

**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.

•Período 1951 - 1955:

-Lenguajes tipo assembly: se caracterizan por ser lenguajes de bajo nivel, que se acercan mucho al lenguaje de la máquina, y que permiten escribir programas de manera más eficiente que en lenguaje máquina.

•Período 1956 - 1960:

-FORTRAN: el primer lenguaje de programación de alto nivel, diseñado específicamente para aplicaciones científicas y de ingeniería.

-ALGOL 58 y ALGOL 60: lenguajes de programación de alto nivel que influenciaron en el diseño de muchos otros lenguajes posteriores.

-LISP: primer lenguaje de programación funcional, que se enfoca en la evaluación de funciones y no en la ejecución de comandos secuenciales.

•Período 1961 - 1965:

-COBOL: primer lenguaje de programación diseñado para aplicaciones comerciales y empresariales.

-SNOBOL: primer lenguaje de programación orientado a procesamiento de texto.

-JOVIAL: lenguaje de programación diseñado para aplicaciones militares y aeroespaciales.

•Período 1966 - 1970:

-APL: lenguaje de programación orientado a procesamiento de datos matemáticos.

-FORTRAN 66: mejora del lenguaje FORTRAN original, con nuevas características para programación científica y de ingeniería.

-BASIC: lenguaje de programación diseñado para ser fácil de aprender y utilizado por personas sin experiencia previa en programación.

-PL/I: lenguaje de programación orientado a aplicaciones comerciales, científicas e industriales.

-SIMULA 67: primer lenguaje de programación orientado a objetos, que sentó las bases para el desarrollo posterior de lenguajes como C++ y Java.

-ALGOL-W: mejora de ALGOL 60, con nuevas características para programación de sistemas y sistemas operativos.

•Período 1971 - 1975:

-Pascal: lenguaje de programación diseñado para enseñar programación estructurada y de alta calidad.

-C: lenguaje de programación diseñado para programación de sistemas y aplicaciones de bajo nivel, que se convirtió en uno de los lenguajes más populares y ampliamente utilizados.

-Scheme: lenguaje de programación funcional y de programación estructurada.

-Prolog: lenguaje de programación orientado a la programación lógica, utilizado para inteligencia artificial y sistemas expertos.

•Período 1976 - 1980:

-Smalltalk: primer lenguaje de programación orientado a objetos puro, que influyó en el diseño de muchos otros lenguajes posteriores.

-Ada: lenguaje de programación diseñado para aplicaciones militares y críticas para la seguridad, con características para programación estructurada y orientada a objetos.

-FORTRAN 77: mejora de FORTRAN 66, con nuevas características para programación científica y de ingeniería.

-ML: lenguaje de programación funcional que se enfoca en la evaluación de expresiones matemáticas.

•Período 1981 - 1985:

-Smalltalk 80: mejora de Smalltalk, con nuevas características para programación orientada a objetos y entornos de programación visual.

-Turbo Pascal: lenguaje de programación diseñado para ser rápido y fácil de aprender, utilizado principalmente para programación de aplicaciones y juegos.

-Postscript: lenguaje de programación para describir gráficos vectoriales y páginas de documentos, utilizado para la impresión y publicación.

•Período 1986 - 1990:

-FORTRAN 90: mejora de FORTRAN, con nuevas características para programación científica y de ingeniería, como arreglos dinámicos y programación paralela.

-C++: mejora de C, con características para programación orientada a objetos y manejo de excepciones.

-SML: lenguaje de programación funcional que se enfoca en la evaluación de expresiones matemáticas complejas.

•Período 1991 - 1995:

-TCL: lenguaje de programación orientado a scripting y automatización de tareas, utilizado en sistemas operativos y aplicaciones de software.

-PERL: lenguaje de programación orientado a texto y procesamiento de datos, utilizado para scripting y desarrollo de aplicaciones web.

-HTML: lenguaje de marcado utilizado para la creación de páginas web y sitios web estáticos.

•Período 1996 - 2000:

-Java: lenguaje de programación orientado a objetos y plataforma de software diseñado para ser portátil y seguro, utilizado en aplicaciones web y móviles.

