

Teoría en Computación

Semana 01: Introducción a la Teoría de la Computación

Mg. Roberto Zárate Mendoza



**Universidad
Tecnológica
del Perú**

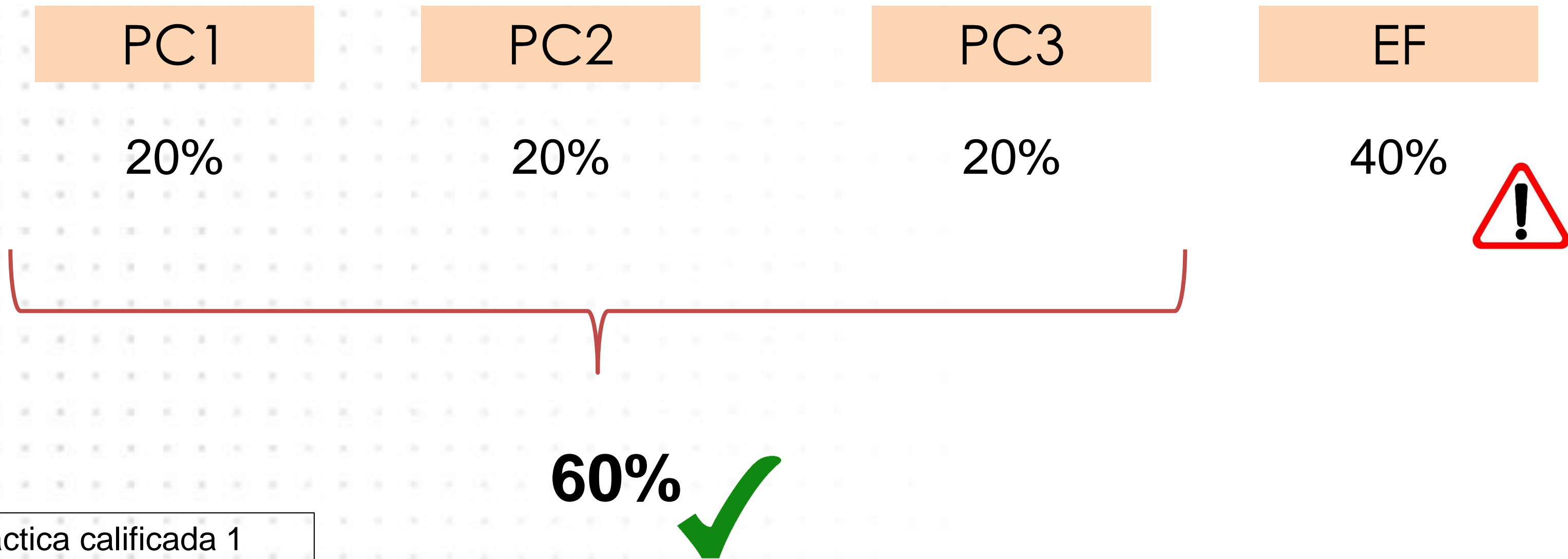
Inicio

Sílabo del curso



Revisamos el sílabo en la plataforma virtual de aprendizajes.

Metodología de Evaluación:



PC1 : práctica calificada 1
PC2: práctica calificada 2
PC3: práctica calificada 3
EF: evaluación final

Elección del Delegado:

