

Antoni Oliver. Departament de Ciències Matemàtiques i Informàtica. Universitat de les Illes Balears.

El repositori d'exercicis autocorregits està basat en la llista d'exercicis de l'assignatura de Programació – Informàtica I, que van ser redactats pels professors de l'assignatura.

## Introducció

Aquest document acompanya un repositori moodle d'exercicis autocorregits en Java, per a l'assignatura Programació – Informàtica I, que està basat en el sistema CodeRunner.

En assignatures tècniques, en aquest cas, concretament, les de programació, els alumnes sovint tenen una gran llista d'exercicis per practicar però a l'hora de comprovar si han fet bé la tasca que se'ls demanava, han d'assegurar-se de comprovar tots els casos possibles (per exemple: què passa si en fer un programa que resol una equació de segon grau, el terme quadràtic és zero?). Moltes vegades, això no passa i els alumnes donen per acabada una solució incompleta.

Una manera de donar solució a aquest problema és la correcció automàtica dels exercicis. Hi ha diversos sistemes d'aquest tipus, com ara *CodingBat*<sup>1</sup>, *DOMJudge*<sup>2</sup>, *Codewars*<sup>3</sup>, *Qualified*<sup>4</sup>, o *repl.it*<sup>5</sup>, però en tots els casos són sistemes externs al moodle que es fa servir com a Aula Digital a la UIB.

Un altre sistema és CodeRunner<sup>6</sup>, que sí que s'integra en el moodle (tot i que fa ús d'un servidor extern, però de manera transparent a l'usuari).

## Els exercicis

He adaptat la llista d'exercicis de l'assignatura Programació – Informàtica I a aquest sistema i l'he fet servir durant el curs 2021-22 en els estudis d'Enginyeria Telemàtica. Aquests exercicis han estat dissenyats com a preguntes de qüestionari de Moodle, que

---

<sup>1</sup> <https://codingbat.com/java>

<sup>2</sup> <https://www.domjudge.org/>

<sup>3</sup> <https://www.codewars.com/>

<sup>4</sup> <https://www.qualified.io/for-education>

<sup>5</sup> <https://repl.it/site/classrooms>

<sup>6</sup> <https://coderunner.org.nz/>

fan servir el tipus de pregunta CodeRunner<sup>7</sup> que ha de ser instal·lat per l'administrador del moodle (a la UIB ja està disponible). Llavors, aquestes preguntes es poden incorporar a qüestionaris que poden tenir efecte a la qualificació de l'assignatura d'una manera o de l'altra. En el cas d'aquesta assignatura, fer aquests exercicis no compta per nota, però sí que compten a una "qualificació" que és "exercicis fets", que serveix per motivar un poc els alumnes a fer-los. Alguns exercicis d'aquests són un poc més complets i són "exercicis de final de tema", que compten per "punts de gamificació".

Amb CodeRunner, les preguntes fan servir una "pregunta base" (prototype) que defineix com funciona, que és, a la seva vegada, una pregunta (però no està pensada per ser resposta, perquè no té casos de prova). S'han dissenyat tres preguntes base:

1. **Mètode Java amb extra:** basada en la pregunta "Java Method", avalua un mètode Java que ha escrit l'alumne, comparant-lo amb un mètode que ha indicat el professor a "global extra". A cada cas de prova, el professor pot definir un "extra template data" per executar els dos mètodes i comprovar si el resultat és el mateix.
2. **Programa LT:** basada en "Java Program", hi incorpora dues millores: primer, elimina la definició de Package indicada per l'alumne, ja que el sistema no ho suporta i segon, incorpora la biblioteca de codi "LT.java" que es fa servir a l'assignatura per facilitar la lectura de valors des del teclat.
3. **Programa LT avançat:** substitueix l'anterior i és implementada des de zero. En aquest cas, es permet adjuntar també fitxers de codi i permet l'adaptació del template de la pregunta en exercicis més avançats.

Llavors, s'adjunten 106 exercicis basats en aquest sistema i que fan servir una d'aquestes preguntes base. Els exercicis estan organitzats d'aquesta manera: amb el codi T<tema>E<nombre>, els exercicis per practicar el que es veu després de les classes teòriques del tema indicat; amb el codi T<tema>P<nombre>, els exercicis per practicar durant o després de les classes pràctiques del tema indicat; amb el codi T<tema>F<nombre>, els exercicis de final del tema indicat, per demostrar que s'han assolit les competències d'aquell tema.

Per exemple, un exercici que demana fer un programa que llegeixi un nombre des del teclat i digui si era entre -1 i +1, estaria basat en "Programa LT avançat", hauria d'incorporar una descripció de la tasca a fer, tenir una solució model per si els alumnes volen comprovar una versió model, i comptar amb múltiples casos de prova, com per exemple:

---

<sup>7</sup> [https://moodle.org/plugins/qtype\\_coderunner](https://moodle.org/plugins/qtype_coderunner)

Entrada estàndard	Sortida estàndard
0	Introdueix un nombre real: 0.0 és entre -1.0 i 1.0.
-0.5	Introdueix un nombre real: -0.5 és entre -1.0 i 1.0.
1.0	Introdueix un nombre real: 1.0 no és entre -1.0 i 1.0.
0.999	Introdueix un nombre real: 0.999 és entre -1.0 i 1.0.
-1.234	Introdueix un nombre real: -1.234 no és entre -1.0 i 1.0.

Els exercicis que demanen implementar un mètode fan servir el tipus de pregunta “Mètode Java amb extra”. Per exemple, un exercici que demana fer un mètode que retorni el nombre primer primer a partir d'un nombre passat per paràmetre pot incorporar un mètode *resposta* que ho solucioni a *global extra* i llavors fer-lo servir al cas de prova com a *extra template data* dins un bucle per provar els nombres des de 1 fins a 1000.

Finalment, s'adjunten exercicis que tracten fitxers. En aquest cas, es pot simular l'accés a fitxers a través del camp *extra template data* dels casos de prova. Però per fer això, s'ha d'adaptar el *template* de la pregunta segons les necessitats de l'exercici. Un exemple d'això pot ser l'exercici T7E04, que demana comptar les paraules que es troben en un fitxer i per comprovar la resposta de l'usuari primer escriu el contingut d'aquest camp *extra template data* a un fitxer abans d'executar el cas de prova. Un altre exemple és l'exercici T7E07 que demana escriure en un fitxer les paraules d'un altre fitxer que compleixen un requisit: en aquest cas, també s'escriu el contingut d'*extra template data* al fitxer d'entrada, s'executa el cas de prova i llavors es llegeix el fitxer de sortida i es mostra com a sortida estàndard i és comparat normalment. Es poden combinar la sortida estàndard normal del programa i un fitxer com a l'exercici T7F01, que mostra les dues coses separades per un text arbitrari i llavors ho compara segons aquest mateix format.

## Conclusió

En definitiva, si sou un professor d'aquesta assignatura i voleu engrescar els alumnes a fer aquesta llista d'exercicis, podeu fer servir el repositori que he preparat on he adaptat tots aquests exercicis a aquest format.

Si sou un professor d'una altra assignatura, podeu fer servir aquesta implementació com a model per fer el mateix amb els vostres exercicis.

En qualsevol cas, no dubteu en posar-vos en contacte amb mi:  
<antoni.oliver@uib.cat>.