

ACTIVIDAD	Evaluación Individual de PROGRAMACIÓN III – Asociaciones entre clases en C++					
DOCENTE	JAIRO ARMANDO RIAÑO HERRERA					
ASIGNATURA	PROGRAMACIÓN III (60)		Cod. Asignatura	8108257	Primer 50%	<input checked="" type="checkbox"/>
TIPO ACTIVIDAD	Individual	<input checked="" type="checkbox"/>	Colaborativa	<input type="checkbox"/>	Número de Semanas	NA
FECHAS	Fecha Inicio	17-Nov-2023		Fecha Finalización	17-Nov-2023	

Competencia:

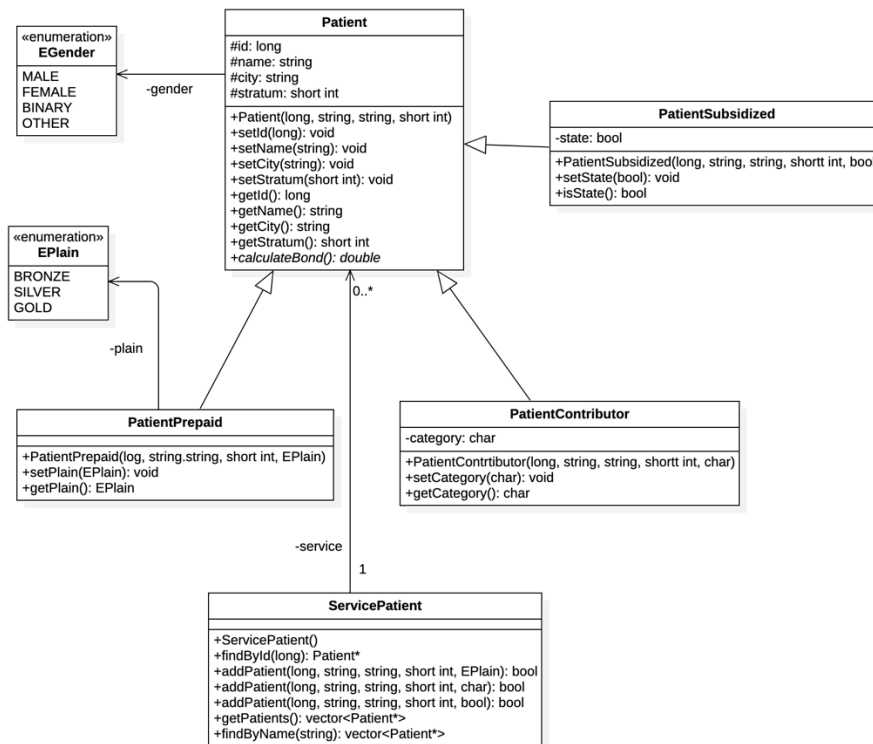
El estudiante implementa el sistema con base en los requisitos dados

Propósitos de Formación:

- ☐ Interpretar Diagramas de Clases para su implementación en un lenguaje POO
- ☐ Usar C++ como lenguaje de programación para la implementación del Diagrama de Clases

Descripción de la Actividad:

En un proyecto en C++, implemente el siguiente Diagrama de Clases:



ACTIVIDAD	Evaluación Individual de PROGRAMACIÓN III – Asociaciones entre clases en C++				
DOCENTE	JAIRO ARMANDO RIAÑO HERRERA				
ASIGNATURA	PROGRAMACIÓN III (60)		Cod. Asignatura	8108257	Primer 50% <input checked="" type="checkbox"/> Segundo 50% <input type="checkbox"/>
TIPO ACTIVIDAD	Individual <input checked="" type="checkbox"/>	Colaborativa <input type="checkbox"/>	Número de Semanas		NA
FECHAS	Fecha Inicio	17-Nov-2023	Fecha Finalización	17-Nov-2023	

MÉTODOS DE LA CLASE DE SERVICIO:

- **findById()** → Busca un paciente por el Id, retorna un apuntador al Paciente, o NULL si no existe
- **findByName()** → Busca un paciente por Nombre, retorna un arreglo (vector) de apuntadores a Paciente, ya que pueden existir varios con el mismo nombre
- **addPatient()** → Método sobrecargado que permite agregar un paciente, en cada uno se instancia el objeto correspondiente al tipo de paciente. Para los pacientes de régimen contributivo, validar que la categoría esté entre A..D. Si no, no se puede agregar. El método retorna verdadero o falso indicando si se puede o no agregar el paciente.
- **getPatients()** → Método que retorna un arreglo (vector) con apuntadores a todos los Pacientes

