Flujo, ciclos y métodos

Introducción a la programación y Java

{desafío}
latam_

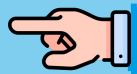
¿Qué aprenderemos en este módulo?

Codificar piezas de software de baja/mediana complejidad en Java utilizando el paradigma de orientación a objetos para resolver una problemática de acuerdo a las buenas prácticas de la industria.

{desafío} latam_

Reconocer las características fundamentales del lenguaje Java para el desarrollo de aplicaciones empresariales

- Unidad 1: Flujo, ciclos y métodos
- Unidad 2: Arreglos y archivos
- Unidad 3: Programación orientada a objetos
- Unidad 4: Pruebas unitarias y TDD



Te encuentras aquí

{desafío} latam_

¿Qué aprenderás en esta sesión?

- *Emplear algoritmos básicos mediante técnicas de pseudocódigos y diagramas de flujo para resolver un problema informático.*
- *Ejecutar Java en el Sistema Operativo para comenzar a crear los primeros programas desarrollados en el lenguaje.*

{desafío} latam_

¿Qué es programar?

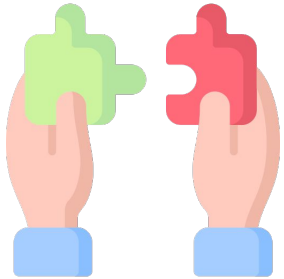
¿Qué es Java?

/* Introducción a la programación */

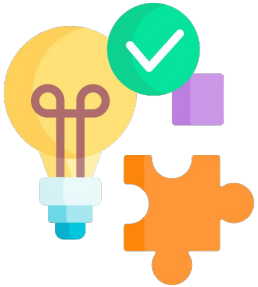
¿Qué es programar?

Es más que escribir código, es:

Analizar problemas



Descomponer en partes



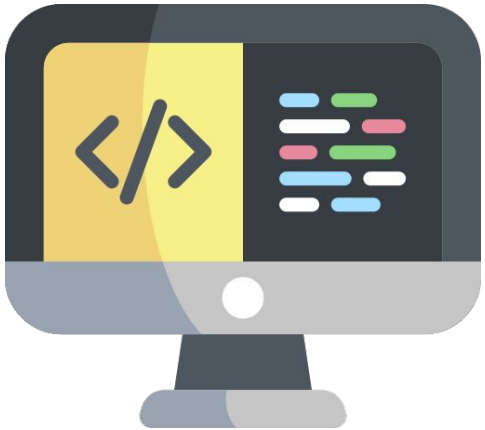
Solucionar el problema

Implementar las soluciones

¿Código o algoritmos?

Código es la codificación en un lenguaje de programación.

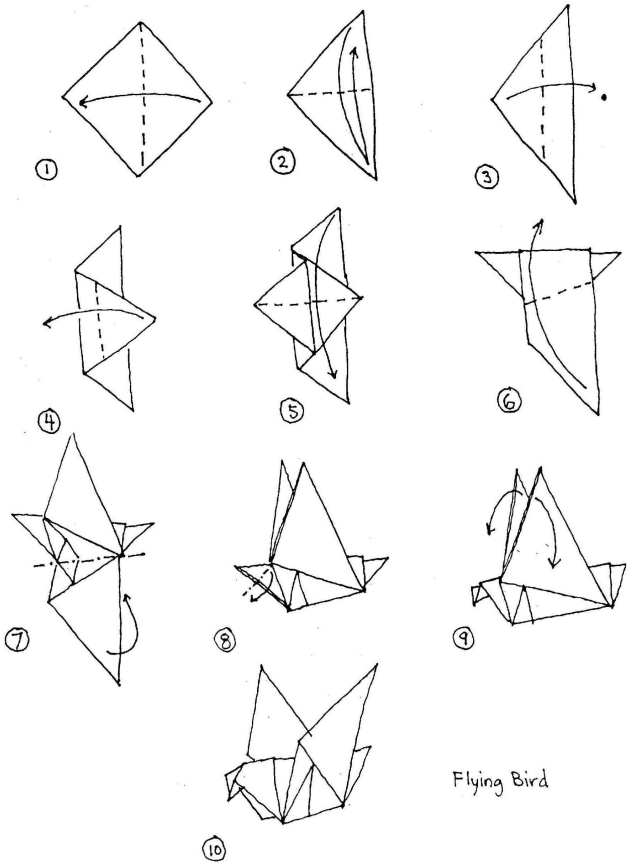
Algoritmo es la secuencia de pasos a realizarse, para solucionar el problema.



