

Diseño arquitectónico de una aplicación



Ismael Ramos Alonso

Introducción

El diseño arquitectónico de una aplicación no se refiere simplemente a cómo se ve o se siente al usuario, sino a cómo está estructurada y cómo sus componentes interactúan entre sí. Es una disciplina que se centra en definir la estructura general, los componentes, las relaciones y las propiedades de esos componentes. Esta estructura debe ser capaz de satisfacer los requisitos técnicos y funcionales, al tiempo que garantiza la escalabilidad, la seguridad y el rendimiento.

El objetivo principal es llevar a cabo el diseño básico de una aplicación móvil empleando un patrón de diseño estructural. Esto implica que nos centraremos en establecer la sólida base sobre la cual se construirá la aplicación, definiendo la disposición e interacción de sus elementos fundamentales.

1.- Definición del dominio

El dominio de nuestra aplicación se centra en un mercado digital que promueve y conecta la artesanía canaria con los usuarios interesados. Los Usuarios Básicos disfrutan de una experiencia de compra y colaboración única, mientras que los Usuarios Artesanos obtienen las herramientas necesarias para gestionar sus negocios y llegar a un público más amplio. Esta plataforma busca preservar y promover la rica tradición artesanal de las Islas Canarias, al tiempo que fomenta la economía local y la apreciación por la artesanía auténtica.

Dentro de este dominio, hemos definido una serie de funcionalidades clave para dos tipos de usuarios: los Usuarios Básicos y los Usuarios Artesanos.

Funcionalidades para el Usuario Básico:

- **Búsqueda Avanzada:** Los usuarios pueden realizar búsquedas de artículos y artesanos utilizando la sección de búsqueda.
- **Exploración de Categorías:** Los usuarios pueden explorar y navegar a través de diversas categorías de artículos para encontrar productos de su interés.
- **Gestión de Favoritos:** Los usuarios pueden guardar artículos y artesanos en su lista de favoritos para un acceso rápido y conveniente.

- **Compra de artículos:** Los usuarios pueden comprar artículos a los artesanos.
- **Realización de pedidos:** Los usuarios pueden realizar pedidos públicos de artículos, especificando los detalles que crean convenientes. Además, podrán seleccionar la oferta que consideren más convenientes entre las propuestas por los artesanos.
- **Gestión de Perfil:** Los usuarios pueden editar su perfil, actualizar información personal y gestionar la configuración de su cuenta.

Funcionalidades para el Usuario Artesano:

Además de las funcionalidades mencionadas anteriormente, el usuario artesano tiene acceso a herramientas adicionales para gestionar su negocio de manera eficiente.

- **Búsqueda de Pedidos:** Los artesanos pueden buscar y visualizar pedidos realizados por usuarios que necesitan servicios de artesanía.
- **Realizar ofertas de servicio:** Los artesanos pueden responder a los pedidos de los usuarios ofreciendo sus servicios y estableciendo acuerdos.
- **Gestión de Productos:** Los artesanos pueden agregar, editar y eliminar productos de su perfil, especificando detalles como imagen, precio, descripción y disponibilidad.
- **Seguimiento de Ventas y Pedidos:** Los artesanos tienen acceso a un registro completo de sus ventas y pedidos, incluyendo tanto los pedidos aceptados como los pendientes de confirmación

2.- Identificación de los principales elementos de la arquitectura elegida

Se ha seleccionado la arquitectura hexagonal, la cual es un enfoque de diseño que se centra en la separación clara de las diferentes capas y componentes de una aplicación. En este enfoque, se identifican varios "puertos" a través de los cuales la aplicación interactúa con el mundo exterior, como el acceso a datos, servicios externos e interfaces de usuario. Estos puertos actúan como puntos de entrada y salida de la aplicación, y están conectados a través de "adaptadores" que facilitan la comunicación entre la aplicación y estos diferentes contextos.

Elementos de la arquitectura hexagonal

La arquitectura hexagonal comprende tres componentes centrales:

- **Modelo de Dominio:** representa el núcleo de la aplicación de software y encapsula las reglas comerciales y la lógica central que definen el comportamiento de la aplicación.
- **Puertos:** representan conductos que conectan el modelo de dominio con el mundo exterior. Los puertos primarios exponen la lógica empresarial de la aplicación y sirven como puerta de entrada al núcleo de la aplicación.
- **Adaptadores:** actúan como traductores entre el modelo de dominio y el mundo externo. Su función principal es convertir datos entre el formato utilizado por los sistemas externos y el empleado por la lógica empresarial. Estos adaptadores facilitan la interacción fluida entre la aplicación y los diversos componentes externos, garantizando una comunicación eficiente en todas las direcciones.

