



UNIVERSIDAD DE LOS LAGOS

DIPLOMADO LENGUAJE DE PROGRAMACIÓN C#

MÓDULO 1 - ARQUITECTURA

CLASE 2 - PROGRAMACIÓN ORIENTADA A OBJETOS

JOEL TORRES CARRASCO
CRISTHIAN AGUILERA CARRASCO
CRISTIAN VALLEJOS VEGA

DEPARTAMENTO DE CIENCIAS DE LA INGENIERÍA
INGENIERÍA CIVIL EN INFORMÁTICA

Campus Osorno

Av. Fuchslocher 1305
Teléfono +56 64 2333 000
Fax +56 64 2333 774
Osorno, Chile

Campus Puerto Montt

Camino a Chingihue Km 6
Teléfono +56 65 2322 536
Puerto Montt, Chile

Sede Santiago

República 517
Barrio Universitario
Teléfono +56 02 2675 3057
Santiago, Chile

Sede Chiloé

Ubaldo Mansilla Barrientos 131
Teléfono 56 65 2322 409
Castro, Chile
Eleuterio Ramírez 348
Teléfono +56 65 2322 476
Ancud, Chile



5 UNIVERSIDAD ACREDITADA
Nivel de 2011 / Revalidada en 2016
Criterio internacional. Sistema de Aseguramiento
de la Calidad - Indicador para el 2011
AVANZADA

www.ulagos.cl

TABLA DE CONTENIDO

- 1 Modelo Tradicional del Desarrollo
- 2 Introducción al patrón MVC
- 3 Modelo
- 4 Vista
- 5 Controlador
- 6 Implementación

SECCIÓN SIGUIENTE

1 Modelo Tradicional del Desarrollo

2 Introducción al patrón MVC

3 Modelo

4 Vista

5 Controlador

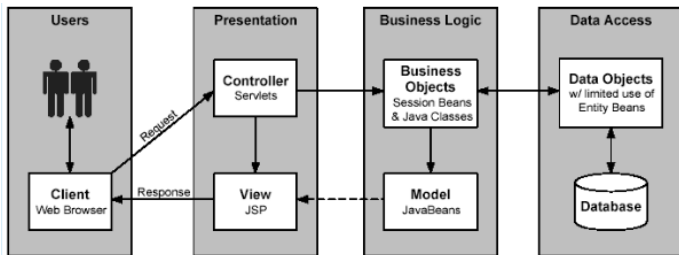
6 Implementación

MODELO TRADICIONAL DE DESARROLLO

Modelo de 3 niveles

Es un modelo utilizado para separar las funcionalidades de la solución entre el: Acceso a los datos, Lógica de negocio y Presentación.

- ▶ **Acceso a los datos:** Corresponde al software responsable de entablar la conexión a la base de datos y permitir la comunicación.
- ▶ **Lógica del Negocio:** Corresponde al software que recibe, analiza y procesa la información relacionada con el problema.
- ▶ **Presentación:** Corresponde al software que permite visualizar la información dados por la lógica de negocio.



SECCIÓN SIGUIENTE

1 Modelo Tradicional del Desarrollo

2 Introducción al patrón MVC

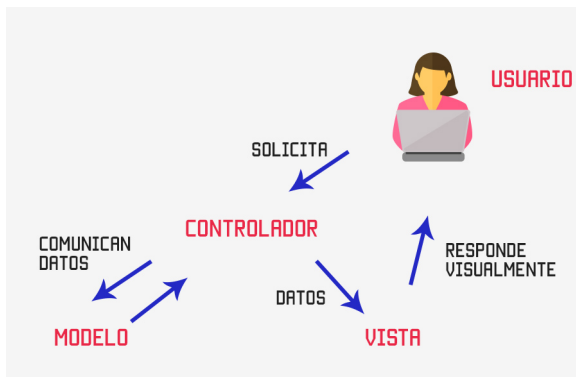
3 Modelo

4 Vista

5 Controlador

6 Implementación

INTRODUCCIÓN AL PATRÓN MVC



Es un patrón de arquitectura llamado **Modelo - Vista - Controlador (MVC)** que separa el desarrollo de la lógica de negocio y de presentación en tres componentes lógicos.

Tradicionalmente, ha sido desarrollada para soluciones que involucran **interfaces gráficas de usuario (GUIs)**. Hoy en día, ha llegado a ser un patrón popular para el desarrollo de soluciones en diseño web y aplicaciones móviles.

INTRODUCCIÓN AL PATRÓN MVC

Características

- ▶ Esta arquitectura fue discutida en 1979 por Trygve Reenskaug, pero fue utilizada por primera vez en el lenguaje de programación Smalltalk en 1987.
- ▶ Es un framework altamente testeable, extensible y adaptable a otras soluciones.
- ▶ Actualmente, en el desarrollo de soluciones web, ofrece control total sobre la estructura y URLs.
- ▶ Varios frameworks de desarrollo de software web ya incorporan MVC, tales como ASP.NET con C#, CakePHP, Laravel, Symphony o Yii en PHP, CherryPy o Django en Python, Ruby on Rails en Ruby, Spring MVC en Java.
- ▶ Separa la lógica del negocio, la forma de interactuar con el usuario, el ingreso de datos y la visualización de los datos.
- ▶ Soporta el uso de metodologías de desarrollo como Test Driven Development (TDD)

