



CHANET Zoran
CHARLOT Servan

Réseaux Informatiques et Communication Multimédia
Rapport de Projet 5^{ème} année



CONTRIBUTION AU PROJET EDCAMPUS

Tome Principal

Du 28 Janvier au 15 Mars 2019

Table des matières

1	Rappel du sujet	2
2	Technologies employées	2
3	Architecture technique	2
4	Réalisations techniques	3
4.1	Correction de bugs	3
4.1.1	Système de fichiers	3
4.1.2	Système de droits	3
4.1.3	Système de tâches (Kanban)	4
4.1.4	Corrections diverses (système de mail, appels serveur...)	4
4.2	Mise en place de techniques DevOps	5
4.2.1	CI	5
4.2.2	Sonar	5
4.3	Création de la vitrine	5
4.3.1	Vitrine d'université	5
4.3.2	Vitrine de projet	6
4.4	Autres	6
4.5	Difficultés	6
4.5.1	Tests unitaires	6
5	Gestion de projet	7
5.1	Rôles des membres	7
5.2	Méthode de travail	7
5.3	Gestion des risques	7
5.4	Planning prévisionnel et effectif	7
6	Outils	7
7	Métriques logiciels	7
7.1	Lignes de code	7
7.2	Langages	8
7.3	Performance	8
7.4	Temps ingénieur	8
8	Conclusion	9
9	Glossaire	10

1 Rappel du sujet

La plateforme EDCampus est une plateforme de gestion de projet développée par Waza Education, puis achetée et maintenue par Disrupt Campus Grenoble. Pensée pour le web, cette plateforme vise à rassembler de nombreux outils utilisés par les universités et les étudiants pour les projets universitaires, afin de faciliter une collaboration interdisciplinaire entre étudiants, universités et entreprises. Parmi les fonctionnalités proposées se trouvent le dépôt de documents et de livrables, un tableau Kanban, un mur de post-it, la gestion de l'avancée des tâches, la production de diagrammes de Gantt, l'évaluation des projets, une vitrine de projets...

Cependant, cette plateforme ambitieuse a encore beaucoup de manquements, et nombre de ces fonctionnalités ne sont pas correctement implémentées. Il a donc été demandé que des étudiants viennent contribuer au développement et à la correction de la plateforme, afin d'en accélérer la mise en production.

2 Technologies employées

Le front-end de la plateforme a été majoritairement développé en utilisant la technologie AngularJS, venant en renfort à la base HTML/CSS de la plateforme. Il reste également quelques pages produites par des fichiers PHP, reliquats d'une version précédente de la plateforme.

Le back-end de la plateforme est basé sur le langage PHP, et s'articule autour d'une base de données MySQL.

Le serveur est hébergé sur une machine virtuelle universitaire à l'aide d'Apache. Ce serveur Apache, comme la base de données et les autres composants, est isolé dans un conteneur Docker.

3 Architecture technique

Le projet possède une architecture client-serveur basée sur Apache. On peut aussi remarquer un début d'architecture MVC dans le front-end AngularJS, mais cela reste assez brouillon. La figure 1 représente le lien entre le front-end et le back-end.

