



Agencia de  
Aprendizaje  
a lo largo  
de la vida

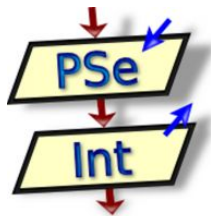
# Codo a Codo inicial

## Clase 4

# Les damos la bienvenida

Vamos a comenzar a grabar la clase

# Herramientas que utilizamos en clases



**PSeInt** es una herramienta para asistir a un estudiante en sus primeros pasos en programación



**Whimsical** es el espacio de trabajo visual y colaborativo que ayuda a tu equipo a comunicarse a la velocidad del pensamiento ya que ofrece esquemas de base, diagramas de flujo, mapas mentales, notas adhesivas y documentos (beta) que son increíblemente rápidos.

## Clase 03

**Programación - Conceptos básicos**

- Datos.
- Algoritmos.  
Características, diseño, verificación.
- Diagrama de flujo
- Desafíos.

## Clase 04

**Análisis y resolución de problemas**

- **Programa Informático. Concepto.**
- **Estructura de un programa.**
- **Metodologías para la resolución de problemas.**
- **Instalación de PSeint.**
- **Desafíos.**

## Clase 05

**Instalacion de Software : Sintaxis y variables**

- Variables y ejemplos.
- Sintaxis.
- Ejemplos.

# Pensamiento Computacional

# ¿Qué es el pensamiento computacional?

- Es un proceso que **permite abordar un problema complejo, comprenderlo y desarrollar las soluciones** más adecuadas.
- El pensamiento computacional tiene **4 etapas**, que las veremos a continuación.



# Etapas del pensamiento computacional

- **La descomposición:** se trata de desarticular el problema complejo en unidades más chicas que resulten más fáciles de resolver.
- **Reconocimiento de patrones:** consiste en buscar similitudes o serie que se repiten dentro o fuera de la situación problemática
- **La abstracción:** es simplificar un problema complejo, dejando de lado los detalles pocos relevantes para centrar la atención en los datos importantes y definir un plan acotado.
- **El algoritmo:** es la definición de los pasos ordenados necesarios para la resolución del problema.



# Habilidades del pensamiento computacional

El pensamiento computacional permite:

- Ampliar habilidades de **comunicación**.
- Aprender a **trabajar con otros** para lograr un objetivo común.
- Lidiar con **problemas complejos** y situaciones diversas.
- Poner en juego la **creatividad**.
- Desarrollar el **pensamiento crítico** y el **razonamiento lógico**.





# EL PROBLEMA

# Resolver Problema

Como mencionamos anteriormente los algoritmos resuelven problemas, pero ¿Cómo definimos a estos problemas? ¿Existenciales, pasionales, naturales?

