Práctica 1

Historia, evolución y características de lenguajes de programación

1. Los lenguajes de programación más representativos, indique para cada uno de los periodos presentados cuáles son las características nuevas que incorporan y cuál de ellos la incorpora.

   1951- 1955: lenguajes tipo assembly

   Permitían escribir código de bajo nivel usando una sintáxis legible para los programadores. Algunas características incorporadas fueron

   - Permitían a los programadores escribir código de bajo nivel de forma accesible y legible.
   - Podían ser portados entre diferentes plataformas de hardware.

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

   Se desarrollaron varios lenguajes de alto nivel, que permitían una programación más abstracta.

   - FORTRAN: uno de los primeros lenguajes de alto nivel que permitió la programación de cálculos matemáticos complejos. Se incorporó el uso de variables y subrutinas.
   - ALGOL 58: uno de los primeros lenguajes diseñados para que sea independiente de la plataforma de hardware. También permitía la definición de estructuras de control complejas.
   - ALGOL 60: versión mejorada de ALGOL 58, incorporó la gestión dinámica de memoria y la definición de funciones recursivas.
   - LISP: uno de los primeros lenguajes diseñado para el procesamiento de listas y datos simbólicos. Incorporó la evaluación de expresiones de manera recursiva.

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

   Las nuevas novedades fueron orientación a negocios, flexibilidad, manipulación de texto y programación de sistemas en tiempo real

   - COBOL: se convirtió en el primer lenguaje diseñado para negocios. Incorporó la capacidad de manejar grandes volúmenes de datos y la capacidad de procesar operaciones aritméticas eficientemente.
   - ALGOL 60: introdujo la idea de bloques estructurados.
   - SNOBOL: incorporó la manipulación de texto y operaciones de cadenas para manipular texto eficientemente.
   - JOVIAL: se convirtió en el lenguaje estándar de la industria miliar. Incorporó características para programar sistemas en tiempo real, lo que lo hizo adecuado para aplicaciones militares.

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

   - Programación matricial: APL fue uno de los primeros lenguajes en usar matrices y operaciones matriciales como elementos básicos de programación. Fue popular en la comunidad científica y matemática debido a su capacidad de procesar datos matriciales eficientemente.
   - Mejoras en FORTRAN: incorporó la capacidad de hacer operaciones vectoriales y matriciales y la inclusión de subrutinas recursivas.

- Lenguaje de programación para principiantes: BASIC fue uno de los primeros lenguajes diseñados para principiantes. Popular en la computación personal.
- Lenguajes para aplicaciones de sistemas: PL/I fue un lenguaje d diseñado para aplicaciones de sistemas y comerciales. Permitió la programación estructurada y modular.
- Programación oreintada a objetos: SIMULA 67: permitió la definición de clases y objetos, lo que lo hizo muy adecuado para la programación de simulaciones y sistemas complejos.
- Mejoras en ALGOL: incorporó la inclusión de punteros y la capacidad de manejar estructuras de datos complejas.

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

- Pascal: nuevo lenguaje estructurado ue incorporó sintáxis clara y concisa. Fue uno de los primeros lenguajes que incluyó el concepto de tipos de datos definidos por el usuario, lo que permitió a los programadores crear nuevos tipos de datos que se ajustaran a sus necesidades específicas. También introdujo la modularización.
- C: nuevo lenguaje de bajo nivel usado para escribir sistemas operativos, controladores de dispositivos y programas que requieran acceso más cercano al hardware. Lo innovador de C fue la capacidad de trabajar con punteros, definir estructuras complejas y la capacidad de realizar operaciones aritméticas en direcciones de memoria.
- Scheme: lenguaje funcional y dinámico que incorpora un conjunto de primitivas reducido. Introdujo la novción de cierre lo que permitió a los programadores escribir funciones anónimas.
- Prolong: lenguaje lógico usado para resolver problemas de inteligencia artificial.

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

- Smalltalk: lenguaje orientado a objetos caracterizado por la simplicidad y flexibilidad. Incorporó el concepto de objetos y clases. Además fue uno de los primeros en incorporar un entorno de desarrollo integrado (IDE) para poder crear, depurar y ejecutar programas en una sola aplicación.
- Ada: lenguaje seguro y robusto que incorpora características de programación concurrente y la gestión de excepciones.
- FORTRAN 77: incorporó la capacidad de trabajar con matrices y la posibilidad de usar estructuras de control de flujo mas complejas
- ML: caracterizado por ser un lenguaje funcional puro. Permite definir tipos de datos algebráicos y poder trabajar con patrones de datos complejos, incorporó la noción de inferencia de tipos lo que permitió omitir la especificació explícita de tipos en su código.

