

Test i prova pilot d'una web de correcció automàtica de pràctiques de Compiladors

Marc Morales

Resum— L'aplicació per a la correcció automàtica de pràctiques de l'assignatura de Compiladors va ser desenvolupada per Christian García Miñano [1]. És una eina de suport via web que principalment ajuda a corregir de forma automàtica pràctiques, treballs i exàmens de qualsevol assignatura. Com a altres funcionalitats, també permet fer altes, baixes i modificacions de professors, assignatures i alumnes, així com l'execució de scripts que corregeixen el material entregat.

Aquest projecte sorgeix de la necessitat per part del professor Javier Sánchez Pujadas (professor de l'assignatura de compiladors) de fer un testeig intensiu d'aquesta aplicació per assegurar que es cobreixen els aspectes més importants d'un bon software (robustesa, fiabilitat i qualitat). Això permetrà que, una vegada s'hagi implantat en un entorn real, el personal docent de l'assignatura de Compiladors pugui estalviar-se temps en la correcció de pràctiques i l'alumnat pugui gaudir d'una nova eina, garantint en tot moment una aplicació segura, fiable i tolerant a fallades.

Paraules clau— Cosel, Scrum, TFG, Compiladors, Correcció automàtica, Scripts, Informes, Plataforma web, Spring, MVC.

Abstract— The application for the automatic correction of practices of Compiler subject is developed by Christian García Miñano [1]. It is a support tool via web that mainly helps to correct practices, works and exams from any subject automatically. As other features, it also allows additions, deletions and modifications of teachers, students and subjects, as well as the execution of scripts that correct material delivered.

This project emerges from the need of teacher Javier Sánchez Pujadas (teacher of the Compilers subject) to do an intensive testing of this application to ensure that it covers the most important aspects of a good software (robustness, reliability and quality). This will allow, once it has been implanted in a real environment, the teaching staff of Compilers subject to save time in correcting practices and students will be able to enjoy a new tool, ensuring at all times a safe, reliable and fault tolerant application.

Index Terms— Cosel, Scrum, TFG, Compilers, Automatic correction, Scripts, Reports, Web platform, Spring, MVC.



1 INTRODUCCIÓ

ACTUALMENT, la correcció de pràctiques es resol mitjançant uns scripts en llenguatge Cosel que realitzen diferents tests i proporcionen els resultats al professor. La gestió i manteniment d'aquests scripts s'està tornant pesada i neix la necessitat d'automatitzar aquest treball per tal d'optimitzar el temps invertit per part del professorat i facilitar als alumnes una eina de autoavaluació.

Per cobrir aquesta necessitat, es va desenvolupar una aplicació per gestionar les correccions de les pràctiques. Aquesta aplicació però, estava en una fase de desenvolupament inicial producte de l'anterior TFG i per tant es desconeixia si l'aplicació era suficientment robusta per poder ser usada en un entorn real i per assegurar una certa tolerància a errors.

Aquest projecte, tracta precisament d'assegurar que aquesta aplicació compleix amb els nivells de qualitat, fiabilitat i robustesa esperats. Per aconseguir-ho s'ha fet ús de diferents proves de software que validen aquestes necessitats.

El present document es divideix en les següents seccions:

- **Introducció:** Introdueix al lector sobre el projecte desenvolupat i es detallen els diferents punts de l'article.
- **Objectius:** Es descriuen tots els objectius planificats des del començament del projecte fins al seu tancament incloent també una llista de les seves prioritats.
- **Estat de l'art:** Aquest apartat fa un anàlisi de les diferents eines existents en el mercat i com aquestes podrien arribar a fer les mateixes funcions de l'aplicació desenvolupada.
- **Metodologia:** Es descriu quina ha sigut la metodologia escollida pel desenvolupament al llarg del projecte així com la seva justificació.

• E-mail de contacte: marc.morales@e-campus.uab.cat
 • Menció realitzada: Enginyeria del Software.
 • Treball tutoritzat per: Javier Sánchez Pujadas (DCC)
 • Curs 2015/16

- **Desenvolupament:** Es descriu quina ha sigut la planificació global per satisfer els objectius del projecte i l'estat final d'aquests. També es comenta de manera general els problemes que han anat sorgint i com s'han resolt.
- **Test:** Aquest apartat explica quines eines s'han utilitzat per realitzar els tests de l'aplicació, quins tipus de proves s'han fet, quins errors s'han corregit amb aquests tests i per últim quines millores s'han afegit a l'aplicació.
- **Resultats obtinguts:** En aquest apartat, es dona a conèixer quins han sigut els resultats que s'han obtingut al finalitzar el projecte.
- **Conclusions:** S'ofereix una visió global sobre el transcurs del projecte des de l'inici fins la seva fase final, així com propostes de millora cara al futur.
- **Agraïments:** Es realitza un esment a totes les persones que han donat suport d'una manera u altre en el desenvolupament d'aquest projecte.
- **Bibliografia:** En aquest últim apartat es detalla la bibliografia utilitzada en aquest article perquè el lector pugui aprofundir més en els temes que s'han referenciat.

2 OBJECTIUS

Per un correcte desenvolupament de l'aplicació s'han marcat els següents objectius des de l'inici del projecte fins al final, aquests objectius es mostren a la Taula 1 ordenats per la seva prioritat.

Nº	Objectius	Prioritat
1	Instal·lar l'aplicació de correcció de pràctiques en un entorn real amb data límit 2 de Març.	CRÍTIC
2	Implementar la funcionalitat d'informar al alumne sobre la qualificació de les pràctiques.	CRÍTIC
3	Elaborar i executar els tests de correcció d'entregues de les pràctiques dels alumnes.	CRÍTIC
4	Elaborar i executar els tests de tots els mòduls de l'aplicació.	PRIORITARI
5	Elaborar i executar els tests de següent.	PRIORITARI
6	Tests de varis accessos concurrents per trobar erros en l'aplicació.	PRIORITARI
7	Afegir noves funcionalitats com per exemple (estadístiques, entregues de problemes, etc.)	SECUN-CUN-DARI

8	Crear un sistema de verificació de pràctiques copiades.	SECUN-CUN-DARI
---	---	----------------

Taula 1 - Objectius globals del projecte ordenats per prioritat

3 ESTAT DE L'ART

A dia d'avui hi ha una gran varietat de software que dona suport a la gestió d'assignatures, el problema és que les funcionalitats que proporcionen cobreixen aspectes molt generals i per tant no en tenim prou si necessitem característiques més avançades.

