

Arquitectura de Desarrollo de software DDD (Domain-Driven Design)

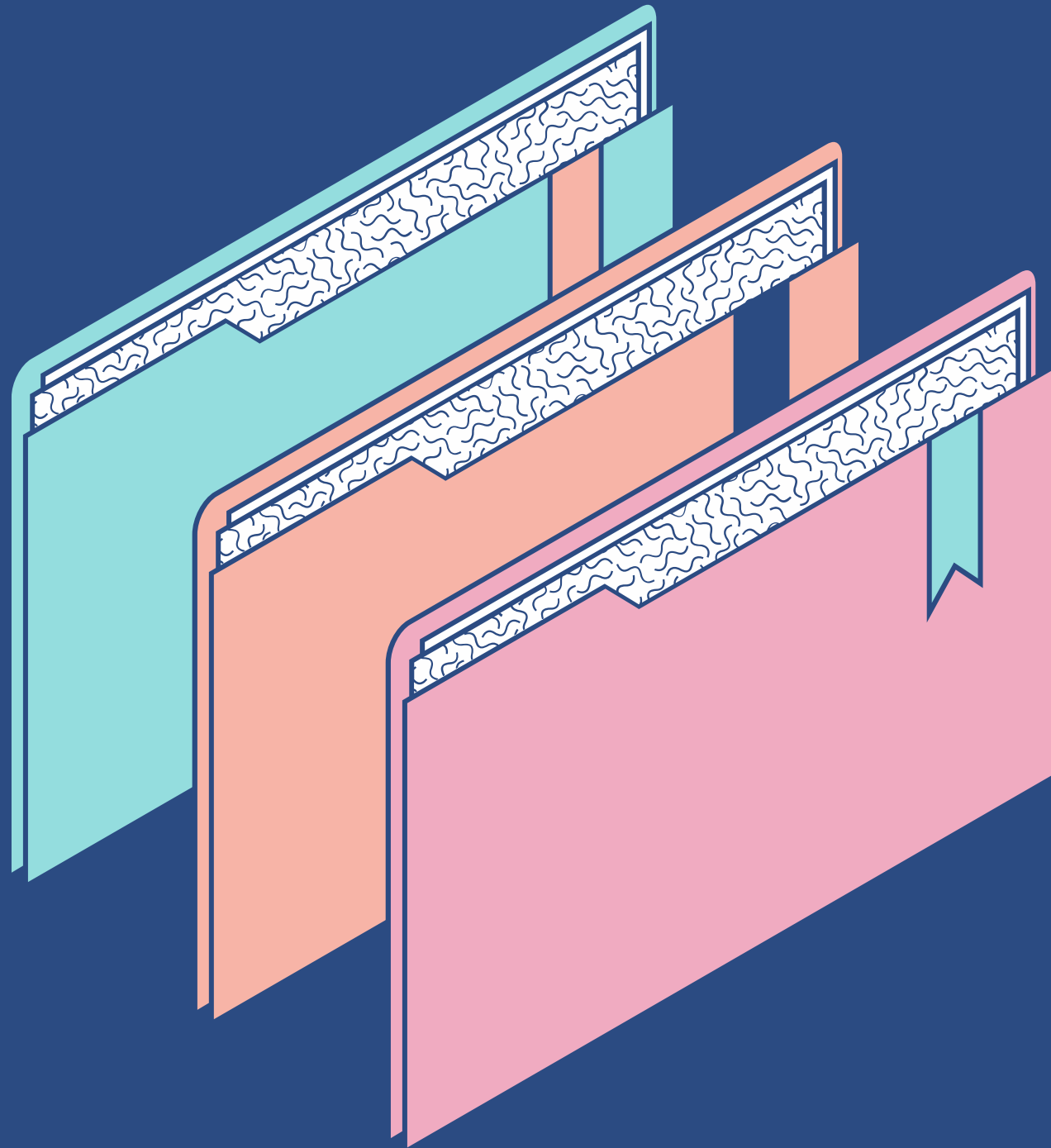
Estudiantes:

- Rene Manuel Avilés Montecinos.
- José Milton Mamani Callizaya.
- Roberto Carlos Aruquipa Arzala.
- Diego Carlos Rojas Apaza.

