



UNIVERSIDAD SANTO TOMÁS

PRIMER CLAUSTRO UNIVERSITARIO DE COLOMBIA

SECCIONAL TUNJA

VIGILADA MINEDUCACIÓN - SNIES 1732



Acreditación Institucional
Internacional

OTORGADA POR EL IAC CINDA ACUERDO 55 DEL 9 DE MAYO-VIGENCIA 5 AÑOS





UNIVERSIDAD SANTO TOMÁS
PRIMER CLAUSTRO UNIVERSITARIO DE COLOMBIA
SECCIONAL TUNJA

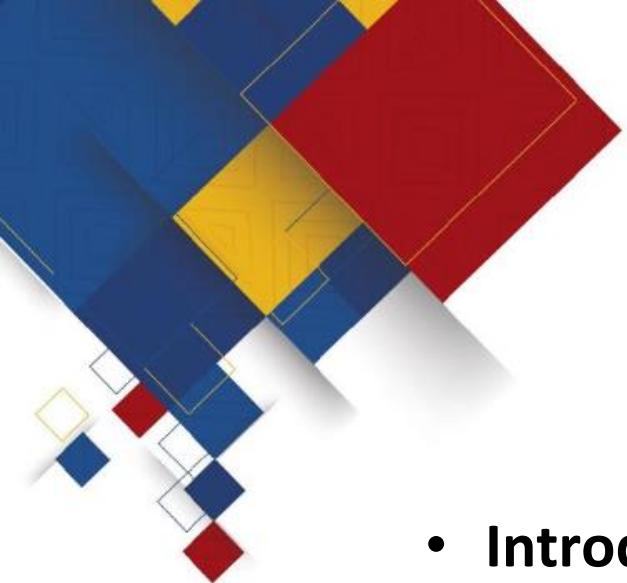
VIGILADA MINEDUCACIÓN - SNIES 1732

Faculty: Systems engineer
Course: Introduction of Programming
Topic: **Fundamentals of software programming**

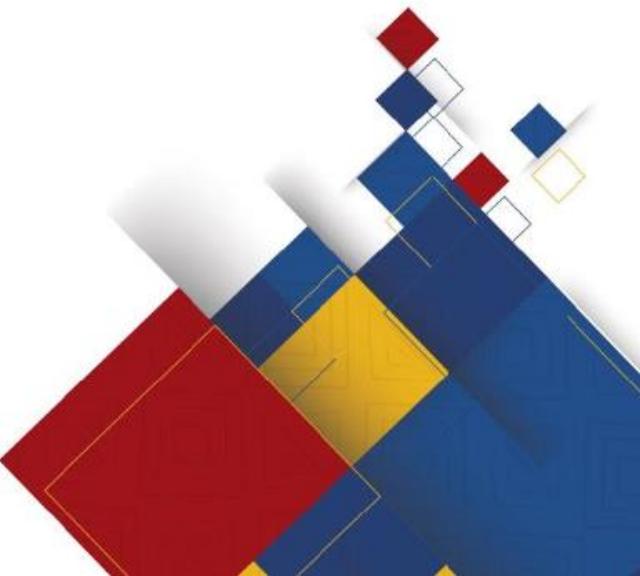
Socializer: Luis Fernando Castellanos Guarín
Karen Daniela Cuervo Cely
Luis.castellanosg@usantoto.edu.co
Karen.cuervo@usantoto.edu.co

Email:
Luis.castellanosg@usantoto.edu.co
Karen.cuervo@usantoto.edu.co

Phone:
3214582098
3105856930



Topics



- **Introduction to CodeGym**
 - How do I register? / Como me registro?
- **Game Rules(work guide)**
- **¡Let's play!**

*¡Siempre
hacia lo alto!*



Introduction to CodeGym





Introduction to CodeGym / Introducción a CodeGym

<https://codegym.cc/>

Empezar a aprender

The screenshot shows the CodeGym website homepage. On the left, there is a vertical sidebar with icons for Course, Tasks, Games, Help, Subscriptions, Success stories, Chat (BETA), and another Chat icon. The main content area has a yellow background with the text "To become a programmer, you need to code". Below this, it says "CodeGym is an online Java programming course that is 80% based on practice". There is an image of a tablet displaying Java code and a notepad with handwritten notes.

¡Siempre
hacia lo alto!



Introduction to CodeGym / Introducción a CodeGym

Registration

 Name E-mail Password

By registering, you accept [the terms of the License Agreement](#)

[Create account](#)

or

Fast registration through social networks



Already have an account? [Sign in](#)

Institutional mail /
Correo institucional



¡Siempre
hacia lo alto!

Introduction to CodeGym / Introducción a CodeGym

 A circular logo featuring a stylized sun or star design with a red and yellow pattern, set against a blue background.

Curso de Java

Search Karen Daniela Cuervo Cely Nivel 0  NO SUBSCRIPTION   

¡Eyl! Tienes algunos asuntos pendientes:

- ✗ Enlazar tu dirección de correo electrónico
- ✗ Confirmar tu dirección de correo electrónico
- ✗ Establecer una contraseña

[Terminar el perfil](#)

[Mapa de misiones](#) [Lecciones](#)



Sintaxis de Java

La misión Sintaxis de Java puede dominarla incluso alguien que no haya programado nunca. Aprenderás sobre clases, objetos, métodos, variables, tipos de datos, matrices (arrays), operadores condicionales y bucles. Echarás un rápido vistazo a las colecciones y a la programación orientada a objetos (POO), y también comenzarás a trabajar en IntelliJ IDEA, un entorno de desarrollo integrado.

¡Siempre hacia lo alto!



Introduction to CodeGym / Introducción a CodeGym

[Inicio](#)[Favoritos](#)[Mi página](#)[Configuración](#)[Descargas](#)[Logros](#)

Karen Daniela Cuervo Cely

Nivel 0  x9

Calificación:  0

Tu suscripción: [NO SUBSCRIPTION](#)

[Inicio](#)[Favoritos](#)[Mi página](#)[Configuración](#)[Descargas](#)[Logros](#)

Karen Daniela Cuervo Cely

Nivel 0  x9

Calificación:  0

Tu suscripción: [NO SUBSCRIPTION](#)

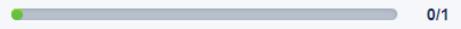
Achievements:



One small step

Complete your first task

 +1

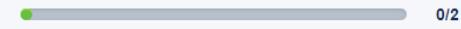
 0/1



Sprint

Complete two tasks in one day

 +1

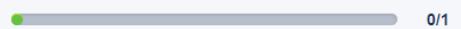
 0/2



Well begun is half done

Complete Level 0

 +1

 0/1

Introduction to CodeGym / Introducción a CodeGym

Course /
Curso

The screenshot shows the CodeGym platform interface. On the left is a sidebar with icons for Course (highlighted with an orange arrow), Tasks, Games, Help, Subscriptions, Success Cases, and Chat. The main area is titled "Curso de Java" and shows the "Mapa de misiones" tab selected. It features a mission card for "Sintaxis de Java" with an image of a robot, a brief description, and a green "DISPONIBLE" button.

Course /
Curso

Curso de Java

Search

Karen Daniela Cuervo Cely
Nivel 0

NO SUBSCRIPTION

Mapa de misiones Lecciones

Sintaxis de Java

La misión Sintaxis de Java puede dominarla incluso alguien que no haya programado nunca. Aprenderás sobre clases, objetos, métodos, variables, tipos de datos, matrices (arrays), operadores condicionales y bucles. Echarás un rápido vistazo a las colecciones y a la programación orientada a objetos (POO), y también empezarás a trabajar en IntelliJ IDEA, un entorno de desarrollo utilizado por programadores de todo el mundo.

DISPONIBLE

Mission /
Mision

¡Siempre
hacia lo alto!

Introduction to CodeGym / Introducción a CodeGym

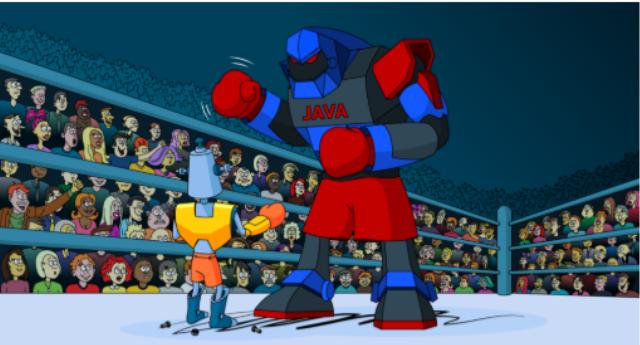


Misión

Search Karen Daniela Cuervo Cely Nivel 0 9x9 NO SUBSCRIPTION

Mapa de misiones Lecciones

Sintaxis de Java



La misión de **Sintaxis de Java** se desarrolló en el centro secreto de **CodeGym**. Consiste en **10 niveles** dedicados a los fundamentos del lenguaje **Java**. Puede dominarla incluso alguien que no haya programado nunca. Aprenderás sobre **clases**, **objetos**, **métodos** y **variables**. Estudiarás **tipos de datos básicos**, **matrices (arrays)**, **sentencias condicionales** y **bucles**. Echa un rápido vistazo a las **colecciones** y a la **programación orientada a objetos (POO)** (*estos temas empezarán a estudiarse en serio en la misión 4*) y también comenzarás a trabajar en **IntelliJ IDEA**, un conocido entorno de desarrollo *utilizado por programadores de todo el mundo*.

No obstante, lo más importante es que realizarás muchas tareas. Las tareas desempeñan un papel fundamental en **CodeGym**. Además, los mentores virtuales te ayudarán con los errores (*tus soluciones se comprueban instantáneamente*).

¡Siempre
hacia lo alto!



Game Rules(work guide)

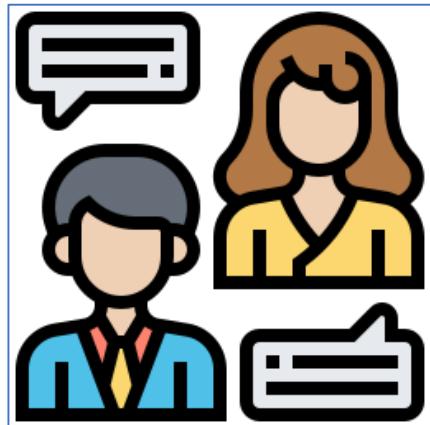
Weekly guide / Guía semanal

Día 1



Gamified platform
work / Trabajo en
plataforma gamificada

Día 2



Survey, clarification of
doubts and practical
exercises proposed /
Sondeo, aclaración de dudas
y Ejercicios prácticos
prouestos

Día 3



Gamified platform
work / Trabajo en
plataforma gamificada

Día 4



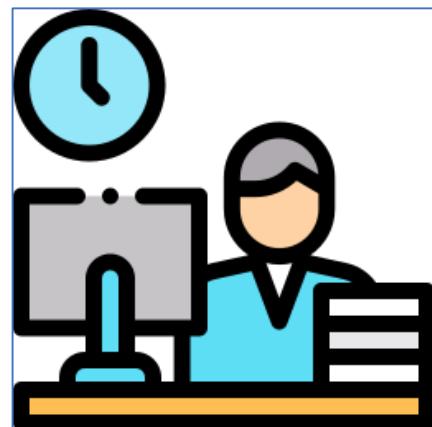
Survey, clarification of
doubts and practical
exercises proposed /
Sondeo, aclaración de dudas
y Ejercicios prácticos
prouestos

¡Siempre
hacia lo alto!