L'aplicació d'aquest projecte, cobreix requisits més específics que necessita el client i que les aplicacions actuals no poden satisfer. Per exemple, en el propi campus de la universitat tenim varies eines per a la gestió de material docent i de cursos. Per una part, tenim el PSG [2] que es fa servir per l'entrega telemàtica de pràctiques d'informàtica i per altra banda el campus virtual (CV) [3]. La primera només funciona per assignatures relacionades en l'àmbit d'informàtica, en canvi l'aplicació d'aquest projecte pot donar d'alta diverses assignatures siguin de l'àmbit que siguin. La segona, té bastantes funcionalitats similars al projecte actual en canvi no permet la correcció automatitzada.

També tenim altres eines externes com Coursera [4] o Moodle [5], Coursera tampoc cobreix la necessitat de poder corregir pràctiques a partir d'un llenguatge de programació propi com és Cosel, ja que només ofereix vídeos, qüestionaris i text interactiu pels seus cursos online. En canvi, Moodle sí que pot cobrir totes les necessitats que necessita el client, el problema és que s'hauria de desenvolupar un mòdul específic per aquesta plataforma i això pot comportar problemes de seguretat, per tant s'ha preferit desenvolupar una aplicació independent en lloc de la creació d'aquest mòdul.

Com es pot observar, l'aplicació de l'assignatura de compiladors cobreix aspectes molt específics que en altres plataformes son més difícils de cobrir. Per tant podem dir que (per ara) s'ha desenvolupat una nova eina a mida que actualment pot diferenciar-se d'altres eines existents en el mercat.

4 METODOLOGIA

La metodologia que s'ha utilitzat en aquest projecte és la de desenvolupament àgil basat en alguns aspectes de la metodologia SCRUM [6].

Aquest desenvolupament s'ha realitzat de forma iterativa i incremental. Cada iteració denominada Sprint ha tingut una durada de 4 setmanes que han coincidit amb les entregues importants d'aquest treball (informes de progrés, article, dossier del TFG, etc.), això ha fet possible entregar al client parts funcionals del treball tot i que

globalment encara no s'hagués finalitzat.

A diferència d'SCRUM, les reunions han sigut setmanals (no diàries) on s'ha revisat i comentat l'estat de la feina feta. Per últim, una altra diferència a tindre en compte es que l'equip de desenvolupament no ha estat format per un grup de persones sinó per un únic desenvolupador, en aquest cas l'alumne involucrat en el TFG.

Per tancar aquesta secció, es pot afirmar que aquesta metodologia ha encaixat perfectament amb les necessitats de la planificació del TFG ja que ha permès tindre un seguiment del progrés del projecte i ha aconseguit que el client pugui haver observat aquests avenços.

5 DESENVOLUPAMENT

5.1 PLANIFICACIÓ

La planificació del projecte ha sigut variable en funció de la prioritat dels objectius. La planificació s'ha basat en Sprints que s'han fet coincidir amb en les dates d'entrega pactades en el desenvolupament del TFG.

S'ha de tindre en compte que els tests que s'han realitzat s'han continuat executant-se al llarg de tot el projecte, però només s'han definit en la planificació la seva elaboració inicial.

La planificació global del projecte es desglossa de la següent manera:

Planificació inicial:

Sprint 1 (11/02/2016 - 06/03/2016)

- Reunió inicial amb el tutor.
- Instal·lar l'aplicació de correcció de pràctiques en un entorn real amb data límit 2 de Març (**objectiu 1**).
- Elaboració i prova del test d'anàlisi sintàctic (**objectiu 3**).
- Elaboració informe inicial.
- Lliurament del informe inicial.

Sprint 2 (07/03/2016 - 17/04/2016)

- Elaboració del feedback informatiu de correcció de la pràctica (**objectiu 2**).
- Elaboració i prova del test d'anàlisi semàntic (**objectiu 3**).
- Elaborar els test de tots els mòduls de l'aplicació (**objectiu 4**).
- Elaborar i executar els tests de seguretat (**objectiu 5**).
- Tests de varis accessos concurrents per trobar errors en l'aplicació (**objectiu 6**).
- Elaboració del document de Test.
- Elaboració del informe de progrés I.
- Lliurament informe de progrés I.

Sprint 3 (18/04/2016 - 22/05/2016)

- Elaboració i prova del test de generació de codi (**objectiu 3**).

- Fase inicial del desenvolupament de noves funcionalitats (**objectiu 7**).
- Fase inicial del desenvolupament del sistema de verificació de pràctiques copiades (**objectiu 8**).
- Elaboració del informe de progrés II.
- Lliurament informe de progrés II.

Sprint 4 (23/05/2016 - 07/07/2016)

- Elaboració proposta d'article.
- Elaboració dossier TFG.
- Lliurament proposta article.
- Lliurament article.
- Lliurament dossier TFG.
- Elaboració de la defensa.

5.2 PROBLEMES TROBATS

Al llarg del desenvolupament del projecte, han anat apareixent diversos problemes que han causat retards importants en la planificació això ha provocat que alguns dels objectius s'hagin tingut que tornar a planificar o bé directament descartar-los. A continuació es comenta quins han sigut aquests problemes i quines mesures de prevenció s'han pres.

El primer problema, va sorgir després d'assolir l'objectiu 1 i abans de començar amb l'objectiu 3. Després d'instal·lar l'aplicació en l'entorn real, es va fer una prova inicial per comprovar que les funcionalitats d'aquesta funcionaven correctament sota els paràmetres establerts. El cas és que, després de fer varies proves bàsiques, l'aplicació mostrava diversos problemes greus, provocant que la seva utilització en un entorn real fos irrealitzable.

Aquest fet va provocar que els objectius del projecte es paralitzessin, ja que, fins que les funcionalitats més bàsiques no funcionessin dins d'uns paràmetres d'acceptació, no es va poder continuar amb la realització d'aquests, provocant que la planificació global del projecte es veiés afectada (3 setmanes de retràs).

El segon problema, va ser causat per la excessiva dificultat de generar tests pel Framework Spring MVC [7], ja que la seva configuració era més complicada del que en un principi es pensava. Després de solucionar aquest impediment, va sorgir un altre causat per una incompatibilitat entre les llibreries per fer test JUnit i Hamcrest tal com s'observa a la Figura 1.

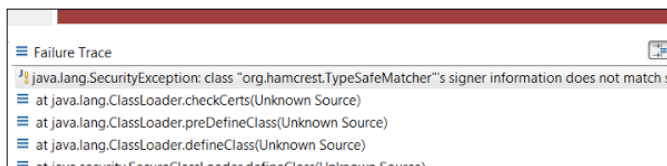


Figura 1 - Conflicte de les llibreries de test JUnit i Hamcrest

L'últim problema ha sigut causat per la inexperiència en el món del testeig, ja que en un principi es pensava que dues setmanes serien suficients per testejar tota

l'aplicació, però degut a la suma dels retards produïts per els problemes anteriors i no haver tingut en compte la gran quantitat de tests ha realitzar, ha provocat que l'objectiu 4 (un dels més importants) no s'hagi pogut completar totalment.