Conjunto ordenado de operaciones sistemáticas que permite hacer un cálculo y hallar la solución a un problema.

Ejemplos de algoritmo

- Para ensamblar un mueble debemos seguir todos los pasos del manual de forma secuencial
- Si queremos hacer un pastel, no podemos meter al horno la harina, sin haberla mezclado antes con los otros ingredientes en un orden específico para que quede bien hecha la mezcla.
- En el caso de un modelo de origami, debemos seguir una serie de pasos ordenados.



Resolvamos el siguiente problema: "Hacer un huevo frito"

Pasos para resolver el problema

1. Encender el fuego en la cocina.
2. Poner la paila en el fuego.
3. Colocar aceite en la paila.
4. Romper el huevo.
5. Colocar el huevo en la paila.
6. Colocar sal al huevo.
7. Si se desea huevo revuelto.
    a. Revolver.
8. Si se desea huevo entero.
    a. Tapar la paila.
9. Esperar dos minutos.
10. Apagar el fuego.
11. Servir.

Formas de escribir un algoritmo

1. Diagrama de flujo
2. Pseudocódigo
3. Implementando directamente en algún lenguaje de programación

1. Diagrama de flujo

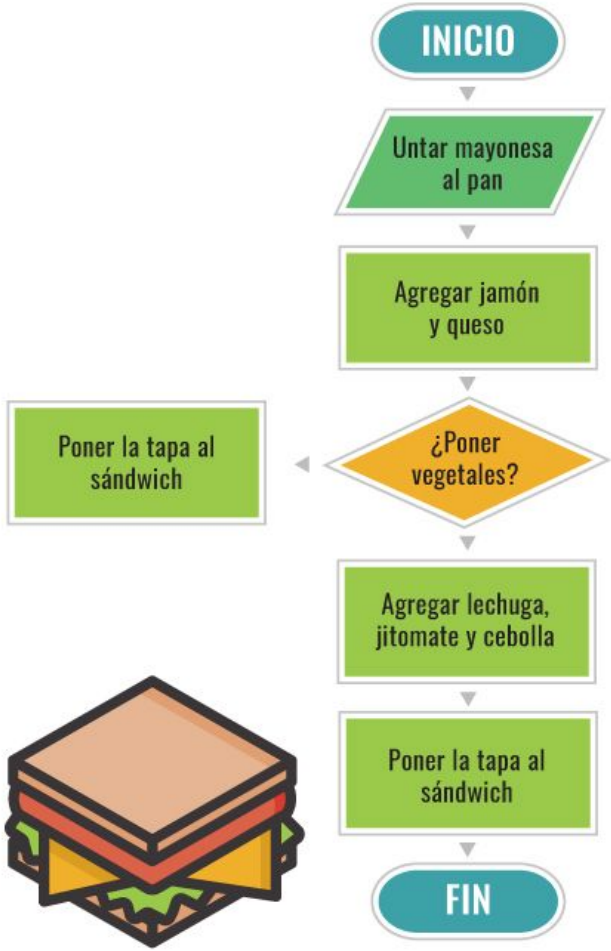
Símbolos

Inicio y fin del proceso

Datos de entrada y salida

Procesos *(instrucciones que se le entrega a la máquina)*

Decisiones

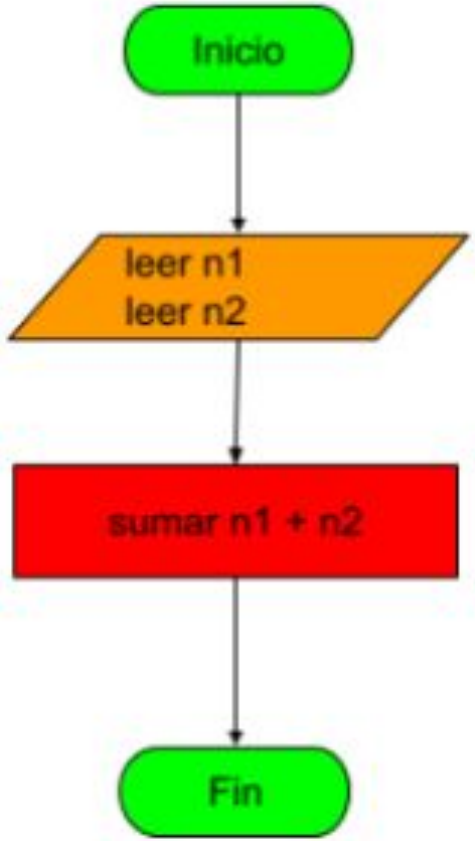


Revisemos ejemplos de diagrama de flujo

Ejemplo 1

Sumar dos números y mostrarlos en la salida

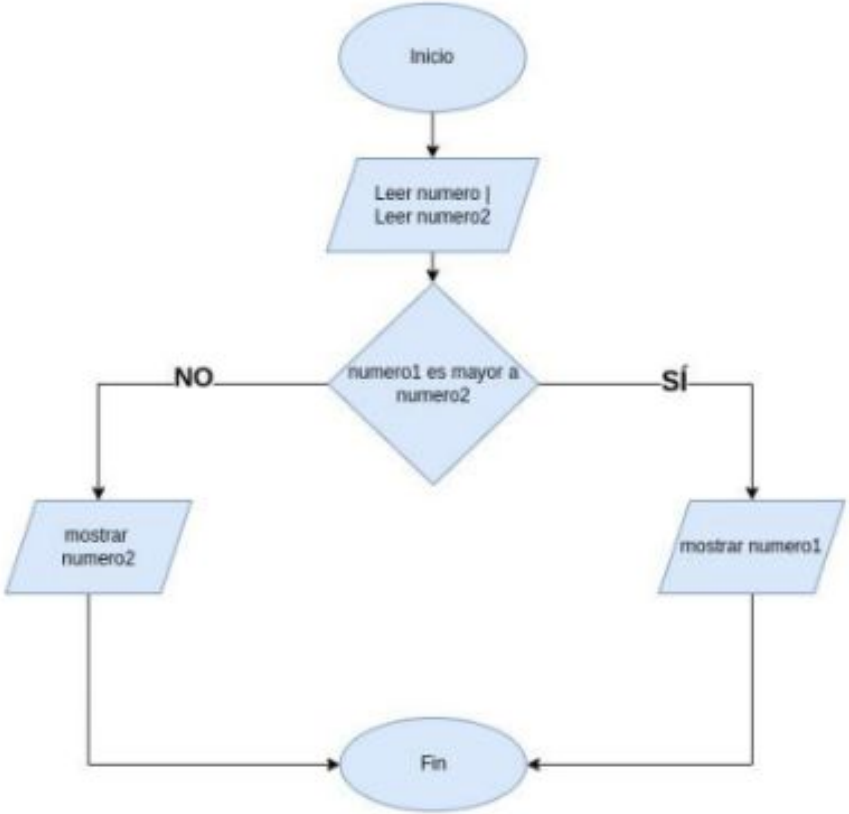
El diagrama representa dos entradas de datos mediante las variables n1 y n2, las cuales guardarán el valor de cada número ingresado y el proceso de sumar, quien se encarga de representar el proceso de sumar dos números en cuestión.



Ejemplo 2

Ingresar dos números y mostrar el mayor de ellos

Representa el algoritmo que da solución al problema de identificar cuál es el número mayor y mostrarlo por pantalla. En este flujo se implementa el rombo de decisión.



