

IIC2133

Estructuras de Datos y Algoritmos

Las Clases

Nos importa que aprovechen las clases, por lo que hemos preparado un sistema de instancias de aprendizaje activo.

Podrán reconocer las distintas instancias por sus íconos:



Den ideas



Nos importa que puedan deducir los contenidos del curso

No tengas miedo en decir lo que piensas, nada es obvio

Cuando veas la ampolleta, es el momento de dar ideas

Control



Los controles son para diagnóstico: no llevan nota

En su lugar, se darán décimas por respuestas correctas

Cuando veas la hojita, relájate y da lo mejor de ti

Discute con tus compañeros



Nos importa que puedas discutir ideas con otras personas

¿Cuál crees que es la respuesta y por qué?

Cuando veas los globos de texto, prepara tu argumento

Las Tareas

En las primeras dos ayudantías se hará una actividad introductoria al lenguaje de programación **C** (con nota)

Durante el semestre habrá 4 tareas de programación en **C**

La nota de tareas es el promedio entre la actividad y las tareas

Las Pruebas

El curso consta de 3 interrogaciones y un examen

I1	3 de Abril
I2	3 de Mayo
I3	30 de Mayo
Ex	27 de Junio

La nota de interrogaciones se calcula borrando la peor

Las Notas

$$NT = \frac{A + T_1 + T_2 + T_3 + T_4}{5}$$

$$NI = \frac{I_1 + I_2 + I_3 + 2 \cdot Ex - \min(I_1, I_2, I_3, Ex)}{4}$$

$$NF = \begin{cases} \frac{NI + NT}{2}, & \text{si } NI \geq 4 \\ \min(NI, NT), & \text{en otro caso} \end{cases}$$

GitHub: Plataforma Oficial del Curso

En el repositorio del curso en GitHub podrán encontrar:

- Guías de instalación de C y algunas librerías
- El foro para dudas de tareas, materia, etc.
- Los enunciados de las tareas y las diapositivas de clases
- Sus propios repositorios para entregar las tareas

Deben contestar la encuesta en el **SIDING** para poder acceder

Algoritmos



Nuestra primera ampolleta

¿Qué es un algoritmo?

Algoritmos

Un **algoritmo** y su **implementación** pueden ser distintos

En clases veremos los algoritmos de manera conceptual

En las tareas tendrán que pensar en la implementación

Como hacer Puré de Papas

1. Pelar las papas y echarlas en una olla
2. Llenar una olla con agua. Agregar sal a gusto
3. Poner la olla al fuego hasta que las papas estén blandas
4. Sacar las papas del agua y vaciar la olla
5. **Moler** las papas y volverlas a poner en la olla
6. Agregar leche, calentar y revolver. Servir caliente.

Como hacer Puré de Papas

¿Cómo **moler** las papas?



Todos muelen... pero no de igual manera

Algoritmos



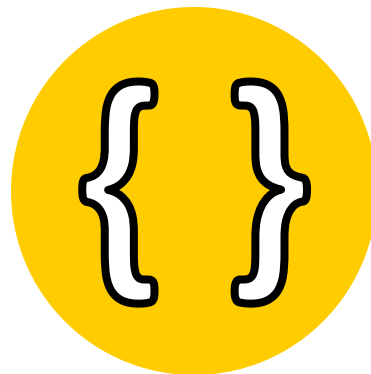
¿Qué hace que una solución a un problema sea buena?

¿Cuál será la mejor?

Detalles de Implementación

Si bien el análisis algorítmico será abstracto, no podemos olvidar que hay que implementarlos en un computador

A la hora de pensar en código, verás este ícono:



Complejidad

$$f(x) \in O(g(x)):$$

$$\exists x_0, k > 0$$

$$f(x) < k \cdot g(x), \quad \forall x > x_0$$

$$f(x) \in \Omega(g(x)):$$

$$g(x) \in O(f(x))$$

$$f(x) \in \Theta(g(x)):$$

$$f(x) \in O(g(x)) \quad y \quad f(x) \in \Omega(g(x))$$

Complejidad

- Si un algoritmo tiene varias partes, su complejidad es la **suma** de las complejidades de cada parte
- Por lo anterior, si una parte se repite x veces, entonces se puede **multiplicar** su complejidad por x , y luego sumar
- Al final, sólo queda el término de **mayor** grado

Complejidad

Nos interesan dos tipos de complejidad para algoritmos:

- Complejidad de tiempo: $T(n)$
- Complejidad de memoria: $M(n)$

Ambos relativos al tamaño del input, n

Si bien $T(n)$ es nuestra prioridad, nunca olvidar que

$$T(n) \in \Omega(M(n))$$

¿Por qué C?

El lenguaje C posee las siguientes características:

- Control absoluto de memoria
- Control absoluto del algoritmo
- No trae nada que no sea necesario

Esto significa que el lenguaje tiene mínimo *overhead*

¿Por qué no Python?

El lenguaje Python posee las siguientes características:

- Estructuras de datos integradas
- Fácil y rápido de escribir
- Se encarga de casi todo para comodidad del usuario

Esto significa que el lenguaje muchas veces hace cosas sin que uno sepa, aumentando la complejidad del programa.

Próxima Clase

La próxima clase comenzaremos con los contenidos del curso

Deben traer leída la guía de **arreglos** que está en el repositorio

Deben tener clara la diferencia entre una lista y un arreglo

Actividad 1

Para la ayudantía de este viernes necesitan lo siguiente:

- Haber leído y seguido la guía de instalación de **C**
- Haber leído la guía de **arreglos**
- Haber leído la guía de **punteros**
- Traer su computador con todo listo para trabajar en C