



# Business Solution Design - DDD

28-11-2022

---

Leandro Baier

## Metas

Temas a revisar:

1. ¿Qué es domain driven design y qué ventajas trae al diseño de una solución?
2. Entender conceptos de dominio, subdominio, Ubiquitous language y bounded context


### 1. ¿Qué es domain driven design y qué ventajas trae al diseño de una solución?

El DDD provee una estructura de prácticas y terminologías para tomar decisiones de diseño que enfoquen y aceleren el manejo de dominios complejos en los proyectos de software.

Nos permite conocer de modo extenso el problema del negocio y que desean desarrollar los clientes. De esta forma, podemos dividirlo en subdominios con el objetivo de lograr así soluciones modulares, que premian la adaptabilidad de cada subdominio y, por lo tanto, la adaptabilidad de la solución en general.

### 2. Entender conceptos de dominio, subdominio, Ubiquitous language y bounded context

- Dominio: La lógica de dominio es el propósito del modelado. Comúnmente se conoce como la lógica de negocios. Aquí es donde las reglas de negocio definen la forma en que se crean, almacenan y modifican los datos.
- Subdominio: Un dominio tiene varios subdominios que hacen referencia a diferentes partes de la solución de negocios. Por ejemplo, una tienda minorista en línea podría tener un catálogo de productos, inventario y entrega como sus subdominios.
- Ubiquitous Language: El lenguaje ubicuo es una metodología que se refiere al mismo lenguaje que los expertos y desarrolladores usan cuando hablan sobre el dominio en el que trabajan. Esto es necesario porque los proyectos pueden enfrentar problemas graves con un lenguaje interrumpido. Vale decir, entre los desarrolladores de software y el cliente que solicita el desarrollo. Por eso es necesario definir un conjunto de términos que las dos partes utilicen. Todos los términos en el lenguaje ubicuo se estructuran en torno al modelo de dominio.

- 
- Bounded Context: El contexto delimitado es un patrón central en el diseño guiado por dominio. En el contexto es donde se implementa el código, después de haber definido el dominio y los subdominios. Los contextos delimitados representan los límites en los que se define y se aplica un determinado subdominio. Aquí, el subdominio específico tiene sentido, mientras que otros no. Una entidad puede tener nombres diferentes en contextos diferentes.