## ¿Qué tipo de problemas se pueden resolver?

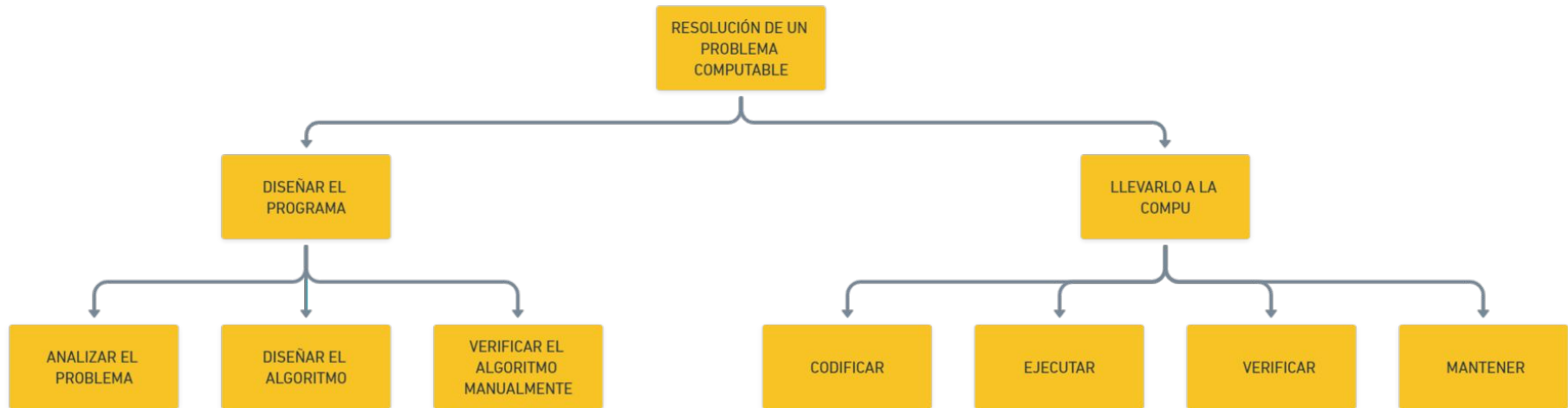
- Computables

## ¿Qué métodos hay para resolver problemas computables?

- Metodología de la programación (centrado en los algoritmos)



# Fases de resolución de problemas computables



# Fases para resolver un problema computable

Las fases para resolver un problema computable son

1. **Diseñar el programa.**
2. **Llevarlo a la computadora.**



# 1. Diseñar el programa

El diseño de programa podemos resumirlo en los **siguientes pasos**

## 1.1 Análisis del problema

- Es el análisis propiamente dicho

## 1.2 Diseño del algoritmo

- Es el diseño, lo que aprendimos en la clase pasada, el refinamiento paso a paso

## 1.3 Verificación manual del algoritmo



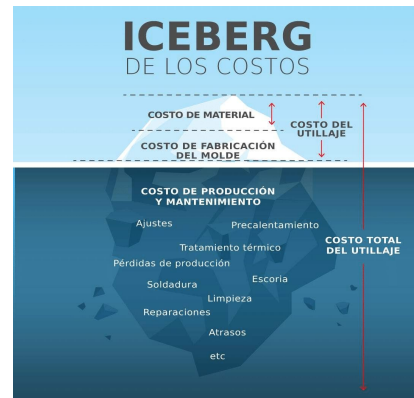
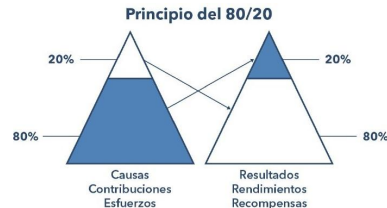
# 1.1 Análisis del problema

- El **Análisis del problema**, consiste en:
  - a. **Definir y entender el problema.**
  - b. **Identificar los datos de entrada.**
  - c. **Identificar los datos de salida** es decir qué voy a obtener como **resultado.**



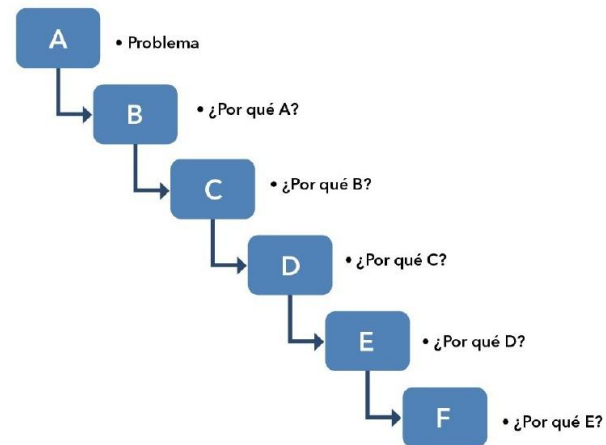
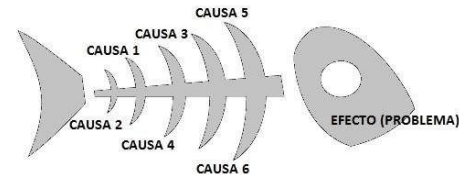
# 1.1 Análisis del problema

- El **primer paso** a seguir **para encontrar la solución** a un problema computable es el **análisis del problema**.
- En el análisis del problema se **requiere** del máximo de **creatividad e imaginación**.
- Debido a que se **busca una solución** se debe **examinar cuidadosamente el problema** a fin de identificar que tipo de **información es necesaria producir**. Enseguida se deben identificar aquellos **elementos de información ofrecidos por el problema** y que resulten útiles para **obtener la solución al problema**.
- **Finalmente**, elaborar un procedimiento para producir los resultados deseados a partir de los datos, es decir, **el algoritmo**.



