

École polytechnique de Louvain

INGInious High School

**Adapter INGInious aux besoins des enseignants
du secondaire**

Auteur: **Benjamin SIMON**
Promoteurs: **Kim MENS, Olivier GOLETTI**
Lecteur: **Anthony GÉGO**
Année académique 2019–2020
Master [120] : ingénieur civil en informatique

Résumé

À l'heure actuelle, un grand nombre d'établissements scolaires intègrent des cours de programmation dans leurs grilles horaires et l'enseignement secondaire n'est pas en reste. Malheureusement, les outils disponibles à l'heure actuelle ne répondent pas entièrement aux besoins des enseignants. Certains outils proposent des cours préformatés, mais offrent peu voire pas de flexibilité pour adapter ces cours. À l'opposé, d'autres outils permettent beaucoup de flexibilité, mais n'offrent pas la possibilité de réutiliser des contenus pédagogiques existants. Dans ce mémoire, nous allons tenter de fournir une approche hybride permettant à l'enseignant de s'appuyer sur une base existante et de personnaliser cette base au fur et à mesure afin de s'adapter à ses besoins spécifiques.

Pour ce faire, nous partons de la plateforme inginius et l'étendons selon 3 axes principaux. Premièrement, nous ajoutons un système de sections afin de permettre aux enseignants de mieux structurer leurs cours. Ensuite, nous introduisons le principe de template, des cours génériques pouvant servir de base pour la création de nouveaux cours. Pour finir, nous proposons la possibilité d'importer et d'exporter des sections et des exercices dans des cours existants. Ces extensions ont ensuite fait l'objet d'une validation par des utilisateurs dont les retours ont été très positifs.

Ce travail a donc permis de mieux adapter inginius aux principaux besoins des enseignants du secondaire et ouvre la porte à de nouveaux développements dans cette direction.

L'ensemble de notre travail est disponible sur la plateforme Github à l'adresse suivante : <https://github.com/bensim602/INGInious>.

Une instance de cette plateforme contenant les plug-ins implémentés durant ce mémoire est hébergée à l'adresse suivante : <http://tfe-simon.info.ucl.ac.be>. Vous pouvez vous y connecter en utilisant l'identifiant **teacher** et le mot de passe **IngiProf**.

Remerciements

Tout d'abord, je tiens à remercier mes promoteurs Kim Mens et Olivier Goletti pour leur supervision tout au long de l'année, leurs précieux conseils et leur support.

J'aimerais également remercier Anthony Gégó et François Michel pour leurs conseils et avis qui m'ont guidé tout au long de ce travail.

Ensuite, je remercie Denis Lemoine d'avoir pris le temps de partager son expérience d'enseignant avec moi.

Merci à mes colocataires et amis qui m'ont supporté et aidé tout au long de cette année, en particulier Arthur Sluÿters et Jean-Martin Vlaeminck.

Je tiens aussi à remercier toutes les personnes ayant accepté de donner un peu de leur temps pour participer à la validation de ce travail.

Finalement, je tiens aussi à remercier mes parents, mes grands-parents et ma sœur pour leur support et leurs conseils ainsi que pour leur aide dans la relecture de ce travail.

Table des matières

1	Introduction	1
1.1	Contexte	1
1.2	Problèmes	2
1.3	Motivations	2
1.4	Objectifs	3
1.5	Approche	4
1.6	Contributions	4
1.7	Plan	5
2	Travaux similaires	6
2.1	Code.org	6
2.2	OpenClassrooms	8
2.3	edX	8
2.4	Moodle	9
2.5	Oscar	9
2.6	Conclusion	10
3	Plateforme INGIInious	12
3.1	Description de la plateforme	12
3.1.1	Exercices	13
3.1.2	Cours	14
3.1.3	Groupes	15
3.2	Fonctionnement	15
3.2.1	Plateforme	16
3.2.2	Docker	17
3.2.3	Système de fichiers	18
3.2.4	Base de données	19
3.3	Étendre la plateforme	19
3.3.1	Modifier la plateforme	20
3.3.2	Plug-ins	20

4	Organisation	21
4.1	Approche générale	21
4.2	Planning	22
4.3	Méthodologie	23
4.3.1	Itérations	23
4.3.2	Backlog	23
4.4	Outils utilisés	24
4.4.1	Trello	24
4.4.2	Github	24
5	Plateforme EZLearn	25
5.1	Architecture	25
5.2	Fonctionnalités	26
5.2.1	Éditer un cours	26
5.2.2	Utilisation des templates	27
5.2.3	Gestion des utilisateurs	28
5.2.4	Établissements	29
5.2.5	Progression des étudiants	30
5.3	Défauts de l'approche	31
6	Solution	32
6.1	Structure du cours	32
6.1.1	Fonctionnalités	32
6.1.2	Implémentation	35
6.1.3	Contribution	38
6.2	Importer et exporter des exercices	39
6.2.1	Fonctionnalités	39
6.2.2	Implémentation	41
6.2.3	Contribution	43
6.3	Template	44
6.3.1	Fonctionnalités	44
6.3.2	Implémentation	50
6.3.3	Contribution	54
7	Validation	55
7.1	Méthodologie	55
7.2	Résultats	56
7.3	Améliorations	57
7.3.1	Erreurs	57
7.3.2	Indications	57
7.3.3	Glisser-déposer	58

7.3.4	Fusionner les onglets	58
7.3.5	Simplifier l'utilisation	59
8	Améliorations futures	60
8.1	Intégration à INGIInious	60
8.2	Privilèges enseignants	60
8.3	Plateforme globale	60
8.4	Suivi des étudiants	61
8.5	Création d'étudiants	61
9	Conclusion	62
A	Guide d'installation	66
B	Exemple de structure d'un cours	69
C	Protocole de validation	71
D	Questionnaire de validation	76
D.1	Questions	76

Chapitre 1

Introduction

1.1 Contexte

Cette thèse ainsi que «Source Code : un catalogue open source d'exercices informatiques» [8] et «Exploration d'une application adaptative pour évaluer les compétences en informatique» [4] sont nées de la prise de conscience qu'il existe un manque d'outils pour les enseignants en informatique du secondaire. De plus, en Belgique ainsi que partout dans le monde, on constate une tendance à trouver peu, voire aucun cours de programmation dans le cursus des étudiants du secondaire.

En revanche, on retrouve de plus en plus de cours de bureautique, souvent qualifiés de cours d'informatique, mais bien différents des cours de programmation. Un bon exemple de cette tendance est la mise en place du passeport TIC¹ par la fédération Wallonie Bruxelles. Ce passeport a pour but de fournir un référentiel des compétences liées aux Technologies de l'Information et de la Communication (TIC) afin de permettre aux enseignants d'intégrer ces compétences dans leurs cours.

Le peu de cours de programmation est problématique pour le domaine de l'informatique. En effet, comme certains articles [6] le démontrent, une des principales raisons pour le manque d'intérêt des étudiants pour se lancer dans les études d'informatique est le manque de connaissances du domaine. Cela entraîne une vision biaisée et donc peu d'attrait pour ce domaine.

Malheureusement, ce désintérêt mène aussi à une pénurie d'informaticiens, des profils pourtant très recherchés à l'ère du numérique. Il résulte donc de tout cela un important écart entre l'offre et la demande².

Tous ces éléments font que, comme beaucoup d'autres pays, la Belgique com-

1. <http://www.enseignement.be/index.php?page=27182&navi=3683>

2. <https://www.rtl.be/info/vous/temoignages/la-2-penurie-d-informaticiens-continue-de-s-accroitre-il-y-a-assez-de-techniciens-mais-on-s-arrache-les-developpeurs-et-analystes-1130683.aspx>

mence timidement à essayer d'introduire des cours d'informatique dans l'enseignement secondaire. Cette volonté se manifeste par exemple dans le pacte d'excellence et la plateforme Digital Wallonia. Ces derniers ont, entre autres, créé un groupe de travail SI² [15] dans le but d'introduire l'enseignement des sciences informatiques dans le secondaire inférieur.

1.2 Problèmes

Le manque de matériel pédagogique informatique pour les enseignants du secondaire est la principale préoccupation de ce mémoire. Ce problème général peut être subdivisé en plusieurs sous-problèmes. Tout d'abord, l'accès à des exercices informatiques pouvant être réutilisés par les enseignants est relativement compliqué. Pour remédier à ce problème, la thèse «Source Code : un catalogue open source d'exercices informatiques» tente de créer une librairie pour de tels exercices. Ensuite, il n'existe pas encore de plateformes permettant d'apprendre l'informatique tout en s'adaptant aux besoins spécifiques de l'étudiant. Ce problème est abordé dans la thèse «Exploration d'une application adaptative pour évaluer les compétences en informatique».

Finalement, nous avons aussi remarqué que les plateformes d'apprentissage existantes sont souvent mal adaptées aux besoins des enseignants du secondaire (voir chapitre 2). C'est ce problème que nous allons aborder dans ce mémoire.

Nous avons identifié plusieurs contraintes liées au contexte particulier de l'enseignement secondaire, notamment le fait que les établissements disposent de ressources limitées. Par exemple, il n'est pas réaliste de demander à chaque école d'héberger sa propre plateforme, car beaucoup d'établissements ne disposeront pas des ressources nécessaires. Ensuite, il faut prendre en compte que la discipline est relativement nouvelle et que l'expérience des enseignants dans le domaine est souvent limitée. Une autre conséquence de cette nouveauté est qu'il est parfois nécessaire de fournir la liste des compétences transversales du cours afin de permettre aux enseignants de justifier la place du cours dans le programme. Finalement, ces cours sont soumis à des contraintes de temps très variables d'un établissement à un autre.

1.3 Motivations

À l'ère actuelle, l'informatique occupe une place de plus en plus importante dans de nombreux aspects de la société. Nous estimons qu'il est important que ce changement se manifeste dans le programme de l'enseignement secondaire. En effet, l'apprentissage de l'informatique présenterait plusieurs avantages pour les

étudiants. Tout d’abord, il permettrait aux étudiants sortants de l’enseignement secondaire de se faire une idée moins biaisée de ce qu’est l’informatique et cette meilleure information pourrait orienter le choix de leurs études supérieures vers ce domaine. Ensuite, l’apprentissage de l’informatique permettrait de développer de nombreuses compétences utiles aux étudiants du secondaire [13].

Toutefois, les plateformes d’apprentissage en ligne ne sont malheureusement pas au niveau des attentes des enseignants. Cela a d’ailleurs été mis en avant avec les récents événements liés aux COVID-19 et les mesures de confinement qui en ont découlé.

Finalement, ce mémoire permet aussi de participer au développement et à l’amélioration d’une plateforme existante utilisée par de nombreux étudiants de l’Université Catholique de Louvain ainsi que par d’autres universités partout dans le monde.

1.4 Objectifs

Ce mémoire a pour objectif de fournir une plateforme facilitant l’apprentissage de concepts de programmation aux étudiants du secondaire. Nos recherches nous ont permis d’identifier plusieurs aspects importants pour une telle plateforme.

Tout d’abord, nous estimons qu’il est important que les enseignants puissent créer et personnaliser leurs propres cours afin que ceux-ci correspondent exactement à leurs besoins. Toutefois, nous réalisons que créer entièrement un tel cours peut représenter un obstacle majeur pour un enseignant du secondaire. Par conséquent, nous voulons leur fournir la possibilité de créer un cours à partir d’une base existante et de modifier ce cours selon leurs désirs. Des cours structurés en chapitres permettent aussi aux étudiants de mieux se repérer dans les cours et de comprendre leurs structures. De plus, ils permettent aux enseignants de facilement sélectionner les éléments qu’ils veulent garder quand ils créent un cours à partir d’une base existante.

Ensuite, il est intéressant de fournir un support pédagogique pour les cours pouvant être réutilisés par les enseignants. En effet, beaucoup de ces enseignants ne sont pas habitués à donner des cours d’informatique et ne sont pas familiers avec la matière abordée. Un guide pédagogique conçu pour aider les enseignants et leur permettant de mieux identifier les concepts abordés par un cours peut donc s’avérer utile.

Finalement, nous voulons aussi permettre l’évaluation des compétences des étudiants. Cela peut se décliner sous deux formes. Tout d’abord, des questionnaires pouvant être corrigés automatiquement permettent de tester les compétences théoriques de l’étudiant. En ce qui concerne les compétences pratiques, la réalisation de petits projets semble plus appropriée. Ces projets peuvent ensuite être évalués

par l’enseignant, mais dans certains cas il peut être intéressant de fournir la possibilité d’évaluer automatiquement ces projets afin d’alléger la charge de travail des enseignants. Nous souhaitons aussi permettre aux enseignants de visualiser les progrès de leurs étudiants sur l’ensemble de ces évaluations.

1.5 Approche

Deux approches ont été envisagées pour ce mémoire. La première consiste à créer une nouvelle plateforme. Celle-ci permettrait de créer des cours contenant de la théorie et des exercices, d’évaluer les étudiants et de suivre leur progression. Elle permettrait d’héberger des cours pour plusieurs établissements et utiliserait inginius [7] pour la création et la correction d’exercices. Cependant, cette approche a dû être abandonnée, car elle s’est avérée irréalisable dans le cadre de ce mémoire.

La seconde approche qui a finalement été retenue consiste à étendre la plateforme inginius. Cette extension se fait par le biais de trois plug-ins ayant chacun pour objectif de répondre à certains besoins des enseignants du secondaire. Le premier plug-in permet de structurer un cours en sections et de modifier cette structure. Le second plug-in, quant à lui, permet de créer un nouveau cours tout en se reposant sur une base existante grâce au système de template. Finalement, le dernier plug-in permet d’importer et exporter des sections et exercices par l’intermédiaire d’une archive au format prédéfini. Ensuite, nous avons conduit des entretiens avec différents utilisateurs afin de valider le bon fonctionnement de chaque extension.

1.6 Contributions

Ce mémoire contribue au développement de la plateforme inginius en proposant une série de plug-ins qui pourront, par la suite, y être intégrés. Chacun de ces plug-ins apporte un élément nouveau à la plateforme.

Le premier plug-in apporte aux enseignants la possibilité de structurer leurs cours en sections. Le besoin d’un tel ajout avait déjà été identifié par certains professeurs de l’UCLouvain utilisant cette plateforme. De plus, avec l’ajout du système de template, cette extension permet aussi de sélectionner les chapitres que l’on veut utiliser lors de la création d’un cours à partir d’un template.

Le second plug-in permet de se reposer sur ce qui existe déjà lors de la création d’un cours. Il permet aux enseignants de choisir une base pour leurs cours et de la modifier pour s’adapter à leurs besoins spécifiques.

Finalement, le dernier apporte la possibilité de facilement réutiliser des exercices et des sections d’un cours à un autre. Tout cela aurait déjà été possible en copiant des dossiers, à condition d’avoir accès au système de fichiers de chacun des cours

concernés. Cependant, ce plug-in facilite le processus en fournissant une interface pour faire ces imports et exports ainsi qu'un format spécifique pour l'archive.

Durant la validation, nous avons pu démontrer le bon fonctionnement et perfectionner chacun de ces plug-ins. Grâce à ces plug-ins, inginius répond maintenant en grande partie à nos objectifs tout en respectant la majorité des contraintes liées à l'enseignement secondaire. Cependant, elle ne permet pas encore de suivre la progression globale des étudiants ou de fournir une instance générique pour l'ensemble des établissements.

Si ces plug-ins sont intégrés à la plateforme, notre mémoire participerait donc à l'évolution de l'exerciseur utilisé par les enseignants et étudiant de l'UCLouvain et d'autres universités. De plus, il permettrait aussi à certains établissements secondaires de profiter eux aussi de cette plateforme.

1.7 Plan

Dans la suite de ce document, le chapitre 2 présentera brièvement les plateformes existantes qui offrent certains éléments de solution aux problématiques de ce mémoire. Ensuite, le chapitre 3 montrera plus en détail le fonctionnement de la plateforme inginius que nous étendons. Le chapitre 4 présentera l'organisation pour la réalisation de ce mémoire. Ensuite, le chapitre 5 offrira un aperçu de la première approche envisagée pour ce mémoire suivie d'une description détaillée de chacun des plug-ins ayant été réalisé dans le chapitre 6. Finalement, le chapitre 7 expliquera la méthode utilisée pour valider notre travail et les résultats obtenus durant cette validation. Avant de conclure avec le chapitre 9, le chapitre 8 indiquera brièvement le travail qu'il reste à accomplir.

Chapitre 2

Travaux similaires

Ce chapitre présente les plateformes de cours en ligne les plus connues et répondant au mieux aux objectifs présentés à la section 1.4. Il présente aussi brièvement les avantages et les inconvénients de chacune de ces plateformes.

2.1 Code.org

Code.org¹ est un organisme sans but lucratif dont l’objectif est de rendre l’informatique accessible à tous. Pour ce faire, le site propose un catalogue de cours. Étant donné l’objectif de cette ASBL, l’offre du catalogue est principalement orientée vers des cours d’introduction à l’informatique. La figure 2.1 présente les principaux éléments du site.

Un cours est séparé en plusieurs modules représentant chacun une unité d’apprentissage. Ces modules sont constitués principalement de textes et vidéos expliquant les différents concepts de l’unité. Afin de tester les connaissances des étudiants, ces cours contiennent aussi des exercices. Ces derniers peuvent prendre la forme de questionnaires à choix multiples et de questions ouvertes. D’autres sont basés sur l’écriture de programmes Blockly et sont testés et évalués automatiquement. Blockly est un langage de programmation graphique. Le principe général est d’assembler des blocs dans un éditeur visuel. Chacun de ces blocs représente une instruction de base de la programmation. Ce système permet d’apprendre les concepts de programmation sans s’encombrer de la syntaxe liée à un langage spécifique. Vous pouvez trouver plus d’information à ce sujet sur la page officielle du projet Blockly².

Le site web propose aux étudiants deux possibilités pour suivre les cours. Tout d’abord, il est possible pour l’étudiant de suivre n’importe quel cours en parcours libre. Pour ce faire, il lui suffit de consulter le catalogue, choisir un cours et suivre

1. <https://code.org>

2. <https://blockly.games/about>

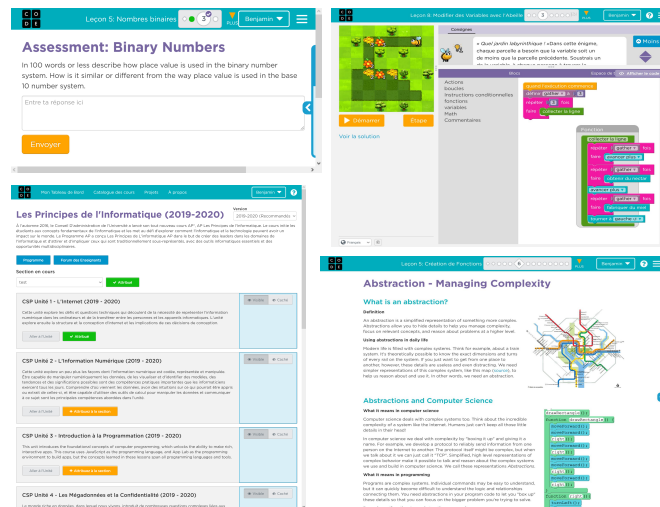


FIGURE 2.1 – Site Code.org

Source: <https://studio.code.org>

les modules désirés. L'autre possibilité est de s'inscrire dans une classe créée par un enseignant. Il est alors possible pour l'enseignant de suivre sa progression de manière visuelle et de choisir les modules qu'il doit suivre.

Plusieurs systèmes sont mis en place afin d'aider les enseignants qui désirent utiliser le site dans le cadre de leurs cours. Tout d'abord, chaque cours propose une description détaillée à leur attention. Celle-ci a pour objectif de leur permettre de se faire une idée de la matière abordée et de mieux comprendre les différents enjeux du cours. Des guides d'apprentissage sont aussi mis à leur disposition ainsi que des conseils pour préparer le cours. Afin de répondre aux questions que les enseignants pourraient se poser et permettre l'entraide entre enseignants, une section forum est aussi disponible pour chaque cours. Une fois que l'enseignant a créé une classe, il lui est possible d'y inscrire ses étudiants ou de laisser ceux-ci s'y inscrire. Il peut ensuite observer la progression de ses étudiants de manière visuelle et récupérer les réponses aux questions ouvertes et sondages.