{desafío} latam_

Ejercicio guiado

Calcular el IVA

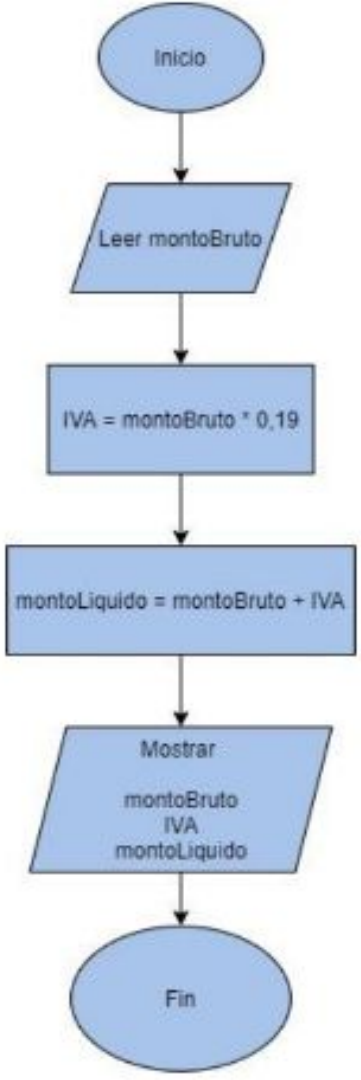
en base al monto total de una factura

Paso 1: Ingresar o leer el monto bruto de la factura.

Paso 2: Calcular el IVA de la factura en base a la siguiente función: `iva = montoBruto * 0,19`

Paso 3: Calcular el monto total o monto líquido de la factura en base a la siguiente función: `montoLiquido = montoBruto + iva`

Paso 4: Mostrar `montoBruto` y `montoLiquido`.



2. Pseudocódigo

- Representación detallada de representar la solución de un algoritmo.
- Parecida al lenguaje que posteriormente se utilizará para la codificación del mismo.
- Permite pensar en términos independientes al lenguaje de programación y concentrarnos en describir lo que estamos tratando de hacer y los pasos necesarios, en lugar de cómo lograrlo.

Ejemplo:

```
Algoritmo Suma
     Leer valor1
     Leer Valor2
     Mostrar valor1 + valor2
FinAlgoritmo
```

En pseudocódigo se utiliza la instrucción `leer` para especificar que el usuario tiene que ingresar un valor y mostrar para imprimir el valor en pantalla.

Revisemos ejemplos de pseudocódigo

Ejemplo 1

Calcular el área de un cuadrilátero

Fórmula: `area = base * altura`

```
Algoritmo AreaCuadrilatero
     Leer base
     Leer altura
     Mostrar base * altura
FinAlgoritmo
```

Ejemplo 2

Mostrar por pantalla cuál de las dos personas es mayor.

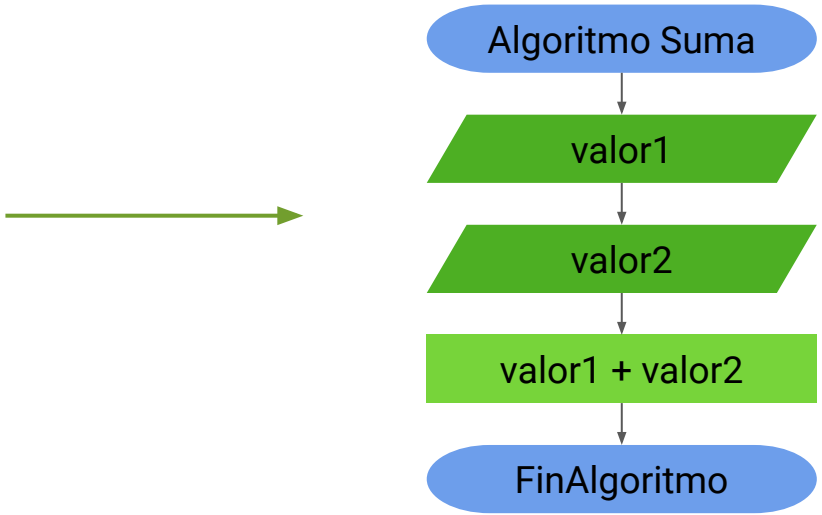
```
Algoritmo QuienEsMayor
     Leer nombrePersona1
     Leer fechaNacimiento1
     Leer nombrePersona2
     Leer fechaNacimiento2
     si fechaNacimiento1 es mayor a fechaNacimiento2 entonces
          Mostrar nombrePersona1 + " es mayor a " + nombrePersona2
     si no
          si fechaNacimiento2 es mayor a fechaNacimiento1 entonces
               Mostrar nombrePersona2 + " es mayor a " + nombrePersona1
          si no
               Mostrar "Las personas tienen la misma edad"

FinQuienEsMayor
```

