



ex  LATAM  
CODING  
DOJO

# Introducción al Curso

**Cynthia Castillo**

Líder de Instrucción





[ccastillo@skillnest.com](mailto:ccastillo@skillnest.com)

Desarrollador  
**Full Stack**  
**Python**

---



# ¿Qué vamos a Aprender?

-  Perfil y contexto laboral del desarrollador Full-Stack Python
-  Estructura del plan formativo y productos obtenidos
-  Metodología de trabajo: colaboración y ética profesional
-  Uso de herramientas de Inteligencia Artificial para aumentar la productividad



# Módulos



NÚMERO DE MÓDULOS	NOMBRE DEL MÓDULO
Módulo N°1	ORIENTACIÓN AL PERFIL Y METODOLOGÍA DEL CURSO
Módulo N°2	FUNDAMENTOS DE DESARROLLO FRONT-END
Módulo N°3	FUNDAMENTOS DE PROGRAMACIÓN EN PYTHON
Módulo N°4	PROGRAMACIÓN AVANZADA EN PYTHON
Módulo N°5	FUNDAMENTOS DE BASES DE DATOS RELACIONALES
Módulo N°6	DESARROLLO DE APLICACIONES WEB CON PYTHON DJANGO
Módulo N°7	ACCESO A DATOS EN APLICACIONES PYTHON DJANGO
Módulo N°8	DESARROLLO DE PORTAFOLIO DE UN PRODUCTO DIGITAL
Módulo N°9	DESARROLLO DE EMPLEABILIDAD EN LA INDUSTRIA DIGITAL

# Glosario

## Primer reto...

---



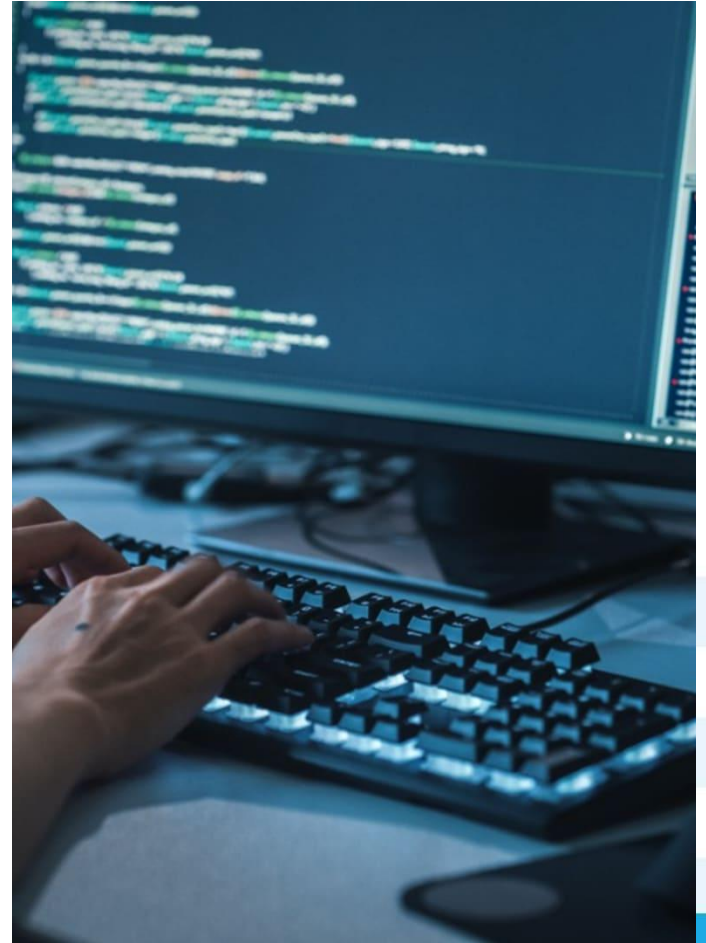




# **Perfiles Laborales y Competencia**

# Principales **Perfiles Laborales**

- Desarrollador Backend
- Desarrollador Frontend
- Desarrollador Full-Stack
- Ingeniero de Datos:
- DevOps:
- QA Tester



# Competencias técnicas del perfil



## Backend con Python/Django

- Lenguaje Python
- Django
- Bases de datos relacionales
- APIs y RESTful
- Autenticación y seguridad
- Pruebas
- Despliegue
- Versionado con Git



