Documentación Kincaps (Sprint #1)

Descripción:

Kincaps es una aplicación web desarrollada para administrar una tienda virtual dedicada a la venta de gorras urbanas y deportivas. Se permite el registro e inicio de sesión de usuarios, gestión de productos, carritos de compras y facturación, con un backend en Java EE, JSP, Servlets, JPA/Hibernate y MySQL.

Arquitectura:

- Frontend: JSP con Bootstrap para diseño responsivo y estilo visual moderno.
- Backend: Java EE con Servlets y JSP para lógica de negocio y vistas dinámicas.
- Persistencia: JPA con Hibernate como proveedor ORM, conectado a base de datos MySQL.
- Servidor: GlassFish para despliegue y ejecución de la aplicación.
- Seguridad: Almacenamiento de contraseñas usando SHA-256 para hash de las contraseñas.
- Gestión de sesión: Para mantener estado de usuario tras autenticación.

Base de Datos:

Nombre: DB_KinCaps
Motor: MySQL
Tablas:

- cliente: Encargada de almacenar: (nombre, email, teléfono, dirección, contraseña hash).
- empleados: Almacena los datos de los empleados sus roles y fecha de contratación.
- proveedor: Posee la informacion sobre los proveedores de las gorras.
- gorras: El catálogo de los productos referenciando a los proveedores.
- carrito, detalleCarrito: Mueve las compras en proceso.
- factura, detalleFactura: Registra las compras completadas.

Procedimientos almacenados:

- sp_AgregarCliente, sp_AgregarEmpleado: Insertar nuevos usuarios con contraseñas cifradas.
- sp_Login: Validar credenciales y retornar tipo de usuario.

Seguridad:

Las contraseñas se cifran usando SHA – 256:

En la base de datos se guarda el hash hexadecimal de 64 caracteres.

La clase PasswordHash implementa esta función, pudiendo comparar las contraseñas ingresadas con las guardadas sin mostrar la contraseña real.

Usuario:

Registro:

- El usuario llena un formulario con datos personales.
- Se valida y se ejecuta el procedimiento sp_AgregarCliente para almacenar el cliente con la contraseña hasheada.

Inicio de Sesión:

Registro:

- El usuario llena un formulario con sus datos.
- Se valida y se ejecuta el procedimiento de agregar para guardar al cliente con la contraseña hasheada.

Inicio de Sesión:

- El usuario envía email y contraseña.
- Se invoca el procedimiento login que verifica si los datos concuerdan con algún cliente o empleado.
- Se inicia sesión y se crea una sesión HTTP con el objeto Usuario.
- Según el tipo de usuario, se dirigir cada uno a una página especifica ya sea cliente o empleado.

Interacción:

- Los clientes pueden navegar el catálogo, añadir sus compras al carrito y realizar la compra.
- Los empleados tienen una vista para gestionar productos y usuarios.

Estructura:

- **Paquete util:** Clase para hashing de contraseñas.
- **Servlets:** Controladores para manejar peticiones HTTP, como LoginServlet.
- **DAO:** Acceso a datos para verificar usuarios y realizar consultas a la base.
- **Modelo:** Clases Java que representan entidades como Cliente y Empleado.
- **JPA:** Clase para obtener EntityManager de JPA y manejar la conexión con la base.
- **JSP:** Formularios para login y registro, páginas de usuario y administrador con Bootstrap para UI.
- **Configuraciones:**
- **persistence.xml:** Configura la unidad de persistencia para Hibernate con conexión a MySQL.
- **Glassfish-web.xml:** Configura acciones específicas de GlassFish.
- **Pool JDBC:** Creado manualmente con comandos as admin para conexión eficiente a la base de datos.