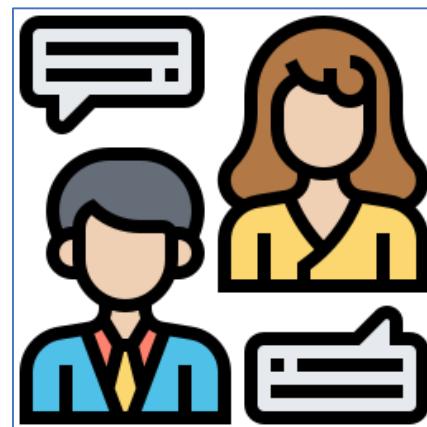


Game Rules(work guide)

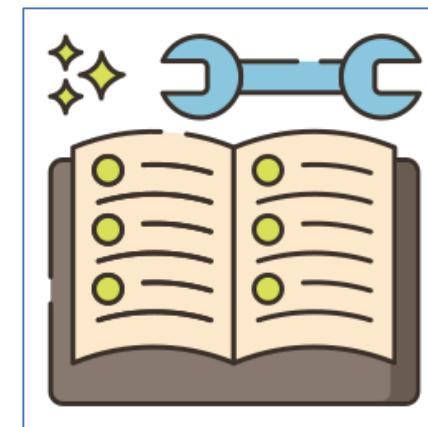
Day 1 and Day 3: Gamified platform work/ Trabajo en plataforma gamificada.



Interacción del estudiante con CodeGym / Student interaction with CodeGym



Retroalimentación / Feedback for each lesson

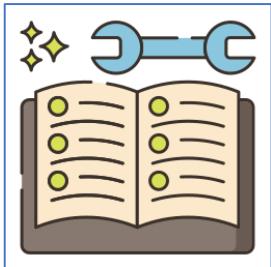


Documentación / Workbook





Game Rules(work guide)



Documentation
(Workbook)

Documentación (Libro de trabajo)

Mostrar el cuadrado de un número

Requirements:

- No cambies la variable numero. Solo puedes cambiar su valor.
- El programa debe generar texto.
- El texto mostrado debe ser un número.
- El texto mostrado debe ser el cuadrado de la variable numero.

Solution.java

```
1 package es.codegym.task.task01.task0140;
2
3 /*
4  * Mostrar el cuadrado de un número
5  */
6
7 public class Solution {
8     public static int numero = 25;
9
10    public static void main(String[] args) {
11        //escribe aquí tu código
12        System.out.println(numero * numero);
13    }
14 }
```

Mi poema favorito

Requirements:

- El programa debe generar texto.
- La línea «Mi poema favorito» debe mostrar antes del verso.
- El texto mostrado debe contener menos de cinco líneas (las espacios también cuentan).
- El texto mostrado debe tener menos de cincuenta caracteres.

The task passed testing!

1	1.50	1
Number of attempts made	Average number of attempts	Your reward

Name of the exercises /
Nombre de los ejercicios

Level /
Nivel

NO_LO_E1

Lesson / Lección

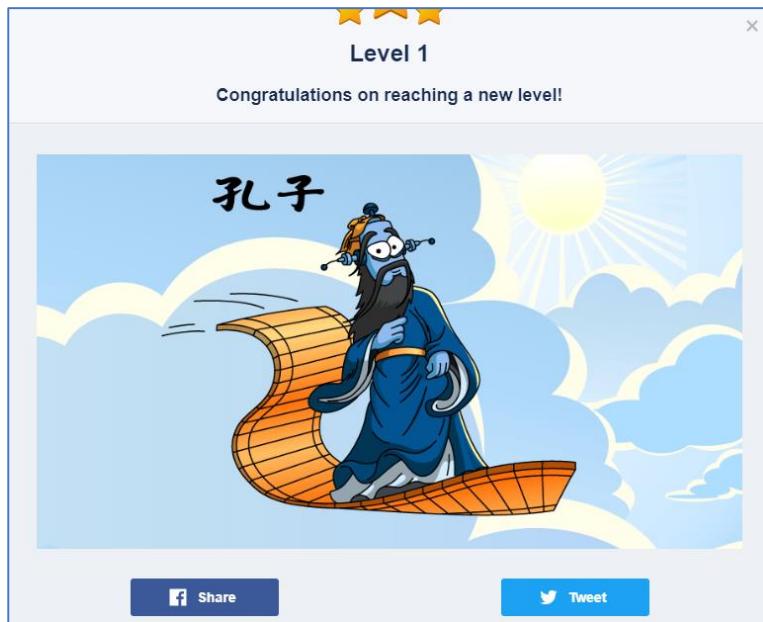
Excercise / Ejercicio

¡Siempre hacia lo alto!



Game Rules(work guide)

When you finish each level (weekly) / Cuando finalices cada nivel (semanalmente).



¡Siempre
hacia lo alto!



Game Rules(work guide)

Day 2 and Day 4: Survey, clarification of doubts and practical exercises proposed / Sondeo, aclaración de dudas y ejercicios prácticos propuestos.



This has no note, It's to clarify doubts / Esto no tiene nota, es para aclarar dudas o falencias.



To practice! / A practicar!

¡Siempre
hacia lo alto!



CodeGym

LET'S
GO!

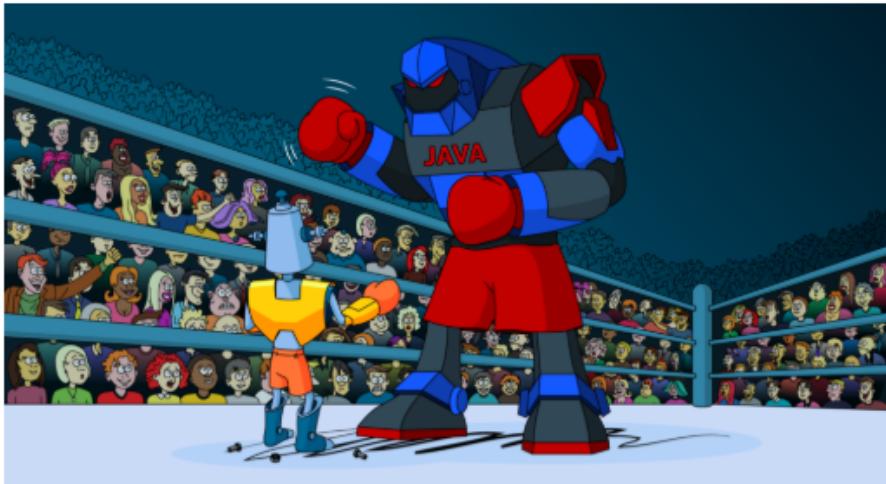


¡Siempre
hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Sintaxis de Java



La misión de **Sintaxis de Java** se desarrolló en el centro secreto de **CodeGym**. Consiste en **10 niveles** dedicados a los fundamentos del lenguaje **Java**. Puede dominarla incluso alguien que no haya programado nunca. Aprenderás sobre **clases**, **objetos**, **métodos** y **variables**. Estudiarás **tipos de datos básicos**, **matrices (arrays)**, **sentencias condicionales** y **bucles**. Echa un rápido vistazo a las **colecciones** y a la **programación orientada a objetos (POO)** (estos temas empezarán a estudiarse en serio en la **misión 4**) y también comenzarás a trabajar en **IntelliJ IDEA**, un conocido entorno de desarrollo utilizado por **programadores de todo el mundo**.

No obstante, lo más importante es que realizarás muchas tareas. Las tareas desempeñan un papel fundamental en **CodeGym**. Además, los mentores virtuales te ayudarán con los errores (*tus soluciones se comprueban instantáneamente*).

Nivel 0

EN CURSO

Introducción



CONTINUAR
How to use CodeGym

¡Siempre
hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 0 - Lesson 0 / Nivel 0 – Lección 0



Programación en Java: aprende partiendo de cero

Sintaxis de Java

Nivel 0, Lección 0



Bueno, aquí estás, jy en una situación difícil! Este oscuro corredor conduce a una lejana puerta (mira, allí hay un montón de robots jóvenes como tú). Detrás de la puerta se encuentra el laboratorio secreto CodeGym, donde vas a aprender Java. No sé nada de ti, ni dónde estudiaste o qué estudiabas. Solo sé que ascender por cuarenta niveles con más de 1200 tareas que hacer te convertirá en programador.

DISPONIBLE



Programación en Java: aprende partiendo de cero

Sintaxis de Java

Nivel 0, Lección 0

DISPONIBLE



Hola. Ya que estás leyendo esto, te confirmaré que, sí, esto son lecciones de Java. Este curso está lleno de oportunidades para el

**¡Siempre
hacia lo alto!**



Gamified platform work / Trabajo en plataforma gamificada.

Level 0 - Lesson 1 / Nivel 0 – Lección 1



Cómo usar CodeGym

Sintaxis de Java
Nivel 0, Lección 1

DISPONIBLE



¡Siempre
hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 0 - Lesson 2 / Nivel 0 – Lección 2



La máquina virtual y el primer comando

Sintaxis de Java
Nivel 0, Lección 2

DISPONIBLE



¡Siempre
hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 0 - Lesson 2 / Nivel 0 – Lección 2

El programa se compone de tres comandos:

```
1 System.out.println("Los humanos y los robots son amigos para siempre");
2 System.out.println("Los humanos y los robots son amigos para siempre");
3 System.out.println("Los humanos y los robots son amigos para siempre");
```

```
public class Hogar
{
    VARIABLE A
    VARIABLE Z
    MÉTODO 1
    MÉTODO N
}
```

```
1 public class Hogar
2 {
3     int a;
4     int b;
5
6     public static void main(String[] args)
7     {
8         System.out.print("1");
9     }
10
11    public static double pi()
12    {
13        return 3.14;
14    }
15 }
```

```
public class Hogar
{
    // Parte inalterable
    public static void main(String[] args)
    {
        CÓDIGO DEL MÉTODO
    }
}
```

¡Siempre
hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 0 - Lesson 3 / Nivel 0 – Lección 3



Listo para el primer programa

Sintaxis de Java
Nivel 0, Lección 3

DISPONIBLE

Comando	Descripción (qué hace)
1 System.out.println(1);	Muestra el número 1 en la pantalla
1 System.out.println("Amigo");	Muestra Amigo en la pantalla
1 System.out.println("Rishi y Amigo");	Muestra Rishi y Amigo en la pantalla

¡Siempre
hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 0 - Lesson 3 / Nivel 0 – Lección 3

Comandos		Qué se mostrará en la pantalla
1	<pre>1 System.out.println("Amigo"); 2 System.out.println("Es El"); 3 System.out.println("Mejor");</pre>	Amigo Es El Mejor
2	<pre>1 System.out.print("Amigo"); 2 System.out.println("Es El"); 3 System.out.print("Mejor");</pre>	AmigoEs El Mejor
3	<pre>1 System.out.print("Amigo"); 2 System.out.print("Es El"); 3 System.out.print("Mejor");</pre>	AmigoEs ElMejor

System.out.println(); VS System.out.print();

¡Siempre
hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 0 - Lesson 3 / Nivel 0 – Lección 3

Creo que ser programador es genial

Verificar Pista

Conditions

Escribe un programa que muestre:
«Creo que ser programador es genial».

