Resolver el taller y agregarlo a la carpeta creada con su nombre en git.

1. Indique las capas en las que se dividen la arquitectura limpia.

- Domain layer.

- Application layer.

- Adapter layer.

- Infrastructure layer.

2. Clasifique los siguientes componentes de acuerdo a las capas de clean arquitectura:

Entidades -> Domain layer.

Infraestructura -> Infrastructure layer.

API's -> Adapter layer.

Gateways -> Adapter layer.

UI -> Infrastructure layer.

Casos de Uso -> Application layer.

Web -> Infrastructure layer.

Databases -> Infrastructure layer.

Dispositivos -> Infrastructure layer.

3. Describa el principio de abstracción definido en arquitecturas limpias.

La lógica del negocio debe estar más al interior de las capas de arquitectura limpia para que las capas de infraestructura y adaptadores se conecten a esta más no la puedan definir.

4. Es posible que una capa haga referencia a una capa interna de la arquitectura?

Sí porque si se hace referencia en una capa externa debe existir en las capas debajo de ésta.

5. Cual es la definición de API?

Application programming interface. Permite la comunicación entre los componentes de software, como el front y el back.

6. En sus propias palabras especifique acoplamiento y cohesión he indique cual es la mejor configuración que se debe mantener para que un sistema conserve los principios de arquitecturas limpias.

Acoplamiento se refiere a la dependencia entre dos o varios componentes para existir o realizar una operación.

Cohesión se refiere cuando existe relación entre los componentes pero tienen límites establecidos y no dependen uno de otro.

La mejor configuración es un bajo acoplamiento y una alta cohesión para tener bien definido el alcance y los límites entre cada componente y aunque se relacionen no dependa uno del otro.

7. Mencione y describa las partes de los casos de uso.

-Caso de uso de Aplicación. Se modelan los actores que utilizan un sistema para lograr un objetivo a través de actividades y variantes.

Partes:  -Actor.

-Caso de uso.

-Relación.

-Límites del sistema.

-Caso de uso de Negocio. Este representa cómo se hace el trabajo actual manualmente y no necesariamente lo realiza el sistema ni tiene la intención de automatizarse en el sistema de destino.

Partes:  -Solo un objetivo

-Un solo punto de partida

-Un solo punto final.

-Múltiples caminos para llegar de principio a fin

-Especificar el comportamiento para una variedad de posibles condiciones

-Cada condición puede requerir acciones específicas.

8. Diagramación de casos de uso (tiempo 40 minutos):

Una   empresa   comercializadora,   desea   invertir   en  el   desarrollo   de   un   sistema orientado al  control de  las ventas de sus productos, los

requerimientos que el sistema debe satisfacer son:

A. Permitir   que   los  Vendedores  registren   las   ventas,   para   ello   deberán consultar el saldo de los productos.

B. Permitir al Jefe de Recursos Humanos administrar al personal de ventas, registrando,   actualizando   y   eliminando   a   los   diferentes   vendedores

que ingresen o salgan de la empresa.

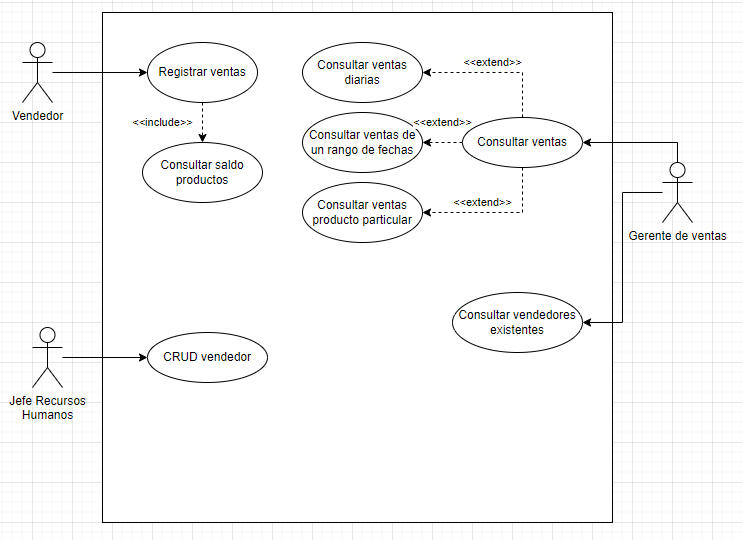
C. El Gerente de Ventas podrá consultar los siguientes reportes:

Ventas diarias.

Ventas dentro de un rango de fechas.

Ventas de un producto en particular.

Vendedores existentes en la empresaoVendedores que tienen una ruta a seguir.



9. Teniendo en cuenta el ejemplo anterior y luego de definir los casos de uso especifique que componentes debería tener y en a que capa de arquitecturas

limpias pertenece.

Entities -> Domain layer.

Application business rules -> Application layer.

Database, devices -> Infrastructure layer.