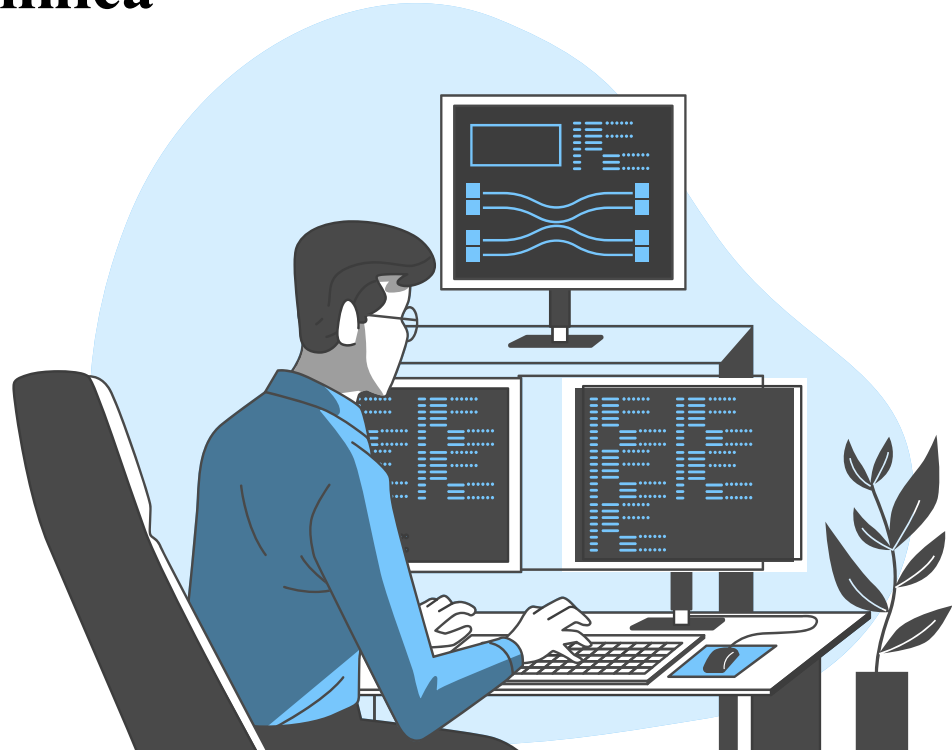


Frameworks para Pruebas Automáticas

Clínica

- Achig Erika
- Wagner Cuenca
- Kevin Quimuña



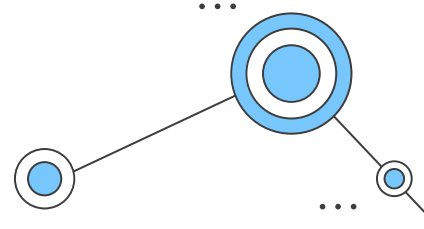


1. Objetivos

1.1 General

- Desarrollar una aplicación que tenga back end y front end que gestione una clínica para el desarrollo de pruebas automáticas.

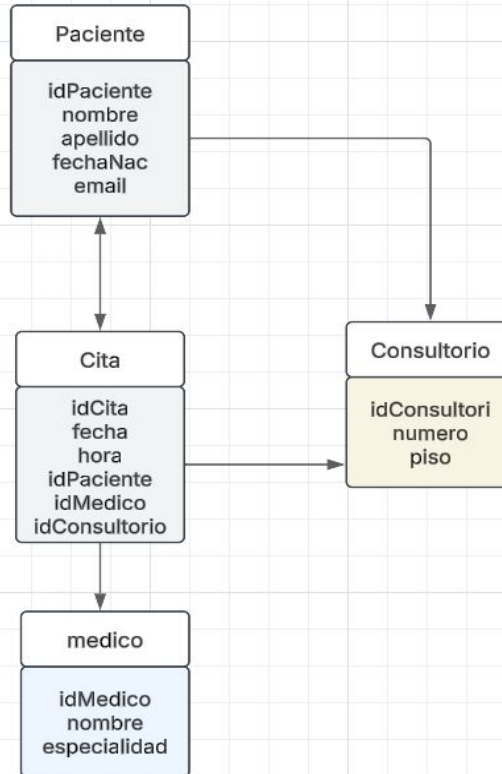
1.2 Específicos

- Desarrollar pruebas automáticas como lo son las pruebas de estrés para ver el rendimiento de la aplicación.
 - Aplicar pruebas de integración para verificar cómo interactúan los módulos de la aplicación.
 - Realizar pruebas para la creación, modificación y eliminación de registros.
- 

...