Per tant, a causa d'aquests fets, els objectius 6, 7 i 8 no s'han pogut completar ja que el marge de temps que quedava per la finalització del projecte era massa just per intentar completar-los.

A la Taula 2 es mostra l'estat final dels objectius que s'havien proposat a l'inici del projecte. Podent observar que la majoria han sigut finalitzats amb èxit excepte per l'objectiu 4 que ha faltat poc per ser completat però que compleix (en general) amb els criteris de satisfacció per part del client.

ID	Objectius	% Completat
1	Instal·lar l'aplicació de correcció de pràctiques en un entorn real amb data límit 2 de Març.	100%
2	Implementar la funcionalitat d'informar al alumne sobre la qualificació de les pràctiques.	100%
3	Elaborar i executar els tests de correcció d'entregues de les pràctiques dels alumnes.	100%
4	Elaborar i executar els tests de tots els mòduls de l'aplicació.	80%
5	Elaborar i executar els tests de seguretat.	100%
6	Tests de varis accessos concurrents per trobar errors en l'aplicació.	0%
7	Afegir noves funcionalitats com per exemple (estadístiques, entregues de problemes, etc.)	0%
8	Crear un sistema de verificació de pràctiques copiades.	0%

Taula 2 - Estat final dels objectius del projecte amb el % de completesa

6 TEST

6.1 EINES UTILITZADES

En el transcurs del projecte, s'han anat utilitzant diverses eines per realitzar les proves de software a l'aplicació de gestió de pràctiques i poder validar el seu bon funcionament. Aquestes eines es descriuen a continuació:

- **Spring-test** [8]: És part dels mòduls de Spring per fer test. S'ha utilitzat per fer els "rollbacks" de les proves a base de dades, per carregar els arxius de configura-

ció del context de prova i el més important, per fer peticions personalitzades del tipus GET i POST.

- **Mockito** [9]: Utilitzat com a "mock object" [10] per simular part del comportament de diverses classes (p. ex. la classe de la base de dades, els serveis, fitxers,...)
- **Hamcrest** [11]: S'ha fet servir per escriure els nostres tests amb un llenguatge més semblant al natural de tal manera que ens sigui més fàcil entendre que fa el nostre test.
- **JUnit** [12]: S'ha fet servir per fer els testejos més bàsics i senzills de l'aplicació.
- **GreenMail**: [13] És una suite de codi obert que proporciona SMTP, POP3 i IMAP lleugers i de manera aïllada. S'ha fet servir per fer diferents proves amb l'enviament de e-mails.
- **HyperSQL** [14]: És un sistema de gestió de bases de dades relacionals, està escrit en el llenguatge de programació Java. S'ha utilitzat per fer les proves de base de dades, ja que ofereix la possibilitat de treballar amb les bases de dades en memòria.
- **EclEmma** [15]: És una eina per Eclipse que s'ha utilitzat per visualitzar la cobertura del codi dels tests unitaris que hem fet.

6.2 Tipus de proves

Per tal de validar que l'aplicació de gestió de pràctiques compleix amb uns nivells de qualitat i robustesa esperats s'ha fet ús de les denominades **proves unitàries**. Aquestes proves, s'han utilitzat per validar que cada component de l'aplicació compleix amb el comportament esperat. Per fer-ho, els tests s'han dividit en cinc "mòduls" que cobreixen diferents aspectes molt concrets de l'aplicació i que s'ha trobat oportú tractar-los de manera separada.

Aquests mòduls son:

- **Controladors**: Són totes les classes que faran d'intermediàries entre les vistes i el serveis. També són les encarregades de rebre les peticions GET i POST i de retornar els resultats a la vista.
- **Serveis**: Són aquelles classes cridades pels controladors i que faran crides a les funcions dels repositoris. A més també són les encarregades de preparar els resultats per servir-los al controlador.
- **Repositoris**: Són les classes que accediran a les bases de dades i proporcionaran els resultats al serveis.
- **Validadors**: Els validadors, són les classes encarregades de validar que les dades introduïdes en els formularis de les vistes són correctes i en cas contrari avisar al controlador.
- **Classes bàsiques**: Són classes POJO per ajudar a Hibernate a fer la persistència a base de dades [16].

Gràcies a que es coneix l'estructura interna de l'aplicació (codi), en tots els tests s'han utilitzat **proves de caixa blanca** com a suport a les proves unitàries. Amb aquestes proves, s'ha examinat cada línia de codi dels diferents mòduls.

Per als mòduls "controladors" i "serveis", s'ha fet ús de "mock objects". En els cas dels controladors, s'han utilitzat per simular el comportament dels serveis i en cas dels

serveis s'han fet servir per simular el comportament dels repositoris. A la Figura 2, es pot observar com simulem el comportament de la classe **entityManager** (un servei) a través del mètode **when** perquè retorni un determinat estudiant (Student). Les eines que han estat més presents en aquests mòduls han sigut Mockito i Hamcrest.

```
Student student = getStudent();
when(studentManager.getStudentByNiu(anyString())).thenReturn(student);
// Sin session
mockMvc.perform(get("/delivery/send/{id_work}.html", ""))
.andExpect(status().isOk())
.andExpect(view().name("redirect:/login.html"))
.andExpect(redirectedUrl().contains("/login.html"))
.andExpect(model().size(0));
```

Figura 2 - Simulació d'un servei gràcies a un "mock object"

Per els mòduls que no tenen dependències com els **validadors** i les **classes bàsiques**, s'ha utilitzat de manera general la eina JUnit, ja que els tests a realitzar són més senzills. Més detalladament, els repositoris han fet servir l'eina HyperSQL per poder carregar en memòria una base de dades temporal i realitzar els diferents test. Per altra banda, els validadors han utilitzat en ocasions molt específiques "mock objects", per simular la pujada d'arxius al servidor, tal com es pot observar a la Figura 3.

```
@Test
public void validateOKFile() {
    MultipartFile file = null;
    file = new MockMultipartFile("file", "test.txt", "text/plain", "test".getBytes());
    student.setFile(file);
    ValidationUtils.invokeValidator(validator, student, errors);
    assertEquals(0, errors.getFieldErrorCount("file"));
}
```

Figura 3 - Simulació del comportament d'una pujada d'un arxiu

6.3 Errors corregits

Al llarg del projecte, s'han anat trobant errors ja sigui gràcies als testejos intensius, al feedback proporcionat pels usuaris o bé simplement fent-ne ús de l'aplicació en l'entorn de desenvolupament.