A continuación, destacamos los principales puertos que la aplicación requerirá, junto con los adaptadores correspondientes:

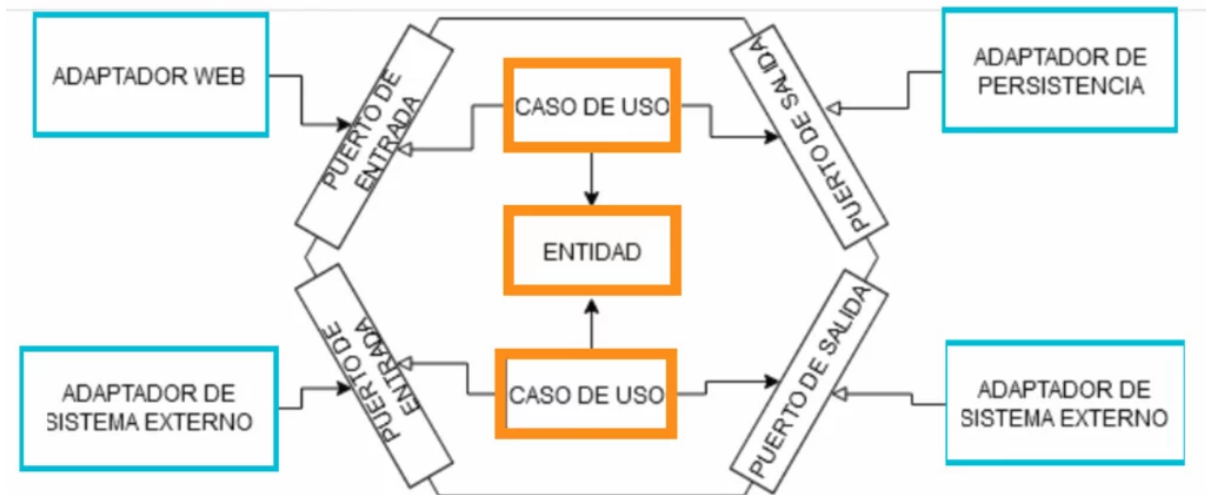
- **Puerto de Acceso a la Base de Datos:** Este puerto actúa como la interfaz que permite a nuestra aplicación interactuar con la capa de almacenamiento de datos. Proporciona una abstracción de acceso a la base de datos subyacente, lo que nos permite cambiar la tecnología de la base de datos sin afectar a otras partes de la aplicación. El adaptador correspondiente a este puerto se encargará de implementar las operaciones de lectura y escritura en la base de datos, manteniendo la lógica de negocio aislada de los detalles de la persistencia.

- **Puerto de Acceso a Servicio Externo de Pago:** En esta arquitectura, es esencial separar la lógica relacionada con los servicios externos de pago. El puerto de acceso a servicios externos de pago actúa como una interfaz que permite a la aplicación comunicarse con proveedores de servicios de pago externos. Los adaptadores vinculados a este puerto se encargan de la implementación específica de la interacción con cada proveedor de pago. Esto garantiza que podamos cambiar o agregar nuevos proveedores de pago sin modificar el núcleo de nuestra aplicación.

Estos puertos y adaptadores son componentes cruciales de la arquitectura hexagonal, ya que promueven la flexibilidad y la modularidad al aislar las dependencias externas y permitir que la lógica de la aplicación se centre en sus objetivos principales. Esta estructura facilita la adaptación a cambios futuros, la prueba unitaria y la expansión de la aplicación sin comprometer su estabilidad y mantenibilidad.

3.- Diseño de la Arquitectura

A continuación, se muestra un diagrama de la arquitectura hexagonal con sus principales elementos:



4.- Casos de Uso

1. Buscar Producto:

El usuario desea buscar un producto específico en la aplicación para ver sus detalles y precio.

