



## UNIVERSIDAD NACIONAL DE INGENIERIA

### Facultad de Ingeniería Industrial y de Sistemas

#### SI302 - PROGRAMACION ORIENTADA A OBJETOS

PROFESOR	Eric Gustavo Coronel Castillo	FECHA	16/10/2023
EVALUACION	EXAMEN PARCIAL	SECCION	V

### INDICACIONES

- La prueba es individual.
- Crear una carpeta con el siguiente formato: **POO\_EP\_AAAAA**, donde AAAAA representa su apellido paterno, esta carpeta será su carpeta de trabajo.
- En su carpeta de trabajo, utilizando Apache NetBeans crear los proyectos **PROY1\_AAAAA** y **PROY2\_AAAAA** correspondientes a los proyectos solicitados, donde AAAAA representa su apellido paterno, estos nombres para los proyectos son requisitos fundamentales.
- Es importante recordarte que los proyectos debes asegurarte de crearlos en tu carpeta de trabajo, es tu responsabilidad.
- 10 minutos antes de finalizar el tiempo del examen debe cerrar el IDE y empaquetar su carpeta de trabajo en un archivo **RAR** o **ZIP**, es importante recordarte que los proyectos los debiste crear en tu carpeta de trabajo.
- El archivo empaquetado es el que debe subir al aula virtual (<https://univirtual.uni.pe>) en la sección que le corresponde, es la única forma de entregar la solución del examen, no existen otras formas o medios de entrega de entrega.
- Se calificará con nota A0 soluciones parecidas o iguales.
- Los proyectos deben resolverse bajo la programación en capas, el enfoque de servicios y las buenas prácticas de desarrollo de software..



## UNIVERSIDAD NACIONAL DE INGENIERIA

### Facultad de Ingeniería Industrial y de Sistemas

## Proyecto 1 (8 Puntos)

Se necesita una aplicación que permita evaluar lo siguiente:

### Número de Amstrong

Un número de Amstrong es aquel que es igual a la suma de sus dígitos elevados a la potencia de su número de cifras.

Por ejemplo, el número 371 es un número que cumple dicha característica ya que tiene tres cifras y:

$$371 = 3^3 + 7^3 + 1^3 = 27 + 343 + 1 = 371$$

### Evaluar serie

Dado un número entero positivo  $n$ , calcule la suma de la siguiente serie:

$$\text{Serie} = 1 + 1/2 + 1/3 + 1/4 + 1/5 + \dots + 1/n$$

### Fibonacci

Encontrar los primeros “ $n$ ” números de Fibonacci. Los números de Fibonacci comienzan con 0 y 1, y cada término siguiente es la suma de los anteriores: 0, 1, 2, 3, 5, 8, 13, 21, ...

Se solicita:

- La clase de servicios que implemente los requerimientos solicitados.
- La clase o clases de prueba necesarios para probar los servicios.
- Una interface de usuario amigable.



## UNIVERSIDAD NACIONAL DE INGENIERIA

### Facultad de Ingeniería Industrial y de Sistemas

## Proyecto 2 (12 Puntos)

Se necesita una aplicación que permita analizar dos arreglos de números enteros, para lo cual se ha propuesto la siguiente clase de servicios:

ArregloService
<ul style="list-style-type: none"><li>- arreglo1 : int[]</li><li>- arreglo2 : int[]</li></ul>
<ul style="list-style-type: none"><li>+ ArregloService( int n )</li><li>+ getArreglo1() : int[]</li><li>+ getArreglo2() : int[]</li><li>+ arregloDiferencia() : int[]</li><li>+ arregloInterseccion() : int[]</li></ul>

### Donde:

- **arreglo1 y arreglo2** : Se trata de 2 variables donde se guardan los datos de 2 arreglos que el constructor debe generar de manera aleatoria.
- **ArregloService** : Se trata del constructor, recibe el valor de “n” que representa el tamaño de los arreglos. Debe generar los 2 arreglos con números aleatorios que deben estar en el rango [21,30].
- **getArreglo1** : Este método retorna el **arreglo1**.
- **getArreglo2** : Este método retorna el **arreglo2**.
- **arregloDiferencia** : Retorna los elementos del arreglo1 que no se encuentran en arreglo2, sin considerar los duplicados.
- **arregloInterseccion** : Retorna los elementos comunes a ambos arreglos, sin considerar los duplicados.



## UNIVERSIDAD NACIONAL DE INGENIERIA

### Facultad de Ingeniería Industrial y de Sistemas

Debe considerar lo siguiente:

- Las firmas de los métodos solicitados en la clase ArregloService no pueden ser modificados.
- Si usted necesita o considera necesario crear otros métodos privados puede hacerlo, tal vez con el criterio de no repetir código u otra buena practica.
- Si conoce alguna librería dentro de Java (No externa) para el tratamiento de arreglos puede utilizarla.

Por ejemplo, para un valor de  $n=10$ , una posible respuesta podría ser:

Arreglo 1: 21 27 27 30 21 26 27 30 23 30

Arreglo 2: 24 30 24 30 23 25 29 28 29 24

Arreglo diferencia: 21 27 26

Arreglo intersección: 30 23

Se solicita:

- La clase ArregloService totalmente implementada.
- Una clase de prueba, debe verificar el funcionamiento completo de la clase ArregloService.
- Una interface de usuario amigable.

## CRITERIOS DE EVALUACIÓN

ITEM	PUNTAJE MAXIMO
<b>Proyecto 1</b> <ul style="list-style-type: none"><li>▪ Capa service: 4 puntos</li><li>▪ Prueba de servicios: 3 Puntos</li><li>▪ Capa view: 3 Puntos</li></ul>	10
<b>Proyecto 2</b> <ul style="list-style-type: none"><li>▪ Capa service: 4 puntos</li><li>▪ Prueba de servicios: 3 Puntos</li><li>▪ Capa view: 3 Puntos</li></ul>	10
<b>PUNTAJE TOTAL MÁXIMO</b>	20