## Frontend

- HTML y CSS
- JavaScript
- Frameworks y librerías
- Consumo de APIs
- State Management
- Testing en frontend





# Habilidades personales **del perfil laboral**

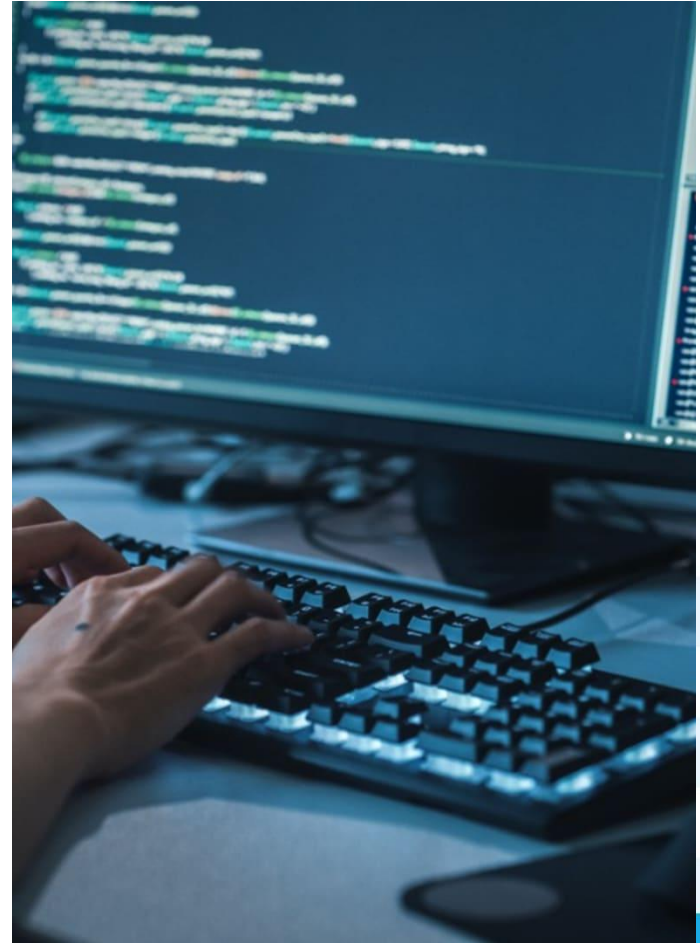
- Resolución de problemas
- Trabajo en equipo
- Adaptabilidad
- Gestión del tiempo
- Comunicación efectiva

# Niveles de experiencia y seniority en el desarrollo de software

Nivel	Años de experiencia	Características
Junior	0 - 2 años	Necesita supervisión, realiza tareas básicas, aprende buenas prácticas.
Mid-Level	2 - 5 años	Maneja proyectos con autonomía, optimiza código, propone mejoras.
Senior	5+ años	Lidera proyectos, mentoriza a otros, toma decisiones técnicas.

# Expectativas laborales **del mercado actual**

- Experiencia con Django Rest Framework para el desarrollo de APIs.
- Conocimientos en bases de datos
- Uso de React o Vue
- Despliegue de aplicaciones
- Implementación de pruebas en backend y frontend.
- Trabajo en equipo bajo metodologías ágiles





# Proyección laboral en **Python/Django y frontend**

- Desarrollo Web con Django y React.
- Ciencia de Datos y Machine Learning con Python.
- Automatización de procesos con scripts en Python.
- Aplicaciones móviles utilizando React Native.

# ¿Qué vamos a Aprender?

- 📌 Perfil y contexto laboral del desarrollador Full-Stack Python
- 🧭 Estructura del plan formativo y productos obtenidos
- 🤝 Metodología de trabajo: colaboración y ética profesional
- 🤖 Uso de herramientas de Inteligencia Artificial para aumentar la productividad