3. De pseudocódigo a diagrama de flujo

Al ser el pseudocódigo y los diagramas de flujo dos formas de representar un algoritmo, un mismo problema podemos escribirlo de ambas formas.

Ejemplo:

```
Algoritmo Suma
     Leer valor1
     Leer Valor2
     Mostrar valor1 + valor2
FinAlgoritmo
```



Ejercicio propuesto

Crear diagrama de flujo junto a su pseudocódigo

Calcular el promedio de 3 notas y mostrar si el/la estudiante aprueba o reprueba

Requerimientos

1. Ingresar 3 notas entre 1 y 7.

2. Calcular el promedio de notas en base a la siguiente función:

`promedio = nota1 + nota2 + nota3 / cantidadDeNotas` (3 en este caso)

3. Si el promedio es menor a 4 se mostrará por pantalla que el alumno está reprobado.

4. Si el promedio es igual o mayor a 4, se mostrará por pantalla que el alumno está aprobado.

Dado lo visto anteriormente: ¿Qué es para ti el pensamiento lógico?

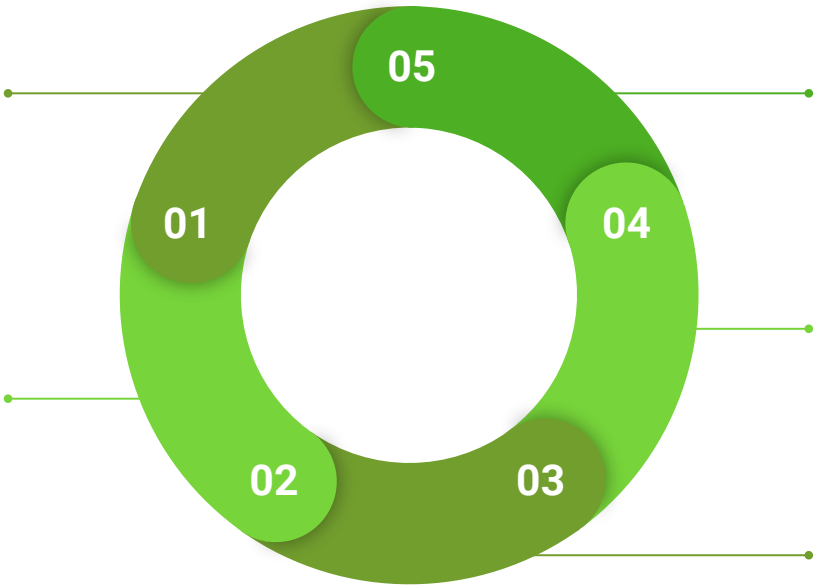
/* Introducción a Java */

{desafío}
latam_

Java permite:

Escribir software en una plataforma y ejecutar virtualmente en otra.

Crear programas que se puedan ejecutar en un explorador y acceder a servicios Web disponibles.



Desarrollar aplicaciones de servidor para foros en línea, almacenes, encuestas, procesamiento de formularios HTML, entre otros.

