

# **PRUEBA TÉCNICA**

**Integrantes:**  
**Michael Vargas**

**Presentado a:**  
**Colombia Compra Eficiente**

**17 / Noviembre / 2021**  
**Bogotá D.C.**

## Tabla de contenido

1.	Introducción .....	3
1.1	Objetivo .....	3
1.2	Alcance .....	3
1.3	Definiciones, Acrónimos y Abreviaturas .....	3
2.	Vista Lógica .....	4
2.1	Diagrama de Clases: .....	4
2.2	Vista de Procesos .....	6
3.	Vista de Desarrollo .....	7
3.1	Patrones .....	8
4.	Vista de Física .....	9
5.	Tecnologías de implementación .....	10
6.	Referencias .....	10

## Ilustraciones

Ilustración 1	Modelo de vista arquitectónica 4+1 [1] .....	3
Ilustración 2	Diagrama de Entidad Relación .....	4
Ilustración 3	Diagrama de Procesos .....	6
Ilustración 4	Diagrama de Componentes .....	7
Ilustración 5	Diagrama de Despliegue .....	9

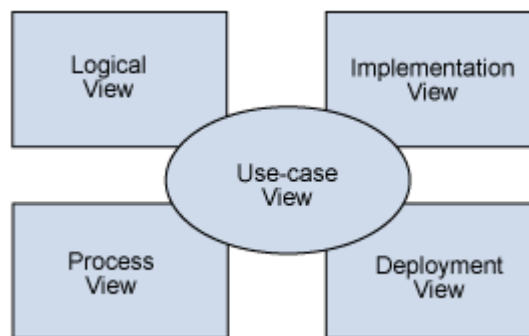
# 1. Introducción

Con el objetivo de mostrar la aplicación de patrones de software, en este documento se hará uso de diferentes diagramas para enseñar la complejidad de los diferentes aspectos que encierra la arquitectura de la solución propuesta, para el ejercicio del desarrollar la prueba técnica en la Agencia Nacional de Contratación Pública – Colombia Compra Eficiente (ANCP-CCE).

## 1.1 Objetivo

El presente SAD proporciona una visión general arquitectónica de la herramienta E-Procurement, que puede ser utilizada por ANCP-CCE y sus usuarios.

Para describir la solución con la mayor precisión posible, la estructura de este documento se basa en el modelo de arquitectura 4+1 que se puede observar en la *Ilustración 1*.



*Ilustración 1 Modelo de vista arquitectónica 4+1 [1]*

## 1.2 Alcance

Esta solución se restringe al registro de usuarios y creación de órdenes de compra, junto con la representación de la plataforma E-Procurement.

## 1.3 Definiciones, Acrónimos y Abreviaturas

- **UML:** Unified Modeling Language

## 2. Vista Lógica

A continuación, se muestran las vistas correspondientes al diseño arquitectural de la aplicación.