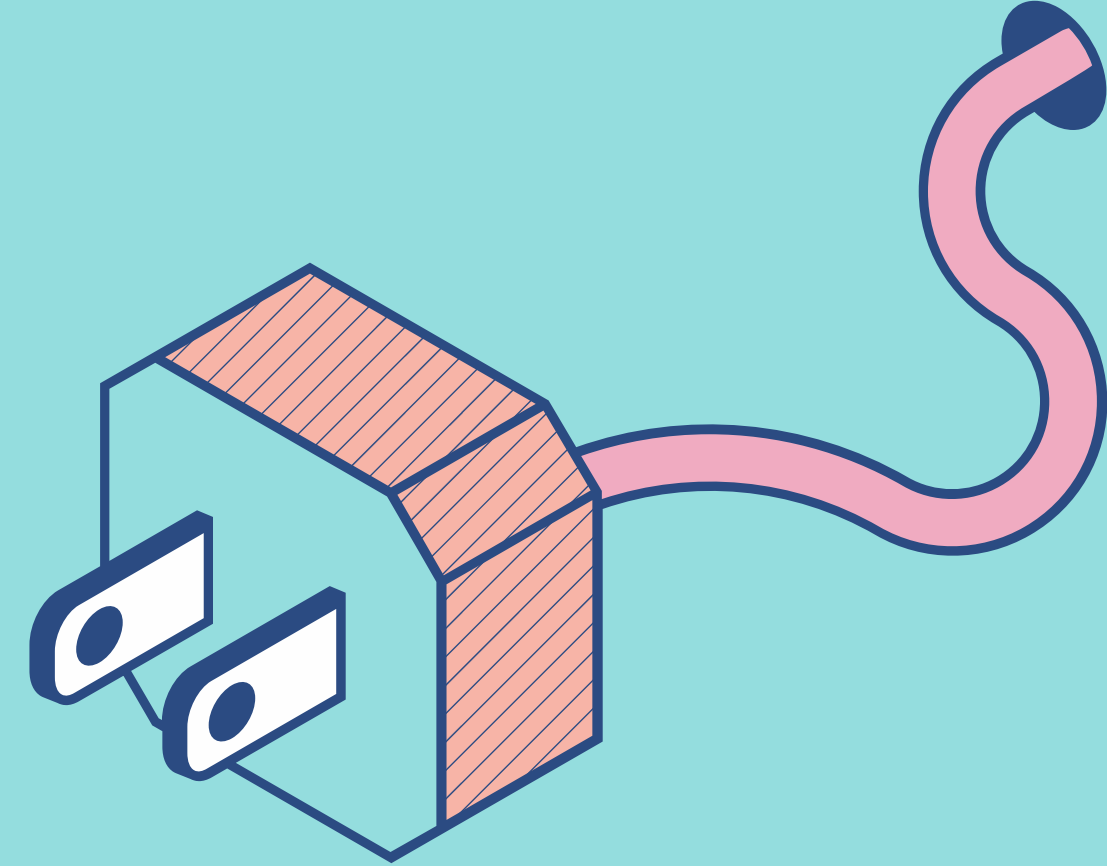