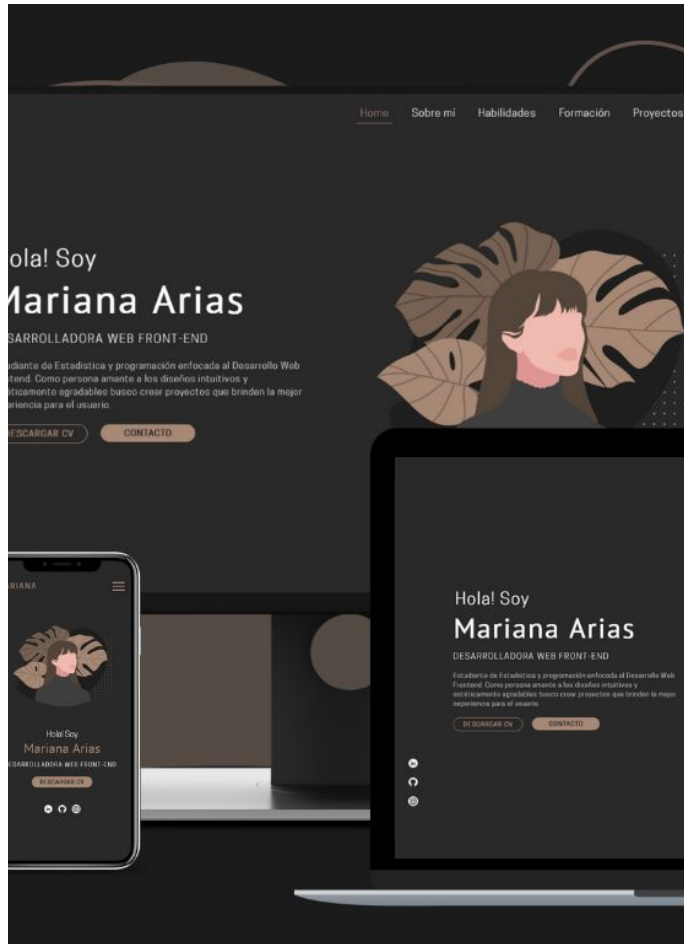


# Módulos



NÚMERO DE MÓDULOS	NOMBRE DEL MÓDULO
Módulo N°1	ORIENTACIÓN AL PERFIL Y METODOLOGÍA DEL CURSO
Módulo N°2	FUNDAMENTOS DE DESARROLLO FRONT-END
Módulo N°3	FUNDAMENTOS DE PROGRAMACIÓN EN PYTHON
Módulo N°4	PROGRAMACIÓN AVANZADA EN PYTHON
Módulo N°5	FUNDAMENTOS DE BASES DE DATOS RELACIONALES
Módulo N°6	DESARROLLO DE APLICACIONES WEB CON PYTHON DJANGO
Módulo N°7	ACCESO A DATOS EN APLICACIONES PYTHON DJANGO
Módulo N°8	DESARROLLO DE PORTAFOLIO DE UN PRODUCTO DIGITAL
Módulo N°9	DESARROLLO DE EMPLEABILIDAD EN LA INDUSTRIA DIGITAL





# Características de un **portafolio de productos**

- Variedad de proyectos
- Código limpio y bien documentado
- Proyectos completos
- Visibilidad en línea

# Módulo 1: Orientación al perfil y metodología del curso



## Competencias

Introducción al curso, comprensión del perfil de egreso y metodología de trabajo.



## Herramientas

NA



## Trabajo Técnico

Orientación sobre el curso, perfil de egreso y la metodología de trabajo, incluyendo actividades de integración y planificación.



## Producto Obtenido

NA - Actividades integración y planificación

# Módulo 2: Fundamentos de desarrollo front-end



## Competencias

Diseño y desarrollo de interfaces de usuario, conocimiento básico de HTML, CSS y JavaScript.



## Herramientas

HTML, CSS, JavaScript, VSCode, Chrome Developer Tools.



## Trabajo Técnico

Crear páginas web estáticas utilizando HTML, CSS y JavaScript. Crear diseños responsivos y a manipular el DOM.



## Producto Obtenido

Páginas web estáticas y dinámicas utilizando HTML, CSS y JavaScript.

# Módulo 3: Fundamentos de programación en Python



## Competencias

Desarrollo de aplicaciones en Python, comprensión de estructuras de datos y control de flujo.



## Herramientas

Python, VSCode, Git.



## Trabajo Técnico

Proyectos de programación básicos en Python, como calculadoras, juegos sencillos y otros programas con estructuras de datos.



