Ciclo de Vida del Desarrollo de Software

¿Que es el SDLC? ¿Cuantos procesos tiene? 1. Identification of problems or objectives Identificación del problemas u objetivos

2. Analysis
Análisis

6. Maintenange
Mantenimiento

3. Design
Diseño

5. Implementation / Operation Implementación / Operación

4. Development and testDesarrollo y pruebas

Implantación/operación de sistemas

Actividades que hacen que un **sistema de software** esté **disponible para su operación**, en estas, el sistema es **instalado** para su uso.

Ciclo de Vida del Desarrollo de Software Implementación / Operación

Se instala la solución, y se:

- Construye la **estructura de datos**
- Configura la plataforma final
- Capacita a los usuarios finales y personal técnico que administrara el sistema
- Planea y ejecuta la migración del sistema antiguo al nuevo, si se requiere.

Existen cuatro tipos de implementación:

- Directo
- Paralelo
- Piloto
- fases

Poner en operación



Mantenimiento de sistemas

Modificación de un producto software después de haber sido entregado [a los usuarios o clientes] con el fin de corregir defectos, mejorar el rendimiento u otros atributos, o adaptarlo a un cambio en el entorno.

Ciclo de Vida del Desarrollo de Software Mantenimiento

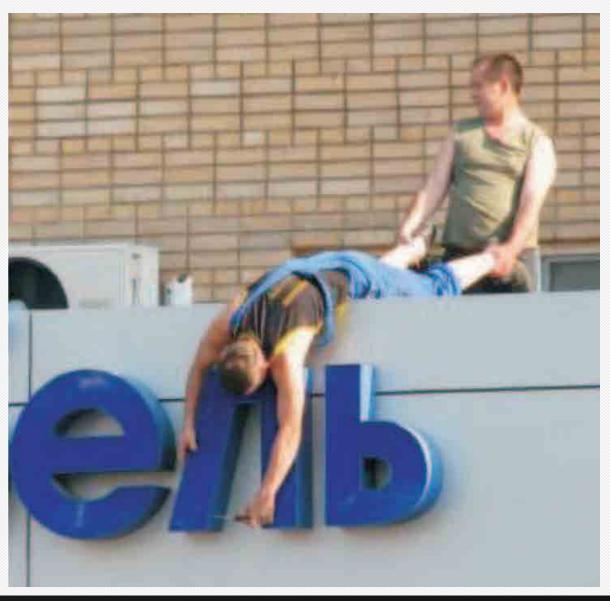
Se modifica el software para corregir, mejorar, o adaptar las necesidades cambiantes del cliente o el entorno, es decir, se aplica una evaluación y el software sufre una evolución.

Los tipos de mantenimientos pueden ser:

- Mantenimiento Correctivo
- Mantenimiento Adaptativo
- Mantenimiento Perfectivo
- Mantenimiento Preventivo

Incluye la retirada del Software

Evaluación y evolución



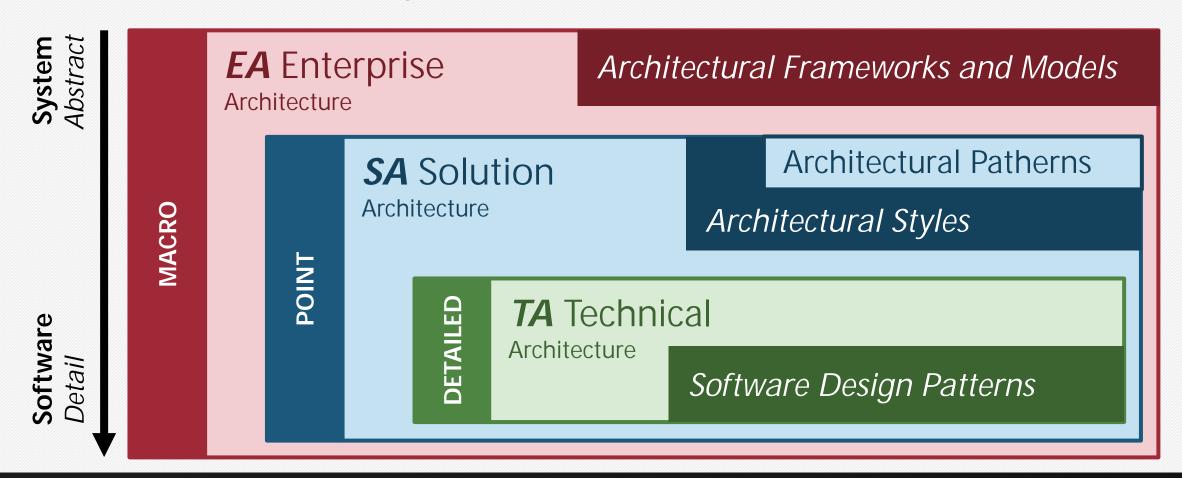
Arquitectura de software y sistemas

¿Que es la Arquitectura? ¿Cuantos tipos de arquitectura existen?