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

- Smalltalk 80: es altamente interactivo y ofrece un ambiente de desarrollo integrado. Incorporó el concepto de objetos persistentes.
- Turbo Pascal: incorporó el concepto de unidades, lo que permitió hacer programas mñas grandes y complejos fácilmente.
- Postscript: incorporó el concepto de dispositivos de salida independiente, permitió que los documetnos generados en PostScript se impriman en cualquier impresora, además logró la capacidad de incorporar fuentes vectoriales en documentos, lo que permitió que los documentos generados tuvieran una apariencia uniforme en diferentes dispositivos de salida.

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

- FORTRAN 90: incorporación de estructuras de control de flujo adicionales, capacidad de crear matrices dinámicas, capacidad de realizar cálculos en paralelo y de interactuar con otros lenguajes de programación.
- C++: permitió escribir programas tanto en estilo estructurado como orientado a objetos, sobrecarga de operadores, herencia múltiple y constructores y destructores.
- SML: incorporó la inferencia de tipos y las funciones de orden superior, lo que permite escribir programas cortos y legibles y características de programación concurrente y gestión de expreciones.

1991 - 1995: TCL, PERL, HTML.

- TCL: lenguaje de scripting para la automatización de tareas y en la programación de herramientas de software.
- PERL: lenguaje de scripting usado para manipular y en la programación web. Incorpotó el manejo de expresiones regulares, el acceso a bases de datos y la manipulación de archivos.
- HTML: caracterizado por ser un lenguaje para crear páginas web. Incorpora etiquetas que permiten crear hipevínculos, inserción de imágenes y creación de tablas.

1996 - 2000: Java, JavaScript, XML.

- Java: lenguaje multiplataforma para el desarrollo de aplicaciones web, móviles y de escritorio. Incorporo características de gestión automática de memoria, POO, excepciones y manejo de hilos.
- JavaScript: lenguaje de scripting usado para desarrollo de páginas web. Incorporó características coo el acceso al DOM (Document Object Model), la manipulación de eventos, creación de animaciones y el soporte para la POO.
- XML (eXtensible Markup Language): lenguaje para describir datos y documentos estructurados. Incorpora etiquedas que permiten la definición de estructuras de datos y documentos y es usado para la comunicación entre aplicaciones.

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

   Python, la historia del lenguaje de programación Python comienza a fines de los años 80 y principios de los 90, cuando Guido Van Rossum, quien trabajaba en el Centro de Investigación de Matemáticas y Ciencias de la Computación de los Países Bajos, comenzó a trabajar en su implementación como un pasatiempo. Van Rossum había formado parte del equipo de desarrollo del lenguaje de programación ABC, que se enfocaba en ser fácil de usar y aprender manteniendo potencia en su desempeño, pero no había logrado trascender debido a las limitaciones del hardware disponible en la época de su creación. Python se inspiró en ABC y fue diseñado para ser fácil de leer, escribir y entender. El nombre "Python" proviene de la afición de Van Rossum por el grupo de comedia británico Monty Python. Van Rossum sigue siendo el autor principal de Python y desempeña un papel central en la toma de decisiones sobre el lenguaje, siendo conocido en la comunidad de Python como el "Benevolente Dictador Vitalicio" o "Benevolent Dictator for Life".

   Java fue creado en 1991 en Sun Microsystems para ser usado en un proyecto de set-top-box. Inicialmente se llamó Oak, pero luego se renombró como Green y finalmente como Java. Se quería implementar una máquina virtual y un lenguaje con una estructura y sintaxis similar a C++. Se decidió reorientar la plataforma hacia la Web y crear un prototipo de navegador llamado WebRunner, que luego se conocería como HotJava. Proporcionando un lenguaje independiente de la plataforma y

un entorno de ejecución ligero y gratuito para las plataformas más populares. La seguridad del entorno de ejecución y la capacidad de ejecutar applets Java en los navegadores web impulsaron su popularidad. Java ha experimentado numerosos cambios desde la versión primigenia y ha aumentado enormemente el número de clases y paquetes que componen la biblioteca estándar.

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.

   Los atributos que tiene que tener un buen lenguaje de programación es que sea simple, legible, claro en los bindings, confiable, ser confidencial, tener soporte, teer abstracción, ortogonalidad y eficiente.