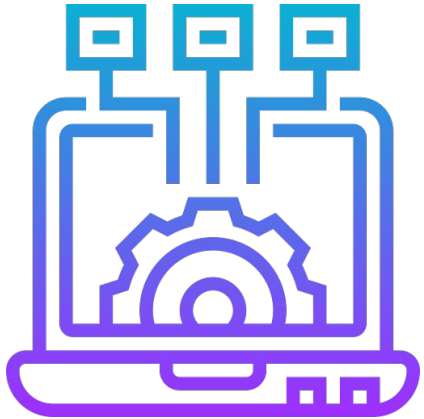
Combinar aplicaciones o servicios que utilizan el lenguaje Java para crear aplicaciones o servicios con un gran nivel de personalización.

Escribir aplicaciones potentes y eficaces para teléfonos móviles, procesadores remotos, microcontroladores, módulos inalámbricos, sensores, gateways, productos de consumo y prácticamente cualquier otro dispositivo electrónico.

JVM

Java Virtual Machine

- Entorno en el que se ejecutan los programas Java.
- Su misión principal es la de garantizar la portabilidad de las aplicaciones Java.
- Cuando se compila una aplicación escrita en lenguaje Java, en realidad no se compila en lenguaje de máquina del sistema operativo del dispositivo, sino a un lenguaje intermedio denominado "Byte Code".



¿Por qué utilizar Java?

- Máquina Virtual de Java (JVM).
- Orientado a objetos, que es el paradigma que más se acerca a la manera de pensar del ser humano.
- No existen problemas con la liberación de la memoria.
- Es relativamente fácil de aprender comparado con otros lenguajes.
- Tiene librerías estándar: Java API
- Variedad de IDEs.

No confundir:

Java JRE

Java Runtime Environment

Paquete que contiene todo lo necesario para correr un programa ya compilado en Java, incluyendo la Java Virtual Machine (JVM), entre otros.

Java JDK

Java Development Kit

Contiene todo lo que trae JRE y, además, incluye el compilador (Javac), por lo que es capaz de crear y compilar programas.

IDE

Ventajas

- Tienen soporte del lenguaje, agregando autocompletados, repositorios.
- Permite debuguear código.
- Muestra los ficheros donde existan errores de sintaxis.
- Conoce las funciones declaradas en la clase.
- Desplazamiento ágil entre las funciones y ficheros

IDE

Elección

- Eclipse (recomendada)
- Apache NetBeans
- BlueJ
- jGRASP
- SlickEdit
- IntelliJ IDEA

Instalación de Java y Eclipse

{desafío} latam_

Instalando Java

Puedes descargar e instalar java desde su documentación oficial.

Acá se presentan los links de descarga para los distintos sistemas operativos.

En caso que la página oficial de java presente problemas, en la plataforma, tendrás ejecutable con el nombre “**Material de apoyo - Instalador de Java**”.

Instalando Java

Instrucciones

- Instalar Java
- Cambiar variables de entorno
- Comprobar la correcta instalación
  - Windows: (cmd) `java --version` y `javac --version`
  - Linux y Mac: (terminal) `java --version` y `javac --version`
- Agregar a las variables de entorno la ruta donde quedo instalado nuestro jdk
  - Panel de control → Sistema → Configuración avanzada del sistema
  - Opciones avanzadas → Variables de entorno
  - Variables de usuario → Crear una nueva variable y la llamaremos JAVA_HOME y el valor de la variable será: C:\Program Files\Java\jdk1.8.0_211

Cabe destacar que la ruta C:\Program Files\Java\jdk1.8.0_211 es referencial, y debe corresponder a la que tendrá cada uno instalada en su computador. Se recomienda ir a la ruta o carpeta y verificar la ruta de instalación.

Instalando Eclipse

- Link descarga
- Instalar Eclipse

¿Pudiste realizar las instalaciones?

Próxima sesión...

- *Comprender las formas de trabajar en Java para ejecutar aplicaciones.*
- *Crear un proyecto para ejecutar bajo la Máquina Virtual de Java instalado.*

{desafío}
latam_

Academia de
talentos digitales