Estructura del diseño
diagrama de esquema de base de datos.

Diagrama UML



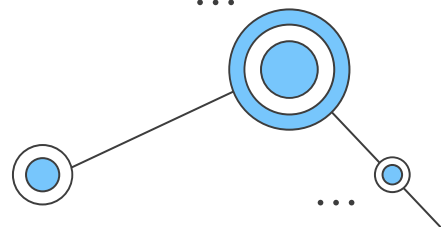
Requerimientos Funcionales

| | |
|-----------------------|--|
| Id. Requerimiento | REQ-001 |
| Nombre | Gestión de Pacientes |
| Registrar pacientes | Información personal (nombre, apellido, fecha de nacimiento, email). |
| Actualizar y eliminar | Datos de pacientes. |
| Consultar | Los detalles de los pacientes registrados. |

| | |
|-------------------|----------------------------|
| Id. Requerimiento | REQ-002 |
| Nombre | Gestión de Médicos |
| Registrar médicos | Datos de médicos. |
| Consultar | Información de los médicos |

| | |
|----------------------|---|
| Id. Requerimiento | REQ-003 |
| Nombre | Gestión de Citas |
| Agendar citas médica | Para pacientes con un médico y fecha específicas. |
| Modificar o cancelar | Citas agendadas. |
| Consultar | Citas programadas para un paciente o médico |

| | |
|------------------------------------|-------------------------|
| Id. Requerimiento | REQ-004 |
| Nombre | Gestión de Consultorios |
| Registrar y gestionar consultorios | Número de piso |
| Modificar o cancelar | Citas agendadas. |
| Asignar consultorios | Citas médicas. |



Configuración del Entorno de Desarrollo:

The screenshot displays the phpMyAdmin web interface. The left sidebar shows a tree view of databases, with 'clinica' selected. The main panel shows the 'Estructura' (Structure) tab for the 'clinica' database. A table list is displayed with columns: Tabla, Acción, Filas, Tipo, Cotejamiento, Tamaño, and Residuo a depurar. Below the table list, there is a section for 'Crear nueva tabla' (Create new table) with fields for 'Nombre de la tabla' and 'Número de columnas'.

| Tabla | Acción | Filas | Tipo | Cotejamiento | Tamaño | Residuo a depurar |
|---------------------------------------|---|----------|---------------|---------------------------|-----------------|-------------------|
| <input type="checkbox"/> citas | ★ Examinar Estructura Buscar Insertar Vaciar Eliminar | 1 | InnoDB | utf8mb4_general_ci | 64.0 KB | - |
| <input type="checkbox"/> consultorios | ★ Examinar Estructura Buscar Insertar Vaciar Eliminar | 2 | InnoDB | utf8mb4_general_ci | 16.0 KB | - |
| <input type="checkbox"/> medicos | ★ Examinar Estructura Buscar Insertar Vaciar Eliminar | 1 | InnoDB | utf8mb4_general_ci | 16.0 KB | - |
| <input type="checkbox"/> pacientes | ★ Examinar Estructura Buscar Insertar Vaciar Eliminar | 1 | InnoDB | utf8mb4_general_ci | 32.0 KB | - |
| 4 tablas | Número de filas | 5 | InnoDB | utf8mb4_general_ci | 128.0 KB | 0 B |

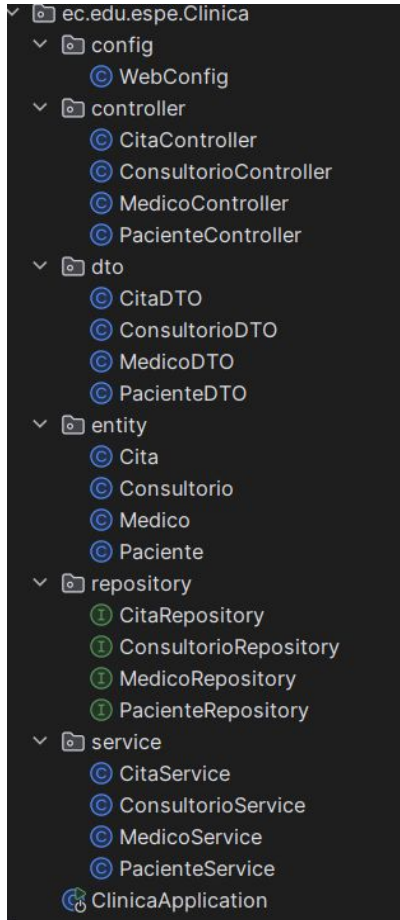
↑ ☐ Seleccionar todo Para los elementos que están marcados: ▾