Presentación de la Plataforma Utp +Class

Consulta a la clase

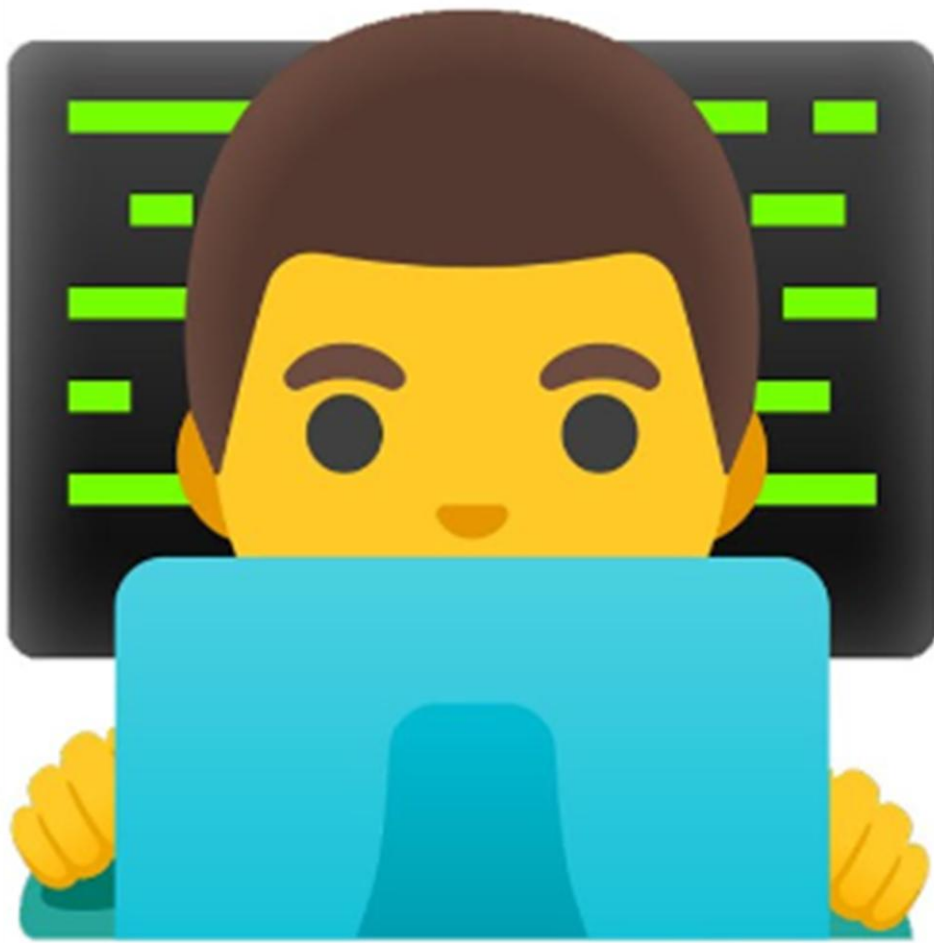
Pregunta a la clase: ¿Qué entendemos por Teoría de la Computación?

Logro de la Semana 1

Al finalizar la semana 1, el estudiante logra entender los conceptos iniciales como compilador, traductor, interprete y el proceso de compilación.



Agenda: Sesión



- **Introducción a la teoría de la computación.**
- **Traductores.**
- **Compiladores.**
- **Interpretes.**

Utilidad

¿Cuál es la importancia de conocer la teoría de la Computación en el desarrollo de aplicaciones?



- Análisis léxico

Utilidad en el campo laboral



Autocompletado y aplicativos en donde se debe revisar texto, herramienta: Autómata finito.

Introducción

Procesos involucrados en la construcción de compiladores, como el análisis léxico , sintáctico y semántico . Estos conceptos son fundamentales para entender cómo se traduce código fuente escrito por humanos en lenguajes de programación (como Python, Java o C++) en código máquina que una computadora puede ejecutar.

Traductores

Un traductor es un programa que toma como entrada un texto escrito en un lenguaje fuente (por ejemplo, un lenguaje de programación de alto nivel como Python o C++) y lo convierte en un lenguaje objetivo (como código máquina o bytecode). Este proceso implica analizar, transformar y generar una salida equivalente en el lenguaje objetivo.

En términos generales, un traductor realiza las siguientes tareas:

Análisis : Descompone el lenguaje fuente en componentes más pequeños (tokens, estructuras gramaticales, etc.).

Transformación : Convierte estas componentes en una representación intermedia o directamente en el lenguaje objetivo.

Generación : Produce la salida final en el lenguaje objetivo.

Compilador

Un compilador es un traductor que convierte un programa completo escrito en un lenguaje de alto nivel (lenguaje fuente) en un lenguaje de bajo nivel (como código máquina o bytecode).

