



UNIVERSIDAD ABIERTA INTERAMERICANA
FACULTAD DE TECNOLOGÍA INFORMÁTICA
MATERIA: SEMINARIO DE APLICACIÓN PROFESIONAL
1er Parcial
Teórico y Práctico

Alumno: Krauss Nahuel

Fecha: 19/05/2025

Tema 1

Comisión – Localización – Turno: 5°A – Norte – Noche

Práctica:

Teoría:

Nota:

Temas para evaluar: Procesos y productos para la creación de software. Análisis y diseño de una aplicación web. Arquitectura de base. POO, Conocimientos de bases de datos. Tecnologías web. Patrones de diseño. Modelado.

Objetivos:

- Comprender las pautas del proceso para la creación de software desde la perspectiva de la Ing. SW.
- Comprender cómo se desarrolla una aplicación con características ingenieriles.

Modalidad: Parcial domiciliario

Requisitos para aprobar: Para que el parcial esté aprobado el alumno deberá tener correctamente desarrolladas el 60% de la teoría y resuelto el ejercicio práctico.

Tiempo:

Recomendaciones:

- a) Lea todo el parcial antes de comenzar a responder.
- b) Desarrolle una redacción clara y precisa contestando lo que la pregunta requiere.
- c) Observe la ortografía ya que la misma es parte del parcial.
- d) Si considera que no comprende alguna consigna antes de comenzar consulte a su profesor.

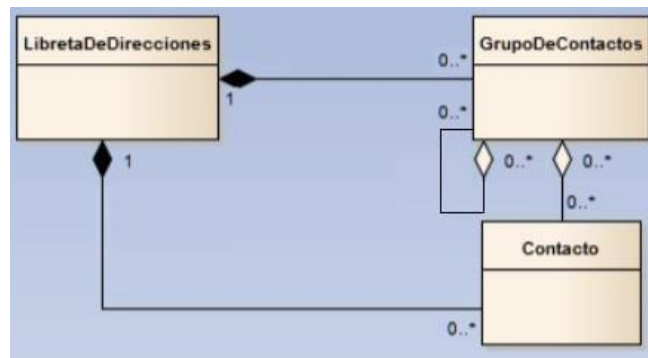
Notas: Las preguntas en las que se seleccionen opciones se deberá optar solo por una de las posibilidades. La indicación se efectuará con una X sobre su lateral izquierdo

Las preguntas que solicitan justificación serán consideradas válidas si poseen la misma correctamente.

