



Practica 1

[Siguiete](#)

[Anterior](#)

Historia, evolución y características de Leng. de Programación

Objetivo: Conocer la evolución de los lenguajes de programación y sus características.

- [Ejercicio 1 Los lenguajes de programación más representativos son](#)
- [Ejercicio 2 Escriba brevemente la historia del lenguaje de programación](#)
- [Ejercicio 3 ¿Qué atributos debería tener un buen lenguaje de programación?](#)
- [Ejercicio 4 Tome uno o dos lenguajes de los que ud. Conozca y](#)
- [Ejercicio 5 Describa las características más relevantes de Ada, referida a](#)
- [Ejercicio 6 Diga para qué fue, básicamente, creado Java](#)
- [Ejercicio 7 ¿Sobre qué lenguajes está basado?](#)
- [Ejercicio 8 ¿Qué son los applets?](#)
- [Ejercicio 9 ¿Cómo es la estructura de un programa escrito en C?](#)
- [Ejercicio 10 Describa el manejo de expresiones que brinda el lenguaje](#)
- [Ejercicio 11 ¿Qué tipo de programas se pueden escribir con cada uno de estos lenguajes?](#)
- [Ejercicio 12 Cite otras características importantes de Python](#)
- [Ejercicio 13 ¿A qué tipo de paradigma corresponde este lenguajes?](#)
- [Ejercicio 14 Cite otras características importantes de javascript](#)

Ejercicio 1

Los lenguajes de programación más representativos son:

- **1951 - 1955:** Lenguajes tipo assembly
- **1956 - 1960:** FORTRAN, ALGOL 58, ALGOL 60, LISP
- **1961 - 1965:** COBOL, ALGOL 60, SNOBOL, JOVIAL
- **1966 - 1970:** APL, FORTRAN 66, BASIC, PL/I, SIMULA 67, ALGOL-W
- **1971 - 1975:** Pascal, C, Scheme, Prolog
- **1976 - 1980:** Smalltalk, Ada, FORTRAN 77, ML
- **1981 - 1985:** Smalltalk 80, Turbo Pascal, Postscript
- **1986 - 1990:** FORTRAN 90, C++, SML

- **1991 - 1995:** TCL, PERL, HTML
- **1996 - 2000:** Java, Javascript, XML

Indique para cada uno de los períodos presentados cuales son las características nuevas que se incorporan y cual de ellos la incorpora.

- **1951-1955: Lenguajes tipo assembly**

Características nuevas: Estos son los primeros lenguajes de programación que se crearon. Los lenguajes tipo assembly se basan en la arquitectura de la computadora y proporcionan una forma de escribir código de bajo nivel para controlar la máquina.

- **1956-1960: FORTRAN, ALGOL 58, ALGOL 60, LISP**

Características nuevas: Se introducen los primeros lenguajes de programación de alto nivel. Estos lenguajes proporcionan una forma más fácil y natural de expresar algoritmos que los lenguajes tipo assembly. FORTRAN y ALGOL 60 se convierten en los primeros lenguajes de programación ampliamente utilizados.

- **1961-1965: COBOL, ALGOL 60, SNOBOL, JOVIAL**

Características nuevas: Se desarrollan lenguajes de programación específicos para aplicaciones empresariales, como COBOL. Además, SNOBOL introduce la idea de patrones de búsqueda en los lenguajes de programación.

- **1966-1970: APL, FORTRAN 66, BASIC, PL/I, SIMULA 67, ALGOL-W**

Características nuevas: Se desarrollan lenguajes de programación para la programación orientada a objetos, como SIMULA 67. También se crean lenguajes de programación más fáciles de usar para los usuarios no expertos, como BASIC.

- **1971-1975: Pascal, C, Scheme, Prolog**

Características nuevas: Se introducen lenguajes de programación más eficientes y flexibles para la programación de sistemas, como C. Además, Pascal introduce conceptos de programación estructurada que hacen que el código sea más fácil de entender y mantener.