Ce site convient donc aux enseignants débutant dans l'enseignement de l'informatique, car il fournit beaucoup de support pour les guider et les aider à donner leurs cours.

L'inconvénient majeur de ce site est qu'il propose très peu de flexibilité concernant la gestion du cours. En effet, il est uniquement possible de masquer certains modules. Il est en revanche impossible de créer de nouveaux exercices ou modules, de réorganiser un cours ou module ou de créer un cours soi-même.

2.2 OpenClassrooms

OpenClassrooms³ (anciennement Site du Zéro) est un site de MOOC francophone. Il propose de nombreux cours sur divers sujets, mais reste principalement orienté sur la programmation informatique. Les cours sont ouverts à tous, mais la rédaction de cours est limitée aux partenaires.

Le suivi des cours est principalement fait de manière autonome avec la possibilité d'obtenir une certification. Cela signifie qu'il est possible pour les enseignants de vérifier que les étudiants ont réussi le cours, mais pas de suivre le progrès de leurs étudiants au fur et à mesure et d'adapter l'enseignement en fonction.

Les cours sont séparés en modules et en chapitres. Chaque module théorique peut contenir du texte et des vidéos explicatives concernant la matière. Il existe deux types de modules pratiques pour tester les connaissances des étudiants : les questionnaires, qui sont corrigés automatiquement, et les activités. Les activités sont des petits projets réalisés par les étudiants et évalués par un système de peer-reviewing.

L'avantage de ce site est qu'il propose une grande variété de cours de qualité en français.

L'inconvénient majeur est qu'il n'est pas possible pour l'enseignant de suivre le progrès de ses étudiants ou de créer et personnaliser un cours pour ses besoins.

2.3 edX

edX⁴ est aussi un site de MOOC similaire à Open Classroom et d'autres sites de ce type. Une des différences est que ce dernier est multilingue et ne contient donc pas que des cours en français. Une autre différence est que l'écriture de cours sur ce site est ouverte à tous. De plus, il est possible de consulter les résultats des étudiants pour les cours que vous avez créés. Il existe aussi une plateforme open source, Open edX, permettant à chacun de créer sa propre instance d'edX, similaire à edX.org.

Finalement, ce site propose aussi d'utiliser des exercices provenant d'autres plateformes par le biais d'outils LTI⁵. Cet outil permet d'afficher un exercice provenant d'une autre plateforme, de réaliser les exercices et d'afficher les résultats. Par exemple, il est possible d'intégrer à un cours EDX un exercice provenant d'un exerciceur tel que la plateforme INGIInious.

Ce site est intéressant, car il permet d'intégrer d'autres types d'exercices, de suivre dans une certaine mesure la progression des étudiants et de créer son propre

3. <https://openclassrooms.com>

4. <https://www.edx.org>

5. <https://www.imsglobal.org/activity/learning-tools-interoperability>

cours.

Malheureusement, pour ce faire, il est nécessaire de créer entièrement son cours et ses exercices. Il n'est donc pas possible de réutiliser le travail produit par d'autres enseignants et il est nécessaire de fournir beaucoup d'efforts avant de réussir à produire un cours.

2.4 Moodle

Moodle⁶ est une plateforme d'apprentissage en ligne, c'est-à-dire une plateforme permettant de créer et partager des documents servant à l'apprentissage. Elle est cependant relativement différente des autres plateformes.

Chaque cours sur cette plateforme est composé de fichiers (PDF, vidéo ...) et d'exercices pouvant être séparés en sections. Les exercices prennent la forme de questionnaires ou de soumissions de fichiers. Les questionnaires sont à choix multiples ou à réponses prédéfinies et sont corrigés automatiquement. Les soumissions de fichiers, en revanche, sont évaluées de manière externe à la plateforme. Il est aussi possible d'utiliser des exercices provenant d'autres plateformes par le biais d'outils LTI.

Cette plateforme est généralement utilisée comme un espace numérique de travail (ENT). Il s'agit d'un espace pour centraliser les différents supports du cours et permettre aux étudiants de soumettre leurs devoirs, mais qui ne contient pas le cours en lui-même. Elle est donc utile pour rendre les supports du cours accessibles aux étudiants et organiser certains types d'évaluation.

Cependant, la plateforme ne permet pas de présenter un cours, mais plutôt de rassembler les supports pour ce cours.

2.5 Oscar

Oscar⁷ est un site d'enseignement en ligne pour étudiants de l'enseignement de primaire et secondaire de la Fédération Wallonie-Bruxelles.

Le principal atout de ce site est qu'il permet d'identifier les difficultés rencontrées par les étudiants. Il permet ensuite d'adapter leur parcours en fonction de leurs forces et faiblesses. Ainsi chaque étudiant peut évoluer selon le parcours qui lui convient le mieux. Tout cela est rendu possible grâce à un graphe des compétences pour la matière enseignée et l'identification claire des compétences évaluées par chaque exercice.

6. <https://moodle.org>

7. <https://oscar.education>

L'inconvénient majeur est que, pour le moment, ce site est limité aux cours de mathématique.

Ce mémoire ne se concentre pas sur la possibilité de fournir des parcours différents en fonction des étudiants. Si ce sujet vous intéresse, vous pouvez consulter le mémoire de Maxime Bénére [4].

2.6 Conclusion

Cette section compare tous les sites par rapport aux différentes fonctionnalités qu'ils proposent. Ces fonctionnalités correspondent aux besoins des enseignants mentionnés dans la section 1.4.

Outils mis à disposition des enseignants

	Code.org	Open Classrooms	EDX	Moodle	Oscar
Support pédagogique pour l'enseignant	✓	✗	✗	✗	✓
Suivi de la progression des étudiants	✓	✗	✓	✗	✓
Création de parcours personnalisés	✗	✗	✗	✗	✓

Édition de cours

	Code.org	Open Classrooms	EDX	Moodle	Oscar
Création de cours	✗	✗	✓	✓	✓
Création de cours à partir d'une base existante	✗	✗	✗ ⁸	✗ ⁹	✗
Cours d'informatique disponible	✓	✓	✓	✓	✗
Séparation en chapitre	✓	✓	✗	✓	✓

8. Il est possible pour les enseignants de dupliquer certains de leur cours, mais pas de réutiliser des cours mis à leur disposition par d'autres enseignants.

9. Voir note de bas de page précédente.

Évaluations

	Code.org	Open Classrooms	EDX	Moodle	Oscar
Évaluation par questionnaires	✓	✓	✓	✓	✓
Évaluation par projet auto évalué	✓	✗	✓	✗	✗
Évaluation par projet évalué de manière externe	✗	✓	✓	✓	✗

On remarque qu'aucun des sites ne répond entièrement à tous les besoins des enseignants du secondaire. Ceci montre le besoin d'une alternative permettant de mieux répondre aux besoins des enseignants et, en particulier, d'une plateforme rendant possible l'utilisation et la modification de bases de cours créées par d'autres enseignants. C'est ce que nous allons présenter dans ce travail.

Chapitre 3

Plateforme INGInious

La partie implémentation de ce mémoire consiste en une série d’extensions de la plateforme inginiuous. Ce chapitre présente la plateforme aux lecteurs qui ne seraient pas encore familiers avec cette dernière. Il contient une description du principe général de la plateforme, de son fonctionnement ainsi que des différents moyens de l’étendre. Si vous désirez déployer cette plateforme, un guide simplifié est disponible à l’annexe A.

3.1 Description de la plateforme

Inginious est une plateforme créée et maintenue par le département d’ingénierie informatique de l’UCLouvain. Elle est disponible librement sous licence GNU AGPL v3¹ sur la plateforme Github [7]. Il s’agit avant tout d’un exerciceur. Cela signifie que la plateforme permet de tester les compétences des étudiants. Cela se fait par le biais d’une série d’exercices et d’un système d’évaluation des solutions proposées par l’enseignant. La plateforme est principalement orientée vers l’informatique et permet la correction d’exercices de programmation. Ces exercices sont regroupés dans des cours. Les étudiants peuvent s’inscrire à ces cours et effectuer les exercices seuls ou en groupe. Les enseignants, quant à eux, peuvent gérer leurs cours, créer des exercices et suivre la progression des étudiants.

À première vue, la plateforme peut sembler complexe, mais un bon moyen de se familiariser avec celle-ci est de suivre le tutoriel disponible sur inginiuous.org [1]. Ce tutoriel présente les différentes fonctionnalités de la plateforme et la manière de créer un cours et des exercices.

1. <https://www.gnu.org/licenses/agpl-3.0.en.html>

3.1.1 Exercices

The screenshot displays the INginious platform interface for a 'Q* FizzBuzz' exercise. The top navigation bar includes the INginious logo, a breadcrumb trail ('Python 1 - Introduction à la programmation...' > 'Session 2: Q* FizzBuzz'), and links for 'Liste des cours' and 'Validation teacher'. The exercise title 'Q* FizzBuzz' is prominently displayed, along with a 'Masquer l'énoncé' button. The instructions describe the FizzBuzz game and the task: to write a program that takes an integer `i` and returns a string in `temp` based on divisibility rules. A list of requirements is provided: register 'fizz' for numbers divisible by 3, 'buzz' for numbers divisible by 5, 'fizzbuzz' for numbers divisible by both, and 'no' for numbers not divisible by either. Below the instructions is a code editor showing a solution. A green notification bar indicates that the user's response passed all tests with a score of 100.0%. On the right, an 'Informations' table shows details about the exercise, including the author (Tanguy De Bels), the date limit, the state (Succeeded), the score (100%), and the number of attempts (5). Below this, a 'Pour évaluation' section highlights the 'Meilleure soumission' (Best submission) with a score of 100.0% on 22/10/2019. A 'Historique des soumissions' (Submission history) table lists previous attempts with their dates, times, and scores.

Auteur(s)	Tanguy De Bels
Date limite	Pas de date limite
Etat	Succeeded
Note	100%
Poids de la note	1.0
Nombre d'essais	5
Limite de soumission	Pas de limite
Étiquettes de catégories	Session 2

Pour évaluation	
Meilleure soumission	
22/10/2019 12:13:02 - 100.0%	

Historique des soumissions	
21/10/2019 11:07:04	100.0%
20/10/2019 15:16:35	86.67%
16/10/2019 11:54:57	80.0%
15/10/2019 22:49:42	66.67%
14/10/2019 23:15:24	53.33%

FIGURE 3.1 – Exercice inginius

Les exercices sont au cœur de la plateforme inginius. Ils permettent aux professeurs de tester les compétences de leurs étudiants. Les solutions proposées par les étudiants sont testées et notées automatiquement, de manière sécurisée dans des conteneurs individuels (voir section 3.2). Un exemple d'exercice est présenté à la figure 3.1.

Il existe plusieurs types d'exercices disponibles sur inginius et il est possible d'en rajouter grâce aux extensions de la plateforme. Voici les principaux types proposés :

- **Code** : Il est demandé à l'étudiant d'écrire du code afin de réaliser une tâche précise. Pour ce faire, il dispose d'un éditeur de code basique intégré à la plateforme. Lors de la correction, ce code peut éventuellement être inséré dans un template et testé par le biais d'une série de tests définis par l'auteur de l'exercice. Une note est ensuite attribuée en fonction du résultat des tests.
- **Dépôt de fichier** : L'étudiant soumet un ou plusieurs fichiers pour évaluation. Il est alors possible pour les professeurs et assistants de les télécharger et de les évaluer manuellement. Si les fichiers contiennent du code, il est

également possible, comme précédemment, d'exécuter une série de tests afin de les évaluer automatiquement.

- **Correspondance** : L'étudiant doit donner une courte réponse textuelle. Celle-ci est ensuite comparée à la réponse attendue.
- **QCM** : Un classique questionnaire à choix multiple. Le professeur définit les réponses correctes. Celles-ci sont ensuite comparées aux choix de l'étudiant et une note est attribuée.

Pour tous ces exercices, il existe aussi un système de feedback permettant de guider l'étudiant de manière personnalisée. Enfin, d'autres réglages permettent de définir :

- Le nombre de soumissions pouvant être envoyées par l'étudiant
- Quand l'étudiant a accès à l'exercice
- Quelle soumission est évaluée
- Les catégories de l'exercice
- ...

3.1.2 Cours

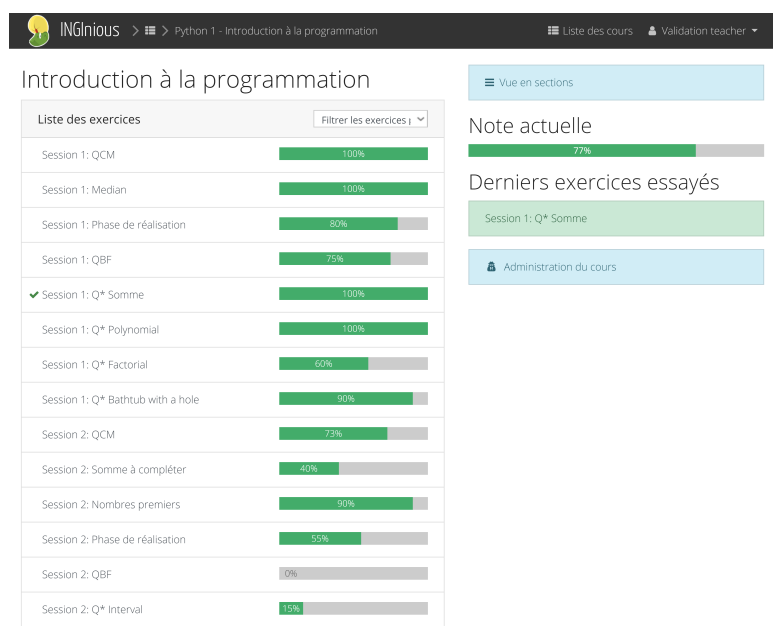


FIGURE 3.2 – Page d'accueil d'un cours

Un cours est un recueil d'exercices permettant d'évaluer les étudiants sur une matière spécifique. Un exemple d'exercice est présenté à la figure 3.2. Chaque cours est géré par un ou plusieurs enseignants. Ceux-ci peuvent définir plusieurs

Le schéma de la figure 3.3 représente l'architecture d'inginius [2]. La plateforme est implémentée en Python et les utilisateurs peuvent s'y connecter par le biais d'un navigateur internet. Elle utilise le système de fichier du serveur pour stocker les exercices et les cours. En ce qui concerne les données des utilisateurs, celles-ci sont stockées sur une base de données MongoDB. La plateforme communique aussi avec les agents afin d'évaluer les soumissions (voir section 3.2.2).

3.2.1 Plateforme

La plateforme est écrite en Python dans un style orienté objet. Elle est divisée en plusieurs parties correspondant chacune à un répertoire.

Tout d'abord, la partie **common** contient les blocs de base utilisés par les autres modules. Par exemple, c'est là que l'on retrouve les blocs de base pour les cours, les exercices, la gestion des fichiers, les exceptions . . .

Le backend est la partie de la plateforme qui envoie les solutions des étudiants pour évaluation. Elle s'occupe de toutes les interactions avec l'agent inginius du Docker. Elle est indépendante du frontend de telle sorte que celui-ci puisse être remplacé.

Le frontend utilise Python avec le framework webpy ainsi que les technologies web classiques : HTML, CSS avec Bootstrap et Javascript avec jquery. Il définit l'entièreté de l'interface utilisateur et est interchangeable. Il est donc possible de réimplémenter cette partie et de l'utiliser avec le reste d'inginius.

Le diagramme de classe présenté à la figure 3.4 décrit les classes les plus importantes dans le cadre de ce mémoire. Il utilise les conventions enseignées à l'UCLouvain [11]. Lorsqu'elles ne sont pas mentionnées explicitement, les multiplicités sont uniques, les variables d'instance sont privées et les méthodes sont publiques. Les classes et les méthodes abstraites sont représentées en italique et les dictionnaires sont représentés à l'aide du qualificatif².

Le **FileSystemProvider** est une abstraction d'un dossier et propose plusieurs méthodes pour manipuler le système de fichier. Il s'agit d'une classe abstraite dont une implémentation concrète est donnée par **FileSystemLocal**.

Les **CourseFactory** et **TaskFactory** permettent de créer les objets pour les cours et les exercices à partir du système de fichier et contiennent la liste de ces objets. Ils fournissent aussi les méthodes pour modifier ces cours et exercices.

Les classes **Course**, **Task** et **Problem** représentent respectivement un cours, un exercice et un sous-problème. Elles contiennent principalement des données et leurs accesseurs. Les classes **Task** et **Problem** contiennent aussi des fonctions pour évaluer une soumission.

2. <https://www.uml-diagrams.org/property.html#qualifier>

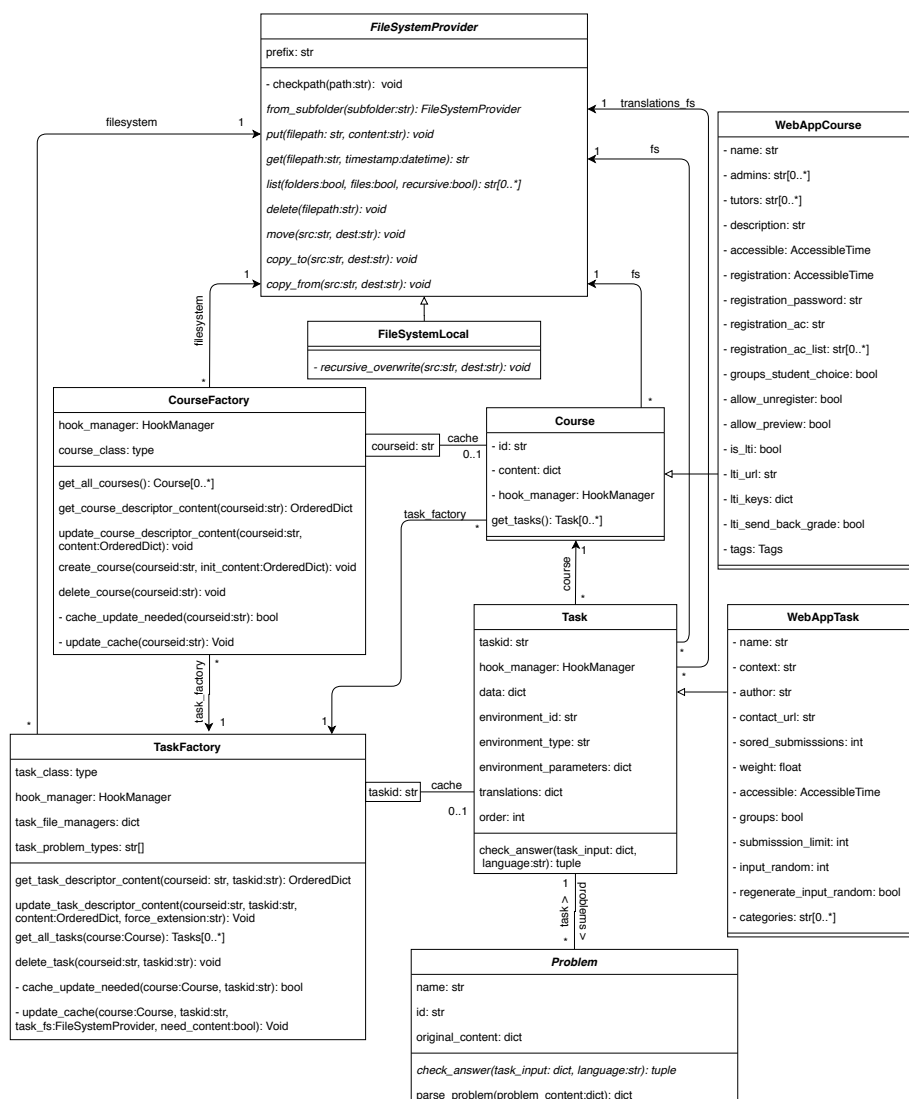


FIGURE 3.4 – Diagramme de classe de la plateforme inginius

Les classes **WebAppCourse** et **WebAppTask** étendent les classes **Course** et **Task**. Elles fournissent des accesseurs plus avancés pour le frontend (traduction, mise en forme ...) et stockent les différentes informations des dictionnaires dans des variables d'instance spécifiques.

3.2.2 Docker

Lorsqu'un étudiant envoie ses réponses pour un exercice, une soumission est créée. Cette soumission contient la solution de l'étudiant et le script pour l'évaluer.

