



IIC2113 – Diseño Detallado de Software

CLASE 4 – Patrones de diseño y arquitectura

*Pontificia Universidad Católica de Chile
2020-2*

Índice

- 01 **Patrones de diseño**
- 02 **Patrones de arquitectura**
- 03 **DDD**
- 04 **CLEAN**
- 05 **Próxima clase**

02. Patrones de Arquitectura

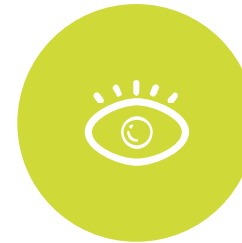
Un breve resumen



¿Qué es la arquitectura de software?

En el mundo del desarrollo se separan las nociones de **Arquitectura de Software** del **Diseño de Software**, en que la arquitectura busca una abstracción general del sistema que se construirá, y define las prioridades de la solución respecto a los atributos de calidad esenciales.

Existe un curso enfocado a la arquitectura (IIC2173 - Arquitectura de sistemas de software) por lo que veremos solo una pincelada para conectarla con el diseño.



Arquitectura

- Visión General
- **Estructura**, interfaces y comportamiento
- Atributos de calidad
- No ahonda en los detalles de implementación



Diseño

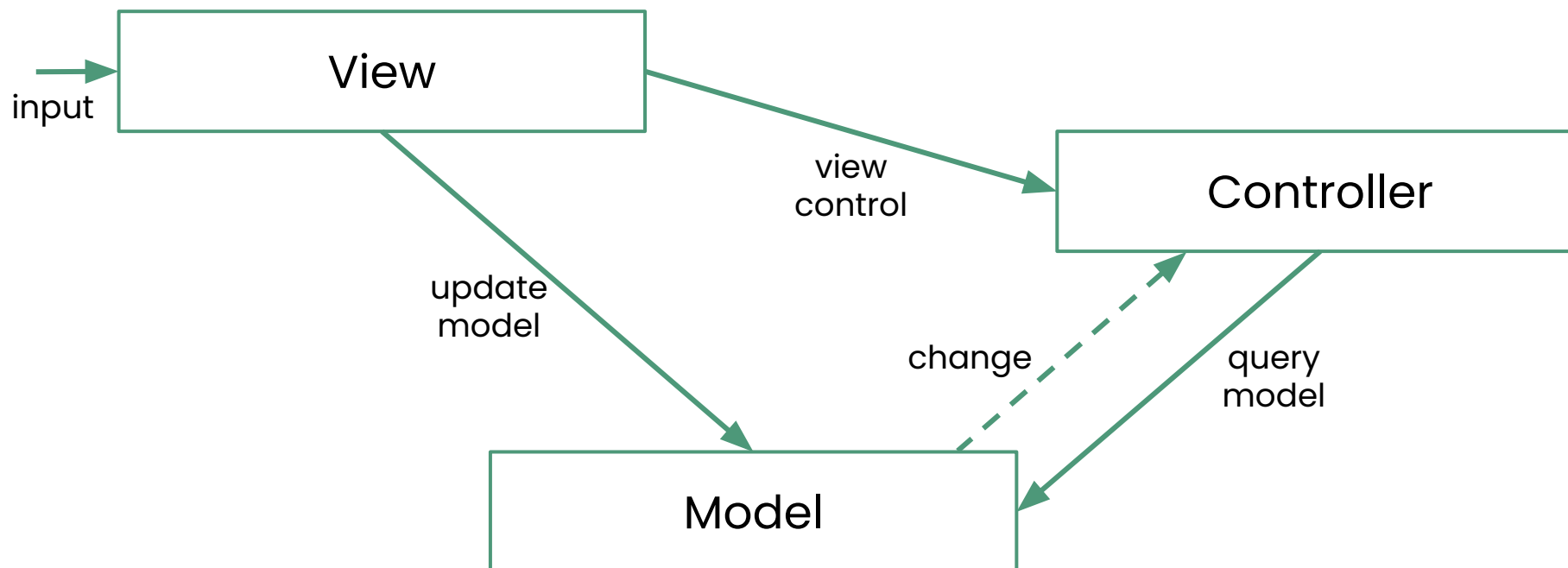
- Detalle de la implementación
- Paso de los **requisitos** a código
- Algoritmos, estructuras de datos y comportamientos a nivel de implementación
- Guía el desarrollo

Estilo arquitectónico

Son el resultado de una serie de decisiones que llevan a un patrón común para resolver cierto problema.
Hay muchos estilos o patrones arquitectónicos.