   Uno de los más destacados es Python porque

   - Es un lenguaje simple y fácil de aprender, tiene sintaxis clara y legible que lo hace ideal para principiantes.
   - Sus bindings son intuitivos y su estructura de código es clara y fáxil de entender.
   - Es confiable y ganó reputación en cuanto a su estabilidad y robustez.
   - Gran comunidad de desarrolladores que dan soporte, documentación y recursos en línea para adyudar a los programdores a resolver problemas.
   - Gran abstracción
   - Es un lenguaje muy ortogonal, significa que los conceptos son coherentes y consistentes en toda la estructura del lenguaje.
   - Es muy eficiente, aprovecha bibliotecas de bajo nivel y modulos optimizados.

4. Tome uno o dos lenguajes de los que usted 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 lenguaje elegido y cuál no posee.

   Expresiones que se pueden escribir en Java

```
/* Declaración y asignación de variables*/
int edad = 25;
String nombre = "Juan";
double altura = 1.75;

/* Operaciones aritméticas básicas*/
int suma = 5 + 3;
int resta = 8 - 2;
double division = 10.0 / 2.0;
int multiplicacion = 4 * 7;

/* Operaciones de comparación*/
boolean esMayor = 10 > 5;
boolean esIgual = 7 == 7;
boolean noEsIgual = 3 != 4;
boolean esMenor = 2 < 6;

/* Estructuras de control*/
if (edad >= 18) {
    System.out.println("Eres mayor de edad");
} else {
    System.out.println("Eres menor de edad");
}

for (int i = 0; i < 10; i++) {
    System.out.println("El valor de i es " + i);
}
```

```
while (altura < 2.0) {
    altura += 0.1;
    System.out.println("La altura ahora es " + altura);
}

/* Uso de métodos y clases*/
String mensaje = "Hola, mundo!";
System.out.println(mensaje);

Scanner scanner = new Scanner(System.in);
System.out.println("Ingresa tu edad:");
int edad = scanner.nextInt();
System.out.println("Tu edad es: " + edad);
```

Organización provista para programas

1. Paquetes: permiten organizar las clases relacionadas en un grupo común. Los paquetes ayudan a evitar conflictos de nombres y a estructurar el código de manera más clara.
2. Modificadores de acceso: permiten controlar el acceso a los miembros de una clase desde otras partes del programa. Los moficadores de acceso son: public, private, protected y default.
3. Interfaces: permiten definir un conjunto de métodos que deben ser implementados por una clase que implemente la interfaz.
4. Clases abstractas.
5. Herencia.
6. Polimorfismo.
7. Excepciones.

Java no cumple con todas las características del ejercicio anterior.

- Simple: Java se diseñó para ser un lenguaje fácil de aprender y usar.
- Legible: se enfoca en una sintaxis clara y fácil de entender.
- Claro en los bindings: Java tiene una forma clara y consistente de manejar las interfaces de programación de apliaciones, lo que hace que sea fácil trabajar con ellas.
- Confiable: Java se diseñó con la seguridad y confiabilidad. El sistema de gestión de memoria de Java ayuda a prevenir errores de memoria comunes, y el modelo de seguridad de Java restringe el acceso a recursos del sistema y promueve un ambiente seguro.
- Confidencialidad: Java permite el uso de modificadores de acceso para controlar la visiilidad de los datos y métodos de una clase.
- Tiene soporte: cuenta con una amplia comunidad de desarrolladores y una gran cantidad de recursos y herramientas disponibles, por lo que es fácil encontrar soporte para cualquier problema que pueda surgir.
- Abstracción: es una característica importante en Java, permite a los desarrolladores trabajar con conceptos abstractos y simplificar la complejidad de los programas.
- Ortogonalidad: la ortogonalidad se logra mediante la capacidad de combinar diferentes componentes del lenguaje para crear programas complejos. Por ejemplo, los programadores pueden utilizar clases y objetos, métodos y herencia para construir programas más grandes.

- Eficiencia: nque puede no ser tan eficiente como otros lenguajes de programación de bajo nivel, como C, ofrece una eficiencia razonable en términos de velocidad de ejecución y uso de recursos.

5. Describa las características más relevantes de Ada. Referido a tipos de datos, tipos abstractos de datos, estructuras de datos, manejo de excepciones, manejo de concurrencia.

