

# Generació automàtica via web de qüestionaris Moodle

Daniel Vilà Esteban

**Resum–** Aquest paper conté explicacions i procediments exemplificats sobre com s'ha dut a terme i s'ha solucionat la problemàtica que l'eina de treball Moodle suposava. Aquesta eina oferia una dificultat a l'hora de crear preguntes per fer qüestionaris online d'una manera intuïtiva i ràpida. S'ha solucionat amb la creació d'aquesta pàgina web per generar qüestionaris automàticament, fàcilment importables a l'eina de treball Moodle. D'aquesta manera s'aconsegueix una major eficiència i facilitat a l'hora de crear qüestionaris en format xml. També s'expliquen altres millores com la importació de fitxers en format xlsx. A més de les millores, també s'expliquen tots els processos i funcionalitats implementats, eines utilitzades i mesures de seguretat establertes.

**Paraules clau–** Flask, Moodle, Python, Qüestionari, Sprint, CSRF, Jinja2, Bootstrap, Session, JavaScript, HTML.

**Abstract–** This paper contains explanations and exemplified procedures on how it has been carried out and solved the problems that the Moodle work tool was supposed. This tool offers a difficulty in creating questions to make online questionnaires in a intuitive and fast way. It has been solved with the creation of this web page to automatically generate questionnaires and import it into the Moodle work tool easily. In this way, it is achieved greater efficiency and ease when creating questionnaires in xml format. Other improvements are also explained, such as importing files in xlsx format. In addition to the improvements, There is also explained all the processes and functionalities implemented, tools used and security measures established.

**Keywords–** Flask, Moodle, Python, Questionnaire, Sprint, CSRF, Jinja2, Bootstrap, Session, JavaScript, HTML.

## 1 INTRODUCCIÓ

AQUEST projecte consisteix a implementar una pàgina web que sigui capaç de generar qüestionaris amb diferent tipus de preguntes, per poder-los importar a l'eina de treball Moodle [8]. L'objectiu principal és millorar la generació de preguntes de tipus Moodle que ofereix aquest mateix. La generació de preguntes al Moodle, resulta costosa en temps i oposa una certa dificultat a l'hora de generar-les, per tant s'ha volgut crear aquesta pàgina web, prenent millorar les funcionalitats que ofe-

reix el Moodle i afegir-ne de noves. L'objectiu, doncs, és crear una pàgina web que englobi les funcionalitats que el Moodle ofereix, afegint de noves, com la importació de diferents tipus de fitxers, la implementació dels tipus més bàsics de preguntes Moodle d'una manera senzilla i ràpida, la creació d'un fitxer exportable compatible amb l'eina de treball Moodle. A més, ha de ser una aplicació web segura.

En aquest document s'explicaran les raons per les quals s'ha volgut optar per aquest projecte, quina ha sigut la necessitat de realitzar-lo, a més de la metodologia i desenvolupament portats a terme per a la realització d'aquest. També s'explicaran els processos de validació duts a terme i quins han sigut els resultats obtinguts en aquest projecte, per tal de deixar ben clarificades totes les funcionalitats disponibles a la pàgina web. També s'explicarà quines han estat les eines i llibreries utilitzades per aconseguir les funcionalitats esmentades anteriorment. D'altra banda s'aportaran alguns exemples per complementar l'explicació. Finalment, es do-

---

- E-mail de contacte: daniel.vilae@e-campus.uab.cat
- Menció realitzada: Tecnologies de la Informació
- Treball tutoritzat per: Cristina Fernández Córdoba (Dept. of Information and Communications Engineering)
- Curs 2017/18

narà una conclusió i línies futures de treball.

Les parts d'aquest article seran, primerament, una posada en escena, d'on sorgeix la necessitat de crear aquest projecte, qui el requereix i perquè a la Secció 2. Estat de l'art. Seguidament, a la Secció 3. Metodologia de treball, s'explica sobre quines metodologies de treball s'han adoptat de cara a afrontar aquest projecte, perquè s'han escollit i de quina manera han sigut adaptades si ha sorgit la necessitat, després, s'explicarà la Secció 4. Desenvolupament de la pàgina web, on es veurà quines característiques i funcionalitats té la pàgina web, dintre d'aquesta secció tindrem, la subsecció de l'explicació de les funcionalitats bàsiques, la subsecció del tipus de preguntes implementats i, per últim, la subsecció de les mesures de control i seguretat establertes per tal de fer de la pàgina web, robusta i segura. A la Secció de Validació, la número 5, es mostrarà quines proves s'han dut a terme per demostrar que totes les funcionalitats esmentades amb anterioritat funcionen correctament. A la Secció 6, Resultats, s'explicaran aquests resultats el més detallat possible, amb gràfiques comparatives entre el Moodle i la pàgina web, l'abast del projecte i com s'ha arribat a aquests resultats. Més endavant, s'explicaran també la Secció 8, respecte a les possibles línies futures del projecte i per acabar les seccions de conclusions i agraïments.

## 2 ESTAT DE L'ART

Amb l'objectiu d'avaluar l'alumnat via online, mitjançant qüestionaris amb autocorrecció i feedback instantani, el professorat pot optar per utilitzar l'eina Moodle [8], d'aquí sorgeix la necessitat del projecte proposat, ja que amb el Moodle, és molt difícil i costós en temps per crear i generar moltes preguntes d'un mateix tipus de cop. Una altra dificultat és la necessitat de crear preguntes per als qüestionaris amb paràmetres per tal de generar la mateixa pregunta, però amb la variació d'aquests paràmetres introduïts. En l'avaluació basada en qüestionaris, és important el fet que hi hagi un volum molt gran de preguntes. Si és possible, és interessant poder generar moltes versions d'una mateixa pregunta (preguntes similars amb petites variacions) de manera que a l'hora de fer un qüestionari es pugui escollir una pregunta de manera aleatòria entre les diferents versions d'aquesta, per tal que cada estudiant obtingui una versió diferent de la mateixa pregunta quan accedeix al seu qüestionari. Per exemple, si es vol afegir una pregunta de càlcul, seria interessant generar diverses preguntes diferents en funció d'un o més paràmetres. D'aquesta manera, cada estudiant tindrà al seu qüestionari valors de càlcul diferent i, a més, hi ha l'opció de poder repetir el qüestionari amb valors de càlcul diferent per poder practicar més.

En grups grans, aproximadament 250 alumnes o més, per tal de poder minimitzar la probabilitat que alumnes diferents tinguin les mateixes preguntes (o que s'obtingui exactament la mateixa pregunta si es repeteix el qüestionari), caldria generar un mínim de 100 preguntes diferents del mateix tipus. Afegir una pregunta al banc de preguntes Moodle no és ràpid. Si per a cada pregunta que es vol afegir al qüestionari, s'han d'introduir 100 variacions diferents, aquest procés es torna realment molt tediós i poc eficient. Dins de la plataforma Moodle sí que hi ha l'opció de generar preguntes amb paràmetres però el seu funcionament és molt limitat i dona alguns problemes quan s'intenten fer servir

més d'un paràmetre. Si bé és cert que el Moodle et proveeix aquesta funcionalitat, no és gaire útil un paràmetre, a l'hora d'afegir-ne més el cost en temps i dificultat sofreix un gran augment per al professorat.

A partir de voler generar els qüestionaris online per rebre feedback immediat i altres beneficis que proporciona, va sorgir la necessitat de generar preguntes automàticament, per això es va decidir aquest projecte, per solucionar aquests inconvenients que el Moodle suposa i poder generar totes les preguntes de diferents tipus, amb el nombre de paràmetres desitjat i una gran quantitat de preguntes d'un mateix tipus, en un qüestionari fàcilment importable al Moodle.

