Manual Técnico

1. Introducción

Este documento describe la arquitectura, tecnologías y funcionalidades clave del sistema de administración de "La Computadora Feliz".

2. Arquitectura del Sistema

El sistema sigue una arquitectura cliente-servidor con los siguientes componentes:

• Frontend: Aplicación JSP

• Backend: API desarrollada con Jakarta EE

• Base de Datos: MySQL

3. Tecnologías Utilizadas

- Frontend:
 - o HTML, CSS, JSP
 - Bootstrap para estilos
- Backend:
 - Jakarta EE
 - JDBC para acceso a MySQL
 - o Servlets
- Base de Datos:
 - o MySQL

4. Diagramas UML

Diagrama Entidad/Relacion

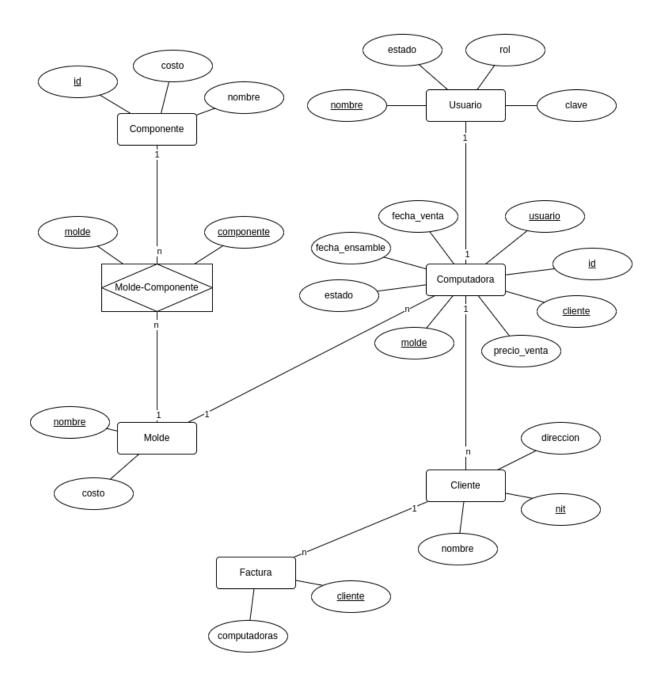


Diagrama de Tablas

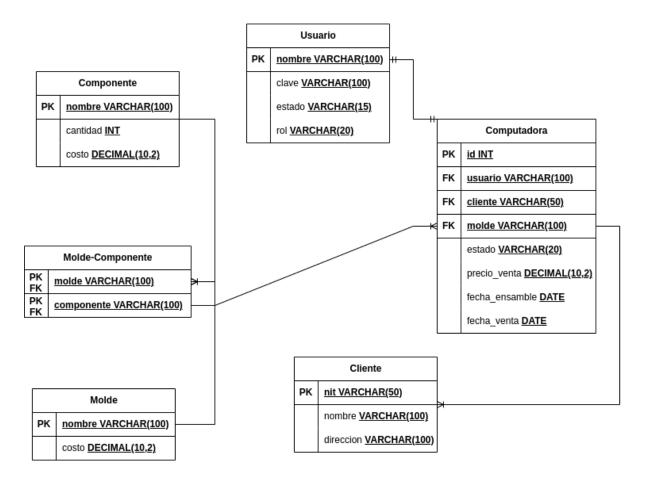


Diagrama de Clases

Vista

Controlador

Data Access Object ComponenteDAO Componente InicioView ComponenteController nombre: String + field: type crearComponente(): void obtenerComponente(): void actualizarComponente(): void eliminarComponente(): void cantidad: int RegistrarComponenteView costo: double + method(type): type Molde-ComponenteController + field: type ActualizarComponenteView + method(type): type Molde EliminarComponenteView MoldeController nombre: String costo: double CrearMoldeView + field: type MoldeDAO + method(type): type crearMolde(): void obtenerMolde(): void actualizarMolde(): void eliminarMolde(): void VerMoldesView nombre: String clave: String UsuarioController estado: String rol: String ActualizarMoldeView + field: type UsuarioDAO + method(type): type EliminarMoldeView Cliente crearUsuario(): void obtenerUsuario(): void nit: String nombre: String ClienteController actualizarUsuario(): void + field: type eliminarUsuario(): void direccion: String + method(type): type ClienteDAO Computadora ComputadoraController id: int crearCliente(): void obtenerCliente(): void actualizarCliente(): void eliminarCliente(): void usuario: String cliente: String molde: String + field: type + method(type): type estado: String precio_venta: double fecha_ensamble: LocalDate ComputadoraDAO fecha_venta: LocalDate

Modelo

5. Mapeo Físico Base de Datos

-- Creacion del Esquema CREATE SCHEMA TechSolutionsHub; **USE TechSolutionsHub**;

crearComputadora(): void obtenerComputadora(): Computadora actualizarComputadora(): void eliminarComputadora(): void consultarCompras():Computadora[] consultarDevoluciones():Computadora[]

-- Tabla Usuario CREATE TABLE usuario (nombre VARCHAR(100) PRIMARY KEY, clave VARCHAR(100) NOT NULL, estado VARCHAR(20) NOT NULL, rol VARCHAR(20) NOT NULL);

```
-- Tabla Componente
CREATE TABLE componente (
      id INT AUTO INCREMENT PRIMARY KEY,
      nombre VARCHAR(100),
      costo DECIMAL(10,2) NOT NULL,
      estado VARCHAR(100)
);
-- Tabla Molde
CREATE TABLE molde (
      nombre VARCHAR(100) PRIMARY KEY,
      costo DECIMAL(10,2)
);
-- Tabla Molde-Componente (Relación)
CREATE TABLE molde componente (
      molde VARCHAR(100),
      componente INT,
      PRIMARY KEY (molde, componente),
      FOREIGN KEY (molde) REFERENCES molde(nombre),
      FOREIGN KEY (componente) REFERENCES componente(id)
);
-- Tabla Cliente
CREATE TABLE cliente (
      nit VARCHAR(50) PRIMARY KEY,
      nombre VARCHAR(100) NOT NULL,
      direccion VARCHAR(100) NOT NULL
);
-- Tabla Computadora
CREATE TABLE computadora (
      id INT AUTO INCREMENT PRIMARY KEY,
      usuario VARCHAR(100),
      cliente VARCHAR(50),
      molde VARCHAR(100),
      estado VARCHAR(20) NOT NULL,
      precio venta DECIMAL(10,2),
      fecha ensamble DATE NOT NULL,
      fecha venta DATE,
      FOREIGN KEY (usuario) REFERENCES usuario(nombre),
      FOREIGN KEY (cliente) REFERENCES cliente(nit),
      FOREIGN KEY (molde) REFERENCES molde(nombre)
);
```