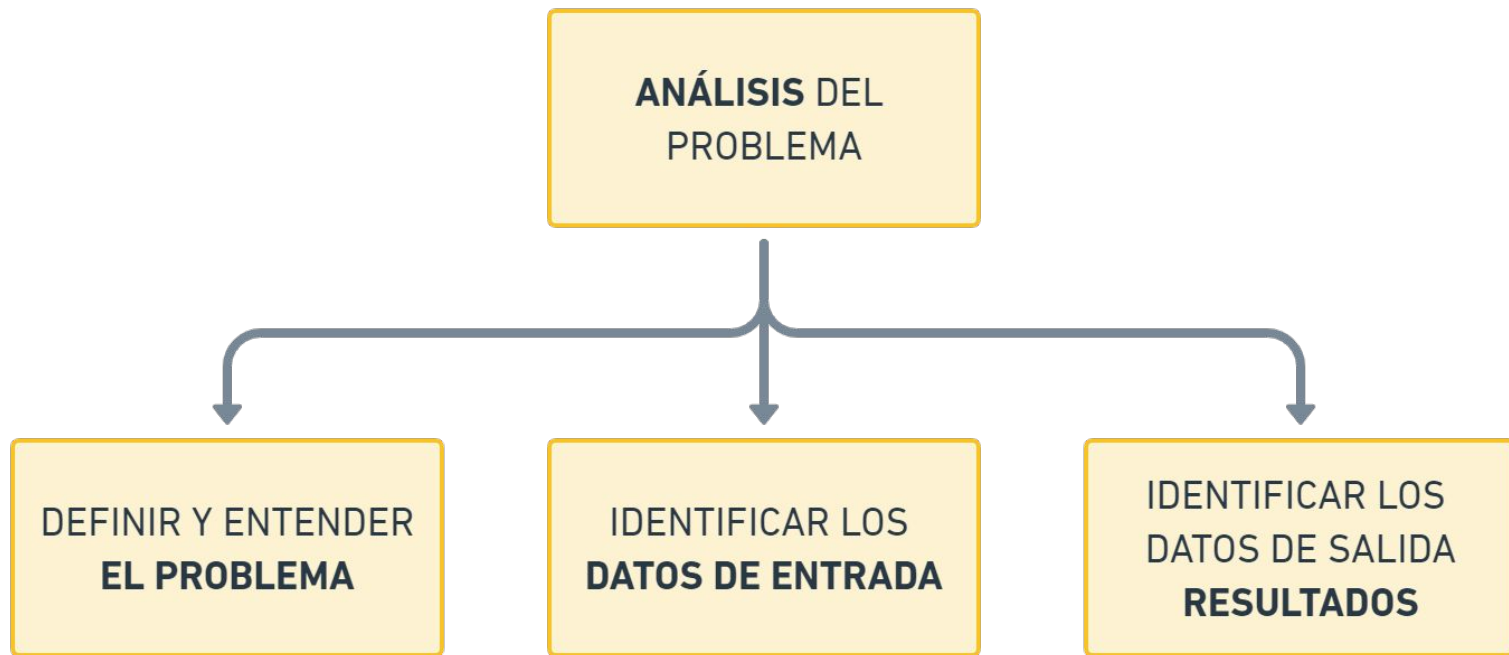
# 1.1 Análisis del problema

- Para identificar un problema existen variados métodos y son objetos de especialidades en resolución de problemas, por ejemplo:
  - Las entrevistas
  - Diagrama de Ishikawa
  - El método de los 5 por qué





# Resumido en un esquema conceptual



## 2. Llevarlo a la computadora

Llevar el diseño anterior a la **compu** consiste simplemente en:

- **Codificación** del algoritmo
- **Ejecución** del programa
- **Verificación** del programa (pruebas)
- **Mantenimiento** y documentación.



# Un ejemplo de análisis

Miriam va a estudiar programación y para ello decide comprar dos libros, uno de algoritmos y otro de Python, quiere saber el monto total con la promoción que le llegó por correo ofreciéndole un descuento del 20% para la compra de ambos libros.