- **1976-1980: Smalltalk, Ada, FORTRAN 77, ML**

Características nuevas: Se desarrollan lenguajes de programación para la programación orientada a objetos, como Smalltalk y Ada. Además, FORTRAN 77 y ML se enfocan en mejorar la eficiencia y el rendimiento de los programas.

- **1981-1985: Smalltalk 80, Turbo Pascal, Postscript**

Características nuevas: Smalltalk 80 se convierte en uno de los primeros lenguajes de programación completamente orientado a objetos. Turbo Pascal hace que la programación estructurada sea más fácil para los usuarios no expertos. Postscript introduce la idea de describir la página completa en lugar de simplemente dibujarla.

- **1986-1990: FORTRAN 90, C++, SML**

Características nuevas: C++ se convierte en el primer lenguaje de programación ampliamente utilizado que combina programación orientada a objetos con programación estructurada. FORTRAN 90 introduce mejoras significativas en la eficiencia y el rendimiento de los programas.

- **1991-1995: TCL, PERL, HTML**

Características nuevas: Se introducen lenguajes de programación específicos para aplicaciones de Internet, como HTML. Además, TCL y PERL se convierten en lenguajes de programación populares para scripting y automatización de tareas.

- **1996-2000: Java, Javascript, XML**

Características nuevas: Java se convierte en un lenguaje de programación ampliamente utilizado para la

Ejercicio 2

Escriba brevemente la historia del lenguaje de programación que eligió en la encuesta u otro de su preferencia.

Python es un lenguaje de programación de alto nivel y propósito general que fue creado a finales de la década de 1980 por Guido van Rossum, un programador holandés. Van Rossum comenzó a trabajar en Python en diciembre de 1989, mientras trabajaba en el Centro para las Matemáticas y la Informática (CWI) en los Países Bajos.

El nombre "Python" fue elegido como un homenaje al grupo de comediantes británicos Monty Python, del cual Van Rossum es fanático. La primera versión pública de Python (versión 0.9.0) fue lanzada en febrero de 1991.

A lo largo de los años, Python ha evolucionado y se ha convertido en uno de los lenguajes de programación más populares del mundo. Python es conocido por su sintaxis clara y legible, lo que lo hace fácil de aprender y usar. Además, Python tiene una gran cantidad de bibliotecas y marcos de trabajo que permiten a los programadores crear aplicaciones de alta calidad de manera rápida y eficiente.

Hoy en día, Python se utiliza en una amplia variedad de aplicaciones, incluyendo el desarrollo web, la inteligencia artificial, el análisis de datos, la automatización de tareas y la robótica, entre otros. La comunidad de Python es muy activa y está formada por programadores de todo el mundo que contribuyen al desarrollo del lenguaje y de sus herramientas.

Ejercicio 3

¿Qué atributos debería tener un buen lenguaje de programación? Por ejemplo, ortogonalidad, expresividad, legibilidad, simplicidad, etc. De al menos un ejemplo de un lenguaje que cumple con las características citadas.

Un buen lenguaje de programación debería tener varios atributos, entre ellos:

- **Ortogonalidad**
esto significa que las construcciones básicas del lenguaje deben ser coherentes y consistentes. Por ejemplo, si un lenguaje permite la definición de funciones, entonces debería ser posible pasar una función como argumento a otra función.
- **Expresividad**
el lenguaje debe permitir al programador expresar claramente sus ideas de manera concisa y legible. Esto puede lograrse mediante la inclusión de constructores de lenguaje expresivos y la eliminación de elementos innecesarios.
- **Legibilidad**
el código escrito en el lenguaje debe ser fácil de leer y entender, incluso por personas que no estén familiarizadas con el lenguaje. Esto se puede lograr mediante el uso de una sintaxis clara y consistente, así como mediante el uso de nombres significativos para variables y funciones.

- **Simplicidad**

el lenguaje debe ser fácil de aprender y usar, sin requerir demasiado conocimiento previo de programación. Esto puede lograrse mediante la eliminación de características innecesarias y la inclusión de una sintaxis intuitiva.