-Javascript: lenguaje de programación orientado a la web y al lado del cliente, utilizado para la programación de interacciones y dinamismo en páginas web.

-XML: lenguaje de marcado utilizado para el intercambio de datos entre aplicaciones y sistemas, utilizado en aplicaciones web y empresariales.

---

**Ejercicio 2:** Escriba brevemente la historia del lenguaje de programación que eligió en la encuesta u otro de su preferencia.

### Historia de **Java** y **JavaScript**

Java y JavaScript son dos lenguajes de programación diferentes con historias distintas.

Java fue desarrollado por Sun Microsystems en los años 90 como una plataforma independiente de hardware y software, lo que significa que los programas escritos en Java pueden ejecutarse en cualquier dispositivo o sistema operativo que tenga una máquina virtual de Java instalada. Java se convirtió rápidamente en uno de los lenguajes de programación más populares para el desarrollo de aplicaciones empresariales y web.

Por otro lado, JavaScript fue creado por Brendan Eich de Netscape en 1995 como un lenguaje de scripting para aplicaciones web. Inicialmente, su objetivo era permitir la interactividad en las páginas web del navegador, pero rápidamente se convirtió en un lenguaje de programación muy popular para el desarrollo de aplicaciones web complejas.

A pesar de que los nombres de Java y JavaScript son similares, los dos lenguajes no están relacionados entre sí. Java es un lenguaje de programación de propósito general, mientras que JavaScript es un lenguaje de scripting para aplicaciones web. Ambos lenguajes son ampliamente utilizados en la actualidad y han evolucionado mucho desde sus primeras versiones.

---

**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 deseables, como:

- Ortogonalidad: como se mencionó anteriormente, las características de un lenguaje de programación deben ser independientes entre sí y combinarse de forma coherente.

- **Expresividad:** el lenguaje debe permitir que el programador exprese sus ideas de forma clara y concisa.
- **Legibilidad:** el código debe ser fácil de leer y entender, tanto para el programador que lo escribió como para otros programadores.
- **Simplicidad:** el lenguaje debe ser fácil de aprender y utilizar, y no debe tener características innecesariamente complicadas.
- **Eficiencia:** el lenguaje debe permitir la creación de programas eficientes, tanto en términos de velocidad de ejecución como de uso de memoria.

Un ejemplo de un lenguaje de programación que cumple con estos atributos es Python. Python es un lenguaje de programación interpretado y de alto nivel que es conocido por su simplicidad y legibilidad. Python es muy expresivo, lo que significa que los programas escritos en Python suelen ser más cortos y concisos que los escritos en otros lenguajes. Python también es altamente ortogonal, lo que significa que las características del lenguaje pueden combinarse de forma coherente. Además, Python es un lenguaje eficiente, ya que está diseñado para ejecutarse rápidamente y con un uso mínimo de memoria.

---

**Ejercicio 4:** Tome uno o dos lenguajes de los que ud. Conozca y

- Describa los tipos de expresiones que se pueden escribir en él/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.

## **C#**

- **Tipos de expresiones:**

- Expresiones aritméticas: permiten realizar operaciones matemáticas, como sumar, restar, multiplicar y dividir números.
- Expresiones lógicas: permiten realizar operaciones lógicas, como comparar valores o realizar operaciones booleanas (AND, OR, NOT).
- Expresiones de asignación: permiten asignar valores a variables.
- Expresiones condicionales: permiten tomar decisiones en función del resultado de una evaluación lógica.
- Expresiones de bucles: permiten repetir una sección de código mientras se cumpla una condición.

- **Facilidades de organización del programa:**

- Espacios de nombres: permiten agrupar clases y otros elementos relacionados en un espacio de nombres común.
- Clases y estructuras: permiten definir tipos de datos personalizados.

-Interfaces: permiten definir un conjunto de métodos que deben ser implementados por cualquier clase que las implemente.

-Métodos y funciones: permiten definir bloques de código reutilizables que realizan una tarea específica.

-Propiedades y campos: permiten definir variables para almacenar datos en una clase.

- Atributos del lenguaje:

-Ortogonalidad: C# es un lenguaje altamente ortogonal, lo que significa que sus características pueden combinarse de forma coherente e independiente.