La fórmula a aplicar es:

$$\text{costo} = ((\text{primerLibro} + \text{segundoLibro}) * 0.8)$$

# Entendiendo el problema

Podemos deducir que **este problema busca una solución simple** recurriendo a un poco de matemáticas, pero para ello se deben identificar los precios de ambos libros.

# Identificando las entradas y salidas

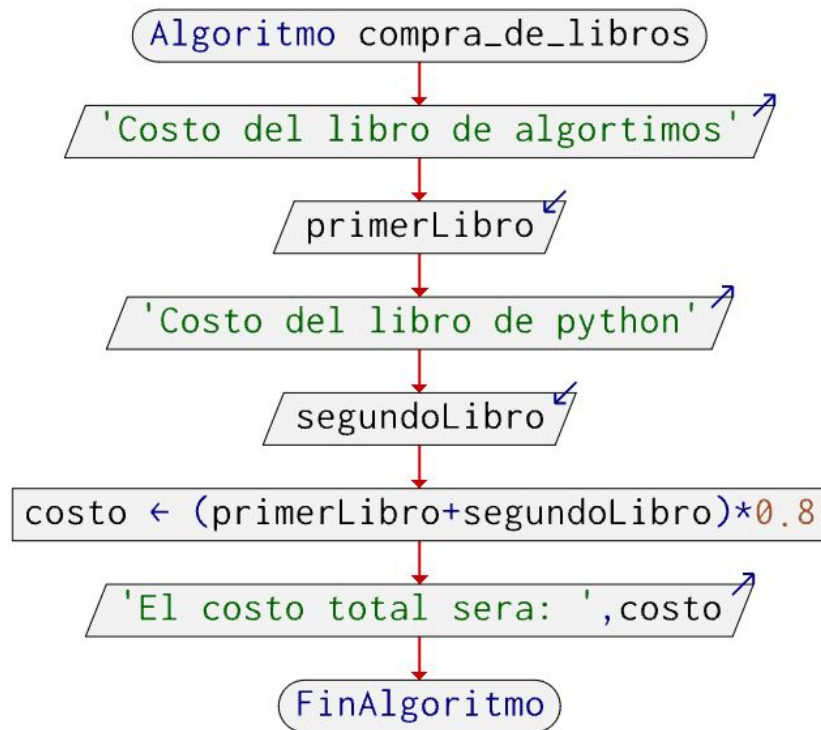
## Entradas

- Costo del libro de algoritmos.
- Costo del libro de Python.

## Salidas

- Monto total a pagar.

# Diseñar el algoritmo - DF

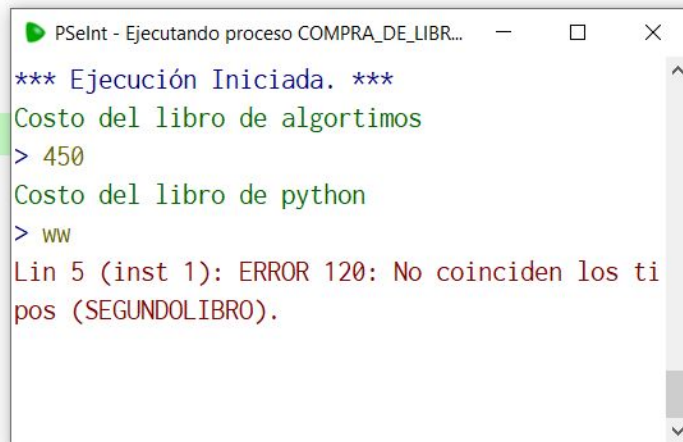


# Codificar

```
1 Algoritmo compra_de_libros
2     Escribir "Costo del libro de algortimos"
3     Leer primerLibro
4     Escribir "Costo del libro de python"
5     Leer segundoLibro
6     costo = (primerLibro + segundoLibro) * 0.8
7     Escribir "El costo total sera: ", costo
8 FinAlgoritmo
```

# Depurar

```
1 Algoritmo compra_de_libros
2   Escribir "Costo del libro de algortimos"
3   Leer primerLibro
4   Escribir "Costo del libro de python"
5   Leer segundoLibro
6   costo = (primerLibro + segundoLibro) * 0.8
7   Escribir "El costo total sera: ", costo
8 FinAlgoritmo
```