Pour évaluer les soumissions inginius utilise le logiciel Docker. Un agent est en charge de recevoir les soumissions, les évaluer et renvoyer les résultats. Pour ce faire, ce dernier crée un conteneur logiciel spécifique pour chaque soumission. Un conteneur est un environnement logiciel isolé permettant d'exécuter les soumissions des étudiants de façon indépendante et sécurisée. Il est possible d'utiliser plusieurs agents afin de séparer la charge de travail sur plusieurs serveurs.

Étant donné que ce mémoire ne modifie pas cette partie d'inginius, nous ne nous attarderons pas ici sur son fonctionnement. Plus de détails sont disponibles dans la documentation [2].

3.2.3 Système de fichiers

Le système de fichiers du serveur est utilisé pour stocker plusieurs données de la plateforme.

Abordons tout d'abord le fichier de configuration au format yaml. Ce fichier contient différentes informations sur l'instance telles que l'emplacement des cours et backup et la liste des identifiants des administrateurs. Pour la liste complète, consultez la documentation³.

Les fichiers relatifs aux cours sont stockés dans un dossier spécifié dans le fichier de configuration. Ce dossier respecte le format suivant présenté à la figure 3.5 :

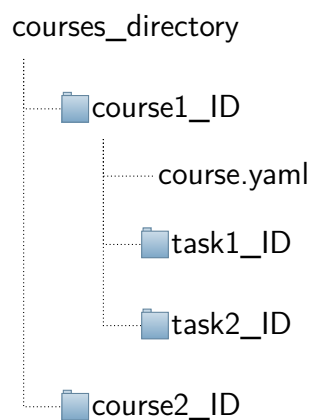


FIGURE 3.5 – Structure de fichier des cours

À chaque cours correspond un dossier dont le nom est l'identifiant du cours. Dans ce dossier se trouve un fichier **course.yaml**. Celui-ci décrit le cours avec par exemple le titre du cours, sa description, la liste de ses administrateurs ... Il contient aussi un dossier pour chaque exercice.

3. https://inginius.readthedocs.io/en/v0.6/install_doc/config_reference.html

Ce dossier a comme nom l'identifiant de l'exercice et la structure présentée à la figure 3.6 :

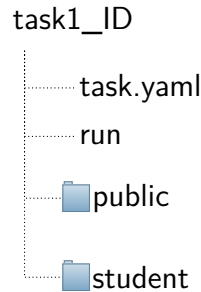


FIGURE 3.6 – Structure de fichier d'un exercice

Il contient au minimum le fichier **task.yaml** décrivant l'exercice. Celui-ci regroupe plusieurs informations telles que le titre de l'exercice, sa description en **restructured text**, l'auteur ... Il peut aussi contenir un script **run** qui est exécuté pour évaluer une soumission. Le dossier **public** contient les fichiers accessibles depuis le frontend et pouvant être utilisés dans la description (images, documents fournis aux étudiants ...). Le dossier **student** est utilisé pour isoler les soumissions des étudiants afin de l'exécuter avec l'API **run__student**. Pour plus de détails, consulter le tutoriel [1].

3.2.4 Base de données

La base de données MongoDB stocke toutes les données relatives aux utilisateurs. MongoDB est un système de base de données NOSQL basé sur des documents⁴. Grâce à la fonction de réplication de MongoDB, il est possible de configurer inginius avec plusieurs frontend⁵.

3.3 Étendre la plateforme

Cette section présente les deux méthodes disponibles pour étendre inginius ainsi qu'une discussion sur leur pertinence dans le cadre de ce mémoire.

4. <https://www.mongodb.com>

5. https://inginius.readthedocs.io/en/v0.6/dev_doc/understand_inginius.html#scalability-of-the-inginius-frontend

3.3.1 Modifier la plateforme

Inginious étant une plateforme open source, il est possible de participer à son développement. Ceci se fait en effectuant une pull request sur le répertoire Github [7]. Si celle-ci est acceptée, les changements sont alors intégrés à la plateforme.

Ces changements sont donc appliqués à toutes les instances d'inginius à jour. Il faut donc être sûr de leur validité avant de les intégrer. C'est pour cette raison qu'il est préférable de passer d'abord par une phase intermédiaire. Nous implémenterons donc des plug-ins qui pourront ensuite être intégrés une fois validés.

Cette technique sera en revanche utilisée afin de généraliser certains éléments d'inginius et éviter la duplication de code dans les plug-ins.

3.3.2 Plug-ins

Finalement, il est aussi possible de créer des plug-ins à la plateforme. Un plug-in est un module que l'on peut choisir d'intégrer ou non à la plateforme.

La principale différence est qu'un plug-in ne modifie pas le code d'inginius. L'entièreté du plug-in est contenue dans un dossier spécifique. Une méthode du plug-in est appelée au démarrage d'inginius et les ajouts sont apportés par le biais d'appels à une API spécifique aux plug-ins. Pour plus de détails sur le fonctionnement des plug-ins, consultez la documentation⁶.

Il existe principalement deux inconvénients à cette technique. Le premier est d'être limité à ajouter des éléments définis par l'API relative aux plug-ins. Malheureusement, celle-ci propose uniquement une liste finie d'éléments et elle ne permet pas toujours d'ajouter ou modifier ce que l'on souhaite. Le deuxième est que sans la possibilité de modifier le corps d'inginius, la duplication de code est parfois inévitable. En effet, lorsqu'une fonctionnalité présentant des similarités avec un module de la plateforme est implémentée, il n'est pas possible de généraliser ce module afin de réutiliser les parties similaires.

L'avantage est qu'il est possible de sélectionner ou non les plug-ins d'une instance d'inginius à l'autre et que le code du plug-in est séparé du reste d'inginius.

Dans un premier temps, c'est cette technique qui a été choisie pour ce mémoire en combinaison avec quelques modifications d'inginius pour éviter la duplication de code. Ensuite, si ces plug-ins sont validés et estimés utiles sur toutes les instances, il sera possible de les intégrer directement dans inginius. Lors du développement, il a donc fallu à la fois penser à comment implémenter chaque plug-in et à comment ceux-ci pourraient être intégrés au reste de la plateforme par la suite.

6. https://inginius.readthedocs.io/en/v0.6/dev_doc/plugins.html

Chapitre 4

Organisation

4.1 Approche générale

Durant la première phase de ce mémoire, nous nous sommes concentrés sur l'analyse des besoins. Nous avons essayé de définir exactement les besoins des enseignants qui ne pouvaient pas être comblés par les plateformes existantes. Pour ce faire, nous avons interrogé un enseignant d'informatique du niveau secondaire afin de clairement définir ses besoins. Ensuite, nous avons consulté les différentes plateformes mises à disposition des enseignants. Cela nous a permis d'identifier les besoins qui n'étaient pas comblés par ces plateformes, mais aussi d'observer les éléments intéressants apportés par chacune de ces plateformes.

Une fois ces besoins identifiés, nous avons cherché à trouver des solutions qui permettraient de répondre au mieux à ces derniers. La première approche explorée consistait à créer une nouvelle plateforme utilisant INGINious pour les exercices. Elle est décrite au chapitre 5. Nous avons commencé à nous familiariser avec le fonctionnement d'INGInious et à modéliser notre solution. Un modèle pour l'architecture de la plateforme ainsi que quelques mock-ups ont été créés.

Cependant, cette approche n'a malheureusement pas pu être implémentée. À la place de créer une nouvelle plateforme, nous avons donc décidé d'étendre la plateforme INGINious. Cette approche est décrite au chapitre 3. Nous avons donc continué à nous renseigner sur le fonctionnement de la plateforme afin d'être capables d'implémenter des plug-ins.

Pour chacun des plug-ins, nous avons commencé par une phase de modélisation. Durant cette phase, nous avons défini l'architecture du plug-in et ses interactions avec le reste de la plateforme. Nous avons aussi créé des diagrammes de classes lorsque cela semblait utile. Ensuite, nous avons entamé l'implémentation du plug-in en suivant la méthodologie décrite à la section 4.3.

Une fois que tous les plug-ins ont été implémentés, nous avons **rebase**¹ le répertoire afin de rendre propre et structuré l’historique des commits. La figure 4.1 présente un exemple de l’effet d’un tel **rebase**. Chaque commit correspond à une fonctionnalité indépendante et la plus petite possible. Cette étape est très importante, car elle est nécessaire à l’intégration des plug-ins à la plateforme INGINious.

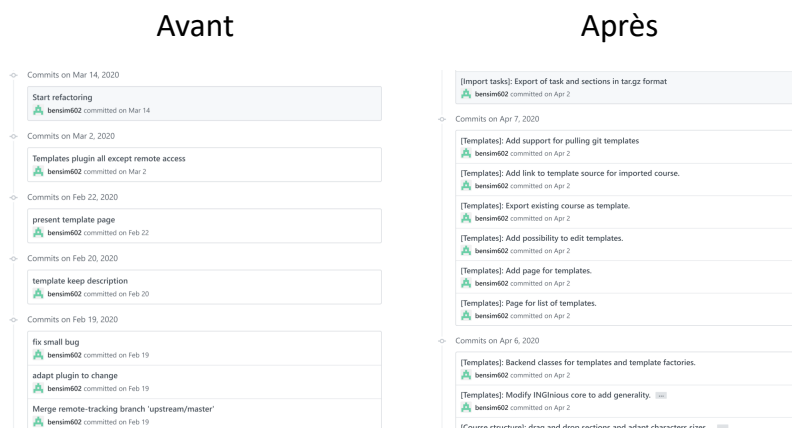


FIGURE 4.1 – Exemple de commits rebase

Finalement, nous avons mis la plateforme dans les mains de vrais utilisateurs. Nous leur avons demandé de réaliser une série d’opérations utilisant les nouvelles fonctionnalités. Ceci a été fait en entretien virtuel et les utilisateurs ont ensuite rempli un formulaire. En fonction des retours obtenus durant l’entretien et dans le formulaire, nous avons pu identifier les points forts et les problèmes des plug-ins et nous avons adapté certains éléments.

4.2 Planning

La figure 4.2 présente le planning pour la réalisation de ce mémoire sous la forme d’un diagramme de Gantt. Ce planning a évidemment évolué au cours de l’année, mais cette figure présente la version finale. La partie supérieure du diagramme présente les différentes approches utilisées dans ce mémoire et leur décomposition. La partie inférieure, quant à elle, présente les différentes phases de l’implémentation, celles-ci sont reprises plus en détail dans la section 4.1. On peut remarquer qu’aucune analyse des besoins n’a été effectuée pour les plug-ins. Cela

1. <https://git-scm.com/docs/git-rebase>

est dû au fait que nous avons pu réutiliser celle faite pour la plateforme EZLearn, car elle répond aux mêmes besoins.

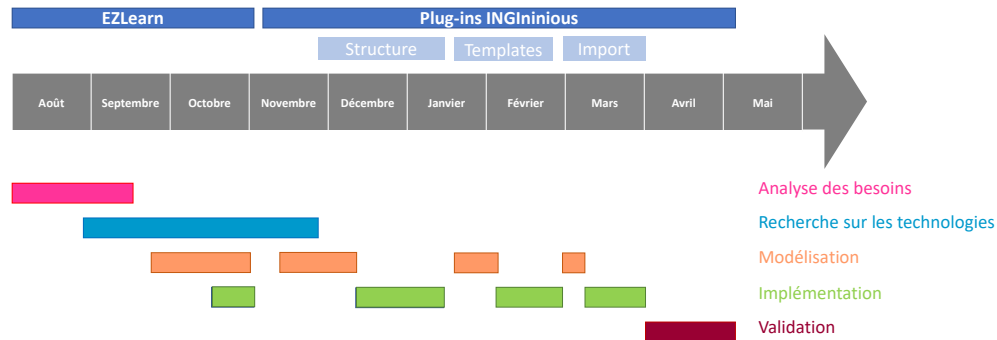


FIGURE 4.2 – Diagramme de Gantt de l'organisation du mémoire

4.3 Méthodologie

Pour l'implémentation des différents plug-ins, nous avons décidé de nous inspirer de la méthodologie Agile et plus particulièrement de Scrum [14]. Certains aspects de cette méthodologie sont spécifiques pour une équipe d'implémentation, ce qui n'est pas le cas pour ce mémoire, cependant certains aspects restent intéressants dans le cadre de ce travail.

Cette méthodologie est flexible aux changements, elle permet d'adapter le développement de semaine en semaine, en fonctions des retours et des changements éventuels.

4.3.1 Itérations

Une des spécificités de cette méthodologie est le fait de fonctionner par courtes itérations (sprint). Dans le cadre de ce mémoire, chaque itération dure une semaine. Cela permet de débriefer le sprint pendant la réunion hebdomadaire avec les promoteurs. Nous prenons ensuite ce débriefing en compte afin de définir ce qui sera réalisé lors du prochain sprint.

4.3.2 Backlog

Dans Scrum, le backlog contient une liste des fonctionnalités à implémenter et éléments à corriger accompagnée d'une estimation de la durée et de l'importance afin de les ordonner entre eux. Pour ce mémoire, cette liste correspond au Trello

(voir section 4.4.1). Cependant, les fonctionnalités étant généralement dépendantes les unes des autres, l'ordre s'établit assez naturellement. Nous avons donc décidé de ne pas ajouter une estimation du temps et de priorité, mais uniquement de marquer les tâches urgentes.

4.4 Outils utilisés

4.4.1 Trello

Nous utilisons Trello² afin de visualiser les tâches à réaliser et organiser le travail. Un tableau a été créé pour chacune des phases du mémoire. Ces tableaux contiennent la liste des fonctionnalités à implémenter, sous la forme de user stories [11] ainsi que des éléments à améliorer ou corriger. Les éléments urgents sont marqués en rouge. Un exemple simplifié d'un tableau est présenté à la figure 4.3.

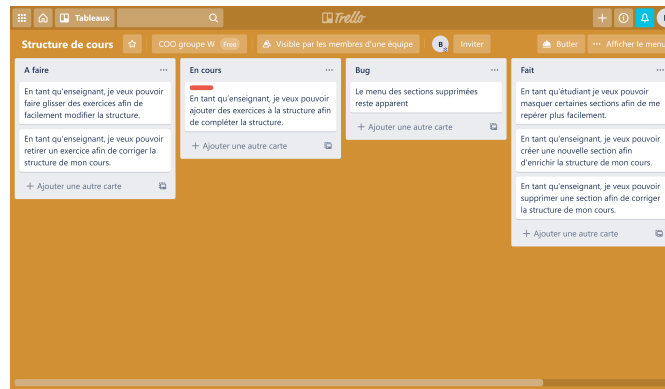


FIGURE 4.3 – Tableau Trello simplifié pour l'organisation de l'implémentation de la structure d'un cours

4.4.2 Github

Pour ce mémoire, nous avons voulu collaborer à la plateforme INGINIOUS. Pour ce faire, il est nécessaire d'effectuer une **pull request** sur le répertoire Github d'INGInious [7]. Nous avons donc hébergé notre projet sur cette plateforme et fait des commits soignés afin de faciliter la relecture de notre pull request.

2. <https://trello.com>

Chapitre 5

Plateforme EZLearn

Durant la première phase de ce mémoire, nous avons exploré la possibilité d’implémenter une plateforme permettant de faciliter l’apprentissage, EZLearn. Malheureusement, celle-ci n’a pas pu être implémentée, mais une analyse du problème a pu être réalisée. C’est cette analyse que nous présentons dans ce chapitre. Elle servira par la suite pour l’implémentation de nos plug-ins et pourrait éventuellement servir de base pour d’autres travaux.

Cette plateforme fonctionnerait comme une couche supplémentaire au-dessus de la plateforme inginius. Cette nouvelle couche permettrait de créer des modules théoriques contenant du texte et des vidéos tandis que les exercices seraient gérés par la plateforme inginius. Elle permettrait aussi de créer et éditer des cours contenant des exercices inginius. D’autre part, ezlearn permettrait de facilement dupliquer un cours pour plusieurs classes ou de se servir de cours template pour créer son propre cours. Elle fournirait aussi certains outils permettant de visualiser la progression des étudiants. Finalement, ezlearn présenterait l’avantage d’être conçu pour fonctionner avec une seule instance pour toutes les écoles et de supporter le trafic engendré par ce choix de design. Ceci permettrait d’éliminer la contrainte forçant chaque école à avoir sa propre instance sur un serveur propre.

5.1 Architecture

Le schéma présenté à la figure 5.1 présente l’architecture de la plateforme ezlearn. Il est en partie basé sur le schéma de l’architecture de la plateforme inginius¹.

Ezlearn ainsi que l’instance d’inginius se trouvant sur le même serveur, les deux plateformes partagent le même système de fichiers. Ceci permet à ezlearn de créer et modifier des cours sur l’instance d’inginius.

1. https://inginius.readthedocs.io/en/v0.6/dev_doc/understand_inginius.html

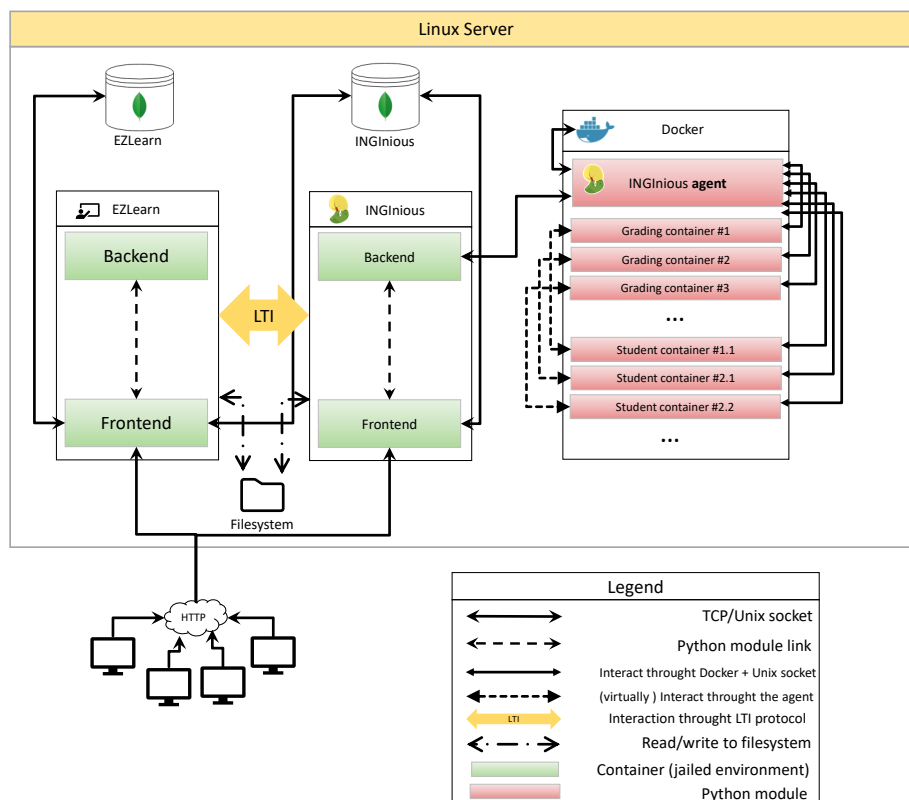


FIGURE 5.1 – Architecture de la plateforme ezlearn

La récupération des données de progression des étudiants est possible en accédant directement à la base de données d'inginius ou par le biais du protocole LTI. Pour le reste, elle utiliserait les mêmes technologies qu'inginius afin de maximiser l'interopérabilité avec cette plateforme.

5.2 Fonctionnalités

5.2.1 Éditer un cours

Un cours ezlearn serait composé de plusieurs modules soit de type théorique, soit de type pratique. Ces modules pourraient éventuellement être organisés en chapitres.