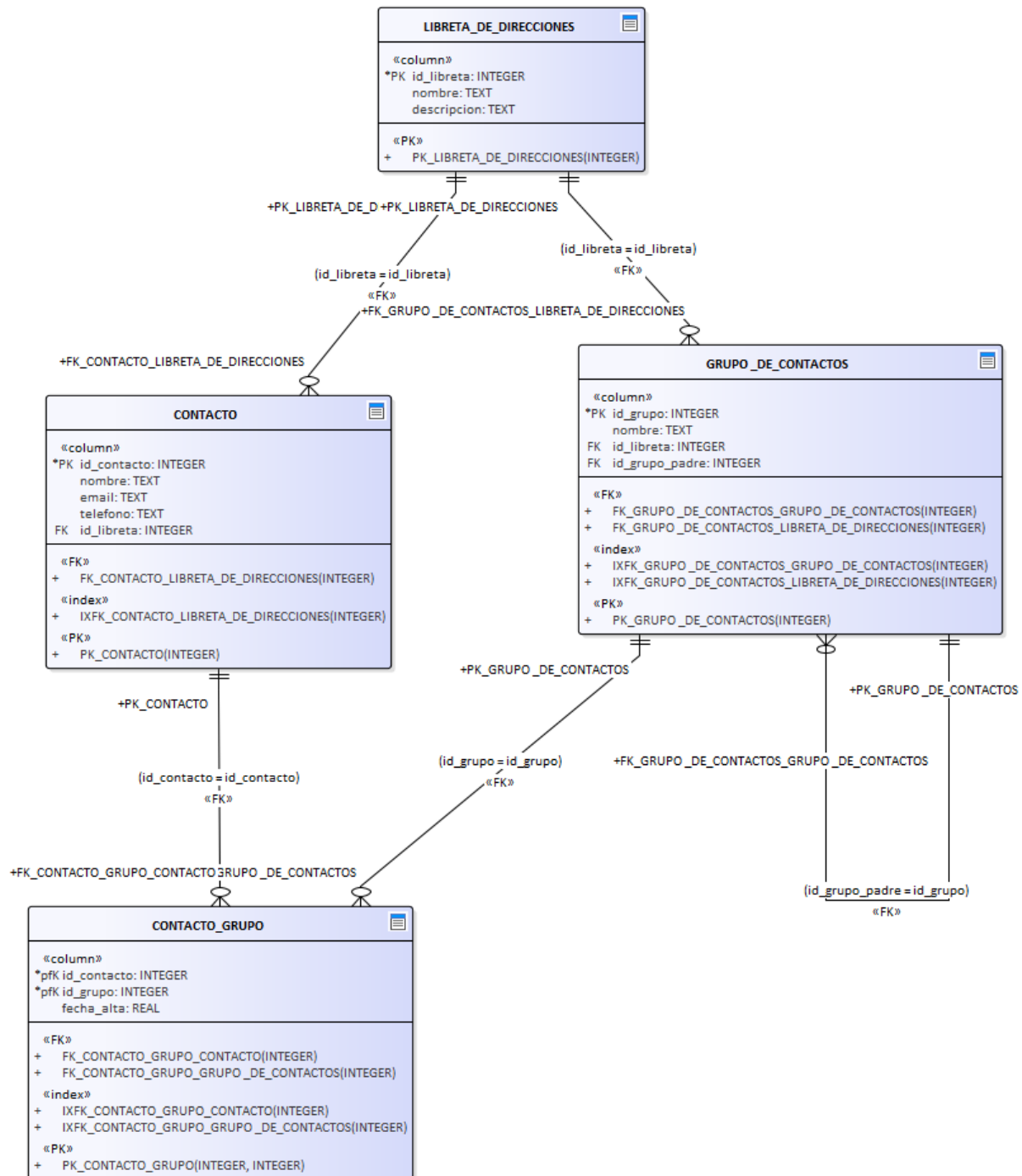
Las preguntas de múltiples posibilidades y verdadero / falso restan 0.50 puntos en caso de estar mal contestadas. En las preguntas verdadero / falso se debe tachar la opción incorrecta.

(*) la cifra entre paréntesis en cada pregunta es la cantidad de puntos sobre 100.

1. Nos solicitan desarrollar un DER a partir del siguiente diagrama de clases (Agregue los atributos que considere necesarios al modelo): (50) (Pegar directamente la imagen DER generada con la herramienta que usted desee: por ejemplo draw.io, E.A, etc.)



dm DER





UNIVERSIDAD ABIERTA INTERAMERICANA
FACULTAD DE TECNOLOGÍA INFORMÁTICA
MATERIA: SEMINARIO DE APLICACIÓN PROFESIONAL
1er Parcial
Teórico y Práctico

2. ¿Cómo podría relacionar la plataforma Auth0 (Similitudes/diferencias) con el concepto del patrón (Composite) Usuario-Patente-Familia? (50)

La relación entre Auth0 y el patrón Composite, aplicado al modelo Usuario-Patente-Familia, puede abordarse desde diversas perspectivas que combinan similitudes y diferencias en la gestión jerárquica y estructural de entidades. Auth0 administra identidades y permisos mediante roles, grupos o políticas que pueden heredarse, ofreciendo una organización jerárquica similar a cómo el patrón Composite estructura elementos individuales (como una patente) dentro de composiciones más amplias (familias). Así como Composite facilita el tratamiento uniforme de objetos simples y compuestos, Auth0 unifica el proceso de autenticación y autorización independientemente del nivel de complejidad del recurso, sea una aplicación individual, una API o un ecosistema completo de microservicios.