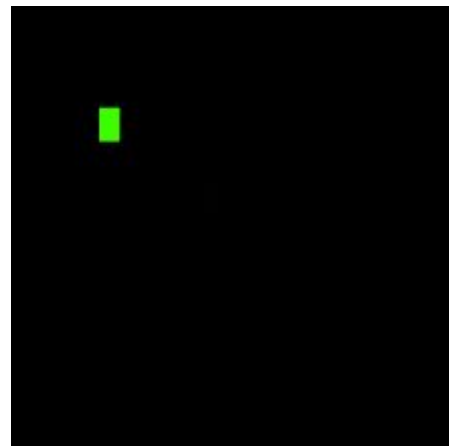
```
PSeInt - Ejecutando proceso COMPRA_DE_LIBR...
*** Ejecución Iniciada. ***
Costo del libro de algortimos
> 450
Costo del libro de python
> ww
Lin 5 (inst 1): ERROR 120: No coinciden los ti
pos (SEGUNDOLIBRO).
```



# Programa Informático

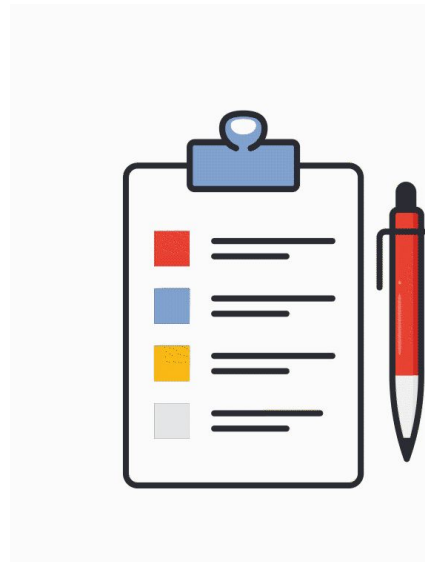
# Programa informático - *review de la 2da clase*

Un programa informático es una secuencia de acciones (**instrucciones**) que manipulan un conjunto de objetos (**datos**).



# ¿Cómo se compone un programa?

- **Cabecera:** A modo de comentarios se suele especificar:
  - Nombre del programa
  - Fecha de creación
  - Autoría y versionado
- **Funciones:** Definición de funciones propias creadas por el programador para usarlas en varias ocasiones.
- **Declaraciones:** Definiciones y declaraciones de tipo
  - variables
  - constantes
  - nuevos tipos de datos



# ¿Cómo se compone un programa?

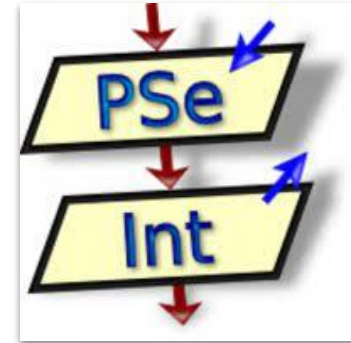
- **Asignaciones:** Valores iniciales de los identificadores declarados previamente.
- **Entradas:** Instrucciones para almacenar en memoria los valores de algunos identificadores.
- **Control:** Instrucciones de control de flujo del programa. Pueden ser:
  - Alternativas
  - Repetitivas
- **Salidas:** Instrucciones para devolver los resultados obtenidos.