Un module théorique est un module ayant pour but d'expliquer un point de matière précis. Il se présente sous la forme d'une page contenant du texte et éventuellement des vidéos et des images. Afin de créer et modifier ces modules, la plateforme proposerait un éditeur permettant d'écrire et modifier le texte, le

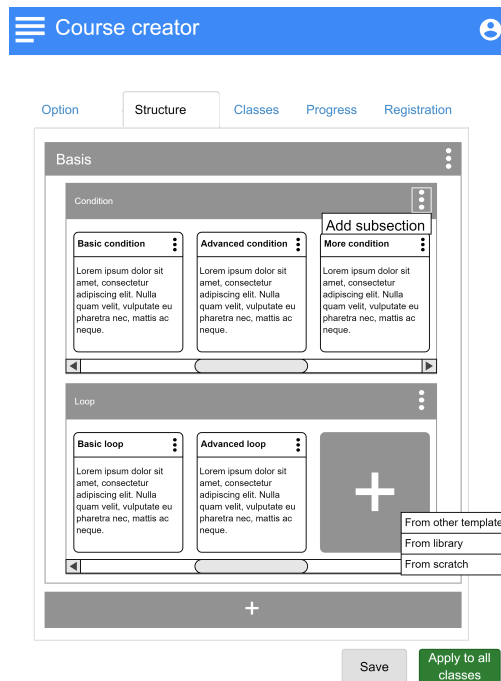


FIGURE 5.2 – Éditer la structure d'un cours ezlearn

mettre en forme et insérer des images, des vidéos et des liens hypertextes.

Un module pratique, en revanche, a pour but d'évaluer si les connaissances ont été bien acquises par l'étudiant. Ces modules correspondent à des exercices inginius. Deux méthodes seraient mises à disposition de l'enseignant pour créer de tels modules. La première serait de réutiliser un exercice existant et disponible en libre accès. L'autre méthode consiste à écrire entièrement un nouvel exercice comme cela est possible sur la plateforme inginius.

Concernant la décomposition en chapitres et l'édition de la structure, celle-ci se ferait principalement par un système de glisser-déposer. Cette méthode a été choisie dans le but de rendre le processus le plus intuitif possible. Cette page est présentée à la figure 5.2

Lorsqu'un cours ezlearn est créé ou modifié, un cours inginius correspondant est aussi créé ou modifié. Ce dernier comprend tous les exercices présents dans un module du cours ezlearn.

5.2.2 Utilisation des templates

La fonctionnalité précédente permet de créer un cours à partir de rien. Cependant dans beaucoup de cas il peut s'avérer utile de partir d'une base. Pour ce faire, ezlearn introduit la notion de template. Un template est un cours modèle sur lequel

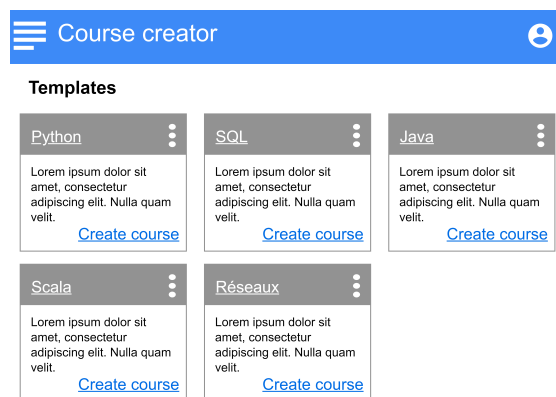


FIGURE 5.3 – Catalogue de template ezlearn

on se base pour créer un cours. Il se présente comme un cours sans étudiants ou enseignants. De plus, il contient des descriptions et listes de compétences pour le cours, ses chapitres et éventuellement ses exercices. Ces descriptions ont pour but de présenter le cours aux enseignants, les guider dans l’enseignement du cours et les aider à identifier les compétences qui seront abordées.

Un catalogue présente les différents templates disponibles aux enseignants. Ceux-ci peuvent être utilisés afin de créer un cours contenant les chapitres et modules du template. Une fois le cours créé il est alors possible pour l’enseignant de le modifier comme il le souhaite. Un rendu possible de ce qu’aurait pu être ce catalogue est présenté à la figure 5.3

5.2.3 Gestion des utilisateurs

Afin d’utiliser la plateforme, les utilisateurs doivent se connecter. Plusieurs niveaux de privilèges sont définis.

Tout d’abord l’administrateur système. Ce dernier a le niveau maximum de privilèges. Il peut supprimer et modifier n’importe quel cours ou utilisateur de la plateforme.

Les enseignants correspondent à un autre niveau de privilèges. Ceux-ci ont le droit de créer des cours et de modifier les cours qu’ils ont créés. Pour les cours qu’ils ont créés, ils peuvent aussi consulter le progrès des étudiants et ajouter de nouveaux étudiants. Finalement, ils ont aussi accès à une vue étudiante leur permettant de consulter le cours comme il est vu par les étudiants.

Le dernier type d’utilisateur correspond aux étudiants. Un étudiant peut s’inscrire à des cours, suivre les différents modules et consulter sa progression. Les différents niveaux et leurs privilèges sont récapitulés dans le tableau suivant.

	Étudiant	Enseignant	Administrateur
S'inscrire à un cours	✓	✓	✓
Suivre un cours	✓	✓	✓
Créer et modifier ses cours	✗	✓	✓
Consulter le progrès des étudiants	✗	✓	✓
Supprimer ou modifier n'importe quel cours	✗	✗	✓
Supprimer ou modifier des étudiants	✗	✗	✓

Tous ces types utilisateurs ont un compte protégé par un mot de passe. Ce compte contient plusieurs informations qui peuvent être modifiées par l'utilisateur (par exemple le nom, prénom, numéro d'identification ...).

En ce qui concerne les comptes enseignants, il est important de constater que permettre à n'importe qui de créer ce type de compte présente un risque. En effet, ces comptes peuvent créer des cours qui prennent de la place sur le serveur et encombrant la liste des cours disponibles. Il peut donc être utile de faire valider ces comptes par un administrateur ou par un système de vérification de l'identité des enseignants (par exemple un numéro d'identification de l'enseignant).

5.2.4 Établissements

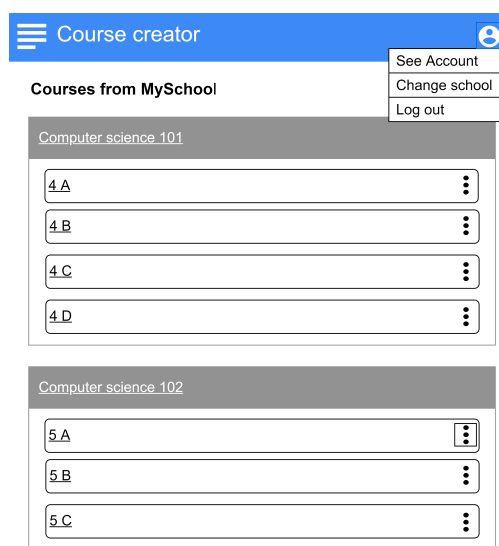


FIGURE 5.4 – Page d'accueil ezlearn

Un des avantages de cette plateforme est qu’une seule instance est nécessaire pour toutes les écoles qui l’utilisent. Cependant, si la plateforme commence à être beaucoup utilisée, il deviendra plus difficile de se repérer parmi tous les cours.

Pour des établissements du secondaire ayant un nombre limité de cours, ce concept d’établissements permet de résoudre ce problème. Chaque compte et cours serait lié à un ou plusieurs établissements. Ainsi les enseignants peuvent uniquement créer des cours liés à leur établissement et les étudiants ont uniquement accès aux cours de leur établissement. Comme on peut le voir à la figure 5.4, cela limite les cours affichés à l’étudiant. Il est aussi possible de rendre certains cours accessibles par tous les étudiants.

5.2.5 Progression des étudiants

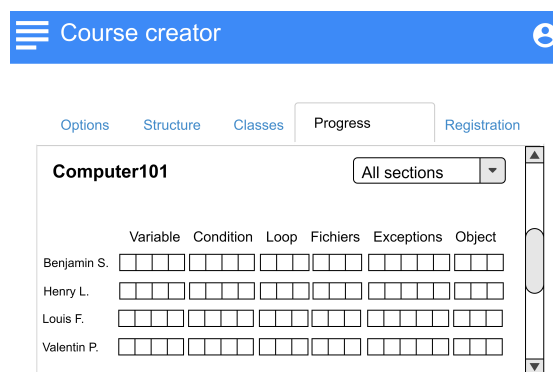


FIGURE 5.5 – Progrès des étudiants ezlearn

Pour les enseignants, un aspect important d’une telle plateforme est de pouvoir consulter le progrès de leurs étudiants. C’est pourquoi ezlearn propose aux enseignants un tableau de bord permettant de visualiser le progrès des étudiants inscrits à leurs cours.

Celui-ci prend la forme d’un tableau pour lequel chaque ligne représente la progression d’un étudiant et chaque colonne représente un module. Chaque élément du tableau est un carré coloré en vert selon le même pourcentage que le score obtenu par l’étudiant pour ce module. Ce tableau est disponible pour l’ensemble du cours ou pour un chapitre en particulier.

La figure 5.5 présente cette page

5.3 Défauts de l'approche

Nous avons eu une discussion avec deux employés du pôle informatique de l'UCLouvain, Anthony Gego qui travaille sur la plateforme inginius et François Michel qui a travaillé sur le syllabus interactif² utilisant inginius par le biais de l'interface LTI. Après cette discussion et une analyse plus approfondie de ce qu'il serait nécessaire d'implémenter, cette approche a été abandonnée. En effet, la charge de travail nécessaire pour implémenter correctement toutes ces fonctionnalités se serait avérée être trop importante dans le cadre de ce mémoire. Nous avons donc décidé à la place d'enrichir la plateforme inginius par une série de plug-ins.

Cependant, nous avons estimé intéressant de présenter cette solution, car malgré le fait qu'elle ne puisse pas être implémentée dans le cadre de ce mémoire, elle pourrait, malgré tout, servir de base pour un travail plus conséquent. De plus, certains éléments se sont avérés utiles dans l'implémentation des plug-ins. Le tableau ci-dessous présente l'état d'intégration dans inginius des fonctionnalités présentées dans ce chapitre.

	État de la fonctionnalité dans inginius après ce travail
Éditer un cours	Présent de base dans inginius. Le principe d'édition de la structure des cours a été réutilisé dans le plug-in Structure du cours présenté à la section 6.1.
Utilisation des templates	Ajouté à inginius avec le plug-in Template présenté à la section 6.3.
Gestion des utilisateurs	Géré de base par inginius. La possibilité d'ajouter le niveau de privilège enseignant est présentée dans les améliorations futures à la section 8.2.
Établissements	Non implémenté sur inginius. Présenté dans les améliorations futures à la section 8.3.
Progression des étudiants	Présent de base sur inginius. Cependant, il manque la possibilité de visualiser la progression globale. Ceci est présenté dans les améliorations futures à la section 8.4.

2. <https://syllabus-interactif.info.ucl.ac.be>

Chapitre 6

Solution

La plateforme INGIInious décrite dans le chapitre 3 offre beaucoup de fonctionnalités pour évaluer les connaissances informatiques des étudiants. Cependant, elle ne répond pas à tous les besoins des enseignants du niveau secondaire. Afin d’essayer de résoudre ce problème, nous allons étendre la plateforme par une série de plug-ins visant chacun à répondre à certains de ces besoins. Dans le cadre de ce mémoire, trois plug-ins ont été implémentés. Le premier permet de structurer et organiser un cours grâce à un système de chapitres et sections. Le deuxième permet de créer un cours à partir d’une base existante appelée template. Enfin, le dernier permet d’importer et d’exporter des exercices afin de facilement réutiliser le travail d’autres enseignants. Chacune des sections suivantes présente un de ces plug-ins.

6.1 Structure du cours

Ce plug-in permet de structurer un cours en sections. Un exemple d’une telle structure est donné à la figure 6.1.

6.1.1 Fonctionnalités

Interface étudiant

Du point de vue de l’étudiant, et avant que cette extension ne soit implémentée, chaque cours pouvait être vu comme une liste d’exercices. Lorsqu’un étudiant consultait un cours, une liste d’exercices ainsi que sa progression lui était présentée. Cependant, pour un cours contenant beaucoup d’exercices, il peut devenir compliqué de se repérer parmi les exercices et de comprendre dans quel cadre chacun d’eux s’inscrit.

Ce nouveau plug-in a pour objectif de résoudre ce problème. Pour ce faire, il ajoute un lien vers une vue en section sur le menu principal du cours. Un exemple

de cette vue en section est présenté à la figure 6.1. Elle présente la liste des exercices, cette fois séparés sections. Afin de mieux se repérer dans la structure, il est possible de masquer le contenu d’une section grâce à la flèche à droite de celle-ci.

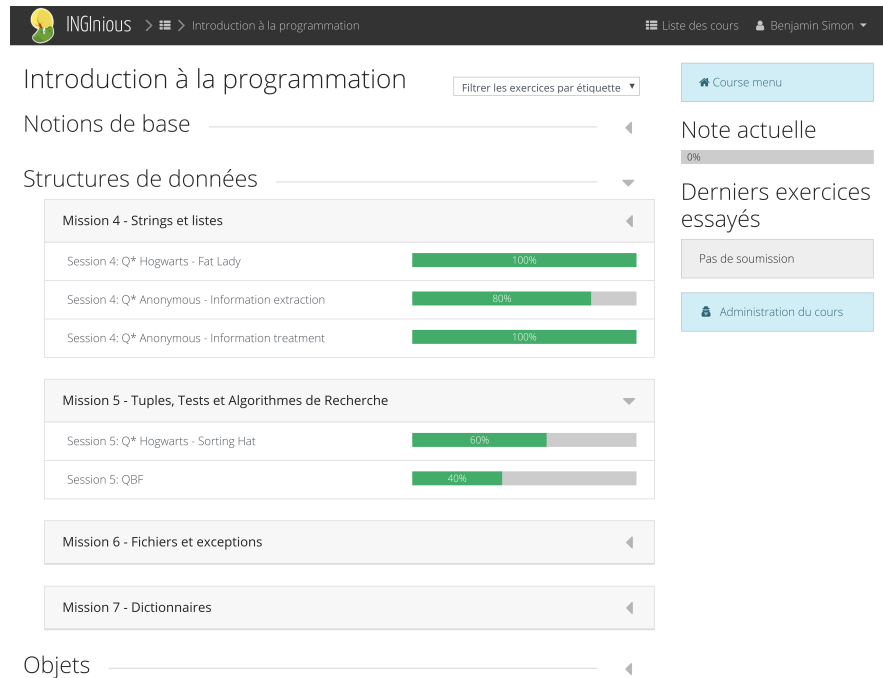


FIGURE 6.1 – Vue en sections d’un cours

Chaque section est composée d’un ensemble d’éléments parmi les 2 natures suivantes :

- un ensemble de type terminal (dont tous les éléments sont des exercices)
- un ensemble de type intermédiaire (dont tous les éléments sont de type terminal et/ou intermédiaire)

Pour plus de détails sur l’organisation de cette structure, consultez la figure 6.4.

Afin de s’assurer que ce plug-in soit utilisable avec d’anciens cours, il est possible qu’aucune structure ne soit définie. Dans ce cas, tous les exercices sont affichés dans une seule section, de manière similaire à ce qui était fait avant cette extension.

Pour l’implémentation du plug-in, cette page a été ajoutée comme un page supplémentaire. Cependant, l’objectif est que cette page remplace la page principale lors de l’intégration à la plateforme.

Éditer un cours

Pour que cette fonctionnalité soit utile, il est nécessaire de permettre aux enseignants de créer la structure du cours. Pour ce faire, nous avons créé un éditeur

pour cette structure. Cet éditeur est disponible dans l'administration du cours et un exemple est présenté à la figure 6.2. Celui-ci propose les différentes fonctionnalités listées ci-dessous.

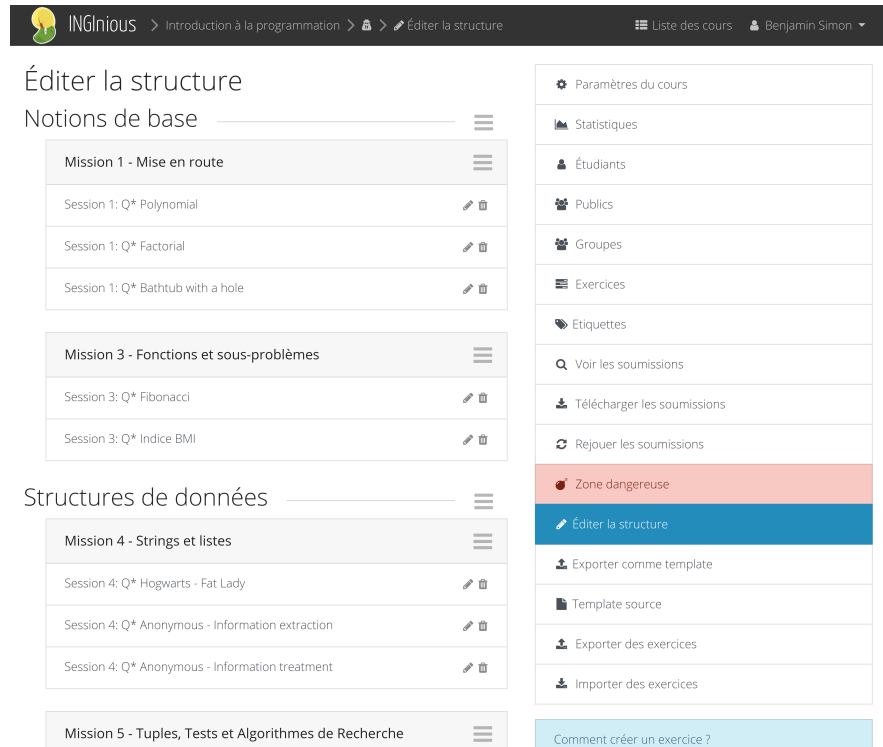


FIGURE 6.2 – Éditer la structure d'un cours

- **Glisser-déposer** : Il est possible de déplacer des exercices et des sections grâce au glisser-déposer. Les exercices peuvent être glissés dans n'importe quel ensemble de type terminal ou vide. Les sections peuvent être glissées dans n'importe quel ensemble de type intermédiaire ou vide. Lorsqu'un élément est glissé, la structure s'adapte pour refléter les changements.
- **Créer une section** : Il est possible de créer une nouvelle section vide soit à la base de la structure, soit dans un ensemble de type intermédiaire ou vide.
- **Ajouter des exercices** : Il est possible d'ajouter des exercices à un ensemble de type terminal ou vide. Lors de l'ajout d'exercices, une fenêtre s'ouvre présentant tous les exercices qui ne sont pas encore intégrés dans la structure. Il est alors possible de faire une recherche et de sélectionner tous les exercices que l'on souhaite ajouter à l'ensemble.
- **Supprimer des éléments** : Il est possible de supprimer des exercices et des sections de la structure. Il est important de noter que les exercices sont

- simplement retirés de la structure, mais pas supprimés du cours.
- **Renommer des éléments** : Il est aussi possible de renommer des sections ou des exercices.

6.1.2 Implémentation

Stockage des données

Les données relatives à la structure d'un cours sont liées au cours en lui-même et non aux utilisateurs. Conformément à ce qui est expliqué dans la section, 3.2.3 elles sont donc stockées dans le système de fichiers. Un fichier **toc.yaml** à la base du dossier de chaque cours permet de stocker cette structure. La structure du dossier contenant les cours correspond donc maintenant à la figure 6.3.

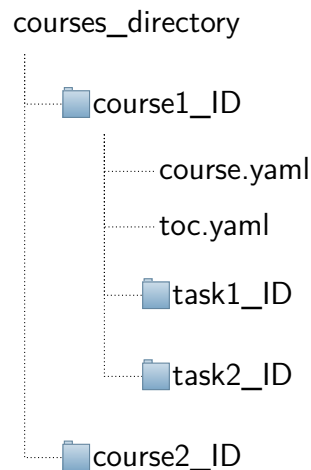


FIGURE 6.3 – Structure de fichiers des cours

Cependant, le fichier **toc.yaml** est optionnel et la plateforme reste compatible avec les cours ne contenant pas ce fichier. Afin de pouvoir être reconnu, le fichier **toc.yaml** doit respecter une structure stricte décrite à la figure 6.4.

```

Toc ::= [Section, ..]
Section ::= {"id": ID,
            "rank": Rank,
            "title": Title,
            "sections_list": [Section, ..] | tasks_list: Tasks}
Tasks ::= {TaskID: Rank, ..}
  
```

FIGURE 6.4 – Pseudo structure du fichier toc.yaml

Ce fichier se compose d'une liste de sections. Chaque section est représentée par un dictionnaire contenant plusieurs éléments. Le **title** correspond au titre qui est affiché pour la section, l'**ID** à l'identifiant unique de la section et le **rank** à la position de la section. Dans le cas d'un ensemble de type intermédiaire, **sections_list** est associé à la liste des sous-sections. Pour les ensembles de type terminal, **tasks_list** correspond à un dictionnaire contenant les identifiants des exercices de la section. Les clés de ce dictionnaire sont les identifiants des exercices et la valeur associée correspond à la position de l'exercice. Un exemple d'un tel fichier est fourni à l'annexe B.