## 3 METODOLOGIA DE TREBALL

Els passos principals del treball han sigut, la creació d'un *paper prototyping*, amb l'objectiu d'aconseguir una visió més propera i exacta al projecte, però sense perdre massa temps amb el prototip, trobant un equilibri entre afinitat i temps, això ha permès comprendre millor el que es demanava abans de començar i evitar la realització de treball extra. El següent pas ha sigut crear la pàgina web generant qüestionaris amb un únic tipus de pregunta, per tal de començar amb la part lògica de la web, això amb l'eina sol·licitada, Flask [1], un microframework de *Python* per a la creació de pàgines web i aplicacions. El següent pas va ser la implementació de les funcionalitats d'importar fitxers per a processar a l'aplicació web preguntes ja creades. A més, d'un exportable en format xml, amb totes les preguntes generades a la pàgina web, amb aquest exportable es pot importar al Moodle per tenir més preguntes al banc de preguntes del Moodle i així tenir més opcions per crear nous qüestionaris amb les preguntes del Moodle. I per últim, ampliar la pàgina web amb distintes funcionalitats útils, com esborrar i afegir moltes preguntes de cop alterant els resultats per a cada pregunta, diferents tipus de preguntes que l'eina Moodle accepta i la importació d'altres fitxers amb distintes extensions, per tal de poder recuperar qüestionaris fets anteriorment en fulls excel. A més de tot això, també s'ha tingut molt en compte la seguretat de la pàgina web, prevenint cert tipus d'atacs, entre altres mesures de seguretat.

En aquest projecte s'ha adoptat una metodologia de treball basada en SCRUM [2], que consisteix a dividir el projecte en *sprints*, des de l'inici fins a la finalització del projecte, cada *sprint* està format per distintes etapes. A grans trets serien les següents. Inicialment, una reunió amb la persona sol·licitant, per tal de comprendre que és el que es demana exactament, pressupost, flexibilitat relacionada amb el producte i finalment per decidir que es farà durant aquest *sprint*. Una segona fase és la de desenvolupament, en la qual es duu a terme tota la implementació de funcionalitats del projecte. En el nostre cas, el desenvolupament de la pàgina web i totes les funcionalitats que veurem en apartats posteriors. Per últim, una fase de revisió, en la qual s'ha anat mostrant el que s'havia aconseguit al final de cada *sprint*, es mencionaven les problemàtiques sorgides durant la realització del projecte, possibles retards respecte a la planificació inicial del projecte, a més del feedback sobre el treball efectuat fins al moment per part de la persona sol·licitant. Gràcies a aquesta metodologia, és podem detectar implementacions no requerides, errors tant en la planifi-

cació com en el mateix projecte en un espai de temps menys costós, abans de l'hora d'entrega del projecte. Segueix una estructura cíclica, el que provoca un cost reduït de temps i recursos, ja que, amb aquesta revisió cada cert temps, ens permet trobar abans els errors de planificació o implementació, i així corregir-los sense perdre més temps en allò que no s'ajusta a la demanda de la persona sol·licitant, a nivell visual o de funcionalitat.

A més d'aquesta metodologia adaptada al nostre projecte, s'ha treballat amb una altra metodologia addicional per tal de poder realitzar un desenvolupament del projecte el més robust possible. Consisteix a fer servir *Exploratory testing* [3] el qual consisteix a adoptar una mentalitat destructiva amb el codi. Es tracta d'aconseguir que el codi deixi de funcionar per trobar els errors de programació. D'aquesta manera s'aconsegueix una aplicació robusta i consistent de cara a la utilització per part de l'usuari, tracta de fer proves amb diferents tipus de inputs inesperats, efectuar accions repetidament o jugar amb els elements gràfics per tal d'aconseguir resultats inesperats, com ja s'ha esmentat abans.

Durant tot el projecte l'eina de treball principal ha estat Flask [1], un *microframework* de programació basat en llenguatge *Python* pel desenvolupament d'aplicacions i pàgines web, Flask s'utilitza per gestionar la lògica del sistema i les funcionalitats de la pàgina web, després a nivell de visualització dels elements web, s'ha fet servir Jinja2 [4] i per decorar aquesta vista per tal que sigui atractiva i intuïtiva de cara a l'usuari, s'ha fet servir Bootstrap [9], una eina de desenvolupament de codi HTML, CSS i JS de codi obert. Flask, disposa de dues llibreries principals, dos pilars per a la seva potencialitat, Jinja2 i Werkzeug [5], les quals permeten la fàcil utilització de plantilles, elements decoratius, entre altres característiques úniques que ofereix Flask. En el nostre cas, s'ha utilitzat un dels dos pilars de Flask, Jinja2, que ens permet representar de manera dinàmica tota la vista de l'aplicació carregant variables i estructures Python, a més, juntament amb els llenguatges de marques HTML i CSS d'estil propi i utilitzant la potencialitat de Bootstrap, els quals ens permeten mostrar tots els elements de la pàgina web i com els volem mostrar. Per acabar, s'ha utilitzat JavaScript per gestionar tota la part d'avaluació i seguretat a la banda del client, per tal de prevenir atacs o errors a l'hora de completar formularis, o realitzar accions no desitjades per part de l'usuari.

## 4 DESENVOLUPAMENT DE LA PÀGINA WEB

En aquest apartat s'explicarà i es mostrarà amb exemples els resultats del projecte, tant a nivell visual, com a nivell de lògic de programació. S'explicaran les funcionalitats bàsiques que s'han implementat per poder substituir tota la funcionalitat que oferia el Moodle, a més de millorar-la, fent-la més còmoda per a l'usuari i més intuïtiva. També s'introdueixen noves funcionalitats que aquest no oferia. Seguidament es mostrarà quins tipus de preguntes s'han implementat i la funcionalitat requerida per a implementar-les. Seguidament, s'explicarà com i quins fitxers es poden importar a l'aplicació web, per tal de poder llegir antics qüestionaris en diferents formats, per no haver de començar de nou cada cop que es vulgui fer un qüestionari diferent. S'explicarà quin format han de tenir aquests fitxers i quines extensions estan permeses. Per acabar s'explicaran les me-

asures de seguretat preses per evitar codi malintencionat, atacs per part de tercers. A més també s'explicaran quins controls s'han establert per tal d'aconseguir que l'usuari no introdueixi dades errònies i d'aquesta manera evitar que ho hagi de refer tot un altre cop o que deixi de funcionar l'aplicació web. Amb aquests controls es dona feedback a l'usuari sobre què està mal introduït i per què.

### 4.1 Funcionalitats bàsiques

Per entendre perfectament aquest projecte, hem de conèixer que el Moodle necessita una ruta o també anomenada categoria, per tal d'ordenar les preguntes importades a l'aplicació. A més, aquestes preguntes han de tenir un nom de pregunta, un enunciat i una resposta com a mínim, després hi ha altres variants de preguntes que tenen uns altres paràmetres addicionals com pot ser la tolerància o múltiples respostes combinades entre respostes certes i falses. Un exemple molt clar de l'aspecte d'aquestes preguntes i la seva visualització a la pàgina web és el de la Figura 1.

Generació automàtica de qüestionaris			
Banc de preguntes Importar Exportar Afegeir pregunta Esborra tot			
Entropia			
Nom pregunta	Enunciat pregunta	Resposta pregunta	Tolerància
Pregunta 1	Quant val $H(1/2,0,1/2)$ ?	1.0	0.0

Fig. 1: Exemple de la pantalla principal de l'aplicació.