# Descargamos PseInt

# PSeInt

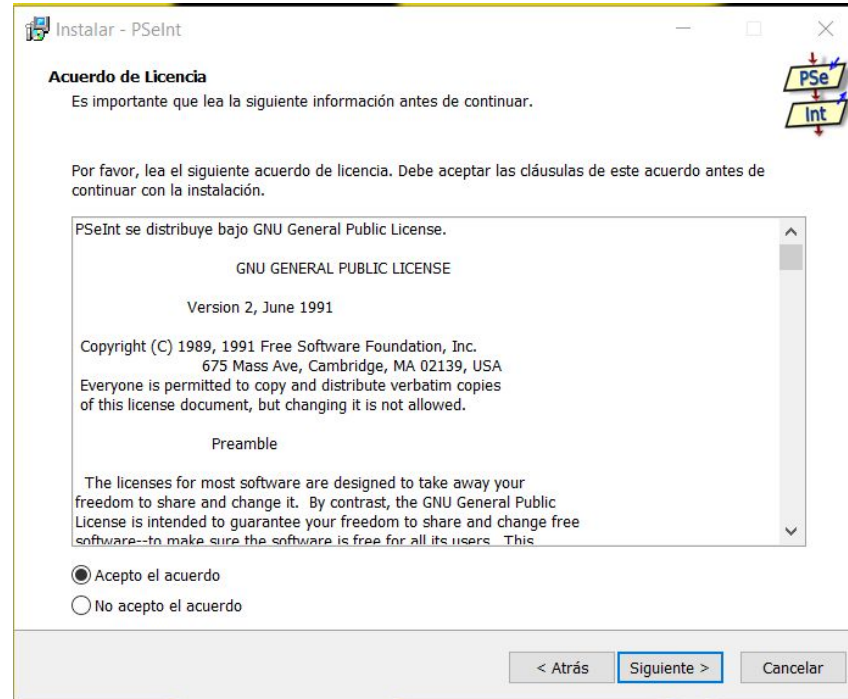
Herramienta que nos permite dar los primeros pasos en programación, mediante pseudocódigo y complementado con diagrama de flujo.