   Ada es un lenguaje de programación diseñado para aplicaciones de alta integridad y seguridad crítica. A continuación, se describen algunas de sus características más relevantes:

   - Tipos de datos: Ada ofrece una amplia gama de tipos de datos, incluyendo tipos escalares, como enteros y reales, tipos compuestos, como arrays y registros, y tipos de punto flotante. También tiene tipos definidos por el usuario, que permiten a los programadores definir nuevos tipos de datos para representar conceptos específicos de su aplicación.
   - Tipos abstractos de datos: Ada soporta tipos abstractos de datos, que son tipos definidos por el usuario que encapsulan datos y operaciones relacionadas con esos datos. Esto permite a los programadores definir interfaces claras y bien definidas para sus tipos de datos, lo que a su vez puede mejorar la modularidad y la reutilización del código.
   - Estructuras de datos: Ada ofrece una variedad de estructuras de datos, como arrays, listas enlazadas, pilas, colas y árboles. Estas estructuras de datos pueden ser definidas por el usuario o utilizadas de las bibliotecas estándar de Ada.
   - Manejo de excepciones: Ada tiene un sistema de manejo de excepciones que permite a los programadores capturar y manejar excepciones en tiempo de ejecución. Los programadores pueden definir excepciones personalizadas para sus aplicaciones y especificar cómo se deben manejar las excepciones en diferentes situaciones
   - Manejo de concurrencia: Ada tiene un modelo de concurrencia que permite a los programadores crear programas concurrentes que ejecutan múltiples tareas al mismo tiempo. El modelo de concurrencia de Ada utiliza un enfoque basado en procesos, en el que cada proceso tiene su propio espacio de direcciones y puede comunicarse con otros procesos mediante la comunicación por paso de mensajes.

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

   Java originalmente fue diseñado como un lenguaje para dispositivos electrónicos, específicamente para controlar dispositivos de consumo como televisores y reproductores. Pero a medida que fue desarrollando se convirtió en un lenguaje más amplio, el objetivo principal era proporcionar una plataforma de programación independiente de la plataforma, lo que significa que los programas escritos en Java podrían ejecutarse en cualquier sistema operativo sin cambios.

   Con respecto a la web, algunos de los cambios fueron Servlets y JSP, Applets y seguridad

   - Servlets y JSP: introdujo la tecnología de servlets y JavaServer Pages (JSP), que permiten a los programadores crear aplicaciones web dinámicas y escalables. Los servlets y JSP son utilizados para procesar y responder a las solicitudes del cliente, y pueden interactuar con bases de datos y otras aplicaciones.
   - Applets: permiten a los programadores crear pequeñas aplicaciones que se ejecutan dentro de un navegador web. Los applets proporcionan una forma de enriquecer las páginas web con gráficos interactivos y otras características avanzadas.

- Seguridad: verificación de bytecode y el manejo de excepciones. Estas características ayudan a prevenir errores y proteger contra virus y malware.

Sobre la plataforma en la que se ejecuta, Java es un lenguaje de programación en gran medida independiente de la plataforma, lo que significa que los programas escritos en Java pueden ser portátiles y ejecutarse en diferentes sistemas operativos y arquitecturas sin necesidad de modificaciones. Sin embargo, aún existe cierta dependencia de la plataforma donde se ejecuta, ya que Java se compila en bytecode y requiere una implementación de la Máquina Virtual Java (JVM) para ejecutar los programas. Además, ciertas características del lenguaje de programación Java pueden no ser portables y pueden requerir la escritura de código específico de la plataforma.

7. ¿Sobre qué lenguajes está basado?

   Está basado en gran medida en C++.

8. ¿Qué son los applets?¿Qué son los servlets?

   - Applets: pequeñas aplicaciones que se ejecutan dentro de un navegador web usando la tecnología Java. Están escritos en Java y se pueden incrustar en una página web para proporcionar contenido interactivo y dinámico. Los applets se cargan en el navegador web a través de un archivo HTML que hace referencia al archivo de applet de Java. El navegador web descarga el archivo de applet y lo carga en la Máquina Virtual Java (JVM) para su ejecución.
   - Servlets: son programas Java que se ejecutan en un servidor web para procesar solicitudes y respuestas HTTP. Son usados para crear aplicaciones web dinámicas que generan contenido en función de las solicitudes de los clientes. Los servlets siguen el modelo de programación "request-response" y se ejecutan en un contenedor de servlets. Son importantes para la creación de aplicaciones web en Java.