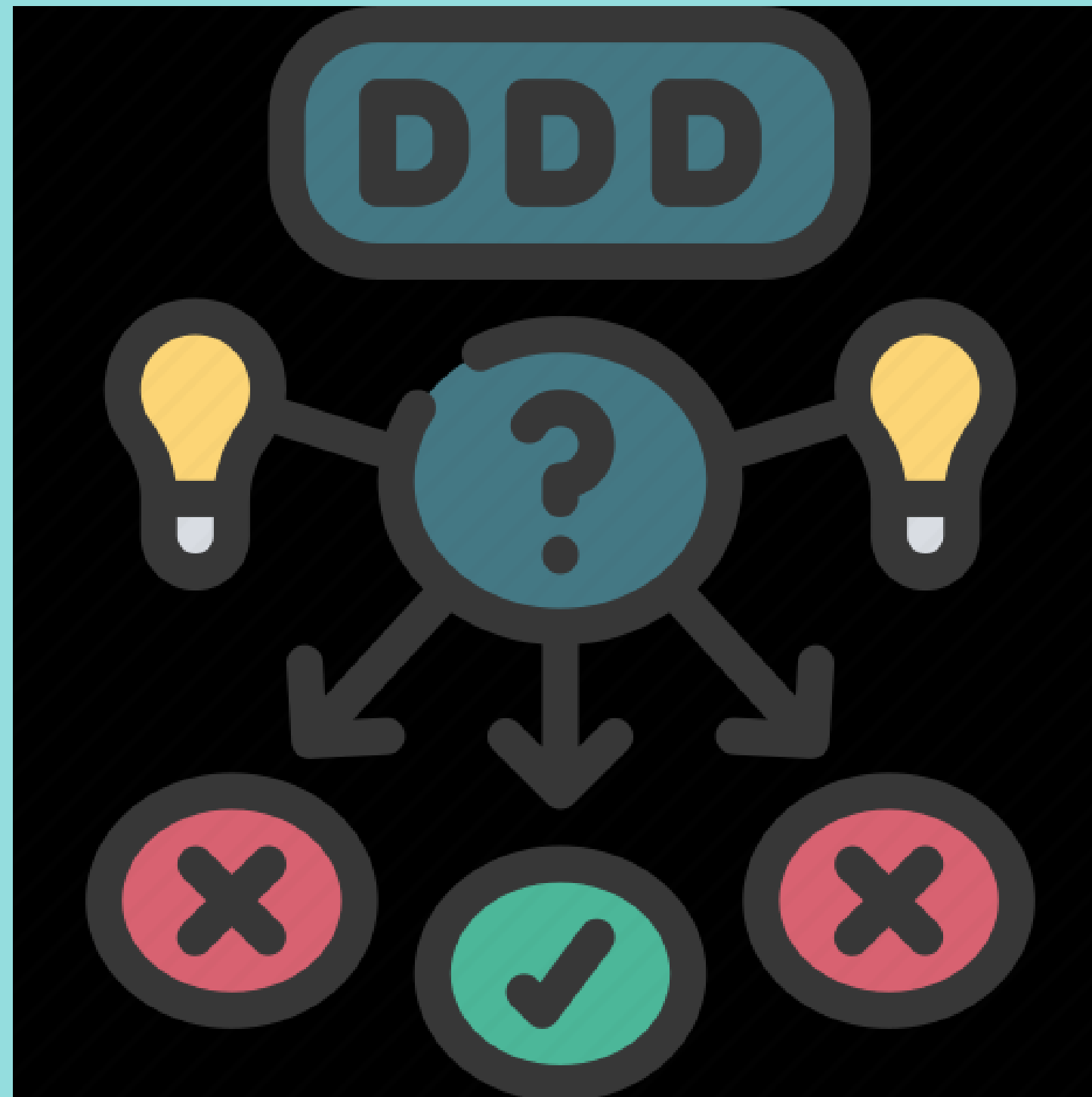


