[C2-Q1: CONSTRUIR LAYOUT A PARTIR DE COMPONENTES](https://virtual.corhuila.edu.co/uh/mod/assign/view.php?id=45571)

**Enfermero:**

**Funcionalidad del componente**

1. **Formulario de entrada de datos**:
   * El formulario permite agregar o modificar la información de un enfermero.
   * Los campos del formulario incluyen:
     + Nombre
     + Apellido
     + Edad
     + Correo
     + Teléfono
     + Dirección
     + **Turno Asignado**
     + **Área de Atención**
   * Los datos ingresados se vinculan al objeto nuevoEnfermero mediante v-model.
2. **Operaciones CRUD**:
   * **Agregar**: Al enviar el formulario, si no está en modo edición, se agrega un nuevo enfermero a la lista enfermeros.
   * **Editar**: Al hacer clic en el botón "Editar" de un enfermero, los datos se cargan en el formulario para ser modificados.
   * **Modificar**: Si el formulario está en modo edición, los cambios realizados se guardan en la lista.
   * **Eliminar**: Permite eliminar un enfermero de la lista.
3. **Visualización de datos**:
   * Los enfermeros se muestran en una tabla con columnas correspondientes a los campos del formulario.
   * Cada fila incluye botones para editar o eliminar el enfermero.
4. **Interactividad**:
   * Botón para alternar entre mostrar y ocultar la lista de enfermeros.
   * Validación de campos obligatorios mediante el atributo required.

**Justificación de los campos adicionales**

1. **Turno Asignado**:
   * Este campo es relevante para identificar el horario en el que el enfermero trabaja (por ejemplo, turno diurno, nocturno, etc.).
   * Es esencial para la planificación de turnos y garantizar que siempre haya personal disponible en las diferentes franjas horarias.
2. **Área de Atención**:
   * Este campo especifica la sección o departamento en el que el enfermero presta sus servicios (por ejemplo, urgencias, pediatría, UCI, etc.).
   * Es importante para asignar responsabilidades y coordinar al personal según las necesidades específicas de cada área.

**Medico:**

**Funcionalidad del componente**

1. **Formulario de entrada de datos**:
   * Permite agregar o modificar la información de un médico.
   * Los campos del formulario incluyen:
     + Nombre
     + Apellido
     + Edad
     + Correo
     + Teléfono
     + Dirección
     + **Especialidad**
     + **Número de Licencia**
   * Los datos ingresados se enlazan al objeto nuevoMedico mediante v-model.
2. **Operaciones CRUD**:
   * **Agregar**: Al enviar el formulario, si no está en modo edición, se agrega un nuevo médico a la lista medicos.
   * **Editar**: Al hacer clic en el botón "Editar", los datos del médico seleccionado se cargan en el formulario para ser modificados.
   * **Modificar**: Si el formulario está en modo edición, los cambios realizados se guardan en la lista.
   * **Eliminar**: Permite eliminar un médico de la lista.
3. **Visualización de datos**:
   * Los médicos se muestran en una tabla con columnas correspondientes a los campos del formulario.
   * Cada fila incluye botones para editar o eliminar al médico.
4. **Interactividad**:
   * Botón para alternar entre mostrar y ocultar la lista de médicos.
   * Validación de campos obligatorios mediante el atributo required.

**Justificación de los campos adicionales**

1. **Especialidad**:
   * Este campo es fundamental para identificar el área de conocimiento o práctica médica del profesional (por ejemplo, cardiología, pediatría, neurología, etc.).
   * Es crucial para asignar médicos a casos específicos según las necesidades de los pacientes y garantizar una atención especializada.
2. **Número de Licencia**:
   * Este campo asegura que el médico cuenta con una licencia válida para ejercer la medicina, lo cual es un requisito legal y ético.
   * Permite llevar un registro formal y verificar la autenticidad de los profesionales médicos.

**Paciente:**

**Funcionalidad del componente**

1. **Formulario de entrada de datos**:
   * Permite agregar o modificar la información de un paciente.
   * Los campos del formulario incluyen:
     + Nombre
     + Apellido
     + Edad
     + Correo
     + Teléfono
     + Dirección
     + **Número de Historia Clínica**
     + **Tipo de Afiliación**
   * Los datos ingresados se enlazan al objeto nuevoPaciente mediante v-model.
2. **Operaciones CRUD**:
   * **Agregar**: Al enviar el formulario, si no está en modo edición, se agrega un nuevo paciente a la lista pacientes.
   * **Editar**: Al hacer clic en el botón "Editar", los datos del paciente seleccionado se cargan en el formulario para ser modificados.
   * **Modificar**: Si el formulario está en modo edición, los cambios realizados se guardan en la lista.
   * **Eliminar**: Permite eliminar un paciente de la lista.
3. **Visualización de datos**:
   * Los pacientes se muestran en una tabla con columnas correspondientes a los campos del formulario.
   * Cada fila incluye botones para editar o eliminar al paciente.
4. **Interactividad**:
   * Botón para alternar entre mostrar y ocultar la lista de pacientes.
   * Validación de campos obligatorios mediante el atributo required.

**Justificación de los campos adicionales**

1. **Número de Historia Clínica**:
   * Este campo es esencial para identificar de manera única el expediente médico del paciente.
   * Permite acceder rápidamente a los antecedentes médicos, diagnósticos, tratamientos y otros datos relevantes para brindar una atención adecuada.
2. **Tipo de Afiliación**:
   * Este campo especifica el tipo de cobertura médica del paciente (por ejemplo, seguro público, privado, subsidiado, etc.).
   * Es importante para gestionar los servicios médicos según las condiciones de afiliación y garantizar que el paciente reciba los beneficios correspondientes.

**Recepcionista:**

**Funcionalidad del componente**

1. **Formulario de entrada de datos**:
   * Permite agregar o modificar la información de un recepcionista.
   * Los campos del formulario incluyen:
     + Nombre
     + Apellido
     + Edad
     + Correo
     + Teléfono
     + Dirección
     + **Horario laboral**
     + **Extensión telefónica**
   * Los datos ingresados se enlazan al objeto nuevoRecepcionista mediante v-model.
2. **Operaciones CRUD**:
   * **Agregar**: Al enviar el formulario, si no está en modo edición, se agrega un nuevo recepcionista a la lista recepcionistas.
   * **Editar**: Al hacer clic en el botón "Editar", los datos del recepcionista seleccionado se cargan en el formulario para ser modificados.
   * **Modificar**: Si el formulario está en modo edición, los cambios realizados se guardan en la lista.
   * **Eliminar**: Permite eliminar un recepcionista de la lista.
3. **Visualización de datos**:
   * Los recepcionistas se muestran en una tabla con columnas correspondientes a los campos del formulario.
   * Cada fila incluye botones para editar o eliminar al recepcionista.
4. **Interactividad**:
   * Botón para alternar entre mostrar y ocultar la lista de recepcionistas.
   * Validación de campos obligatorios mediante el atributo required.

**Justificación de los campos adicionales**

1. **Horario laboral**:
   * Este campo es esencial para definir las horas de trabajo del recepcionista, lo que permite organizar turnos y garantizar la cobertura en la recepción durante todo el horario de atención.
   * Es importante para coordinar las actividades del personal y evitar conflictos en los horarios.
2. **Extensión telefónica**:
   * Este campo permite asignar un número de extensión único al recepcionista, facilitando la comunicación interna dentro de la institución.
   * Es útil para redirigir llamadas de manera eficiente y garantizar que los pacientes o visitantes puedan comunicarse con la persona adecuada.

EVIDENCIA VISUAL







