

## Hoja de Trabajo N° 2: Un Estudiante

**Enunciado.** Analice el siguiente enunciado e identifique el mundo del problema, lo que se quiere que haga el programa y las restricciones para desarrollarlo.

Se desea construir una aplicación para el manejo de información de los cursos que está tomando un estudiante. El estudiante toma solo 4 cursos en el semestre. Los datos personales del estudiante que maneja la aplicación son código, nombre y apellido.

De cada curso se conoce:

- Código. Es el identificador del curso y no pueden haber dos cursos con el mismo código.
- Nombre.
- Departamento. Puede ser Matemáticas, Física, Sistemas o Biología.
- Cantidad de créditos.
- Nota obtenida en el curso. Este valor debe estar entre 1.5 y 5.

Para poder calcular el promedio del estudiante, se deben ponderar las notas, teniendo en cuenta la cantidad de créditos de las materias. Para esto, para cada curso se debe multiplicar la nota del curso con su cantidad de créditos, sumar estos valores y dividir esta suma por la cantidad total de créditos vistos por el estudiante. Por ejemplo, si el estudiante ha terminado dos materias, "Cálculo 1" y "Física 1", la primera de 4 créditos y la segunda de tres, con las siguientes notas:

- Cálculo 1: 4,5
- Física 1: 3,5

El promedio del estudiante es:

- $(4,5 * 4 + 3,5 * 3) / 7 = 4,07$

Adicionalmente, se quiere poder saber si un estudiante está en prueba académica o si es candidato para beca. Para esto se debe tener en cuenta las siguientes reglas:

- Se considera que un estudiante está en prueba académica si su promedio es inferior a 3.25.
- Se considera que un estudiante es candidato a beca si su promedio es igual o superior a 4.75.

La aplicación debe permitir: (1) visualizar la información del estudiante, (2) visualizar la información de los cursos, (3) modificar la información de un curso, (4) asignar una nota a un curso (5) calcular el promedio del estudiante (6) indicar si el estudiante está en prueba académica, (7) indicar si el estudiante es candidato a beca.

**Requerimientos funcionales. Especifique los siete requerimientos funcionales descritos en el enunciado.**

**Requerimiento Funcional 1**

Nombre:	R1 – Visualizar la información del estudiante.
Resumen:	El programa me debe permitir visualizar la información de cada estudiante, en el cual se debe indicar el código con el cual se identifica a cada estudiante, también se debe visualizar el nombre y apellido del estudiante.
Entradas:	Ninguna
Resultado:	Se indicará la información del estudiante

**Requerimiento Funcional 2**

Nombre:	R2 – Visualizar la información de los cursos.
Resumen:	<p>El programa debe permitir visualizar la información de cada una de las materias que se manejan, como lo son Matemáticas, Física, Sistemas y Biología, la información que se debe indicar para cada materia es.</p> <p>El código con la cual se identifica a cada materia y que debe de ser único</p> <p>La Cantidad de créditos de cada materia.</p> <p>Nota obtenida en el curso. Este valor debe estar entre 1.5 y 5.</p>
Entradas:	Ninguna
Resultado:	La información de las diferentes materias que se manejan

### Requerimiento Funcional 3

Nombre:	R3 – Modificar la información de un curso.
Resumen:	El programa me debe permitir modificar la información del curso.
Entradas:	Identificación del curso: Nombre del curso: Créditos del curso: Nota mínima y máxima del curso:
Resultado:	Se guarda la nueva información del curso.

### Requerimiento Funcional 4

Nombre:	R4 - Asignar una nota a un curso.
Resumen:	El programa me debe permitir asignar la nota a cada curso, la nota debe de ser entre 1.5 y 5.
Entradas:	<b>Asignar nota</b>
Resultado:	<b>Se guarda la información de la nota</b>

### Requerimiento Funcional 5

Nombre:	R5 - Calcular el promedio del estudiante.
Resumen:	<ul style="list-style-type: none"><li>El programa debe permitir calcular el promedio del estudiante, para calcular la nota tenemos la siguiente formula (Primera nota * Segunda nota + Tercera nota * 3 ) / 7 = Resultado</li></ul>
Entradas:	Ninguna
Resultado:	Promedio del estudiante