Imprimir Diccionario de datos

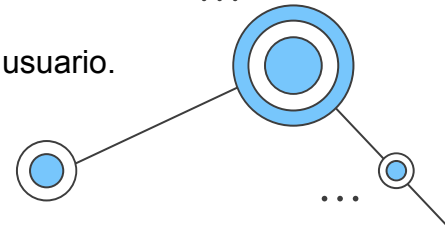
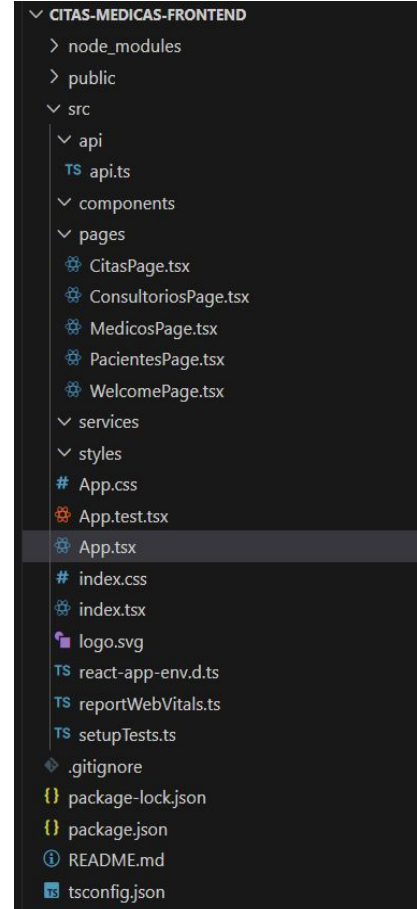
Crear nueva tabla

Nombre de la tabla: Número de columnas: **Crear**

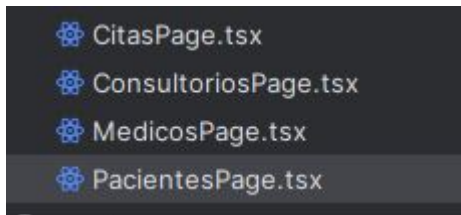
Vistas de clases



Implementar las vistas y la lógica de la interfaz de usuario.



Desarrollo



paquetes

```
{
  "dependencies": {
    "react": "^17.0.2",
    "react-dom": "^17.0.2"
  },
  "devDependencies": {
    "typescript": "^4.1.2",
    "eslint": "^7.14.0"
  }
}
```

Dependencias

```
const CitasPage: React.FC = () => { Show usages
  const [citas : Cita[], setCitas : React.Dispatch<React.SetStateA... ] = useState<Cita[]>({ initialState: []});
  const [pacientes : Paciente[], setPacientes : React.Dispatch<React.SetStateA... ] = useState<Paciente[]>({ initialState: []});
  const [medicos : Medico[], setMedicos : React.Dispatch<React.SetStateA... ] = useState<Medico[]>({ initialState: []});
  const [consultorios : Consultorio[], setConsultorios : React.Dispatch<React.SetStateA... ] = useState<Consultorio[]>({ initialState: []});
  const [modalMode : "crear" | "editar" | "borrar" ..., setModalMode : React.Dispatch<React.SetStateA... ] = useState<"crear" | "editar" | "borrar" | null>({ initialState: null});
  const [selectedCita : Cita | null, setSelectedCita : React.Dispatch<React.SetStateA... ] = useState<Cita | null>({ initialState: null});
  const [nuevaCita : Cita, setNuevaCita : React.Dispatch<React.SetStateA... ] = useState<Cita>({ initialState: {
```

Lógica