Accès aux données

Plusieurs approches ont été envisagées afin de permettre de récupérer et de modifier la structure. Tout d'abord, l'approche orientée objet consiste à étendre la classe **Course** par une classe contenant une variable d'instance correspondant à la structure et des méthodes permettant de la manipuler. Il est aussi nécessaire d'étendre la classe **CourseFactory** afin de passer la structure au constructeur de cette nouvelle classe et de mettre à jour les instances en cas de modification de la structure. Ces changements sont présentés au diagramme de classe de la figure 6.5. Il utilise les mêmes conventions que celui de la figure 3.4 et ne présente pas en détail les classes déjà présentées dans celui-ci. Cependant, ce design nécessite de changer les classes utilisées par la plateforme et ceci n'est pas possible avec un plug-in.

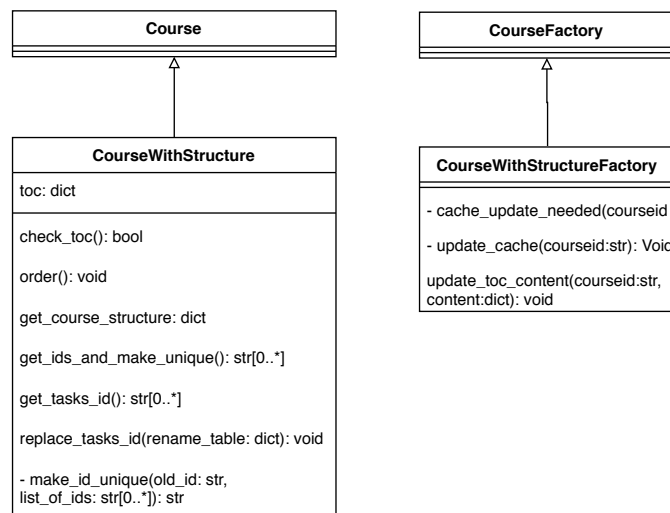


FIGURE 6.5 – Diagramme de classe des cours avec structure

Une autre possibilité est d'utiliser la réflexion [12]. Celle-ci permet d'ajouter

directement les méthodes aux classes **Course** et **CourseFactory**. Cependant, à cette échelle, il est préférable d'éviter d'utiliser la réflexion. En effet, cela rend le design de la plateforme plus confus avec la structure des différentes classes modifiées dans des fichiers différents. De plus, cette technique a un impact négatif sur les performances.

Finalement, il est possible d'utiliser une approche fonctionnelle. Avec cette approche, un module fournit des méthodes statiques pour accéder et modifier la structure d'un cours. Nous avons choisi cette approche, car elle nous permet de facilement réaliser tout ce que nous souhaitons et elle est compatible avec le système de plug-ins. Cependant, des commentaires et une **issue** Github¹ ont été ajoutés afin de pouvoir facilement passer à une approche orientée objet lors de l'intégration, car cela constitue une meilleure pratique pour une plateforme orientée objet.

Les fonctions mises à disposition par le module sont les suivantes :

- **get_toc** : récupère le contenu brut du fichier **toc.yaml** pour le cours passé en argument ou une liste vide si le fichier n'existe pas.
- **update_toc_content** : enregistre la structure passée en argument dans le fichier **toc.yaml** du cours.
- **check_toc** : vérifie que la structure passée en argument respecte le format défini à la section 6.1.2. Si ce n'est pas le cas, renvoie aussi un message précisant l'erreur rencontrée.
- **order** : ordonne la structure passée en argument en fonction des **rank**.
- **get_course_structure** : récupère la structure du cours ordonnée si celle-ci existe et est valide, autrement retourne une structure composée d'une section contenant tous les exercices du cours.
- **get_ids_and_make_unique** : récupère la liste des identifiants des sections et s'assure que ceux-ci sont uniques.
- **get_tasks_id** : récupère la liste des identifiants des exercices
- **replace_tasks_id** : modifie les identifiants des exercices en fonction de la table de renommage passée en argument.
- **make_id_unique** : une fonction d'aide permettant de rendre un identifiant unique par rapport à une liste d'identifiants.

1. <https://github.com/bensim602/INGInious/issues/1>

6.1.3 Contribution

Nouvelles pages	Vue en sections du cours Édition de la structure
Classes impactées	Aucune (approche fonctionnelle)
Statut d'intégration	Directement intégrable comme plug-in - Commentaires et issue pour intégration directe à inginius
Nombres de lignes de code	382++
Nombres de lignes de HTML	545++
Nombres de lignes de JavaScript	328++

6.2 Importer et exporter des exercices

Ce plug-in permet d'importer et d'exporter des exercices et des sections par le biais d'archives. Il permet de facilement réutiliser ces éléments d'un cours à un autre.

6.2.1 Fonctionnalités

Exporter des exercices

L'objectif de cette extension est de permettre aux enseignants de partager les exercices qu'ils ont créés. La première étape pour atteindre cet objectif est de permettre aux enseignants d'exporter ces exercices dans un fichier. Pour ce faire, nous avons créé une page d'exportation d'exercices dans l'administration du cours. Cette page est présentée à la figure 6.6. Dans l'onglet **Exercices**, il est possible de sélectionner les différents exercices que nous souhaitons exporter. Une fois la sélection terminée, l'utilisateur clique sur le bouton **Exporter les éléments sélectionnés** et une archive est téléchargée contenant les exercices sélectionnés.

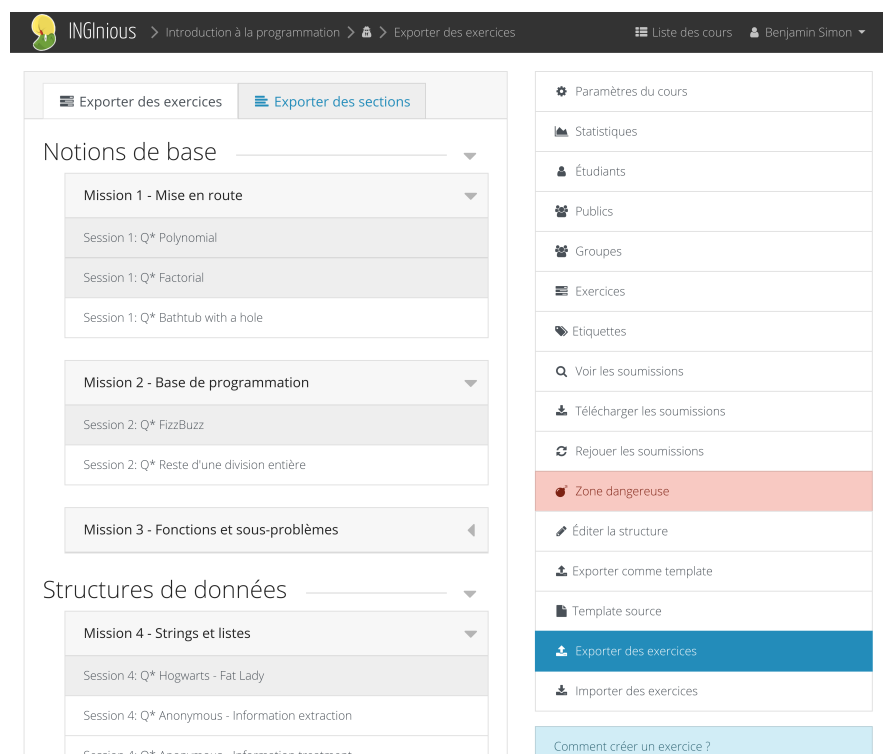


FIGURE 6.6 – Page d'exportation d'exercices

Exporter des sections

Avec les exercices nous souhaitons aussi pouvoir exporter des sections. Pour ce faire, il suffit d'utiliser l'onglet **Sections** présenté à la figure 6.7. La structure des sections sélectionnées est ajoutée à l'archive et sera ajoutée aux cours lors de l'importation. Par défaut, lorsqu'une section est sélectionnée, tous ses exercices sont aussi ajoutés à l'archive. Cependant, il est possible d'en dé-sélectionner certains dans l'onglet exercices. Ceux-ci ne sont alors pas ajoutés dans l'archive et sont retirés de la structure des sections.

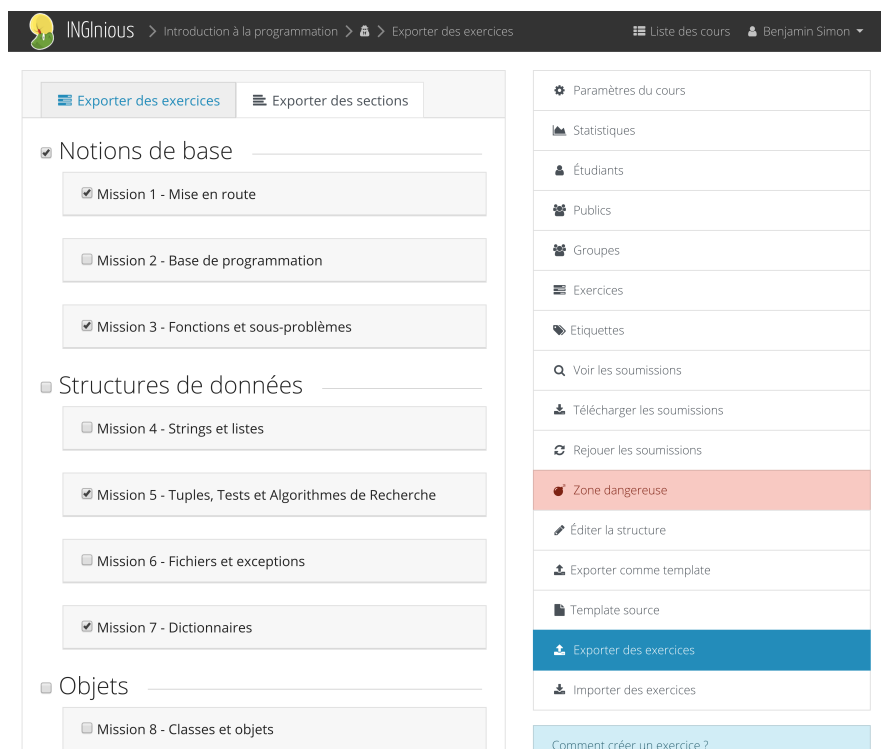


FIGURE 6.7 – Page d'exportation de sections

Importer des exercices et sections

Après avoir exporté des exercices et des sections dans une archive, il est possible de les importer dans un cours. Pour ce faire, il suffit de consulter la page d'importation d'exercices de l'administration du cours, sélectionner l'archive et appuyer sur importer. Cette page est présentée à la figure 6.8. Tous les exercices de l'archive sont alors ajoutés aux cours et la structure des chapitres de l'archive est ajoutée à la fin de la structure du cours, si cela est supporté par la plateforme. Nous avons aussi envisagé la possibilité de permettre aux utilisateurs de sélectionner les

exercices et sections de l'archive qu'ils souhaitent importer, de manière similaire à la sélection des éléments à exporter. Cependant, nous avons manqué de temps pour implémenter cette fonctionnalité.

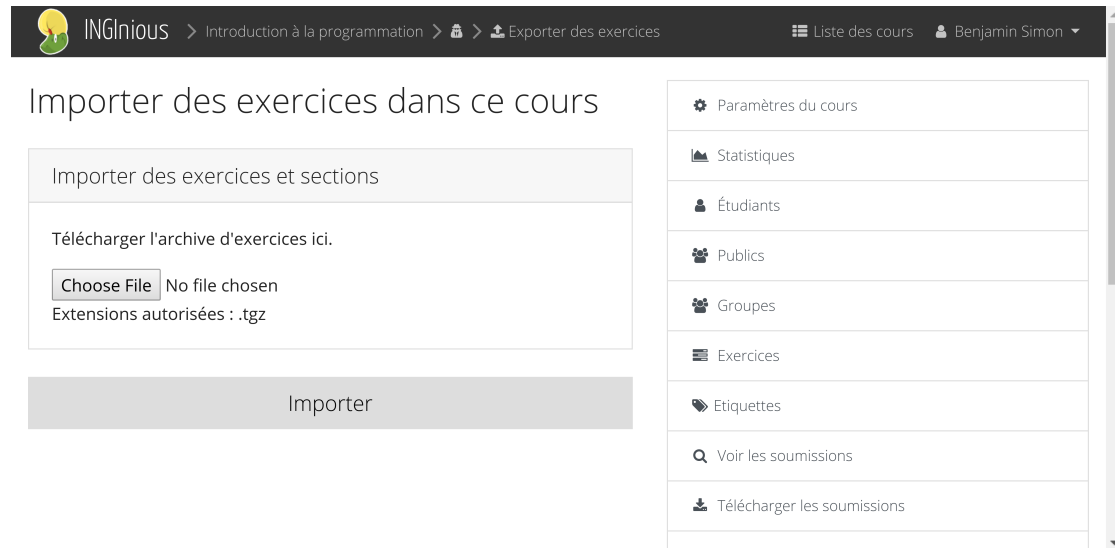


FIGURE 6.8 – Page d’importation d’exercices et de sections

6.2.2 Implémentation

Format de l'archive

Afin de pouvoir être importée, l'archive doit respecter un format strict. Cette archive doit être au format **tar.gz**. Chaque dossier de l'archive correspond à un exercice. Le nom de ce dossier correspond à l'identifiant de l'exercice et il contient les fichiers relatifs à cet exercice (voir section 3.2.3). Si des sections sont importées, leur structure doit respecter le format décrit à la section 6.1.2. Cette structure est stockée dans un fichier nommé **sections.yaml**. Une fois décompressée, l'archive prend donc le format décrit à la figure 6.9.

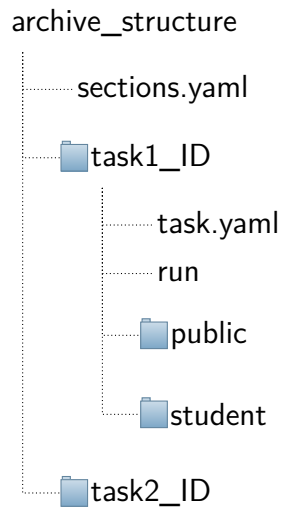


FIGURE 6.9 – Contenu d’une archive pour l’importation d’exercices

Ajout de méthode d’import

Ce plug-in permet aussi d’ajouter de nouvelles méthodes pour importer des exercices. Pour ce faire, il suffit d’ajouter un hook comme présenté ci-après.

```
plugin_manager.add_hook("import_tasks", import_tasks)
```

Où **import_tasks** est une fonction avec les spécifications suivantes

```
import_tasks(template_helper, plugin_manager, user_manager, course)
```

- **template_helper** : l’objet permettant de produire un rendu pour les fichiers HTML.
- **plugin_manager** : l’objet permettant de gérer les plug-ins.
- **user_manager** : l’objet permettant de gérer les utilisateurs.
- **course** : le cours dans lequel les cours sont importés.
- **retourne** : un tuple contenant le titre du type d’import et le rendu à afficher à l’utilisateur

C’est cette méthode qui est utilisée pour ajouter l’importation d’exercices provenant de templates décrits à la section 6.3.1.

Gestion des requêtes HTTP POST

La création de l’archive se fait par le biais de la méthode **POST** de la page d’exportation d’exercices. Elle place dans l’archive les dossiers correspondant à chaque exercice et un nouveau fichier contenant la structure des sections et leurs exercices sélectionnés. Par contre, les exercices non sélectionnés et leur place dans

la structure ne sont pas repris dans l'archive. Cette archive est ensuite envoyée pour être téléchargée par l'utilisateur.

L'importation d'exercices dans un cours se fait par le biais de la méthode **POST** de la page d'importation d'exercice. Chacun des dossiers de l'archive correspondant à un exercice est copié dans le système de fichier du cours. Si un exercice du cours porte déjà le même identifiant, l'exercice importé est renommé afin de garder des identifiants uniques. La structure présente dans l'archive est ajoutée à la suite de la structure du cours. À nouveau, les identifiants de la structure importée sont renommés en cas de conflit avec ceux de la structure du cours.

6.2.3 Contribution

Nouvelles pages	Importation d'exercices Exportation d'exercices
Classes impactées	Aucune (tout est implémenté dans les requêtes POST)
Statut d'intégration	Directement intégrable comme plug-in
Nombres de lignes de code	295++
Nombres de lignes d'HTML	325++

6.3 Template

Ce plug-in permet de créer des cours à partir d'une base grâce à la notion de templates. Un template est très similaire à un cours à l'exception qu'il ne peut pas contenir d'étudiant, mais permet d'instancier des cours. De plus il contient des descriptions visant à aider les enseignants. Lorsqu'un cours est créé à partir d'un template, un nouveau cours est créé et les exercices et la structure du template lui sont ajoutés. Le processus inverse permet aussi d'exporter un cours comme template. Les templates peuvent être distants ou locaux, mais sont toujours stockés de la même façon. Les templates distants sont listés dans le fichier **list.yaml** et sont synchronisés avec leurs répertoires distants lors du démarrage de la plateforme et lorsque l'administrateur appuie sur le bouton **Pull templates**. Cependant ces répertoires distants doivent être créés manuellement et ce plug-in permet uniquement de créer et modifier des templates locaux. Le fonctionnement général des templates est décrit à la figure 6.10.

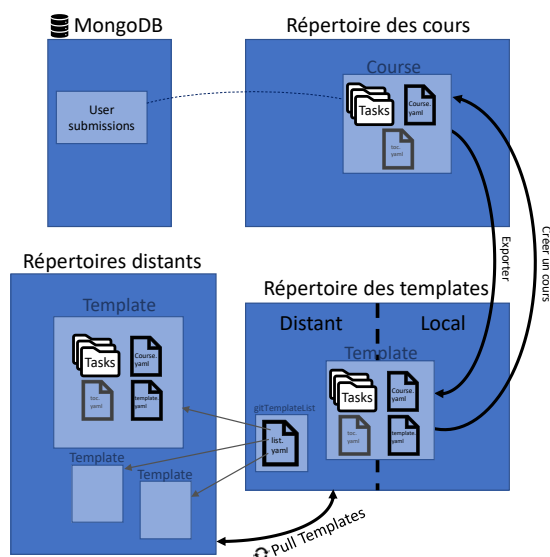


FIGURE 6.10 – Fonctionnement général des templates

6.3.1 Fonctionnalités

Importer un template

Le rôle principal de ce plug-in est de permettre la création de cours en s'appuyant sur une base existante appelée template. Pour ce faire, il suffit à l'utilisateur de choisir un template et de définir l'identifiant du cours qu'il souhaite créer.

Si l'identifiant n'existe pas encore et ne contient pas de caractères spéciaux, un nouveau cours est créé. Son nom correspond au nom du template et à l'identifiant du cours séparé par un tiret. Il contient tous les exercices et la structure du template ainsi que ses tags et sa description. De plus, l'utilisateur est ajouté à la liste des administrateurs et le cours est défini comme non accessible aux étudiants.

Une fois le cours créé, l'utilisateur est redirigé vers la page d'administration. Il peut alors modifier ce nouveau cours comme il le souhaite.