### 2.1 Diagrama de Clases:

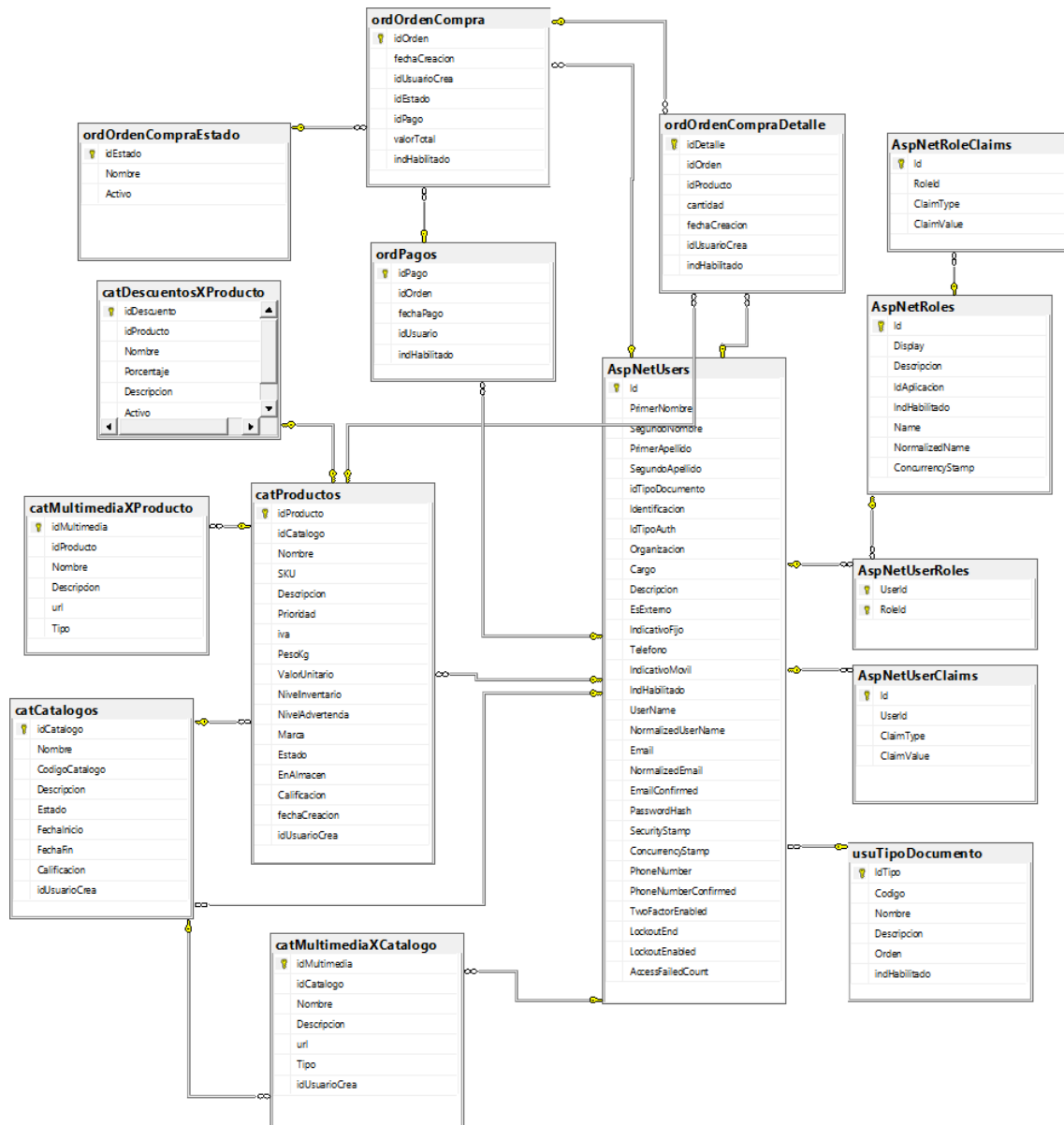


Ilustración 2 Diagrama de Entidad Relación <sup>1</sup>

En la Ilustración 2 se muestra el diagrama entidad relación contenido de la siguiente manera:

- **Autenticación:** Encierra las entidades que almacenan y la información de autenticación. Las tablas son las siguientes:

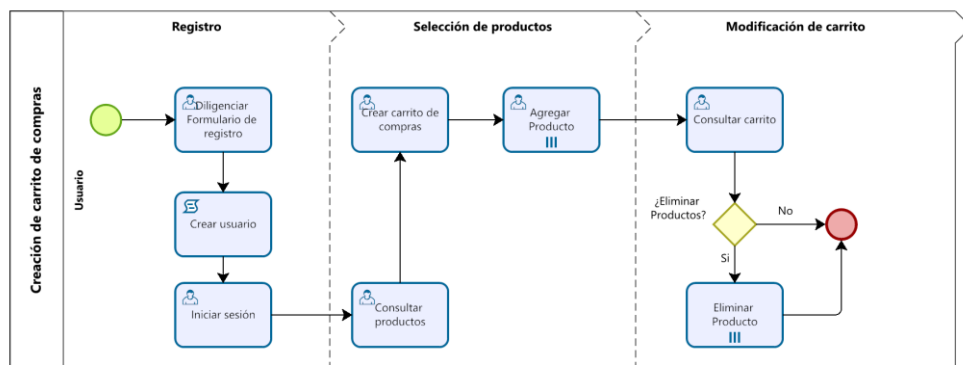
- *AspNetAlgoritmosDeSeguridad*
- *AspNetAplicacion*
- *AspNetRoleClaims*
- *AspNetRoles*
- *AspNetTiposAutenticacion*
- *AspNetUserClaims*
- *AspNetUserLogins*
- *AspNetUserRoles*
- *AspNetUsers*
- *AspNetUserTokens*

