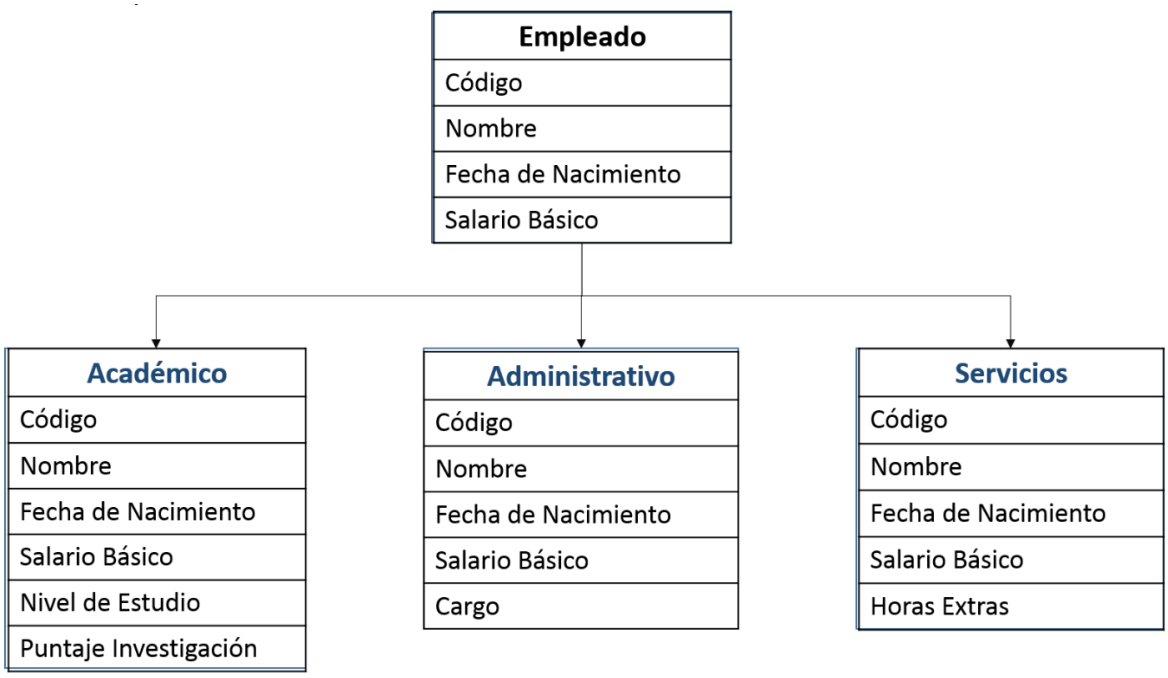
LA CLASE EMPLEADO

USANDO HERENCIA Y POLIMORFISMO

Lea cuidadosamente hasta entender el enunciado del taller, pueda que no encuentre lógica en algunos datos, es un ejercicio para conceptualizar el tema de herencia y polimorfismo. El taller pide el diseño de un programa en C++ que calcula una nómina sencilla teniendo en cuenta los siguientes datos.



Clase base empleado

Los atributos de la clase **empleado** son:

- **Código:** variable de tipo string,
- **Nombre:** variable de tipo string,
- **Fecha de Nacimiento:** es un arreglo int con 3 posiciones {día, mes, año},
- **Salario Básico:** variable de tipo double.

Los métodos de la clase **empleado** son:

1. **void PonCedula (string),** este método permite modificar el atributo Código del objeto asociado al método.

2. **void PonNombre (string),** este método permite modificar el atributo Nombre del objeto asociado al método.
3. **void PonFechaNacimiento (int[ ]),** este método permite modificar el atributo Fecha de Nacimiento del objeto asociado al método, recuerde que el parámetro de entrada se debe validar, día (1 – 31), mes (1 – 12) año (> 1900 && < Hoy) (Hoy debe ser una const definida que pueda ser cambiada en el código fuente en un solo lugar), existen algoritmos que validan fechas teniendo en cuenta los diferentes meses y los años bisiestos, haga uso de esos algoritmos.
4. **void SalarioBasico (double),** este método permite modificar el atributo Salario Básico del objeto asociado al método, recuerde que el parámetro de entrada se debe validar, pues el dato a recibir no puede ser menor que cero.
5. **virtual void imprimir() const,** Este método va a ser polimorfo, va a estar en todas las clases, en la clase EMPLEADO debe imprimir todos los atributos del objeto asociado al método.
6. **virtual double neto() const,** este método va a ser polimorfo, va a estar en todas las clases, las operaciones a realizar son diferentes para las tres clases. En la clase EMPLEADO este método retorna el Sueldo Básico.