Catalogue

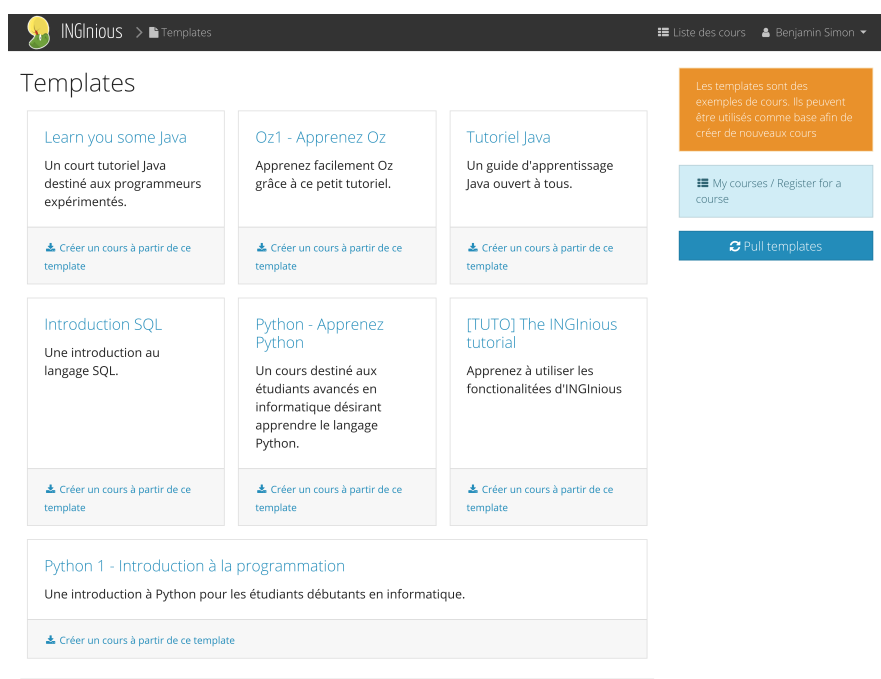


FIGURE 6.11 – Catalogue de templates

Un catalogue présente les templates de la plateforme accessible à l'utilisateur. Un template est accessible à l'utilisateur s'il est public ou s'il est privé, mais que l'utilisateur est administrateur de celui-ci. Un titre et une courte description sont disponibles pour chaque template. Il est soit possible d'importer un template à partir de cette page ou de le consulter plus en détail. Un exemple de catalogue est présenté à la figure 6.11.

Visualisation d'un template

L'interface utilisateur de la figure 6.12 présente un template. Sur cette page sont affichés le titre ainsi que la description complète du template. Cette description a

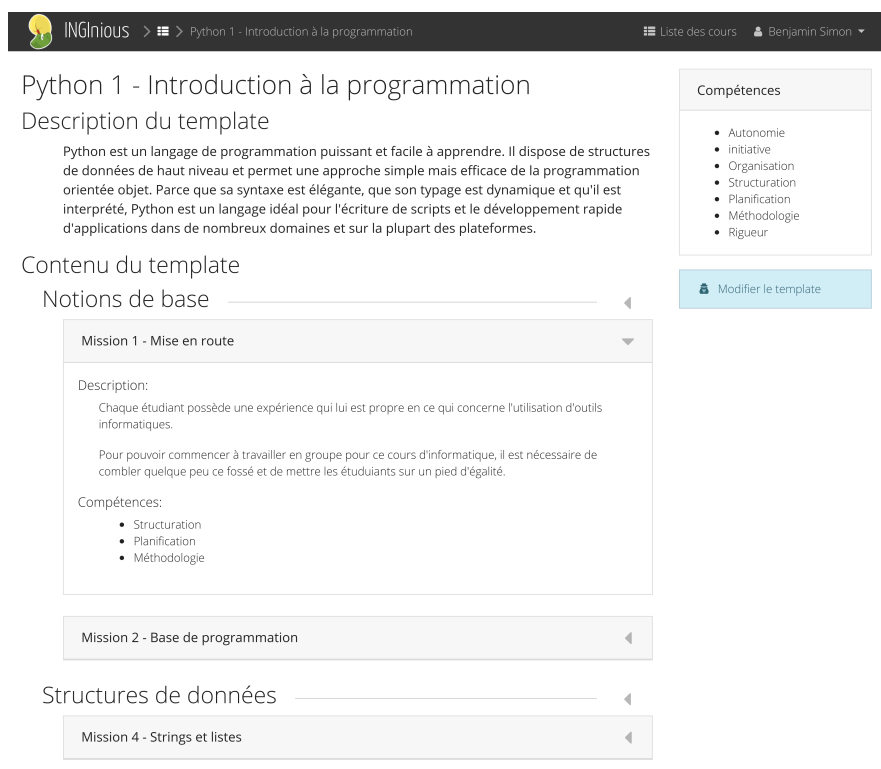


FIGURE 6.12 – Page d’un template

plusieurs objectifs. Tout d’abord, elle vise à expliquer à l’enseignant le contenu du cours. Elle a aussi pour but de guider l’enseignant et de l’aider à donner son cours. Par exemple, il serait possible de donner des idées d’activités à effectuer en classe ou des points importants à aborder durant le cours. Cette description est écrite dans le markup langage **restructured text**², il est donc possible d’y intégrer des images, des liens hypertextes... À droite se trouve la liste des compétences qui seront enseignées par un cours créé à partir de ce template. En dessous de la description apparaît la structure du template si celle-ci existe. Il est possible de cliquer sur une section afin de consulter la description et les compétences de celle-ci. Ces éléments sont similaires à ceux du template, mais pour une section spécifique.

Accès à des templates distants

Il est important de remarquer qu’inginius n’est pour le moment pas prévu pour fonctionner avec une seule instance disponible pour tous. En effet, une instance globale existe³ mais celle-ci n’est pas prévue pour supporter plusieurs établissements.

2. <https://docutils.sourceforge.io/rst.html>

3. <https://inginius.org>

Pour le moment, nous allons donc supposer l'utilisation d'une seule plateforme par établissement. Ceci signifie que chaque école ou université doit posséder et héberger sa propre instance. Les templates n'étant pas partagés d'une instance à l'autre, ils ne sont pas non plus partagés entre les différentes écoles et universités.

Pour remédier à ce problème, nous avons voulu rendre certains templates accessibles depuis toutes les instances. Pour ce faire, nous utilisons Git⁴. Une liste définit une série de templates stockés sur des répertoires distants.

Tous ces templates sont ajoutés aux instances de la plateforme et sont disponibles dans le catalogue. Ceux-ci sont ajoutés par l'administrateur lorsque celui-ci met à jour ces templates grâce au bouton présent sur le catalogue des templates ou lors du démarrage de la plateforme. Il n'est cependant pas possible de modifier ces templates à partir de la plateforme.

Création de templates

À nouveau, pour que tout ceci soit utile, il est nécessaire que des utilisateurs puissent créer des templates. Pour ce faire, il existe deux méthodes. Tout d'abord, il est possible de créer un template vide à partir du catalogue des templates. Ceci se fait de manière très similaire à la création d'un cours.

Une autre possibilité est d'utiliser un cours existant et de le transformer en template. Pour cela, il suffit d'aller dans l'administration du cours en question et de sélectionner l'onglet exporté comme template. Le template ainsi créé contient les exercices, la structure et les tags du cours. C'est ensuite à l'utilisateur d'enrichir le template en complétant les descriptions manquantes pour chaque section et pour le template via l'édition du template.

Éditer un template

Des outils sont fournis à l'utilisateur afin de créer et modifier un template. Ils sont disponibles à la page de modification d'un template, accessible depuis la page principale du template. Ceux-ci correspondent et héritent des outils de la page d'administration d'un cours sans ceux relatifs à la gestion des étudiants. Il est possible de créer et modifier des exercices, créer et modifier des tags et éditer la structure. Il est aussi possible de modifier les paramètres du template, c'est-à-dire son titre, les personnes ayant le droit de le modifier, la description à l'attention des étudiants et la visibilité du template. Un template privé ne peut être utilisé que par ses créateurs tandis qu'un template public peut être utilisé par n'importe qui.

4. <https://git-scm.com>

Éditer les descriptions d'un template

INGinious > Python 1 - Introduction à la program... > 1 Description du template

Liste des cours Benjamin Simon

Description du template

Courte description
(max. 250 caractères)

Une introduction à Python pour les étudiants débutants en informatique.

Description du template

- 1 Python est un langage de programmation puissant et facile à apprendre. Il dispose de structures de données de haut niveau et permet une approche simple mais efficace de la programmation orientée objet. Parce que sa syntaxe est élégante, que son typage est dynamique et qu'il est interprété, Python est un langage idéal pour l'écriture de scripts et le développement rapide d'applications dans de nombreux domaines et sur la plupart des plateformes.

Compétences du template

Autonomie, initiative, Organisation, Structuration, Planification, Méthodologie, Rigueur

Structure description

Notions de base

Mission 1 - Mise en route

Description de la section

- 1 Chaque étudiant possède une expérience qui lui est propre en ce qui concerne l'utilisation d'outils informatiques.
- 2
- 3 Pour pouvoir commencer à travailler en groupe pour ce cours d'informatique, il est nécessaire de combler quelque peu ce fossé et de mettre les étudiants sur un pied d'égalité.

Compétences de la section

Structuration, Planification, Méthodologie

Mission 2 - Base de programmation

Template settings

Description du template

Exercices

Etiquettes

Zone dangereuse

Structure editor

Comment créer un exercice ?

Documentation

Save changes

FIGURE 6.13 – Édition des descriptions

Une partie importante des templates est qu'ils fournissent des descriptions et un guide pour aider les enseignants (voir section 6.3.1). Il est donc important de pouvoir écrire et modifier ces descriptions. Ceci est possible dans l'onglet description du template de la page de modification du template. Cette page présentée à la figure 6.13 est très similaire à celle de la page principale d'un template à la différence qu'il est possible d'éditer les différents éléments.

Privilèges des utilisateurs

Laisser ces fonctionnalités accessibles à n'importe quels utilisateurs n'est malheureusement pas possible. En effet, elles permettent de créer de nouveaux cours dont l'utilisateur est administrateur. Or, nous ne souhaitons pas que cela soit possible. En effet, il serait alors possible aux étudiants de créer de nombreux cours qui encombreraient la plateforme et prendraient de la place sur le serveur.

Malheureusement, il n'existe pas encore de niveau de privilège intermédiaire entre un simple utilisateur et un administrateur. Ces fonctionnalités sont donc actuellement réservées aux administrateurs. Cependant, les développeurs d'inginius ont prévu d'ajouter un niveau de privilège enseignant. Lorsque ce niveau existera, il sera alors possible de rendre ces fonctionnalités disponibles à tous les enseignants.

Accès au template

Il est parfois utile de consulter les descriptions et guides qui accompagnent le template ayant servi de base à un cours après la création de ce dernier. Afin de rendre ceci possible, un lien vers le template source est disponible dans l'administration des cours dérivés d'un template.

Importer des exercices

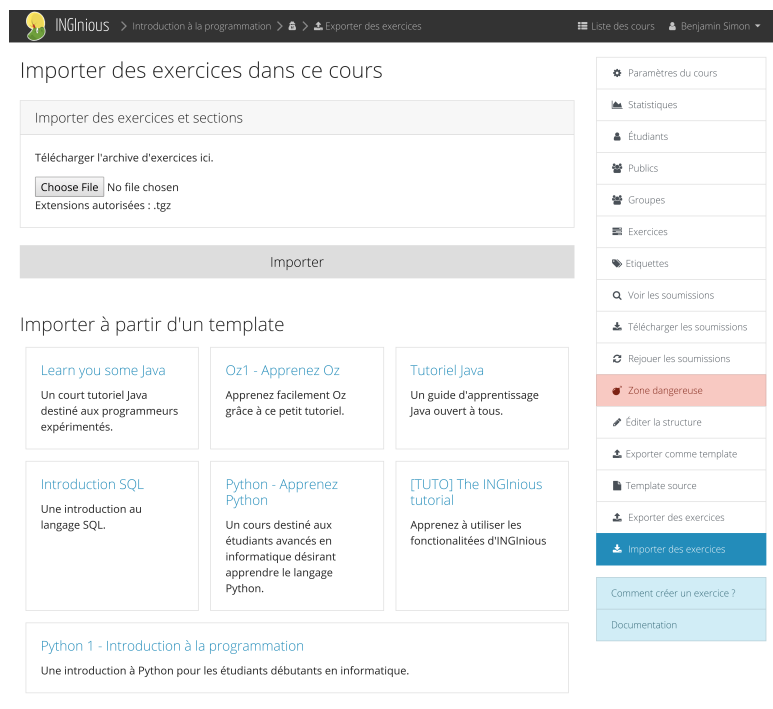


FIGURE 6.14 – Importation à partir de templates

Le plug-in permet aussi d'importer des exercices ou sections provenant de templates dans un cours existant. Pour ce faire, il suffit de sélectionner le template sur la page d'importation d'exercices. Cette page est présentée à la figure 6.14. Ensuite, la sélection des exercices et sections se fait de manière similaire à l'exportation

d'exercices et de sections décrite à la section 6.2.1. Une fois les exercices et sections sélectionnés, ceux-ci sont copiés dans le dossier du cours comme le décrit la figure 6.15.

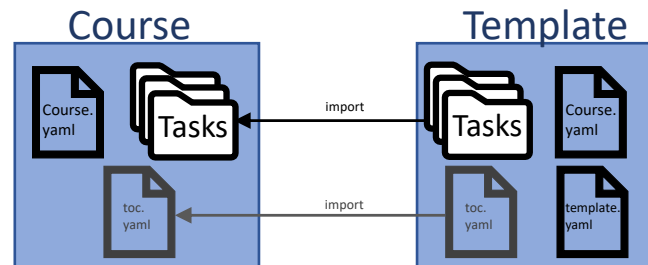


FIGURE 6.15 – Fonctionnement de l'importation

6.3.2 Implémentation

Stockage des données

Les templates étant très similaires à des cours, ces derniers sont stockés de la même façon. Étant donné qu'ils ne contiennent pas de données sur les utilisateurs, tout peut être sauvegardé dans le système de fichiers. Dans ce but, nous utilisons un dossier spécifique dont l'adresse est définie dans le fichier de configuration. Ce dossier respecte le même format que celui présenté à la figure 3.5. La seule différence est la présence d'un fichier **template.yaml** à la base du dossier de chaque template. Ce fichier contient les différentes compétences et descriptions du template. La structure de ce fichier est décrite à la figure 6.16.

Il s'agit donc d'un dictionnaire contenant la longue et courte description du template ainsi que la liste de ses compétences. Il contient aussi un dictionnaire pour les descriptions des sections et un pour la liste de leurs compétences. Ces dictionnaires ont comme clés les identifiants des sections et comme valeurs les descriptions ou listes des compétences correspondantes.

Gestions des templates distants

Comme expliqué dans la section 6.3.1, il est possible d'utiliser sur la plateforme des templates distants. Cela se fait par le biais d'un système à deux niveaux. Le premier niveau correspond à un premier répertoire distant contenant la liste des templates distants. Il peut s'agir d'un répertoire par défaut ou d'un répertoire fourni dans le fichier de configuration. Ce répertoire contient uniquement un fichier **list.yaml**. Dans ce fichier est stocké un dictionnaire ayant pour clés les identifiants

```

short__description: Courte description
description: Description complete
skills:
- competence1
-..
structure__description:
  section__1: description section 1
  ..
structure__skills:
  section__1: description section 1
- competence1
-..
..

```

FIGURE 6.16 – Pseudo structure du fichier template.yaml

des templates et comme valeurs les adresses de leurs répertoires. Le deuxième niveau correspond aux templates de la liste, présents sur des répertoires distants.

Au démarrage de la plateforme, si cela n'est pas déjà fait, le répertoire du premier niveau est cloné dans un dossier nommé **gitTemplateList**. Lors du démarrage et à chaque fois que l'administrateur réclame de synchroniser, un pull est effectué sur le répertoire du premier niveau. Ensuite, pour chaque template de la liste, si cela n'est pas déjà fait, le répertoire est cloné dans un dossier portant l'identifiant du template suivi de "_g1t". Cette extension permet de différencier les templates distants et éviter les conflits d'identifiant. Chaque répertoire de la liste est ensuite pull.

Le dossier contenant les templates prend donc le format de la figure 6.17

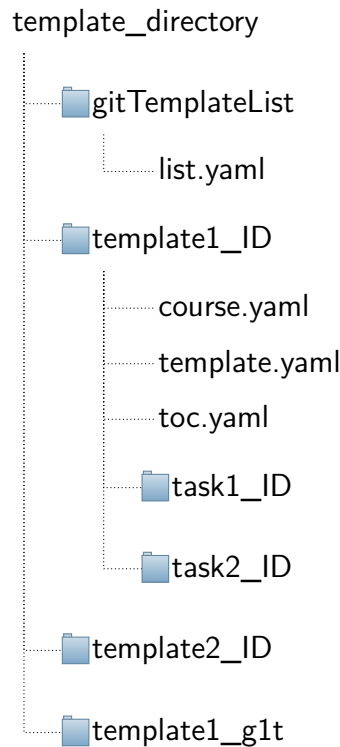


FIGURE 6.17 – Structure de fichiers des templates

Implémentation des classes

En nous calquant sur l'implémentation des cours, nous avons décidé de définir trois classes principales permettant de gérer les templates. Il s'agit de **Template** pour la gestion des données des templates, **WebAppTemplate** pour la présentation de ces données à l'utilisateur et **TemplateFactory** pour la création de classes **Template** à partir du système de fichier. La figure 6.18 présente le diagramme de classe des templates. Il utilise les mêmes conventions que celui de la figure 3.4 et ne présente que les éléments modifiés par rapport à ce dernier.

La classe **Template** hérite de la classe **Course** et y ajoute une variable d'instance **template_content** contenant les descriptions du template. Elle contient aussi une méthode statique **export** permettant de créer un cours à partir d'un template et inversement.

La classe **WebAppTemplate** hérite de **Template** et ajoute plusieurs variables permettant de présenter à l'utilisateur le template et de gérer ce dernier. Afin de réutiliser les mêmes éléments pour les pages d'administration des templates et cours, plusieurs variables ont aussi été ajoutées à **WebAppCourse** et **WebAppTemplate**. En effet, les pages de cours et de templates ne se trouvent pas aux mêmes

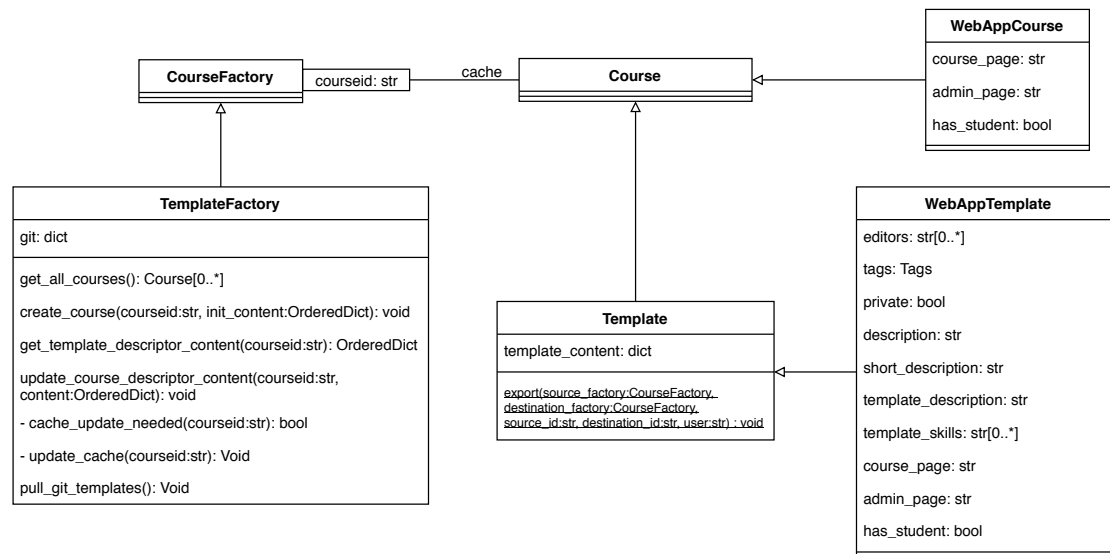


FIGURE 6.18 – Diagramme de classes des templates

adresses, il est nécessaire de connaître ces adresses afin que les liens fonctionnent correctement. Il est aussi nécessaire de savoir si le type de cours présentés supporte la gestion des étudiants ou non afin de savoir si ces options doivent être affichées.