- **Catálogos:** Contiene las tablas que almacenan la información correspondiente a los catálogos y productos:
  - *catCatalogos:*
  - *catDescuentosXProducto*
  - *catMultimediaXCatalogo*
  - *catMultimediaXProducto*
  - *catProductos*
- **Órdenes de Compra:** En este segmento se encierran las tablas que almacenen información sobre las ordenes de pago:
  - *ordOrdenCompra*
  - *ordOrdenCompraDetalle*
  - *ordOrdenCompraEstado*
  - *ordPagos*

## 2.2 Vista de Procesos

En la ilustración 3 se observa el diagrama de procesos que se identificó para la arquitectura de la solución propuesta.

El proceso inicia con el usuario registrándose y luego iniciando sesión. Una vez hecho esto el usuario consulta los productos y crea el carrito de compras. A continuación el usuario selecciona los productos e indica la cantidad que desea ordenar. Para finalizar el proceso el usuario puede modificar el carrito eliminando los elementos agregados o finalizar la selección de productos.



Powered by  
IBM  
Modeler

Ilustración 3 Diagrama de Proceso

### 3. Vista de Desarrollo

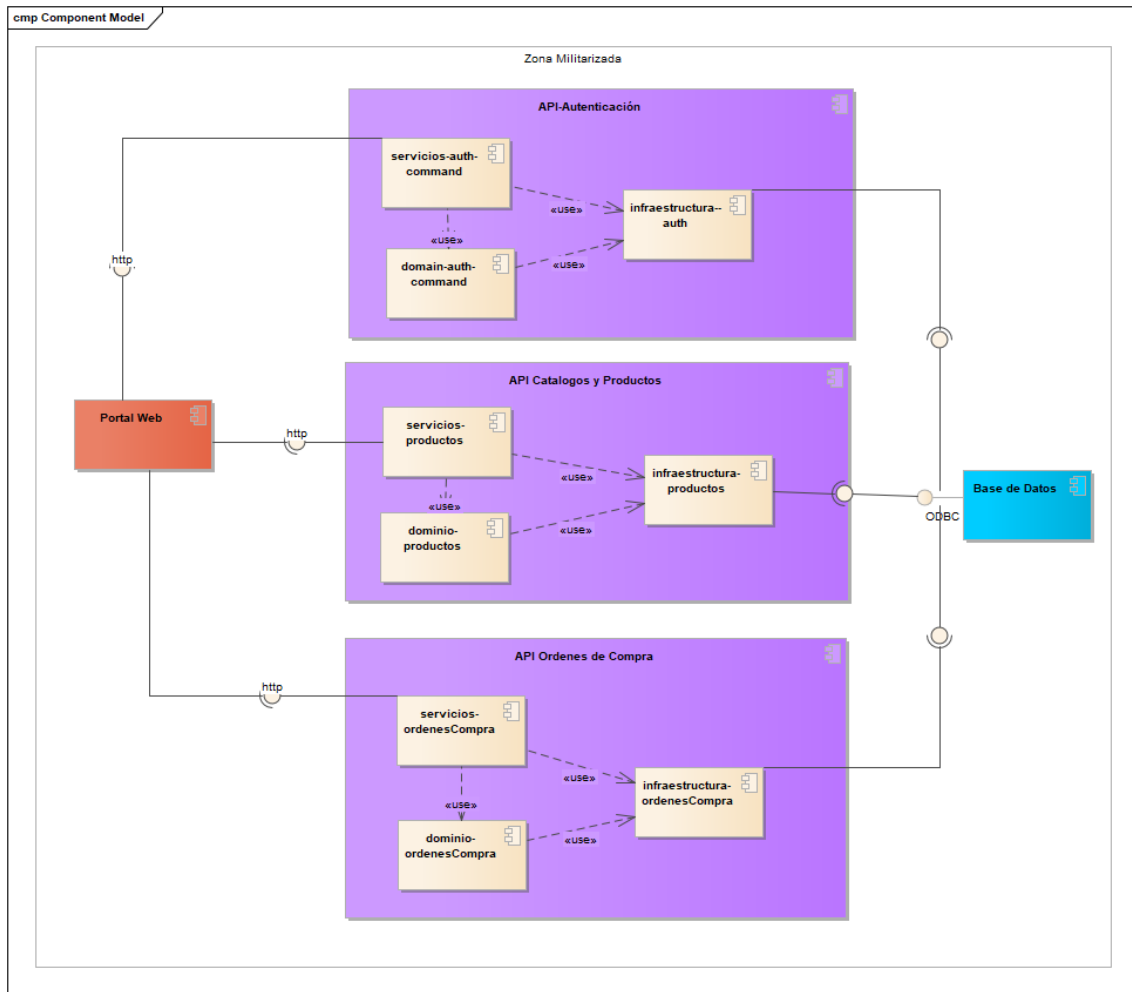


Ilustración 4 Diagrama de Componentes <sup>2</sup>

En la *Ilustración 4* se puede ver el diagrama de componentes del sistema. Este incluye los siguientes componentes:

- **Portal Web**  
Este componente abarca los subcomponentes encargados de la presentación, estructura de datos y controladores de comunicación para presentar la información por medio de un portal web. Consume los servicios web que exponen en cada una de las API.
