

# Taller de Random Testing

Ingeniería de Software II  
Departamento de Computación  
Facultad de Ciencias Exactas y Naturales  
Universidad de Buenos Aires

Importar el proyecto StackAr en la IDE *IntelliJ IDEA*.

- Para correr los tests, hacer click derecho sobre el módulo test en la IDE y seleccionar Run Tests.

Opcionalmente, se pueden correr los tests por línea de comando: `./gradlew test`.

- Para correr *pitest*, hacerlo directamente por línea de comando: `./gradlew pitest`.

El output del report de PiTest queda en `build/reports/pitest/index.html`. Abrir ese archivo con un navegador.

- Para correr *Randoop*, hacerlo directamente por línea de comando: `./gradlew randoop`.

Pueden encontrar más información de como usar Randoop en:

- <https://randoop.github.io/randoop/>
- <https://randoop.github.io/randoop/manual/index.html>

También, puede revisar los argumentos de la *Gradle task* **randoop** en el archivo `build.gradle`.

# Ejercicio 1

Ejecutar Randoop sobre la clase StackAr para que genere todos los tests aleatorios posibles durante 15 segundos.

¿Cuántos test cases produjo Randoop? ¿Hay failing test cases?

Ejecutar los casos de tests generados por Randoop (por línea de comando o directamente en la IDE). Usaremos la herramienta JaCoCo para obtener la cobertura lograda por los tests. Puede encontrar el reporte generado en `build/reports/jacoco/test/html/index.html`.

- ¿Cuántas líneas cubiertas reporta JaCoCo?
- ¿Cuántos branches cubiertos reporta JaCoCo?

## Ejercicio 3

Completar el método `StackAr.repoOK()` para que retorne **true** solamente si la estructura del `StackAr` es válida.

- Una instancia de `StackAr` es válida sii:
  - $elems \neq null$
  - $readIndex \geq -1$  y  $readIndex < elems.length$
  - $\forall i > readIndex, elems_i = null$
- Ejecutar Randoop por 1 min sobre `StackAr`.
- Reportar el número de failing tests.

Si hay failing tests, reparar el programa hasta que Randoop no reporte más failing tests usando un minuto de presupuesto.

Ejecutar PiTest sobre el último test suite generado por Randoop.

- 1 ¿Cuántas mutantes construye PiTest? ¿Cuál es el mutation score?
- 2 Extender manualmente el test suite para obtener el mejor mutation score posible con PiTest. ¿Cuál es el mejor mutation score que pudo obtener? ¿Qué mutantes equivalentes encontró?



Entregar un archivo comprimido que incluya todo el proyecto StackAr modificado.

El taller debe ser entregado a través del campus y debe incluir el siguiente material (preferentemente colocar todo en un archivo zip).

- 1 Un informe con las respuestas a las preguntas de cada punto. Para esto, puede agregar un archivo `informe.txt` o `informe.pdf` en la raíz del proyecto.
- 2 Los tests generados para el ejercicio #1.
- 3 Los tests generados para el ejercicio #3.
- 4 Los tests finales generados para el ejercicio #4.
- 5 La versión reparada de StackAr.