**Prueba 1 Tema 5**

**Aplicación de las estructuras de almacenamiento**

**Índice**

1. **C++………………………………………………………….1-3**

**1.1Listas enlazadas…………………………............1-2**

**1.2Pila…………………………………………………………2**

**1.3Cola………………………………………………………..3**

1. **C#……………………………………………………………3-5**

**2.1Listas enlazadas………………………………………3**

**2.2Pila…………………………………………………………4**

**2.3Cola………………………………………………………..5**

1. **Pre-conclusión……………………………………………..5**
2. **Conclusión……………………………………………………6**
3. **Bibliografía………………………………………………….6**

* **RESUELVE**

1. En la actualidad los principales lenguajes de programación de alto nivel son Java, C# y C++. En la unidad hemos estudiado cómo trabajar con listas en Java. Se pide indicar sintaxis y ejemplos de implementación de listas, pilas y colas en el lenguaje de programación C++ y en C#.

**A continuación se irán indicando las sintaxis e implementaciones que estarán en el mismo apartado, ya que se indican en las implementaciones que son estática y dinámicas, la sintaxis de cada uno.**

C++:

Listas enlazadas:

Dinámica:

**typedef Nodo \* Puntero;**

**struct Nodo**

**{**

**Valor Info;**

**Puntero Sig;**

**};**

**class Lista**

**{**

**public:**

**...**

**private:**

**Puntero ini; /\* Puntero al inicio de la lista \*/**

**Puntero fin; /\* Puntero al inicio de la lista \*/**

**Puntero pto; /\* Punto de interés \*/**

Estática**:**

**class Lista**

**{**

**public:**

**...**

**private:**

**Vector info;**

**int fin;**

**int pto;**

**}**

Pila**:**

Dinámica:

**class nodo {**

**private:**

**int valor;**

**nodo \*siguiente;**

**public:**

**nodo(int v, nodo \*sig)**

**{**

**valor = v;**

**siguiente = sig;**

**}**

**int getValor() {return (valor);}**

**nodo \*getSiguiente() {**

**return(siguiente); }**

Estática:

**class Pila {**

**private:**

**static const int MAX = 3;**

**int tope;**

**int valores[MAX];**

**public:**

**Pila() {**

**tope = -1;**

**}**

Cola**:**

Dinámica:

**public**

**Nodo\_cola = Registro**

**Info: Valor**

**Sig : Puntero\_a\_Nodo\_cola**

**private**

**Cola = Registro**

**Ini, Fin: Puntero\_a\_Nodo\_cola**

Estática:

**class Cola**

**{**

**public:**

**...**

**private:**

**Vector info;**

**int ini, fin;**

**}**

C#:

Listas enlazadas:

Dinámicas:

**using System;**

**using System.Collections.Generic;**

**using System.Linq;**

**using System.Text;**

**namespace ListaCircular1**

**{**

**public class ListaCircular**

**{**

**class Nodo**

**{**

**public int info;**

**public Nodo ant, sig;**

**}**

**private Nodo raiz;**

**public ListaCircular()**

**{**

**raiz = null;**

**}**

Estáticas:

**public class Nodo {**

**public Object dato; // dato contenido en el nodo**

**public Nodo siguiente; //puntero al siguiente nodo**

**}**

Pila:

Dinámicas:

**using System;**

**using System.Collections.Generic;**

**using System.Linq;**

**using System.Text;**

**namespace ListasTipoPila1**

**{**

**class Pila**

**{**

**class Nodo**

**{**

**public int info;**

**public Nodo sig;**

**}**

**private Nodo raiz;**

**public Pila()**

**{**

**raiz = null;**

**}**

Estáticas:

**#define STACKSIZE 100**

**typedef struct stack{**

**int top;**

**int nodes[STACKSIZE];**

**}Stack;**

**Stack \*create\_stack**

**(Stack \*S) {**

**S= (Stack \*)**

**malloc(sizeof(Stack));**

**S->top=-1;**

**return S;**

**}**

**main(){**

**Stack \*S;**

**S=create\_stack(S);**

**:**

**}**

Colas:

Dinámica:

**using System;**

**using System.Collections.Generic;**

**using System.Linq;**

**using System.Text;**

**namespace ListasTipoCola1**

**{**

**class Cola**

**{**

**class Nodo**

**{**

**public int info;**

**public Nodo sig;**

**}**

**private Nodo raiz,fondo;**

**public Cola()**

**{**

**raiz=null;**

**fondo=null;**

**}**

Estática:

**public class Cola**

**{**

**private int[] cola;**

**private int posicion;**

**public Cola(int tamanyo)**

**{**

**posicion = -1;**

**cola = new int[tamanyo];**

**}**

**Estas serían las sintaxis y tipos de implementaciones de cada tipo, listas enlazadas, pila y cola tanto en C++ como en C# como se podrá ver es algo parecido a java pero de formas diferentes, aun así los concepto de los punteros, nodos entre otros siguen siendo los mismos, en lo único que difiere es a la hora de implementarlo y claramente, aunque es algo que ya se sabe, en el lenguaje en sí.**

**Conclusión**

**Al ver las formas que hay y la sintaxis de cada uno he visto, que el más sencillo a mi gusto sería el que se está enseñando, java, ya que va más al grano y es a la vez más sencillo, también puede ser porque es el único lenguaje que he trabajado, pero a priori sería el mejor en estos aspecto. Otro asunto sería C# que ha sido el que más problemas ha dado a la hora de buscar las sintaxis de sus implementaciones, sobre todo por la parte estática ya que la que más salía era la dinámica, de la estática salía el concepto pero no las sintaxis, por lo cual ha sido el más complejo a la hora de buscar y elaborar las preguntas. De hecho es de los dos el único que tiene los apartados de estático y dinámico separado en la bibliografía por ese mismo motivo que ya se ha explicado con anterioridad, por lo cual si se tuviera que hacer un top 3 de los lenguajes entre java, C++ y C#, el orden sería:**

1. **Java**
2. **C++**
3. **C#**

**Aunque este top podría variar en un futuro, ya que se podría llegar a una mayor comprensión de los lenguajes aun no enseñados, aunque a priori como ya se ha dicho anteriormente, el mejor de los tres por lo menos en este aspecto y caso sería java por todo lo ya mencionado con anterioridad en este mismo apartado y también en otros ejercicios de la misma índole que también se dijeron las opiniones de ambos lenguajes, por lo cual de momento en el cómputo global sería de esta forma la clasificación de los lenguajes de programación.**

# Bibliografía

* **[Listas C++](http://informatica.uv.es/iiguia/AED/oldwww/2001_02/Teoria/Tema_13.pdf)**
* **[Pila C++](https://www.uv.mx/personal/edbenitez/files/2010/09/6-Pilas.pdf)**
* **[Colas C++](http://informatica.uv.es/iiguia/AED/oldwww/2002_03/Teoria/AED.Tema.12.pdf)**
* **[Listas dinámicas C#](https://www.tutorialesprogramacionya.com/csharpya/detalleconcepto.php?codigo=174&inicio=40)**
* **[Listas estáticas C#](https://analisisyprogramacionoop.blogspot.com/2017/07/lista-simplemente-enlazada-C-sharp.html)**
* [**Pila dinámica C#**](https://www.tutorialesprogramacionya.com/csharpya/detalleconcepto.php?codigo=167&inicio=40)
* [**Pila estática C#**](https://ccc.inaoep.mx/ingreso/programacion/corto2015/Curso-PROPE-PyED-5-Pilas-Colas.pdf)
* [**Cola dinámica C#**](https://www.tutorialesprogramacionya.com/csharpya/detalleconcepto.php?codigo=169&inicio=40)
* [**Cola estática C#**](https://www.exercisescsharp.com/es/colecciones-genericas/cola-con-arrays/)