En aquesta pàgina principal, també s'anomena "Banc de preguntes", com podem veure a l'anterior figura. Amb aquest botó de la barra de navegació es pot tornar des de qualsevol lloc a la vista principal amb la utilització de funcions que ofereix Flask. Altres funcionalitats bàsiques que es poden realitzar utilitzant la barra de navegació a la part superior de la vista, són les següents:

- **Importar**, ens permet importar fitxers amb unes extensions i formats determinats, l'aplicació processa i afegeix les preguntes al nou qüestionari que s'està generant en el moment de la importació.
- **Afegeir preguntes**, fent servir una estructura de *dropdown* ens mostra tots els tipus de preguntes disponibles.
- **Esborrar tot**, aquest botó ens permet netejar el banc de preguntes, és a dir, deixar buida l'estructura de dades que conté totes les preguntes del banc de preguntes.
- **Exportar**, ens descarrega un fitxer, com es pot veure més endavant a la Figura 2, aquest fitxer conté totes les preguntes del banc de preguntes en format *.xml*.

Altres funcionalitats implementades fora de la barra de navegació són. Per exemple, la manera en què generem el contingut de la taula del banc de preguntes, es fa mitjançant la llibreria Jinja2 que ofereix Flask, amb aquesta llibreria podem agafar dinàmicament el contingut de les variables de Flask i les mostrem en format taula, com es pot veure a la Figura 1, de manera que sigui fàcilment comprensible per l'usuari, la taula es mostra amb la categoria a dalt, una per

a cada taula, això indica que les preguntes contingudes en aquella taula es troben únicament dins d'aquella categoria. Cada taula pot contenir qualsevol tipus de pregunta, a més, pot tenir diferents tipus de preguntes en una mateixa taula. A la taula es mostra nom de la pregunta, enunciat, resposta. Com ja s'ha mencionat anteriorment, hi ha una columna addicional, la qual conté una creu de color vermell. Aquesta columna ens permet la funcionalitat bàsica d'esborrar una pregunta que havíem afegit anteriorment al nostre banc de preguntes.

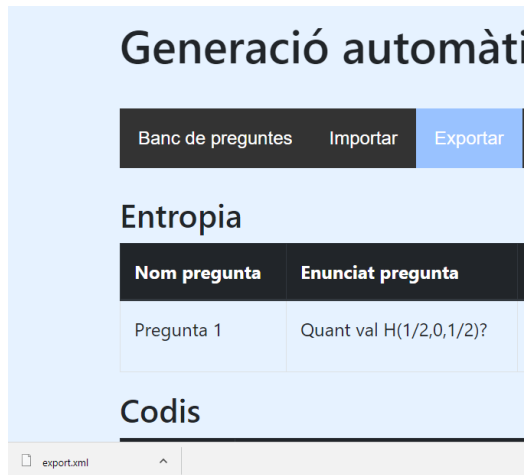


Fig. 2: Exemple de la funcionalitat d'exportar un fitxer.

A l'hora d'afegir una pregunta, uns dels camps que hem d'introduir és a quina categoria es trobarà aquesta pregunta. La web té en compte totes les categories que es troben al banc de preguntes de la web i les mostri en un llistat per escollir alguna de les existents. En molts casos, és molt possible que els usuaris o usuàries vulguin afegir moltes preguntes d'una mateixa categoria, la web permet que no calgui tornar a escriure la categoria en qüestió. En qualsevol cas, també hi ha l'opció d'escriure una nova categoria. Aquesta funcionalitat és possible gràcies a Jinja2, ja que, un altre cop, ens permet recollir dades de variables o estructures Flask i mostrar-les dinàmicament dins dels tags HTML que es poden anar creant amb bucles i condicionals gestionats per Jinja2. Sí no escollim cap categoria, aquella pregunta s'afegirà a l'última categoria utilitzada al banc de preguntes, si no hi ha cap, es posarà automàticament una categoria per defecte. Podem apreciar aquesta funcionalitat a la Figura 3.

Una altra funcionalitat bàsica als formularis d'afegir preguntes és la d'afegir camps dinàmicament amb un botó. Això permet, en alguns tipus de preguntes, ajustar aquella pregunta perquè s'adapti a com la vol l'usuari, permetent afegir o eliminar alguns camps, com podem veure a les següents figures. Aquesta funcionalitat s'ha implementat a nivell de client-side, amb codi JavaScript, creant o eliminant elements HTML per col·locar el nou camp al lloc desitjat.

## 4.2 Tipus de preguntes

En aquest apartat s'explicarà, quins tipus de preguntes s'han implementat, quina funcionalitat tenen i com es visualitzen dintre de la pàgina web. Hi han tres tipus de preguntes implementades, les preguntes de resposta breu, les preguntes d'opció múltiple i les preguntes de resposta numèrica.

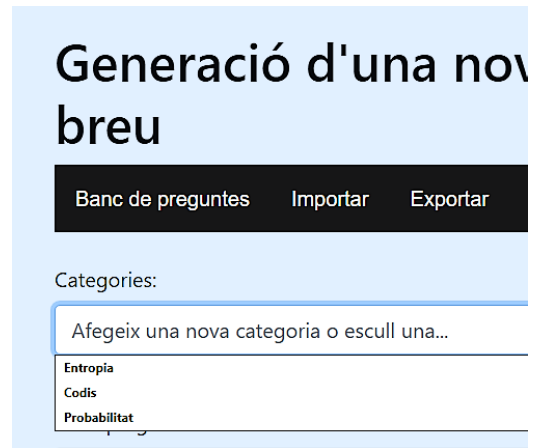


Fig. 3: Exemple de la funcionalitat d'escollir categoria.

Les preguntes de resposta breu i numèrica, tenen una opció addicional per ser afegides, es poden afegir també amb paràmetres. Els camps comuns per a tots aquests tipus de pregunta són, categoria, que s'afegix una per defecte si no s'especifica, nom de pregunta i enunciat, la resta de camps depenen del tipus de pregunta, s'explicarà a continuació.

### 4.2.1 Pregunta de resposta breu

En primera instància tenim el tipus de pregunta resposta breu. És el tipus de pregunta més bàsic.

Fig. 4: Exemple del formulari de la pregunta amb resposta breu.

Com hem pogut veure a la Figura 4, conté els camps:

- **Categoria**, el qual consisteix en la ruta que utilitzarà el Moodle per guardar la pregunta.
- **Nom de la pregunta**, un camp del Moodle, que et permet distingir la pregunta ràpidament.
- **Enunciat de la pregunta**, valor que conté l'explicació sobre allò que s'ha de contestar o resoldre.
- **Resposta de la pregunta**, valor que conté el resultat de la pregunta.

Aquests tipus de preguntes, de resposta breu, la resposta donada al qüestionari serà considerada com una cadena de caràcters i serà considerada certa si i només si coincideix exactament amb la solució, per tant l'usuari ha de donar una resposta exacta. A la Figura 5, a continuació, veiem com es representen aquests camps esmentats en la vista principal de l'aplicació web.

Tipus criptosistema				
Nom pregunta	Enunciat pregunta	Resposta pregunta	Tolerància	
Pregunta tipo crypto 1	De quin tipus de criptosistema és el criptosistema RSA (clau pública/clau privada)?	clau pública	-	<span style="color: red;">✖</span>

Fig. 5: Exemple de la vista de la pregunta amb resposta breu.

