

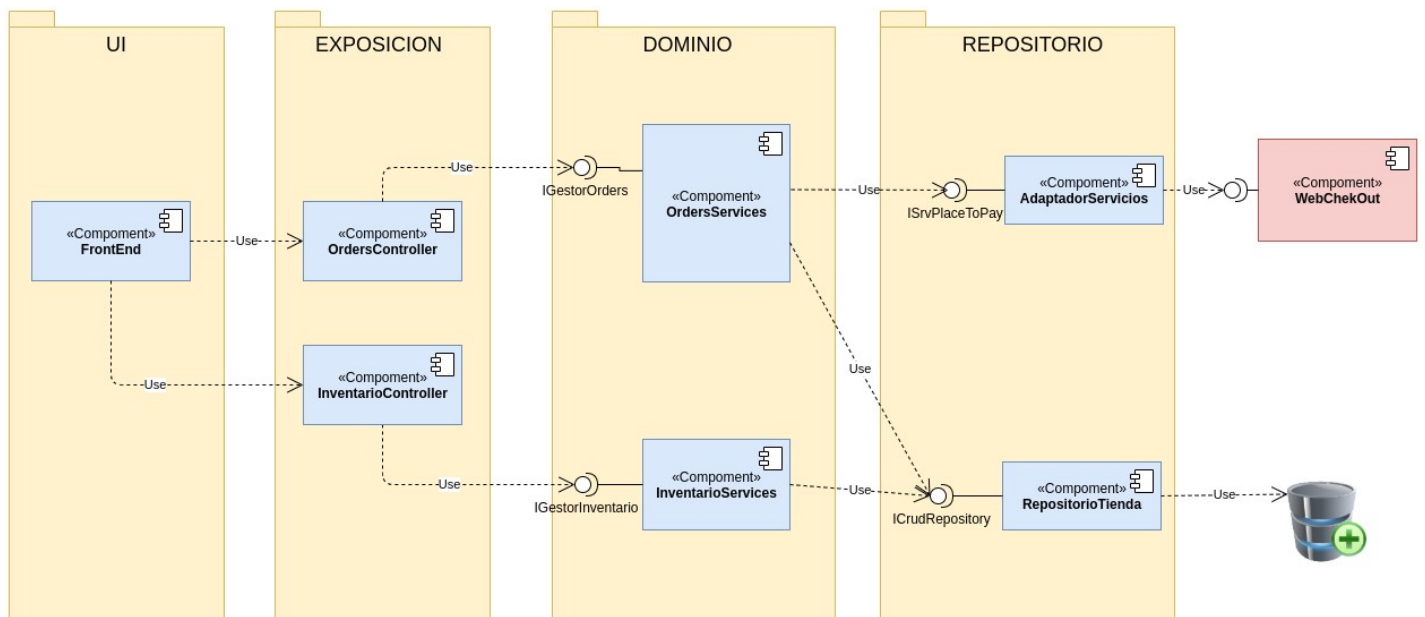
DOCUMENTO TÉCNICO DE SOLUCIÓN DE CHALLENGE

Título:	Prueba de Ingreso Ingeniero de Desarrollo
Solicitado por:	EVERTEC
Solucionado por:	Oscar Martín Bohórquez Rojas
Fecha:	23-Agosto-2021

1. CONTEXTO SOLUCIÓN

Como base de la arquitectura se ha utilizado el estándar o patrón de Arquitectura Hexagonal para el diseño del microservicio y para modelar la lógica del negocio Domain Driven Design (DDD).