A continuació, es llisten tots els problemes que han anat sorgint i que s'han anat solucionant.

- Al fer una entrega per part de l'alumne, l'aplicació mostra una excepció.
- Quan un professor vol donar d'alta una assignatura, l'enllaç que permet al professor modificar el seu perfil desapareix.
- Si es visita la secció d'assignatures, no es pot accedir a les diferents opcions del menú perquè la capçalera de la pàgina es queda bloquejada.
- Si el professor crea una nova entrega, la pàgina mostra una excepció.
- Si el professor puja el llistat per donar d'alta alumnes nous, la pàgina genera una excepció.
- A vegades al introduir dades (altes, modificacions, validacions, etc.) la pàgina mostra excepcions.
- Si varis usuaris es validen a l'aplicació i un d'ells tanca la seva sessió, la sessió dels altres usuaris també queda tancada.
- Si un alumne que té un e-mail erroni demana les credencials per poder utilitzar l'aplicació, aquest queda bloquejat sense poder entrar.
- Si un alumne entrega una pràctica la qual provoca

que es generi un bucle a l'hora de corregir-la, aquest alumne podrà obtenir informació addicional de la seva correcció quan descarregui el seu informe.

- En general, en tota l'aplicació no es notifica als usuaris sobre accions dutes a terme (p. ex. entregues de pràctiques, enviament de e-mails, modificacions de dades personals, altes d'alumnes, termini d'entrega expirat, etc.).
- Si un alumne modifica el seu perfil amb dades incorrectes, al recarregar la pàgina no es mostren les seves dades.

Els tests que validen aquest elements mencionats anteriorment, es troben en el document de test que s'ha inclòs en l'informe del projecte.

A la Figura 4, s'indica quina ha sigut la cobertura total de línies de codi de l'aplicació. En aquest cas, s'ha cobert 1.262 línies d'un total de 1.618 (aproximadament un 80%).

Element	Coverage	Covered Lines	Missed Lines	Total Lines
src/main/java - corrector	78,0 %	1.262	356	1.618
com.compiladores.corrector.service	6,1 %	23	355	378
com.compiladores.corrector.validator	100,0 %	158	0	158
com.compiladores.corrector.repository	100,0 %	175	0	175
com.compiladores.corrector.domain	99,8 %	415	1	416
com.compiladores.corrector.web	100,0 %	491	0	491

Figura 4 - Cobertura total de l'aplicació

A la Figura 5, es mostra quina és la quantitat total de tests de l'aplicació i el seu temps d'execució (que en aquest cas és aproximadament de mig minut). A l'apèndix d'aquest document es mostra de manera més detallada el total de tests per cada mòdul de l'aplicació.

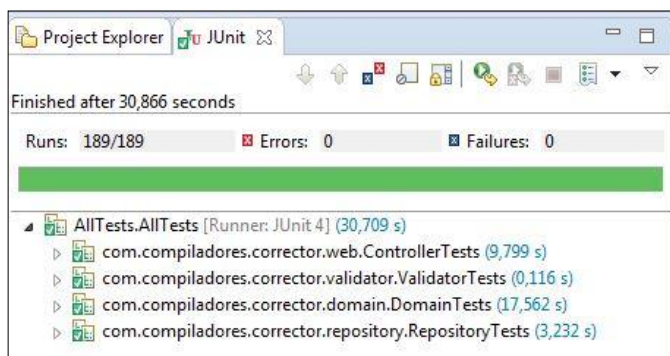


Figura 5 - Tests totals de l'aplicació

6.4 Millores afegides a l'aplicació

A part de corregir els errors trobats al llarg del projecte, també s'ha intentat millorar el codi perquè sigui més fàcil poder fer modificacions cara al futur. Totes les millores que s'han fet fins ara es llisten a continuació:

- S'ha fet servir variables globals per evitar tindre que repetir las variables més utilitzades, com pot ser les que es fan servir per les rutes dels arxius, programes, noms de carpetes, etc.
- S'han "sanejat" totes les consultes de l'aplicació per prevenir atacs d'injecció SQL. A la Figura 6 i a la Figura 7 es pot veure un exemple.

```
public String getPasswordTeacher(String niu) {
    List<Teacher> t = em.createQuery(
        "from Teacher t where t.niu = '" + niu + "'").getResultList();
    if (!t.isEmpty()) {
        return t.get(0).getPassword();
    }
    return null;
}
```

Figura 6 - Consulta SQL sensible a atacs

```
public String getPasswordTeacher(String niu) {
    Query query = em.createQuery("from Teacher t where t.niu = :niu_teacher");
    query.setParameter("niu_teacher", niu);
    List<?> t = query.getResultList();
    return t.isEmpty() ? null : ((Teacher) t.get(0)).getPassword();
}
```

Figura 7 - Consulta SQL sanejada contra atacs d'injecció

- S'ha millorat l'alta d'alumnes, ara la operació es fa de manera transaccional.
- Anteriorment es creaven arxius per lots (.bat) per crear carpetes o per executar les correccions, actualment es fan servir les pròpies llibreries de Java per fer aquestes operacions.
- L'aplicació ja no depèn de si el programa Winrar està instal·lat en el sistema o no, ja que ara s'utilitzen llibreries pròpies de Java per la descompressió d'arxius.
- Es millora el codi per la correcció de pràctiques fent-lo més personalitzat, per exemple podent canviar el temps màxim d'execució, la ruta d'execució, llegir la sortida de la correcció, etc.
- S'ofereixen missatges informatius al professor si no té assignatures assignades.
- S'agafa la ruta d'instal·lació del programa Cosel directament de les variables de sistema, així es pot saber si el programa està instal·lat en el sistema o no, tal com s'observa a la Figura 8.

```
private boolean executeCorrection(Path path file, Path path entregues) {
    String installation_cosel = System.getenv("CrossVisions");
    if (installation_cosel == null) {
        logger.info("Puede que COSEL no esté instalado");
        return false;
    }
    Path path_cosel = Paths.get(installation_cosel, "Bin", "CVCon.exe");
}
```

Figura 8 - Obtenció de la ruta d'instal·lació del programa Cosel mitjançant variables del sistema

- Ja no s'utilitza la classe singleton "HttpSessionControl" per fer el control de les sessions, en comptes d'això s'utilitzen les sessions del servidor per emmagatzemar la informació bàsica de cada usuari. Això soluciona el problema que un usuari pogués tancar les sessions de tothom. A més amb aquest mètode, el codi per gestionar les sessions és molt més fàcil i clar.
- Es modifica el codi de l'entrega de pràctiques perquè sigui més fàcil de mantenir i d'entendre.
- Totes les pàgines de l'aplicació redirigeixen a la pantalla de login si no es té permís.
- Algunes pàgines de l'aplicació només són accessibles per determinats tipus de rols (professors, alumnes).
- Es modifica la classe "Subject" perquè es pugui comparar amb altres assignatures (equals, contains, etc.).
- S'ha implementat la funcionalitat de descarregar un tipus o un altre d'informe depenent si l'entrega de la pràctica ha expirat o no. També s'ha implementat la

possibilitat de poder descarregar els arxius de les pràctiques que ha penjat un alumne.