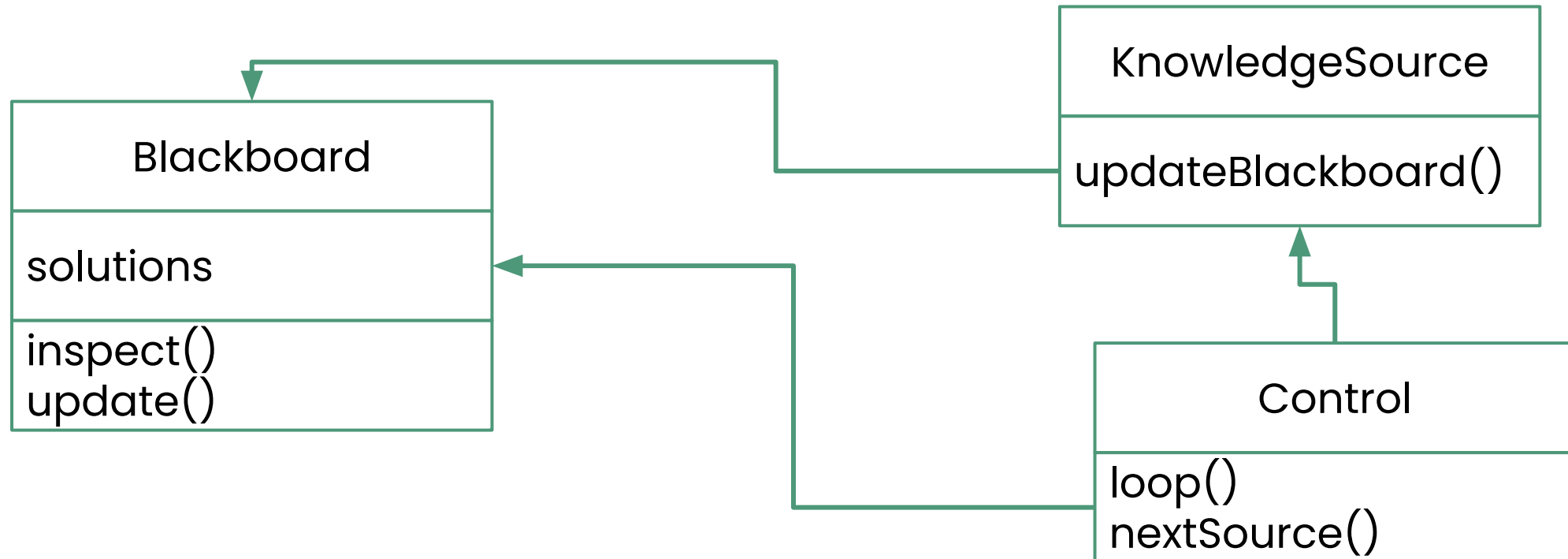
1. MVC

Modelo-Vista-Controlador, como su nombre indica divide las responsabilidades en 3: el modelo que contiene el core y datos, controladores se encargan de gestionar los inputs, y vistas que exponen la información al usuario..



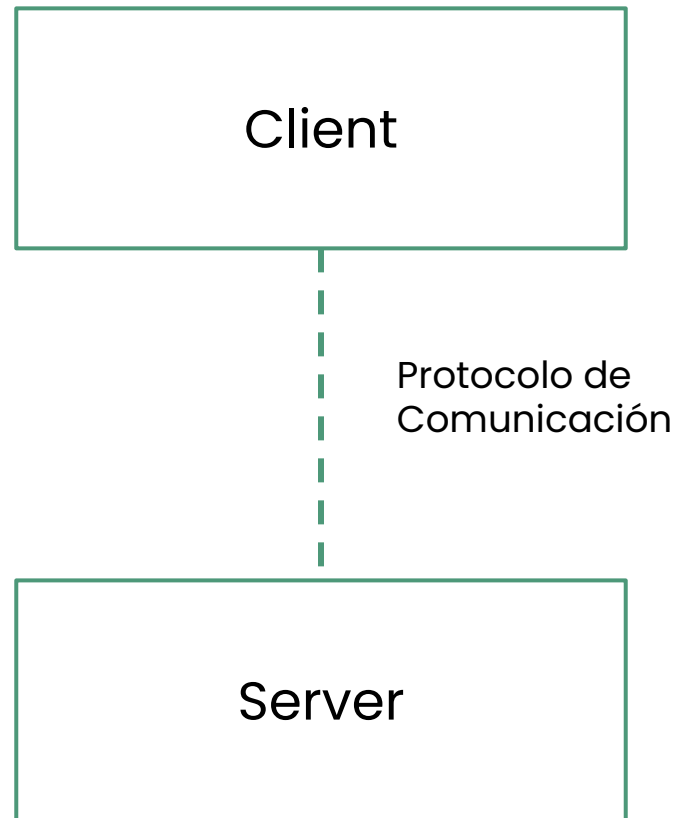
2. Blackboard

Problemas que no tienen una solución determinística. El pizarrón contiene objetos de la solución. Los componentes pueden añadir nuevos elementos al pizarrón.