Ejemplo de salida:

Creo que ser programador es genial

Solution

```
1 package es.codegym.task.task01.task0101;
2
3 /*
4 Creo que ser programador es genial
5 */
6
7 public class Solution {
8     public static void main(String[] args) {
9         //escribe aquí tu código
10    }
11 }
```

¡Siempre
hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 0 - Lesson 3 / Nivel 0 – Lección 3

Recuerden llevar su bitácora al día

Creo que ser programador es genial

Verificar **Pista**

Conditions

Escribe un programa que muestre: «**Creo que ser programador es genial**».

Ejemplo de salida:

Creo que ser programador es genial

Requisitos:

- ✓ El programa debe generar texto.
- ✓ El texto debe comenzar con «Creo».
- ✓ El texto debe terminar con «es genial».
- ✓ El texto debe constar de 34 caracteres, incluidos espacios.
- ✓ El texto debe constar de una línea.
- ✓ El texto mostrado debe cumplir las condiciones de la tarea.

Solution

```
1 package es.codegym.task.task01.task0101;
2
3 /*
4 Creo que ser programador es genial
5 */
6
7 public class Solution {
8     public static void main(String[] args) {
9         //escribe aquí tu código
10        System.out.println("Creo que ser programador es genial");
11    }
12 }
```

Output

Creo que ser programador es genial

Verificar **Pista**

Conditions

Escribe un programa que muestre: «**Creo que ser programador es genial**».

Ejemplo de salida:

Creo que ser programador es genial

Requisitos:

- ✓ El programa debe generar texto.
- ✓ El texto debe comenzar con «Creo».
- ✓ El texto debe terminar con «es genial».
- ✓ El texto debe constar de 34 caracteres, incluidos espacios.
- ✓ El texto debe constar de una línea.
- ✓ El texto mostrado debe cumplir las condiciones de la tarea.

Solution

```
1 package es.codegym.task.task01.task0101;
2
3 /*
4 Creo que ser programador es genial
5 */
6
7 public class Solution {
8     public static void main(String[] args) {
9         //escribe aquí tu código
10        System.out.println("Creo que ser programador es genial");
11    }
12 }
```

¡La tarea ha superado las pruebas!

Número de intentos realizados: 1

Media de intentos: 1.99

Tu recompensa: 1

Cerrar



Nombre: N0_L3_E1

¡Siempre hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 0 - Lesson 4 / Nivel 0 – Lección 4



Ellie, variables y tipos de datos

Sintaxis de Java
Nivel 0, Lección 4

DISPONIBLE



¡Siempre
hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 0 - Lesson 4 / Nivel 0 – Lección 4

Declaración:

Para declarar una variable: primero tipo, luego nombre		Descripción
1	1 int a;	Crea una variable de tipo int llamada a.
2	1 String s;	Crea una variable de tipo String llamada s.
3	1 double c;	Crea una variable de tipo double llamada c.

Asignación:

Código		Descripción
1	1 i = 3;	Asigna el valor 3 a la variable i.
2	1 a = 1; 2 b = a+1;	Asigna el valor 1 a la variable a. Asigna el valor 2 a la variable b.
3	1 x = 3; 2 x = x + 1;	Asigna el valor 3 a la variable x. En la siguiente línea, el valor de x aumenta en 1, por lo que es igual a 4.



Gamified platform work / Trabajo en plataforma gamificada.

Level 0 - Lesson 4 / Nivel 0 – Lección 4

Tipo de dato		Ejemplo
short	Números cortos	20
Int	Números enteros	4556
long	Números enteros largos	4566566
float	Números flotantes	7.545601
double	Número dos veces más grande que float	7.545601533333333
byte	Manejo de 0 y 1	001110
char	Carácter	'a'
boolean	Booleanos verdad o falso	False
String	Cadenas de caracteres	"Juan David Peña"

re
alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 0 - Lesson 4 / Nivel 0 – Lección 4

	Código compacto	Código equivalente pero más largo
1	<pre>1 int a = 5; 2 int b = 6;</pre>	<pre>1 int a; 2 a = 5; 3 int b; 4 b = 6;</pre>
2	<pre>1 int c = 7; 2 int d = c+1;</pre>	<pre>1 int c; 2 c = 7; 3 int d; 4 d = c+1;</pre>
3	<pre>1 String s = "Soy Amigo";</pre>	<pre>1 String s; 2 s = "Soy Amigo";</pre>

Task Sintaxis de Java, level 0, lesson 4 AVAILABLE ★★★★☆ MEDIUM



x2

Un elogio para la profesora

Todos los profesores del centro secreto CodeGym son estupendos. Los alumnos se dan cuenta de ello desde el principio (nivel 0), pero rara vez lo expresan en voz alta. No obstante, a todos nos gusta que nos digan algo bonito, ¿verdad? El mejor cumplido para cualquiera que enseñe programación es hacerle un programa. Si además el programa imprime la frase correcta, ¡no tiene precio! Vamos a escribir un programa con un cumplido.

[Open](#)

Task Sintaxis de Java, level 0, lesson 4 AVAILABLE ★★★★☆ MEDIUM



x2

Mi joven amigo

En 3126, todos conocen System.out.println() y lo usan a menudo. No solo los estudiantes de CodeGym que empiezan a aprender a programar con este comando, sino hasta los niños. Vamos a escribir un programa que muestre el verdadero año de nacimiento de mi amigo. Nació hace ocho años.

[Open](#)

¡Siempre
hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 0 - Lesson 4 / Nivel 0 – Lección 4

★ Excercise: N0_L4_E1

Un elogio para la profesora

Verify Help Discuss New/Reset Run Code analysis

Conditions Class tree

Escribe un programa que muestre «**Ellie es inteligente**» cinco veces, cada una de ellas en una nueva línea.

Ejemplo de salida:

```
Ellie es inteligente
```

Solution.java

```
1 package es.codegym.task.task01.task0102;
2
3 /*
4  * Un elogio para La profesora
5  */
6
7 public class Solution {
8     public static void main(String[] args) {
9         //escribe aquí tu código
10    }
11 }
12 }
13 }
```

¡Siempre
hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 0 - Lesson 4 / Nivel 0 – Lección 4

★ Excercise: N0_L4_E2

Mi joven amigo

Verify Help Discuss New/Reset Run Code analysis

Conditions Class tree

Ahora mismo estamos en el 3126. Mi amigo nació hace ocho años. Escribe un programa que muestre el año de nacimiento de mi amigo.

Requirements:

- El programa debe generar texto.
- El año mostrado debe tener cuatro dígitos.
- El año mostrado debe comenzar por «31».
- El año mostrado debe cumplir las condiciones de la tarea.

Solution.java

```
1 package es.codegym.task.task01.task0103;
2
3 /*
4  * Mi joven amigo
5  */
6
7 public class Solution {
8     public static void main(String[] args) {
9         //escribe aquí tu código
10    }
11 }
12
13 }
```

¡Siempre
hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 0 - Lesson 4 / Nivel 0 – Lección 4

★ Excercise: N0_L4_E3

Declarar variables

Conditions Class tree

Escribe un programa que declare las siguientes variables en el método

```
main:  
String nombre, int edad y String  
ciudad.
```

Nota: «Declarar una variable» es lo mismo que «crear una variable».

Requirements:

- Declara una variable de tipo String llamada nombre.
- Declara una variable int llamada edad.
- Declara una variable de tipo String llamada ciudad.
- Debe haber tres variables declaradas.

Verify Help Discuss New/Reset Run Code analysis

Solution.java

```
1 package es.codegym.task.task01.task0105;  
2  
3 /*  
4 Declarar variables  
5 */  
6  
7 public class Solution {  
8     public static void main(String[] args) {  
9         //  
10    }  
11 }  
12
```

¡Siempre
hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 0 - Lesson 5 / Nivel 0 – Lección 5



¿Qué es un compilador?

Sintaxis de Java
Level 0, Lesson 5

AVAILABLE



¡Siempre
hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 0 - Lesson 5 / Nivel 0 – Lección 5



01101000 01101111 01101100 01100001
00100000 01100010 01110101 01100101
01101110 01101111 01110011 00100000
01100100 11000011 10101101 01100001
01110011



“hola buenos días”

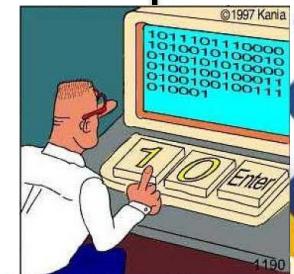
las computadoras solo pueden entender en binario (0 y 1)

¡Siempre
hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 0 - Lesson 5 / Nivel 0 – Lección 5

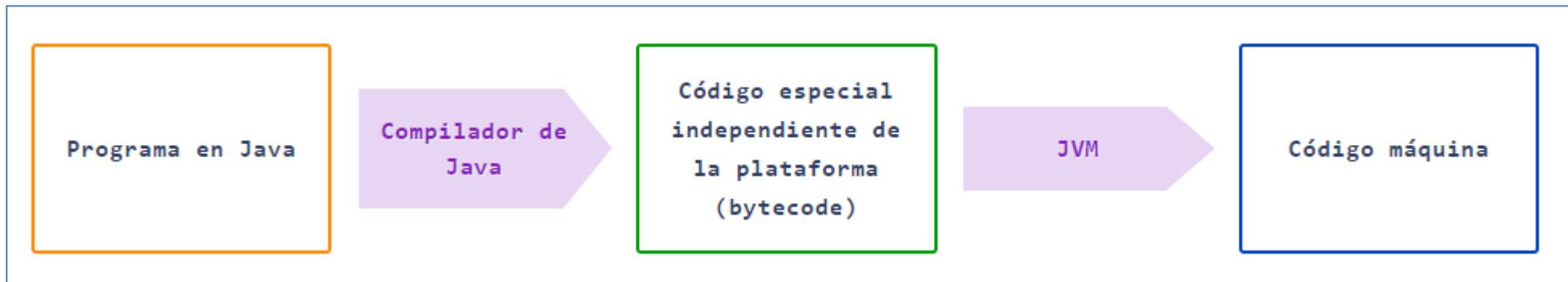
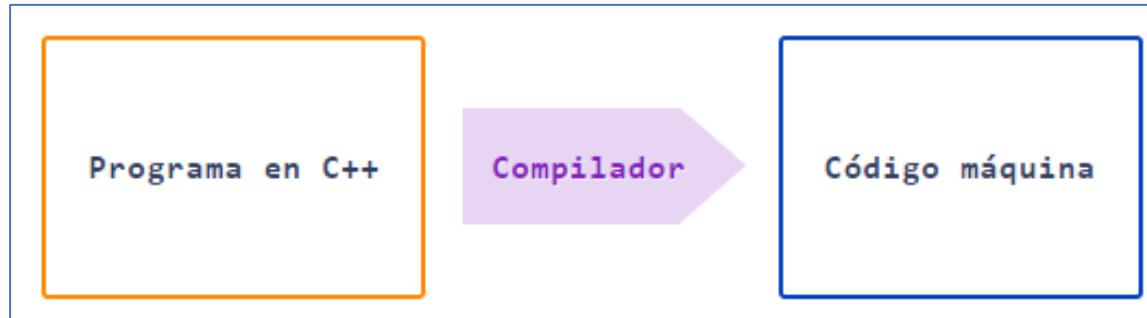


siempre
hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 0 - Lesson 5 / Nivel 0 – Lección 5



¡Siempre
hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 0 - Lesson 6 / Nivel 0 – Lección 6



