



ugr

Universidad  
de Granada



## FUNDAMENTOS DE PROGRAMACIÓN

### EXAMEN PRÁCTICO 2

Dpto. Ciencias de la Computación e Inteligencia Artificial  
E.T.S. de Ingenierías Informática y de Telecomunicación  
Universidad de Granada

**!Importante!** No se corregirá ninguna práctica que no cumpla escrupulosamente las normas que aparecen a continuación.

#### Para la realización del examen:

- Para iniciar sesión en las aulas de prácticas, tendrá que introducir su identificador de usuario y contraseña como siempre. En la casilla correspondiente a *Código*: **examenfp** . Con este código sólo tendrá acceso Internet a la página de la asignatura. No tendrá disponible su unidad en red U: ni los puertos USB.
- Se creará un fichero fuente con la solución del problema, nombrándolo *problema.cpp*.
- Este archivo debe contener en la cabecera, como comentario, la siguiente información: nombre y apellidos, DNI y grupo de prácticas al que se asiste.
- En la evaluación de las prácticas se tendrá en cuenta, además de la corrección de la solución propuesta, el estilo de programación, el uso correcto de espacios y tabuladores, así como la claridad del código fuente.
- La entrega de la práctica se hará durante el tiempo de duración del examen, y se hará en entrega de prácticas accesible a partir de la página web de la asignatura <https://decsai.ugr.es> (Menú Entrega prácticas). Bien directamente <https://150.214.191.180> en caso de problemas.
- Debéis asegurarnos de entregar en vuestro grupo de prácticas.
- La práctica se puede entregar tantas veces como se quiera durante el examen.
- El sistema guarda la última entrega. De hecho, **se recomienda que se entregue varias veces a lo largo del examen**, ya que si el ordenador se quedara colgado, habría que reiniciarlo y se perdería toda la información.

**Tiempo de examen: 1.30 HORA**

Partiendo de la implementación básica de la Clase SecuenciaEnteros disponible en decsai, se plantearán extensiones que deberá implementar como métodos de la clase. Los métodos son:

- **Unión:** devolverá una nueva secuencia de enteros que contendrá todos los elementos, sin repeticiones, de la secuencia actual y una nueva secuencia que se pasa como parámetro
- **Intersección:** devolverá una nueva secuencia de enteros que contendrá los elementos que se repiten en la secuencia actual y una nueva secuencia que se pasa como parámetro.
- **Diferencia:** devolverá una nueva secuencia de enteros con los elementos que están en la secuencia actual pero no en una nueva secuencia que se pasa como parámetro.
- **OPCIONAL] Diferencia simétrica:** devolverá una nueva secuencia de enteros como los elementos de la secuencia actual y de una nueva secuencia que se pasa como parámetro siempre que no estén en ambas.

#### NOTAS:

- En todos los casos es necesario devolver un nuevo objeto de la clase SecuenciaEnteros.
- Realice cada método de manera iterativa, es decir, primero la unión, luego la intersección, y chequee la validez de los mismos. Recuerde que es mejor tener dos métodos bien implementados que 4 mal hechos.
- Muestre en el “main” ejemplos de uso de los métodos implementados.
- Luego de aplicar cada método, muestre los valores de la secuencia resultante.
- Si hace falta implementar algún método adicional, hágalo y justifique si debe ser un método public o privado.
- Ejemplos de uso:
  - Dada las secuencias {1, 5, 3, 8, 4} y {0, 6, 7, 5, 1}
  - UNIÓN: {1, 5, 3, 8, 4, 0, 6, 7}
  - INTERSECCIÓN: {1, 5}
  - DIFERENCIA: {3, 8, 4}
  - DIFERENCIA SIMÉTRICA: {3, 8, 4, 0, 6, 7}