## Producto Obtenido

Programas y ejercicios básicos en Python, como juegos y aplicaciones simples.

# Módulo 4: Programación avanzada en Python



## Competencias

Dominio de conceptos avanzados de programación en Python, manejo de módulos, funciones y manejo de excepciones.



## Herramientas

Python, VSCode, Git.



## Trabajo Técnico

Aplicar conceptos avanzados de Python, creando proyectos más complejos que involucran manejo de excepciones y procesamiento de archivos.



## Producto Obtenido

Proyectos más complejos en Python, incluyendo manejo de archivos y excepciones.

# Módulo 5: Fundamentos de bases de datos relacionales



## Competencias

Creación y gestión de bases de datos relacionales, uso de SQL para consultas.



## Herramientas

MySQL, SQL.



## Trabajo Técnico

Creación y manipulación de bases de datos relacionales, así como el desarrollo de consultas SQL para gestionar datos.



## Producto Obtenido

Bases de datos relacionales con consultas SQL, integradas en aplicaciones pequeñas.



# Módulo 6: Desarrollo de aplicaciones web con Python Django



## Competencias

Desarrollo de aplicaciones web con Django, gestión de vistas, modelos y plantillas.



## Herramientas

Django, MySQL, Git, Postman.



## Trabajo Técnico

Desarrollar aplicaciones web utilizando Django, creando modelos, vistas y plantillas. Implementar funcionalidades básicas de un sistema web.



## Producto Obtenido

Aplicaciones web completas en Django, con funcionalidad de bases de datos y plantillas

# Módulo 7: Acceso a datos en aplicaciones Python Django



## Competencias

Integración de bases de datos en aplicaciones Django, implementación de autenticación y autorización de usuarios.



## Herramientas

Django ORM, MySQL, Django Rest Framework.



## Trabajo Técnico

Integración de bases de datos en aplicaciones Django, implementando operaciones CRUD y manejando autenticación de usuarios.



## Producto Obtenido

Aplicaciones Django con integración de bases de datos y autenticación de usuarios.

# Módulo 8: Desarrollo de portafolio de un producto digital



## Competencias

Creación de un portafolio profesional con proyectos completos que demuestren habilidades adquiridas durante el curso.



## Herramientas

GitHub, VSCode.



## Trabajo Técnico

Desarrollar un portafolio con proyectos completos que incluyan aplicaciones web frontend y backend, reflejando el conocimiento adquirido a lo largo del curso.



## Producto Obtenido

PORTAFOLIO COMPLETO!

# Módulo 9: Desarrollo de empleabilidad en la industria digital



## Competencias

Preparación para la inserción laboral, desarrollo de habilidades blandas, elaboración de CV y preparación para entrevistas.



## Herramientas

LinkedIn, GitHub, plataformas de empleo.



## Trabajo Técnico

Desarrollar su perfil profesional mediante la creación de CV, entrevistas simuladas y actividades de networking.