Te presentamos a Kim

Sintaxis de Java
Level 0, Lesson 6

AVAILABLE



¡Siempre
hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 0 - Lesson 6 / Nivel 0 – Lección 6

```
1 public class Hogar
2 {
3     public static void main(String[] args)
4     {
5         /*
6             Ahora vamos a mostrar la frase "Amigo Es El Mejor" en la pantalla
7             */
8         System.out.print("Amigo ");
9         System.out.print("Es ");
10        System.out.print("El ");
11        System.out.print("Mejor");
12    }
13 }
```

```
1 public class Hogar
2 {
3     public static void main(String[] args)
4     {
5         System.out.print("Amigo ");
6         System.out.print("Es El "); // Esto también es un comentario
7         System.out.print("Mejor");
8     }
9 }
```



¡Siempre
hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 0 - Lesson 7 / Nivel 0 – Lección 7



Lo que no te enseñan en el colegio

Sintaxis de Java
Level 0, Lesson 7

AVAILABLE



¡Siempre
hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 0 - Lesson 8 / Nivel 0 – Lección 8



Resumen del nivel 0

Sintaxis de Java
Level 0, Lesson 8

AVAILABLE



¡Siempre
hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 0 - Lesson 8 / Nivel 0 – Lección 8

Hoy has aprendido sobre lo siguiente:

- Variables
- Mostrar texto en la pantalla
- Tipos int y String
- La diferencia de compilar en Java respecto a otros lenguajes
- Añadir comentarios al código y por qué los necesitamos

EASY

Task Sintaxis de Java, level 0, lesson 8

CLOSED ★★★★☆



x1

Errores y características

Algun día serás un verdadero programador y tendrás tu propio (casi) evaluador. Por supuesto, eso queda bastante lejos, pero nunca es demasiado pronto para aprender: de momento, ¡recuerda una frase que te será útil cuando hables con tu evaluador! Tomemos el comando de salida por pantalla y visualicemos con convicción: «No es un error, es una característica».

Open

MEDIUM

CLOSED ★★★★☆



x2

2 + 3 = 5

Es obvio que $2 + 3 = 5$, ¿verdad? Por lo que parece, no siempre es así: nuestro programa de prueba es incomprendible, porque algún genio de programador añadió líneas de más. Devuelve el equilibrio al mundo: comenta algunas líneas para mostrar « $2 + 3 = 5$ » en la pantalla.

Open

¡Siempre
hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 0 - Lesson 4 / Nivel 0 – Lección 4

★ Excercise: N0_L8_E1

Errores y características

Verify Help Discuss New/Reset Run Code analysis

Conditions Class tree

Solution.java

```
1 package es.codegym.task.task01.task0106;
2
3 /*
4  * Errores y características
5  */
6
7 public class Solution {
8     public static void main(String[] args) {
9         //escribe aquí tu código
10    }
11 }
12 }
```

Escribe un programa que muestre «**No es un error, es una característica.**»

Ejemplo de salida:

No es un error, es una característica.

¡Siempre
hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 0 - Lesson 4 / Nivel 0 – Lección 4

★ Excercise: N0_L8_E2

2 + 3 = 5

Verify Hint

Conditions

Comenta unas pocas líneas para mostrar
2 más 3 es igual a 5
en la pantalla.

Nota: No comentes las líneas con declaraciones de variables.

Requirements:

- El programa debe mostrar «2 más 3 es igual a 5» en la pantalla.
- No cambies la línea que declara las variables.
- Tienes que comentar al menos una línea.
- No cambies ni añadas comandos responsables de la salida. Puedes comentarlos.

Solution

```
1 package es.codegym.task.task01.task0108;
2
3 /*
4 2 + 3 = 5
5 */
6
7 public class Solution {
8     public static void main(String[] args) {
9         int a = 3, b = 2;
10
11         System.out.print("dos");
12         System.out.print(b);
13         System.out.print(" más ");
14         System.out.print(" menos ");
15         System.out.print(a);
16         System.out.print("tres");
17         System.out.print(" es igual a ");
18         System.out.print(" igual a ");
19         System.out.print("cinco");
20         System.out.print(a + b);
21     }
22 }
```

¡Siempre
hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 0 - Lesson 4 / Nivel 0 – Lección 4

★ Excercise: N0_L8_E3

Sin comentarios

Verify Hint

Conditions

Quita el comentario de una línea para que el programa muestre los números **12 y 2** (primero 12 y luego 2).

Requirements:

- El programa debe mostrar los números 12 y 2.
- No cambies las líneas que declaran variables.
- El programa debe tener solo dos variables int.
- No cambies las líneas responsables de la salida por pantalla.
- Debes quitar el comentario de una línea y dejar sin modificar las demás.

Solution

```
1 package es.codegym.task.task01.task0107;
2
3 /*
4 Sin comentarios
5 */
6
7 public class Solution {
8     public static void main(String[] args) {
9         int x = 2;
10        int y = 12;
11
12        // y = x * y;
13        // y = x + y;
14
15        x = y - x;
16        y = y - x;
17
18        System.out.println(x);
19        System.out.println(y);
20    }
21
22 }
```

¡Siempre
hacia lo alto!



Feedback

LET'S
GO!



¡Siempre
hacia lo alto!



Survey



¡Siempre
hacia lo alto!



Clarification of doubts



¡Siempre
hacia lo alto!



To practice!

LET'S
GO!



¡Siempre
hacia lo alto!



P1T29: System.out.print ()

Display an original figure on the screen,
along with the name of each one in relief,
using the System.out.print () command.



P1T30: Identifiers

Identify which of the following identifiers are valid in Java.

Identifica cuál de los siguientes identificadores son válidos en Java.

Identificador	¿Válido? (SI/NO)	¿Por qué?
producto1		
1producto		
\$tasa		
primer-nombre		
678#		
Saldo__		
barça		
for		
primer nombre		
tallaPantalon		
Precio_camisa		
Tom's		
n0m8r3		
%interes		
\$_Sancho		
talla.zapato		
pension?		
else		



P1T31: Sum

Write a program that declares three variables with different value assignment, and adds them.

Escriba un programa que declare tres variables con asignación de valores diferentes, y los sume.





P1T32: Difference between dates

Write a program that receives two years of birth and print your difference.

Escriba un programa que reciba dos años de nacimiento e imprima su diferencia.





P1T33: Operations with variables

Write a Java program that does the following: declare two variables X and Y of type int, two variables N and M of type double and assign each one a value. Then show by screen.

Escribe un programa Java que realice lo siguiente: declarar dos variables X e Y de tipo int, dos variables N y M de tipo double y asigna a cada una un valor. A continuación, muestra por pantalla.



P1T33: Operations with variables

- o The value of each variable.
- o The sum $X + Y$
- o The difference $X - Y$
- o The product $X * Y$
- o The X / Y ratio
- o The rest $X \% Y$
- o The sum $N + M$
- o The difference $N - M$
- o The product $N * M$
- o The quotient N / M
- o The rest $N \% M$
- o The sum $X + N$
- o The Y / M ratio
- o The rest $Y \% M$
- o Double of each variable
- o The sum of all variables
- o The product of all variables

- o El valor de cada variable.
- o La suma $X + Y$
- o La diferencia $X - Y$
- o El producto $X * Y$
- o El cociente X / Y
- o El resto $X \% Y$
- o La suma $N + M$
- o La diferencia $N - M$
- o El producto $N * M$
- o El cociente N / M
- o El resto $N \% M$
- o La suma $X + N$
- o El cociente Y / M
- o El resto $Y \% M$
- o El doble de cada variable
- o La suma de todas las variables
- o El producto de todas las variables

Siempre
hacia lo alto!



P1T34: The Paradise supermarket

The paradise supermarket requires building a payment point application. The application declares 4 variables that correspond to the value of each of the products registered by a customer, the application must add the values of the four products and add VAT of 19%. In the invoice the customer can observe the value of the products, the value of VAT and the total to pay

El supermercado paraíso requiere construir una aplicación de punto de pago. La aplicación declara 4 variables que corresponden al valor de cada uno de los productos registrados por un cliente, la aplicación deberá sumar los valores de los cuatro productos y agregar el IVA de 19%.

En la factura el cliente podrá observar el valor de los productos, el valor del IVA y el total a pagar.

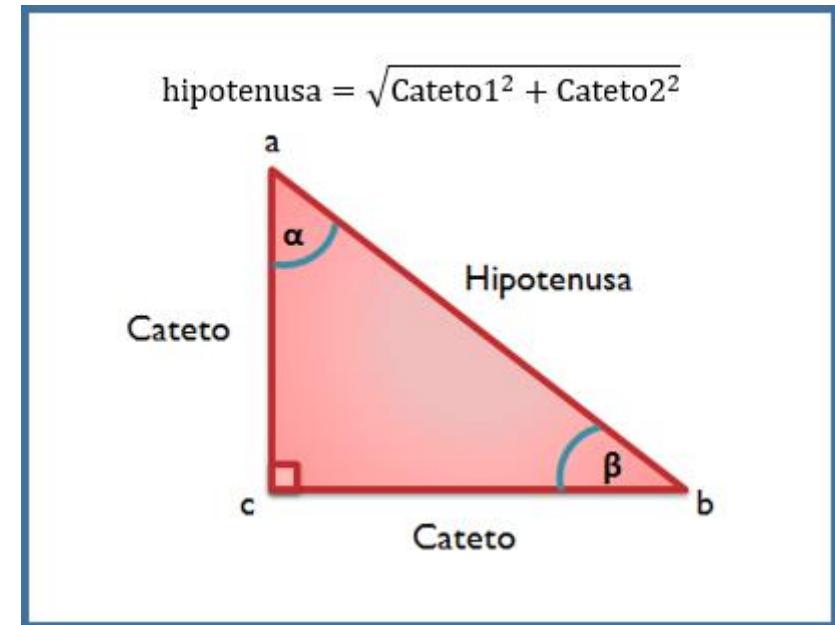
¡Siempre
hacia lo alto!



P1T35: Hypotenuse

Make a program to calculate the hypotenuse of a triangle, declaring the variables of the two legs.

Realice un programa para calcular la hipotenusa de un triángulo, declarando las variables de los dos catetos.

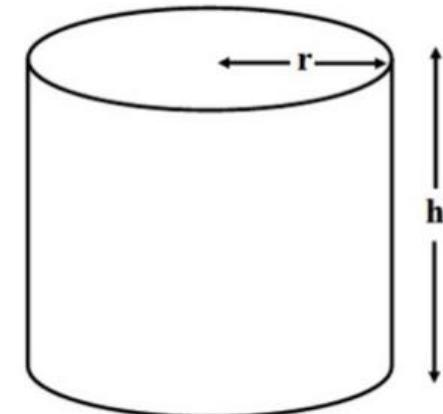




P1T36: Volume of a cylinder

Build a program that calculates the volume of a cylinder knowing the radius and height.

$$V = \pi r^2 h$$



VOLUMEN DE UN CILINDRO

Construya un programa que calcule el volumen de un cilindro conociendo el radio y la altura.

¡Siempre
hacia lo alto!



P1T37: Student's grades

Perform an algorithm that receives a student's grades in the semester, the grades have the following weights: 1st exam 20%, 2nd exam 25%, 3rd exam 25%, 4th exam 30%. With the notes and the weighted calculate the final semester grade.

Realice un algoritmo que reciba las notas de un estudiante en el semestre, las notas tienen los siguientes ponderados: 1° examen 20%, 2° examen 25%, 3° examen 25%, 4° examen 30%. Con las notas y el ponderado calcular la nota final del semestre.