-Expresividad: C# es un lenguaje altamente expresivo, lo que permite a los programadores expresar sus ideas de forma clara y concisa.

-Legibilidad: C# es un lenguaje legible gracias a su sintaxis clara y estructurada.

-Simplicidad: C# es un lenguaje relativamente simple y fácil de aprender, aunque puede ser más complejo en su uso avanzado.

-Eficiencia: C# es un lenguaje eficiente gracias a su compilación just-in-time (JIT) y su capacidad para trabajar con código nativo.

En resumen, C# permite escribir expresiones aritméticas, lógicas, de asignación, condicionales y de bucles. Además, ofrece facilidades para la organización del programa como espacios de nombres, clases, interfaces, métodos y propiedades. C# es un lenguaje altamente ortogonal y expresivo, legible, relativamente simple y eficiente gracias a su compilación JIT. Sin embargo, su simplicidad puede ser cuestionable en algunas situaciones avanzadas donde la complejidad del lenguaje puede aumentar.

---

### **Ejercicio 5:** Describa 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 diseñado para aplicaciones de alta confiabilidad y seguridad, y tiene varias características relevantes en cuanto a tipos de datos, tipos abstractos de datos, estructuras de datos, manejo de excepciones y manejo de concurrencia.

- Tipos de datos:

Ada ofrece un conjunto muy completo de tipos de datos, incluyendo tipos numéricos, tipos de caracteres, tipos de cadenas, tipos booleanos y tipos de punto flotante. Además, Ada permite definir tipos de datos personalizados y tipos enumerados, lo que facilita la creación de programas más legibles y mantenibles.

- Tipos abstractos de datos - paquetes:

Ada permite definir tipos abstractos de datos y paquetes, lo que permite al programador encapsular la implementación de los tipos y proporcionar una interfaz clara y bien definida para su uso en otros lugares del programa. Esto mejora la modularidad y la reutilización del código.

- Estructuras de datos:

Ada ofrece una variedad de estructuras de datos, incluyendo matrices, listas enlazadas, colas y pilas, así como estructuras de datos personalizadas definidas por el usuario. También proporciona acceso directo a los bits individuales de una variable, lo que es útil para programar dispositivos de bajo nivel.

- Manejo de excepciones:

Ada tiene un sistema de manejo de excepciones muy robusto que permite al programador definir excepciones personalizadas y manejarlas de manera específica. También proporciona un mecanismo de propagación de excepciones que facilita la identificación y resolución de errores.

- Manejo de concurrencia:

Ada ofrece un conjunto de características para el manejo de concurrencia, como tareas y entradas-salidas bloqueantes. Las tareas son similares a los hilos de ejecución y permiten al programador escribir programas concurrentes de manera fácil y segura. Además, Ada ofrece un sistema de comunicación interproceso basado en entradas-salidas bloqueantes que simplifica el manejo de la concurrencia en programas de alta confiabilidad.

En resumen, Ada es un lenguaje de programación de alto nivel con una amplia gama de características para tipos de datos, tipos abstractos de datos, estructuras de datos, manejo de excepciones y manejo de concurrencia. Su enfoque en la seguridad y la fiabilidad lo hace popular en aplicaciones críticas, como sistemas de defensa, aeronaves y dispositivos médicos.

---

**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? ¿Por qué?

Java fue creado principalmente como un lenguaje de programación para dispositivos electrónicos, en particular para los electrodomésticos inteligentes (smart appliances) de la época. Sin embargo, a medida que el lenguaje se fue desarrollando, se convirtió en una herramienta importante para la creación de aplicaciones empresariales y de servidores, y más tarde en una plataforma para la creación de aplicaciones web.

Java introdujo importantes cambios a la web, en particular a través de la introducción de la tecnología Java applet. Los applets de Java permitieron a los desarrolladores crear aplicaciones web interactivas y dinámicas que podían ser ejecutadas en cualquier sistema operativo y en cualquier navegador web, lo que eliminó la necesidad de descargar y ejecutar aplicaciones nativas en el ordenador local. Además, Java también introdujo

importantes mejoras en cuanto a la seguridad de las aplicaciones web, ya que los applets de Java se ejecutan en un entorno de sandbox que limita su capacidad para acceder a los recursos del sistema.