Esta similitud también puede observarse en la existencia de una interfaz común: el patrón Composite utiliza una interfaz que define operaciones homogéneas para hojas y compuestos, mientras que Auth0 emplea tokens estandarizados (por ejemplo, JWT) como interfaz única para autenticar usuarios y asignar scopes que determinan permisos. Estos tokens pueden aplicarse tanto a recursos individuales como a conjuntos o grupos de recursos, proporcionando una experiencia coherente y simplificada desde el punto de vista del acceso.

Otra característica común es la capacidad de extensión. Auth0 admite agregar funcionalidades mediante mecanismos como rules o hooks, que permiten modificar o extender los flujos de autenticación sin alterar la estructura central, algo parecido al enfoque del patrón Composite, que permite integrar nuevos componentes como subfamilias o patentes adicionales sin afectar la lógica global ya existente. Además, el Marketplace de Auth0 posibilita la integración con soluciones externas, actuando como componentes adicionales que enriquecen el sistema sin modificar la estructura interna.

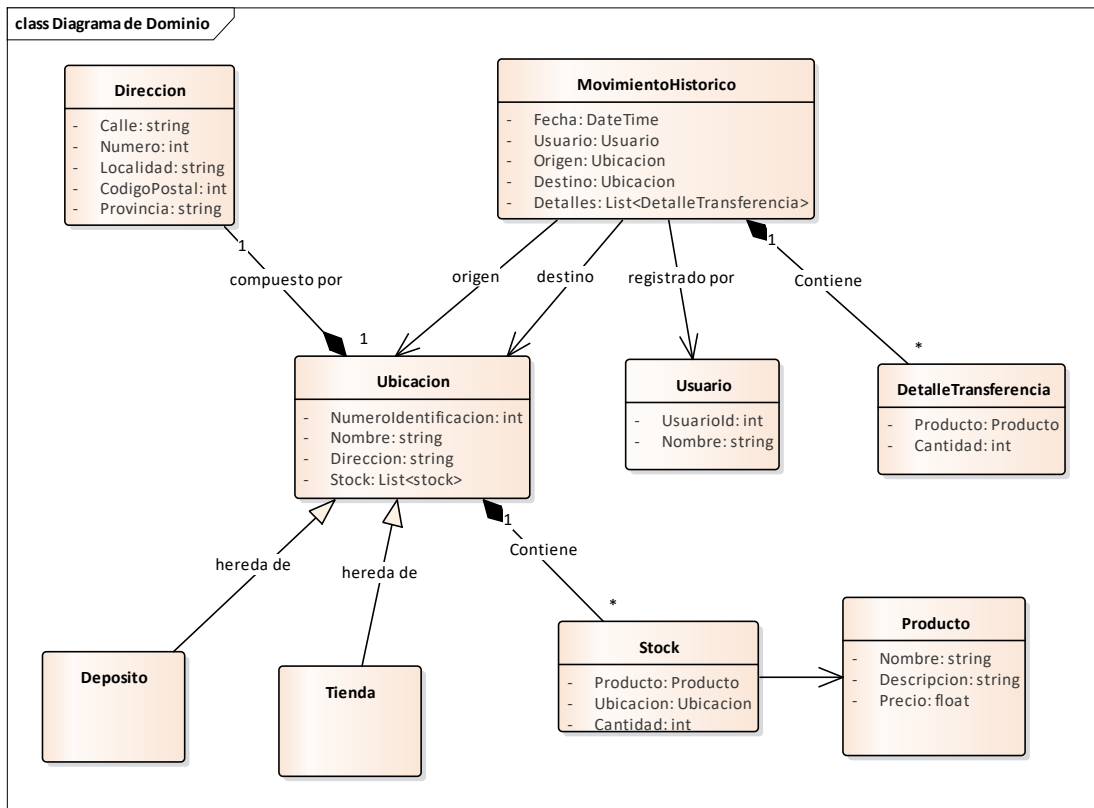
Sin embargo, a pesar de estas similitudes, ambos enfoques poseen diferencias significativas en objetivos y aplicaciones prácticas. Auth0 está orientado principalmente hacia la gestión de identidad y acceso (IAM), asegurando la seguridad y cumplimiento normativo mediante procesos centralizados de autenticación y autorización. En contraste, el patrón Composite se centra en modelar relaciones parte-todo, optimizando operaciones jerárquicas sobre estructuras complejas, como la gestión unificada de patentes y familias para tareas específicas como el cálculo de royalties.