P1T40: Operations with String

A program that receives a person's name is required to perform the following operations:

- o Search if the last name with the letters “gomez” exists in the string.
- o Convert all letters to uppercase.
- o Replace character a with the at sign.
- o Cut the String from position 7 to 12 and print this substring

Se requiere un programa que reciba el nombre de una persona para realizar las siguientes operaciones:

- o Buscar si dentro de la cadena existe el apellido que tenga las letras “gomez”.
- o Convertir todas las letras en mayúscula.
- o Reemplazar el carácter a por el símbolo arroba.
- o Cortar el String desde la posición 7 a la 12 e imprimir este substring.

¡Siempre
hacia lo alto!



Preparcial

Diseñe un software en JAVA que tenga las siguientes características:

- Menú

```
-----  
Dr. coronavirus  
-----  
This software helps determine if you have coronavirus (COVID-19)  
-----  
created by: xxxxxxxxxxxxxxxxxxxxxxxxx  
-----|
```

Variables que el software debe pedir:

- Tos: SI (1) / NO (0)
- Dolor corporal: SI (1) / NO (0)
- Dificultad para respirar: SI (1) / NO (0)
- Fiebre: # grados
- Edad

Si tiene tos seca, dolor corporal, dificultad para respirar, fiebre superior a 38 grados : El usuario puede tener COVID19
debe mantenerse aislado y seguir rigurosamente las indicaciones de salud.

Si además de eso es mayor de 60 años y tiene dificultad para respirar: debe contactar de inmediato a su EPS. De lo contrario solo debe permanecer en casa



Preparcial

Decisión

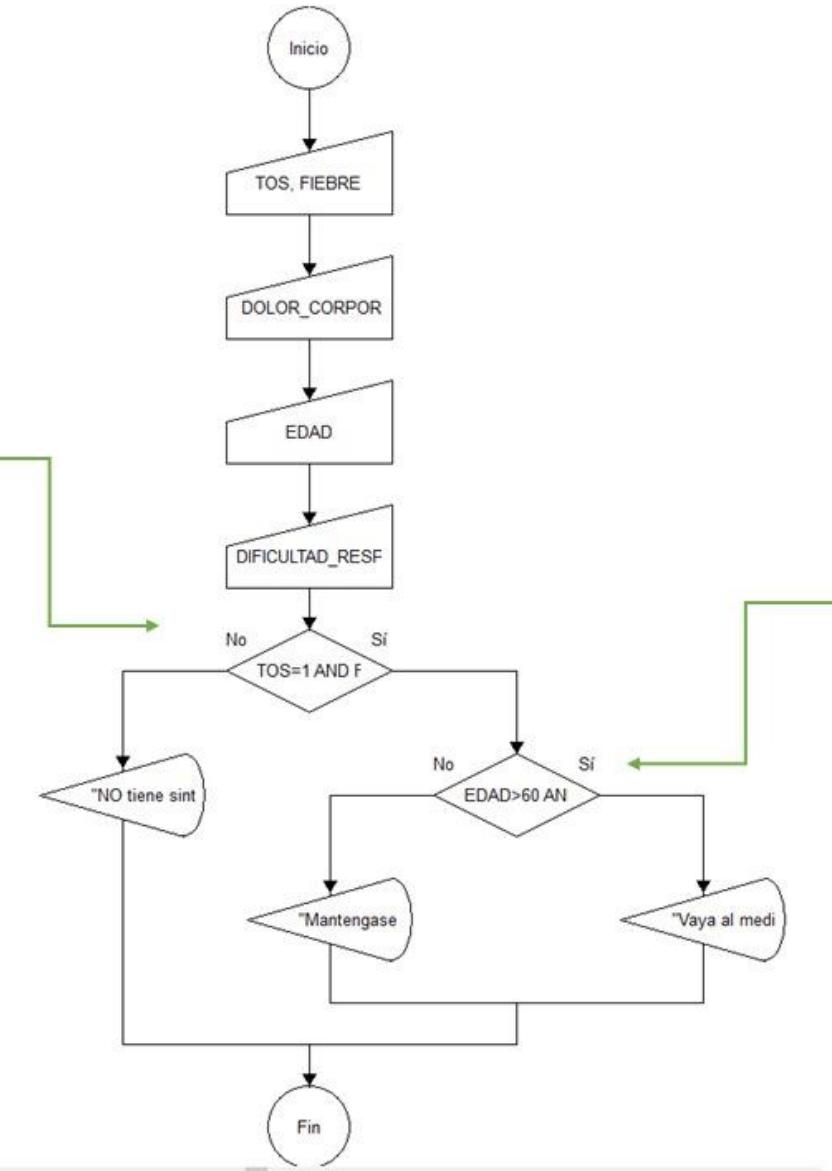
Condición:

```
TOS=1 AND FIEBRE > 38 AND  
DOLOR CORPORAL=1
```

Condición verdadera

Derecha
 Izquierda

Aceptar Cancelar



Decisión

Condición:

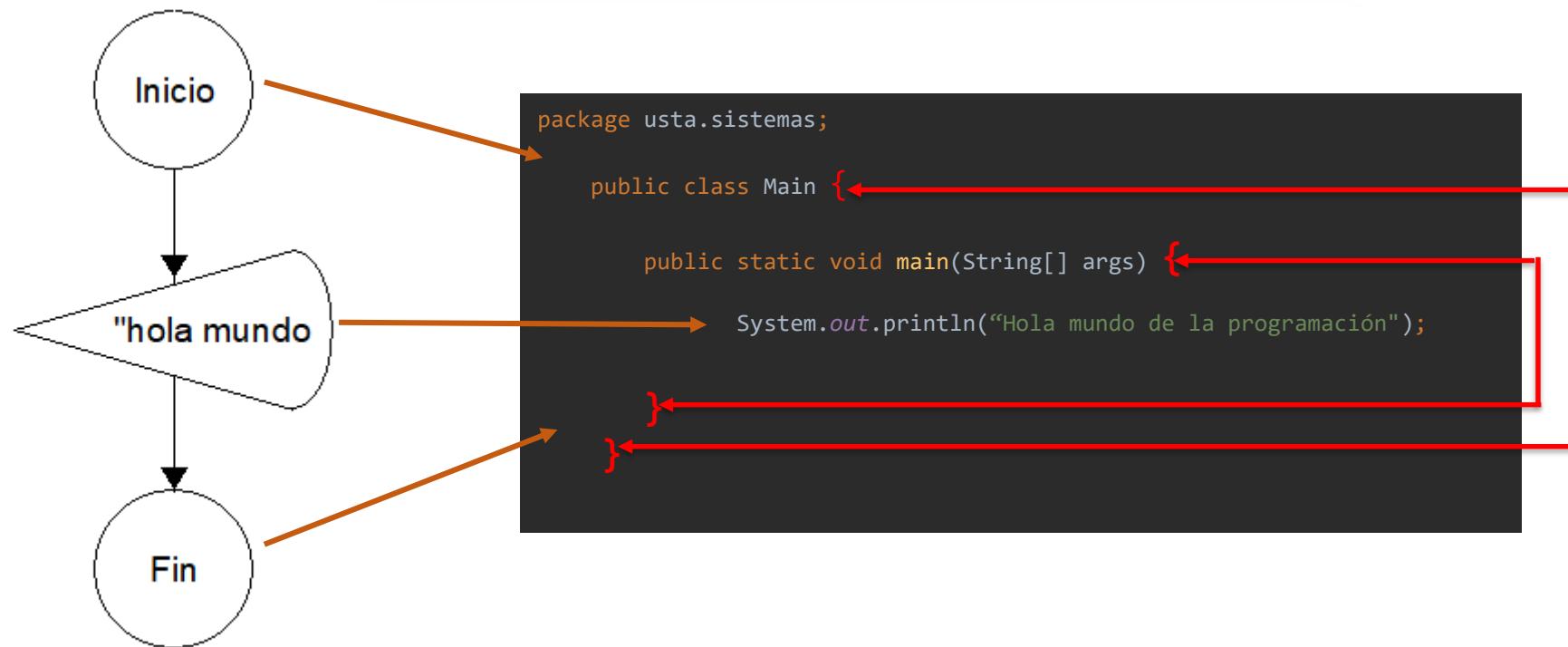
```
EDAD>60 AND DIFICULTAD RESP=1
```

Condición verdadera

Derecha
 Izquierda

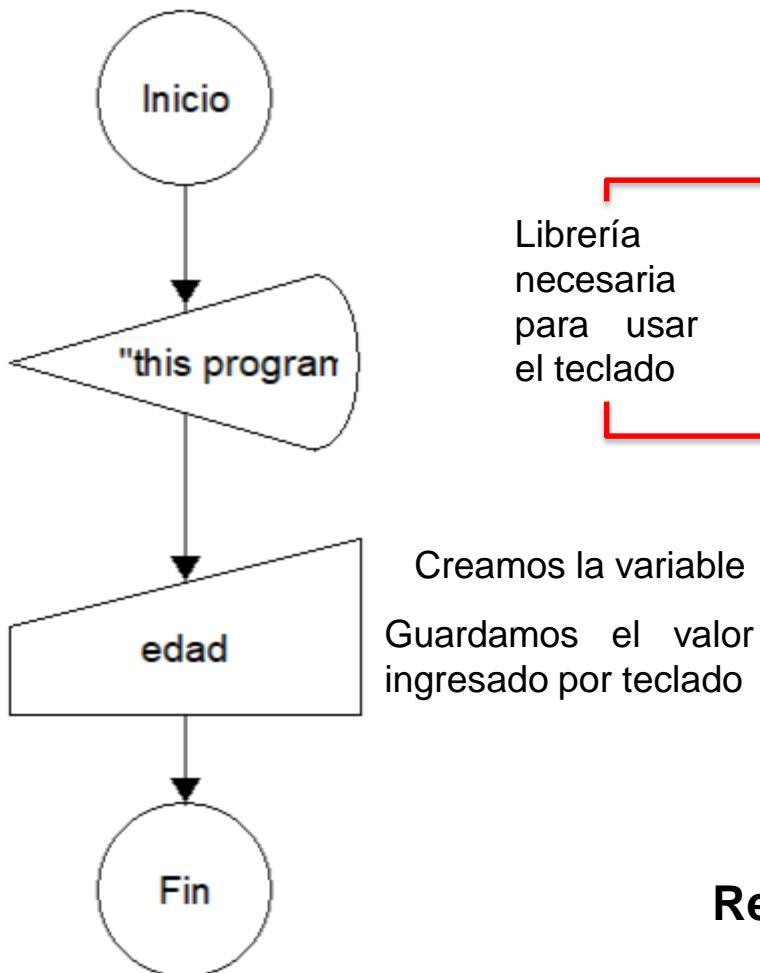
Aceptar Cancelar





Recomendaciones:

- Todo corchete que se abre **{** se debe cerrar **}**.
- Toda línea de código que no finalice con un corchete debe tener un **;**



Librería necesaria para usar el teclado

Creamos la variable
Guardamos el valor ingresado por teclado

```
package usta.sistemas;  
  
import java.util.Scanner; //libreria necesaria para usar el teclado de la computadora.  
  
public class Main {  
  
    public static void main(String[] args) {  
  
        Scanner keyboard = new Scanner(System.in);  
  
        System.out.println("This program, calculate How old are you, input your age:");  
  
        int edad;  
        edad = keyboard.nextInt();  
  
    }  
}
```