Clase derivada académico

Académico hereda de **empleado**, sus atributos y sus métodos, además va a tener otros dos atributos que son:

- **Nivel de Estudio**: variable de tipo char, se debe validar que los niveles son,

Nivel de estudio	Cálculo de bonificación
T → Técnico o Tecnológico…	5 % del Salario Básico
P → Profesional……..	8 % del Salario Básico
E → Especialista……..	10 % del Salario Básico
M → Magíster……..	20 % del Salario Básico
D → Doctor……..	30 % del Salario Básico

- **Puntaje Investigación**: variable de tipo int, se debe validar que los puntajes son:

Categoría según puntos de investigación	Cálculo de bonificación
Desde **0** hasta **10** puntos → No categorizado investigador	# puntos * (0,01 * Salario Básico)
Desde **10** hasta **100** puntos → Investigador Junior	# puntos * (0,03 * Salario Básico)
Desde **101** hasta **500** puntos → Investigador Asociado	# puntos * (0,05 * Salario Básico)
Más de **501** puntos → Investigador Senior	# puntos * (0,08 * Salario Básico)

Bonificación en ambos casos se tiene en cuenta para el cálculo del salario neto a pagar:

Los métodos de la clase **académico** son:

1. **void PonNiveldeEstudio (char),** este método permite modificar el atributo Nivel de Estudio del objeto asociado al método, recuerde que el parámetro de entrada se debe validar.
2. **void PonPuntajeInvestigacion (char),** este método permite modificar el atributo Puntaje Investigación del objeto asociado al método, recuerde que el parámetro de entrada se debe validar.
3. **virtual void imprimir() const,** este método es polimorfo y permite imprimir objetos de la clase ACADÉMICO.
4. **virtual double neto() const,** este método retorna el sueldo neto a pagar, que se calcula teniendo en cuenta el Nivel de Estudio y el Tipo de Investigador así:

(Salario Básico + Bonificación Nivel de Estudio + Bonificación Puntaje en Investigación)* 0.89

Clase derivada administrativo

Administrativo hereda de **empleado**, sus atributos y sus métodos, además va a tener otro atributo que es:

- **Cargo:** variable de tipo char, se debe validar los cargos son:

Cargo	Bonificación
D → Directivo	30 % del Salario Básico
J → Jefe	20 % del Salario Básico
A → Auxiliar	10 % del Salario Básico

Los métodos de la clase **administrativo** son:

1. **void PonCargo (char),** este método permite modificar el atributo Cargo del objeto asociado al método, recuerde que el parámetro de entrada se debe validar.
2. **virtual void imprimir() const,** este método es polimorfo y permite imprimir objetos de la clase administrativo.
3. **virtual double neto() const,** este método retorna el sueldo neto a pagar, que se calcula teniendo en cuenta el Cargo así:

(Salario Básico + Bonificación por Cargo)* 0.89

Clase derivada servicios

Servicios hereda de **empleado**, sus atributos y sus métodos, va a tener otro atributo que es:

- **Horas Extras:** variable de tipo double por qué se va a operar con el Salario Básico, se debe validar que no sea negativo.

Los métodos de la clase **servicios** son:

1. **void PonHorasExtras (double),** este método permite modificar el atributo Horas Extras del objeto asociado al método, recuerde que el parámetro de entrada se debe validar, no puede ser negativo.
2. **double baseHorasExtras(),** este método retorna el valor base para calcular el monto a pagar por Horas Extras y se calcula así: **(Salario Básico / 30 / 8)*1.20**
3. **virtual void imprimir() const,** este método es polimorfo y permite imprimir objetos de la clase servicios.
4. **virtual double neto() const,** este método retorna el sueldo neto a pagar, que se calcula teniendo en cuenta las Horas Extras así:

(Salario Básico + (Horas Extras * baseHorasExtras())) * 0.89

NOTA: debe hacer un archivo main.cpp donde realice un programa que pruebe todas las clases anteriores, generando un programa int main(), que defina objetos **empleado**, **académico**, **administrativo** y **servicios** que hagan uso de los métodos de la clase **empleado** en todos los objetos, y también haga uso de los métodos de las clases derivadas en los objetos respectivos de cada clase.