Arquitectura

Es la organización fundamental de un sistema que representa sus componentes y sus relaciones entre sí y con el medio ambiente, además de los principios que rigen su diseño y evolución

Enterprise Architecture | Architectural Frameworks and Models Solution Architecture | Architectural Styles | Architectural Patherns Technical Architecture | Software Design Patterns



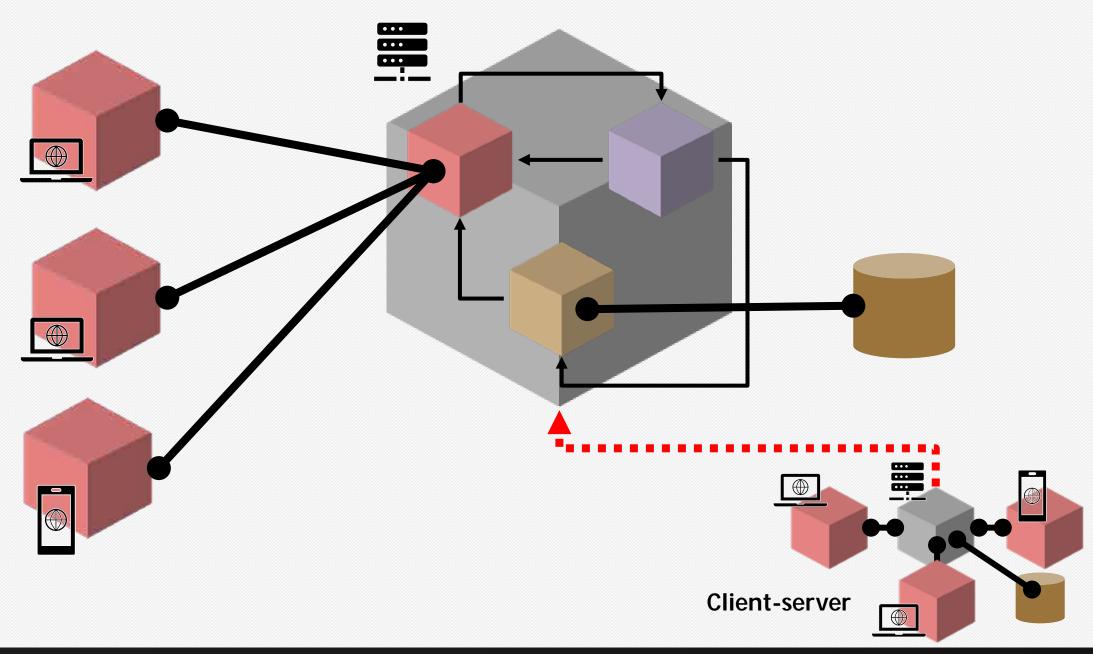


Patrones arquitectónicos

Esquema de organización estructural que describe los elementos y sus relaciones a soluciones de problemas de estilos arquitectónicos.

También llamados **arquetipos**, ofrecen una solución a un **problema de estilo arquitectónico de software**.