- Un ejemplo de un lenguaje de programación que cumple con estos atributos es Python. Python tiene una sintaxis clara y legible que hace que sea fácil de aprender y usar. Además, su enfoque en la simplicidad y la ortogonalidad lo hace fácil de entender y leer. Python también es muy expresivo y permite a los programadores expresar sus ideas de manera clara y concisa, lo que lo convierte en un lenguaje popular para la inteligencia artificial y el análisis de datos.
-

Ejercicio 4

Tome uno o dos lenguajes de los que ud. Conozca y

- Describa los tipos de expresiones que se pueden escribir en el/ellos
- Describa las facilidades provistas para la organización del programa
- Indique cuáles de los atributos del ejercicio anterior posee el/los lenguaje/s elegidos y cuáles no posee, justifique en cada caso

Tipos de expresiones

En Python, las expresiones son combinaciones de valores, variables y operadores que se evalúan para producir un resultado. Hay varios tipos de expresiones en Python, como expresiones constantes, aritméticas, integrales, booleanas, de comparación, de asignación, de llamada de función, de indexación, de corte y de atributo

- Expresiones constantes: `2` , `3.14` , `"Hola"` , `True` , `False` .
- Expresiones aritméticas: `2 + 3` , `5 - 1` , `4 * 2` , `6 / 3` , `2 ** 3` .
- Expresiones integrales: `5 // 2` , `10 % 3` .
- Expresiones booleanas: `True and False` , `True or False` , `not True` .
- Expresiones de comparación: `2 > 1` , `2 < 1` , `2 == 2` , `2 != 3` .
- Expresiones de asignación: `x = 2` , `y = 3` .
- Expresiones de llamada de función: `print("Hola, mundo!")` .
- Expresiones de indexación: `lista[0]` , `cadena[1]` .
- Expresiones de corte: `lista[1:3]` , `cadena[2:5]` .
- Expresiones de atributo: `objeto.atributo` .

Facilidades de la organización

Python es un lenguaje de programación de alto nivel que tiene una sintaxis elegante y un sistema de programación orientado a objetos simple pero efectivo. El código de un software escrito en Python estará organizado en uno o varios ficheros con definiciones de Python. Cada uno de estos archivos es un módulo. Desde cada módulo podemos importar otros módulos. Para organizar los módulos, Python proporciona el concepto de “paquetes” y “subpaquetes”. Un paquete Python es un directorio del sistema de archivos en el que existe un fichero [init.py](#). Dentro del directorio puede haber uno o varios módulos de código Python, y también otros paquetes

Atributos que posee

Python es un lenguaje de programación que se caracteriza por su sintaxis simple y expresiva. Su objetivo principal es aumentar la legibilidad y la facilidad de desarrollo. En cuanto a los atributos mencionados, Python posee expresividad y legibilidad.

La simplicidad es una característica que puede entrar en conflicto con la expresividad

Lenguajes - ADA

Ejercicio 5

Describe las características más relevantes de Ada, referida a:

- Tipos de datos
- Tipos abstractos de datos – paquetes
- Estructuras de datos
- Manejo de excepciones
- Manejo de concurrencia

Ada es un lenguaje de programación orientado a objetos y multipropósito, desarrollado originalmente para aplicaciones militares y de defensa. En cuanto a sus características más relevantes en cuanto a tipos de datos, tipos abstractos de datos - paquetes, estructuras de datos, manejo de excepciones y manejo de concurrencia, podemos decir lo siguiente:

- **Tipos de datos:**

Ada es un lenguaje de programación fuertemente tipado, lo que significa que los tipos de datos son muy importantes. El lenguaje proporciona tipos de datos básicos, como enteros, caracteres, flotantes, etc. Además, también permite definir tipos de datos personalizados, incluyendo tipos de datos enumerados y tipos de subrangos.