#### 4.2.2 Pregunta de resposta numèrica

Un segon tipus de pregunta és el de resposta numèrica. Conté els següents camps:

- **Categoria**, el qual consisteix en la ruta que utilitzarà el Moodle per guardar la pregunta.
- **Nom de la pregunta**, un camp del Moodle, que et permet distingir la pregunta ràpidament.
- **Enunciat de la pregunta**, valor que conté l'explicació sobre allò que s'ha de contestar o resoldre.
- **Resposta de la pregunta**, valor que conté el resultat de la pregunta.
- **Tolerància**, valor que permet l'usuari que respon la pregunta un cert marge d'error per donar-la com a vàlida.

Aquest tipus de pregunta difereix de l'anterior només en un parell d'aspectes. En primer lloc el camp extra, la tolerància, és un camp numèric. L'altra variació és que el camp resposta també ha de ser un valor numèric. La pregunta de resposta numèrica, te la mateixa funcionalitat que la pregunta de resposta breu, però el resultat ha de ser un valor numèric, però amb una altra petita variació, el resultat no ha de ser exacte, ja que segons la tolerància introduïda, es permetrà a l'usuari errar per dalt o per sota de la resposta, respecte a aquella tolerància. Per exemple, imaginem que la resposta és 55 i la tolerància és 1, sé li donaria com a resposta correcta a l'usuari quan introduís, 54, 55 o 56. A nivell visual l'única diferència amb el primer tipus de pregunta explicat, és que a la columna de tolerància hi ha el valor introduït al formulari, en comptes d'un guió, com hi havia a la pregunta de resposta breu.

#### 4.2.3 Pregunta de resposta múltiple

Un tercer tipus de pregunta implementat en aquesta aplicació és la pregunta resposta múltiple.

A la Figura 6 es poden veure els nous camps que aquest tipus de pregunta conté:

- **Categoria**, el qual consisteix en la ruta que utilitzarà el Moodle per guardar la pregunta.

Enunciat pregunta d'opció múltiple

Resposta correcta per la pregunta d'opció múltiple

Afegir resposta correcta    Esborrar resposta correcta

Resposta falsa per la pregunta d'opció múltiple

Afegir resposta falsa    Esborrar resposta falsa

Afegir pregunta

Fig. 6: Exemple del formulari de la pregunta amb resposta múltiple.

- **Nombre de preguntes**, el qual consisteix a definir el nombre de preguntes que volem crear.
- **Nom de la pregunta**, un camp del Moodle, que et permet distingir la pregunta ràpidament.
- **Enunciat de la pregunta**, valor que conté l'explicació sobre allò que s'ha de contestar o resoldre.
- **Resposta correcta**, valor que conté un resultat correcte de la pregunta, mínim ha d'haver-hi 1 d'aquest tipus, es poden afegir totes les que es vulguin.
- **Resposta falsa**, valor que conté un resultat fals de la pregunta, mínim ha d'haver-hi 1 d'aquest tipus, es poden afegir totes les que es vulguin.

Com es pot veure, igual que en cas de la pregunta amb resposta breu i la pregunta amb resposta numèrica, tenim els camps de categoria, nom pregunta, enunciat. Ara bé, en aquest tipus de preguntes, no tenim una única resposta sinó un llistat de possibles respostes certes, possibles respostes falses i un camp on l'usuari o usuària pot indicar quantes preguntes vol generar. Té dos tipus de respostes diferents, les correctes i les falses, de les quals es requereixen 5 entre els dos camps, com a mínim, perquè el formulari sigui vàlid, aquestes 5 respostes es poden aconseguir utilitzant els botons d'afegir i esborrar resposta, funcionalitat bàsica ja esmentada anteriorment, pot ser qualsevol combinació de respostes certes i falses, i no hi ha un màxim de respostes de qualsevol tipus, però sí que ha d'haver-hi mínim una de cada. L'aplicació web recollirà ambdues llistes i farà una selecció aleatòria d'un total de 5 respostes barrejades entre els dos tipus per a cada pregunta de resposta múltiple que es vulgui introduir. Per últim aquest formulari ens ofereix la possibilitat d'afegir tantes preguntes de resposta múltiple com es vulgui, les 5 respostes de cada pregunta introduïda es seleccionaran aleatòriament d'entre les respostes. Es poden donar combinacions de 2 certes i 3 falses, 3 certes i 2 falses, 4 certes i una falsa, etc. Donant així sempre 5 respostes aleatòries per a cada pregunta.

Com es pot apreciar a la Figura 7, les respostes correctes estan destacades amb un color verd amb l'objectiu de saber

Nom pregunta	Enunciat pregunta	Resposta pregunta	Tolerància
Pregunta RSA	En un criptosistema RSA, on l'usuari A vol enviar un missatge xifrat a l'usuari B...	<p>Tant A com B coneixen la clau pública d'A</p> <p>Només B el podria desxifrar</p> <p>Cal que la clau pública de B sigui coneguda</p> <p>Es fa servir la clau pública de B per xifrar</p> <p>Tant A com B coneixen la clau privada d'A</p>	-

Fig. 7: Exemple de la vista de la pregunta amb resposta múltiple.

a simple vista quines i quantes correctes conté aquella pregunta en concret. Si generem varies preguntes, no sempre agafarà les mateixes respostes, ni certes ni falses, ni tampoc el mateix nombre de respostes de cada tipus.

#### 4.2.4 Pregunta de resposta breu amb paràmetres

Un altre tipus de pregunta implementada en aquest projecte és la pregunta de resposta breu amb paràmetres.

Fig. 8: Exemple del formulari de la pregunta amb resposta breu amb paràmetres.

A la Figura 8 es poden veure els nous camps que aquest tipus de pregunta conté:

- **Categoria**, el qual consisteix en la ruta que utilitzarà el Moodle per guardar la pregunta.
- **Nombre de preguntes**, el qual consisteix a definir el nombre de preguntes que volem crear.
- **Nom de la pregunta**, un camp del Moodle, que et permet distingir la pregunta ràpidament.
- **Enunciat de la pregunta**, valor que conté l'explicació sobre allò que s'ha de contestar o resoldre, també conté el paràmetre o paràmetres a variar per cada pregunta generada.
- **Paràmetres**, valor mínim i màxim que es substituirà de manera aleatòria per valors entre el mínim i el màxim, pel paràmetre o paràmetres introduïts al camp enunciat i resposta.
- **Resposta de la pregunta**, valor que conté el resultat de la pregunta, també conté el paràmetre o paràmetres a variar per cada pregunta generada.

Com es pot veure, igual que en cas de la pregunta amb resposta breu, té els camps, categoria, nom pregunta, i enunciat. La idea és la mateixa que l'altra pregunta de resposta

breu, però amb funcionalitats afegides, com la de seleccionar el nombre volgut de preguntes a generar a la vegada i la d'afegir tants paràmetres com vulgui l'usuari, fent servir la funcionalitat d'afegir elements HTML esmentada anteriorment. A més, el camp resposta, es pot expandir, en tags HTML seria el textarea, per tal de poder introduir la fórmula o codi desitjat amb comoditat. S'ha de donar una resposta concreta, que és correcta, si i només si coincideix exactament amb el camp resposta. A la Figura 9, podem veure un exemple de preguntes breu generades amb paràmetres.

Nom pregunta	Enunciat pregunta	Resposta pregunta	Tolerància
Sumatori 3 params 1	Quant val el sumatori de 2, 9 i 7	18	-

Fig. 9: Exemple de la vista de la pregunta amb resposta breu amb paràmetres.