- **API Autenticación**  
Esta API expone servicios para crear y autenticar usuario de la aplicación. Mantiene la estructura estándar definida para cada una de los componentes Backend.
- **API Catálogos y Productos**  
Este componente expone servicios de consulta para ver catálogos y productos.

- **API órdenes de Compra**

Esta API provee servicios para consultar y realizar operaciones CRUD sobre la información de las órdenes de compra de cada usuario.

### 3.1 Patrones

En el diseño de implementación de la solución se tuvieron en cuenta los siguientes patrones:

- **BFF (Building a Backend for Frontend)**: Usado en la arquitectura de diseño definiendo componentes de servicios web y portal web.
- **DDD (Domain Driven Design)**: Este patrón se aplica en cada uno de los componentes que exponen servicios web para el portal web. Se contemplan los dominios *API*, *Dominio e Infraestructura*.
- **UnitOfWork**: Usado para centralizar los recursos de repositorio en una sola entidad accesible para los componentes que contienen la lógica de negocio.
- **Repository**: Este patrón ha sido usado en el dominio de Infraestructura como recurso de operaciones genéricas para acceder a la información persistida.
- **Specification**: Se usa para diseñar y definir las reglas comerciales del negocio al momento de interactuar con los datos.

Cabe aclarar que toda la solución sigue el estilo arquitectural Cliente/Servidor.



