

## DOCUMENTACIÓN TÉCNICA

### Estructura general del despliegue de la práctica Hola Mundo

Todo la práctica se compone de un sistema MVC(Modelo, Vista, Controlador) de esta forma:

- **Controladores** (HomeController, ReporteController, MedicoController, PacienteController, ExamenController)
- **Modelos** (Médico, Paciente, Orden, ResultadoOrina, ResultadoHeces, Database)
- **Vistas**
  1. Vista exámenes (create\_heces, create\_orina, edit\_heces, edit\_orina, detalle\_heces, detalle\_orina, index)
  2. Vista home (index)
  3. Vista médico (create, edit, index)
  4. Vista Paciente (create, edit, index)
  5. Vista Reportes (visitas)
  6. Vista layouts (footer, header)
- Se cuenta con una paquete config, que contiene una clase llamada database que tiene toda la configuración de la base de datos (host, dbname, user, pass, charset), está también otro paquete llamado core que tiene una clase Database que usa la clase config dataBase para poder hacer la conexión.
- Clase Database:
  1. Tiene una propiedad estática donde guardamos la única conexión PDO
  2. Tiene un constructor privado para que nadie pueda hacer new Database()
  3. Cuenta con un método público y estático que devuelve la conexión
  4. Si la conexión es null quiere decir que todavía no se ha creado la conexión, y se crea una conexión
  5. Luego lee la configuración de la clase database
  6. Creamos la dirección hacia esa base de datos
  7. Luego creamos la conexión con el user y el password
  8. Usamos un try..catch para capturar la excepción si hay algún error de conexión y mostrar el error
  9. Luego si ya hay una conexión retornamos siempre la misma conexión
  10. Creamos un método privado para evitar la clonación de la instancia Singleton

## Modelos

Los modelos encapsulan toda la lógica de datos y conexión a la Base de Datos .

- **Database:**
  1. Clase singleton que asegura que solo haya una conexión PDO en todo el sistema.
  2. Todos los demás modelos dependen de ella
- **Médico / Paciente**
  1. Tienen métodos CRUD: all, find, create, update, delete.
  2. Internamente, cada uno abre conexión vía Database::getConnection().
- **Orden**
  1. Representa las órdenes de exámenes.
  2. Se relaciona con un paciente (obligatorio) y un médico (opcional).
  3. Métodos: crear orden, buscar orden, listar todas, y un reporte visitasPorPaciente.
- **ResultadoOrina / ResultadoHeces**
  1. Cada uno se enlaza 1 a 1 con una orden.
  2. Métodos: create, find, updateByOrden.
  3. Así se asegura que cada orden tenga solo un resultado de orina o sólo un resultado de heces.

## Controladores

Los controladores no guardan datos, solo tienen métodos que:

- Reciben la petición (create, edit, delete).
- Usan los modelos para acceder a la base de datos.

- Finalmente llaman a una vista (require \_\_DIR\_\_ . '/../views/paciente/index.php').

Por ejemplo la relación entre ellos quedaría de esta forma:

- MedicoController usa el modelo Médico para listar (index), crear (store), actualizar (update) o eliminar (delete).

## **Flujo de la práctica**

1. Un controlador recibe la acción (por ejemplo: "crear examen de orina").
2. Ese controlador llama al modelo correspondiente (Orden + ResultadoOrina).
3. El modelo usa la conexión PDO única (Database).
4. Se guarda en la BD y luego el controlador carga la vista para mostrar el resultado.