Architectural Pattern



Patrones arquitectónicos

Mi crokernel

PM

Presentation-Model

SCA

Sensor-Controller-Actuator

MVP

Model-View-Presenter

MVC

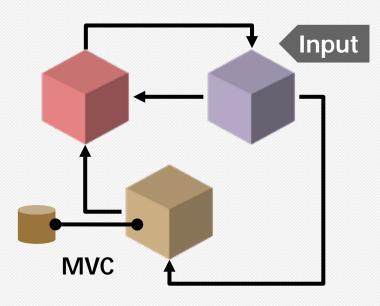
Model - View-Controller

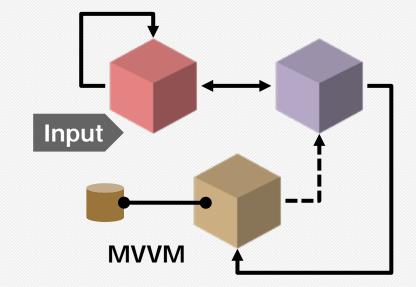
MVVM

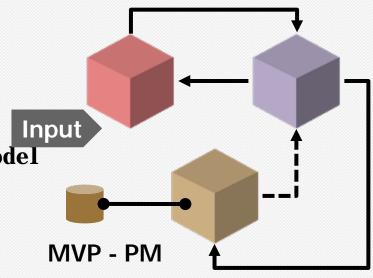
Mbdel-View-ViewMbdel

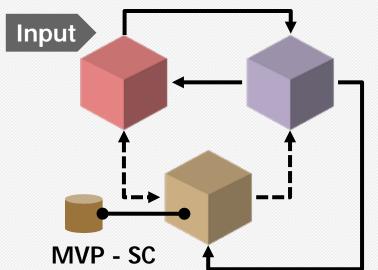
MVPVM

Model - Vi ew-Presenter- Vi ewModel





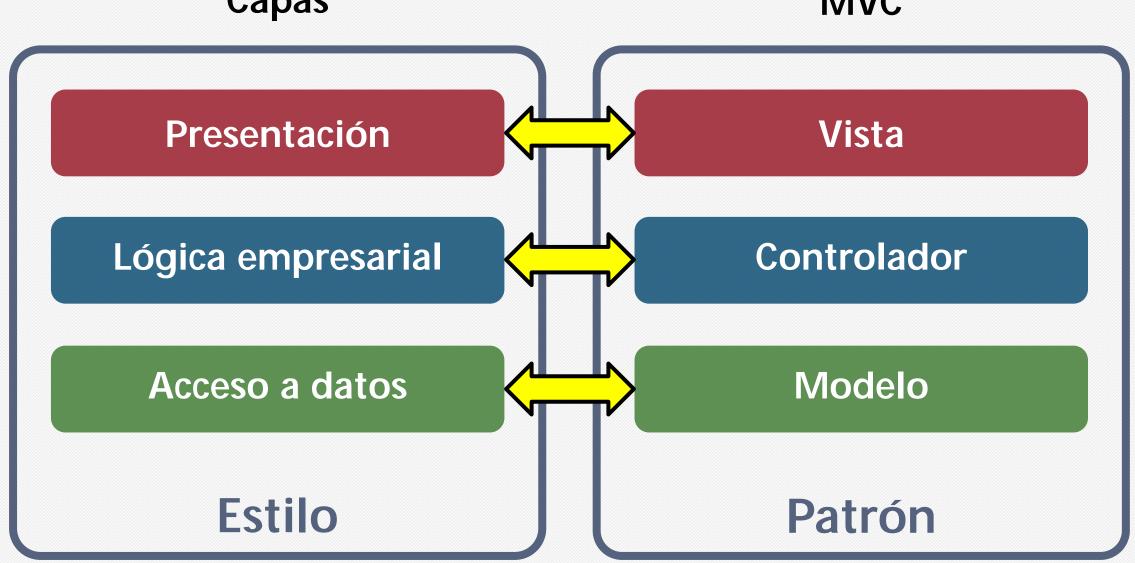




Arquitectura

Estilo Arquitectónico en Capas

Patrón arquitectónico MVC

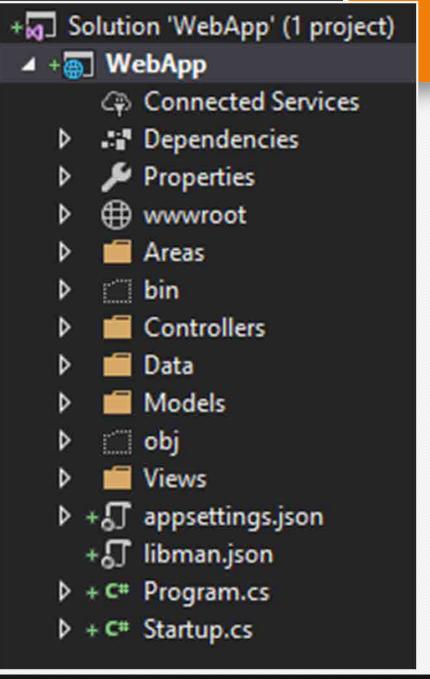


