



## UNIVERSIDAD NACIONAL DE INGENIERIA

### Facultad de Ingeniería Eléctrica y Electrónica

BMA15 - PROGRAMACION ORIENTADA A OBJETOS			
PROFESOR	Eric Gustavo Coronel Castillo	FECHA	01/02/2024
EVALUACION	EXAMEN PARCIAL	SECCION	M
ESTUDIANTE			

## 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 debes asegurarte de crear tus proyectos en tu carpeta de trabajo, es tu responsabilidad.
- 10 minutos antes de finalizar el tiempo del examen debes 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.
- 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 Eléctrica y Electrónica

## Proyecto 1 (8 Puntos)

Se necesita una aplicación que permita evaluar lo siguiente:

### Número perfecto

Evaluar si un número es perfecto. Un número perfecto es aquel que es igual a la suma de sus divisores positivos.

Por ejemplo, el número 28 es un número perfecto ya que sus divisores son: 1, 2, 4, 7 y 14, y la suma de estos números es 28.

### Números amigos

Evaluar si dos números son amigos. Dos números enteros positivos se consideran amigos si la suma de los divisores de uno es igual al otro número y viceversa.

Por ejemplo, los números 220 y 284 son amigos. Los divisores del número 220 son: 1, 2, 4, 5, 10, 11, 20, 22, 44, 55 y 110, y suman 284. Los divisores de 284 son: 1, 2, 4, 71 y 142, que suman 220.

### Serie

Evaluar la siguiente serie:

$$S = \sum_{i=0}^n \frac{(-1)^i}{2i+1} x^{2i+1} = x - \frac{x^3}{3} + \frac{x^5}{5} - \dots$$

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 Eléctrica y Electrónica

## 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:

VectorService
- vector3 : int[] - vector5 : int[]
+ VectorService( int n ) + getVector3() : int[] + getVector5() : int[] + vectorComun() : int[] + vectorUnion() : int[]

### Donde:

- **vector3** : Se trata de una variable donde se guardan los datos de 1 arreglo de números múltiplos de 3 que el constructor debe generar de manera aleatoria.
- **vector5** : Se trata de una variable donde se guardan los datos de un arreglo de números múltiplos de 5 que el constructor debe generar de manera aleatoria.
- **VectorService** : Se trata del constructor, recibe el valor de “n” que representa el tamaño de los vectores. Debe generar los 2 vectores con números aleatorios que deben estar en el rango [10,50]. Recuerda, que los números generados deben ser múltiplos de 3 y 5 respectivamente.
- **getVector3** : Este método retorna el **vector3**.
- **getVector5** : Este método retorna el **vector5**.
- **vectorNoComun** : Este método retorna un vector con los elementos no comunes de ambos vectores, sin considerar los duplicados.
- **vectorUnion** : Este método retorna un vector con todos los elementos de ambos vectores, sin considerar los duplicados.



## UNIVERSIDAD NACIONAL DE INGENIERIA

### Facultad de Ingeniería Eléctrica y Electrónica

Debe considerar lo siguiente:

- Las firmas de los métodos solicitados en la clase VectorService 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 práctica.
- 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:

Vector 3: [27,15,24,18,21,48,24,30,24,30]

Vector 5: [25,30,45,50,40,35,15,40,10,50]

Vector NO Común: [27,24,18,21,48,25,45,50,40,35,10]

Vector Unión: [27,15,24,18,21,48,30,25,45,50,40,35,10]

Se solicita:

- La clase **VectorService** totalmente implementada.
- Una clase de prueba, debe verificar el funcionamiento completo de la clase VectorService.
- 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: 2 Puntos</li><li>▪ Capa view: 2 Puntos</li></ul>	8
<b>Proyecto 2</b> <ul style="list-style-type: none"><li>▪ Capa service: 6 puntos</li><li>▪ Prueba de servicios: 3 Puntos</li><li>▪ Capa view: 3 Puntos</li></ul>	12
<b>PUNTAJE TOTAL MÁXIMO</b>	20