FUNCIONALIDAD DE LA APLICACIÓN:

- A través de una instancia de la clase de Servicios de Pacientes en el archivo de la aplicación, adicionar uno de cada tipo de paciente.
- Usar el método de cálculo del valor del bono para validar el funcionamiento del Polimorfismo (buscando un paciente para la llamada al método)
 - o Para los pacientes de Medicina Prepagada, el valor es el 20% del SMMV
 - o Para los pacientes de Régimen Contributivo se calcula así:
 - Categoría A → \$4.500
 - Categoría B → \$12.000
 - Categoría C → \$23.500
 - Categoría D → \$43.000
 - o Para los pacientes de Régimen subsidiado el valor del bono es el 1% del SMMV
- Listar todos los pacientes indicando el valor del aporte de cada uno (cálculo del bono)
- Listar pacientes por un mismo nombre.

ACTIVIDAD	Evaluación Individual de PROGRAMACIÓN III – Asociaciones entre clases en C++				
DOCENTE	JAIRO ARMANDO RIAÑO HERRERA				
ASIGNATURA	PROGRAMACIÓN III (60)		Cod. Asignatura	8108257	Primer 50% <input checked="" type="checkbox"/> Segundo 50% <input type="checkbox"/>
TIPO ACTIVIDAD	Individual <input checked="" type="checkbox"/>	Colaborativa <input type="checkbox"/>	Número de Semanas		NA
FECHAS	Fecha Inicio	17-Nov-2023	Fecha Finalización		17-Nov-2023

RÚBRICA DE EVALUACIÓN				
ASPECTOS EVALUADOS	Niveles de Desempeño Individual de la Actividad			PUNTAJE
	VALORACIÓN			
	ALTA	MEDIA	BAJA	
Definición de miembros de las clases	Se define en forma apropiada tanto los atributos, como los métodos, su firma es la especificada en el Diagrama de las Clases.	La definición o firma de los métodos no está conforme a lo especificado en el Diagrama de las Clases.	No es posible validar el ítem	20
	(Hasta 20 Puntos)	(Hasta 10 Puntos)	(0 Puntos)	
Implementación de asociaciones entre las clases	El código de la implementación de las asociaciones es pertinente al tipo de asociación y cardinalidad. Los elementos definidos son los adecuados. Aplica en forma adecuada los elementos de la asociación de generalización o herencia	No se definen los elementos necesarios con base en el tipo y cardinalidad de la asociación. No especifica adecuadamente los elementos de la asociación de generalización o herencia	No es posible validar el ítem	30
	(Hasta 20 Puntos)	(Hasta 10 Puntos)	(0 Puntos)	
Código de la Aplicación	El código en el archivo de aplicación evidencia que se instancia la clase de servicio, se agregan objetos y se envían mensajes al objetos para visualizar el resultado del procesamiento de los datos y se aplica en forma adecuada el polimorfismo	Se instancia la clase de servicio, pero no se agregan objetos. O no hay llamadas o paso de mensajes entre objetos para obtener la información pertinente, no se evidencia la aplicación del polimorfismo	No es posible validar el ítem	20

ACTIVIDAD	Evaluación Individual de PROGRAMACIÓN III – Asociaciones entre clases en C++				
DOCENTE	JAIRO ARMANDO RIAÑO HERRERA				
ASIGNATURA	PROGRAMACIÓN III (60)		Cod. Asignatura	8108257	Primer 50% <input checked="" type="checkbox"/> Segundo 50% <input type="checkbox"/>
TIPO ACTIVIDAD	Individual <input checked="" type="checkbox"/>	Colaborativa <input type="checkbox"/>	Número de Semanas		NA
FECHAS	Fecha Inicio	17-Nov-2023	Fecha Finalización		17-Nov-2023

	(Hasta 30 Puntos)	(Hasta 15 Puntos)	(0 Puntos)	
Funcionalidad del Programa	Se construye en forma adecuada la aplicación y los resultados son los esperados con base en el estado de los objetos.	Se construye la aplicación, pero los resultados son incompletos o no son los esperados	No es posible validar el ítem	30

JAIRO ARMANDO RIAÑO HERRERA
DOCENTE