El compilador de C++ convierte código C++ en código máquina ejecutable.

Características :

Procesa todo el programa de una vez.

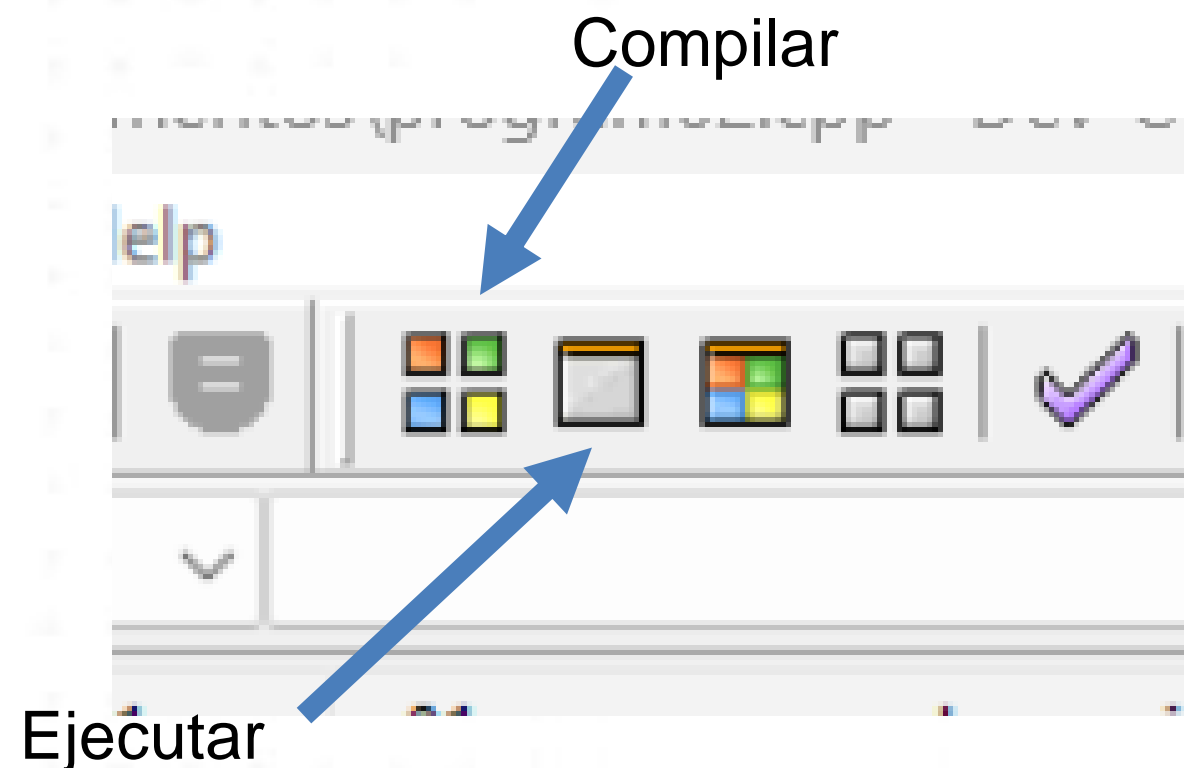
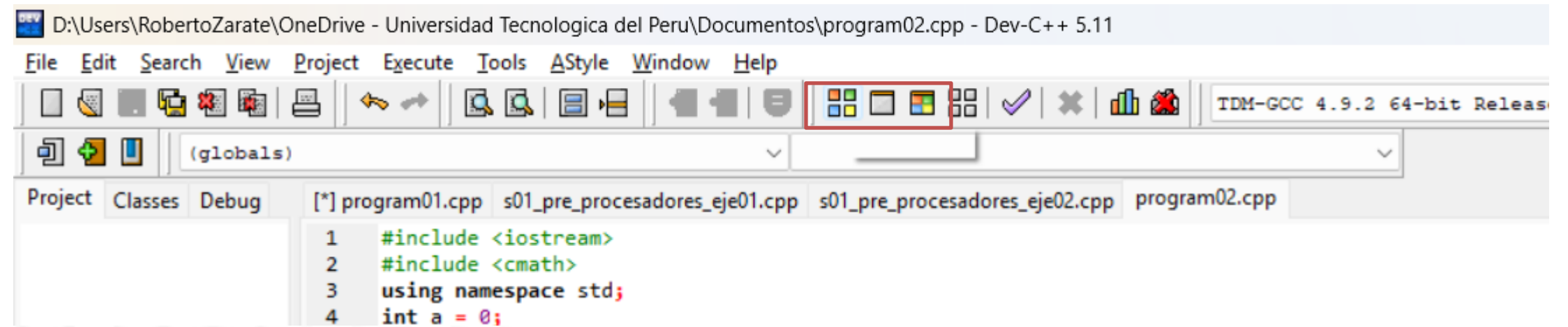
Genera un archivo ejecutable independiente.