SECCIÓN SIGUIENTE

1 Modelo Tradicional del Desarrollo

2 Introducción al patrón MVC

3 Modelo

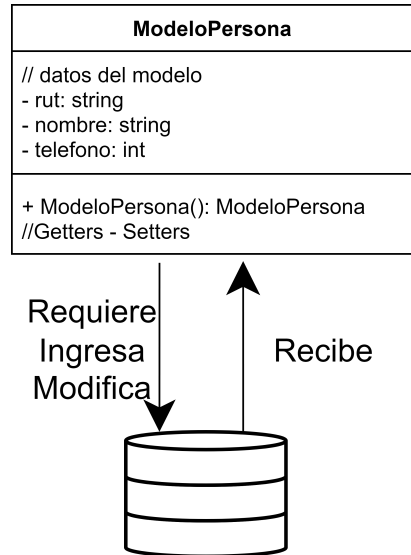
4 Vista

5 Controlador

6 Implementación

MODELO

Un **Modelo** incluye toda la información de la solución y su lógica relacionada. Entonces, este componente almacena datos y procedimientos. Se representan los datos que son transferidos desde el componente **Controlador** o cualquier otro procedimiento de lógica de negocio de la solución. El modelo responde a los requerimientos desde las **Vistas** y también responde a las instrucciones desde el **Controlador** para actualizarse continuamente. Finalmente, es el componente de más bajo nivel del patrón, ya que es el responsable del mantenimiento y conexión a la base de datos.



SECCIÓN SIGUIENTE

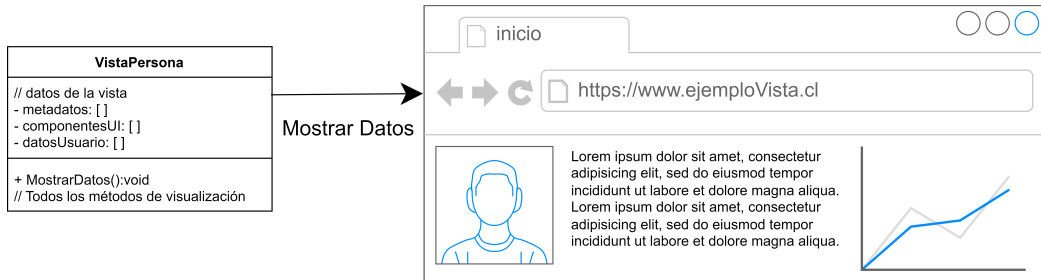
- 1 Modelo Tradicional del Desarrollo
- 2 Introducción al patrón MVC
- 3 Modelo
- 4 Vista**
- 5 Controlador
- 6 Implementación

VISTA

Una **Vista** es la parte de la aplicación que implementa la presentación de los datos al usuario final.

Las **Vistas** son creadas para visualizar los datos recolectados por el **modelo**. Entonces, una **Vista** requiere que el **Modelo** entregue un conjunto de información para que sea presentada en algún formato requerido por la solución.

Esta presentación puede ser realizada a través de gráficos, diagramas y tablas. Además, se puede incluir formularios con componentes visuales (UI) como *text boxes*, *dropdowns* y *buttons*, entre otros.



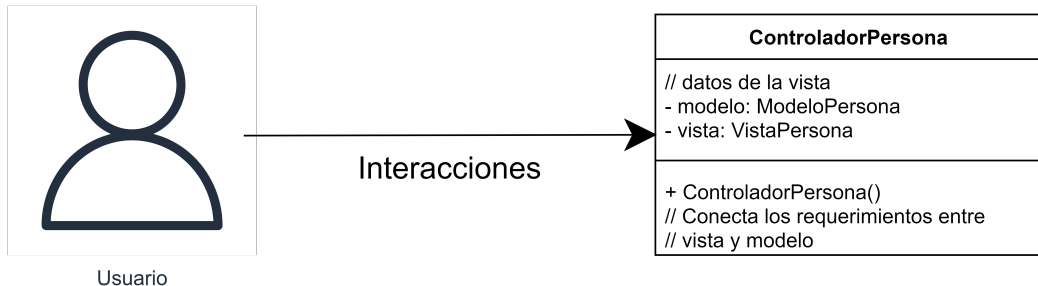
SECCIÓN SIGUIENTE

- 1 Modelo Tradicional del Desarrollo
- 2 Introducción al patrón MVC
- 3 Modelo
- 4 Vista
- 5 Controlador**
- 6 Implementación

CONTROLADOR

El **Controlador** es la parte de la aplicación que maneja la interacción con el usuario a través de la interpretación de los eventos del mouse y teclado, informando al **Vista** sobre los cambios visuales que se deben realizar y entregando los datos provistos por el usuario hacia el **Modelo**.

Un **Controlador** ordena actualizar el estado del modelo. También, ordena que la vista realice los cambios en la presentación de los datos.



SECCIÓN SIGUIENTE

- 1 Modelo Tradicional del Desarrollo
- 2 Introducción al patrón MVC
- 3 Modelo
- 4 Vista
- 5 Controlador
- 6 Implementación**

IMPLEMENTACIÓN DEL MODELO-VISTA-CONTROLADOR