- S'ha corregit el format de fitxer que es fa servir per donar d'alta a alumnes per una nova versió amb format diferent.
- S'eliminen classes i fitxers que no s'utilitzaven.
- Es milloren els validadors a nivell general, per exemple que els NIUs tinguin el format correcte, que els arxius que es pugen al servidor siguin vàlids, etc.).
- S'ha millorat l'arxiu de missatges de l'aplicació dividint els missatges per cada mòdul.
- Es desacobla el codi de la classe TeacherController perquè la feina de creació d'estudiants passi a la classe CreateStudents. També es fa el mateix per a la classe DeliveryController perquè la feina de gestió d'entregues recaigui en una nova classe.
- En general, s'han implementat missatges que informen al usuari sobre les interaccions que es duen a terme en l'aplicació (ja que en la versió inicial no s'oferia aquest servei i es necessitava tindre un feedback per saber si les operacions dutes a terme eren satisfactòries o no).

7 RESULTATS OBTINGUTS

Al llarg de tot aquest projecte, s'han obtingut els següents resultats:

- S'ha fet una instal·lació de l'aplicació en un entorn real i totalment funcional on alumnes i professors poden fer-ne ús a través de la direcció <http://compiladors.uab.es/>. Això ha estalviat temps al personal docent i ha ofert un nou servei als alumnes per poder corregir les serves pràctiques.
- S'han creat els tests de controladors, validadors, accés a base de dades i de classes que validen el correcte funcionament de l'aplicació.
- S'han generat tests de seguretat que podrien comprometre el funcionament de l'aplicació. Aquests tests han tingut alguns aspectes de seguretat com: injecció SQL, restricció de pàgines depenent del rol alumne/professor, accés a pràctiques d'altres alumnes, etc.
- S'ha millorat part del codi de l'aplicació pensant en futurs desenvolupadors que repleguin el projecte i per tant els sigui més fàcil d'implementar noves funcionalitats
- També s'ha obtingut una millora significativa en els resultats acadèmics de l'assignatura. Això és degut a la funcionalitat que ofereix l'aplicació per corregir les pràctiques tantes vegades com es vulgui. Aquest fet motiva a l'alumne a intentar superar els seus resultats obtinguts.

A la Figura 9, s'observa que hi ha hagut un increment en les notes de la practica de generació de codi si les comparem amb la primera nota i l'última. A més, a la Figura 10 també es pot veure quin ha sigut el % de millora de cada alumne respecte la primera pràctica que van entregar. En general, la pràctica a tingut una millora al voltant del 24%.

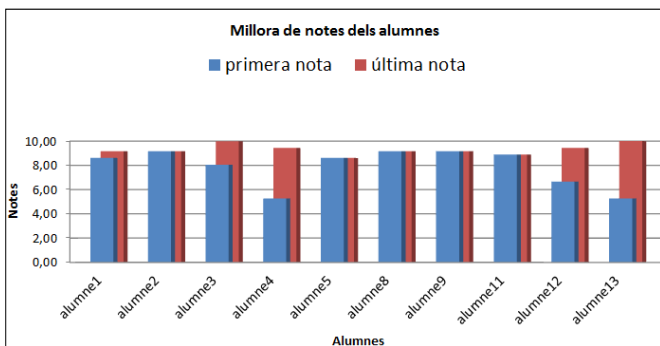


Figura 9 - Millora dels alumnes entre la primera nota de pràctiques i l'última.

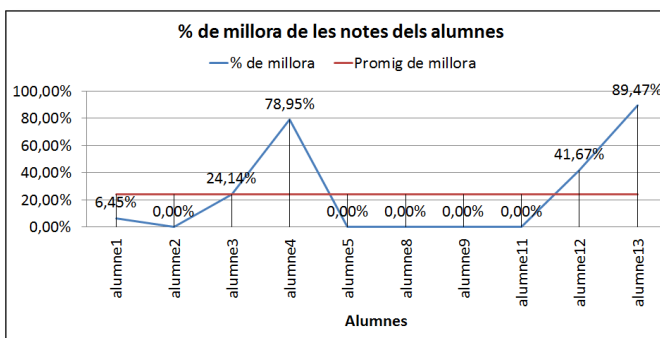


Figura 10 - % de millora de cada alumne respecte la primera nota de la pràctica i l'última.

Cal destacar que, globalment, la nota de pràctiques respecte curs passat també ha millorat gràcies a l'ús de l'aplicació. La nota mitjana del curs anterior (2014-2015) va ser d'un 9,21 i la del curs actual ha sigut 9,59. Això ha suposat una millora d'un 4,13% respecte el curs anterior, fet que es tradueix en una millora significativa en resultats acadèmics dels alumnes de l'assignatura de compiladors.

7.1 PROVA PILOT

S'ha fet una prova pilot per testejar en una situació real que la correcció de les pràctiques funciona correctament, amb alumnes i professors fent-ne ús de l'aplicació durant un mes (donant alta a alumnes, creant assignatures, corregint pràctiques, etc.). La pràctica que s'ha utilitzat a sigut la de generació de codi, ja que aquesta és acumulativa amb pràctiques anteriors (pràctica de sintàctic i semàntic) i això ha permès validar que la correcció de totes les pràctiques funcionen correctament.

Aquesta prova va constar dels següents passos:

- 1- El professor va crear una nova assignatura en el sistema i seguidament va pujar l'arxiu d'alumnes per donar-los d'alta.
- 2- Posteriorment va crear un nou treball, que en aquest cas va ser una pràctica de generació de codi i seguidament va pujar el script que feia la correcció automàtica.
- 3- Un cop acabat la segona etapa del procés, els alumnes van tindre que demanar les seves cre-

dencials per poder-los donar d'alta en el sistema.