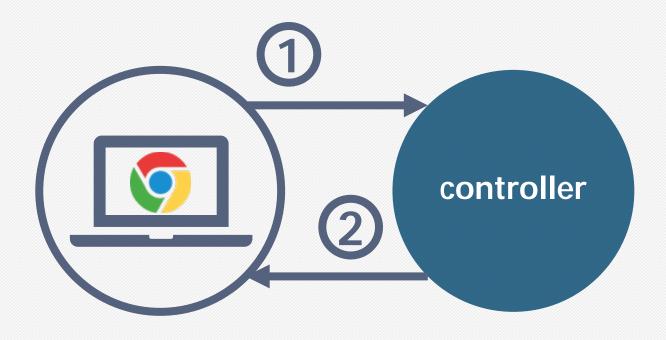


Patrón arquitectónico de diseño MVC Enrutamiento

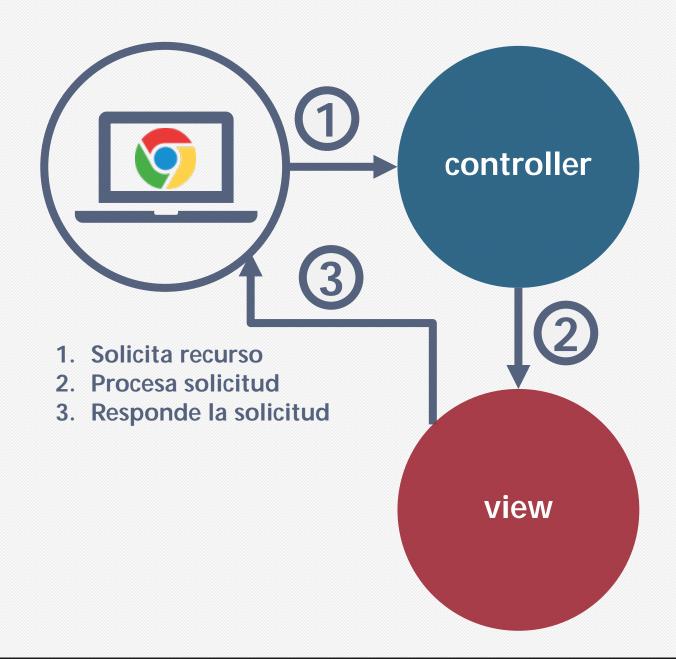
MVC enruta las peticiones al controlador permitiendo aceptar peticiones que no corresponden con ficheros físicos.

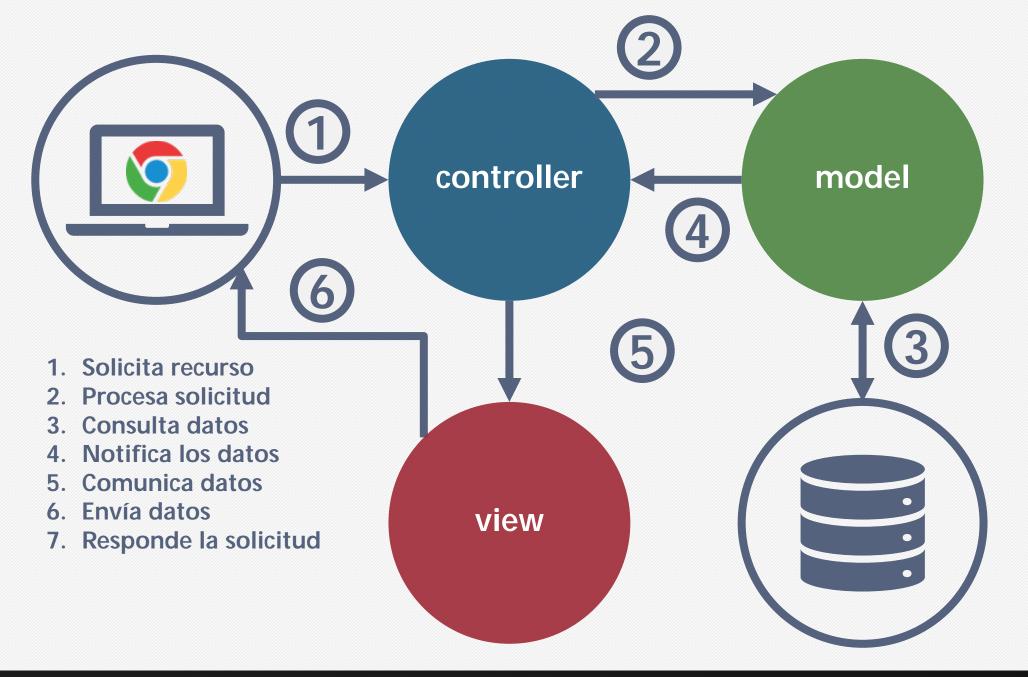
- En WebForms (.Net) y JSP (Java) la url http://localhost/products/index.aspx físicamente existe un fichero index.aspx en la carpeta products del sitio web.
- En MVC http://localhost/products funciona sin que necesariamente exista un fichero, products representa al controlador (programación).