La classe **TemplateFactory** hérite de la classe **CourseFactory** et permet de gérer les templates distants et ajoute une méthode **pull_git_templates** permettant de mettre à jour ces templates. Elles réimplémentent aussi beaucoup de méthodes de la classe mère. Cela est dû principalement au fait que la classe **Template** utilise un fichier en plus. Or plusieurs méthodes utilisent l'hypothèse que tous les types de cours utilisent les mêmes fichiers. En généralisant la classe **CourseFactory** avec une liste des fichiers utilisés par le cours, il est possible de simplifier le diagramme par celui de la figure 6.19. Malheureusement, ces modifications ne peuvent pas être implémentées avec un plug-in, mais des commentaires ainsi qu'une **issue** Github⁵ ont été ajoutés afin de pouvoir facilement utiliser cette approche lors de l'intégration.

5. <https://github.com/bensim602/INGInious/issues/2>

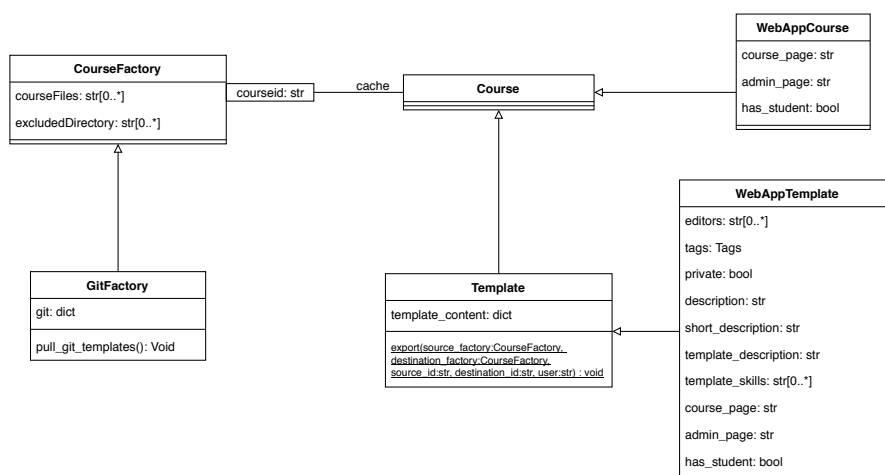


FIGURE 6.19 – Diagramme de classes des templates modifié

6.3.3 Contribution

Nouvelles pages	Catalogue des templates Présentation d'un template Administration d'un template Édition des descriptions d'un template
Classes impactées	WebAppCourse Template TemplateFactory WebAppTemplate
Statut d'intégration	Directement intégrable comme plug-in - Commentaires et issue pour intégration directe à inginius
Nombres de lignes de code	1113++ 25–
Nombres de lignes d'HTML	739++ 16–

Les nombres de lignes peuvent sembler élevés, mais beaucoup de pages ont été ajoutées et, à défaut de pouvoir généraliser, du code a été dupliqué.

Chapitre 7

Validation

Cette partie du mémoire a pour objectif d'évaluer la qualité ainsi que le bon fonctionnement de cette application dans des situations réelles. Pour ce faire, nous l'avons mise dans les mains d'utilisateurs afin de vérifier si ceux-ci parvenaient à l'utiliser facilement. La démarche entreprise s'inspire des recherches présentées sur **Usability.gov** [9].

7.1 Méthodologie

Dans un premier temps, nous avons développé une série de scénarios. Chaque scénario contient une mise en contexte et une liste d'étapes permettant d'effectuer une action (par exemple créer un template). Ces actions correspondent à des situations réelles pouvant être rencontrées par des enseignants du secondaire. Nous avons essayé de développer ces scénarios de telle sorte qu'ils utilisent toutes les fonctionnalités ayant été implémentées durant ce mémoire.

Nous avons ensuite fait passer des entretiens à différents utilisateurs. Afin d'obtenir les résultats les plus pertinents possible, nous avons choisi des utilisateurs étant enseignants en informatique dans le secondaire et le supérieur, étudiants et tuteurs en informatiques ainsi que des utilisateurs étrangers au domaine de l'informatique. Au long de cette validation, nous avons pu faire passer des entretiens à 21 utilisateurs, dont 10 étaient familiers avec la plateforme.

Les entretiens pour la validation se déroulent en plusieurs étapes. La première étape consiste à mettre l'utilisateur dans la situation d'un enseignant en informatique dans le niveau secondaire. On lui demande ensuite de réaliser les scénarios développés précédemment tout en partageant ses impressions. La mise en situation et les scénarios présentés à l'utilisateur sont décrits à l'annexe C. Ces scénarios peuvent toujours être réalisés avec les adresses et identifiants fournis dans les scénarios. Ensuite, nous demandons à l'utilisateur de remplir un questionnaire

présenté à l'annexe D.

Étant donné les conditions particulières liées au COVID-19 et les mesures de confinement qui en ont découlé, nous avons décidé de mener ces entretiens par le biais d'appels audio avec partage d'écran sur des plateformes telle que Teams, Discord ou Skype. Ce format prend plus de temps, car il est nécessaire d'être présent durant toute la durée de la validation de chaque utilisateur. Cependant, il permet d'observer plus en détail les difficultés rencontrées par les utilisateurs et d'obtenir leurs impressions sur la plateforme de manière plus interactive.

7.2 Résultats

Les résultats obtenus lors de cette validation sont assez encourageants. Les réponses récoltées dans le questionnaire à choix multiple sont présentées sur la figure 7.1. Nous avons décidé de séparer les résultats provenant d'utilisateurs familiers avec la plateforme (I) de ceux ne l'étant pas. Cela permet d'identifier quelles difficultés sont liées à la familiarisation avec la plateforme (NI).

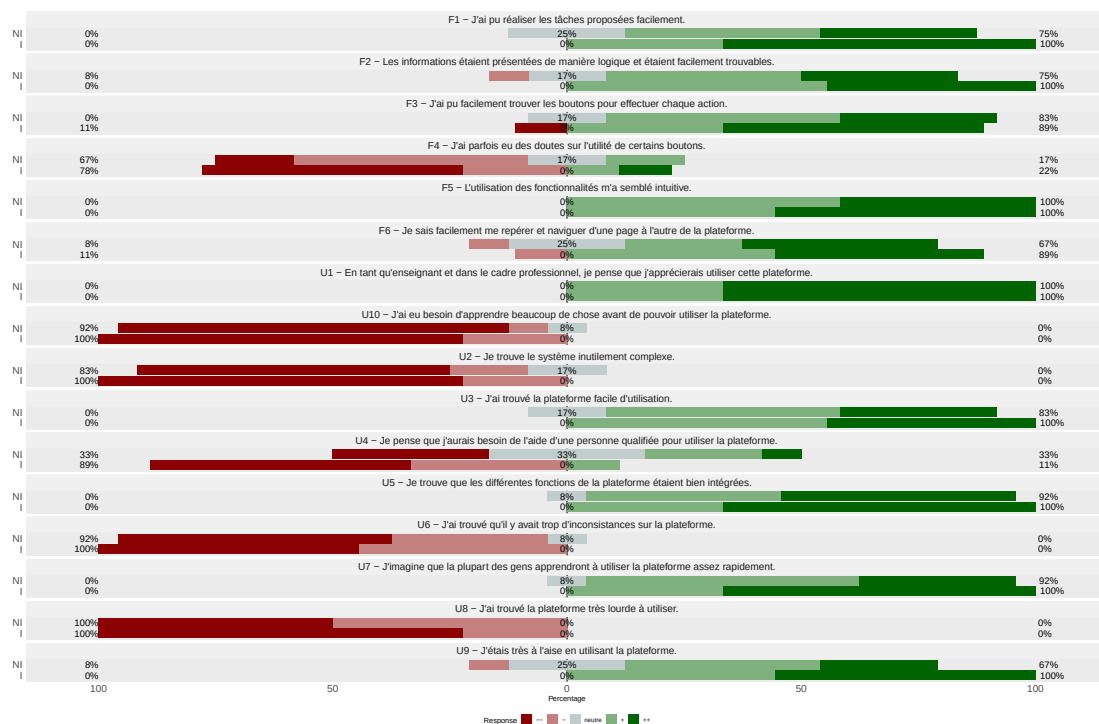


FIGURE 7.1 – Répartition des réponses au questionnaire de validation

En analysant ce graphe, on remarque tout d'abord que plusieurs utilisateurs, principalement ceux n'étant pas familiers avec la plateforme, estiment avoir be-

soin d'assistance pour utiliser la plateforme. La section 7.3 présente les solutions proposées pour ce problème. On remarque aussi que plusieurs utilisateurs ont des difficultés à se repérer, mais cela est plutôt lié au fonctionnement de la plateforme et non aux plug-ins. Finalement, plusieurs utilisateurs ont eu des doutes sur l'utilité de certains boutons. Cependant, cette question est la seule de la première série posée sous la forme négative et après avoir interrogé certains utilisateurs sur cette réponse, la plupart ont reconnu ne pas avoir vu la forme négative. Pour le reste, les réponses étaient généralement très positives, surtout chez les utilisateurs familiers avec la plateforme.

Les 10 dernières questions de ce questionnaire correspondent au test de **System Usability Scale (SUS)** [5]. Elles permettent de donner à l'application un score entre 0 et 100. Ce score ne correspond pas à un pourcentage, mais il permet de comparer à d'autres scores. Des études empiriques montrent que la moyenne pour ce test est de 68 et qu'un score supérieur à 80 correspond à un excellent score [3]. Lors de nos tests, nous obtenons un score de 85,7. Il s'agit donc d'un bon score qui augmente lorsque les utilisateurs sont familiers avec la plateforme (le score est de 90.6 pour les utilisateurs utilisant fréquemment la plateforme contre 82.1 pour les autres).

Finalement, cette validation a aussi permis d'obtenir des remarques sur des éléments précis pouvant être améliorés ou d'identifier certains bogues de la plateforme. Pour la plupart, ceux-ci ont pu être corrigés et sont présentés à la section suivante.

7.3 Améliorations

7.3.1 Erreurs

La validation a permis de trouver certaines erreurs des plug-ins. Tout d'abord, les exercices ajoutés à la structure ne pouvaient pas être glissés et déposés. De plus, la barre de recherche des exercices disparaissait lorsque l'utilisateur faisait défiler les exercices. Ces deux problèmes ont donc été résolus.

7.3.2 Indications

Durant la validation, il a parfois été nécessaire de donner certaines indications aux utilisateurs. Nous avons donc décidé d'intégrer ces indications à la plateforme afin que les utilisateurs puissent effectuer toutes les opérations sans aide extérieure. Par exemple, plusieurs utilisateurs ne se rendaient pas compte que les exercices n'étant pas présents ou ayant été retirés de la structure pouvaient être ajoutés via le menu **Ajouter des exercices**. Nous avons donc ajouté un avertissement à la fin de la structure expliquant ce concept. Cet avertissement est présenté à la figure

7.2. De plus, quelques utilisateurs quittaient la page afin de trouver les exercices sans sauvegarder. Pour remédier à ce problème, nous avons ajouté un avertissement avant que l'utilisateur quitte la page sans sauvegarder.

Les exercices du cours n'étant pas présents ou ayant été retirés de la structure peuvent être ajoutés à une section. Pour ce faire, utiliser l'option Ajouter des exercices du menu de la section.

FIGURE 7.2 – Page d'édition de la structure avec avertissements

7.3.3 Glisser-déposer

Selon les navigateurs, certains utilisateurs avaient des difficultés à faire défiler la page tout en glissant des éléments. Cela étant dû au glisser-déposer du navigateur, il n'était pas possible pour nous de résoudre directement ce problème. Cependant, pour limiter ce type d'action nous avons réduit toutes les sections terminales lorsqu'un élément est glissé. Lorsqu'un exercice est maintenu au-dessus d'une autre section, celle-ci s'ouvre afin que cet exercice puisse y être glissé. Ce nouveau système est présenté à la figure 7.3.

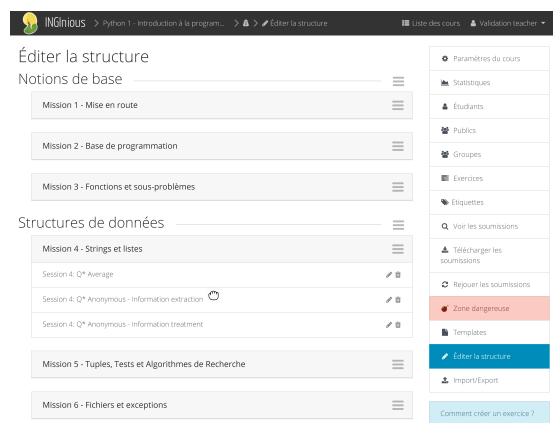


FIGURE 7.3 – Nouveau système de glisser-déposer

7.3.4 Fusionner les onglets

Certains utilisateurs ont éprouvé des difficultés à se repérer dans l'administration du cours, car celle-ci contenait beaucoup d'onglets. Il n'est pas possible de réduire le nombre d'onglets initial, cependant une **issue**¹ Github a été ajoutée afin de

1. <https://github.com/UCL-INGI/INGInious/issues/541>

reporter le problème. De plus, il a aussi été possible de fusionner les onglets liés aux plug-ins et de n'en garder qu'un par plug-in. Ce changement est illustré à la figure 7.4.

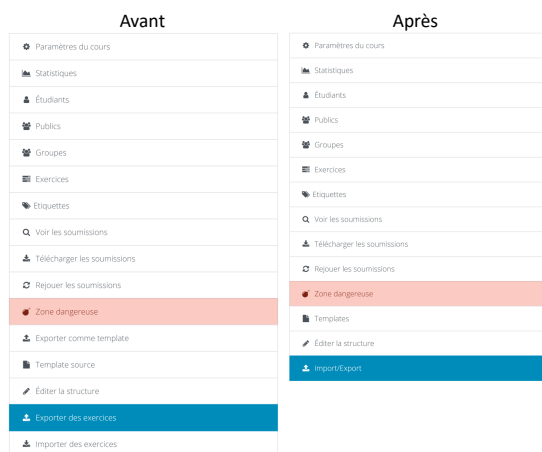


FIGURE 7.4 – Fusion des onglets d'administration

7.3.5 Simplifier l'utilisation

Finalement, nous avons aussi apporté plusieurs modifications afin de faciliter l'utilisation de la plateforme. Par exemple, nous avons changé la couleur des éléments sélectionnés afin que ceux-ci soient plus visibles. Ce changement est présenté à la figure 7.5. Nous avons aussi fait en sorte de pouvoir réduire les sections en cliquant n'importe où sur l'en-tête.

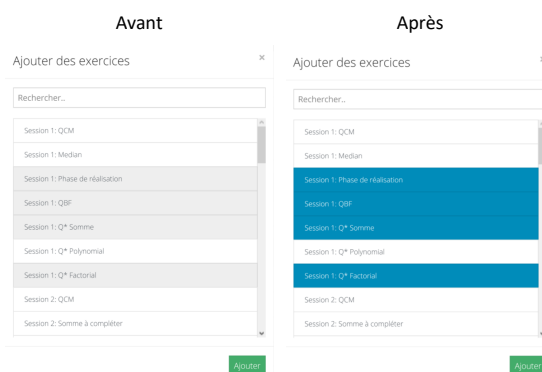


FIGURE 7.5 – Changement de la couleur de sélection

Chapitre 8

Améliorations futures

8.1 Intégration à INGINious

Au terme de ce travail, nous avons fourni et validé trois plug-ins pour la plateforme inginius. Chacun de ces plug-ins est disponible sur le répertoire Github associé à ce mémoire¹. Cependant, tout au long de ce mémoire, nous avons porté une attention particulière dans le but de faciliter l'intégration future à inginius (voir section 4.1). Nous espérons donc, dans le futur, avoir l'occasion de travailler sur l'intégration de ces plug-ins à la plateforme.

8.2 Privilèges enseignants

Comme expliqué à la section 6.3.1, le plug-in template est limité par le manque d'un niveau de privilège intermédiaire entre un simple utilisateur et un enseignant. Cependant, il est prévu qu'un tel niveau soit créé dans le futur. Une fois que cela sera fait, il sera alors possible d'adapter le plug-in conformément aux commentaires et à l'**issue** Github correspondante², afin de profiter pleinement du potentiel de l'extension.

8.3 Plateforme globale

Lors de l'analyse du problème (voir section 1.2), nous avons identifié une limitation importante liée aux ressources des établissements secondaires. En effet, nous nous sommes rendu compte qu'il n'était pas possible de demander à chaque établissement d'héberger sa propre plateforme. Cependant, dans le fonctionnement

1. <https://github.com/bensim602/INGInious>

2. <https://github.com/bensim602/INGInious/issues/3>

actuel d'inginius, chaque établissement possède sa propre instance. Il existe bien une instance globale **inious.org**³ mais celle-ci n'est pas prévue pour supporter les cours de plusieurs établissements.

Tout d'abord, car elle ne possède pas l'infrastructure physique nécessaire pour supporter le trafic et la quantité de données engendrée par plusieurs établissements.

Ensuite, la plateforme ne supporte pas la séparation en établissement et la création de cours par des utilisateurs n'étant pas administrateurs. Il faudrait donc implémenter un système semblable à celui présenté à la section 5.2.4.

Malheureusement, cela aurait représenté une charge de travail trop importante pour ce mémoire. De plus, fournir une telle plateforme ne fait pas partie pour le moment des objectifs de l'équipe de développement d'inginius. Pour l'instant, nous avons donc décidé de ne pas fournir de solution à ce problème. Cependant, il est tout à fait envisageable dans le futur qu'un travail implémente une telle solution.

8.4 Suivi des étudiants

La plateforme inginius propose un outil de suivi de la progression des étudiants. Cependant, cet outil ne permet pas une visualisation claire et schématique de la progression globale des étudiants. Une solution telle que celle proposée à la section 5.2.5 pourrait être envisagée pour la plateforme inginius également. Cependant, ceci n'a pas pu être réalisé durant ce mémoire, mais peut toujours faire l'objet d'un autre travail.

8.5 Création d'étudiants

Nous avons aussi envisagé la possibilité de créer une extension permettant aux enseignants de créer des comptes étudiants. En effet, dans le niveau secondaire, il est courant que les enseignants créent le compte des étudiants pour leurs cours. De plus, cette extension faciliterait l'accès à la plateforme en permettant aux étudiants n'ayant pas d'adresse mail d'obtenir un compte. Cependant, le temps étant limité, nous avons préféré mettre la priorité sur d'autres extensions.

3. <https://inginius.org>

Chapitre 9

Conclusion

Durant ce mémoire, nous avons eu l'occasion de discuter avec des enseignants et d'examiner plusieurs plateformes de cours en ligne. Cette analyse nous a permis de découvrir quelles étaient les forces et faiblesses des outils mis à la disposition des enseignants pour les accompagner dans l'enseignement de l'informatique. Nous nous sommes malheureusement rendu compte qu'aucune plateforme ne répondait pleinement aux attentes des enseignants.

Une telle plateforme devrait permettre aux enseignants de facilement créer et modifier des cours tout en permettant de réutiliser des bases de cours existantes. Elle devrait aussi proposer une décomposition des cours en chapitres. Afin d'aider les enseignants, il est important qu'elle propose des guides pédagogiques pour les cours ainsi que la possibilité de suivre facilement la progression des étudiants. En raison des contraintes spécifiques de l'enseignement secondaire, il faudrait aussi fournir une instance globale pour tous les établissements. Finalement, elle devrait offrir des types d'exercices variés tels que des questionnaires et des projets pouvant être évalués automatiquement pour faciliter la correction.

Nous avons donc envisagé une première approche pour fournir une telle plateforme, tout d'abord une nouvelle plateforme utilisant ingenious pour la gestion des exercices. Un travail d'analyse et de modélisation a été entrepris pour cette approche, mais il n'a malheureusement pas pu être implémenté. Cependant, le travail déjà fourni pourrait servir de base à un travail plus conséquent et certains éléments de l'analyse ont pu être réutilisés pour la suite.