#### 4.2.5 Pregunta de resposta numèrica amb paràmetres

El últim tipus de pregunta és la pregunta de resposta numèrica amb paràmetres, la qual no difereix gaire del tipus de pregunta anterior. Aquest tipus de pregunta té els següents camps:

- **Categoria**, el qual consisteix en la ruta que utilitzarà el Moodle per guardar la pregunta.
- **Nombre de preguntes**, el qual consisteix a definir el nombre de preguntes que volem crear a la mateixa vegada.
- **Nom de la pregunta**, un camp del Moodle, que et permet distingir la pregunta ràpidament.
- **Enunciat de la pregunta**, valor que conté l'explicació sobre allò que s'ha de contestar o resoldre, també conté el paràmetre o paràmetres a variar per cada pregunta generada.
- **Paràmetres**, valor mínim i màxim que es substituirà de manera aleatòria per valors entre el mínim i el màxim, pel paràmetre o paràmetres introduïts al camp enunciat i resposta.
- **Resposta de la pregunta**, valor que conté el resultat de la pregunta, també conté el paràmetre o paràmetres a variar per cada pregunta generada.
- **Tolerància**, valor que permet l'usuari que respon la pregunta un cert marge d'error per donar-la com a vàlida.

Aquest tipus de pregunta té la mateixa funcionalitat que la pregunta de resposta numèrica, com ja s'ha explicat anteriorment, però amb alguns extres. Hi ha petites variacions de funcionalitat. Una primera variació és que la resposta és obligatòriament un valor numèric, també és un textarea com al cas anterior. Té un camp addicional, la tolerància, que també consisteix en un valor numèric. A més, com a la pregunta anterior, també té paràmetres amb la funcionalitat dels botons i tags HTML esmentada anteriorment i un camp per especificar el nombre de preguntes d'aquest tipus que es volen afegir. I l'última diferència, és a nivell visual, la columna de la tolerància està emplenada amb el valor introduït.

### 4.3 Importació de fitxers

En aquest apartat s'explicarà, quins tipus de fitxers estan permesos per importar en la pàgina web, quina funcionalitat tenen i com es visualitzen dintre de la pàgina web. Actualment es poden importar fitxers amb una extensió .xml i fitxers amb extensió .xlsx, a part aquests fitxers han de tenir una estructura concreta, per tal de poder ser gestionats i importats correctament a la pàgina web.

#### 4.3.1 Gestió de fitxers .xml

Per poder importar fitxers amb l'extensió .xml, han de tenir un format concret, per tal que l'aplicació web pugui saber quines són les parts de cada pregunta, on comença i acaba cada pregunta, entre altres. Un dels factors més importants és que tingui el tag d'inici "quiz", un altre és que tingui una categoria definida per tal de saber a quina categoria pertanyen les preguntes contingudes al fitxer. Per últim que les preguntes vinguin definides per un tipus de pregunta concret, com podria ser "shortanswer", per exemple, per tal de definir quin tipus de pregunta ens trobarem i saber com la gestionarem, a la Figura 10 tenim un exemple.

```

1  <?xml version='1.0' encoding='utf-8'?>
2  <quiz>
3      <question type="category">
4          <category>
5              <text>$course$/Entropia</text>
6          </category>
7      </question>
8      <question type="numerical">
9          <name GQ_id="4046">
10             <text>Pregunta 1</text>
11          </name>
12          <questiontext format="html">

```

Fig. 10: Exemple del format dels fitxers amb extensió .xml.

Un cop definit aquest format, amb el botó d'importar a la barra de navegació, podem introduir el fitxer amb l'extensió .xml al formulari i aquest serà processat. En aquest cas, es farà servir la llibreria de Python lxml etree per gestionar tota l'estructura xml continguda al fitxer. Aquesta llibreria ens permet transformar un fitxer xml en una estructura iterable, sobre la qual es poden realitzar certes funcions que et permeten, gestionar-la i modificar-la. D'aquesta manera podem gestionar l'arxiu, afegint o eliminant les preguntes que desitgi l'usuari o usuària.

#### 4.3.2 Gestió de fitxers .xlsx

Per poder importar fitxers amb l'extensió .xlsx, extensió posada pel conegut programa Microsoft Excel. Aquest fitxer ha de tenir un format concret, per tal que l'aplicació web, pugui de saber quines són les parts de cada pregunta, on comença i acaba cada pregunta, entre altres. En aquest cas el format del fitxer és més senzill, ja que només s'ha d'identificar amb un identificador a la primera cel·la de cada columna, el tipus de camp que contindrà cada columna. Un exemple és la Figura 11.

Un cop definit aquest format, amb el botó d'importar a la barra de navegació, podem introduir el fitxer amb l'extensió .xlsx al formulari i aquest serà processat. En aquest cas, es farà servir la llibreria de Python, pandas ExcelFile, la qual et permet gestionar tota l'estructura xlsx continguda al fitxer. A diferència del cas anterior, l'objectiu aquest cop serà

A	B	C	D	E
Categoria	NomPregunta	Enunciats	Respostes certes	Respostes falses
RSA	Pregunta RSA	En un criptosistema RSA,	Es fa servir la clau p	Es fa servir la clau q
			Tant A com B coneix	Tant A com B coneix
			Tant A com B coneix	Tant A com B coneix
			Cal que la clau públi	Cal que la clau públi
			Només B el podrà	Només A el podrà
			Tothom té accés al r	

Fig. 11: Exemple del format dels fitxers amb extensió .xlsx.

transformar tota aquesta estructura xlsx, a l'estructura lxml etree. Fent servir la mateixa estructura, es pot aprofitar les funcionalitats implementades per al tipus de fitxer anterior.

### 4.4 Controls i seguretat

En aquest apartat s'explicarà, quins tipus de controls a l'hora de permetre a l'usuari introduir dades en la pàgina web i també quines mesures de seguretat s'han implementat, per tal de fer la pàgina web la més segura possible.

#### 4.4.1 Controls a l'import de fitxers

Primerament, com ja s'ha explicat anteriorment, l'usuari pot introduir una sèrie de fitxers per tal de començar un nou qüestionari o continuar omplint un ja creat prèviament. Com que aquesta aplicació web permet els dos tipus de fitxers esmentats abans, es dona un feedback a l'usuari si ha introduït un fitxer que no té l'extensió requerida. També dona feedback si no s'introdueix cap fitxer. Podem apreciar un exemple de feedback donat a l'usuari en la Figura 12.

Fig. 12: Exemple d'un error al introduir un fitxer amb extensió no vàlida.

Aquesta gestió d'errors a l'hora d'introduir segons quins tipus de fitxers, es porta a terme gràcies a la funcionalitat flash de Flask, que ens permet donar feedback a l'usuari carregant els missatges d'error gestionats amb una macro. D'altra banda, el missatge d'error es fa visualment atractiu per l'usuari amb la utilització del ja esmentat Bootstrap. D'aquesta manera obtenim un feedback visible i clar per l'usuari. A més d'aquests missatges d'error, s'han implementat d'altres, per controlar els formats dels fitxers, donant feedback de la mateixa manera, si hi falta alguna capçalera al fitxer xlsx, per exemple.

#### 4.4.2 Controls a la gestió dels formularis

De la mateixa manera que a l'apartat anterior, a l'hora d'omplir els formularis també poden haver-hi una sèrie d'errors que podrien portar a problemàtiques amb la web, sí no es gestionessin correctament. Primerament, s'utilitza codi JavaScript, per tal de controlar que s'hagin introduït el mínim de paràmetres, per a les preguntes amb paràmetres

o el nombre mínim de respostes, tan certes com falses a les preguntes de resposta múltiple, per poder ser gestionades les preguntes correctament. Aquesta funcionalitat es gestiona amb JavaScript, donant feedback abans de validar el formulari, l'usuari rebrà una alerta dient que està mal introduït. Alguns exemples a la Figura 13, Figura 14 i Figura 15.