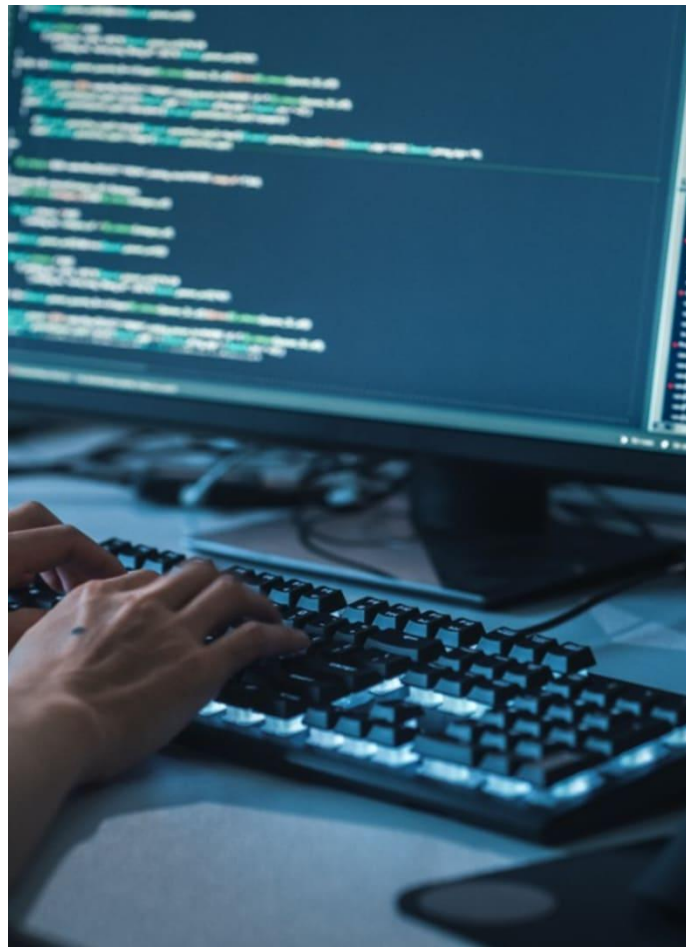


## Producto Obtenido

Listo para trabajar ;)

# Herramientas a utilizar

- Entornos de desarrollo (IDE): Visual Studio Code, PyCharm.
- Lenguajes y frameworks: Python, Django, React, JavaScript.
- Bases de datos: MySQL Workbench
- Control de versiones: Git y GitHub.
- Pruebas de APIs: Postman, Insomnia.
- Despliegue: Docker, AWS, Heroku.
- Colaboración y productividad: Slack, Trello, Notion.

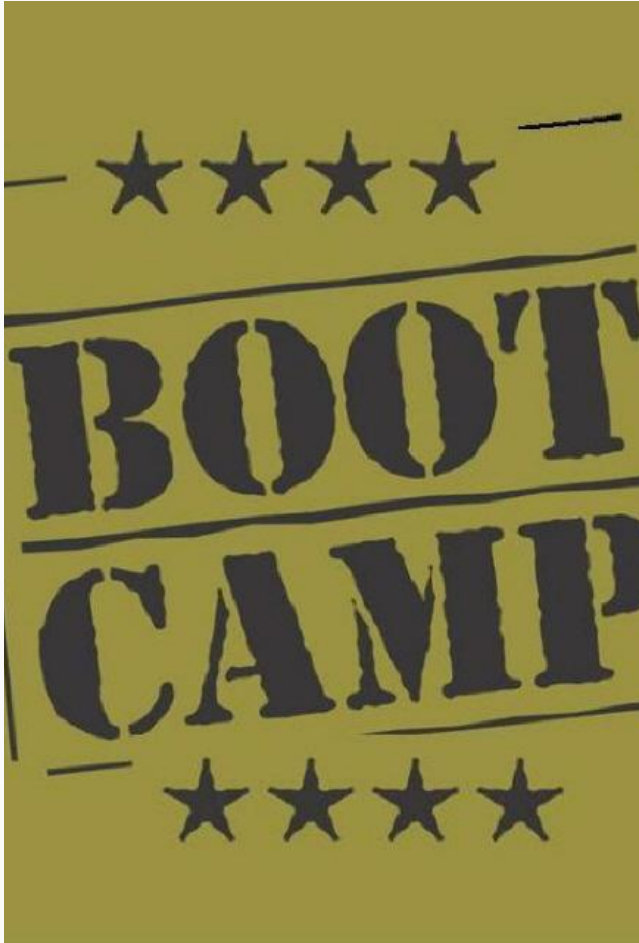


# ¿Qué vamos a Aprender?






- 📌 Perfil y contexto laboral del desarrollador Full-Stack Python
- 🧭 Estructura del plan formativo y productos obtenidos
- 🤝 Metodología de trabajo: colaboración y ética profesional
- 🤖 Uso de herramientas de Inteligencia Artificial para aumentar la productividad







## ¿Qué es un **Bootcamp**?

-  Aprendizaje basado en proyectos
-  Inmersión intensiva
-  Método colaborativo
-  Retroalimentación constante
-  Enfoque práctico



## El Origen del **Bootcamp**

# El día a día del **Bootcamp**

- 🏋️ Sesiones intensivas
- 👥 Trabajo en equipo
- 🕒 Resolución de problemas en tiempo real
- 📅 Revisión de proyectos
- 💌 Mentoría y feedback