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

   En C, un programa típico tiene una estructura básica que consta de una o más funciones. Una función es un bloque de código que realiza una tarea específica. La estructura general de un programa en C se puede resumir de la siguiente manera:

```
#include <stdio.h>

// Declaraciones de funciones
int funcion1(int arg1, int arg2);
void funcion2(int arg1, int arg2);

// Función principal
int main() {
    // Declaraciones de variables
    int var1 = 1;
    int var2 = 2;

    // Llamadas a funciones
    int resultado = funcion1(var1, var2);
    funcion2(var1, var2);

    // Salida
    printf("El resultado es %d", resultado);

    // Retorno
    return 0;
}

// Definiciones de funciones
int funcion1(int arg1, int arg2) {
    // Código de la función
```

```
    return arg1 + arg2;
}

void funcion2(int arg1, int arg2) {
    // Código de la función
    printf("La suma es %d", arg1 + arg2);
}
```

En este ejemplo, la estructura del programa se divide en tres partes: las declaraciones de funciones, la función principal y las definiciones de funciones. Las declaraciones de funciones proporcionan información sobre las funciones que se utilizarán en el programa, como su nombre, tipo de valor devuelto y argumentos. La función principal es el punto de entrada del programa y es donde se declaran y utilizan las variables y se llaman a las funciones. Las definiciones de funciones son los bloques de código que definen cómo se realiza una tarea específica.

En cuanto al anidamiento de funciones, sí es posible anidar funciones en C. Esto significa que una función puede contener otra función como una función interna. Sin embargo, el anidamiento de funciones no se utiliza con frecuencia en C y generalmente se considera una práctica de programación poco común. En general, es más común utilizar funciones independientes para realizar tareas específicas en lugar de anidar funciones.

10. Describa el manejo de expresiones que brinda el lenguaje

    En C, las expresiones son combinaciones de valores, operadores y funciones que se evalúan para producir un resultado. La evaluación de expresiones se realiza utilizando una serie de reglas y prioridades de operadores.Las reglas básicas de evaluación de expresiones en C son las siguientes:

    1. Las expresiones entre paréntesis se evalúan primero.
    2. Los operadores de multiplicación, división y módulo se evalúan antes que los operadores de suma y resta.
    3. Los operadores aritméticos se evalúan antes que los operadores relacionales y lógicos.
    4. Los operadores relacionales se evalúan antes que los operadores lógicos.
    5. Los operadores de asignación tienen la menor prioridad y se evalúan por último.

    Además de estas reglas, es importante tener en cuenta que en C, las operaciones se realizan con tipos de datos específicos. Si una operación se realiza con dos operandos de diferentes tipos de datos, C realizará una conversión automática de tipo de datos para hacer que los operandos sean del mismo tipo.

    También es importante recordar que en C, una expresión puede tener efectos secundarios, es decir, puede modificar valores o realizar operaciones fuera de la propia evaluación de la expresión. Por ejemplo, la expresión `a++` incrementará el valor de la variable `a` en 1, pero también devolverá el valor original de `a` antes de la operación de incremento.

11. ¿Qué tipo de programas se pueden escribir con cada Python, Ruby y PHP?¿A qué paradigma corresponde cara uno? ¿Qué características determinan la pertenencia a cada paradigma?

    Python

    Es un lenguaje de programación de propósito general que se puede utilizar para escribir una amplia variedad de programas, incluyendo aplicaciones de línea de comandos, aplicaciones de escritorio, aplicaciones web, ciencia de datos, inteligencia artificial, aprendizaje automático y robótica. Python es un lenguaje versátil que admite múltiples paradigmas de programación y cuenta con una gran cantidad de bibliotecas y herramientas disponibles.

Python es un lenguaje de programación que admite múltiples paradigmas, pero el paradigma predominante en Python es el paradigma orientado a objetos. Este paradigma permite dividir los programas en objetos que interactúan entre sí para resolver un problema. Además, Python también admite otros paradigmas de programación como el imperativo, el funcional y el estructurado.