- **Tipos abstractos de datos - paquetes:**

Ada es un lenguaje que permite la creación de tipos abstractos de datos, lo que significa que se pueden definir tipos de datos que encapsulan información y comportamiento, y que son independientes de su implementación. Los paquetes son una característica clave de Ada que permite definir tipos abstractos de datos, y también proporcionan una forma de organizar y agrupar conjuntos relacionados de subprogramas y tipos de datos.

- **Estructuras de datos:**

Ada proporciona una variedad de estructuras de datos, incluyendo matrices, registros y punteros. Además, también es posible definir tipos de datos personalizados, que pueden ser estructurados y anidados para crear estructuras de datos más complejas.

- **Manejo de excepciones:**

Ada proporciona una forma robusta y estructurada para manejar excepciones en el código. El lenguaje tiene un sistema de excepciones integrado que permite a los programadores definir excepciones personalizadas y establecer manejadores de excepciones para manejarlas.

- **Manejo de concurrencia:**

Ada es un lenguaje que permite la programación concurrente y distribuida, lo que significa que puede manejar la ejecución de varias tareas al mismo tiempo. Para lograr esto, Ada utiliza hilos, tareas, exclusión mutua y sincronización. Los hilos permiten la ejecución concurrente dentro del mismo proceso, mientras que las tareas permiten la comunicación entre unidades de ejecución. La exclusión mutua evita problemas de acceso concurrente a recursos compartidos, como variables y secciones críticas, y la sincronización coordina la ejecución de las tareas. En conjunto, estos mecanismos permiten la creación de programas robustos y eficientes que pueden ejecutar varias tareas de forma concurrente.

Lenguajes - JAVA

Ejercicio 6

Diga para qué fue, básicamente, creado Java. ¿Qué cambios le introdujo a la Web? ¿Java es un lenguaje dependiente de la plataforma en dónde se ejecuta? ¿Porqué?

Java fue creado por Sun Microsystems en 1991 como una herramienta de programación para ser usada en un proyecto de set-top-box. Java es un lenguaje independiente de la plataforma que permite que los programas escritos en el lenguaje Java puedan ejecutarse en cualquier tipo de hardware, lo que lo hace portable.

En cuanto a los cambios que le introdujo a la Web, Java permitió la creación de applets, que son pequeñas aplicaciones que se ejecutan dentro del navegador web y que pueden interactuar con el usuario.

Sí, Java es un lenguaje dependiente de la plataforma en donde se ejecuta. La máquina virtual de Java (JVM) es la encargada de la ejecución del código de las bibliotecas y las aplicaciones externas y su implementación depende de la plataforma en donde se ejecuta.

Ejercicio 7

¿Sobre qué lenguajes está basado?

Java es un lenguaje de programación de alto nivel, basado en clases y orientado a objetos. No está basado en ningún otro lenguaje de programación.

Ejercicio 8

¿Qué son los applets? ¿Qué son los servlets?

Los applets y los servlets son pequeños programas o aplicaciones de Java. La diferencia básica entre un applet y un servlet es que un applet se ejecuta en el lado del cliente, mientras que un servlet se ejecuta en el lado del servidor. Los servlets son programas para los servidores, mientras que los applets son programas para los clientes.

Lenguajes - C

Ejercicio 9

¿Cómo es la estructura de un programa escrito en C? ¿Existe anidamiento de funciones?

Un programa en C consta de una o más funciones, la función principal se llama main. Cada función debe contener cabecera, argumentos y sentencias. En cuanto al anidamiento de funciones, es posible anidar funciones en C

Ejercicio 10

Describe el manejo de expresiones que brinda el lenguaje

En C, las expresiones son combinaciones de operadores y operandos que se evalúan para producir un valor. Cuenta con una amplia variedad de operadores, incluyendo aritméticos, de asignación, de comparación, lógicos, de bits y ternarios. Los operadores permiten realizar operaciones matemáticas y lógicas complejas de manera fácil y flexible.

Lenguajes - Python - RUBY - PHP

Ejercicio 11

¿Qué tipo de programas se pueden escribir con cada uno de estos lenguajes? ¿A qué paradigma responde cada uno? ¿Qué características determinan la pertenencia a cada paradigma?

