave encriptada
326 • ALTER TABLE Regente
327 ADD COLUMN Contraseña VARBINARY(64);
328
329 -- Insertar los datos con la contraseña en texto plano
330 • INSERT INTO Regente (Id_Regente, Nombre, Apellido, DNI, Fecha_Contratacion, Licencia, Correo, Telefono, Id_Turno, Contraseña) VALUES
331 (1, 'Pablo', 'Lopez', 87654321, '2023-05-10', 987654321, 'Pablo.lopez@example.com', 300000000, 1, UNHEX(SHA2('54321', 256)));
332
333 -- 3. Encriptar la contraseña con SHA2
334 • UPDATE Regente
335 SET Contraseña = UNHEX(SHA2(Contraseña, 256));
336
337 -- 4. Renombrar la columna Contraseña
338 • ALTER TABLE Regente
339 RENAME COLUMN Contraseña TO Contraseña_encriptada;
340
341 -- 5. verificar los resultados de la encriptacion
342 • SELECT Id_Regente, Contraseña_encriptada
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	Id_Regente	Contraseña_encriptada
▶	1	54321
•	54321	54321



3. FrontEnd funcional.

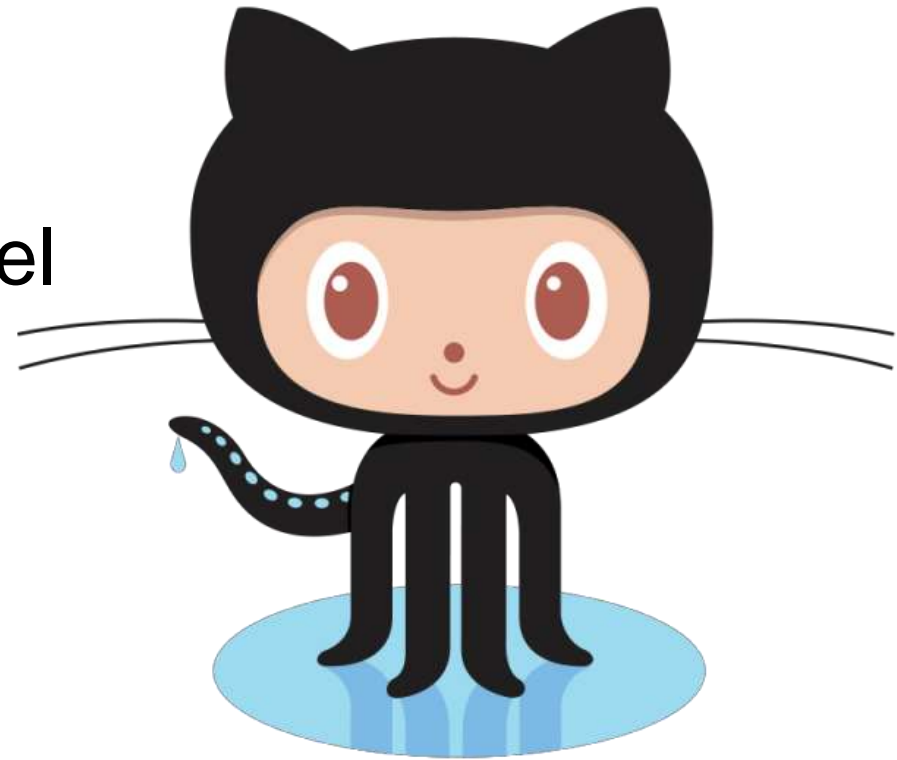
Esta sección muestra la implementación de la interfaz de usuario del proyecto, utilizando Bootstrap, siguiendo el diseño del prototipo.

```
Formulario.html X Registros.html # Registro.css
Formulario.html > html > body > div.container.mt-5 > div.row.justify-content-center > div.col-md-6
2 <html lang="es">
9 <body>
10 <div class="container mt-5">
11 <div class="row justify-content-center">
12 <div class="col-md-6">
13 <div class="card">
14 <div class="card-header text-center">
15 <h3>Inserción Fórmulas Médicas</h3>
16 </div>
17 <div class="card-body">
18 <form id="registroForm">
19 <div class="mb-3">
20 <label for="identificacion" class="form-label">Número de Identificación</label>
21 <input type="text" class="form-control" id="identificacion" placeholder="Ingresa tu identificación">
22 </div>
23 <div class="mb-3">
24 <label for="nombre" class="form-label">Nombre Completo</label>
25 <input type="text" class="form-control" id="nombre" placeholder="Ingresa tu nombre completo">
26 </div>
27 <div class="mb-3">
28 <label for="fecha" class="form-label">Fecha de Inserción</label>
29 <input type="date" class="form-control" id="fecha" required>
30 </div>
31 <div class="mb-3">
32 <label for="imagen" class="form-label">Subir Imagen</label>
33 <input type="file" class="form-control" id="imagen" accept="image/*" required>
34 </div>
35 <button type="submit" class="btn btn-primary w-100">Registrar</button>
36 </form>
37 </div>
```

4. Control de versiones



En nuestro proyecto, la implementación de un control de versiones es una pieza fundamental para asegurar la integridad y la coherencia en el desarrollo del software. A través de técnicas especializadas, gestionamos de manera efectiva los cambios en el código fuente y otros archivos clave.





5. Prototipo navegable

Se presenta un prototipo navegable que simula el funcionamiento del sistema, diseñado con JAVA JS, HTML y CSS, para visualizar el flujo y la experiencia del usuario.

Inserción Fórmulas Médicas

Número de Identificación

Ingresa tu identificación

Nombre Completo

Ingresa tu nombre completo

Fecha de Inserción

dd/mm/aaaa

Subir Imagen

Elegir archivo

No se ha seleccionado ningún archivo

Registrar



G R A C I A S

Línea de atención al ciudadano: 01 8000 910270
Línea de atención al empresario: 01 8000 910682



www.sena.edu.co