Índice

- Introducción
- Objetivos
- Desafíos y Consideraciones
- Marco Teórico
- Marco Práctico
- Conclusiones



Introducción



Objetivos

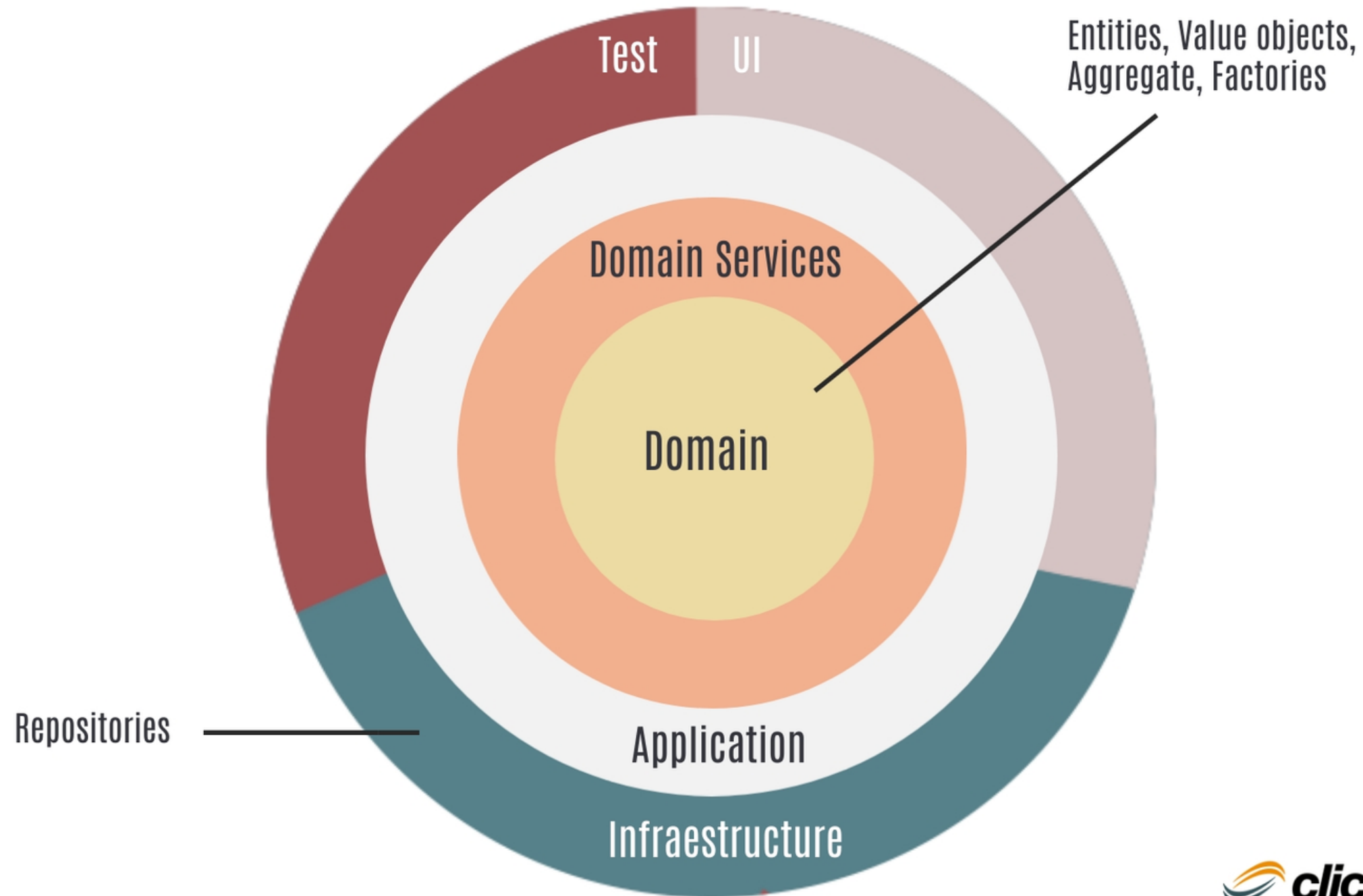
- Comprender el dominio
- Crear un modelo de dominio
- Perfeccionamiento
- Implementación del modelo dominio