- 4- Finalment, una vegada introduïdes les seves credencials els alumnes van poder començar a penjar les seves pràctiques, podent descarregar posteriorment el seu informe de correcció.

D'un total de 57 alumnes, 22 han utilitzat l'aplicació. Això significa que prop d'un 39% han aprofitat la funcionalitat que ofereix l'aplicació per corregir les seves pràctiques. A la Figura 11, es pot observar que en general s'han fet una mitja de 8 entregues per cada alumne, per tant es pot concloure que la prova pilot va ser un èxit.

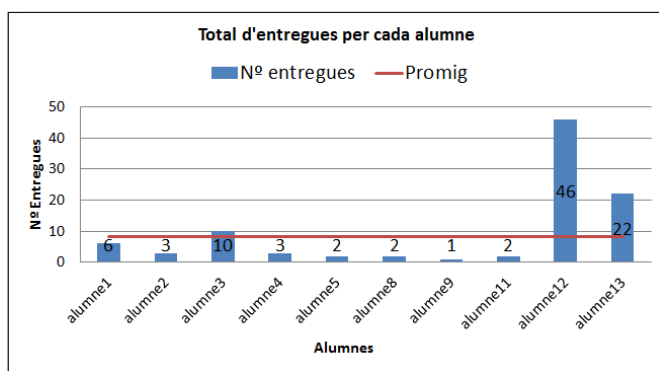


Figura 11 - Total d'entregues de cada alumne en la pràctica de generació de codi.

8 CONCLUSIONS

Com a conclusió, es pot dir que reprendre el desenvolupament d'una aplicació ja creada pot comportar molts problemes dels quals si no es preveuen amb antelació pot implicar retards importants en la planificació global del projecte. També s'ha pogut comprovar que si hagués hagut un testeig mínim previ de l'aplicació, molts dels problemes que es van trobar al principi s'haurien cobert i el temps dedicat a resoldre'ls hauria disminuït dràsticament.

Finalment, cal a dir que no s'esperaven tants problemes al llarg del projecte sobretot per temes de mala programació i falta de testeig, tot i així en termes generals el treball realitzat en aquest projecte ha sigut satisfactori, ja que s'han cobert les parts més importants que el client necessitava i a més hi ha hagut una millora tangible de l'aplicació fet que ha comportat una experiència més agradable pels usuaris finals.

8.1 PROPOSTES DE MILLORA

Es proposen les següents opcions de millora en futures implementacions:

- Completar els objectius que no s'hagin pogut finalitzar en la fase final del projecte.
- Utilitzar el motor de plantilles Thymeleaf [17] en comptes de JSP [18], això farà que el disseny de les vistes sigui més elegant i el codi sigui menys difícil de llegir.
- Millorar el codi intern de la pàgina, intentant fer ús

de bones pràctiques de programació (es recomana fer ús d'alguna eina d'anàlisi de codi [19]).

- Utilitzar paginació en llistats massa llargs, per exemple quan el professor llista les pràctiques entregades, les notificacions, etc.
- Fer servir el format de classe Java en comptes d'XML per a les diverses configuracions. Fer-ho suposa tindre un codi més organitzat i no tindre tants arxius de configuració.
- Comprovar que quan s'envii un e-mail aquest existeixi, ja que si no existeix l'usuari quedarà bloquejat.
- Implementar un controlador que gestioni les excepcions [20] en comptes d'utilitzar "try-catch" en tots els mètodes de l'aplicació.
- Implementar una pàgina d'error 404 millor que l'actual (més informativa, amb millor disseny, etc.)
- Fer que les entregues dels alumnes siguin successives i que no se sobreescriuin.
- Quan es crea un treball per part del professor, que es pugui introduir el temps d'execució que necessita aquest treball.
- No només implementar noves funcions de seguretat en l'aplicació sinó també en el servidor on s'allotja la web.
- Fer tests automàtics de les vistes HTML en comptes de fer-ho manualment.

AGRAÏMENTS

Voldria donar les gràcies a tota aquella gent que ha sigut un suport en algun moment en el transcurs d'aquest projecte de final de grau. Primerament, agrair al meu tutor Javier Sánchez Pujadas la possibilitat de participar en el seu projecte, al tècnic de sistemes Jorge Ramírez Espejo per proporcionar la infraestructura necessària per posar en marxa l'aplicació en l'entorn real i a Christian García Miñano per fer un gran treball en la documentació de l'aplicació i aconseguir que reprenere el projecte de nou hagi sigut menys costós.

Per últim i no menys important, aquelles persones que m'han donat suport en els moments més difícils d'aquest projecte i que sense elles aquest projecte hauria sigut més difícil de portar a terme. Aquestes persones han sigut la meua parella, els meus amics i per descomptat la meua família.

BIBLIOGRAFIA

- [1] C. García Miñano, «Estudio, diseño e implementación de una web de corrección automática de prácticas y trabajos la asignatura Compiladores,» 2015. [En línia]. Disponible a: https://ddd.uab.cat/pub/tfg/2015/tfg_27911/Articulo.pdf. [Últim accés: 24 Febrer 2016].
- [2] F. d. Medicina, «PSG,» [En línia]. Disponible a: <http://clon.uab.es/psg/>. [Últim accés: 17 Juny 2016].
- [3] «Campus Virtual de la UAB,» [En línia]. Disponible a: <https://cv.uab.cat/portada/index.html>. [Últim accés: 17 Juny

