**ARQUITECTURA DE SOFTWARE**

**Aplicación web para la gestión de ventas y control de inventarios y ventas de micromercados de ropa**

**JUAN DE DIOS SANTIAGO ESPINOSA**

**MIGUEL ANGEL PRADO PADILLA**

**JAIME MANUEL PÉREZ GUTIÉRREZ**

**UNIVERSIDAD POPULAR DEL CESAR**

**INGENIERÍAS Y TECNOLOGÍAS**

**ESPECIALIZACIÓN EN INGENIERÍA DE SOFTWARE**

**AGUACHICA CESAR**

**2023**

CONTENIDO

[**1. Propósito y alcance del proyecto 3**](#_heading=h.gkvda0aj6q3i)

[**2. ¿Qué arquitectura se va a implementar en el proyecto? 4**](#_heading=h.v1gjpep8ckdt)

[**3. ¿Por qué implementar la arquitectura de Tres Capas? 4**](#_heading=h.s8mnzxag82zj)

[**4. Diagrama de la arquitectura 5**](#_heading=h.l60iyf2869k0)

# **Propósito y Alcance del Proyecto**

Este documento contempla una propuesta con la descripción de los procedimientos, acciones a seguir y arquitectura de software en el desarrollo de una Aplicación Web para la gestión de inventarios y ventas en micromercados de ropa. De esta manera, facilitará el manejo y gestión de inventario, usuarios, clientes y ventas. Y con base a los resultados obtenidos generar informes estadísticos que faciliten a los vendedores la toma de decisiones.

La Aplicación Web brindará a los micromercados de ropa, la capacidad de gestionar eficazmente la información de los productos que venden. Además, permitirá la administración de datos de clientes y usuarios, el registro de ventas, gestión de los vendedores, así como la generación de informes y notificaciones relacionados con las ventas y el inventario de la empresa. Esta solución tecnológica actuará como una herramienta integral para simplificar y optimizar todas las operaciones críticas de estos micromercados, mejorando significativamente su eficiencia y rendimiento.

Con esta aplicación, los micromercados de ropa podrán hacer un seguimiento preciso de su inventario, lo que les permitirá tomar decisiones más informadas sobre qué productos mantener en stock y cuáles pueden necesitar reabastecimiento. Además, la capacidad de registrar ventas de manera eficiente y generar informes detallados les proporcionará una visión clara de su rendimiento financiero y las tendencias de ventas, lo que será invaluable para la toma de decisiones estratégicas.

La gestión de datos de clientes y usuarios será más sencilla, lo que permitirá un servicio más personalizado y una mayor retención de clientes. Además, la generación de notificaciones automáticas relacionadas con el inventario y las ventas ayudará a los micromercados a mantenerse al tanto de los eventos críticos y a tomar medidas oportunas cuando sea necesario.

En resumen, esta Aplicación Web está diseñada para fortalecer la capacidad de los micromercados de ropa para competir en un mercado dinámico y altamente competitivo, ofreciendo una solución integral que simplificará la gestión diaria y mejorará la experiencia tanto para los clientes como para los propios negocios.

# 

# **¿Qué arquitectura se va a implementar en el proyecto?**

La arquitectura seleccionada para el proyecto del software de gestión de ventas y control de inventarios en micromercados de ropa es la Arquitectura de Tres Capas. Esta elección se justifica por los numerosos beneficios que aporta a la eficiencia y adaptabilidad del sistema.

# **¿Por qué implementar la arquitectura de Tres Capas?**

La elección de una arquitectura de software de capas para un sistema de gestión de ventas y control de inventarios en micromercados de ropa se justifica por una serie de beneficios fundamentales. En primer lugar, esta arquitectura promueve la modularidad, que es esencial en un contexto donde la versatilidad y la adaptabilidad son críticas. La modularidad se logra al dividir el sistema en capas independientes, cada una con responsabilidades específicas. Esto permite que diferentes equipos de desarrollo trabajen en las capas respectivas, facilitando la colaboración y reduciendo el riesgo de conflictos en el código.

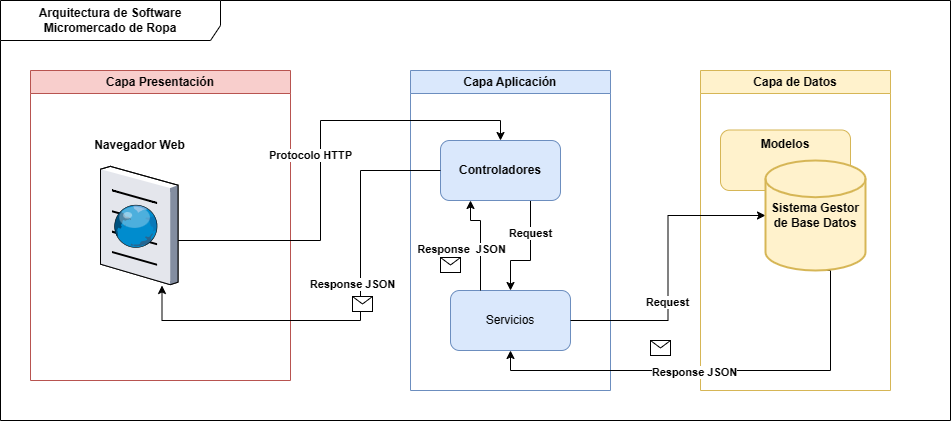
Otro beneficio significativo es la reutilización de código. Al dividir el sistema en capas, se crean componentes independientes que pueden ser utilizados en diversas partes del software. Esto ahorra tiempo y recursos, ya que no es necesario volver a escribir funciones comunes en múltiples secciones del programa. Además, al tener capas bien definidas, la mantenibilidad se ve mejorada. Cuando es necesario realizar cambios o correcciones, estos pueden llevarse a cabo de manera aislada en una capa específica sin afectar otras partes del sistema. Esto no sólo acelera el proceso de mantenimiento, sino que también reduce el riesgo de introducir nuevos errores en otras partes del sistema.