Nous avons finalement changé d'approche et décidé d'étendre la plateforme ingenious par le biais de trois plug-ins. Afin de permettre aux enseignants de personnaliser leurs cours tout en se reposant sur une base existante, un premier plug-in introduit la notion de template. Un template est un cours modèle pouvant servir de base à plusieurs cours. Afin de bénéficier pleinement de ce plug-in, nous avons aussi ajouté un deuxième plug-in ajoutant la possibilité de structurer les cours en sections imbriquées. En effet, cette structure permet aux enseignants

de personnaliser les cours dérivés de templates en sélectionnant uniquement les chapitres qu'ils estiment intéressants pour leurs cours. Elle permet aussi aux étudiants de mieux visualiser l'organisation du cours et de plus facilement se repérer. Selon les développeurs de la plateforme, il s'agit d'ailleurs d'une fonctionnalité très demandée par les utilisateurs actuels. Finalement, un dernier plug-in permet de facilement réutiliser des exercices et sections d'un cours à l'autre grâce à une interface permettant l'importation et l'exportation d'exercices et sections dans une archive au format spécifique. Cette approche répond à la majorité des objectifs. Cependant, elle n'offre pas la possibilité de fournir une instance globale pour tous les étudiants. De plus, elle permet de suivre la progression des étudiants, mais manque d'un outil permettant de visualiser directement la progression globale de l'ensemble des étudiants.

Finalement, nous avons voulu mettre la plateforme dans les mains d'utilisateurs afin de nous assurer que celle-ci pouvait être facilement utilisée et répondait aux besoins des enseignants. Les résultats de cette validation ont été plutôt positifs. En effet, tous les participants ont réussi à réaliser toutes les tâches qui leur étaient demandées. De plus, elle a permis de mettre en lumière certaines erreurs et certains éléments de design méritant d'être améliorés. Ceux-ci ont alors pu être corrigés.

Bibliographie

- [1] The INGINious AUTHORS. *[TUTO] The INGINious tutorial*. URL : <https://inginous.org/course/tutorial> (visité le 14/09/2019).
- [2] The INGINious AUTHORS. *INGInious' documentation v0.6*. URL : <https://inginous.readthedocs.io/en/v0.6/> (visité le 09/09/2019).
- [3] Aaron BANGOR, Philip T. KORTUM et James T. MILLER. « An Empirical Evaluation of the System Usability Scale ». In : *International Journal of Human-Computer Interaction* 24.6 (29 juil. 2008), p. 574-594. ISSN : 1044-7318. DOI : 10.1080/10447310802205776.
- [4] Maxime BÉNÉRE. « Exploration d'une application adaptative pour évaluer les compétences en informatique ». Prom. Mens, Kim et Goletti, Olivier. Mém. de mast. Ecole polytechnique de Louvain, Université catholique de Louvain, 2020.
- [5] John BROOKE. « SUS-A quick and dirty usability scale ». In : *Usability evaluation in industry* 189.194 (1996), p. 4-7.
- [6] Lori CARTER. « Why students with an apparent aptitude for computer science don't choose to major in computer science ». In : *ACM SIGCSE Bulletin* 38.1 (3 mar. 2006), p. 27-31. ISSN : 0097-8418. DOI : 10.1145/1124706.1121352.
- [7] Guillaume DERVAL, Anthony GÉGO, Ludovic TAFFIN et al. *INGInious, GitHub repository*. URL : <https://github.com/UCL-INGI/INGInious> (visité le 27/08/2019).
- [8] Alexandre DEWIT et Jacques YAKOUB. « Source Code : un catalogue open source d'exercices informatiques ». Prom. Mens, Kim et Goletti, Olivier. Mém. de mast. Ecole polytechnique de Louvain, Université catholique de Louvain, 2020. URL : <http://hdl.handle.net/2078.1/thesis:24961>.
- [9] U.S. Department of HEALTH & HUMAN SERVICES. *The Research-Based Web Design & Usability Methods*. Juil. 2013. URL : <https://www.usability.gov/how-to-and-tools/methods/index.html> (visité le 03/04/2020).
- [10] Saul MCLEOD. *Likert Scale, Simply Psychology*. 3 août 2019. URL : <https://www.simplypsychology.org/likert-scale.html> (visité le 07/04/2020).

- [11] Kim MENS. « Conception orientée objet et gestion de données ». Cours universitaire. Ecole polytechnique de Louvain, Université catholique de Louvain. 2017-2018.
- [12] Kim MENS. « Programming paradigms ». Cours universitaire. Ecole polytechnique de Louvain, Université catholique de Louvain. 2019-2020.
- [13] Sarantos PSYCHARIS et Maria KALLIA. « The Effects of Computer Programming on high school students' reasoning skills, and self-efficacy and problem solving in Mathematics ». In : *Instructional Science* 45 (19 oct. 2017), p. 583-602. DOI : 10.1007/s11251-017-9421-5.
- [14] Ken SCHWABER. « SCRUM Development Process ». In : *Business Object Design and Implementation*. Sous la dir. de Jeff SUTHERLAND et al. London : Springer, 1997, p. 117-134. ISBN : 978-1-4471-0947-1. DOI : 10.1007/978-1-4471-0947-1_11.
- [15] *SI² : Sciences Informatiques pour le Secondaire Inférieur*. 23 fév. 2018. URL : <https://sicarre.be/> (visité le 21/08/2019).

Annexe A

Guide d'installation

Ce guide est un guide d'installation simplifié d'INGInious [2]. Il vous permettra d'obtenir une installation basique de la plateforme en local ou sur un serveur. Pour un guide plus détaillé, consultez la documentation¹.

Pour chaque étape la liste des commandes est fournie pour Ubuntu 16.04+.

1. Tout d'abord créer les dossiers nécessaires à l'installation. Vous aurez besoin de dossiers pour le répertoire Github, pour les exercices et pour les backups ainsi qu'un emplacement pour stocker le fichier de configuration.

```
$ mkdir [Dossier d installation]; cd [Dossier d instalation]
$ mkdir App; mkdir Filesystem; mkdir Filesystem/Task; \
mkdir Filesystem/Backup;
```

2. Installer les dépendances d'INGInious.

```
$ apt install git mongodb docker.io python3 python3-pip
$ pip3 install docker docutils pymongo PyYAML web.py Jinja2 lti oauth2 \
httplib2 watchdog msgpack pyzmq natsort psutil zipstream WsgiDAV \
pytidylib sh
```

3. Cloner le répertoire Github.

```
$ git clone https://github.com/UCL-INGI/INGInious.git \
[Dossier d instalation]/App
```

4. Ajoutez le groupe docker à l'utilisateur courant et redémarrer pour appliquer les changements.

```
$ sudo usermod -aG docker ingi
$ reboot
```

1. https://inginius.readthedocs.io/en/v0.6/install_doc/installation.html

5. Démarrez les services de Dockers et MongoDB.

```
$ systemctl start mongod
$ systemctl enable mongod
$ systemctl start docker
$ systemctl enable docker
```

6. Ajouter le répertoire Github à la variable d'environnement PATH afin de pouvoir lancer les scripts INGINious depuis n'importe quel dossier.

```
$ export PATH="[Dossier d installation]/App:$PATH"
```

7. Ajoutez les variables d'environnement relatives à INGINious. Celles-ci correspondent respectivement au chemin vers le fichier de configuration, l'adresse du serveur (localhost pour une installation locale) et le port (par défaut 80 pour HTTP, 443 pour HTTPS et 8080 pour une installation locale).

Ces variables peuvent aussi être définies de manière permanente² ou être passées en argument du script inginiuous-webapp.

```
$ export INGINIOUS_WEBAPP_CONFIG=\
"[Dossier d installation]/Filesystem/configuration.yaml"
$ export INGINIOUS_WEBAPP_HOST="localhost"
$ export INGINIOUS_WEBAPP_HOST="8080"
```

8. Lancez le script d'installation d'INGINious en précisant le chemin vers le fichier de configuration. Pour une installation basique, vous pouvez garder les valeurs par défaut à l'exception des dossiers pour les exercices et les backups ([Dossier d'installation]/Filesystem/Task et [Dossier d'installation]/Filesystem/Backup) et des paramètres relatifs à la création du compte administrateur.

```
$ inginiuous-install --file [Dossier d installation]/Filesystem/configuration.yaml
```

9. Afin de permettre l'inscription sur la plateforme vous devez aussi permettre l'envoi de mail à partir de votre adresse mail. Pour ce faire, ajoutez ces paramètres au fichier de configuration en remplaçant les valeurs par celles relatives à votre adresse mail.

smtp :

```
sendername: INGINious <no-reply@inginiuous.org>
host: smtp.gmail.com
port: 587
username: votre.username
```

2. https://doc.ubuntu-fr.org/variables_d_environnement#variables_d_environnement_persistantes

```
password: MotDePasse  
starttls: true
```

10. Vous êtes maintenant prêts à lancer INGIInious.

```
$ inginius-webapp
```

Annexe B

Exemple de structure d'un cours

```
- id: section_1
  rank: 0
  sections_list:
  - id: subsection1
    rank: 0
    tasks_list:
      01_getting_started: 0
      02_courses: 1
      03_tasks: 2
    title: Sub-Section1
  - id: subsection_2
    rank: 1
    tasks_list:
      04_run_student: 0
      05_fetching_input: 1
      06_feedback: 2
    title: Sub-Section 2
  title: Section 1
- id: section_2
  rank: 1
  sections_list:
  - id: subsection_3
    rank: 0
    tasks_list:
      07_advance_feedback: 0
      08_archive: 1
    title: Sub-section 3
```

```
- id: subsection_4
  rank: 1
  tasks__list:
    09__ssh__debug: 0
    10__environments: 1
  title: Sub-section 4
title: Section 2
```


Annexe C

Protocole de validation

Contexte général

Vous êtes professeur de mathématique dans l’enseignement secondaire. L’établissement pour lequel vous travaillez a pour projet de proposer un cours d’informatique à ses étudiants. Intéressé(e) par le sujet, vous vous portez volontaire pour donner ce cours.

Après discussion avec les différents acteurs concernés, vous décidez d’utiliser Python comme langage de programmation. De plus, vous devez prendre en compte le fait que vos étudiants n’ont généralement pas d’expérience en informatique.

Afin de mener à bien ce projet, vous allez devoir proposer à vos futurs étudiants toute une série d’exercices informatiques. Heureusement, un collègue vous a parlé(e) d’un exerciceur informatique venant d’ajouter de nouvelles fonctionnalités afin de mieux aider les enseignants dans votre situation. Il s’agit de la plateforme INGINious.

Vous décidez donc d’essayer cette plateforme et après avoir créé votre compte, vous voilà prêt(e) à vous lancer.

Votre compte

- **Nom d’utilisateur** : teacher
- **Mot de passe** : IngiProf

Scénario 1 - Importer un template

Contexte

Pour commencer, vous allez vous servir de la fonctionnalité Template afin de créer un nouveau cours à partir d'une base existante.

Marche à suivre

1. Commencez par aller sur la plateforme <http://tfe-simon.info.ucl.ac.be> et connectez-vous avec le compte de la page précédente.
2. Cherchant à créer un cours à partir d'un template, vous cliquez sur l'onglet Templates.
3. Vous arrivez sur la liste des templates, trouvez celui qui correspond à vos besoins (cf. Contexte général) et le consultez plus en détail.
4. Ayant des connaissances limitées dans le domaine, vous consultez plus en détail les descriptions et compétences du chapitre **Structures de données** et de la section **Mission 7 - Dictionnaires**.
5. Ce template vous convient. Vous décidez donc de créer un cours à partir de celui-ci. Vous voulez que ce nouveau cours ait l'identifiant **python-intro**. Si un autre utilisateur effectue la validation en même temps que vous il se peut que vous deviez lui donner un autre identifiant.
6. Vous décidez d'appeler votre nouveau cours **Introduction à la programmation**.
7. Vous décidez aussi que votre cours doit toujours être accessible aux étudiants.
8. Vous n'oubliez pas de sauvegarder et allez consulter le résultat sur la page principale du cours.
9. Ne vous souvenant plus de la matière abordée dans la section **Mission 2 - Base de programmation** vous retournez sur la page du template pour vous renseigner. Pour ce faire, utilisez le lien vers le template source disponible dans la partie administration du cours.

Scénario 2 - Éditer la structure d'un cours

Contexte

Maintenant que vous avez une base pour votre cours, vous allez pouvoir quelque peu la modifier. En particulier, vous allez modifier la structure du cours pour vous adapter aux besoins spécifiques de votre cours.

Marche à suivre

1. Retournez sur la page principale du cours créé lors du scénario 1. Si vous lui avez donné le bon identifiant, celle-ci devrait se trouver à cette adresse <http://tfe-simon.info.ucl.ac.be/course/python-intro>.
2. Allez dans l'administration du cours et sélectionnez l'onglet d'édition de la structure afin de modifier cette dernière.
3. Le temps étant limité, vous décidez de retirer l'entièreté du dernier chapitre sur les **Objets**. Vous décidez aussi de retirer la section **Mission 6 - Fichiers et exceptions**, car celle-ci ne rentre pas dans votre programme.
4. Vous décidez qu'il serait plus pertinent de voir les tuples (mission 5) avant les listes (mission 4) et choisissez d'inverser l'ordre de ces deux missions.
5. Maintenant que vous avez échangé ces deux sections, leurs titres ne correspondent plus à la numérotation logique. Modifiez-les pour que ce soit à nouveau le cas.
6. Il en va de même pour les exercices de ces sections. Modifiez le nom du premier exercice de chacune de ces sections (QCM) et laissez les autres pour plus tard.
7. Sachant que les deux premières semaines sont malheureusement très chargées vous décidez de retirer les exercices sur les QCM des deux premières missions.
8. Vous décidez finalement de créer une nouvelle section dans **Notions de base** pour y mettre les QCM. Naturellement, vous appelez cette section **QCM**.
9. Faites glisser le QCM de la mission 3 dans cette nouvelle section et ajoutez y aussi les exercices **Session 1 : QCM** et **Session 2 : QCM**.
10. Après vous être rendu compte que d'autres exercices de QCM étaient disponibles, vous décidez de les ajouter à la section QCM et de déplacer cette section à la fin du cours.
11. Sauvegardez la structure et allez constater le résultat (allez sur la page principale du cours puis sur vue en sections).

Scénario 3 - Créer un template

Contexte

Suite à vos modifications, vous estimez maintenant avoir un cours différent du template initial et valant la peine d'être partagé. Vous décidez donc de l'exporter comme template afin de le partager à tout le monde.

Marche à suivre

1. Retournez sur la page d'administration du cours créé et modifié dans les précédents scénarios. Si vous lui avez donné le bon identifiant, celle-ci devrait se trouver à cette adresse <http://tfe-simon.info.ucl.ac.be/admin/python-intro>.
2. Exportez ce cours comme template avec l'ID **python1-secondaire**. Si un autre utilisateur effectue la validation en même temps que vous, il se peut que vous deviez lui donner un autre identifiant.
3. Vous décidez de nommer votre template **Python1 - Niveau secondaire** et de le rendre public afin que tout le monde puisse l'utiliser. Comme d'habitude, n'oubliez pas de sauvegarder vos changements.
4. Allez sur l'onglet pour éditer les descriptions du template.
5. Remplacez la courte description par celle-ci : **Une introduction à Python pour les étudiants de secondaire débutant en informatique.**
6. Modifiez la description du template pour ne garder que les deux premiers paragraphes.
7. Ajoutez **Gérer son temps** à la liste des compétences
8. Pour la section QCM, écrivez la description suivante : **Une série de questions à choix multiple afin de tester vos connaissances sur les différentes missions**
Et écrivez les compétences suivantes :
 - **Gérer son temps**
 - **Méthodologie**
9. N'oubliez pas de sauvegarder vos changements.
10. Il vous est ensuite demandé de supprimer ce template dans l'onglet **zone dangereuse** afin de permettre aux autres utilisateurs de pouvoir effectuer la validation à leur tour.

Scénario 4 - Importer, exporter des exercices et des sections

Contexte

Dans un esprit de partage, vous décidez d'échanger des exercices et sections avec vos collègues. Pour ce faire, vous allez utiliser la fonction d'import et d'export.

Marche à suivre

1. Retournez sur la page d'administration du cours créé et modifié dans les précédents scénarios. Si vous lui avez donné le bon identifiant, celle-ci devrait se trouver à cette adresse <http://tfe-simon.info.ucl.ac.be/admin/python-intro>.
2. Vous décidez d'ajouter un exemple d'examen blanc fourni par votre collègue. Pour ce faire, sélectionner l'onglet d'**Import/Export** et importer l'archive fournie par ce dernier et disponible à cette adresse https://drive.google.com/uc?export=download&id=1HF18pHc3vKn-yLj_WybSAVulymEzoOL-.
3. En retour, vous proposez de lui envoyer quelques exercices et sections susceptibles de pouvoir l'intéresser. Pour ce faire, allez dans la partie d'exportation d'exercices et sélectionnez les exercices **Session 7 : Egalité entre structures** et **Session 7 : Création de dictionnaire**.
4. Dans l'onglet sections sélectionnez le chapitre **Notions de base** à l'exception de la section **Mission 2 - Base de programmation**.
5. Revenez à l'onglet exercices et désélectionnez les questions de bilan final (QBF).
6. Exportez votre sélection.
7. Vous disposez de plus de temps pour votre cours et décidez de voir les objets. Pour ce faire, allez dans la partie d'importation à partir d'un template et choisissez le template utilisé dans le scénario 1.
8. Sélectionnez le chapitre **Objets** à l'exception de la section **Mission 12 - Extras** et importer la sélection.
9. Allez consulter le résultat.
10. Il vous est ensuite demandé de supprimer ce cours dans l'onglet **zone dangereuse** de l'administration du cours afin de permettre aux autres utilisateurs de pouvoir effectuer la validation à leur tour.
11. Remplir le questionnaire à l'adresse <https://forms.gle/1XkgcsXHhApH4ud27>.

Annexe D

Questionnaire de validation

Cette annexe présente les questions posées aux utilisateurs après la validation. Elles ont été posées sous la forme d'un Google Form disponible à l'adresse <https://forms.gle/vBnqFcEBQCH6QKRm8>. La section D.1 présente la liste des questions posées ainsi que leurs types :

- [QO] : Question ouverte
- [QCM] : Question choix multiple
- [EL] : Question à choix unique basé sur l'échelle de Likert [10]

D.1 Questions

Général

- Quel âge avez-vous ? [QCM]
- À quelle fréquence utilisez-vous un ordinateur ? [QCM]
- Vous sentez-vous à l'aise avec l'utilisation d'un ordinateur ? [QCM]
- Êtes-vous familier avec la plateforme INGINious ? [QCM]

Qualité de la validation

- J'ai compris toutes les tâches proposées. [EL]
- J'ai compris l'utilité de toutes les tâches proposées. [EL]
- Si non, quelle(s) tâche(s) n'avez vous pas comprise(s) et pourquoi ? [QO]

Réussite des tâches

- Avez-vous pu réaliser toutes les tâches proposées ? [QCM]
- Si non, quelle(s) tâche(s) n'avez vous pas réussie(s) et pourquoi ? [QO]

Fonctionnalités

- J'ai pu réaliser les tâches proposées facilement. [EL]
- Les informations étaient présentées de manière logique et étaient facilement trouvables. [EL]
- J'ai pu facilement trouver les boutons pour effectuer chaque action. [EL]
- J'ai parfois eu des doutes sur l'utilité de certains boutons. [EL]
- L'utilisation des fonctionnalités m'a semblé intuitive. [EL]
- Je sais facilement me repérer et naviguer d'une page à l'autre de la plateforme. [EL]

Usabilité

- En tant qu'enseignant et dans le cadre professionnel, je pense que j'apprécierais utiliser cette plateforme. [EL]
- Je trouve le système inutilement complexe. [EL]
- J'ai trouvé la plateforme facile d'utilisation. [EL]
- Je pense que j'aurais besoin de l'aide d'une personne qualifiée pour utiliser la plateforme. [EL]
- Je trouve que les différentes fonctions de la plateforme étaient bien intégrées. [EL]
- J'ai trouvé qu'il y avait trop d'inconsistances sur la plateforme. [EL]
- J'imagine que la plupart des gens apprendront à utiliser la plateforme assez rapidement. [EL]
- J'ai trouvé la plateforme très lourde à utiliser. [EL]
- J'étais très à l'aise en utilisant la plateforme. [EL]
- J'ai eu besoin d'apprendre beaucoup de choses avant de pouvoir utiliser la plateforme. [EL]

Feedback libre

- Scénario 1 - Importer un template [QO]
- Scénario 2 - Éditer la structure d'un cours [QO]
- Scénario 3 - Créer un template [QO]
- Scénario 4 - Importer, exporter des exercices et des sections [QO]
- Feedback global [QO]

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain

Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | www.uclouvain.be/epl