- 2016].
- [4] «Coursera,» [En línia]. Disponible a: <https://es.coursera.org/>. [Últim accés: 17 Juny 2016].
- [5] «Moodle,» [En línia]. Disponible a: <https://moodle.org/?lang=ca>. [Últim accés: 17 Juny 2016].
- [6] L. Gómez Sanchez, «Metodología Scrum para desarrollo de software - aplicaciones complejas,» [En línia]. Disponible a: <https://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-scrum.html>. [Últim accés: 24 Febrer 2016].
- [7] «Spring Framework Reference Documentation - Testing,» [En línia]. Disponible a: <http://docs.spring.io/autorepo/docs/spring-framework/current/spring-framework-reference/html/testing.html>. [Últim accés: 13 Maig 2016].
- [8] «Spring Framework Reference Documentation - Integration Testing,» [En línia]. Disponible a: <http://docs.spring.io/spring/docs/current/spring-framework-reference/html/integration-testing.html>. [Últim accés: 03 Juny 2016].
- [9] «Mockito,» [En línia]. Disponible a: <http://mockito.org/>. [Últim accés: 13 Maig 2016].
- [10] Wikipedia, «Wikipedia,» [En línia]. Disponible a: https://es.wikipedia.org/wiki/Objeto_simulado. [Últim accés: 03 Juny 2016].
- [11] «Java Hamcrest,» [En línia]. Disponible a: <http://hamcrest.org/JavaHamcrest/>. [Últim accés: 03 Juny 2016].
- [12] «JUnit,» [En línia]. Disponible a: <http://junit.org/junit4/>. [Últim accés: 13 Maig 2016].
- [13] I. Technologies, «GreenMail,» [En línia]. Disponible a: <http://www.icegreen.com/greenmail/>. [Últim accés: 03 Juny 2016].
- [14] T. H. D. Group, «HyperSQL,» [En línia]. Disponible a: <http://hsqldb.org/>. [Últim accés: 03 Juny 2016].
- [15] M. G. & C. K. a. Contributors, «EclEmma,» 27 Setembre 2012. [En línia]. Disponible a: <http://eclEmma.org/>. [Últim accés: 03 Juny 2016].
- [16] «Hibernate ORM documentation,» [En línia]. Disponible a: <https://docs.jboss.org/hibernate/orm/3.6/reference/es-ES/html/persistent-classes.html>. [Últim accés: 04 Juny 2016].
- [17] «Thymeleaf,» [En línia]. Disponible a: <http://www.thymeleaf.org/>. [Últim accés: 13 Maig 2016].
- [18] O. Corporation, «JSP Standard Tag Library,» [En línia]. Disponible a: <https://jstl.java.net/>. [Últim accés: 03 Març 2016].
- [19] «Eclipse Checkstyle Plugin,» [En línia]. Disponible a: <http://eclipse-cs.sourceforge.net/#/>. [Últim accés: 13 Maig 2016].
- [20] «Spring Framework Reference Documentation - Handling exceptions,» [En línia]. Disponible a: <http://docs.spring.io/autorepo/docs/spring/current/spring-framework-reference/html/mvc.html#mvc-exceptionhandlers>. [Últim accés: 16 Maig 2016].

APÈNDIX

A1. N° TOTAL DE TESTS PER CADA MÒDUL DE L'APLICACIÓ

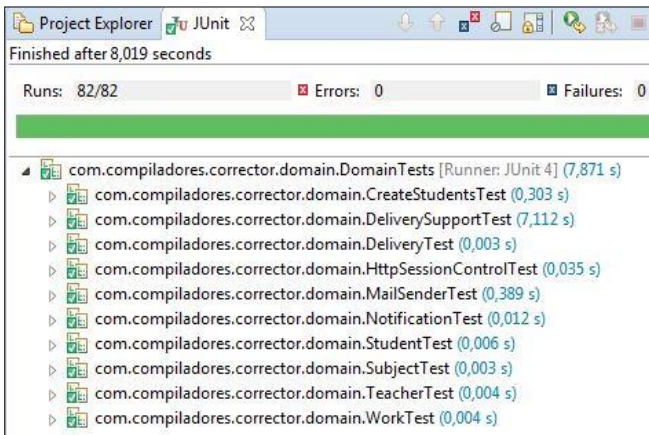


Figura 12 - Tests totals del mòdul "classes POJO" i el seu temps d'execució.

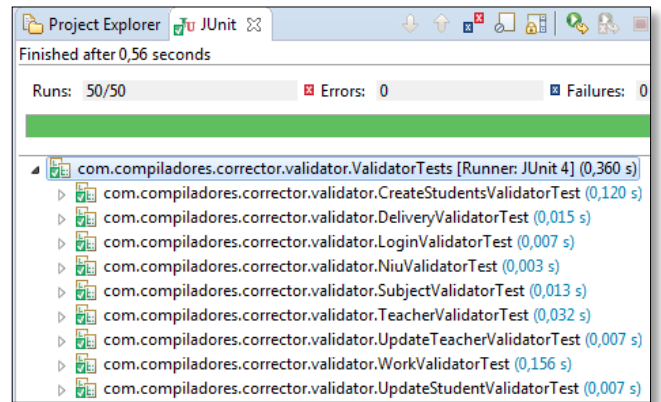


Figura 14 - Tests totals del mòdul "Validadors" i el seu temps d'execució.

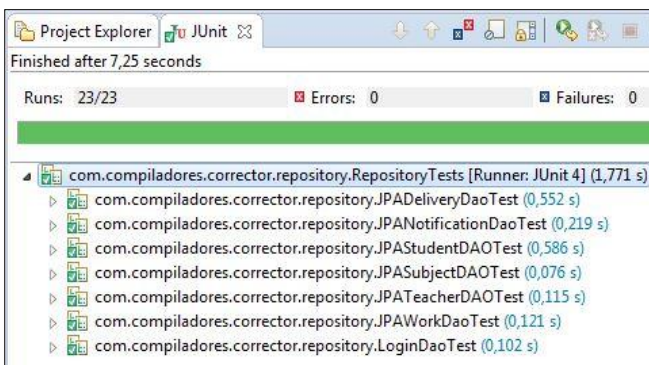


Figura 13 - Tests totals del mòdul "Repositoris" i el seu temps d'execució.

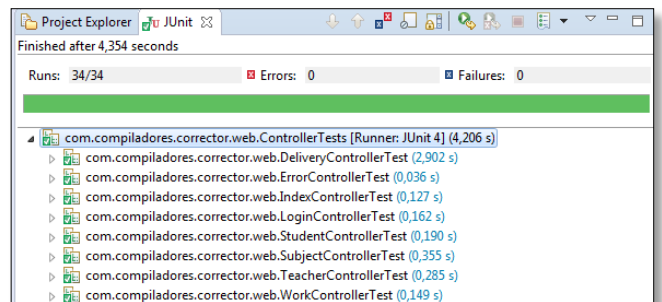


Figura 15 - Tests totals del mòdul "Controladors" i el seu temps d'execució.

A2. MILLORES EN EL CODI

```

public Student LoginStudent(String niu) {
    List<Student> st = em.createQuery(
        "from Student s where s.niu = '" + niu + "' AND s.active= 1")
        .getResultList();
    if (!st.isEmpty()) {
        for (Student s : st) {
            this.student.setId(s.getId());
            this.student.setEmail(s.getEmail());
            this.student.setName(s.getName());
            this.student.setSurname(s.getSurname());
            this.student.setNiu(niu);
            this.student.setPhoto(s.getPhoto());
        }
        return student;
    }
    return null;
}
+ Query query = em.createQuery("from Student s where s.niu = :niu_student and s.active=:activo");
+ query.setParameter("niu_student", niu);
+ query.setParameter("activo", 1);
+ List<?> s = query.getResultList();
+ return (Student) (s.isEmpty() ? null : s.get(0));
}

```

Figura 16 - Previsió d'atacs d'injecció SQL i millora de codi.

```

@RequestMapping(value = "/singoff.html", method = RequestMethod.GET)
public String singoff(HttpServletRequest req) {
    if (req.getSession().getAttribute("Student") != null) {
        sc.setStudent(null);
    } else if (req.getSession().getAttribute("Teacher") != null) {
        sc.setTeacher(null);
    }
    req.getSession().invalidate();
    sc.setLogin(false);
    return "redirect:/login.html";
}

```

↓

```

@RequestMapping(value = "/singoff.html", method = RequestMethod.GET)
public String singoff(HttpServletRequest req, HttpSession session) {
    if (sc.isLogin(session)) {
        session.invalidate();
    }
    return "redirect:/login.html";
}

```

Figura 17 - Millora de codi en el mètode per tancar la sessió.

