

UNIVERSIDAD  
**AUSTRAL**



**INGENIERÍA**

---

# Álgebra III

## TP I

Algoritmos iterativos y recursivos

## Ejercicios

---

Los alumnos deberán formar grupos de 2 personas. Una vez formado, enviar un email con los nombres de los integrantes a la dirección [agustin.olmedo@ing.austral.edu.ar](mailto:agustin.olmedo@ing.austral.edu.ar). Como respuesta, recibirán su número de equipo y los ejercicios que les corresponde solucionar. Los ejercicios se pueden encontrar en la "Guía N° 1" que está en el campus en la Unidad 1.

Los ejercicios a realizar se deberán solucionar tanto de forma iterativa como recursiva. Esto significa que por cada ejercicio asignado deben entregar la solución iterativa y la recursiva. Para ello, recibirán dos clases a implementar: "Guide1IterativeSolution" y "Guide1RecursiveSolution". En las mismas, sólo deberán solucionar los ejercicios que les fueron asignados.

## Entrega

---

La fecha límite de entrega es el día martes 29 de marzo, a dos semanas de la presentación del trabajo práctico.

### Java

Los alumnos deberán modificar dos archivos: "Guide1IterativeSolution.java" y "Guide1RecursiveSolution.java" resolviendo los ejercicios que les fueron asignados. Estos archivos y los tests que serán usados para corregir se pueden encontrar en el repositorio de github: <https://github.com/aolmedo87/algebra3>.

### Python

Los alumnos deberán modificar dos archivos: "Guide1IterativeSolution.py" y "Guide1RecursiveSolution.py" resolviendo los ejercicios que les fueron asignados. Estos archivos y los tests que serán usados para corregir se pueden encontrar en el repositorio de github: <https://github.com/aolmedo87/algebra3>.

La entrega deberá estar conformada por un archivo .zip con la siguiente nomenclatura: "Team[Number]Solution.zip" (ejemplo: Para el equipo 1, el zip debe llamarse "Team1Solution.zip"). Dentro del mismo deben encontrarse todos los archivos necesarios para correr los tests y que pasen.

## Corrección

---

Se evaluarán 2 aspectos:

1. Unit tests: Todos los test de *"Guide1Test.java"* y *"Guide1Test.py"* que corresponden a los ejercicios de la entrega deben pasar. Si el test falla, el ejercicio se da por desaprobado (0 puntos).
2. Análisis del código: Una vez pasados los tests se leerá que el código sea correcto, es decir que sea claro de leer (prestar atención a el nombre de variables y métodos auxiliares entre otras cosas) y no tenga operaciones innecesarias. Esto no penaliza mucho, a menos que haya un error conceptual grave.

En caso de no entregar a tiempo, o de haber desaprobado el alumno tendrá la posibilidad de recuperar el trabajo práctico.

## Nota

---

La tabla a continuación muestra la cantidad de puntos por ejercicio:

	Ej 1	Ej 2	Ej 3	Ej 4	Ej 5
Puntos	2	2	1	3	2

Los puntos por cada ejercicio valen la mitad por cada implementación. Ejemplo, el en ejercicio 1 cuyo total es de 2 puntos, la implementación recursiva vale la mitad (1 punto) y la implementación iterativa vale el resto (1 punto)

La nota final es la suma de los puntos totales (p), utilizando la siguiente fórmula de interpolación:

$$nota = \left( \frac{p-6}{4} \right) \cdot 10 + \left( \frac{p-10}{-4} \right) \cdot 4$$

## Aclaraciones

---

- Todo ejercicio iterativo debe tener al menos un *"loop instruction"*.
- Todo ejercicio recursivo, debe llamar a un método recursivamente. Se pueden usar métodos auxiliares.

A continuación un ejemplo de resolución del ejercicio 2b de la guía 1 (  $2^n$  iterativo) que está mal (0 puntos) aunque pase los tests. En este caso, no se cumple la consigna de resolver de manera iterativa o recursiva, ya que no hay ningún “*loop instruction*” presente.

```
public int exercise_2_b(int n) { return (int) Math.pow(2, n); }
```

## Consultas

---

Para: aclaraciones, consultas, dudas y sugerencias mandar un correo electrónico a [agustin.olmedo@ing.austral.edu.ar](mailto:agustin.olmedo@ing.austral.edu.ar).