Fig. 13: Exemple d'un error al intentar esborrar un camp no permès.

Fig. 14: Exemple d'un error al fer la validació d'un formulari en preguntes amb paràmetres.

Fig. 15: Exemple d'un error al fer la validació del formulari en preguntes de resposta múltiple.

D'altra banda, un cop tenim tots els camps necessaris, procedim a validar el formulari. Aquest cop, en comptes de flash, s'utilitzarà una altra extensió de Flask, anomenada wtforms. Aquesta extensió ens permet avaluar el contingut introduït per l'usuari a cada camp del formulari i

amb una macro, gestionar tots els errors en els camps. Els hi mostrem a l'usuari i l'hi permetem corregir-ho sense la necessitat de refer el qüestionari sencer. Aquesta llibreria Flask, ja té implementades les seves pròpies funcionalitats de control, però se'n poden afegir més, amb l'objectiu de poder personalitzar el control del formulari. A la Figura 16, podem veure un exemple d'un formulari amb algun camp completat incorrectament.

Fig. 16: Exemple d'un error un cop feta la validació del formulari.

Aquest formulari wtforms, està decorat altra vegada amb Bootstrap per tal de ser visualment atractiu per a l'usuari, i que s'entengui perfectament que està fallant. Com podem veure només dona error en un camp, però tots els camps estan controlats amb els seus propis controls perquè tot funcioni correctament un cop s'ha validat el formulari. Tot això es valida amb la llibreria ja esmentada anteriorment, a més, aquesta llibreria et permet personalitzar els missatges de feedback que donen les funcions predefinides.

#### 4.4.3 Mesures de seguretat a la pàgina web

La primera mesura de seguretat és contra el Cross Site Request Forgery (CSRF) [10], consisteix a forçar al navegador web atacat a fer una acció no desitjada en un lloc confiable, per al qual l'usuari està autenticat. Hi ha diverses conseqüències a partir d'aquest atac, com fer que el sistema objectiu realitzi una acció a través del navegador sense que l'usuari objectiu sigui conscient. Un cop l'atacant s'ha validat en algun servei en concret. S'ha protegit els formularis contra CSRF, el mateix Flask t'ofereix una extensió que instal·lant-la i seguint les indicacions adients explicades a la documentació de Flask podem obtenir aquesta protecció d'avant aquest tipus d'atac. També hem d'incloure la comanda `form.csrf.token` als formularis de l'aplicació, per obtenir aquest nivell de seguretat davant aquest tipus d'atac.

La funció `Eval()` del llenguatge de programació JavaScript [11], avalua codi JavaScript representat com una cadena de caràcters, això ens permet introduir operacions complexes als camps de resposta de les preguntes amb paràmetres, aquesta funció avalua la cadena en guarda el resultat d'aquesta avaluació. Per aconseguir això l'`eval`, el que



fa és executar aquell codi JavaScript, això seria devastador a nivell de seguretat per la nostra aplicació, ja que permetria a qualsevol usuari, executar codi dins de l'aplicació, per solucionar aquesta problemàtica, la funcionalitat esmentada s'avalua al client-side. Sí el resultat obtingut no és correcte, retorna error, si és un script malintencionat, l'usuari, podrà realitzar el que vulgui amb la vista de l'aplicació, però mai podrà entrar a la part lògica de l'aplicació per obtenir dades, modificar-les o obtenir el codi de la pàgina web, per tant es conserva la integritat i la disponibilitat de l'aplicació [12].

Com a última mesura de seguretat implementada, s'ha utilitzat una altra llibreria Flask, anomenada Flask-Sessions [13]. Consisteix a crear variables de sessió. Una variable de sessió és un intercanvi d'informació semipermanent entre un ordinador i l'usuari per exemple, en el nostre cas la sessió conté tota la informació d'accions passades per part de cada usuari a la pàgina web, permetent així que l'usuari pugui reprendre el qüestionari que estava fent. També amb les sessions, aconseguim augmentar la seguretat, ja que amb les sessions cap altre usuari podrà accedir a la teva sessió i veure quines preguntes has generat, això afavoreix a la confidencialitat de cada usuari, ja que ningú pot accedir a la teva sessió.

## 5 VALIDACIÓ

Per validar totes les funcionalitats esmentades anteriorment, s'han realitzat una sèrie de proves. Una de les funcionalitats clau a validar ha sigut la importació de fitxers, s'han importat fitxers xml amb moltes preguntes, amb diferents tipus de preguntes i categories en un mateix fitxer. A nivell de fitxers xlsx s'han importat fitxers amb preguntes de resposta breu, de resposta múltiple i també de respostes numèriques. El contingut d'aquests fitxers ha sigut diferent, cadascun d'ells amb varies preguntes del mateix tipus. S'han fet proves deixant algunes columnes buides, l'aplicació dona feedback de l'error. A més, s'han fet proves d'importar diferents fitxers d'un mateix tipus, les últimes proves relacionades amb la importació de fitxers han sigut la introducció de fitxers no vàlids, sigui per l'extensió o format. Altres proves han sigut avaluar les preguntes, s'han generat 1000 preguntes de cada tipus de pregunta que té el camp per afegir el nombre de preguntes desitjat. A més s'han testejat tots els camps dels formularis introduint valors incorrectes, símbols, entre altres entrades diferents. Per validar la creació de sessions, s'han obert 3 navegadors, en cadascun d'ells s'ha obert la pàgina web i allà s'han generat diferents preguntes, per comprovar que realment no es barrejava la informació entre sessions. També s'han exportat fitxers amb tot tipus de preguntes, i aquestes s'han introduït a l'eina Moodle, han pogut ser visualitzades correctament al Moodle.

## 6 RESULTATS OBTINGUTS

S'han realitzat una sèrie de proves amb l'objectiu de veure fins a quin punt la pàgina web millora la funcionalitat que t'ofereix el Moodle. Per demostrar-ho, s'han realitzat diferents proves. S'ha generat amb el Moodle, 1 pregunta de resposta breu, 1 pregunta de resposta múltiple i 1 pregunta amb paràmetres. S'ha calculat el temps que és triga a crear cada una d'aquestes preguntes, després s'han creat aquestes

mateixes preguntes, però a l'aplicació web. Aquesta mateixa prova s'ha realitzat amb 1, 10, 100 i 1000 preguntes de cada tipus per a cada aplicació. Els temps obtinguts es poden veure a la Taula 1.

TAULA 1: TAULA DEL TEMPS QUE ES TRIGA A GENERAR LES DIFERENTS PREGUNTES EN CADA APLICACIÓ.

