



## Preparado para el reto

### Tu Misión

Fuiste contratado por una institución de salud para desarrollar un software basado en Java y que permita gestionar la información de los historiales clínicos de sus pacientes.

Las entidades definidas para este software son:

- **Pacientes:** Es la entidad que representa un paciente afiliado a la entidad de salud. Cada uno de los pacientes deberá estar asignado a su respectivo historial clínico.
- **Médicos:** Es la entidad que representa un médico en la entidad de salud. Cada uno de los médicos deberá estar asignado a respectivos historiales clínicos.
- **Historiales Clínicos:** Es la entidad que representa los historiales clínicos de la entidad de salud, en los cuales se asignan pacientes y médicos para su gestión.

El aplicativo deberá ser desarrollado en Java y deberá tener las siguientes características y funcionalidades:

- El software permitirá registrar, consultar, modificar y eliminar los datos para los atributos de la entidad Paciente, es decir, debes desarrollar una clase "Paciente" donde definas los atributos para dicha entidad, los cuales son: identificación, nombres, apellidos, edad y género.
- El software permitirá registrar, consultar, modificar y eliminar los datos para los atributos de la entidad Medico, es decir, debes desarrollar una clase "Medico" donde definas los atributos para dicha entidad, los cuales son: identificación, nombres, apellidos y especialidad.
- El software permitirá registrar, consultar, modificar y eliminar los datos para los atributos de la entidad Historiales Clínicos, es decir, debes desarrollar una clase "HistorialClinico" donde definas los atributos para dicha entidad, los cuales son: código, fecha, idPaciente, idMedico y observaciones.
- Para cada entidad, se deberá crear los distintos métodos para la manipulación (insertar, eliminar, consultar y modificar) de los atributos de cada entidad incluyendo Getters y Setters. Así como cualquier método que considere necesario.
- Todos los datos de las entidades son cadena de textos excepto el atributo "edad" de la entidad "Paciente", que es numérico de tipo entero.
- El aplicativo debe mostrar un menú principal con las opciones:
  1. Gestionar Pacientes.
  2. Gestionar Médicos.
  3. Gestionar Historial Clínico.
  4. Salir

Luego al entrar a cada entidad se debe mostrar un submenú con las opciones

Chat finalizado



para Pacientes:

1. Registrar Pacientes.
2. Mostrar Pacientes.
3. Modificar Pacientes.
4. Eliminar Pacientes.
5. Atrás

Luego al entrar se debe pedir o mostrar la información según la acción, por ejemplo:

1. Registrar Pacientes: Pedir los datos de un paciente a registrar.
2. Mostrar Pacientes: Mostrar todos los pacientes registrados con sus atributos.
3. Modificar Paciente: Modificar los atributos de un paciente a partir del IdPaciente (Si el idPaciente no existe decir paciente no encontrado.).
4. Eliminar Paciente: Pedir el IdPaciente y eliminarlo (Si el idPaciente no existe decir paciente no encontrado.).
5. Atrás



## Estatus de la entrega

Número de intento	Éste es el intento 1.
Estatus de la entrega	Sin intento
Estatus de calificación	No calificado
Criterio	Desarrollar un software basado en Java y que permita gestionar la

Chat finalizado



para  
calificar

sus pacientes.

Funcionalidad	<p>Experto: (1) Abre el proyecto y compila. Se ejecuta perfectamente. Gestiona pacientes completamente, funciona el menú, es posible listar, agregar y eliminar. (2) Abre el proyecto y compila. Se ejecuta perfectamente. Gestiona médicos completamente. Funciona el menú, es posible listar, agregar y eliminar. (3) Abre el proyecto y compila. Se ejecuta perfectamente. Gestiona Historiales, completamente es posible listar, agregar y eliminar.</p> <p><b>25 puntos</b></p>	<p>Avanzado: (1) Abre, compila y se ejecuta pero tiene clases que no se usan. Gestiona la mayoría de las funciones de pacientes. (2) Abre, compila y se ejecuta pero tiene clases que no se usan. Gestiona la mayoría de las funciones de médicos. (3) Abre, compila y se ejecuta pero tiene clases que no se usan. Gestiona la mayoría de las funciones de Historiales.</p> <p><b>20 puntos</b></p>	<p>Aprendiz: (1) Abre y compila y se ejecuta pero tiene nombres inadecuados y clases de más. Gestiona varias de las funciones de pacientes. (2) Abre y compila y se ejecuta pero tiene nombres inadecuados y clases de más. Gestiona varias de las funciones de médicos. (3) Abre y compila y se ejecuta pero tiene nombres inadecuados y clases de más. Gestiona varias de las funciones de Historiales.</p> <p><b>15 puntos</b></p>	<p>Novato: (1) Abre, pero no compila. Incompleta la Gestión de Pacientes (2) Abre, pero no compila. Incompleta la Gestión de médicos. (3) Abre, pero no compila. Incompleta la Gestión de Historiales.</p> <p><b>5 puntos</b></p>	<p>Incumplido: (1) el proyecto enviado no corresponde a los requerimientos solicitados.</p> <p><b>0 puntos</b></p>
---------------	--	--	---	---	--