Asimismo, la representación de datos en ambos casos difiere notablemente. En Composite, la jerarquía se gestiona de manera explícita, mientras que en Auth0, las relaciones suelen definirse implícitamente mediante metadatos, atributos adicionales como `app_metadata`, o roles específicos asignados a los usuarios. Esto implica que Auth0 no maneja directamente estructuras jerárquicas ni árboles de herencia complejos, sino que deja en manos del desarrollador la implementación de cualquier lógica adicional que pueda requerirse.

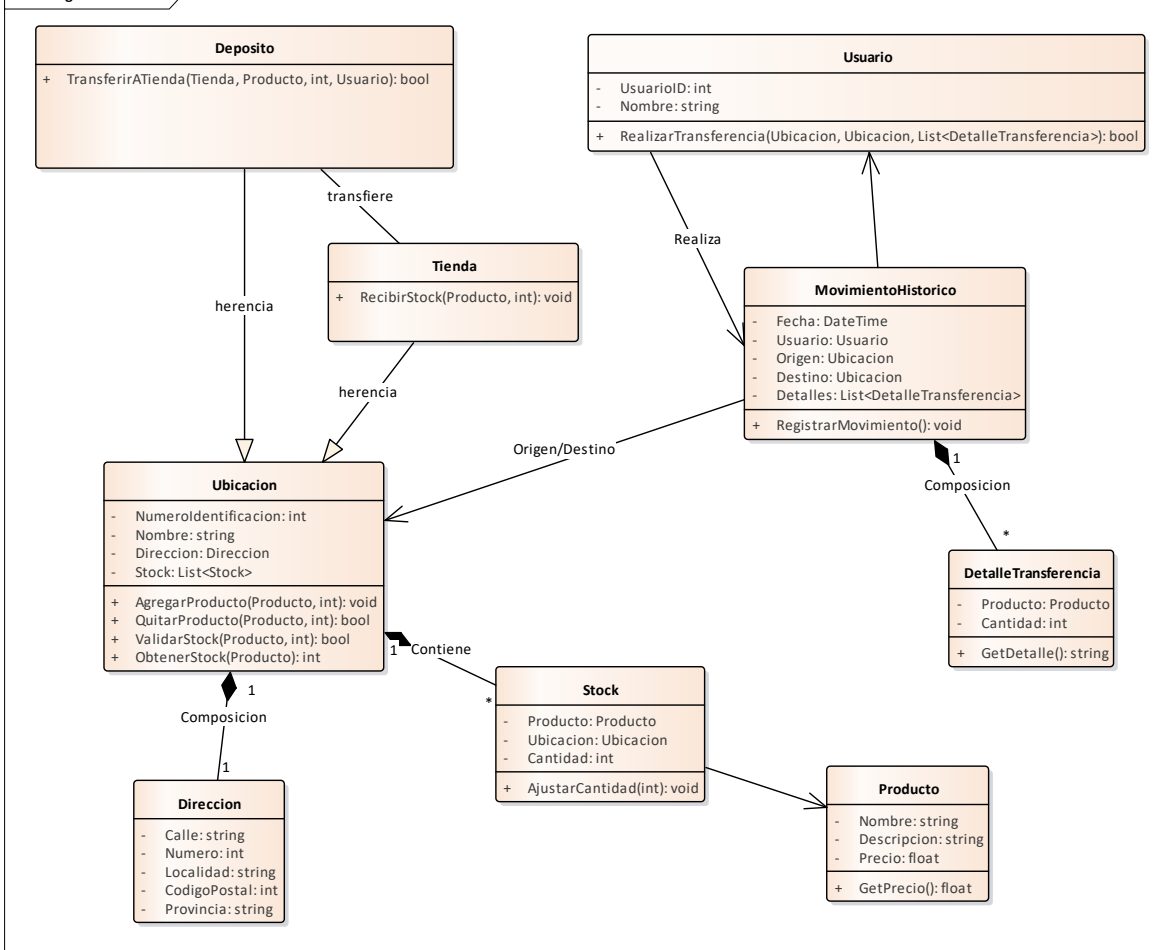
Finalmente, otra diferencia clave radica en la flexibilidad de las operaciones permitidas. El patrón Composite está especialmente diseñado para ejecutar operaciones recursivas sobre estructuras jerárquicas, facilitando cálculos acumulativos o transversales. Auth0, aunque proporciona APIs robustas para gestionar usuarios y sus respectivos permisos, no está optimizado para realizar operaciones complejas que requieran recorrer estructuras anidadas o calcular automáticamente accesos heredados. Por lo tanto, tareas como determinar el acceso acumulado de un usuario mediante roles heredados requerirán lógica externa complementaria.

