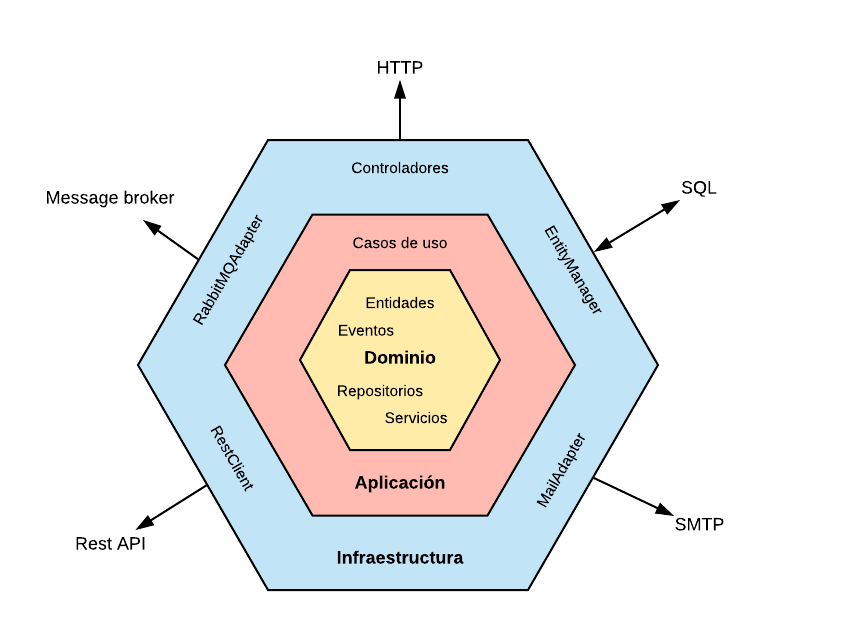
Clean Architecture también conocidas como arquitecturas por capas; las capas desacoplan la lógica del negocio de la infraestructura de la persistencia.



Los beneficios que ofrece son diversos;

* Fácil migración de bases de datos (sólo habrá que agregar un componente).
* Facilidad para actualizar cualquier framework (se separa la lógica de dominio de la infraestructura (capas))
* Escala mejor el código, y programar más rapido

De las arquitecturas por capas, la más usada es la Hexagonal.



Existe una regla de dependencia en las Clean Architectures, siempre se debe ir de a fuera hacía adentro; La capa de infraestructura conoce y puede llamar a la capa de Aplicación, y la capa de Aplicación lo mismo con la de Dominio. La capa de Dominio sólo puede interactuar consigo misma.

* La capa de Infraestructura se refiere a todo lo que tiene una entrada y salida en el sistema
* La capa de Aplicación se refiere a los casos de uso
* La capa de Dominio se refiere a las entidades, valueObjets

De esta manera, se obtienen los beneficios mencionados arriba.

Capa de Dominio

* El valueObject se encarga de mapear los datos que harán representación de la entidad, la entidad en sí, es una interfaz que hace referencia a lo que se va a guardar en la DB, es el Core. El valueObject va a funcionar como un middleware, va a mapear y hacer validaciones contra la entidad.
* El user.repository es una capa de abstración, haciendo referencia a los distintos metodos a utilizar o implementar con la capa de datos. Van todos los metodos sin discriminar que servicios externos o que capas usan.
* Es muy importante tener en cuenta que los archivos de la capa de Dominio no pueden tener relación con las otras capas, lo cual se demuestra en los archivos importados.

Capa de Aplicación

Un UseCase

Capa de Infraestructura