Los lenguajes de programación pueden clasificarse en varios paradigmas, según la forma en que se utilizan para resolver problemas. En esta respuesta, describiré los tipos de programas que se pueden escribir con los lenguajes de programación Python, Ruby y C, así como el paradigma al que responde cada uno y las características que determinan su pertenencia a ese paradigma.

Python:

- **Tipos de programas:** Python es un lenguaje de propósito general que se utiliza en una amplia variedad de aplicaciones, como desarrollo web, análisis de datos, inteligencia artificial, scripting, automatización de tareas y desarrollo de juegos.
- **Paradigma:** Python es un lenguaje multiparadigma que permite programar en diferentes estilos, como programación orientada a objetos, programación funcional y programación procedural.
- **Características del paradigma:**
 - **Programación orientada a objetos:** Python soporta la definición de clases, herencia, polimorfismo y encapsulamiento.
 - **Programación funcional:** Python soporta funciones como ciudadanos de primera clase, funciones anónimas (lambda), funciones de orden superior y programación en estilo funcional.

- **Programación procedural:** Python soporta programación estructurada con funciones y control de flujo.

Ruby:

- **Tipos de programas:** Ruby es un lenguaje de propósito general que se utiliza en una amplia variedad de aplicaciones, como desarrollo web, scripting, automatización de tareas, juegos y aplicaciones de escritorio.
- **Paradigma:** Ruby es un lenguaje orientado a objetos puro que promueve el uso de la programación en estilo declarativo.
- **Características del paradigma:**
 - **Programación orientada a objetos:** Ruby es un lenguaje orientado a objetos puro que permite la definición de clases, herencia, polimorfismo, encapsulamiento y mixins.
 - **Programación en estilo declarativo:** Ruby promueve el uso de la programación en estilo declarativo, lo que significa que los programas se escriben para especificar lo que se desea hacer en lugar de cómo hacerlo.

C:

- **Tipos de programas:** C es un lenguaje de bajo nivel que se utiliza para programar sistemas operativos, controladores de dispositivos, aplicaciones embebidas, compiladores, entre otros tipos de programas.
- **Paradigma:** C es un lenguaje estructurado que promueve la programación en estilo procedural.
- **Características del paradigma:**
 - **Programación estructurada:** C es un lenguaje estructurado que promueve la programación en estilo procedural, lo que significa que los programas se escriben en términos de procedimientos o funciones que realizan tareas específicas.
 - **Variables y tipos de datos:** C tiene un conjunto de tipos de datos básicos, como enteros, caracteres y flotantes, así como la capacidad de definir tipos de datos estructurados mediante estructuras y uniones.
 - **Control de flujo:** C tiene una variedad de operadores y estructuras de control de flujo, como condicionales, bucles y saltos, que permiten controlar la ejecución del programa.

En resumen, cada lenguaje de programación tiene sus propias características que lo hacen más adecuado para ciertos tipos de programas y paradigmas de programación. Python es un lenguaje multiparadigma que permite programar en diferentes estilos, Ruby es un lenguaje orientado a objetos puro que promueve la programación

Ejercicio 12

Cite otras características importantes de Python, Ruby, PHP, Gobstone y Processing.

Por ejemplo: tipado de datos, cómo se organizan los programas, etc.

Python:

- Python es un lenguaje de programación interpretado y de tipado dinámico.
- Se caracteriza por su sintaxis clara y concisa que favorece la legibilidad del código.
- Es un lenguaje multiparadigma, lo que significa que se puede utilizar para programación orientada a objetos, programación funcional y programación procedural.

- Los programas en Python se organizan en módulos y paquetes, que se pueden importar en otros programas para reutilizar el código.
- Python tiene una gran cantidad de bibliotecas estándar y módulos de terceros, lo que lo hace muy versátil para diferentes aplicaciones.