# Instalación

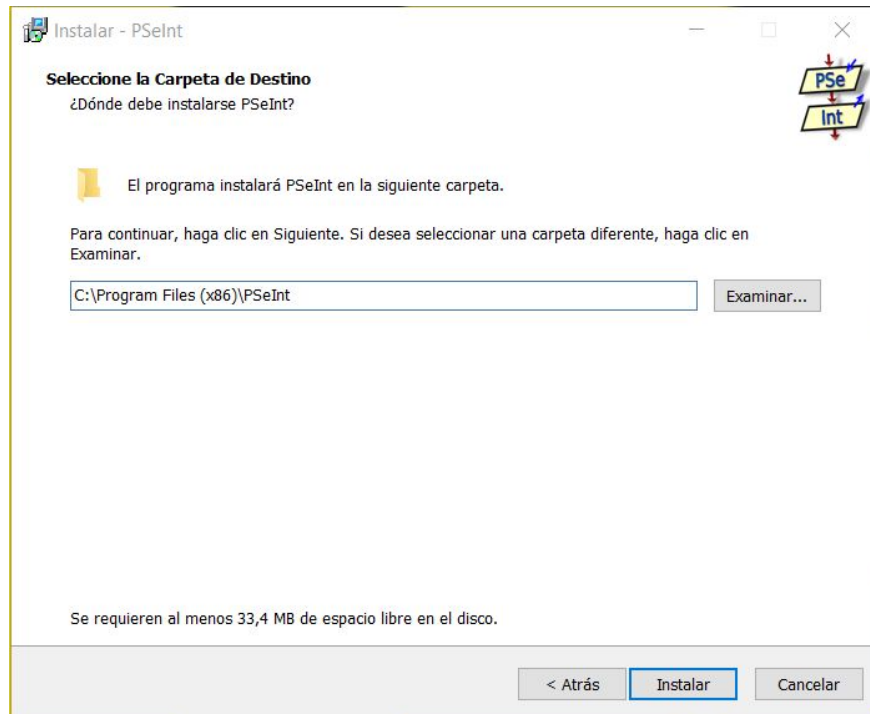


# Instalación

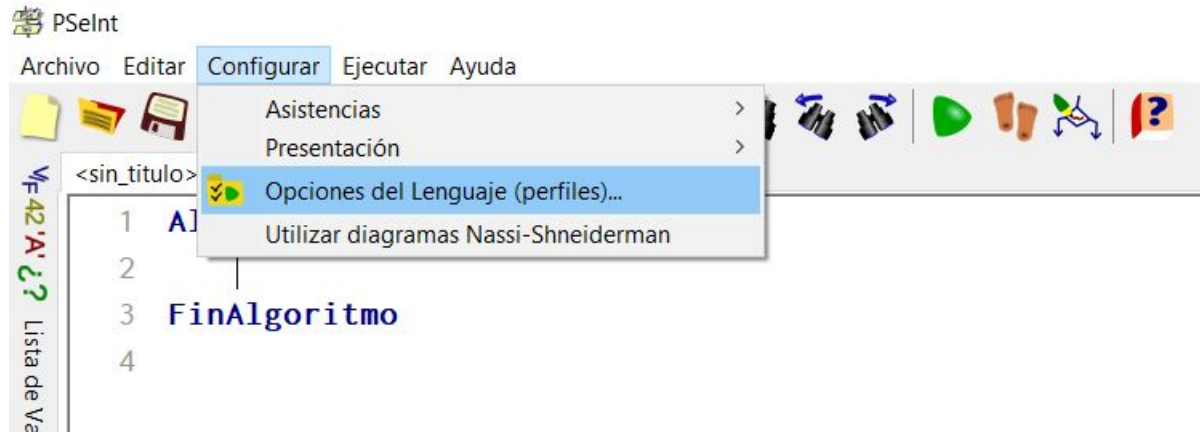




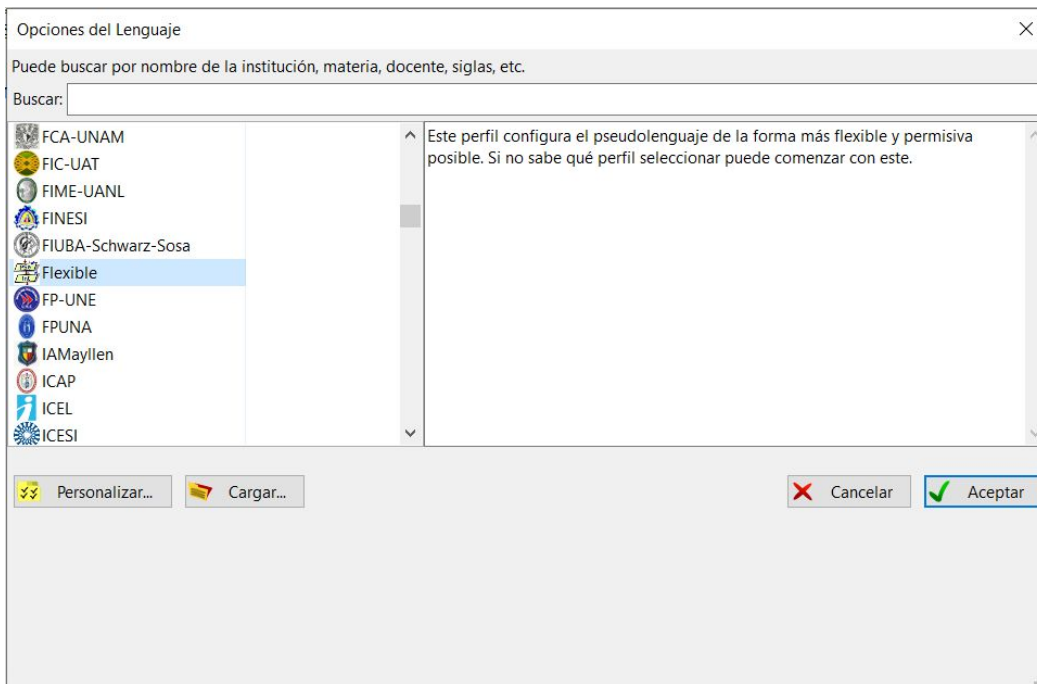
# Instalación



# Configuración







# Configuración



# Configuración

Opciones del Lenguaje

- ☐ No permitir utilizar variables o posiciones de arreglos sin inicializar
- ☐ Obligar a definir los tipos de variables
- ☐ Controlar el uso de ; al final de sentencias secuenciales
- ☒ Permitir concatenar variables de texto con el operador +
- ☒ Habilitar funciones para el manejo de cadenas
- ☒ Permitir las palabras Y, O, NO y MOD para los operadores &, |, ~ y %
- ☐ Utilizar índices en arreglos y cadenas en base 0
- ☒ Permitir utilizar variables para dimensionar arreglos
- ☒ Permitir asignar con el signo igual (=)
- ☒ Permitir definir funciones/subprocesos
- ☒ Utilizar sintaxis flexible
- ☒ Permitir condiciones en lenguaje coloquial
- ☐ Limitar la estructura Según a variables de control numéricas
- ☒ Permitir omitir el paso -1 en ciclos Para
- ☐ Usar diagramas de Nassi-Shneiderman
- ☐ Usar formas alternativas para Leer y Escribir en el diagrama
- ☒ Permitir utilizar acentos en nombres de variables
- ☒ Preferir las palabras clave "Algoritmo" y "FinAlgoritmo"
- ☒ Preferir las palabras clave "Función" y "FinFunción"
- ☒ Permitir la variación "Repetir ... Mientras Que..."
- ☒ Habilitar estructura "Para Cada..."
- ☐ Preferir "Repetir ... Mientras Que..."
- ☐ Proteger contador del Para

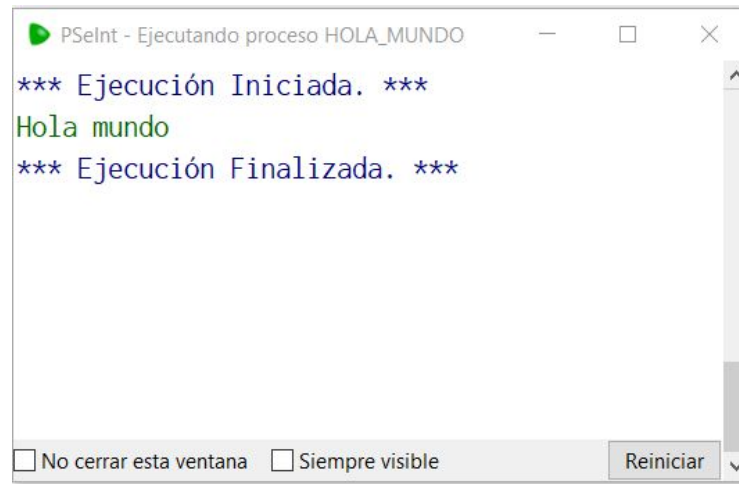
 Cargar...  Guardar...  Cancelar  Aceptar