3. Client-server

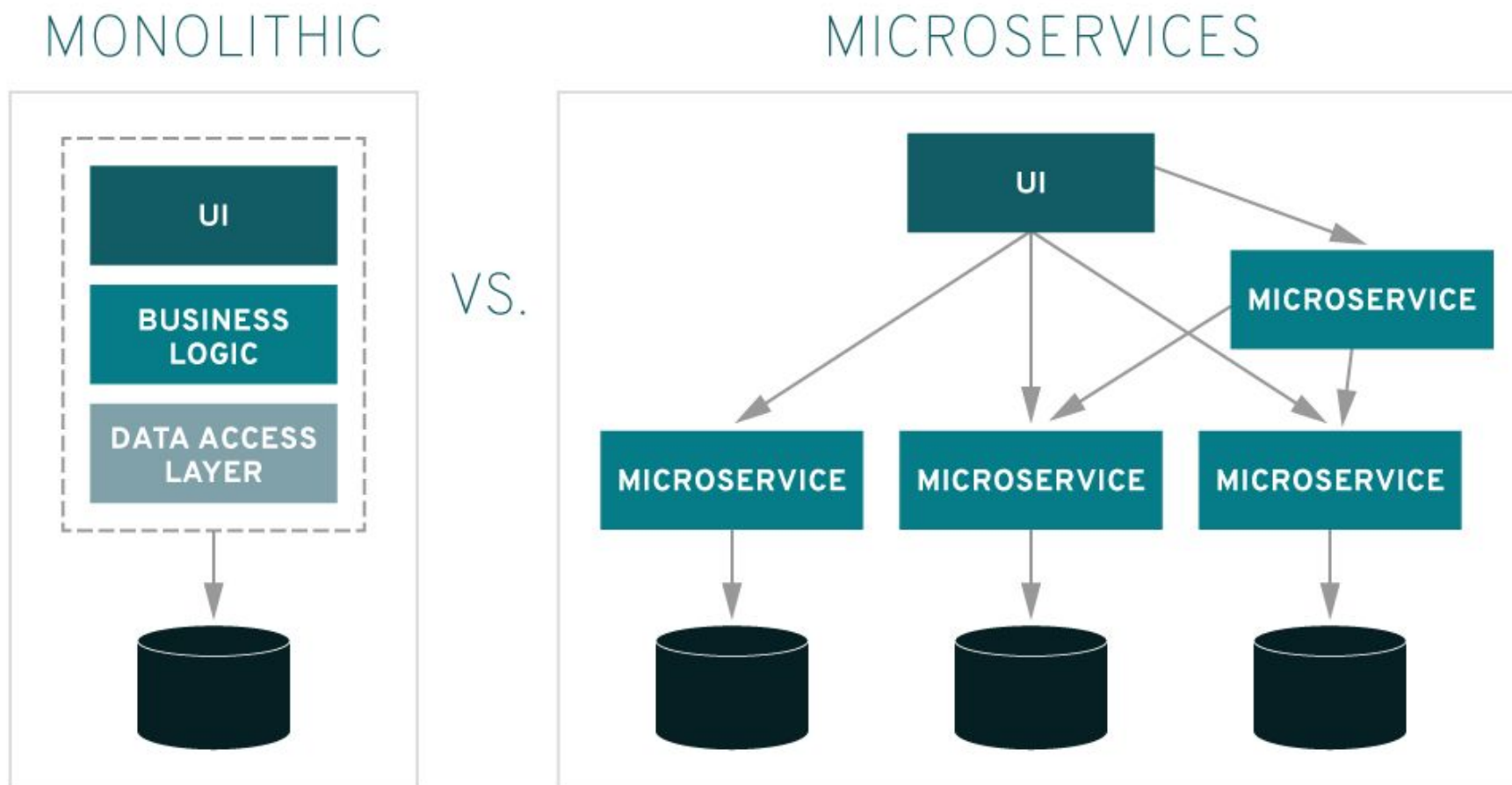
Uno o más clientes se alimentan de la información otorgada por un servidor.



4. Microservicios vs Monolito

Microservicio: El sistema funciona como un conjunto de pequeños servicios cada uno en su proceso que se comunican entre sí.

Monolito: La capa de interfaz, de lógica y de datos corren en el mismo proceso.



Cuando un sistema crece sin control, sin diseño y sin planificación.