Java es un lenguaje de programación dependiente de la plataforma en donde se ejecuta, debido a que el código Java se compila en bytecode, que es un código de bajo nivel que no puede ser ejecutado directamente por el sistema operativo. En su lugar, el bytecode Java debe ser interpretado y ejecutado por una máquina virtual Java (JVM), que se encarga de traducir el bytecode a código nativo para la plataforma en la que se está ejecutando. Esto permite que las aplicaciones Java sean altamente portátiles, ya que pueden ser escritas en cualquier sistema operativo y ejecutadas en cualquier otro sistema operativo que tenga una JVM instalada.

---

### **Ejercicio 7:** ¿Sobre qué lenguajes está basado?

Java no está basado directamente en ningún lenguaje en particular. Sin embargo, el diseño de Java se inspiró en gran medida en varios lenguajes de programación, incluyendo C++, Smalltalk y Objective-C.

En particular, Java heredó muchos conceptos y características de C++, como la sintaxis de los lenguajes, la gestión de la memoria a través del recolector de basura, la capacidad de definir tipos de datos y funciones, y la posibilidad de crear objetos y usar herencia para crear nuevas clases a partir de clases existentes.

Por otro lado, Java también se inspiró en el lenguaje Smalltalk, que fue uno de los primeros lenguajes de programación orientado a objetos. De Smalltalk, Java tomó la idea de que todo en el lenguaje es un objeto y la capacidad de crear objetos y enviar mensajes entre ellos.

Finalmente, Java también se inspiró en Objective-C, otro lenguaje de programación orientado a objetos que se utiliza ampliamente en el desarrollo de aplicaciones para iOS y macOS. De Objective-C, Java tomó la idea de definir clases y objetos, así como la capacidad de enviar mensajes entre objetos en tiempo de ejecución.

---

### **Ejercicio 8:** ¿Qué son los applets? ¿Qué son los servlets?

Tanto los applets como los servlets son tecnologías de Java que se utilizan en el desarrollo de aplicaciones web. Sin embargo, se utilizan en diferentes contextos y para diferentes propósitos.

Los applets son pequeñas aplicaciones de software escritas en Java que se ejecutan en el navegador web del usuario. Los applets se descargan automáticamente desde un servidor web y se ejecutan en el navegador del usuario utilizando la máquina virtual Java (JVM) que se encuentra en el navegador. Los applets se pueden utilizar para crear gráficos interactivos, juegos y otras aplicaciones que se ejecutan en la página web del usuario.



Los servlets, por otro lado, son componentes de software que se ejecutan en el servidor web. Los servlets son programas Java que se ejecutan en el servidor y generan contenido dinámico en respuesta a las solicitudes del cliente. Los servlets se utilizan a menudo para procesar formularios web, generar páginas web dinámicas y proporcionar servicios web RESTful.

En resumen, los applets son pequeñas aplicaciones que se ejecutan en el navegador del usuario, mientras que los servlets son componentes de software que se ejecutan en el servidor web. Ambos se utilizan en el desarrollo de aplicaciones web y aprovechan las características de Java, como la portabilidad y la seguridad.

---

**Ejercicio 9:** ¿Cómo es la estructura de un programa escrito en C? ¿Existe anidamiento de funciones?

La estructura de un programa escrito en C consta de varias partes, que incluyen:

- **Directivas de preprocesador:** estas líneas comienzan con el carácter "#" y se utilizan para incluir archivos de encabezado, definir constantes y realizar otras operaciones de preprocesamiento antes de compilar el código.
- **Funciones:** el cuerpo principal del programa se compone de una o varias funciones. Cada función tiene un nombre, un conjunto de parámetros y un cuerpo que define las operaciones que realiza la función.
- **Variables:** las variables se utilizan para almacenar valores en el programa. Antes de utilizar una variable, es necesario declararla para especificar su tipo y nombre.
- **Estructuras de control de flujo:** las estructuras de control de flujo se utilizan para controlar el flujo de ejecución en el programa. Estas estructuras incluyen declaraciones de control de bucles, condicionales y saltos.