PRÁCTICA (100)

- 1) Desarrollar un diagrama de clases (Sólo del dominio) y el programa donde se observe claramente los campos, propiedades y métodos para un sistema de logística donde nos informan que existen varios depósitos y tiendas donde sus principales datos son: Número de identificación, nombre y dirección (Calle y número, localidad, código postal y provincia). Las acciones que se pueden desarrollar son: Transferencia de stock de diversos productos entre depósitos y transferencia de stock desde depósitos a tiendas. Los depósitos y tiendas cuentan con ciertos productos de los cuales conocemos su nombre, descripción y precio. Se deberá registrar cada movimiento histórico de las operaciones que contendrá: Origen, destino, fecha, usuario y el listado de productos con sus cantidades transferidos. Se deberá contar con al menos un usuario registrado en el sistema para realizar las operaciones. Para poder realizar transferencias de cualquier producto se debe validar el stock sobre el origen. **(10)**



class Diagrama de clases



- 2) Generar un modelo de Repositorio accesible a través de una capa de lógica, que deberá a su vez, ser accedida por una fachada de tipo WebApi (Con los métodos necesarios para las operaciones solicitadas) para persistir la información de transferencias entre los diversos depósitos/tiendas y luego poder consultar las mismas. Puede utilizar cualquier modelo de persistencia, incluso en memoria, para las entidades intervinientes. **(20)**
- 3) El modelo de solución generado deberá contener mínimamente 5 capas (Sitio web para al cliente, WebApi, Business Layer, DomainModel y Repositorio) asignando las responsabilidades correspondientes. Utilizar los patrones de diseño y arquitectura que usted crea conveniente. **(20)**
- 4) Deberá generar casos de prueba para testear las operaciones solicitadas. Mínimamente contemplar:
 - a. Se tienen que poder filtrar los tipos de operaciones a partir de un destino (Escoger un depósito o tienda) y luego mostrar: Origen, producto, cantidad, usuario y fecha, esta última ordenada de manera descendente.
 - b. Se debe de contar con otro informe que nos permita filtrar los destinos que solicitaron más de tres operaciones en el día, mostrando origen, destino, producto, cantidad y usuario que los solicitó ordenado por fecha de manera ascendente.
 - c. A partir de un número de tienda, mostrar los primeros 10 movimientos entre las fechas determinadas por el usuario (Elija los datos que desea mostrar para esta consulta). **(40)**
- 5) Entre la WebApi y el sitio web se dialogará con formato de mensajes en json. Puede utilizar cualquier tecnología de desarrollo para la web: Razor, Blazor, HTML5 puro con invocaciones Ajax, etc. También será valorado el diseño CSS y javascript que permita una buena usabilidad. **(10)**