5. Big ball of mud



Diagrama extraído de <https://www.redhat.com/en/topics/microservices/what-are-microservices>

03. Domain Driven Design

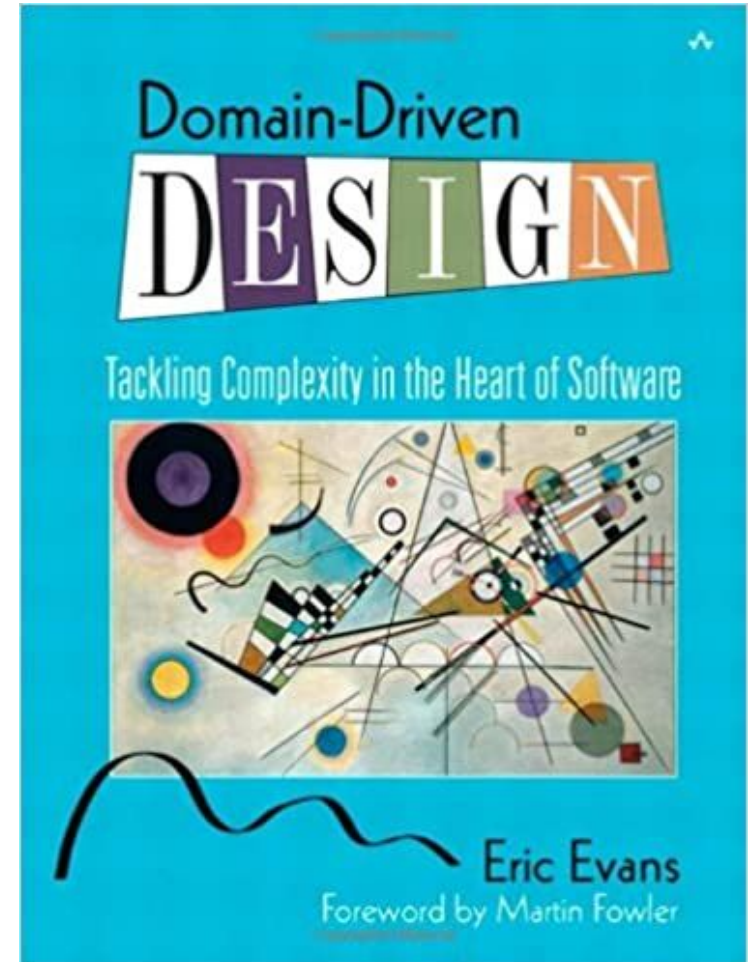
Un breve resumen



Domain-Driven Design

Domain-Driven Design es un enfoque de desarrollo de software que enfoca el desarrollo del dominio del modelo en base al conocimiento los procesos y reglas del dominio de la organización. Busca el punto común entre expertos del dominio y los desarrolladores.

Existe un curso enfocado a las metodologías (IIC3143 - Desarrollo de Software) por lo que veremos solo una pincelada para conectarlo con el diseño.



Dominio

Es el conocimiento del problema que se está intentando resolver.

Modelo

Abstracciones que describen el dominio con el fin de resolver el problema.

Ubiquitous Language

Lenguaje común utilizado por todos los miembros del equipo.

04. CLEAN


Un breve resumen



CLEAN

Término acuñado por Robert C. Martin (Uncle Bob), co-autor del manifiesto ágil y actual autor de [cleancoder](#). Ha escrito una serie de libros motivadas a las buenas prácticas y a mantener el código “limpio”.

- **Clean Architecture:** A Craftsman's Guide to Software Structure and Design (Robert C. Martin Series)
- **Clean Code:** A Handbook of Agile Software Craftsmanship (Robert C. Martin Series)
- **The Clean Coder:** A Code of Conduct for Professional Programmers (Robert C. Martin Series)



Uncle Bob Martin

Programmer, Speaker, Teacher

[Classes & Talks](#) [Invite Me!](#) [Subscribe](#) [Contact](#)

[Recommended Books](#) [Blogs](#) [Ancient Blogs](#) [Old Articles](#) [Clean Code Gear](#)

Classes

Clean Code

Duration: 2-3 days.
Audience: Programmers of any experience.

In this course Uncle Bob teaches the basics of *Clean Code* as described in the [Clean Code](#) book.

[More Info...](#)

Clean Architecture

Duration: 2 days.
Audience: Programmers, leads, architects.

In this course Uncle Bob teaches the basics of Clean Architecture as

Upcoming Events

11 Nov, 2020 (Wed)

Clean Architecture - Week Two - 2 hrs

ON-Line course
This is a 3 week course for Senior programmers, Technical Leads, and Architects who desire to improve their knowledge and skills with regard to high-level software structure and architecture. Outcomes After this class students will: ...understand the goals and aims of software

05. Próxima Clase



Próxima clase

- Ingeniería Inversa y Paradigmas de programación

Clase 4 - Patrones

Bibliografía

- Pressman, R. S. (2009). Software Engineering: A Practitioner's Approach
- Gamma, E., Helm. R., Johnson R., Vlissides, J. (1994). Design Patterns: Elements of Reusable Object-Oriented Software



IIC2113 – Diseño Detallado de Software

Fernanda Sepúlveda – mfsepulveda@uc.cl

Pontificia Universidad Católica de Chile
2020-2