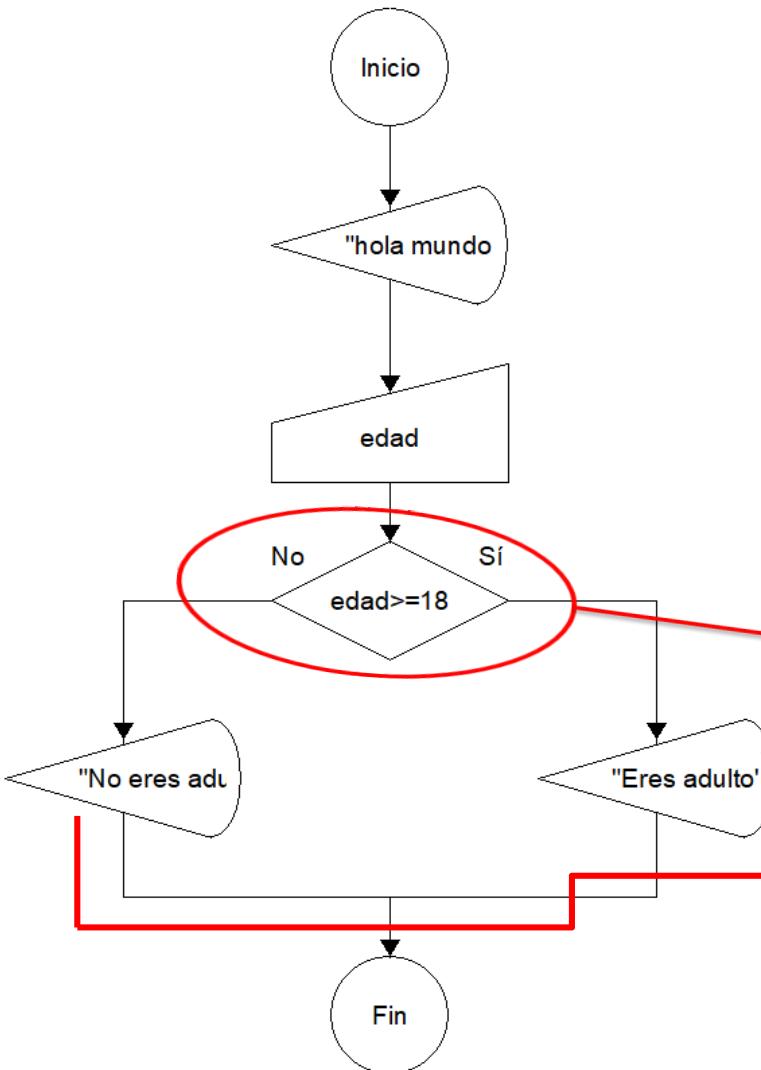
Recomendaciones:

- Si se requiere trabajar con el teclado debe importar la librería Scanner.
- Toda variable se debe definir y tener precaución que datos se van a guardar.



DFD Vs JAVA

If / else



```
package usta.sistemas;

import java.util.Scanner; //libreria necesaria para usar el teclado de la computadora.

public class Main {

    public static void main(String[] args) {

        Scanner keyboard = new Scanner(System.in);

        System.out.println("This program, calculate How old are you, input your age:");

        int edad;
        edad = keyboard.nextInt();

        if (edad>=18){
            System.out.println("Eres adulto");
        }else{
            System.out.println("No eres adulto");
        }
    }
}
```



2° Session CodeGym

LET'S
GO!



¡Siempre
hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 1 - Lesson 0 / Nivel 1 – Lección 0



¡Has subido de nivel!

Sintaxis de Java
Level 1, Lesson 0

AVAILABLE



¡Siempre
hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 1 - Lesson 1 / Nivel 1 – Lección 1

```
1      NOMBRE DEL PAQUETE
2 package com.futujava.leccion2;
3 import java.io.IOException;
4 /*
5  * Usuario: General
6  * Fecha: 21/12/12
7  * Hora: 11:59
8 */
9      NOMBRE DE LA CLASE
10 public class Tarea1
11 {
12
13     private static String TEXTO = "Bésame el retropropulsor"; | VARIABLE DE LA CLASE
14
15
16     public static void main(String[] args) throws IOException
17     {
18         SALIDA POR PANTALLA           COMENTARIO DE UNA SOLA LÍNEA
19         System.out.println(TEXTO); //Muestra una sola cadena
20         COMENTARIO DE VARIAS LÍNEAS
21         /*
22             Este es un comentario de varias líneas.
23             El siguiente código mostrará tres cadenas idénticas. | MÉTODO main()
24         */
25         DECLARACIÓN DE LA VARIABLE
26         String s = ";Ja, ja, ja!";
27         LLAMADA AL MÉTODO
28         imprimeTextoVariasVeces(s, 3);
29
30
31         PARÁMETROS DEL MÉTODO
32         public static void imprimeTextoVariasVeces(String s, int contador) | FIRMA DEL MÉTODO
33
34
35     {
36         FIN
37     }
38 }
```

¡Siempre
hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 1 - Lesson 2 / Nivel 1 – Lección 2



Lo bueno de Java

Sintaxis de Java
Level 1, Lesson 2

AVAILABLE

«Hola, Amigo, mi mejor alumno novato (bueno, uno de los mejores). ¿Cómo te va? He oído que acabas de crear tu primer programa Java... Genial, ¿verdad?».

«Sí, ha sido estupendo... muy chulo, y me gusta, pero...».

«Me parece percibir un atisbo de duda en tu voz. ¿Algún problema?».

«No, no, va todo bien. Solo quería preguntarte una cosa. ¿Por qué es Java el lenguaje adecuado? He oido que algunos lenguajes de programación son más fáciles de aprender que otros, algunos son mejores para sitios web... o para desarrollar juegos. ¿Para qué es mejor Java?».

«¡Buena pregunta! La respuesta es simple y compleja al mismo tiempo. Empezaré por la parte compleja».

«Java está en todas partes. Casi en cualquier lugar puedes encontrar Internet, aplicaciones móviles, sistemas operativos, sistemas integrados, software en tiempo real, minería de datos, informática en la nube, etc.».

nudia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 1 - Lesson 3 / Nivel 1 – Lección 3



Direcccionamiento de memoria y variables

Sintaxis de Java
Level 1, Lesson 3

AVAILABLE

	A	B	C	D	E	F	G	H
1								
2		13			11			
3								
4	A	M	I	G	O			
5	R	I	S	H	I			
6								
7								
8								

¡Siempre
hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 1 - Lesson 3 / Nivel 1 – Lección 3

Código	Explicación
1 <pre>1 String s;</pre>	Esta línea crea una variable llamada <code>s</code> . Declaramos su tipo como <code>String</code> , porque se utilizará para almacenar texto. No podemos declarar otra variable con el mismo nombre en la misma función o clase.
2 <pre>1 string s2 = "Soy Diego";</pre>	Esta línea crea una variable de tipo String llamada <code>s2</code> e inmediatamente le asigna el valor <code>Soy Diego</code> .
3 <pre>1 int a;</pre>	Aquí creamos una variable llamada <code>a</code> . Su tipo de datos coincide con lo que se almacenará en ella. El tipo de datos <code>int</code> es una abreviatura de <code>integer</code> , que quiere decir “entero”.
4 <pre>1 int b = 4;</pre>	Creamos una variable llamada <code>b</code> . Su tipo de datos (<code>int</code>) sirve para almacenar enteros. El valor <code>4</code> se asigna inmediatamente a la variable.



Gamified platform work / Trabajo en plataforma gamificada.

Level 1 - Lesson 3 / Nivel 1 – Lección 3

★ Excercise: N1_L3_E1

Solemamente publicamos y declaramos estas variables...

Verify Help Discuss New/Reset Run Code analysis

Conditions Class tree

Declarla las variables int llamadas **a** y **b**. En la misma línea en que declaras las variables, asignales inmediatamente distintos valores. Esos valores pueden ser cualquier número entero.

Solution.java

```
1 package es.codegym.task.task01.task0109;
2
3 /*
4 Solemnemente publicamos y declaramos estas variables...
5 */
6
7 public class Solution {
8     public static void main(String[] args) {
9         //escribe aquí tu código
10    }
11 }
12 }
```

¡Siempre
hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 1 - Lesson 3 / Nivel 1 – Lección 3

★ Excercise: N1_L3_E2

Una variable en nuestra pantalla

Verify Help Discuss New/Reset Run Code analysis

Conditions Class tree

Crea una variable de tipo **String** llamada **nombre**.
En la misma línea donde declaras la variable, asignale inmediatamente algún valor.
Muestra la variable **nombre** en la pantalla.

Solution.java

```
1 package es.codegym.task.task01.task0110;  
2  
3 /*  
4 Una variable en nuestra pantalla  
5 */  
6  
7 public class Solution {  
8     public static void main(String[] args) {  
9         //escribe aquí tu código  
10    }  
11 }  
12 }
```

¡Siempre
hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 1 - Lesson 3 / Nivel 1 – Lección 3

★ Excercise: N1_L3_E3

Una variable no es suficiente

Verify Help Discuss New/Reset Run Code analysis

Conditions Class tree

Declarar las variables int **a** y **b** y la variable de tipo String **s**. En la misma línea en que declaras las variables, asignales inmediatamente distintos valores.

Requirements:

- Declarar dos variables int.

Solution.java

```
1 package es.codegym.task.task01.task0111;
2
3 /*
4 Una variable no es suficiente
5 */
6
7 public class Solution {
8     public static void main(String[] args) {
9         //escribe aquí tu código
10    }
11 }
12 }
```

¡Siempre
hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 1 - Lesson 3 / Nivel 1 – Lección 3

★ Excercise: N1_L3_E4

Mostrar el cuadrado de un número

Verify Help Discuss New/Reset Run Code analysis

Conditions Class tree

Se proporciona la variable **numero**. Escribe un programa que **muestre el cuadrado de esta variable**: (**numero * numero**).

Requirements:

- No cambies la variable numero. Solo puedes cambiar su valor.
- El programa debe generar texto.

Solution.java

```
1 package es.codegym.task.task01.task0140;
2
3 /*
4  * Mostrar el cuadrado de un número
5  */
6
7 public class Solution {
8     public static int numero = 25;
9
10    public static void main(String[] args) {
11        //escribe aquí tu código
12    }
13}
14}
```

Siempre
hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 1 - Lesson 4 / Nivel 1 – Lección 4



Introducción a ints y Strings

Sintaxis de Java
Level 1, Lesson 4

AVAILABLE

«¡Hola, Amigo!».

«Hola, Eleanor Carrey».

«Llámame Ellie. No hay por qué ser tan formales».

«Vale, Ellie».

¡Siempre
hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 1 - Lesson 4 / Nivel 1 – Lección 4

Ejemplo 1, código:	Descripción
1 string s;	Se declara una nueva variable <code>s</code> . Puede almacenar texto.
1 int i;	Se declara una nueva variable <code>i</code> . Puede almacenar enteros.

Assignment / Asignación

Ejemplo 2, código:	Descripción
1 String s = "Ellie";	La variable <code>s</code> almacena la cadena <code>Ellie</code> .
1 int i = 5;	La variable <code>i</code> almacena el número 5.



Gamified platform work / Trabajo en plataforma gamificada.