# Miedos y Estrategias para vencerlos!



## Algunos miedos...

- Al fracaso
- Inseguridad técnica
- Ritmo acelerado
- Competencia



## Estrategias...

- Mentalidad de crecimiento
- Organización y gestión del tiempo
- Buscar apoyo
- Hábitos saludables
- Perseverancia
- Práctica constante



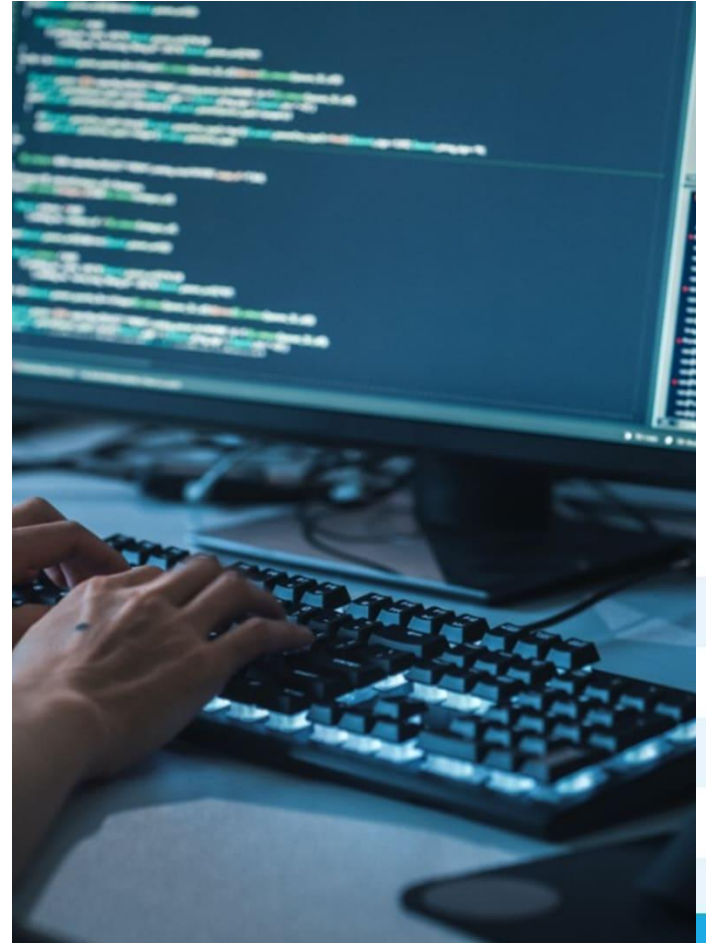
## Recomendaciones

- Compromiso con el aprendizaje
- Disciplina
- Perseverancia
- Balance vida-trabajo



# Exigencias en los Bootcamps

- Carga horaria intensa
- Trabajo constante
- Disciplina
- Perseverancia frente a los obstáculos



# Herramientas Importantes



## Gestión del proceso de aprendizaje

LMS:

- Material
- Progreso
- Evaluación
- Foros
- Tareas



## Coordinación y trabajo colaborativo

- Discord
- Trello
- Google Workspace
- GitHub



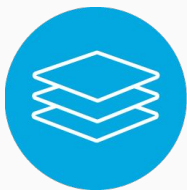
## Competencia Técnica

- IDE
- Bases de Datos
- Control de Versiones
- Frameworks y Bibliotecas
- Herramientas testing



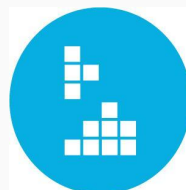


# Trabajo en Equipo



## Buenas Prácticas

- Comunicación efectiva
- Asignación de roles claros
- Trabajo colaborativo
- Revisión de código
- Resolución de conflictos



## Organización

- Definición de objetivos claros
- Estructura jerárquica o distribuida
- Planificación y programación de tareas
- Reuniones periódicas
- Uso de herramientas de colaboración

# ¿Qué es el **Autoaprendizaje?**

---





# Tolerancia a la **Frustración**

- Respirar profundamente
- Tomar un descanso
- Descomponer el problema
- Hablar con otros