Plataforma	Número	Tipus resposta	Temps
Moodle	1	Breu i numèrica Múltiple Amb paràmetres	50seg 2min 10seg 3min 2seg
	10	Breu i numèrica Múltiple Amb paràmetres	8min 20seg 21min 40seg 3min 2seg
	100	Breu i numèrica Múltiple Amb paràmetres	1h 23min 20seg 3h 36min 40seg 3min 2seg
	1000	Breu i numèrica Múltiple Amb paràmetres	13h 53min 20seg 36h 6min 40seg 3min 2 seg
Web	1	Breu i numèrica Múltiple Amb paràmetres	18seg 1min 8seg 53seg
	10	Breu i numèrica Múltiple Amb paràmetres	3min 0seg 51seg 55seg
	100	Breu i numèrica Múltiple Amb paràmetres	30min 0seg 51seg 53seg
	1000	Breu i numèrica Múltiple Amb paràmetres	5h 0min 0seg 57seg 58seg

Als temps de creació de les preguntes a la pàgina web s'ha d'afegir el temps que és triga en exportar l'arxiu e importar-lo al Moodle. S'ha calculat també aquest temps per a fitxers amb 1, 10, 100 i 1000 preguntes, els seus temps respectivament són 43seg, 52seg, 57seg i 1min i 2seg. A la Secció A.1 de l'Apèndix hi ha l'explicació i els gràfics generats a partir de les proves esmentades, amb els resultats de la Taula 1.

Els resultats són favorables, el temps de diferència entre preguntes breus i numèriques no és molt millor, però, a la web és més ràpid i intuïtiu generar-ne una d'aquest tipus, no és òptim sí només es fa servir per generar una única pregunta, però quan en generes moltes, provoca una millora de temps substancial. En aquest primer cas, el temps es redueix considerablement, encara que aquest és el pitjor cas, ja que són preguntes individuals i s'han de generar d'una en una. En el cas de les preguntes de resposta múltiple, es veu una millora considerable, un altre cop, generar només una no és factible, encara que es redueixi el temps, però, ja que la pàgina web et permet generar-ne tantes com vulguis de cop, creant 10, 100 o 1000 preguntes, el temps de creació es redueix moltíssim. Com ja s'ha explicat, les respostes de les preguntes de resposta múltiple, es seleccionen de manera aleatòria entre la llista donada, per tant no sortiran 1000 preguntes iguals. En canvi, el Moodle no et permet generar totes les preguntes que es vulguin d'aquest tipus, has de generar-les d'una en una, fet que incrementa moltíssim el temps de generació al Moodle. Amb l'últim tipus de preguntes, en generar-les al Moodle, et varia els valors dels paràmetres, fet que fa que no hakis de generar gaires per

arribar a les 1000 preguntes diferents, de totes maneres, a la pàgina web és més fàcil i intuïtiu crear-les, ja que aquest era un dels objectius principals del projecte. Com es pot apreciar, s'aconsegueix una millora de temps considerable en crear una única pregunta amb paràmetres. Amb 1000 preguntes d'aquest tipus, la millora continua sent la mateixa pel que he explicat anteriorment, però això sorgeix en un cas senzill, com ja s'ha esmentat, afegir més d'un paràmetre al Moodle amb fórmules complicades dificulta i alenteix molt la creació d'aquestes, en canvi, amb la pàgina web, és més fàcil. Per tant, tot i no mostrar una millora substancial de temps, ja que en crear-les no hi ha gaire diferència, el temps d'aprenentatge, crear-les de manera intuïtiva i amb facilitat si influeix molt en aquest últim cas.

Com a última millora de la pàgina web respecte a l'eina Moodle, ens permet importar els tres tipus de preguntes, resposta breu, resposta numèrica i de resposta múltiple, amb fitxers excel. Aquesta funcionalitat no la té el Moodle i ens permet crear preguntes més fàcilment a l'excel i importar-les a la pàgina web per exportar-les en format Moodle ràpidament, això facilita una major flexibilitat a l'hora de quins tipus de fitxers utilitzar per fer servir el Moodle.

## 7 LÍNIES FUTURES

En vista al futur, aquesta aplicació s'ha començat de zero, encara queden moltes funcionalitats útils per desenvolupar, com podria ser, un botó per editar qualsevol camp de cada pregunta, per si t'has equivocat en alguna part de la qüestió, poder-la modificar sense haver d'esborrar-la i tornar-la a fer. També es poden implementar altres tipus de preguntes que el Moodle també accepta i que podrien ser útils a l'hora de realitzar els qüestionaris. Altres funcionalitats que podrien ser útils en el futur són la de poder importar fitxers amb altres extensions, com ods, xls o txt. També, en comptes de valors aleatoris en els paràmetres, introduir una llista de les duples de paràmetres que es volen fer servir, i així obtenir els resultats volguts més directament. Entre altres funcionalitats que es puguin anar trobant pel camí.

## 8 CONCLUSIONS

Com a conclusió, tots els objectius principals del projecte s'han complert, s'ha creat una aplicació web vàlida i funcional, que, no només aconsegueix millorar els temps de creació respecte a l'eina de treball Moodle, sinó que també ofereix la possibilitat d'afegir tantes preguntes i paràmetres com es vulgui. A més, s'ha incorporat una millora de visualització i feedback a la part visual de l'aplicació web respecte a l'eina Moodle. També, s'ha complert un altre objectiu, d'expandir la funcionalitat Moodle, permetent afegir fitxers amb altres extensions que no siguin xml. Tots els tipus bàsics de preguntes han sigut implementats, les funcionalitats d'importar i exportar fitxers han tingut èxit i com s'ha demostrat a les gràfiques hi ha una gran millora en els temps d'utilització de la web. També s'ha aconseguit una pàgina web segura davant d'atacs de CRSF en els formularis, a més, s'aconsegueix integritat i confidencialitat amb les variables de sessió implementades. Per últim, mantenim la disponibilitat de la pàgina web executant la funcionalitat de l'eval al client-side.

## AGRAÏMENTS

Agraïments a Irene Burgueño, per tot el suport emocional i donar-me forces en molts moments. Agraïments a la professora del TFG, Cristina Fernández, per tot l'ajut de redacció i acompanyament durant el treball, també per l'script en Python facilitat, que ha agilitzat molt el tractat dels tipus de preguntes. Agraïments també al professor Ian Blanes per tot l'ajut oferit en diferents qüestions de programació sorgides durant la realització del projecte i per encaminar amb idees el projecte.

## REFERÈNCIES

- [1] *Flask documentation* <http://flask.pocoo.org/docs/0.12/> [Online]
- [2] *SCRUM* <https://proyectosagiles.org/que-es-scrum/> [Online]
- [3] *Exploratory Testing* <https://www.guru99.com/exploratory-testing.html> [Online]
- [4] *Jinja2* <http://jinja.pocoo.org/docs/2.10/> [Online]
- [5] *Werkzeug* <http://werkzeug.pocoo.org/docs/0.14/> [Online]
- [6] *LXML etree* <http://lxml.de/tutorial.html> [Online]
- [7] *Gestió LXML* [https://www.josedomingo.org/pledin/2015/01/trabajar-con-ficheros-xml-desde-python\\_1/](https://www.josedomingo.org/pledin/2015/01/trabajar-con-ficheros-xml-desde-python_1/) [Online]
- [8] *Moodle* [https://docs.moodle.org/35/en/Main\\_page](https://docs.moodle.org/35/en/Main_page) [Online]
- [9] *Bootstrap* <https://getbootstrap.com/docs/4.1/getting-started/introduction/> [Online]
- [10] *CSRF* [https://www.owasp.org/index.php/Cross-Site\\_Request\\_Forgery\\_\(CSRF\)\\_Prevention\\_Cheat\\_Sheet](https://www.owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF)_Prevention_Cheat_Sheet) [Online]
- [11] *EVAL* [https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos\\_globales/eval](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/eval) [Online]
- [12] *Pilares de la seguridad informática* <https://www.pmg-ssi.com/2018/02/confidencialidad-integridad-y-disponibilidad/> [Online]
- [13] *Sessions* [https://en.wikipedia.org/wiki/Session\\_\(computer\\_science\)](https://en.wikipedia.org/wiki/Session_(computer_science)) [Online]

## APÈNDIX

### A.1 Apèndix resultats obtinguts

En aquesta secció de l'apèndix es mostren 3 gràfiques. Aquestes representen el temps que es triga a generar cada tipus de pregunta, amb el Moodle i la pàgina web. Els valors quantitius estan agafats segons la Taula 1, mostrada a la Secció 6, Resultats obtinguts. Els valors agafats, ja que són, el temps trigat per a la generació de preguntes amb resposta breu o numèrica a l'eina Moodle, comparant-los amb el temps trigat per a la generació d'aquestes amb la pàgina web. S'han fet proves per a 1 pregunta, 10 preguntes, 100 preguntes i 1000 preguntes.