# Mi primer programa en PSeInt

# El “Hola Mundo” en PSeInt

Conocemos la primera instrucción: **Escribir**

```
1  Algoritmo hola_mundo
2      Escribir "Hola mundo"
3  FinAlgoritmo
4
```



# Desafío de clase

Utilizamos Pselnt para desarrollar los siguientes algoritmos:

- Lavar 5 platos, de manera de dejarlos listos para la próxima vez.
- Preparamos una taza de té, obtenemos un té listo para tomar.

## Lavar 3 platos

1. Inicio
2. Necesito 3 platos
3. Necesito agua en la pileta
4. Necesito detergente
5. Apilar plato 1
6. Apilar plato 2
7. Apilar plato 3
8. Colocar el agua en la pileta
9. Colocar el detergente
10. Lavar plato 1
11. Lavar plato 2
12. Lavar plato 3
13. Escurrir plato 1
14. Escurrir plato 2
15. Escurrir plato 3
16. Obtengo platos limpios
17. Fin

## Preparar una taza de té

1. Inicio
2. Necesito el recipiente, agua, te, cuchara.
3. Coloco agua caliente
4. Selecciono el tipo de infusión
5. Coloco la infusión en el agua
6. Coloco una cucharada de azúcar
7. Coloco una cucharada de azúcar
8. Coloco una cucharada de azúcar
9. Espero 5 minutos
10. Obtengo el té
11. Fin



# Desafío para la clase que viene

- Como te habrás dado cuenta durante las tres clases anteriores **diseñamos un programa:**
  - *analizamos un problema,*
  - *diseñamos el algoritmo y*
  - *lo verificamos con el diagrama de flujo*
- Vas a utilizar el diseño del diagrama de flujo en la clase pasada (Clase 3) para llevarlo a la compu:
  - Para lograrlo utilizaremos PSeInt.



# No te olvides de dar el presente

## Recordá:

- Revisar la Cartelera de Novedades.
- Hacer tus consultas en el Foro.

**Todo en el Aula Virtual.**