



Introducción a MVC

.....
De la teoría a la práctica

Modelo

El modelo define qué datos debe contener la aplicación. Si el estado de estos datos cambia, el modelo generalmente notificará a la vista (para que la pantalla pueda cambiar según sea necesario) y, a veces, el controlador (si se necesita una lógica diferente para controlar la vista actualizada).

ClientModel

- Id
- Name
- Lastname
- ...

```
class ClientModel {  
    constructor(name) {  
        this.name = name;  
    }  
    ...  
}
```

NO CONFUNDIR LOS DATOS CON EL MODELO EN SI

Vista

La vista define cómo se deben mostrar los datos de la aplicación.

MyView

- showDetails()
- updateInfo()
- setClearDate()
- ...

```
class MyView {  
  constructor() {  
    this.app = this.getElement('#root')  
    this.form = this.createElement('form')  
  }  
  displayTodos(todos) {  
    ...  
  }  
}
```

¿Cuál agregarías?

Controlador

El controlador contiene una lógica que actualiza el modelo y/o vista en respuesta a las entradas de los usuarios de la aplicación.

MyController

- `handleNameChange()`
- `onAgeChange()`
- `hideDetails()`
- ...

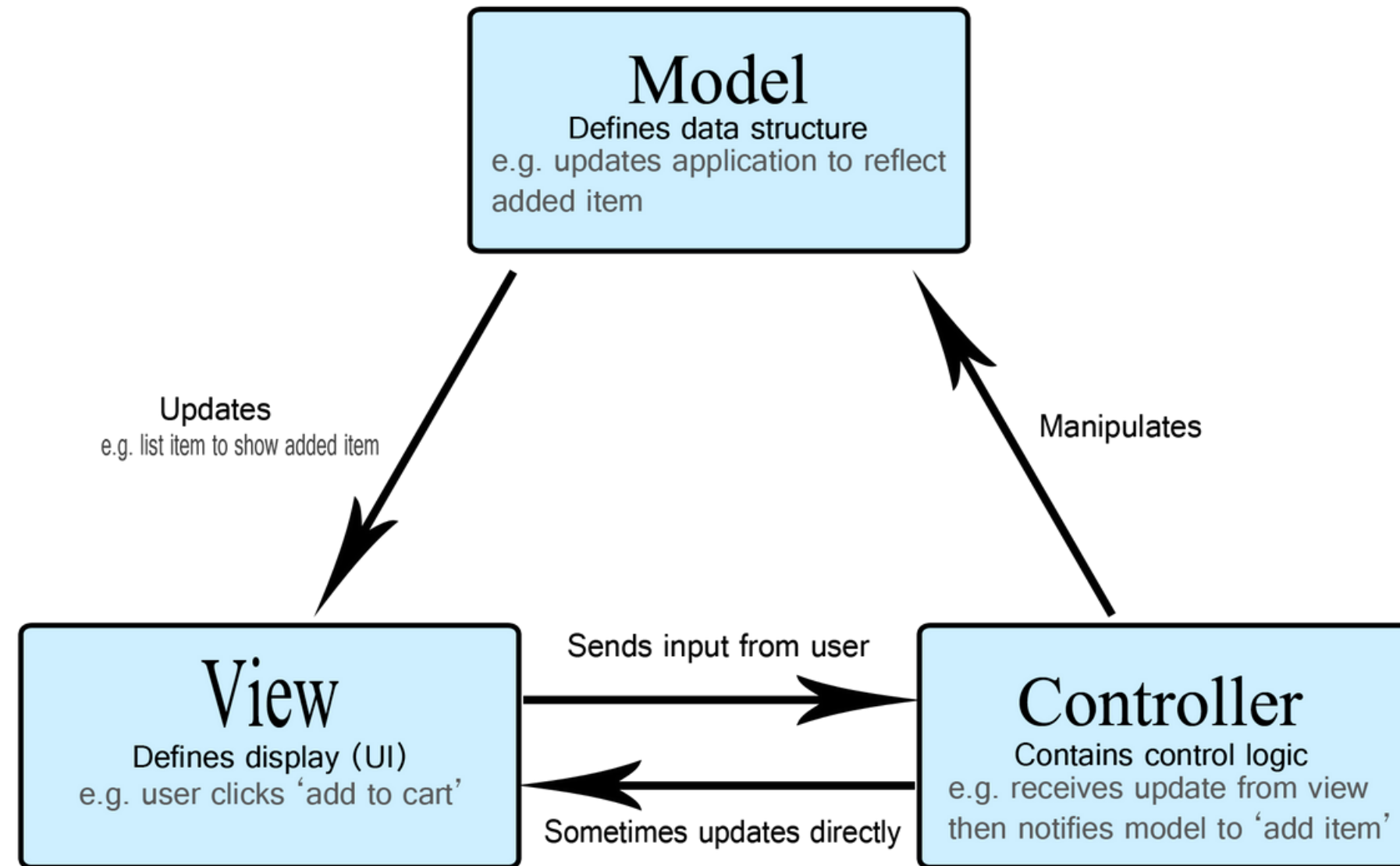
***Uno no debe ir en nuestra lista**

```
class MyController {  
  constructor(model, view) {  
    this.model = model  
    this.view = view  
  }  
  ...  
}
```

Actividad: Lista de tareas

- Completar el código del ejercicio.
- Descomentar partes del código indicado.
- Comprobar y entender el funcionamiento MVC

Flujo MVC



Gracias por tu atención