Aquest mateix cas també s'ha aplicat per a preguntes de resposta múltiple i per a preguntes de resposta amb paràmetres.

Com podem veure a la gràfica 17, no s'ha pogut implementar una manera òptima per reduir el temps de generar aquest tipus de pregunta a l'aplicació, però, gràcies al fet que és més fàcil d'utilitzar i més intuïtiva, el temps que es triga, es veu reduït en la pàgina web.

En la gràfica 18, sí que s'ha trobat una manera de generar totes les preguntes amb resposta múltiple volgudes d'una manera eficient. D'uns 130.000 segons que es trigaria a crear 1000 preguntes d'aquest tipus al Moodle, passem a crear-les en un parell o tres de minuts com a molt. Per tant s'obté un gran benefici de temps, encara que s'hagi d'importar aquelles qüestions al Moodle, amb el que trigariem un parell de minuts més.

En la gràfica 19, tampoc hi ha una millora molt substancial de temps, però, sí que n'hi ha aquesta millora en facilitat, principalment, és triga un parell de minuts menys en generar qualsevol nombre de preguntes amb la web que qualsevol nombre de preguntes amb el Moodle, tot i que aquest temps no és gaire substancial, ha sigut realitzat aquesta comparativa amb un cas senzill, com més paràmetres afegeixis al Moodle i més complicada sigui la fórmula, més trigues a generar aquest tipus de preguntes al Moodle i per tant més temps trigues, en canvi, a la web, la dificultat de generar-la amb un paràmetre o amb més d'un és la mateixa. Per tant, aquesta millora es veu més afectada, en una millora en la dificultat de creació.

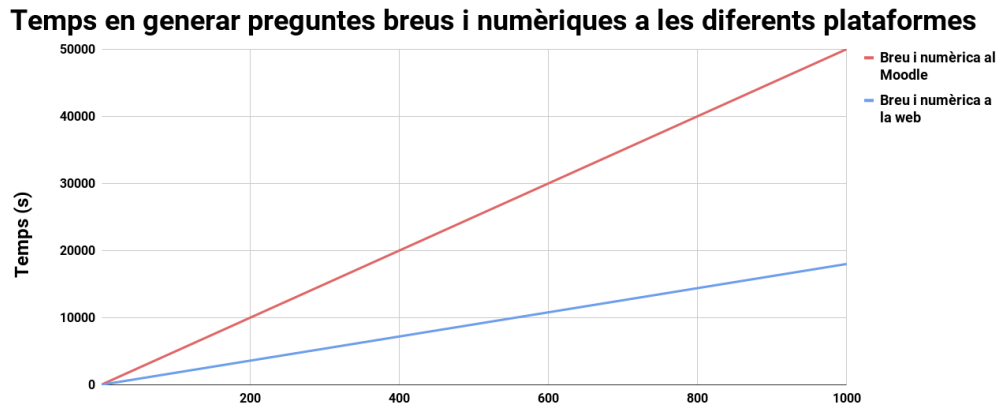


Fig. 17: Gràfic dels resultats de generar preguntes de resposta breu o numèriques en ambdues plataformes

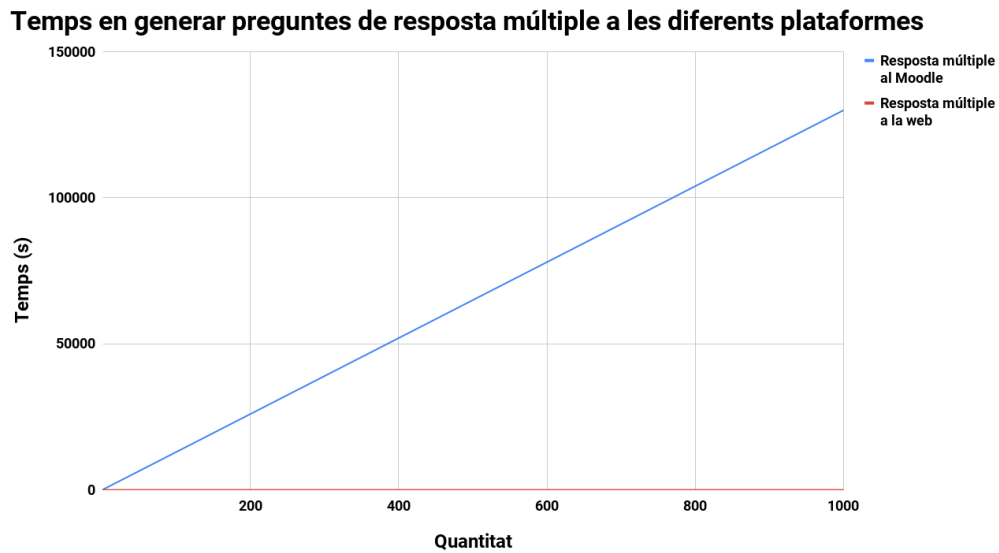


Fig. 18: Gràfic dels resultats de generar preguntes de resposta múltiple en ambdues plataformes

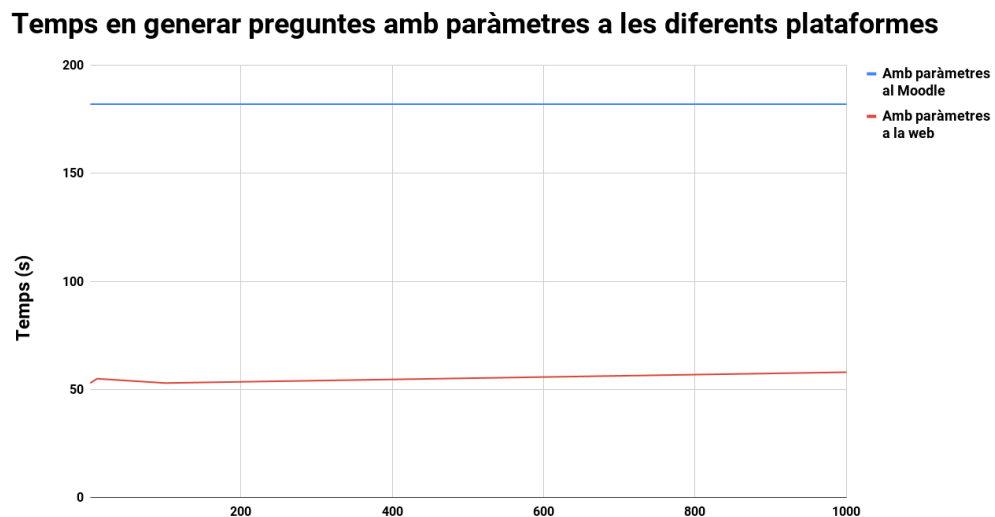


Fig. 19: Gràfic dels resultats de generar preguntes amb paràmetres en ambdues plataformes