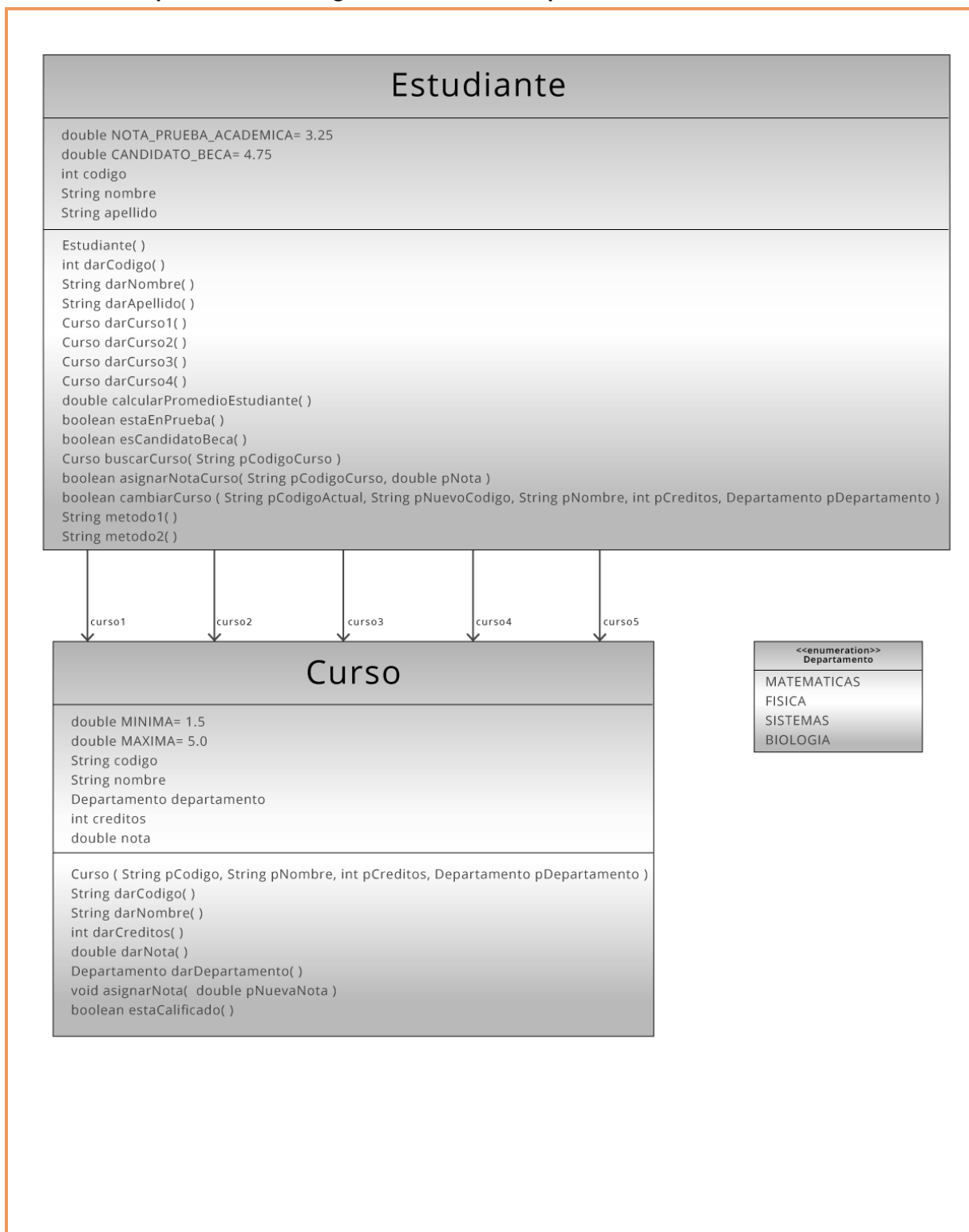
#### Requerimiento Funcional 6

Nombre:	R6 - Indicar si el estudiante está en prueba académica.
Resumen:	El programa me debe permitir saber si el estudiante está en prueba académica, para esto se debe considerar el promedio de la materia, si el promedio es inferior a 3.25, se considera que el estudiante está en prueba académica.
Entradas:	Ninguna
Resultado:	Indicar que el estudiante está en prueba académica

#### Requerimiento Funcional 7

Nombre:	R7 - Indicar si el estudiante es candidato a beca.
Resumen:	El programa me debe indicar si un estudiante es candidato a la beca, para esto debe considerar el promedio del estudiante, si el promedio del estudiante es mayor o igual a 4.75. se indica que el estudiante es un candidato a la beca.
Entradas:	Ninguna
Resultado:	Indicar si el estudiante es candidato a la beca.