<b>Legibilidad y Calidad de Código</b>	<p>Experto: (1) Utiliza herencia adecuadamente para atributos comunes de pacientes y médicos. (2) Implementa los métodos ordenadamente por cada opción del menú. (3) Identifica las clases: Medico, HistorialClinico, Paciente, Personal, Interfaz y maneja clases separadas para las acciones sobre las entidades</p> <p><b>25 puntos</b></p>	<p>Avanzado: (1) Utiliza herencia para la mayoría de los atributos comunes de pacientes o médicos. (2) Implementa métodos por cada opción del menú y no de forma ordenada. (3) Identifica las clases: Medico, HistorialClinico, Paciente, Personal, Interfaz y no maneja clases separadas para las acciones sobre las entidades.</p> <p><b>20 puntos</b></p>	<p>Aprendiz: (1) Utiliza herencia para algunos de los atributos comunes de pacientes o médicos. (2) Tiene métodos para algunas opciones de forma ordenada. (3) Identifica la mayoría de estas clases: Medico, HistorialClinico, Paciente, Personal, Interfaz.</p> <p><b>15 puntos</b></p>	<p>Novato: (1) No utiliza herencia. (2) Tiene métodos para algunas opciones pero no de forma ordenada. (3) La definición de las clases es incompleta.</p> <p><b>5 puntos</b></p>	<p>Incumplido: (1) el proyecto enviado no corresponde a los requerimientos solicitados.</p> <p><b>0 puntos</b></p>
<p><b>Estándares y Buenas Prácticas</b></p>					



<p>Experto: (1) Nombra los elementos, tales como clases y funciones, de manera descriptiva y concisa siempre. (2) Usa siempre nomenclatura CamelCase para atributos y métodos y CapitalCase para clases. (3) Utiliza paquetes de manera adecuada siempre. (4) El código (las clases y funciones declaradas) presentan alta cohesión y bajo acoplamiento. Es decir, que cada clase y función ejecuta aquello para lo que fueron creadas.</p> <p><b>25 puntos</b></p>	<p>Avanzado: (1) Nombra los elementos, tales como clases y funciones, de manera descriptiva y concisa frecuentemente. (2) Usa frecuentemente nomenclatura CamelCase para atributos y métodos y CapitalCase para clases. (3) Utiliza paquetes pero puede separar más adecuadamente. (4) El código (las clases y funciones declaradas) presentan parcialmente alta cohesión y bajo acoplamiento. Es decir, que cada clase y función ejecuta aquello para lo que fueron creadas.</p> <p><b>20 puntos</b></p>	<p>Aprendiz: (1) Nombra los elementos, tales como clases y funciones, de manera descriptiva y concisa algunas veces. (2) Usa algunas veces nomenclatura CamelCase para atributos y métodos y CapitalCase para clases. (3) Utiliza paquetes pero no están separados adecuadamente. (4) El código (las clases y funciones declaradas) presentan poca cohesión y alto acoplamiento.</p> <p><b>15 puntos</b></p>	<p>Novato: (1) No utiliza nombres adecuados. (2) No utiliza nomenclatura estándar. (3) No utiliza paquetes. (4) No implementó separación.</p> <p><b>10 puntos</b></p>	<p>Incumplido: (1) el proyecto enviado no corresponde a los requerimientos solicitados.</p> <p><b>0 puntos</b></p>
---	---	--	---	--

<b>Eficiencia y Robustez</b>	<p>Experto: (1) No presenta redundancias ni código duplicado. (2) No contiene código comentado de más o ignorado. (3) Hace validaciones siempre.</p> <p><b>25 puntos</b></p>	<p>Avanzado: (1) Presenta redundancias algunas veces. (2) Contiene algunas veces código comentado de más o ignorado. (3) Hace validaciones la mayoría de las veces.</p> <p><b>20 puntos</b></p>	<p>Aprendiz: (1) Presenta redundancias muchas veces. (2) Contiene varias veces código comentado de más o ignorado. (3) Hace validaciones algunas veces.</p> <p><b>15 puntos</b></p>	<p>Novato: (1) Siempre es redundante. (2) Contiene muchas veces código comentado de más o ignorado. (3) No realiza validaciones.</p> <p><b>5 puntos</b></p>	<p>Incumplido: (1) el proyecto enviado no corresponde a los requerimientos solicitados.</p> <p><b>0 puntos</b></p>
------------------------------	--	---	---	---	--

Última  
modificación

-

Añadir envío

Hacer cambios a su envío