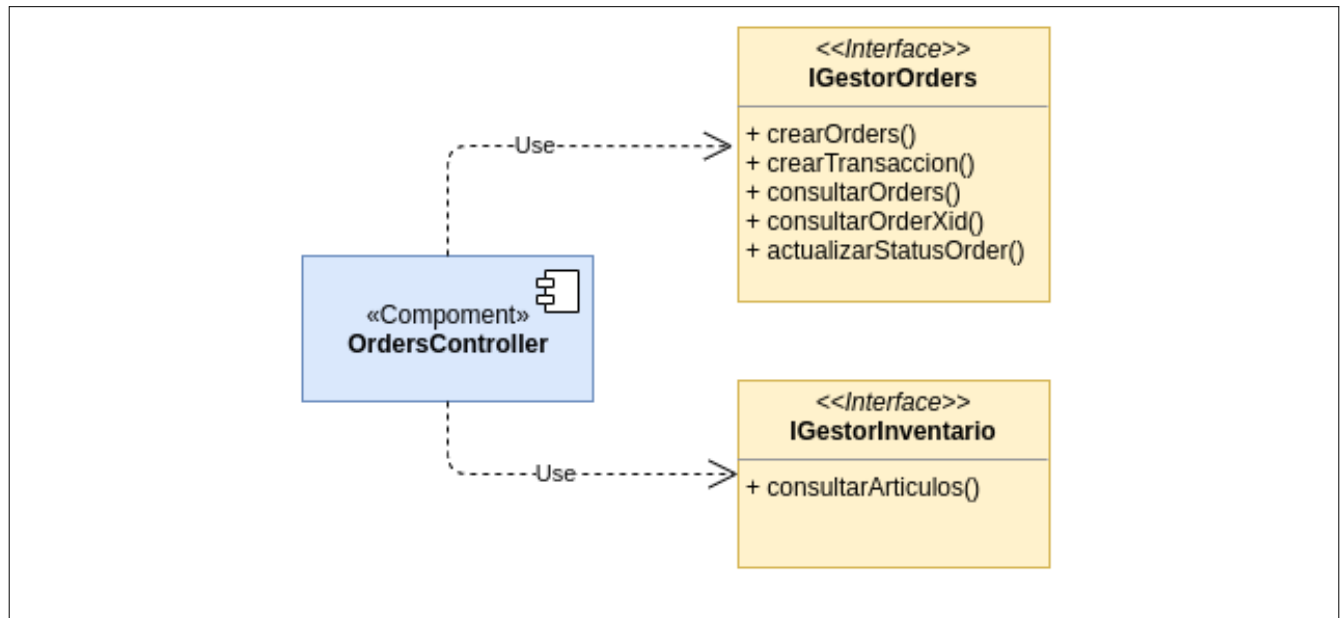
2. MODELO DE COMPONENTES DE ALTO NIVEL



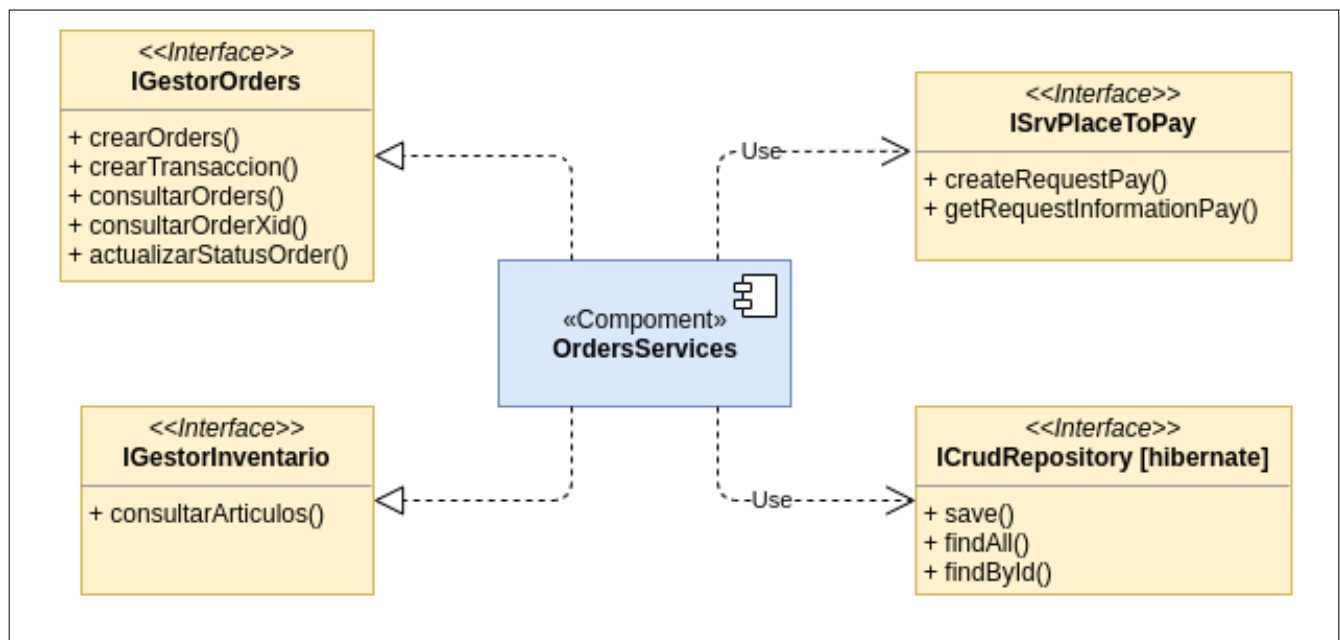
Nombre	Responsabilidades
UI	<ul style="list-style-type: none"> • Interfacez de Usuario (paginas y formularios) para interactuar con la Tienda Pago ONLINE. Comprende las vistas de: <ul style="list-style-type: none"> ◦ Captura Orden ◦ Consulta de Orden Pre Pagar ◦ Consulta de Orden Post Pago ◦ Consulta de Ordenes
OrdersController	<ul style="list-style-type: none"> • Exponer la capacidad (servicio REST) sobre HTTP para crear, modificar y consultar Ordenes. También permite crear un Pago de la Orden • Usar la capa de Dominio para gestionar las Ordenes.
InventarioController	<ul style="list-style-type: none"> • Exponer la capacidad (servicio REST) sobre HTTP para consultar Artículos • Usar la capa de Dominio para consultar los Artículos
OrdersServices	<ul style="list-style-type: none"> • Exponer la interfaz IGestorOrders, que permite gestionar el procesamiento de una Orden y su respectivo Pago. • Usar la interfaz ISrvPlaceToPay del adaptador de los servicios para utilizar/consumir el servicio WebCheckout PlaceTopay • Usar la interfaz ICrudRepository de capa de infraestructura o repositorio para persistir la información de la Orden y la Pasarela de Pago.
InventarioServices	<ul style="list-style-type: none"> • Exponer la interfaz IGestorInventarios, que permite consultar los artículos •
AdaptadorServicios	<ul style="list-style-type: none"> • Expone la interfaz ISrvPlaceToPay para llamar a los servicios de WebCheckout PlaceToPay para iniciar una pasarela de pagos.
RepositorioTienda	<ul style="list-style-type: none"> • Persistir los datos de la Orden y el Pago • Su implementación es basada en Hibernate.

2. DETALLE DE PRINCIPALES COMPONENTES

2.1 OrdersController (capa de exposición) - Diagrama de Interfaces Provistas y Requeridas



2.2 OrdersServices (capa de dominio) - Diagrama de Interfaces Provistas y Requeridas



3. INTERACCIÓN DE COMPONENTES

Principales interacciones y operaciones de cada uno de los componentes