# Código Ético

- Respeto mutuo
- Honestidad
- Colaboración
- Compromiso con el aprendizaje



# ¿Qué vamos a Aprender?

- 📌 Perfil y contexto laboral del desarrollador Full-Stack Python
- 🧭 Estructura del plan formativo y productos obtenidos
- 🤝 Metodología de trabajo: colaboración y ética profesional
- 🤖 Algoritmos y herramientas de Inteligencia Artificial para aumentar la productividad



¿Qué es un  
**Algoritmo?**

---

**DesEntradaación**

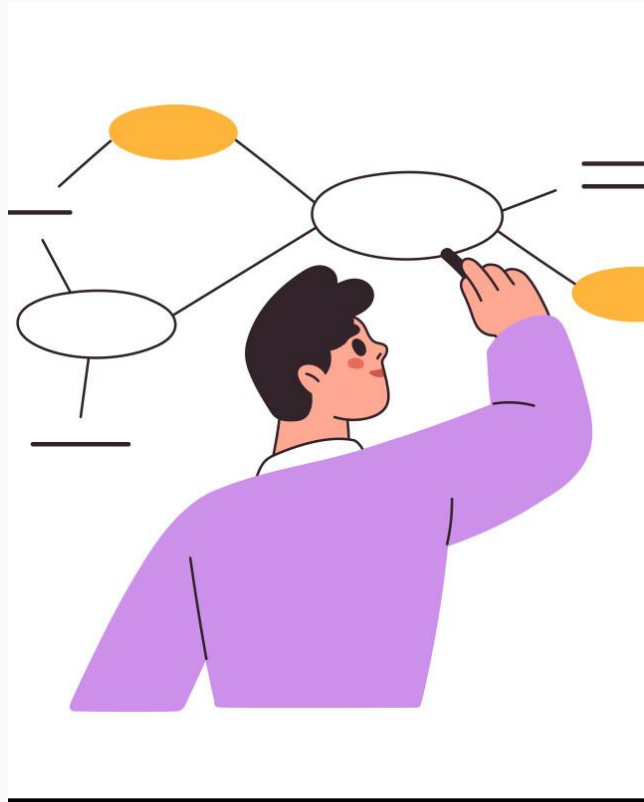
**Proceso**

**Abstracción**

**Salida**

**Condiciones**





```

si (edad > 18) {
    imprime: "Mamermde! Édad"
}
sinomientras
    imprime: "Pequeñuelo"

```

# Elementos Clave Algoritmos

- Variables: numéricas, texto, booleanas  
`edad = 19`  
`nombre = "Juana de Arco"`  
`soleado = true`
- Sentencias de Control:  
 Condicionales y Bucles

# Pseudocódigo

Manera de escribir los pasos de un algoritmo usando palabras simples, como si se lo estuvieras explicando a alguien, pero siguiendo una estructura parecida a la de un lenguaje de programación.

Normalmente comienza con **Inicio**, después tendríamos las instrucciones o pasos a seguir y terminaríamos con **Fin**.

