

Centro Asociado Palma de Mallorca

Introducción

Práctica de

Programación Java



Antonio Rivero Cuesta

Sesión

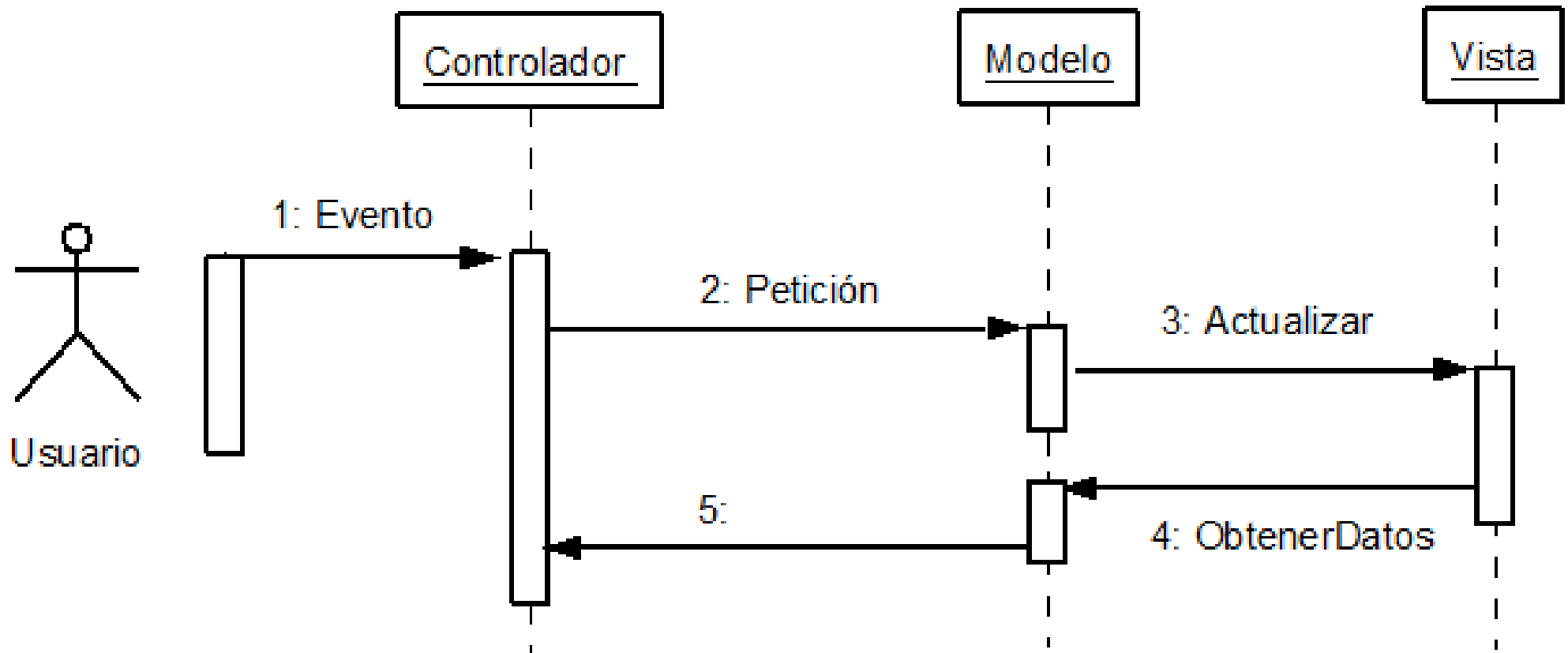
XII

Contenido

Modelo Vista Controlador.....	7
Modelo	9
Vista	11
Controlador	12
Ciclo de vida del MVC.....	14
Ventajas de MVC	17
Desventajas de MVC.....	18

Implementar el MVC 19

Modelo
Vista
Controlador



Modelo Vista Controlador

Es un patrón de diseño de software utilizado para implementar sistemas donde se requiere el uso de *interfaces de usuario*.

Surge de la necesidad de crear software más robusto con un ciclo de vida más adecuado.

Se potencia la facilidad de mantenimiento, reutilización del código y la separación de conceptos.

Su fundamento es la separación del código en tres capas diferentes, acotadas por su responsabilidad:

- Modelo.
- Vista.
- Controlador.

Modelo

Programaremos lo relativo a los *datos* y su encapsulación.

Contiene la lógica de la aplicación.

Es la lógica de negocios.

Son las clases y métodos que se comunican directamente con la base de datos.

Contendrá mecanismos para acceder a la información y también para actualizar su estado.

Los datos los tendremos habitualmente en una base de datos.

En el modelo tendremos todas las funciones que accederán a las tablas y harán los correspondientes SELECT, UPDATE, INSERT, DELETE.

Vista

Programaremos lo relativo a la interfaz gráfica.

Muestra la información al usuario e interactúa con él.

En la vista generalmente trabajamos con los datos.

Sin embargo, no se realiza un acceso directo a éstos.

Las vistas requerirán los datos a los modelos y ellas se generarán las salidas, tal como nuestra aplicación requiera.

Controlador

Es el intermediario entre la Vista y el Modelo.

Se encarga de controlar las interacciones del usuario en la vista.

