

# C#,

# POO y

# Personajes Destacados.

Sebastián Robles.

Marcelo Bustos.

Cristian Vidal.

Chile, region de la Araucanía, Temúco , Julio 2018.

# Introducción.

Existen muchos lenguajes de programación en el mundo, los cuales han ido evolucionando con el paso del tiempo y avances del hardware, y con las multiples plataformas y usos de la programación se hace imperante conocer al menos la estructura básica de estos.

Software y hardware han avanzado de la mano impulsados por grandes personajes que sin duda cambiarón la historia de la humanidad.

El proposito de esta investigación es dar a conocer tanto un lenguaje de programación y su relación con la programación orientada a objetos como un personaje destacado en el mundo de la informaticá para ello hemos elegido como lenguaje C# y a Steve Wozniak.

C# se ha convertido en uno de los LDP más usados en la actualidad debido a su flexibilidad en multiplataforma y su fácil migración desde los lenguajes más utilizados para la enseñanza como son C++ y Java.

Stevie Wozniak, coofundador de appl y creador de computadoras como Apple I y Apple II, considerado uno de los padres de la revolución de las computadoras

# C#.

C# fué lanzado en julio del año 2000 por Microsoft como parte de su plataforma .NET, que hace un énfasis en la transparencía de [redes](https://es.wikipedia.org/wiki/Red_de_computadoras" \o "Red de computadoras), con independencia de plataforma de [hardware](https://es.wikipedia.org/wiki/Hardware" \o "Hardware) y que permite un rápido desarrollo de [aplicaciones](https://es.wikipedia.org/wiki/Aplicaci%C3%B3n_inform%C3%A1tica" \o "Aplicación informática), bajó el liderazgo de Anders Heljsberg, Scott Wiltamuth y Peter Golde. Basado en C++ y Java incorporando , principalmente, la capacidad de combinar operadores propia del primero (no incorpora la herencia múltiple) con la plena orientación a objetos del segundo. La orientación a objetos es tal que el propio programa está encapsulado en una clase. Actualmente C# se encuentra entre los 10 lenguajes más utilizados. Apesar de su corta historia, ha recibido la aprobación del estándar de dos organizaciones: en el 2001 se aprueba el ECMA y en el 2003 el ISO.

El estándar ECMA-334 lista las siguientes metas en el diseño para C#:

-Lenguaje de programación orientado a objetos simple, moderno y de propósito general.

-Inclusión de principios de [ingeniería de software](https://es.wikipedia.org/wiki/Ingenier%C3%ADa_de_software" \o "Ingeniería de software) tales como revisión estricta de los tipos de datos, detección de intentos de usar variables no inicializadas, y recolección de basura automática.

-Capacidad para desarrollar [componentes de software](https://es.wikipedia.org/wiki/Componente_de_software" \o "Componente de software) que se puedan usar en ambientes distribuidos.

-Portabilidad del código fuente.

-Fácil migración del programador al nuevo lenguaje.

-Soporte para [internacionalización](https://es.wikipedia.org/wiki/Internacionalizaci%C3%B3n_y_localizaci%C3%B3n" \o "Internacionalización y localización).

-Permitir crear tanto pequeños proyectos como proyectos a gran escala.

-Ahorro en cuanto a memoria y procesador.

Se utiliza principalmente en el desarrollo de aplicaciones para Windows Phone, desarrollo de paginas web tales como StackOverFlow, cuyo entorno es en base a C#, SQL Server y Windows IIS, también es utilizado en el desarrollo de juegos en tecnologías como Unity 3D, MonoGame y UnityFS.

Los principales IDEs disponibles son:

-Visual Studio: Al ser un IDE perteneciente a Microsoft, los creadores de C#, la integración del LDP es completa, es el más usado.

- Visual Studio Code : Utilizado generalmente para crear paginas webs basadas en C#.

-[#develop](http://www.icsharpcode.net/" \t "https://stackify.com/best-csharp-tools/_blank): es un version ligera que sirve como alternatica a Visual Studio Code.