A3. CANVIS VISUALS

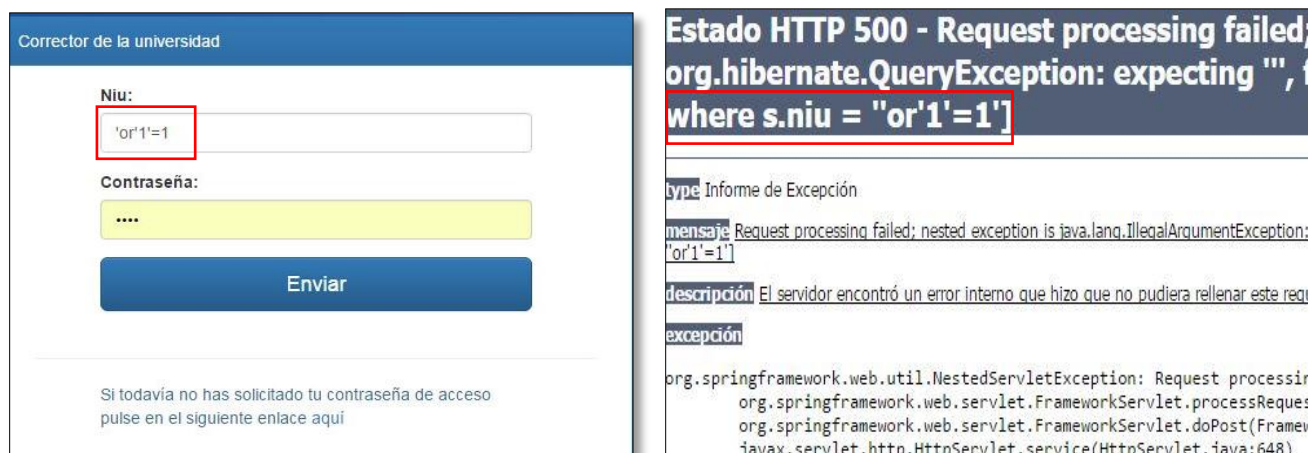


Figura 18 - Error d'injecció SQL a la pantalla de login de la versió original.

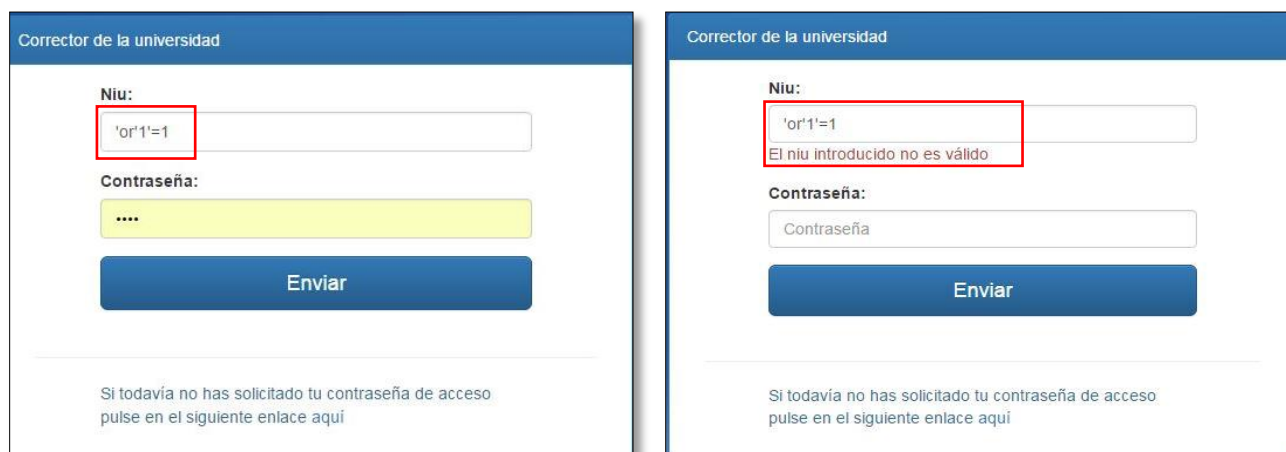


Figura 19 - Missatge informatiu a la pantalla de login quan s'intenta fer un atac d'injecció SLQ a la versió final.

Información entrega

Información asignatura
Nombre: Compiladors 2016
Código: 12345

Información práctica
Nombre: GenCod
Fecha de entrega: 2016-06-30 22:00:00.0

Información última entrega

Ya has realizado una entrega anterior, Si añades una nueva entrega esta quedará eliminada. ✕

Entrega: Script

Archivo: profesores.sql

Enviar

Información entrega

Información asignatura
Nombre: Compiladors 2016
Código: 12345

Información práctica
Nombre: GenCod
Fecha de entrega: 2016-06-30 22:00:00.0

Información última entrega

Archivo: Ningún archivo seleccionado

Enviar

Figura 20 - Entrega d'una pràctica amb una extensió incorrecte sense informar de l'error a la versió original.

Información entrega

Información asignatura
Nombre: Compiladors 2016
Código: 12345

Información práctica
Nombre: GenCod
Fecha de entrega: 2016-06-30 22:00:00.0

Información última entrega
Informe: Informe
Entrega: Script

Archivo: profesores.sql

Enviar

Ya has realizado una entrega anterior, si añades una nueva entrega esta quedará eliminada. ✕

Información entrega

Información asignatura
Nombre: Compiladors 2016
Código: 12345

Información práctica
Nombre: GenCod
Fecha de entrega: 2016-06-30 22:00:00.0

Información última entrega
Informe: Informe
Entrega: Script

Archivo: Ningún archivo seleccionado
El fichero que intentas subir al sistema no es del tipo csl

Enviar

Figura 21 - Missatge informatiu al usuari sobre el format correcte de la pràctica a la versió final.