Un lenguaje de programación *orientado a objetos* se determina por

- Encapsulamiento: la capacidad de agrupar datos y comportamiento en una sola entidad, ocultando la complejidad interna del objeto y protegiendo sus propiedades de cambios no deseados.
- Herencia: la capacidad de crear nuevas clases a partir de clases existentes, lo que permite reutilizar código y extender la funcionalidad de una clase existente.
- Polimorfismo: la capacidad de que diferentes objetos respondan de manera diferente a un mismo mensaje o método, lo que permite escribir código más genérico y flexible.
- Abstracción: la capacidad de modelar objetos del mundo real como entidades abstractas con propiedades y comportamientos relevantes para el problema que se está resolviendo.
- Clases y objetos: los lenguajes orientados a objetos tienen la capacidad de definir clases, que son plantillas para crear objetos. Los objetos son instancias de una clase con valores específicos para sus propiedades.
- Mensajes y métodos: los objetos se comunican enviándose mensajes y ejecutando métodos, que son acciones que los objetos pueden realizar.

Un lenguaje de paradigma *funcional* se determina por

- El énfasis en las funciones como unidad básica de abstracción y composición de programas.
- La inmutabilidad de los datos, lo que significa que los valores de las variables no cambian una vez asignados.
- La evitación de efectos secundarios y la programación sin estado, lo que significa que no se depende del estado global del programa.
- La programación declarativa, en la que se describe qué se quiere hacer, en lugar de cómo se quiere hacer.
- La utilización de funciones de orden superior, que son funciones que reciben otras funciones como argumentos o devuelven funciones como resultado.
- El soporte para la recursión, que es una técnica para definir una función en términos de sí misma.
- La programación concurrente y distribuida, ya que la programación funcional es naturalmente compatible con este tipo de paradigma.

Un lenguaje de paradigma *estructurado* se determina por

- Enfatizar en la utilización de estructuras de control de flujo, como las secuencias, selecciones y ciclos, para controlar el flujo del programa y la ejecución de las instrucciones.
- Evitar el uso de saltos incondicionales, como el GOTO, que pueden hacer que el código sea difícil de leer y mantener.
- La división del programa en módulos o procedimientos con una única entrada y una única salida, con el objetivo de modularizar el código y hacerlo más fácil de entender y mantener.

- El uso de variables y tipos de datos, que permiten la manipulación de la información y la realización de operaciones aritméticas y lógicas.
- La estructuración jerárquica del programa, con subrutinas y funciones que se llaman entre sí para realizar tareas específicas.

Ruby

Ruby es un lenguaje de programación de propósito general que se puede utilizar para escribir una amplia variedad de programas, desde aplicaciones web hasta scripts de línea de comandos, aplicaciones de escritorio, automatización de procesos, juegos y programación de sistemas. Ruby es muy utilizado en el desarrollo de aplicaciones web con el framework Ruby on Rails. Además, se puede utilizar para automatizar tareas en sistemas operativos, crear aplicaciones de escritorio multiplataforma, programar sistemas operativos y dispositivos, entre otras aplicaciones.

Ruby es un lenguaje de programación multi-paradigma, pero su paradigma predominante es el paradigma orientado a objetos. Ruby se destaca por su enfoque en la programación orientada a objetos, su facilidad para trabajar con objetos y su filosofía de "todo es un objeto". Además, Ruby admite otros paradigmas de programación como el paradigma funcional y el paradigma imperativo. Ruby se utiliza principalmente para construir aplicaciones modulares y escalables utilizando patrones de diseño y programación orientada a objetos.

PHP

PHP es un lenguaje de programación utilizado principalmente para el desarrollo de aplicaciones web, incluyendo sitios web dinámicos, sistemas de gestión de contenidos, tiendas en línea y redes sociales. Además, PHP se puede utilizar para escribir scripts de línea de comandos, aplicaciones de consola, aplicaciones de escritorio y servicios web. En resumen, PHP es un lenguaje de programación muy versátil que se utiliza para crear una amplia variedad de programas en diferentes ámbitos.

PHP es un lenguaje de programación multi-paradigma, pero su paradigma predominante es el paradigma imperativo y el paradigma orientado a objetos. En el paradigma imperativo, el enfoque está en describir los pasos que el programa debe seguir para resolver un problema. En el paradigma orientado a objetos, el enfoque está en modelar el mundo real a través de objetos y sus relaciones. Además, PHP también admite otros paradigmas de programación como el paradigma funcional y el paradigma de programación lógica.

Un programa imperativo se caracteriza por

- Enfatizar la descripción de los pasos o instrucciones que el programa debe seguir para resolver un problema.
- Usar estructuras de control, como condicionales y bucles, para controlar el flujo del programa.
- Utilizar variables para almacenar y manipular datos.
- Modificar el estado del programa a medida que se ejecutan las instrucciones.
- Permitir la modularización del código mediante el uso de procedimientos o funciones.
- Tener la capacidad de realizar efectos secundarios, como modificar variables fuera de su alcance o realizar operaciones en el entorno.