Desafíos y Consideraciones

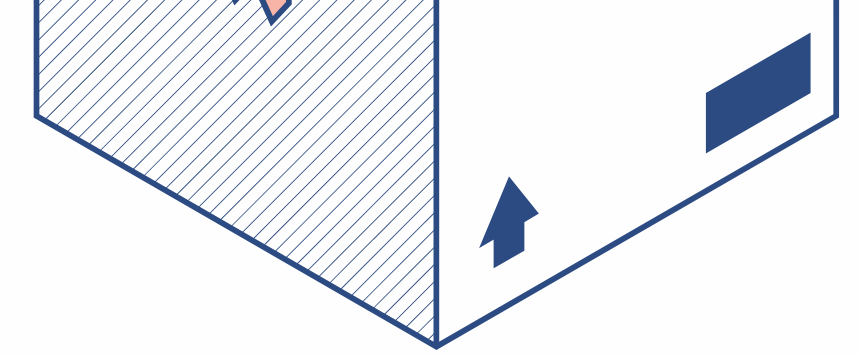
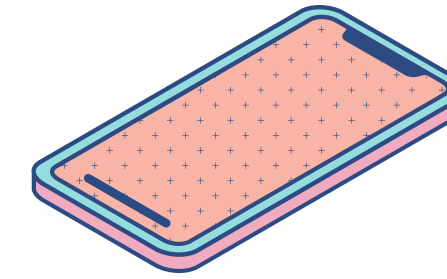
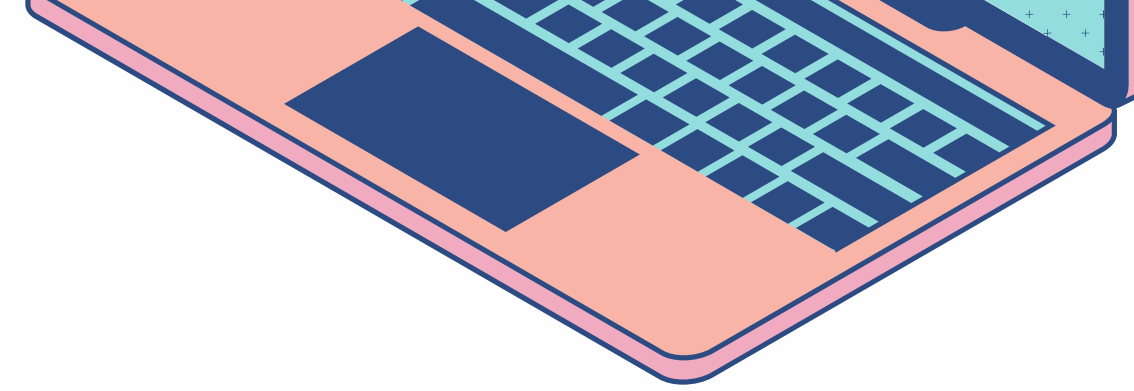
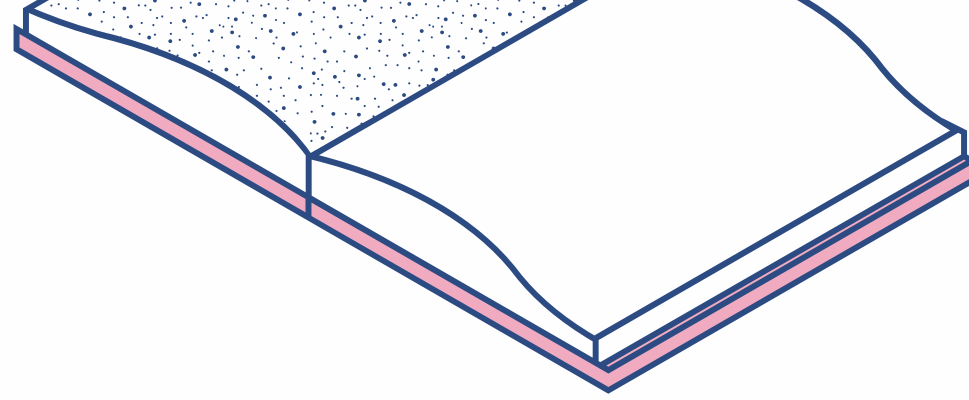
- Complejidad del Dominio
- Expertos en la materia
- Colaboración del Equipo
- Comunicación
- Cambios Constantes