Ejemplo: Pseudocódigo para sumar dos números

**Inicio**

Leer número1

Leer número2

suma = número1 + número2

Escribir suma

**Fin**

## RETO EN VIVO

Escribe un pseudocódigo que lea y calcule el promedio de 3 calificaciones

**Inicio**

Leer cal1

Leer cal2

Leer cal3

prom = (cal1+cal2+cal3) / 3

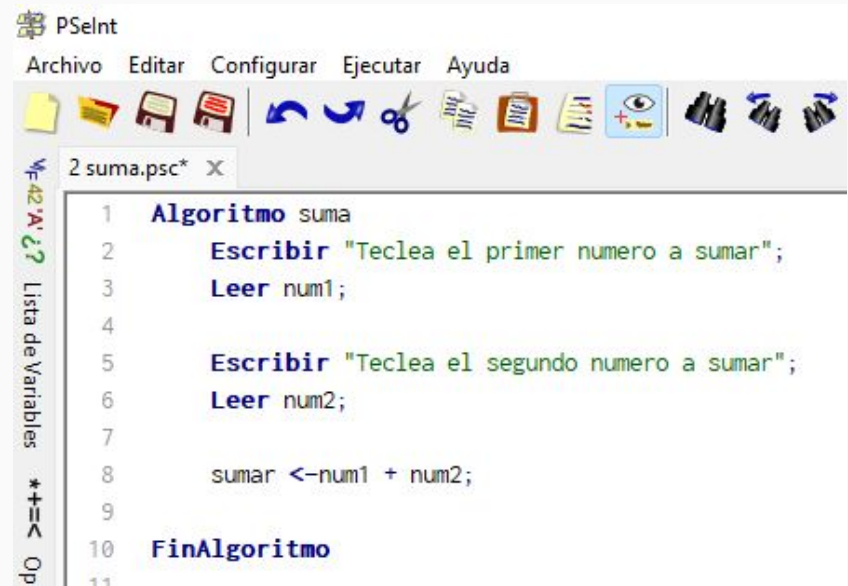
Escribir prom

**Fin**

y si quisiéramos saber si aprobó?



# PSelnt



The screenshot shows the PSelnt application window. The title bar reads "PSelnt". The menu bar includes "Archivo", "Editar", "Configurar", "Ejecutar", and "Ayuda". The toolbar contains icons for file operations (new, open, save, save as, print, delete) and editing (undo, redo, cut, copy, paste, find, replace, zoom in, zoom out, zoom reset). The main window displays a file named "2 suma.psc\*" with the following Pascal code:

```
1  Algoritmo suma
2      Escribir "Teclea el primer numero a sumar";
3      Leer num1;
4
5      Escribir "Teclea el segundo numero a sumar";
6      Leer num2;
7
8      sumar <- num1 + num2;
9
10 FinAlgoritmo
11
```

On the left side of the code editor, there is a vertical toolbar with icons for "Lista de Variables" (List of Variables), "++" (insert), "=", "<" (delete), and "Op" (operators).

# Ejecución del Algoritmo



Inicio

Leer número1

Leer número2

suma = número1 + número2

Escribir suma

Fin

var	valor
numero1	6
numero2	8
suma	14

Inicio

Leer numero

Si numero \* 2 > 10






numero = numero / 2

SiNo

numero = numero \* 2

Fin

# Diagramas de Flujo

Símbolo	Nombre	Función
	Inicio / Final	Representa el inicio y el final de un proceso
	Línea de Flujo	Indica el orden de la ejecución de las operaciones. La flecha indica la siguiente instrucción.
	Entrada / Salida	Representa la lectura de datos en la entrada y la impresión de datos en la salida
	Proceso	Representa cualquier tipo de operación
	Decisión	Nos permite analizar una situación, con base en los valores verdadero y falso

Inicio

Leer calificación

Si calificación > 6

sumar Escribir "Aprobado"

Si No Escribir suma

Escribir "Reprobado"

Fin

Fin

# Reto en Equipos



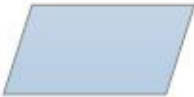


Símbolo	Nombre	Función
	Inicio / Final	Representa el inicio y el final de un proceso
	Línea de Flujo	Indica el orden de la ejecución de las operaciones. La flecha indica la siguiente instrucción.
	Entrada / Salida	Representa la lectura de datos en la entrada y la impresión de datos en la salida
	Proceso	Representa cualquier tipo de operación
	Decisión	Nos permite analizar una situación, con base en los valores verdadero y falso

Diagrama de Flujo para:

- Prepararse para dormir
- Entrar a un Club Nocturno
- Lavar los platos



**GitHub**  
Copilot



**ChatGPT**

# Inteligencia Artificial



CHAT**GPT**

---

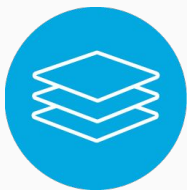


# Buenas prácticas para generación de código

- Especificar el lenguaje de programación o el sistema de bases de datos.
- Indicar claramente el propósito del código.
- Proporcionar detalles adicionales como restricciones o condiciones.
- Solicitar ejemplos de uso cuando sea necesario.



# IA en la Programación

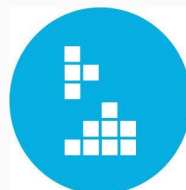


## Riesgos

Código incorrecto o inseguro

Falta de contexto

Dependencia excesiva



## Buenas Prácticas

Revisar siempre el código generado.

Complementar con documentación oficial.

Utilizar IA como apoyo, no como  
reemplazo del aprendizaje.

Verificar la seguridad del código antes de  
implementarlo.





DESBLOQUEA TU POTENCIAL