Ejemplo 1:

En el lenguaje C se genera un ejecutable en lenguaje máquina. (file.exe)

Compilador

```
#include <iostream>
#include <cmath>
using namespace std;
int a = 0;
float b = 0;
int main(){
    cout << "Hola mundo hey!"<<endl;
    for(int i=0; i<9000000000;i++){
        b = sqrt(a) ;
    }
    return 0;
}
```



Compilador

Actividad práctica:

En java, validamos si tenemos los componentes necesarios:

```
C:\Users\Roberto Zarate>javac -version
javac 17.0.10

C:\Users\Roberto Zarate>java --version
java 17.0.10 2024-01-16 LTS
Java(TM) SE Runtime Environment (build 17.0.10+11-LTS-240)
Java HotSpot(TM) 64-Bit Server VM (build 17.0.10+11-LTS-240, mixed mode, sharing)

C:\Users\Roberto Zarate>
```


Compilador

Actividad 2 - práctica:

Creación de clase, compilamos y ejecutamos:

```
C:\Users\Roberto Zarate>rename HolaMundo.java.txt HolaMundo.java
```

```
C:\Users\Roberto Zarate>dir *.java
```

```
El volumen de la unidad C no tiene etiqueta.
```

```
El número de serie del volumen es: B49D-06DF
```

```
Directorio de C:\Users\Roberto Zarate
```

```
28/02/2025  02:46 a. m.          124 HolaMundo.java
```

```
1 archivos          124 bytes
```

```
0 dirs  231,314,210,816 bytes libres
```

```
C:\Users\Roberto Zarate>cat HolaMundo.java
```

```
public class HolaMundo{
```

```
    public static void main(String[] args){  
        System.out.println("Hola Mundo, hey!");  
    }
```

```
}
```

```
C:\Users\Roberto Zarate>javac HolaMundo.java
```

```
C:\Users\Roberto Zarate>java HolaMundo
```

```
Hola Mundo, hey!
```

Compilar:

Ejecutar:



Universidad
Tecnológica
del Perú

Intérprete

Un intérprete traduce y ejecuta el código línea por línea, sin generar un archivo ejecutable independiente.

Ejemplo : El intérprete de Python ejecuta directamente el código fuente sin necesidad de compilarlo previamente.

Características :

No genera un archivo ejecutable.

Es útil para pruebas rápidas y desarrollo interactivo.

Ejemplo práctico: El shell de Python.

¿Todo claro?



Cierre

Traductor: Es un programa que toma como entrada un texto escrito en un lenguaje fuente y lo convierte en el lenguaje objetivo

Compilador: Es un traductor que convierte un programa completo escrito en un lenguaje de alto nivel (lenguaje fuente) en un lenguaje de bajo nivel (como código máquina o bytecode).

Intérprete: Traduce y ejecuta el código línea por línea, sin generar un archivo ejecutable independiente.

Preguntas de Cierre:

- ¿Qué es un compilador?
- ¿Qué es un intérprete?
- ¿Qué es un traductor?





**Universidad
Tecnológica
del Perú**