

## Ficha Técnica

Nombre de la Clase:	<b>ColaPrioridad</b>
Descripción:	Clase que implementa el funcionamiento del TAD Cola de Prioridad, <b>utilizando varias Colas</b> , según el número de prioridades existentes. Serán atendidos, o tienen MAYOR prioridad, los Nodos que se encuentren en las <b>Colas</b> que manejen un valor de prioridad menor.
Módulo:	<b>bed/lineales/cp.py</b>

Método	Tipo de Retorno
<b><i>__init__</i></b> (self)	
<b><i>es_vacia</i></b> (self)	bool
<b><i>encolar</i></b> (self, nuevo_dato:object, prioridad:int) <i>Método que adiciona un nuevo Nodo con su dato a la Cola correspondiente, según la prioridad que éste tendrá.</i> <ul style="list-style-type: none"><li>• <i>prioridad</i> → [1 &gt; 2 &gt; 3 &gt; ... &gt; n]</li></ul>	bool
<b><i>desencolar</i></b> (self)	object/None
<b><i>frente</i></b> (self)	object/None
<b><i>__len__</i></b> (self)	int
<b><i>__str__</i></b> (self)* * Utilizar el mismo formato de una cola normal	str

## Ficha Técnica

Nombre de la Clase:	<b>ColaPrioridad</b>
Descripción:	<p>Clase que implementa el funcionamiento del TAD Cola de Prioridad, <b>utilizando Nodos del tipo NodoPrioridad</b>. Serán atendidos primeramente, o tienen MAYOR prioridad, los Nodos que por el contrario poseen un valor de prioridad menor.</p> <p><b>ATENCIÓN:</b> Esta clase debe soportar el manejo de los mismos métodos que implementa una clase <b>Cola</b>, con la excepción del método <b><i>encolar</i></b> que posee una implementación diferente.</p>
Módulo:	<b>bed/lineales/cp.py</b>

Método	Tipo de Retorno
<b><i>__init__</i></b> (self)	
<b><i>es_vacia</i></b> (self)	bool
<b><i>encolar</i></b> (self, nuevo_dato:object, prioridad:int) <i>Método que adiciona un nuevo <b>NodoPrioridad</b> con su dato a la <b>ColaPrioridad</b>, según la prioridad que éste tendrá. A mayor valor de la prioridad el nuevo <b>NodoPrioridad</b> se ubicará hacia el final de la <b>ColaPrioridad</b>.</i> <i>- Si el nuevo <b>NodoPrioridad</b> tiene la misma prioridad que uno o varios nodos de la <b>ColaPrioridad</b>, el nuevo <b>NodoPrioridad</b> se ubicará después del último <b>NodoPrioridad</b> con la misma prioridad.</i> <ul style="list-style-type: none"> <li><i>prioridad → [1 &gt; 2 &gt; 3 &gt; ... &gt; n]</i></li> </ul>	bool
<b><i>desencolar</i></b> (self)	object/None
<b><i>frente</i></b> (self)	object/None
<b><i>__len__</i></b> (self)	int
<b><i>__str__</i></b> (self)* * Utilizar el mismo formato de una cola normal	str

## Ficha Técnica

Nombre de la Clase:	<b>Prefija</b>
Descripción:	<p>Clase que implementa la <b><i>transformación</i></b> de un expresión matemática <b>Infija</b> a <b>Prefija</b> (<i>notación Polaca</i>) y el <b><i>cálculo</i></b> de la <b>expresión aritmética Prefija</b>.</p> <p>Los operandos utilizados serán de cualquier cantidad de dígitos.</p> <p>Operadores Válidos:</p> <ul style="list-style-type: none"> <li>+ : Suma</li> <li>- : Resta</li> <li>* : Multiplicación</li> <li>/ : División</li> <li>^ : Potenciación</li> <li>( : Paréntesis Izquierdo</li> <li>) : Paréntesis Derecho</li> </ul>
Módulo:	<b>bed/lineales/notaciones.py</b>

Método	Tipo de Retorno
<b><i>__init__</i></b> (self, expresión_infija:str)	
<b><i>infija</i></b> (self) Método que retorna la expresión Infija original, <b><i>separando</i></b> cada operando y cada operador, incluyendo los paréntesis, <b><i>por un espacio en blanco</i></b> .	str
<b><i>prefija</i></b> (self) Método que convierte una expresión Infija a una expresión Prefija, haciendo uso de una <b><i>Pila</i></b> . <b><i>Separar operandos y operadores por un espacio en blanco</i></b> .	str
<b><i>eval_expr_aritm</i></b> (self) Evaluación de la expresión aritmética en notación Prefija, utilizando una <b><i>Pila</i></b> , calculando el resultado final de la expresión.	float

## Ficha Técnica

Nombre de la Clase:	<b>Postfija</b>
Descripción:	<p>Clase que implementa la <b><i>transformación</i></b> de un expresión matemática <b>Infija</b> a <b>Postfija</b> (<i>Polaca Inversa</i>) y el <b><i>cálculo</i></b> de la <b>expresión aritmética Postfija</b>.</p> <p>Los operandos utilizados serán de cualquier cantidad de dígitos.</p> <p>Operadores Válidos:</p> <ul style="list-style-type: none"> <li>+ : Suma</li> <li>- : Resta</li> <li>* : Multiplicación</li> <li>/ : División</li> <li>^ : Potenciación</li> <li>( : Paréntesis Izquierdo</li> <li>) : Paréntesis Derecho</li> </ul>
Módulo:	<b>bed/lineales/notaciones.py</b>

Método	Tipo de Retorno
<b><i>__init__</i></b> (self, expresión_infija:str)	
<b><i>infija</i></b> (self) Método que retorna la expresión Infija original, <b><i>separando</i></b> cada operando y cada operador, incluyendo los paréntesis, <b><i>por un espacio en blanco</i></b> .	str
<b><i>postfija</i></b> (self) Método que convierte una expresión Infija a una expresión Postfija, haciendo uso de una <b><i>Pila</i></b> . <b><i>Separar operandos y operadores por un espacio en blanco</i></b> .	str
<b><i>eval_expr_aritm</i></b> (self) Evaluación de la expresión aritmética en notación Postfija, utilizando una <b><i>Pila</i></b> , calculando el resultado final de la expresión.	float