Sí, en C es posible anidar funciones, es decir, definir una función dentro de otra función. Esto se utiliza a menudo para definir funciones de utilidad que sólo se utilizan dentro de una función principal y que no necesitan ser visibles fuera de esa función.

La estructura de un programa en C puede variar dependiendo del objetivo y la complejidad del mismo, pero en general sigue esta estructura básica.

---

**Ejercicio 10:** Describa el manejo de expresiones que brinda el lenguaje.

El lenguaje C es un lenguaje de programación de bajo nivel que ofrece un amplio conjunto de operadores y constructos para manejar expresiones.

- En C, las expresiones son combinaciones de operadores y operandos que se evalúan para producir un resultado. Los operandos pueden ser variables, valores constantes o llamadas a funciones. Los operadores, por otro lado, pueden ser aritméticos, lógicos, de comparación, de asignación, entre otros.

- Los operadores aritméticos incluyen la suma (+), la resta (-), la multiplicación (\*), la división (/) y el módulo (%). Estos operadores se utilizan para realizar operaciones matemáticas básicas.

- Los operadores lógicos incluyen AND lógico (&&), OR lógico (||) y NOT lógico (!). Se utilizan para realizar comparaciones booleanas y evaluar expresiones condicionales.

- Los operadores de comparación incluyen igualdad (==), desigualdad (!=), mayor que (>), mayor o igual que (>=), menor que (<) y menor o igual que (<=). Estos operadores se utilizan para comparar valores y evaluar expresiones condicionales.

- El lenguaje C también proporciona operadores de incremento (++) y decremento (--), que se utilizan para aumentar o disminuir el valor de una variable en una unidad.

- Además de los operadores, el lenguaje C ofrece constructos de control de flujo como if-else, while, for, switch y do-while, que permiten la ejecución condicional y repetitiva de bloques de código.

En general, el manejo de expresiones en el lenguaje C es muy poderoso y flexible, lo que permite a los programadores escribir código eficiente y sofisticado. Sin embargo, también requiere un buen entendimiento de la sintaxis y semántica del lenguaje para evitar errores y garantizar la correcta ejecución de los programas.

---

## **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?

Python, Ruby y PHP son tres lenguajes de programación populares que tienen diferentes características y se utilizan para diferentes propósitos.

- Python:**

Python es un lenguaje de programación interpretado de alto nivel que se utiliza para una amplia gama de aplicaciones, incluyendo aplicaciones web, ciencia de datos, inteligencia artificial y automatización de tareas. Python es un lenguaje multiparadigma, es decir, puede utilizarse en varios paradigmas de programación, incluyendo programación orientada a objetos, programación imperativa y programación funcional. Python se destaca por su sintaxis clara y legible, su facilidad de uso y su gran cantidad de bibliotecas y marcos de trabajo disponibles.

- Ruby:**

Ruby es un lenguaje de programación interpretado de alto nivel que se utiliza principalmente para el desarrollo web, la automatización de tareas y la programación de scripts. Ruby se enfoca en la programación orientada a objetos y se ha diseñado para ser fácil de leer y escribir. Ruby tiene una sintaxis flexible y dinámica, lo que la hace muy adecuada para la creación de aplicaciones web y la automatización de tareas.

- PHP:**

PHP es un lenguaje de programación de código abierto que se utiliza principalmente para el desarrollo web y la creación de aplicaciones dinámicas. PHP se enfoca en la programación de scripts y en la interacción con bases de datos y servidores web. PHP es un lenguaje de programación interpretado, que se ejecuta en el lado del servidor. PHP es un lenguaje multiparadigma, pero se utiliza principalmente en la programación imperativa y en la programación orientada a objetos.

---

**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:**

- Tipado dinámico:** Python es un lenguaje de tipado dinámico, lo que significa que los tipos de datos de las variables no se declaran explícitamente, sino que se infieren durante la ejecución del programa.
- Sintaxis clara y legible:** Python se enfoca en la legibilidad del código y en una sintaxis clara, lo que lo hace fácil de leer y escribir.
- Gran cantidad de bibliotecas y módulos:** Python tiene una amplia gama de bibliotecas y módulos disponibles, lo que permite a los programadores reutilizar código y desarrollar aplicaciones de manera eficiente.