La separación de preocupaciones es otro aspecto crucial. Cada capa se enfoca en una funcionalidad específica, como la gestión de inventario, la gestión de ventas o la interfaz de usuario. Esta separación hace que el código sea más claro y manejable, lo que facilita la comprensión y el desarrollo continuo del software.

Además, las pruebas se vuelven más efectivas en una arquitectura de capas. Pueden realizarse pruebas unitarias y de integración en cada capa por separado, lo que facilita la detección temprana de errores y problemas de rendimiento. Finalmente, la seguridad se beneficia al aplicar medidas específicas en cada capa. Por ejemplo, la autenticación y el cifrado de datos pueden implementarse en la capa de aplicación, lo que protege de manera más efectiva la integridad de los datos y la privacidad de los usuarios.

En resumen, una arquitectura de capas proporciona una serie de ventajas significativas para un software de gestión de ventas y control de inventarios en micromercados de ropa. Estos beneficios incluyen modularidad, reutilización de código, mantenibilidad, separación de preocupaciones, escalabilidad, integración de terceros, pruebas efectivas y seguridad. Esta estructura optimiza la eficiencia y la adaptabilidad del software, lo que es esencial en un entorno empresarial en constante cambio y crecimiento

# **Diagrama de la arquitectura**



A continuación, se describe de forma detallada cómo es el flujo de información en la Arquitectura de Tres Capas para el presente proyecto

**Capa de Presentación**

**Componentes:**

Interfaz de Usuario (Frontend)

Peticiones HTTP (por ejemplo, utilizando JavaScript para realizar solicitudes AJAX o Fetch)

**Flujo de Información:**

El usuario interactúa con la Interfaz de Usuario (Frontend), por ejemplo, completando un formulario o haciendo clic en un botón.

El Frontend crea y envía una petición HTTP al servidor. Esta petición contiene la información ingresada por el usuario (por ejemplo, datos de un formulario) y la URL del recurso al que se quiere acceder en el servidor.

**Capa de Aplicación**

**Componentes:**

* Controladores
* Servicios

**Flujo de Información**

En el servidor, los Controladores reciben la petición HTTP del Frontend. Los Controladores son responsables de manejar las solicitudes y determinar qué acción se debe tomar.

Los Controladores pueden validar los datos recibidos del Frontend para asegurarse de que sean correctos y cumplan con las reglas de negocio. Si la validación es exitosa, el Controlador pasa la información a los Servicios.

Los Servicios contienen la lógica de negocio. En base a la información recibida, los Servicios realizan las operaciones necesarias. Por ejemplo, pueden crear, actualizar o consultar datos en la base de datos.

Si es necesario, los Servicios interactúan con la Capa de Datos para acceder a la base de datos. Esto implica utilizar modelos y consultas para obtener o modificar información.

Una vez completada la operación, los Servicios reciben la respuesta de la Capa de Datos y procesan cualquier resultado o error.

**Capa de Datos**

**Componentes:**

* Modelos (representación de la estructura de datos en la base de datos)
* Operaciones de Base de Datos

**Flujo de Información:**

La Capa de Datos recibe las solicitudes de los Servicios. Esto puede implicar consultar o modificar la base de datos según las instrucciones proporcionadas.

Utilizando Modelos, se crean consultas y operaciones de base de datos para realizar la acción solicitada.

La Capa de Datos ejecuta estas operaciones en la base de datos y recibe los resultados.

Si la operación fue exitosa, la Capa de Datos devuelve la respuesta al Servicio. Si hubo un error, también se notifica al Servicio.

**Devolución al Usuario**

**Flujo de Información**

El Servicio en la Capa de Lógica de Negocio recibe la respuesta de la Capa de Datos.

El Servicio procesa la respuesta y, si es necesario,realiza cualquier transformación o formato antes de enviarla de vuelta al Controlador.

El Controlador recibe la respuesta del Servicio y la envía de vuelta al Frontend a través de la respuesta HTTP.

Finalmente, el Frontend recibe la respuesta y la muestra al usuario, ya sea como una actualización de la interfaz de usuario o como un mensaje de éxito/fracaso, dependiendo de la operación realizada.