Contiene el código necesario para responder a las acciones que se solicitan en la aplicación.

Si se solicita desarrollar un sistema de administración de usuarios con un **CRUD** (Create, Read, Update, Delete).

Pide los datos al modelo y los devuelve de nuevo a la vista para que ésta los muestre al usuario.

Es decir las llamadas a clases y métodos, y los datos recibidos de formularios.

La conexión entre la parte gráfica y los datos y los *eventos* que se producen cuando manejamos la aplicación.

Ciclo de vida del MVC

El usuario realiza una petición.

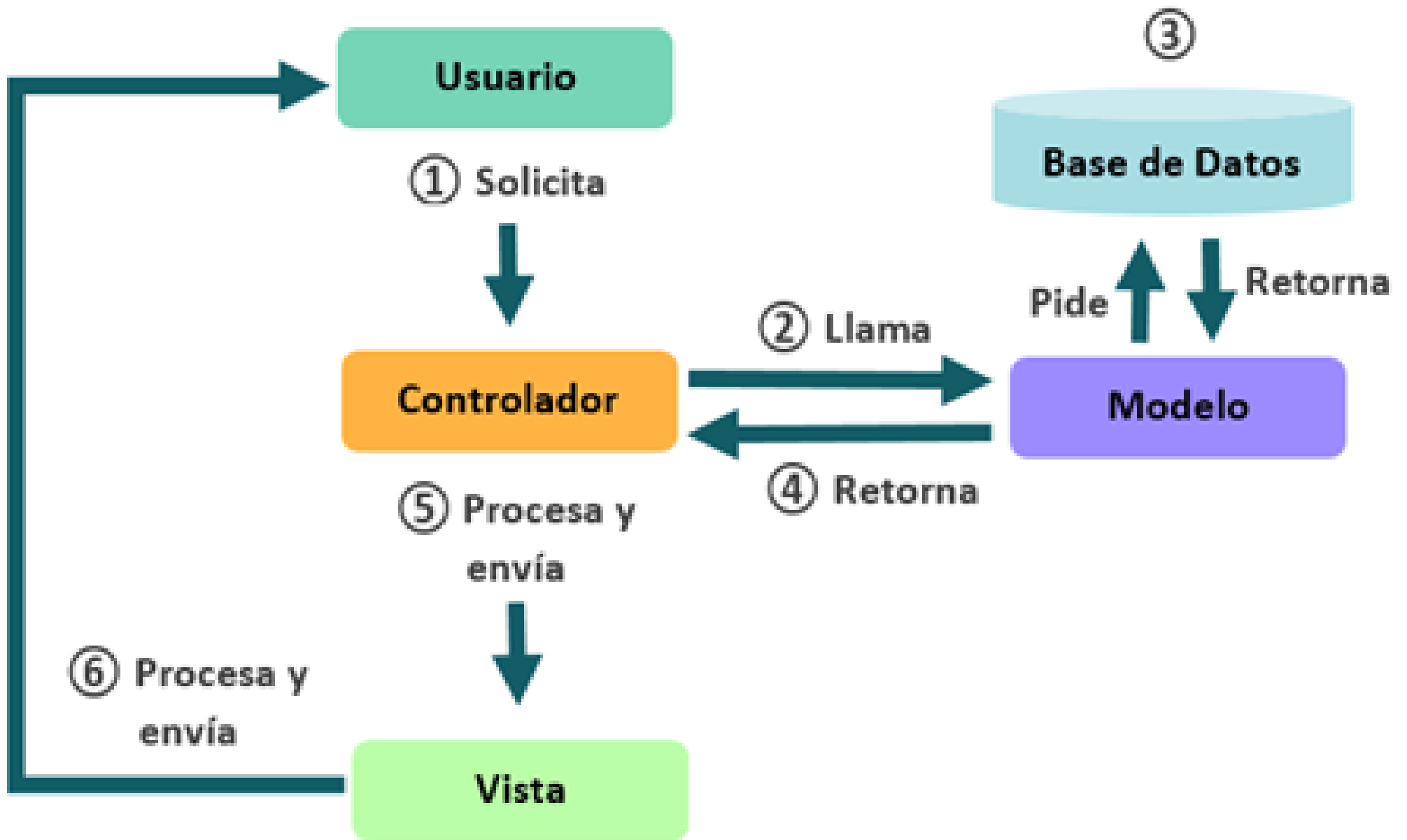
El controlador captura la petición.

Hace la llamada al modelo correspondiente.

El modelo será el encargado de interactuar con la base de datos y retorna la información al controlador.

El controlador recibe la información y la envía a la vista.

- La vista procesa la información recibida y la entrega de una manera visualmente entendible al usuario.



Ventajas de MVC

- La separación del *Modelo* y la *Vista*, lo cual logra separar los datos, de su representación visual.
- Facilita el manejo de errores.
- Permite que el sistema sea escalable si es requerido.
- Es posible agregar múltiples representaciones de los datos.

Desventajas de MVC

- La cantidad de archivos que se deben mantener incrementa considerablemente.
- La curva de aprendizaje es más alta que utilizando otros modelos.
- Su separación en capas, aumenta la complejidad del sistema.

Implementar el MVC