**Ruby:**

- Tipado dinámico:** Ruby es un lenguaje de tipado dinámico, lo que significa que los tipos de datos de las variables no se declaran explícitamente, sino que se infieren durante la ejecución del programa.
- Programación orientada a objetos:** Ruby se enfoca en la programación orientada a objetos y tiene una sintaxis clara y concisa para la definición de clases y objetos.
- Flexibilidad:** Ruby es un lenguaje muy flexible que permite a los programadores redefinir funciones y clases existentes, lo que facilita la creación de código reutilizable y dinámico.

**PHP:**

- Tipado débil:** PHP es un lenguaje de tipado débil, lo que significa que los tipos de datos de las variables pueden cambiar durante la ejecución del programa.
- Interacción con bases de datos y servidores web:** PHP se enfoca en la interacción con bases de datos y servidores web, y se utiliza principalmente para la creación de aplicaciones web y dinámicas.
- Sintaxis similar a C:** PHP tiene una sintaxis similar a la de C, lo que facilita la transición de los programadores que tienen experiencia en C.

**Gobstone:**

- Enfocado en la enseñanza: Gobstone es un lenguaje de programación diseñado específicamente para la enseñanza de programación a niños y jóvenes.
- Interactivo: Gobstone se enfoca en la interacción del programa con el usuario y tiene una sintaxis clara y sencilla.
- Gráficos: Gobstone incluye una biblioteca de gráficos que permite a los programadores crear visualizaciones interactivas.

**Processing:**

- Gráficos y visualización: Processing se enfoca en la creación de gráficos y visualizaciones interactivas y es utilizado principalmente por artistas y diseñadores.
  - Fácil de aprender: Processing tiene una sintaxis sencilla y fácil de aprender, lo que lo hace ideal para personas sin experiencia en programación.
  - Multiplataforma: Processing se puede utilizar en diferentes plataformas y sistemas operativos, lo que lo hace accesible para una amplia audiencia.
- 

**Ejercicio 13:** ¿A qué tipo de paradigma corresponde JavaScript? ¿A qué tipo de Lenguaje pertenece?

JavaScript es un lenguaje de programación que se adapta a varios paradigmas de programación, pero se considera principalmente un lenguaje de programación orientado a objetos y funcional.

En términos generales, JavaScript se puede clasificar como un lenguaje de programación de alto nivel, interpretado y dinámicamente tipado. Se utiliza principalmente en el desarrollo web para crear contenido interactivo y dinámico en el lado del cliente, aunque también puede ser utilizado en el lado del servidor a través de frameworks como Node.js.

JavaScript es uno de los lenguajes más populares y ampliamente utilizados en la actualidad, y es compatible con la mayoría de los navegadores web modernos.

---

**Ejercicio 14:** Cite otras características importantes de javascript. Tipado de datos, excepciones, variables, etc.

Aquí hay algunas características importantes de JavaScript:

- Tipado dinámico: JavaScript es un lenguaje de tipado dinámico, lo que significa que los tipos de datos de las variables no se declaran explícitamente, sino que se infieren durante la ejecución del programa.
- Funciones de primera clase: Las funciones en JavaScript son tratadas como valores y se pueden pasar como argumentos a otras funciones, asignar a variables y retornar de otras funciones.
- Manejo de excepciones: JavaScript tiene un sistema de manejo de excepciones para capturar y manejar errores en tiempo de ejecución.

- Variables: En JavaScript, las variables se declaran usando la palabra clave "var", "let" o "const". Las variables declaradas con "var" y "let" pueden cambiar su valor, mientras que las variables declaradas con "const" son constantes y no pueden cambiar su valor.
- Prototipado: En JavaScript, los objetos se basan en prototipos en lugar de clases como en otros lenguajes orientados a objetos. Esto permite la creación de objetos con propiedades y métodos compartidos.
- DOM y AJAX: JavaScript se utiliza ampliamente para manipular el Document Object Model (DOM) en páginas web y para enviar solicitudes asíncronas a servidores web a través de la tecnología AJAX (Asynchronous JavaScript and XML).