

DEPARTAMENTO: Ciencias de la Ingeniería

CARRERA: Sistemas de Información

CURSO: Séptimo **PARALELO:** "A"

ASIGNATURA: Plataformas de Desarrollo 1

PROFESOR: Mg. Luis Fernando Aguas B.

ESTUDIANTE: Marco Antonio Ayala Lituma

DESCRIPCIÓN: Tarea Semana 1

TEMA: Investigación acerca de las aplicaciones de C#

DESARROLLO:

El lenguaje de Programación C# tiene varias connotaciones de acuerdo con su anteceder, su sintaxis de C# es muy expresiva, pero también sencilla y fácil de aprender. Cualquier persona familiarizada con C, C++, Java o JavaScript reconocerá al instante la sintaxis de llaves de C#. Los desarrolladores que conocen cualquiera de estos lenguajes podrán trabajar, normalmente, en C# de forma productiva en un espacio breve de tiempo.



C# proporciona características eficaces como tipos que aceptan valores NULL, delegados, expresiones lambda, coincidencia de patrones y acceso directo a memoria seguro. C# admite métodos y tipos genéricos, que proporcionan una mayor seguridad de tipos, así como un mejor rendimiento. C# también proporciona iteradores, gracias a los que los implementadores de clases de colecciones pueden definir comportamientos personalizados para el código de cliente. Las expresiones de Language Integrated Query (LINQ) convierten la consulta fuertemente tipada en una construcción de lenguaje de primera clase.

En cuanto lenguaje orientado a objetos, C# admite los conceptos de encapsulación, herencia y polimorfismo. Una clase puede heredar directamente de una clase primaria, pero puede implementar cualquier número de interfaces. Los métodos que invalidan los métodos virtuales en una clase primaria requieren la palabra clave override como una manera de evitar redefiniciones accidentales. En C#, un struct es como una clase ligera; es un tipo asignado en la pila que puede implementar interfaces pero que no admite la herencia. C# también proporciona registros, que son tipos de clase cuyo propósito es, principalmente, almacenar valores de datos. C# facilita el desarrollo de componentes de software mediante varias construcciones de lenguaje innovadoras, entre las que se incluyen las siguientes:

Signaturas de método encapsulado llamadas delegados, que permiten notificaciones de eventos con seguridad de tipos.



Propiedades, que actúan como descriptores de acceso para variables miembro privadas.
Atributos, que proporcionan metadatos declarativos sobre tipos en tiempo de ejecución.
Comentarios de documentación XML insertados Language Integrated Query (LINQ), que proporciona funcionalidades de consulta integradas en diversos orígenes de datos.
Coincidencia de patrones, que permite el flujo de control mediante la inspección de tipos de datos y valores.

Puede interactuar con los componentes nativos mediante un proceso denominado "interoperabilidad". La interoperabilidad permite que los programas de C# hagan casi todo lo que puede hacer una aplicación C++ nativa. C# admite incluso el uso de punteros y el concepto de código "no seguro" en los casos en los que el acceso directo a memoria es crítico.
El proceso de compilación de C# es simple en comparación con C y C++ y más flexible que en Java. No hay ningún archivo de encabezado independiente y ningún requisito de declaración de métodos y tipos en un orden en particular. Un archivo de código fuente de C# puede definir cualquier número de clases, structs, interfaces y eventos.
Actualmente su versión es C# 9.0

COMENTARIO:

El Lenguaje de Programación C# tiene un gran potencial ya que se innova constantemente ya que en la actualidad esta en la versión 9 y con la integración con .NET de la misma plataforma de Microsoft lo hace único ya que con aquello se puede crear aplicaciones multiplataformas con una sola base código, bajo el ambiente de Visual Studio se puede construir aplicaciones multipropósito y multi dispositivo.

BIBLIOGRAFÍA:

Libro Electrónico: Introducción al lenguaje C# y .NET (2020) Recuperado de:
<https://docs.microsoft.com/es-mx/dotnet/csharp/getting-started/>

Libro electrónico: guía de programación C# Recuperado de: <https://docs.microsoft.com/es-mx/dotnet/csharp/programming-guide/types/>

Libro Electrónico: Microsoft (2019) Recuperado de:
[Desarrollo de C# con Visual Studio - Visual Studio | Microsoft Docs](#)