**Modelo conceptual. Estudie el siguiente modelo conceptual.**



**Double = Real**

**String = Cadena de caracteres**

**Int = Entero**

**Declaración de las clases. Complete las declaraciones de las siguientes clases.**

**class Estudiante:**

```
# -----  
# Atributos  
# -----
```

```
#-----
```

# Atributos

```
#-----
```

codigo : int

nombre = ""

apellido = ""

curso = ""

#Constantes

NotaMaxima : int= 5,0

NotaMinima : int= 1,5

NotaPruebaAcademica : float= 3,35

NotaCandidtoBeca : float= 4,75

```
#-----
```

# Metodos

```
#-----
```

def \_\_init\_\_(self,codigo: int, nombre, apellido, curso):

self.codigo =codigo

self.nombre =nombre

self.apellido = apellido

self.curso = curso

**class Curso:**

```
# -----  
# Atributos  
# -----
```

```
#-----
```

```
# Atributos
```

```
#-----
```

```
nombre = ""
```

```
codigo:int
```

```
creditos :int
```

```
nota :float
```

```
#-----
```

```
#Constantes
```

```
#-----
```

```
minima: float =1,5
```

```
maxima: float=5,0
```

```
#-----
```

```
# Metodos
```

```
#-----
```

```
def __init__(self, nombre, codigo: int, credito: int, nota: float):
```

```
    self.nombre = nombre
```

```
    self.codigo = codigo
```

```
    self.credito = credito
```

```
    self.nota = nota
```



**Creación de expresiones. Para cada uno de los siguientes enunciados, escriba la expresión que lo representa. Tenga en cuenta la clase dada para determinar los elementos disponibles.**

<b>Curso</b>	¿El nombre del curso es "Cálculo 1"?	<code>curso = self.nombre</code>
<b>Curso</b>	¿El curso ya tiene una nota asignada?	<code>darNota = self.nota darNota.curso = nota()</code>
<b>Curso</b>	¿El curso tiene más de tres créditos?	<code>darCredito = self.curso darCredito = credito() if darCredito &gt;= 3:     True</code>
<b>Curso</b>	¿El curso fue aprobado?	<code>Curso.nota &gt;= 3,0</code>
<b>Estudiante</b>	¿El código del estudiante es "1234"?	<code>Estudiante.codigo=="1234"</code>
<b>Estudiante</b>	¿El primer curso tiene una nota asignada?	<code>estudiante= self.curso estudiante.curso[0].nota&gt;1.5</code>
<b>Estudiante</b>	¿El segundo curso pertenece al departamento de matemáticas?	<code>Estudiante.curso [1]. departamento == "matematicas"</code>

<b>Estudiante</b>	¿Cuál es el promedio del estudiante?	estudiante.calcular_promedio()

**Desarrollo de métodos. Escriba el código de los métodos indicados.**

<b>Clase: Curso</b>  Retorna el código del curso.	<pre>def darCodigo (self ):     return self.codigo</pre>
<b>Clase: Curso</b>  Indica si el curso ya fue calificado (tiene una nota distinta de cero).	<pre>def estaCalificado(self, nota ):  curso = self.__nota     if curso &gt; 0:         True ("El curso ya fue calificado")     False(curso = self.__nota(nota))</pre>
<b>Clase: Estudiante</b>  Retorna el nombre del estudiante.	<pre>def darNombre( ):     return self.nombre</pre>
<b>Clase: Estudiante</b>	<pre>def pertenecenMismoDepartamento( ):</pre>

Indica si el estudiante ya tiene los cuatro cursos pertenecen al mismo departamento.	<pre>departamentos =(curso.departamento for curso in self.curso) return len (departamento)==1</pre>

<p><b>Clase: Estudiante</b></p> <p>Calcula el promedio de los cursos que ya tienen nota. Si ningún curso tiene nota asignada, retorna cero.</p>	<pre>def calcularPromedioEstudiante( ):     nota1 = self.__nota     nota2 = self.__nota     nota3 = self.__nota     estudiante = self.__codigo     promedio= (nota1 * nota2 + nota3 * 3 ) / 7     = 0     estudiante. codigo = promedio     return promedio</pre>
---	---

--	--

<p><b>Clase: Estudiante</b></p> <p>Busca y retorna el curso que tiene el código que se recibe como parámetro. Si ningún curso tiene dicho código, el método retorna null.</p>	<pre>def buscarCurso( String pCodigoCurso):     For curso in self.curso:         If curso.codigo == pCodigoCurso:             return curso     return None</pre>
---	--

**Clase: Estudiante**

Indica si el estudiante se encuentra en prueba académica. Retorna verdadero si está en prueba académica, false de lo contrario.

```
def estaEnPrueba( ):  
    return self,culcularPromedioEstudiante())<3.25
```