Praktika honen helburua, 2. mailako Software Ingeniaritza irakasgaian garatutako proiektuaren probak garatzea da (estatikoak eta dinamikoak). Ebaluazio jarraituan garatu zen hasierako proiektuaren kodea hurrengo helbidean aurkitu dezakezu: https://github.com/jononekin/Bets2021. Proiektua Mock-friendly izateko errefaktorizatu da, eta src/test/java karpetan **probaProiektualterazio0.pdf** dokumentua aurkitu dezakezu createQuestion metodoaren diseinuarekin eta proba garatzeko arkitektura desberdinekin (oso garrantzitsua da begiratzea).

Proiektua taldeka edo indibidualki egin daiteke, edozein kasuan, **ikasle bakoitzak metodo baten** diseinu eta inplementazioak garatu beharko ditu (eta proba estatikoak).
Garatutako proiektua Github-ean egon beharko da, eta bere analisia kodea sonarcloud-en.

Aukeratutako metodoek %100 estalpena izan beharko dute (sonarcloud.io).
Ariketaren helburua akatsak identifikatzea da. Hasierako kodea **ez da zuzendu behar**.

1. Egin beharreko lana (ikasle bakoitzak):

i. Lehendabiziko laborategian proposatu ziren akasts estatiko batzuk zuzendu eta dokumentatu sonar.pdf fitxategian (Sonar-eko lehendabiziko laborategia ikusi).
ii. DataAccess-eko metodo bat aukeratu. Aukeratutako metodoak konplexutasun ziklomatikoa 4 edo gehiago izan behar du.
iii. **Proba bateragarriak**. **DataAccess**-en aukeratutako metodoaren **kutxa txuriko** (fluxu diagramen grafoa, konplexutasun ziklomatikoa eta proba taulak) eta **kutxa beltzeko** (baliokidetasun klaseak muga balioekin) proba diseinua garatu. Jarraian diseinatutako probak JUnit erabiliz inplementatu bi klase desberdinetan. Proba klaseen izenak metodoaren izena + "DAW" edo "DAB" siglak gehituz izendatuko dira hurrenez-hurren (Abibidez metodoa "createQuestion bada, proba klaseak CreateQuestionDAW.java eta CreateQuestionDAB.java izango dira). Bukatzeko, exekuzio ondoren metodoan aurkitu dituzun akatsak deskribatu. /src/test/java/DataAccessTest.java fitxategian createQuestion metodoaren proba test adibide batzuk aurkitu dezakezu.
iv. **Integrazio probak**. Aurreko metodoari deitzen dion FacadeBL-n dagoen metodoaren **kutxa beltzeko** proba diseinua egin (aldaketak ez badaude aurreko metodoaren diseinua berrerabili dezakezu). Proba klaseen izena metodoaren izena + "MockINT" siglak gehituz izendatuko da (Abibidez metodoa "createQuestion bada, proba klasea createQuestionINT.java izango da). Jarraian diseinatutako probak inplementatu DataAccess klasearen Mock objektu bat erabiliz, hau da, DataAccess-en metodoak akatsik eduki ez balu bezala. Bukatzeko, exekuzio ondoren metodoan aurkitu dituzun akatsak deskribatu. /src/test/java/FacadeMockTest.java fitxategian createQuestion metodoaren proba test adibide batzuk aurkitu dezakezu.

2. Entregatzeko dokumentazioa (PDF dokumentu bat hurrengo informaziorekin)
a) Taldekideak eta proiektua garatzeko behar izan dituzuen ordu kopurua.
b) Aukeratutako metodo bakoitzeko:
   a. Egilea eta Kodea
   b. **Proba bateragarriak**. **DataAccess**-en aukeratutako metodoaren **kutxa txuriko** (fluxu diagramen grafoa, konplexutasun ziklomatikoa eta proba taulak) eta **kutxa beltzeko** (baliokidetasun klaseak muga balioekin) proba diseinua. Aurkitutako akatsak. Probaren exekuzio ondoren, zein proba kasu akatsak aurkitu dituzten, eta akatsaren deskribapena (emandako balioa eta espero zen balioa)
   c. **Integrazio probak**. Aurreko metodoari deitzen dion FacadeBL-n dagoen metodoaren **kutxa beltzeko** proba diseinua. Aurkitutako akatsak. Probaren exekuzio ondoren, zein proba kasu akatsak aurkitu dituzten, eta akatsaren deskribapena (emandako balioa eta espero zen balioa)
c) Github eta sonarcloud.io proiektuaren esteka.

3. Balorazioa: %1,25 (diseinu eta inplementazioa) + %0,25 kode estatikoen analisia. Balorazio berdina taldekide guztientzat.
Entrega data: Urriak 3
Estimatutako denbora kide bakoitzeko: 15 ordu.