- 1. Solicita recurso
- 2. Responde la solicitud





Enrutamiento en Asp.Net MVC

https://gist.github.com/jlcarrillog/bcdc 974e446be5494a197b01946b90c0







Controllers

C# HomeController.cs





- [@] About.cshtml
- [@] About2.cshtml
- [@] Contact.cshtml
- [@] Contact2.cshtml
- [@] Index.cshtml
- [@] Index2.cshtml

Shared

[@] _Layout.cshtml

http://localhost:5000/Home/Index http://localhost:5000/Home/About http://localhost:5000/Home/Contact

```
public class HomeController : Controller
  public IActionResult Index()
    return View();
  public IActionResult About()
    return View(view: "About");
  public IActionResult Contact()
    return Content("About2");
```







Controllers

- C# HomeController.cs
- C# EquipoController.cs

Views



- [@] About.cshtml
- [@] About2.cshtml
- [@] Contact.cshtml
- [@] Contact2.cshtml
- [@] Index.cshtml
- [@] Index2.cshtml

Shared

[@] _Layout.cshtml

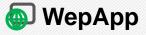
```
http://localhost:5000/Equipo/Index
http://localhost:5000/Equipo/About
http://localhost:5000/Equipo/Contact
```

```
public class EquipoController: Controller
 public IActionResult Index()
   return View(view: "About");
 public IActionResult About()
   return Content("Contact3");
 public IActionResult Contact()
    return View();
```

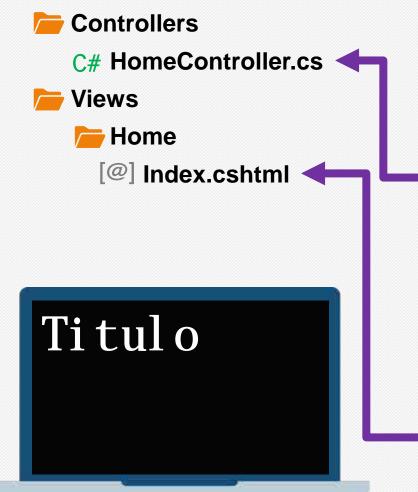
Envío de modelo a vista

https://gist.github.com/jlcarrillog/cc57 0841c8c40c0a53d984e9c08799de









Vista: ~/Home/Index Muestra la vista identificada con Index, la cual contiene un titular h1 con el texto Titulo

```
public class HomeController : Controller
 public IActionResult Index()
    return View();
<h1>Ti tul o</h1>
```







```
public class HomeController : Controller
  Controllers
  C# HomeController.cs
                               public IActionResult Index()
Views
                                 String t = "Hola";
   Home
    [@] Index.cshtml
                                 return View(model: t);
Titulo
Hol a
                             @model String
                             <h1>Ti tul o</h1>
                             @Model
```

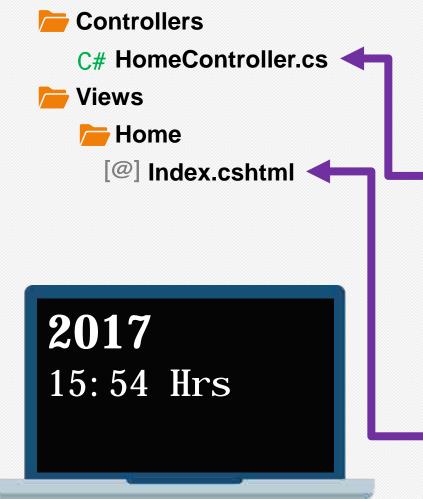


```
public class HomeController : Controller
  Controllers
  C# HomeController.cs
                               public IActionResult Index()
Views
                                 int suma = 5 + 3;
   Home
    [@] Index.cshtml
                                 return View(model: suma);
Titulo
                             @model int
8
                             <h1>Ti tul o</h1>
                             @Model
```









Vista: ~/Home/Index En este ejemplo se puede observar el uso de programación de alto nivel en el HTML

```
public class HomeController : Controller
  public IActionResult Index()
    DateTime fecha = DateTime. Now;
    return View(model: fecha);
@model DateTime
```

<h1>@Model.ToString("yyyy")</h1>

@Model.ToString("HH: mm") Hrs



Controlador

En el **controlador Home** y en la **acción Index** y mostrar la fecha del sistema mediante un **Modelo** con el siguiente formato:

jercicio

Ciudad de Mexico a 01 de marzo de 2020