- MonoDevelop :Developed by Xamarin, tiene muchas funciones similares a Visual Studio Code y tambien es multiplataforma . Con MonoDevelop, puedes crear rapidamente aplicaciones de escritorio como web para Windows, Linux y Mac.

-Otros IDEs menos utilizados son Eclipse, Code::Blocks, Project Rider y Rider.

Plataformas suportadas, C# fue creado con el proposito de tener gran portabilidad de codigo fuente por lo que soporta muchas plataformas tales como Microsoft Windows, Unix, Android, iOS, Windows Phone, Mac OS y GNU/Linux.

# C# y POO.

# - ¿Cómo se implementa el encapsulamiento de atributos y métodos?

Se antepone “public”, “protected” o “private” a la declaración del atributo o método, según el grado de encapsulamiento que se desee.

**-¿Cómo se crea una clase?**

|  |
| --- |
|  |
| encapsulamiento class nombreClase{  } |
|  |

**-¿Cómo se implementa el constructor de una clase?**

namespace NombreDelProyecto{

public class NombreDeLaClase{

public string NombreVariable1;

protected string NombreVariable2;

private int NombreVariable3;

public NombreDeLaClase( Variables de entrada para el constructor ) {

la implementación del Constructor va aquí

}

}

}

**-¿Cómo se implementa la herencia?**

namespace NombreDelProyecto{

public class NombreDeLaClaseHija : NombreDeLaClasePadre{

public string NombreVariable1;

protected string NombreVariable2;

private int NombreVariable3;

public NombreDeLaClaseHija( Variables de entrada para el constructor de la clase hija) : base(Variables de entrada de la clase Padre){

la implementación del Constructor va aquí

}

}

}

**En C# una clase solo puede tener una clase Padre, pero una clase Padre puede tener muchas clases Hijas y cada clase Hija puede a su vez ser la clase Padre de una n-ésima clase.**

**Las secciones rojas hacen referencia a la clase Padre y debiesen ir en la misma línea que las declaraciones de la clase hija. La primera sección roja indica que la clase Hija está heredando de la clase Padre. La segunda sección Roja especifica que la clase hija está heredando el constructor de la clase Padre. Todo lo escrito en la sección “implementación del constructor va aqui” se considera como una extensión del constructor de la clase Padre. La clase Hija no está obligada a heredar el constructor de la clase Padre y puede, en cambio, tener su propio constructor. Si la clase Hija hereda el constructor de la clase Padre, entonces, las variables de entrada de su constructor deben contener al menos las variables de entrada del constructor de la clase Padre.**

**- ¿Cómo se instancia un objeto?**

NombreClase nombreObjeto = new NombreClase( variables de entrada (solo si son necesarias) );

# Conclusiones.

C# es un lenguaje orientado a objetos, debido a su similitud con lenguajes como Java y C++, basados en la investigación y el programa desorrollado en C# podemos concluir que la sintaxis en lenguajes de programación que se orientan a objetos son muy similares variando solo en las palabras mas no en su estructura. C# se convirtío en uno de los lenguajes de programación mas utilizados en el mundo debido a sus ventajas: simpleza y multiproposito, ahorro en memoria y procesador y la portabilidad de su codigo fuente, una desventaja de C# es que a pesar de poder ser utilizado en Linux y Mac OS su funcionalidades son limitadas y de preferencía se sugiere utilizar otro lenguaje para esé tipo de proyecto.

C# nos pareció como se menciono anteriormente un lenguaje al que es muy facil migrar, sobretodo debido a que nuestra formación se basa en Java con el cual comparte muchas caracteristicas, es sencillo y esta desarrollado para Windows de los sistemas operativos mas utilizados.

Enlace al proyecto:

https://github.com/cvidalse/TrabajoLectura.

# Bibliografía.

1. ECMA, Standard ECMA-334 C# Language Specification, Scope, <http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-334.pdf>, 2017.
2. [RyKrD0](https://es.scribd.com/user/3262141/RyKrD0), Historia de C#, Lenguaje orientado a Objetos, <https://es.scribd.com/doc/9018328/Historia-C,> 2008.