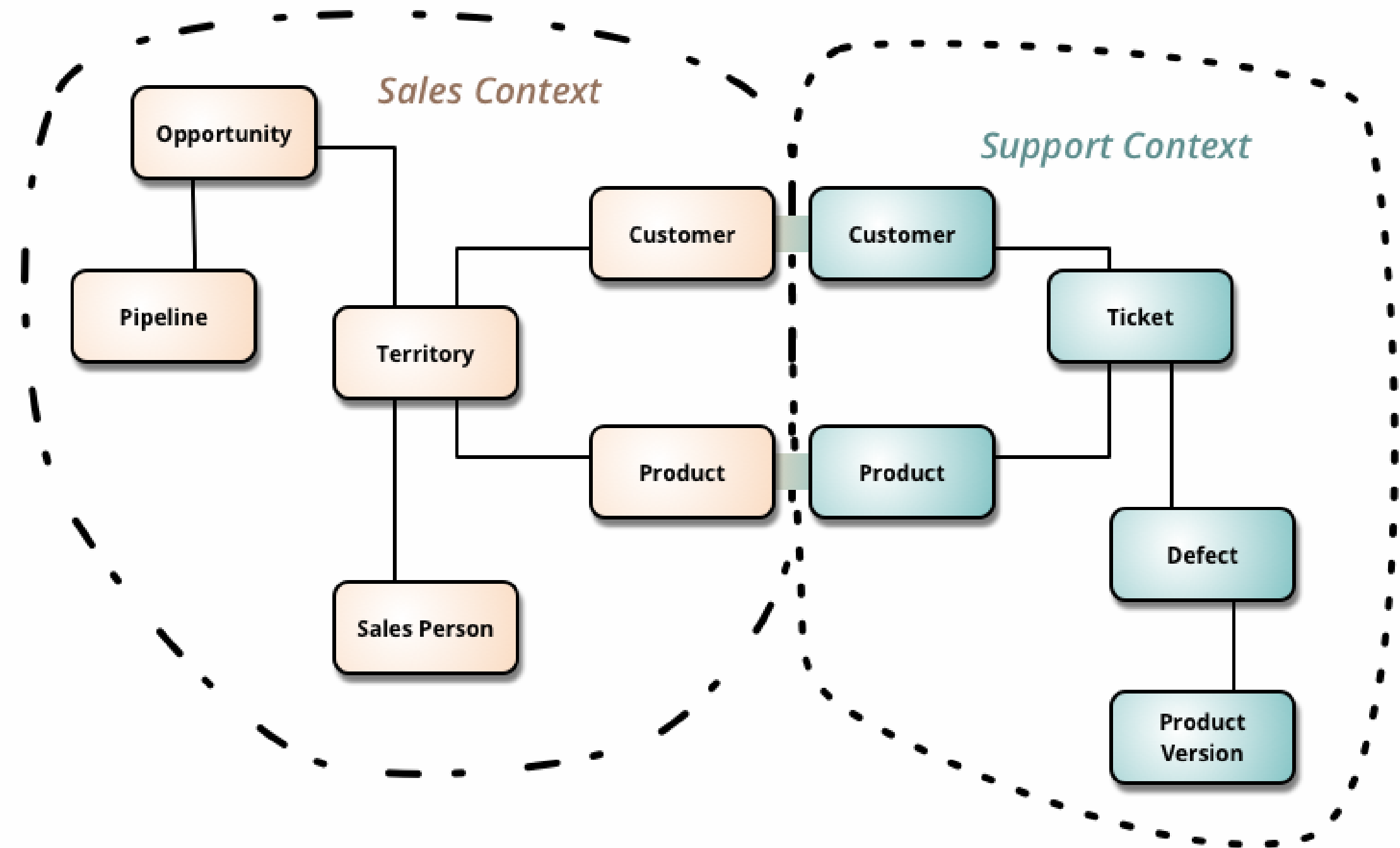
Marco Teórico

- Proceso de diseño con DDD
- Implementación de DDD
- Patrones y Buenas Prácticas de DDD
- Integración con otros Aspectos del Desarrollo
- Herramientas y tecnologías



Conceptos

- Proceso de Diseño con DDD
- Patrones y Buenas Prácticas de DDD
- Integración con Otros Aspectos del Desarrollo
- Herramientas y tecnologías

