

# DP2 2021-2022

## Knowledge of WIS' architecture



Repositorio:

<https://github.com/mpadillatabuenca/Acme-Toolkit.git>

Miembros:

- José Manuel Bejarano Pozo (josbezp@alum.us.es)
- Mario Espinosa Rodríguez (maresprod5@alum.us.es)
- Andrea Meca Sánchez (andmecs@alum.us.es)
- Manuel Padilla Tabuenca (maresprod5@alum.us.es)
- Ezequiel Pérez Sosa (ezepersos@alum.us.es)
- Javier Terroba Orozco (javteroro@alum.us.es)

GRUPO G1-E2-05

Versión 1.0.0

27/02/2022

# Tabla de contenidos

	0
Tabla de contenidos	1
Historial de versiones	2
Resumen ejecutivo	2
Introducción	2
Contenido	2
Modelo-Vista-Controlador	2
Arquitectura en capas	3
Arquitectura de microservicios.	3
Conclusión	3
Bibliografía	3

## Historial de versiones

Fecha	Versión	Descripción de los cambios
27/02/2022	1.0.0	Creación y finalización del documento.

## Resumen ejecutivo

Dada la brevedad del documento estimo poco acertada la realización de un resumen ejecutivo.

## Introducción

En este documento realizado por Ezequiel Pérez Sosa se explica los conocimientos previos del autor sobre las arquitecturas software, previo a la realización de la asignatura.

En el contenido del presente documento se hablará de los patrones arquitectónicos que el autor del documento ha utilizado durante su carrera universitaria o en el ámbito laboral previamente a la asignatura.

## Contenido

A continuación explicamos brevemente los patrones arquitectónicos vistos. Dichos patrones son una solución general y reutilizable para un problema dado. Se diferencian del patrón de diseño en su alcance más amplio.

## Modelo-Vista-Controlador

Este patrón arquitectónico (MVC) consiste en la división de responsabilidades y aplicación en tres partes.

- **Modelo:** Contiene la representación de los datos del sistema así como la lógica de negocio.
- **Vista:** Contiene la información que es enviada al cliente y establece la comunicación con este.
- **Controlador:** Es el intermediario entre el modelo y la vista. Gestiona la información y adapta los datos según necesidades.

Es un patrón ampliamente utilizado. Un framework que se ha utilizado durante la carrera que se basa en este tipo de patrón, es Spring.

## Arquitectura en capas

Este patrón arquitectónico se basa en la distribución de responsabilidades en capas.

Se tienen cuatro capas.

- **Capa de presentación:** Se trata de la interfaz que ve el usuario.
- **Capa de negocio:** Donde se procesa las reglas de negocio
- **Capa de persistencia:** Donde se almacenan u obtienen datos.
- **Capa de base de datos:** Donde se tienen los proveedores de bases de datos.

Este tipo de patrón también está ampliamente extendido.

## Arquitectura de microservicios.

Este tipo de arquitectura basa su funcionamiento en las APIs. Estas establecen una interfaz de comunicación con un servicio en concreto. La construcción de estos sistemas basados en esta arquitectura está compuesto por varias APIs y basan su funcionamiento en estos mismos. Se trata de una arquitectura descentralizada.

## Conclusión

Al final de la realización de este documento, podemos concluir que los conocimientos son repartidos en los patrones arquitectónicos de software. Conocimiento que se espera profundizar y pulir en la realización de la asignatura.

## Bibliografía

1. Apuntes de la asignatura de Diseño y Pruebas 1 ([ev.us.es](http://ev.us.es))
2. Lista resumida de patrones de diseño ([Los 10 patrones comunes de arquitectura de software | by Wilber Ccori huaman | Medium](#))