## 4. Vista de Física

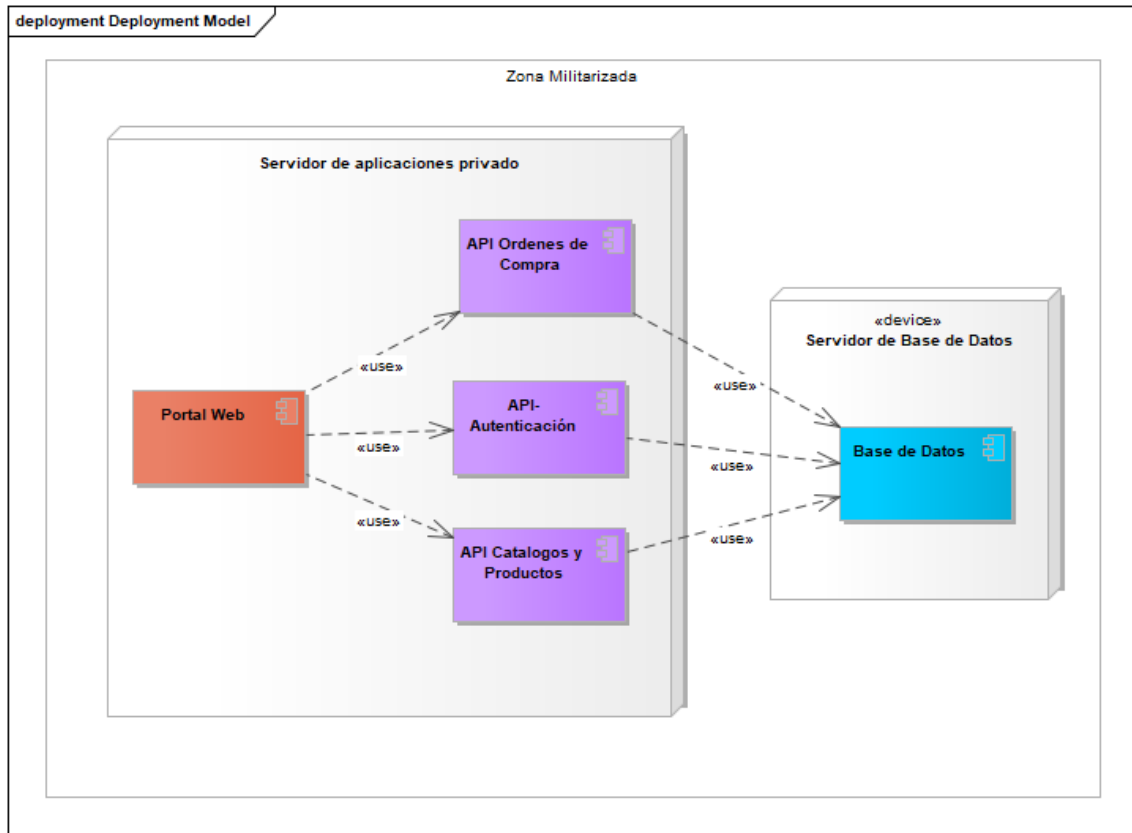


Ilustración 5 Diagrama de Despliegue <sup>3</sup>

En la Ilustración 5 se muestra el diagrama de despliegue con los siguientes nodos:

- **Servidor de aplicaciones privado**  
Este nodo contiene una unidad de despliegue IIS contenida de los componentes *Portal Web*, *API Órdenes de compra*, *API Autenticación*, *API Catálogos*. Su función es procesar y alojar la lógica y modelo del negocio, junto a la persistencia de los datos y la integración con sistemas externos.
- **Servidor de Base de datos**  
Este nodo contiene la base de datos que persiste la información del sistema.

## 5. Tecnologías de implementación

La solución se ha implementado en las siguientes tecnologías:

- **ASP.NET Core 3.1:**  
Se implementaron las API que exponen servicios REST. En los diagramas se pueden ver de color **morado**. Cada componente tiene el Swagger que identifica los servicios que está exponiendo.
- **Angular 13:**  
Se usó para implementar el portal web del sistema el cual presenta la interfaz gráfica y consume los servicios web. Se puede ver de color **rojo** en los diagramas.
- **SQL Server:**  
Se usó para desarrollar la base de datos que persistirá la información del sistema. En los diagramas se ilustra de color **azul**.

## 6. Referencias

- [1] P. Kruchten, «Planos Arquitectónicos: El Modelo de “4+1” Vistas de la Arquitectura del Software», p. 16.