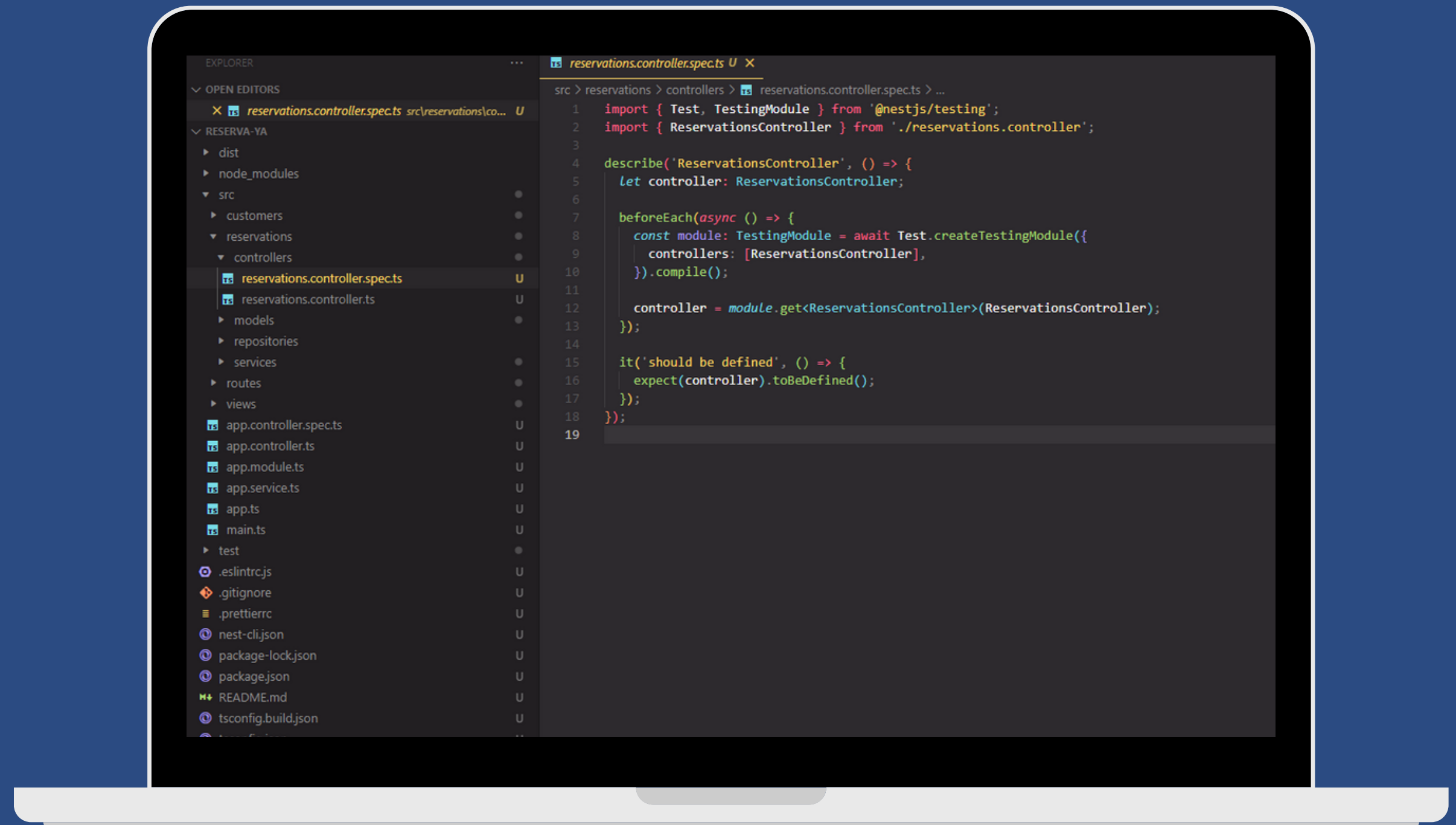


Marco Práctico

Programa de ejemplo:

“Reserva de mesas para
restaurants”

Con NestJS



Conclusiones



El DDD puede ayudar a crear sistemas de software que reflejen con precisión el dominio del problema y sean fáciles de entender, mantener y evolucionar con el tiempo.

¡GRACIAS!