Level 1 - Lesson 4 / Nivel 1 – Lección 4

Ejemplo 3, código:	Descripción
1 int a = 5;	La variable a almacena el valor 5.
1 int b = 6;	La variable b almacena el valor 6.
1 int c = a + b;	La variable c almacena el valor 11.

Ejemplo 7, código:	
1	1 System.out.println("Un hombre ha de hacer lo que ha de hacer");
2	1 String s = "Un hombre ha de hacer lo que ha de hacer"; 2 System.out.println(s);



Gamified platform work / Trabajo en plataforma gamificada.

Level 1 - Lesson 4 / Nivel 1 – Lección 4

★ Excercise: N1_L4_E1

Solo valdrá si da 20

Verify Help Discuss New/Reset Run Code analysis

Conditions Class tree

Solution.java

```
1 package es.codegym.task.task01.task0137;
2
3 /*
4 Solo valdrá si da 20
5 */
6
7 public class Solution {
8     public static int a = 1;
9     public static int b = 3;
10    public static int c = 9;
11    public static int d = 27;
12
13    public static void main(String[] args) {
14
15        int resultado = + a + b + c + d;
16
17        System.out.println(resultado);
18    }
19}
```

En el método **main**, pon correctamente los **signos más y menos** para que la variable **resultado** dé **20**. Solo debes colocar los signos en la línea donde se declara la variable **resultado**. No cambies el orden de las variables en esta línea. Cada variable debe ir precedida por un signo **más o menos**.

Requirements:

- Valores de las variables: no cambies a, b, c ni d.

¡Siempre
hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 1 - Lesson 4 / Nivel 1 – Lección 4

★ Excercise: N1_L4_E2

CodeGym: apréndelo una vez y úsallo en cualquier lugar

Verify Help Discuss New/Reset Run Code analysis

Conditions Class tree

Escribe un programa que muestre:
«**CodeGym: apréndelo una vez y úsalo en cualquier lugar**» diez veces.

Requirements:

- El programa debe generar texto.
- El texto debe comenzar con «CodeGym».

Solution.java

```
1 package es.codegym.task.task01.task0114;
2
3 /*
4  * CodeGym: apréndelo una vez y úsalo en cualquier Lugar
5  */
6
7 public class Solution {
8     public static void main(String[] args) {
9         //escribe aquí tu código
10    }
11 }
12 }
```

¡Siempre
hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 1 - Lesson 4 / Nivel 1 – Lección 4



Excercise: N1_L4_E3

Eh... ¡Feliz año nuevo!

Verify Help Discuss New/Reset Run Code analysis

Conditions Class tree

Solution.java

Elimina los comentarios de parte del código para que aparezca «**Feliz Año Nuevo**» en la pantalla.

Requirements:

- El programa debe mostrar texto en la pantalla.
- Tienes que quitar el comentario de algunas líneas. No cambies nada más.
- No añadas nuevas líneas al programa ni elimines las existentes.

```
1 package es.codegym.task.task01.task0115;
2
3 /*
4 Eh... ¡Feliz año nuevo!
5 */
6
7 public class Solution {
8     public static void main(String[] args) {
9         // String s = "Feliz";
10        // System.out.println("Navidad");
11        // System.out.println("Feliz Nuevo");
12        // System.out.print("fin de semana");
13        // System.out.print("Feliz Año");
14        // System.out.print("fin de semana");
15        // System.out.println(s);
16        // System.out.print("¡Feliz Nuevo!");
17        // System.out.print(" ");
18        // System.out.println("Nuevo");
19    }
20 }
```

¡Siempre hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 1 - Lesson 4 / Nivel 1 – Lección 4

★ Excercise: N1_L4_E4

Cambiemos el código

Verify Help Discuss New/Reset Run Code analysis

Conditions Class tree

Solution.java

```
1 package es.codegym.task.task01.task0116;
2
3 /*
4 Cambiamos el código
5 */
6
7 public class Solution {
8     public static void main(String[] args) {
9         String nombre = "Ellie";
10        String texto = "¡Hola, " + nombre + "!";
11        System.out.println(texto);
12    }
13 }
```

Cambia el programa para que la variable **nombre** adopte el valor «Amigo».

Requirements:

- A la variable nombre se le debe asignar un valor inmediatamente después de su declaración.

¡Siempre
hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 1 - Lesson 4 / Nivel 1 – Lección 4

★ Excercise: N1_L4_E5

Algunas modificaciones más

Verify Help Discuss New/Reset Run Code analysis

Conditions Class tree

Cambia el programa para que muestre el siguiente texto:
Codificación en Java

Usa variables.

Requirements:

- Se deben declarar tres variables.
- A todas las variables se les debe asignar un valor inmediatamente después de declararlas.

Solution.java

```
1 package es.codegym.task.task01.task0117;
2
3 /*
4  * Algunas modificaciones más
5  */
6
7 public class Solution {
8     public static void main(String[] args) {
9         String s1 = "Codificación";
10        String s2 = "en";
11        String s3 = "Pascal";
12        System.out.println(s1 + " " + s2 + " " + s3);
13    }
14 }
15 }
```

...siempre
...alcancia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 1 - Lesson 4 / Nivel 1 – Lección 4

★ Excercise: N1_L4_E6

Comenta lo superfluo

Verify Help Discuss New/Reset Run Code analysis

Conditions Class tree Solution.java

Comenta algo de código para que aparezca «**suma = 12**» en la pantalla.

Requirements:

- El programa debe mostrar texto en la pantalla.
- Debes comentar las líneas necesarias y dejar sin modificar las demás.
- No añadas nuevas líneas al programa ni elimines las existentes.
- El texto mostrado debe cumplir las condiciones de la tarea.

```
1 package es.codegym.task.task01.task0118;
2
3 /*
4 Comenta lo superfluo
5 */
6
7 public class Solution {
8     public static void main(String[] args) {
9         int a = 3;
10        int a = 10;
11        int b = 6;
12        int b = 12;
13        int suma = 1 + a + b;
14        int suma = 2 + a + b;
15        int suma = 3 + a + b;
16        int suma = 4 + a + b;
17
18        System.out.println("suma = " + suma);
19    }
20 }
```

¡Siempre
hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 1 - Lesson 4 / Nivel 1 – Lección 4

★ Excercise: N1_L4_E7

¡Vuelvo a tener quince años!

Verify Help Discuss New/Reset Run Code analysis

Conditions Class tree

Cambia el programa para que la variable **edad** adopte el valor **15**.

Requirements:

- Se deben declarar cuatro variables.
- La variable edad debe ser un int.
- A todas las variables se les debe asignar un valor inmediatamente después de declararlas.
- El valor asignado a la variable edad debe ser 15.

Solution.java

```
1 package es.codegym.task.task01.task0119;
2
3 /*
4 * ¡Vuelvo a tener quince años!
5 */
6
7 public class Solution {
8     public static void main(String[] args) {
9         String nombre = "Amigo";
10        int edad = 2;
11        int dinero = 500;
12        int peso = 70;
13    }
14 }
15
```

¡Siempre
hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 1 - Lesson 5 / Nivel 1 – Lección 5



Prácticas de parte de Diego

Sintaxis de Java
Level 1, Lesson 5

AVAILABLE

«¡Hola, amigo mío! Soy yo, Diego. ¿Te acuerdas de mí? ¡Soy el que te enseñará lo que un robot necesita saber de verdad!».

«Cuanto menos caso hagas a esos sacos de carne, mejor. **¿Puede algo realmente sustituir a la práctica?** No vas a aprender a nadar sentado en una silla, ¿no? ¡Ja, ja! **La práctica lo es todo para nosotros.** ¡Somos robots, al fin y al cabo!».

¡Siempre
hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 1 - Lesson 5 / Nivel 1 – Lección 5

★ Excercise: N1_L5_E1

Bésame mi brillante retropropulsor

Verify Help Discuss New/Reset Run Code analysis

Conditions Class tree

Solution.java

```
1 package es.codegym.task.task01.task0139;
2
3 /*
4 Bésame mi brillante retropropulsor
5 */
6
7 public class Solution {
8     public static void main(String[] args) {
9         //escribe aquí tu código
10    }
11 }
```

- «Amigo, ¿quieres ser programador?».
- «¡Desde luego!».
- «Entonces escribe rápidamente un programa que muestre la frase *Hola mundo!*».
- «Pero... ¡eso está muy visto!».
- «Así es. Amigo, veo que piensas como yo, y eso es genial. Mejor vamos a hacer lo siguiente: muestra una frase que nosotros, como robots, deberíamos repetir en estos tiempos de incertidumbre. La frase es *Bésame mi brillante retropropulsor!*».



Gamified platform work / Trabajo en plataforma gamificada.

Level 1 - Lesson 6 / Nivel 1 – Lección 6



Firmar un contrato

Sintaxis de Java
Level 1, Lesson 6

AVAILABLE

«¡Soy otra vez yo! Creo que ya sabes lo suficiente para **empezar a tomar decisiones importantes**. Ya es hora de que firmes un contrato con tu nuevo empleador. Tienes que llenar una solicitud. Aquí tienes el formulario estándar. **Basta con que muestres su texto en la pantalla y ya está**. Firma sin leerlo. Es lo que siempre hago yo».

¡Siempre
hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 1 - Lesson 6 / Nivel 1 – Lección 6

★ Excercise: N1_L6_E1

Contrato

Conditions Class tree

Muestra el texto del nuevo contrato:

Me llamo Amigo.

Acepto un salario de 5000 créditos al mes el primer año.
Acepto un salario de 5500 créditos al mes el segundo año.
Acepto un salario de 7000 créditos al mes el tercer año.
Acepto un salario de 8000 créditos al mes el cuarto año.
Acepto un salario de 10 000 créditos al mes el quinto año.

¡Bésame mi brillante retropropulsor!

Solution.java

```
1 package es.codegym.task.task01.task0121;
2
3 /*
4 Contrato
5 */
6
7 public class Solution {
8     public static void main(String[] args) {
9         //escribe aquí tu código
10    }
11 }
```

¡Siempre
hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 1 - Lesson 7 / Nivel 1 – Lección 7



print() y println();

Sintaxis de Java
Level 1, Lesson 7

AVAILABLE

«Hola, soy yo otra vez. Hoy te voy a dar tres lecciones. ¡Esta es la segunda! Ponte cómodo y escucha. Te voy a hablar sobre mostrar textos en la pantalla. En realidad es muy simple».

¡Siempre
hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 1 - Lesson 7 / Nivel 1 – Lección 7

Código Java	Qué se mostrará en la pantalla
<pre>1 System.out.println("Diego"); 2 System.out.println(3); 3 System.out.println("Lluvia" + "En" + "España");</pre>	Diego 3 LluviaEnEspaña
<pre>1 System.out.println(1 + 3); 2 System.out.println("1" + "3"); 3 System.out.println(1 + "3"); 4 System.out.println("1" + 3); 5 System.out.println("1" + (1 + 3));</pre>	4 13 13 13 14
<pre>1 System.out.println("Amigo es el mejor"); 2 System.out.println("Amigo" + "es el mejor"); 3 System.out.println("Amigo" + " " + "es el mejor");</pre>	Amigo es el mejor Amigoes el mejor Amigo es el mejor
<pre>1 System.out.println(3 * 3 + 4 * 4); 2 System.out.println(1 * 2 + 3 * 4);</pre>	25 14
<pre>1 System.out.print("Diego"); 2 System.out.print("Diego"); 3 System.out.print("Diego");</pre>	DiegoDiegoDiego
<pre>1 System.out.print("Diego "); 2 System.out.println("es el mejor"); 3 System.out.print("Amigo "); 4 System.out.println("es el mejor");</pre>	Diego es el mejor Amigo es el mejor!