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 de datos dinámico: a diferencia de otros lenguajes de programación como C++ o Java, en Python no es necesario declarar explícitamente el tipo de datos de una variable. El tipo de datos se determina automáticamente en tiempo de ejecución, lo que permite una mayor flexibilidad en la programación.
- Indentación significativa: en Python, la indentación de las líneas de código es significativa y se utiliza para delimitar bloques de código. Esto facilita la legibilidad del código y promueve una estructura más clara y organizada.
- Modularidad: Python facilita la organización de los programas en módulos y paquetes, lo que permite separar el código en componentes más pequeños y fáciles de mantener. También existen numerosas bibliotecas y módulos predefinidos que se pueden importar y utilizar en los programas.
- Interpreteado: Python es un lenguaje de programación interpretado, lo que significa que no se necesita compilar el código fuente antes de su ejecución. Esto permite una mayor facilidad en la depuración y una mayor rapidez en el desarrollo.
- Fuerte énfasis en la legibilidad y facilidad de uso: Python se ha diseñado para ser fácil de leer y escribir, con una sintaxis sencilla y clara que facilita la comprensión del código. Además, la filosofía de Python se centra en la legibilidad del código y en la importancia de la comunidad de programadores.

Ruby

- Tipado de datos dinámico: al igual que Python, Ruby también es un lenguaje de tipado dinámico, lo que significa que no es necesario declarar explícitamente el tipo de datos de una variable.
- Sintaxis expresiva y legible: Ruby se ha diseñado para ser fácil de leer y escribir, con una sintaxis clara y expresiva que facilita la comprensión del código.
- Métodos y bloques: Ruby permite definir métodos y bloques de código, que se pueden utilizar para crear abstracciones y facilitar la reutilización de código.
- Mixins: los mixins son una característica única de Ruby que permiten incorporar funcionalidad de un módulo a una clase. Esto permite una mayor flexibilidad en el diseño de las clases y facilita la reutilización de código.
- Organización en módulos y clases: Ruby facilita la organización del código en módulos y clases, lo que permite una mayor modularidad y reutilización de código.
- Meta-programación: Ruby es un lenguaje muy flexible y permite la meta-programación, es decir, la capacidad de modificar y extender el comportamiento del propio lenguaje. Esto permite una mayor flexibilidad y potencia en el diseño de los programas.
- Interactivo: Ruby incluye un intérprete interactivo que permite ejecutar código y realizar pruebas rápidas sin tener que escribir un programa completo.
- Dinámico: Ruby es un lenguaje dinámico, lo que significa que el código se ejecuta en tiempo de ejecución y puede cambiar su comportamiento en función de las condiciones del programa.

PHP

- Tipado de datos débil: PHP es un lenguaje de tipado débil, lo que significa que las variables no están fuertemente vinculadas a un tipo de datos específico y pueden cambiar de tipo durante la ejecución del programa.

- Orientado a objetos: aunque originalmente se diseñó como un lenguaje de programación procedural, PHP ha evolucionado para incluir programación orientada a objetos. Esto permite una mayor modularidad y reutilización del código.
- Sintaxis sencilla y legible: la sintaxis de PHP es sencilla y legible, lo que facilita su aprendizaje y uso. Además, cuenta con una gran cantidad de documentación y recursos en línea.
- Funciones predefinidas: PHP incluye una gran cantidad de funciones predefinidas para realizar tareas comunes, como manejar cadenas de texto, realizar operaciones matemáticas y trabajar con bases de datos.
- Compatibilidad con bases de datos: PHP cuenta con una amplia compatibilidad con diferentes sistemas de gestión de bases de datos, lo que lo hace popular para aplicaciones web que requieren acceso a bases de datos.
- Integración con HTML: PHP se integra fácilmente con HTML y se utiliza comúnmente para generar contenido dinámico en páginas web.
- Modularidad: PHP permite la organización del código en módulos y librerías, lo que permite una mayor modularidad y reutilización de código.
- Interactivo: PHP incluye un intérprete interactivo que permite ejecutar código y realizar pruebas rápidas sin tener que escribir un programa completo.
- Open source: PHP es un lenguaje de código abierto, lo que significa que está disponible para ser utilizado y modificado de forma gratuita por la comunidad de desarrolladores.