Procesamiento a través de la arquitectura hexagonal:

- **Interfaz de Usuario (UI):** El usuario introduce el nombre o parte del nombre del producto en un campo de búsqueda.
- **Adaptador UI:** Traduce la acción del usuario en una llamada al puerto correspondiente del hexágono.
- **Puerto de Búsqueda de Producto:** Una interfaz dentro del hexágono que define cómo se puede buscar un producto.
- **Lógica de Negocio:** Se ejecuta la lógica para buscar el producto en la base de datos o en memoria.
- **Adaptador de Base de Datos:** Traduce las llamadas de la lógica de negocio en consultas específicas para la base de datos.
- **Base de Datos:** Devuelve los detalles del producto buscado.
- **La respuesta viaja en sentido inverso:** del adaptador de base de datos al adaptador UI, y finalmente se muestra el resultado al usuario en la interfaz.

2. Realizar Compra:

Una vez que el usuario ha encontrado un producto que desea comprar, procede a realizar la compra.

Procesamiento a través de la arquitectura hexagonal:

- **Interfaz de Usuario (UI):** El usuario selecciona el producto y pulsa el botón "Comprar".
- **Adaptador UI:** Traduce la acción del usuario en una llamada al puerto correspondiente del hexágono.
- **Puerto de Realizar Compra:** Una interfaz dentro del hexágono que define cómo se puede realizar una compra.
- **Lógica de Negocio:** Se ejecuta la lógica para:
 - Verificar la disponibilidad del producto.
 - Calcular el precio total.
 - Procesar el pago.
 - Actualizar el inventario.
- **Adaptador de Base de Datos:** Traduce las llamadas de la lógica de negocio en consultas específicas para la base de datos.
- **Base de Datos:** Actualiza la información relevante y devuelve una confirmación.
- **Adaptador de Pago:** Si la aplicación integra un servicio de pago externo, este adaptador traduce las llamadas de la lógica de negocio en solicitudes específicas para el servicio de pago.
- **Servicio de Pago:** Procesa el pago y devuelve una confirmación.

5.- Conclusión

La elección de la arquitectura adecuada para el diseño de una aplicación es crucial para su éxito y mantenimiento a largo plazo. En este sentido, hemos optado por la arquitectura hexagonal por las ventajas que nos ofrece.

En primer lugar, la independencia de la arquitectura hexagonal es fundamental. La capacidad de mantener las diferentes capas del software aisladas e independientes garantiza que cualquier cambio o actualización pueda realizarse sin riesgos significativos de interrupción en la experiencia del usuario. En una plataforma de comercio electrónico, donde la fluidez de las transacciones y la satisfacción del cliente son imperativas, esta independencia es crucial.

La flexibilidad de la arquitectura hexagonal también se adapta perfectamente a las necesidades cambiantes de una aplicación de compra y venta de artesanías. Al permitir que el núcleo del programa permanezca constante mientras se intercambian otros componentes, la arquitectura hexagonal evita la dependencia de tecnologías específicas y facilita la adaptación a nuevos requisitos o la incorporación de herramientas innovadoras. Esto es especialmente relevante en un mercado donde los vendedores y los productos pueden variar considerablemente.

La facilidad de realizar pruebas unitarias es crucial para garantizar la calidad y la confiabilidad de una aplicación de comercio electrónico. La arquitectura hexagonal proporciona un entorno propicio para pruebas automatizadas de la lógica de negocio, lo que es fundamental para asegurar que las transacciones se realicen sin problemas y que los usuarios tengan una experiencia satisfactoria.

La escalabilidad es otra ventaja significativa de la arquitectura hexagonal. La modularidad y la independencia de componentes permiten que la aplicación crezca tanto en capacidad como en funcionalidad de manera eficiente, lo que es esencial para adaptarse a un mercado en constante evolución.

En resumen, la arquitectura hexagonal se destaca como la elección ideal para una aplicación móvil de compra y venta de artesanías debido a su independencia, flexibilidad, facilidad de pruebas, escalabilidad y capacidad para adaptarse a las cambiantes demandas del mercado.

6.- Bibliografía:

Arquitectura Hexagonal. Becas Santander.

<https://www.becas-santander.com/es/blog/arquitectura-hexagonal.html> [en línea] 04/10/23

Introducción a la Arquitectura Hexagonal. CloudAPPi..

<https://cloudappi.net/introduccion-a-la-arquitectura-hexagonal/> [en línea] 05/10/23