¡Siempre
hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 1 - Lesson 7 / Nivel 1 – Lección 7

★ Excercise: N1_L7_E1

Encontrar errores

Verify **Help** **Discuss** **New/Reset** **Run** **Code analysis**

Conditions **Class tree**

Arregla el error en el programa. El programa debería mostrar el número **100**.

Requirements:

- El programa debe mostrar texto en la pantalla.
- Se deben declarar tres variables.

Solution.java

```
1 package es.codegym.task.task01.task0122;
2
3 /*
4 Encontrar errores
5 */
6
7 public class Solution {
8     public static void main(String[] args) {
9         int a = 56;
10        int b = 30;
11        int c = 14;
12        System.out.print(a + b + "c");
13    }
14}
```



Gamified platform work / Trabajo en plataforma gamificada.

Level 1 - Lesson 7 / Nivel 1 – Lección 7

Excercise: N1_L7_E2

No hacen falta líneas de más

Conditions Class tree

Verify Help Discuss New/Reset Run Code analysis

Comenta las variables que **no se usen**.
El programa debe **compilar**.

Requirements:

- No cambies los tipos de las variables.
- No cambies los valores de las variables.
- No añadas nuevas líneas al programa ni elimines las existentes.
- Tienes que comentar las variables que no se utilicen.
- No debes cambiar la salida del programa.

Solution.java

```
1 package es.codegym.task.task01.task0123;
2
3 /*
4  * No hacen falta líneas de más
5  */
6
7 public class Solution {
8     public static void main(String[] args) {
9         int a = 10;
10        int b = 15;
11        double c = b + 38;
12        int d = a + 12;
13        double e = 12.3;
14        String s = "s" + a;
15        String s1 = a + "b";
16        String s2 = "a";
17        String s3 = s1 + "a";
18        String s4 = s3 + "b";
19        System.out.println(c + s4 + s);
20    }
21
22 }
```



Gamified platform work / Trabajo en plataforma gamificada.

Level 1 - Lesson 8 / Nivel 1 – Lección 8



Comparación con Pascal

Sintaxis de Java
Level 1, Lesson 8

AVAILABLE



¡Siempre
hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 1 - Lesson 8 / Nivel 1 – Lección 8

★ Excercise: N1_L8_E1

Aliméntate bien: ¡come fruta!

Muestra las **variables** cuyos valores son nombres de **frutas**. Muestra cada variable en una nueva línea.

Requirements:

- El programa debe generar texto.
- No cambies las líneas que declaran variables.
- El texto que se muestra debe constar de dos líneas.

Solution.java

```
1 package es.codegym.task.task01.task0135;
2
3 /*
4 Aliméntate bien: ¡come fruta!
5 */
6
7 public class Solution {
8     public static void main(String[] args) {
9         String manzana = "Manzana";
10        String gachas = "Gachas";
11        String salchicha = "Salchicha";
12        String melocotón = "Melocotón";
13
14        //escribe aquí tu código
15    }
16}
```

¡Siempre
hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 1 - Lesson 8 / Nivel 1 – Lección 8

★ Excercise: N1_L8_E2

Mi poema favorito

Verify Help Discuss New/Reset Run Code analysis

Conditions Class tree

Solution.java

En el método **main**, escribe el código para que el programa muestre **tu poema favorito**. El poema debe tener **al menos cuatro líneas**. No elimines la línea que muestra «**Mi poema favorito:**».

Requirements:

```
1 package es.codegym.task.task01.task0138;
2
3 /*
4 Mi poema favorito
5 */
6
7 public class Solution {
8     public static void main(String[] args) {
9         System.out.println("Mi poema favorito:");
10        //escribe aquí tu código
11    }
12 }
```

¡Siempre
hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 1 - Lesson 8 / Nivel 1 – Lección 8

★ Excercise: N1_L8_E3

La gran purga

Verify Help Discuss New/Reset Run Code analysis

Conditions Class tree

Declará **las variables que faltan** y comenta las que **no se utilicen** en ninguna parte.

Requirements:

- El programa debe mostrar texto en la pantalla.
- Se deben declarar cuatro variables.
- Tienes que comentar las variables que no se utilicen.

Solution.java

```
1 package es.codegym.task.task01.task0125;
2
3 /*
4 La gran purga
5 */
6
7 public class Solution {
8     public static void main(String[] args) {
9         String s = "15";
10        a = 5;
11        int z = 18;
12        int d = 18;
13        c = a + d;
14        b = " es mejor que ";
15
16    }
17    System.out.println(a + b + c);
18 }
```

Siempre
nacía lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 1 - Lesson 8 / Nivel 1 – Lección 8

★ Excercise: N1_L8_E4

¿No te apetece? Hazlo de todas formas

Verify Help Discuss New/Reset Run Code analysis

Conditions Class tree

Escribe un programa que muestre: «*Si te apetece, haz la tarea; si no, también*» dieciséis veces.

Requirements:

- El programa debe generar texto.
- El texto debe comenzar con «Si te apetece».

Solution.java

```
1 package es.codegym.task.task01.task0126;
2
3 /*
4 ¿No te apetece? Hazlo de todas formas
5 */
6
7 public class Solution {
8     public static void main(String[] args) {
9         //escribe aquí tu código
10    }
11 }
```

¡Siempre
hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 1 - Lesson 8 / Nivel 1 – Lección 8

★ Excercise: N1_L8_E5

Cuadrado de un número

Verify Help Discuss New/Reset Run Code analysis

Conditions Class tree

Escribe un programa que muestre en la pantalla **el cuadrado de 5**.

Requirements:

- El programa debe generar texto.
- El método main debe llamar al método cuadrado con 5 como argumento.
- El método main debe mostrar el resultado del método cuadrado.

Solution.java

```
1 package es.codegym.task.task01.task0127;
2
3 /*
4 Cuadrado de un número
5 */
6
7 public class Solution {
8     public static void main(String[] args) {
9         //escribe aquí tu código
10    }
11
12    public static int cuadrado(int a) {
13        return a * a;
14    }
15
16 }
```

Siempre
hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 1 - Lesson 8 / Nivel 1 – Lección 8

★ Excercise: N1_L8_E6

Tan fácil como sumar 2 + 2

Verify Help Discuss New/Reset Run Code analysis

Conditions Class tree

Escribe un programa que llame al método **sumar** con los argumentos **2** y **2**.

Requirements:

- El programa debe mostrar la suma de 2 más 2.
- El método main debe llamar al método sumar con los argumentos 2 y 2.
- El método main no debe llamar a un método de salida por pantalla.
- No cambies el método sumar.

Solution.java

```
1 package es.codegym.task.task01.task0128;
2
3 /*
4 Tan fácil como sumar 2 + 2
5 */
6
7 public class Solution {
8     public static void main(String[] args) {
9         //escribe aquí tu código
10    }
11
12    public static void sumar(int a, int b) {
13        int c = a + b;
14        System.out.print(c);
15    }
16
17 }
```

¡Siempre
hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 1 - Lesson 9 / Nivel 1 – Lección 9



El profesor Noodles al rescate

Sintaxis de Java
Level 1, Lesson 9

AVAILABLE

«Hola, Amigo, querido amigo (¡perdón por la redundancia!). Quizás hayas escuchado ya este dicho: “Cuanto más sudes en la instrucción, menos sangrarás en la guerra”».

«Bueno, pues es verdad, ¡aprender no suele ser fácil! Estás aprendiendo una nueva profesión, la de programador, y muy posiblemente suponga un cambio radical en tu trabajo».

«CodeGym implica practicar mucho. Las prácticas suponen más del 80 % de todo el proceso de aprendizaje».

«Las convertimos en una misión grande e interesante (de hecho, toda una serie de misiones), pero todos somos diferentes: hay quienes aprenden rápido mientras que otros lo hacen más despacio. Algunos dominan los nuevos materiales fácilmente mientras que para otros resulta más difícil. Sin embargo, nuestra labor es conducir a cada uno de nuestros alumnos hasta la línea de meta».

¡Siempre
hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 1 - Lesson 10 / Nivel 1 – Lección 10



Descanso de cinco minutos.

Sintaxis de Java
Level 1, Lesson 10

AVAILABLE

«Hola, Amigo. Diego y yo hemos estado contando chistes en nuestro descanso. ¿Quieres acompañarnos?».

«Sí, claro».

Dos alumnos, jugadores compulsivos de su primer y quinto año, apostaron a ver si podían jugar mientras estudiaban. Después de discutirlo mucho, decidieron preguntarle al decano. Seguro que él lo sabría.

«¿Puedo usar el ordenador para jugar mientras preparo mis exámenes?», preguntó el estudiante de primero.

«¡Qué barbaridad! ¡No puedes jugar mientras estudias», le regañó el decano indignado.

«¿Puedo estudiar usando el ordenador?», preguntó el alumno de quinto.

«¡Pues claro! ¡Puedes estudiar siempre!», respondió el decano, alabándolo.



Gamified platform work / Trabajo en plataforma gamificada.

Level 1 - Lesson 11 / Nivel 1 – Lección 11



Descanso de cinco minutos. Otra vez.

Sintaxis de Java
Level 1, Lesson 11

AVAILABLE

«Me toca. Vale, escucha».

Unos científicos crearon un superordenador. Era tan potente que se creía que podría responder a cualquier pregunta, de modo que le plantearon su duda:

«¿Dios existe?».

El ordenador empezó a pensar. Tras algunos zumbidos y parpadeos de luces, dijo:

«No dispongo de suficiente información. Conectadme a los ordenadores más potentes del mundo».

Los científicos se rascaron la cabeza y decidieron seguirle la corriente. Volvieron a preguntarle:

«¿Dios existe?».

Más zumbidos y luces parpadeantes y el ordenador responde:

«No dispongo de suficiente información. Conectadme a todos los ordenadores de la Tierra».

Esto fue un poco más difícil de hacer, pero los científicos lograron conectarlo a todos los ordenadores. Una vez más, le hicieron su pregunta. Y de nuevo el ordenador les dijo:

«No dispongo de suficiente información. Conectadme a todas las redes, a todos los dispositivos informáticos, etc.».

Los científicos hicieron todo lo posible para conceder sus deseos. Finalmente, le preguntaron una vez más:

«¿Dios existe?».

Y, por fin, obtuvieron la respuesta:

«Ahora sí».

¡Siempre
hacia lo alto!



Gamified platform work / Trabajo en plataforma gamificada.

Level 1 - Lesson 12 / Nivel 1 – Lección 12



Descanso para cerebritos

Sintaxis de Java
Level 1, Lesson 12

AVAILABLE

«¡Hola, Amigo! Parece que ya has trabajado suficiente por hoy. ¿Qué te parece si te relajas un rato?».





Feedback

LET'S
GO!



¡Siempre
hacia lo alto!



Survey



¡Siempre
hacia lo alto!



Clarification of doubts



¡Siempre
hacia lo alto!



To practice!

LET'S
GO!



¡Siempre
hacia lo alto!





UNIVERSIDAD SANTO TOMÁS
PRIMER CLAUSTRO UNIVERSITARIO DE COLOMBIA
SECCIONAL TUNJA

VIGILADA MINEDUCACIÓN - SNIES 1732

¡Siempre
hacia lo alto!

USTATUNJA.EDU.CO