Gobstone

- Tipado de datos estático: Gobstones es un lenguaje de tipado estático, lo que significa que las variables están vinculadas a un tipo de datos específico y este no puede cambiar durante la ejecución del programa.
- Orientado a objetos: Gobstones es un lenguaje de programación orientado a objetos, lo que permite una mayor modularidad y reutilización del código.
- Sintaxis sencilla y legible: la sintaxis de Gobstones es sencilla y legible, lo que facilita su aprendizaje y uso.
- Gráficos como elemento central: en Gobstones, los gráficos son un elemento central del lenguaje y se utilizan para representar el estado del tablero y las operaciones que se realizan sobre él.
- Evaluación visual: Gobstones cuenta con una herramienta de evaluación visual que permite a los estudiantes ver los resultados de su programa en tiempo real y visualizar cómo se realiza cada operación en el tablero.
- Basado en el paradigma funcional: aunque Gobstones es un lenguaje orientado a objetos, está diseñado para ser utilizado en el contexto de un paradigma funcional. Esto se logra mediante el uso de funciones y la manipulación de listas y conjuntos.

Processing

Processing es un lenguaje de programación que se utiliza para crear gráficos interactivos y aplicaciones multimedia. Algunas de sus características importantes incluyen:

- Tipado dinámico: Processing utiliza tipado dinámico, lo que significa que el tipo de datos de una variable puede cambiar durante la ejecución del programa.

- Biblioteca gráfica: Processing viene con una biblioteca gráfica integrada que facilita la creación de gráficos 2D y 3D, animaciones y aplicaciones interactivas.
- Organización del programa: En Processing, el programa principal se organiza en dos funciones predefinidas: `setup()` y `draw()`. La función `setup()` se utiliza para inicializar variables, cargar recursos y configurar la ventana de visualización, mientras que `draw()` se utiliza para actualizar y dibujar los elementos gráficos.
- Interacción con otros lenguajes: Processing se basa en Java, lo que significa que se puede interactuar con otras bibliotecas de Java y utilizar las capacidades de procesamiento de Java.
- Comunidad activa: Processing cuenta con una gran comunidad de usuarios que comparten proyectos, tutoriales y recursos en línea. Además, hay una gran cantidad de bibliotecas y herramientas disponibles para su uso en proyectos de Processing

13. ¿A qué tpo de paradigma corresponde JavaScript? ¿A qué tipo de lenguaje pertenece?

    JavaScript es un lenguaje de programación multiparadigma, lo que significa que puede ser utilizado para programar utilizando varios paradigmas, incluyendo el paradigma orientado a objetos, el paradigma funcional y el paradigma imperativo.

    Originalmente, JavaScript fue diseñado como un lenguaje de scripting para páginas web, pero en la actualidad se utiliza en una amplia variedad de contextos, desde el desarrollo de aplicaciones web hasta el desarrollo de aplicaciones móviles y de escritorio. Por lo tanto, se puede considerar un lenguaje de propósito general.

14. Cite otras características importantes de JavaScript: tipado de datos, excepciones, variables, etc.

    - Tipado de datos dinámico: JavaScript es un lenguaje de tipado dinámico, lo que significa que el tipo de datos de una variable puede cambiar en tiempo de ejecución.
    - Variables: En JavaScript, las variables se pueden declarar con la palabra clave `var`, `let` o `const`. Las variables declaradas con `var` tienen un alcance de función, mientras que las variables declaradas con `let` o `const` tienen un alcance de bloque.
    - Excepciones: JavaScript permite la captura de excepciones con un bloque `try-catch`, lo que permite manejar errores y excepciones en tiempo de ejecución.
    - Funciones: JavaScript es un lenguaje de funciones de primera clase, lo que significa que las funciones se pueden pasar como argumentos a otras funciones, se pueden asignar a variables y se pueden devolver como valores de retorno.
    - Eventos: JavaScript tiene una gran capacidad para trabajar con eventos. Es muy utilizado en el desarrollo de páginas web para manipular eventos del navegador, como clics de botones o cambios en el contenido de un elemento.
    - Prototipado: JavaScript es un lenguaje prototipado, lo que significa que las clases no existen como en otros lenguajes orientados a objetos. En su lugar, los objetos heredan de otros objetos, lo que permite una gran flexibilidad en la programación orientada a objetos.
    - Bibliotecas y frameworks: JavaScript cuenta con una gran cantidad de bibliotecas y frameworks que facilitan el desarrollo de aplicaciones web y móviles, como React, Angular, Vue, Node.js, entre otros.