Ruby:

- Ruby es un lenguaje de programación interpretado y de tipado dinámico.
- Se caracteriza por su sintaxis expresiva y flexible, que permite escribir código de forma concisa y elegante.
- Es un lenguaje orientado a objetos puro, lo que significa que todo en Ruby es un objeto.
- Los programas en Ruby se organizan en clases y módulos, y se pueden definir métodos y atributos dentro de ellos.
- Ruby tiene una amplia variedad de bibliotecas y frameworks, lo que lo hace muy útil para el desarrollo web y otras aplicaciones.

PHP:

- PHP es un lenguaje de programación interpretado y de tipado débil.
- Se utiliza principalmente para el desarrollo web, y es muy popular en aplicaciones como WordPress, Drupal y Joomla.
- PHP tiene una sintaxis similar a la de C y otros lenguajes de programación, lo que lo hace fácil de aprender para quienes ya tienen experiencia en programación.
- Los programas en PHP se organizan en archivos que se ejecutan en el servidor web.
- PHP tiene una gran cantidad de bibliotecas y frameworks para el desarrollo web, lo que lo hace muy útil para la creación de sitios web dinámicos.

Gobstone:

- Gobstone es un lenguaje de programación especialmente diseñado para enseñar programación a niños y jóvenes.
- Es un lenguaje de tipado estático, lo que significa que se deben definir los tipos de datos de las variables.
- Los programas en Gobstone se organizan en procedimientos y funciones, y se utilizan bloques para estructurar el código.
- Gobstone tiene una sintaxis muy sencilla y fácil de entender, lo que lo hace ideal para enseñar los conceptos básicos de la programación.
- Gobstone se utiliza principalmente para programación de juegos y aplicaciones gráficas.

Processing:

- Processing es un lenguaje de programación diseñado para crear gráficos y animaciones.
- Es un lenguaje de programación basado en Java, y se utiliza principalmente para programación creativa.
- Los programas en Processing se organizan en funciones y se utilizan bibliotecas para crear gráficos y animaciones.
- Processing tiene una sintaxis sencilla y fácil de aprender, lo que lo hace ideal para programadores principiantes y artistas que deseen crear arte digital.

Lenguaje Javascript

Ejercicio 13

¿A qué tipo de paradigma corresponde este lenguajes? ¿A qué tipo de Lenguaje pertenece?

JavaScript es un lenguaje de programación interpretado y de múltiples paradigmas¹. Los paradigmas de programación que se pueden utilizar en JavaScript son el imperativo (también llamado procedimental) y el declarativo (también llamado funcional). JavaScript es un lenguaje orientado a objetos basado en prototipos.

Ejercicio 14

Cite otras características importantes de javascript. Tipado de datos, excepciones, variables, etc.

Tipado de datos en JavaScript:

JavaScript es un lenguaje de programación dinámicamente tipado, lo que significa que no se requiere declarar el tipo de datos de una variable antes de su uso. El tipo de datos se determina automáticamente en tiempo de ejecución. Sin embargo, hay formas de comprobar el tipo de datos de una variable en tiempo de ejecución, como el operador `typeof`.

Excepciones en JavaScript:

JavaScript tiene soporte para excepciones mediante el uso de la declaración `try-catch`. El código que puede lanzar una excepción se coloca dentro del bloque `try`, mientras que el manejo de la excepción se realiza dentro del bloque `catch`. Si una excepción es lanzada dentro del bloque `try`, el código se detendrá y el control se transferirá al bloque `catch`. El bloque `catch` puede contener código para manejar la excepción, como la impresión de un mensaje de error o la recuperación de datos.

Variables en JavaScript:

En JavaScript, las variables se pueden declarar utilizando la palabra clave `var`, `let` o `const`. La palabra clave `var` se utiliza para declarar una variable global o local, mientras que `let` y `const` se utilizan para declarar variables de ámbito de bloque.

La palabra clave `var` permite la redeclaración y la reasignación de variables, mientras que `let` y `const` solo permiten la reasignación. La palabra clave `const` se utiliza para declarar variables que no se pueden reasignar después de la inicialización.