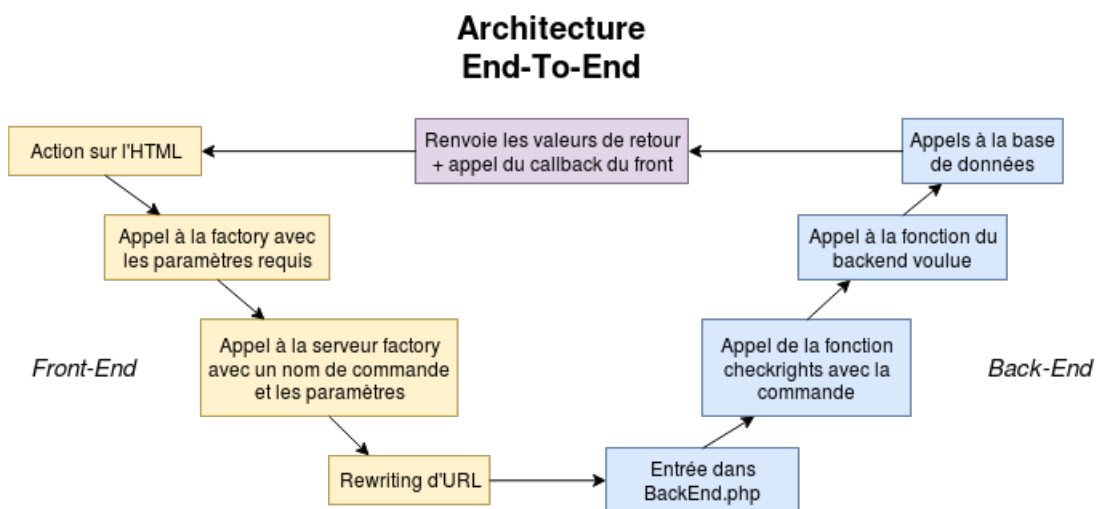


FIGURE 1 – Architecture End-to-End

4 Réalisations techniques

4.1 Correction de bugs

4.1.1 Système de fichiers

Correction complète du système de fichiers qui n'était pas utilisable en l'état. Liste des bugs qui ont été corrigés :

- Les fichiers du projet parent ne s'affichent pas dans le projet fils (en tant que fichiers hérités).
- Erreur d'actualisation si on supprime un fichier.
- Tous les dossiers sont affichés comme des fichiers et nécessitent une interaction pour devenir des dossiers.
- Le bouton retour au dossier parent ne fonctionne pas.
- On ne peut pas entrer dans un dossier hérité.
- Non suppression des fichiers au sein d'un dossier lors de sa suppression.
- Les fichiers ayant un numéro pour nom ne peuvent pas être renommés.
- Lorsque l'on supprime un fichier dans un dossier, on est renvoyé à la racine du projet plutôt que dans le dossier.
- On ne peut pas déplacer un fichier/dossier vers le dossier parent en le glissant sur la flèche retour.
- Correction du système de droits sur les fichiers (détaillé en section suivante).

Pour plus de détails sur la correction de ces bugs, consulter le journal sur le Air[1].

4.1.2 Système de droits

Le système de droits de EDCampus est basé sur un long fichier PHP en back-end, qui vérifie si la commande appelée par le front-end peut être exécutée. Cela se passe principalement dans la fonction `checkRights`. Voici dans les grandes lignes son fonctionnement :

- La commande en cours est envoyée à `checkRights` sous forme de string.
- On vérifie d'abord si elle fait partie des commandes publiques, que l'on peut exécuter lorsque l'on a pas de session. (Via une comparaison avec un tableau contenant toutes les commandes publiques.)
- Si ce n'est pas le cas, on récupère ensuite les informations de session et l'identifiant utilisateur.
- Si on ne trouve pas d'utilisateur, on regarde si on essaie d'accéder à une page publique, sinon on bloque l'accès.
- On a maintenant un utilisateur avec une session, qui essaie d'agir sur une page non publique.
- On vérifie si l'utilisateur est super-administrateur.
- On vérifie si l'utilisateur essaie d'exécuter une commande administrateur, en tant qu'administrateur.
- Puisque la page est non-publique et que l'on n'est pas administrateur, il s'agit forcément d'une page de projet.
- On vérifie si l'utilisateur est propriétaire du projet parent et que cette commande est autorisée.
- On vérifie si l'utilisateur est membre du projet parent et que cette commande est autorisée.
- On vérifie si l'utilisateur est propriétaire du projet courant et que cette commande est autorisée.
- On vérifie si l'utilisateur est membre du projet courant et que cette commande est autorisée.
- Si aucun des tests ci dessus n'a échoué (ou n'a déjà accordé l'accès), on accorde l'accès.

Des corrections ont été nécessaires dans cette fonction et les fonctions qu'elle appelle. En effet, les

tableaux retenant les commandes autorisées pour un certain test étaient très mal faits. La majorité des commandes n'était pas écrite dans le bon format, d'autres avaient le nom de la fonction en back-end plutôt que le nom de la commande, d'autres n'étaient pas dans le tableau alors qu'elles auraient dû y être, et vice-versa. Le cas de l'utilisateur administrateur n'était pas traité, il a donc fallu ajouter un test pour donner les bons droits à ce dernier.

Il existe une surcouche à ce système de droits pour les fichiers, qui a dû être corrigée elle aussi car les utilisateurs pouvaient faire ce qu'ils voulaient dans les fichiers du projet tant qu'ils avaient le lien (voir figure 2).

Droits courants		Admin			Tuteur			Etudiant	
		Owner (Project)	Membre	Non membre	Owner (Project)	Membre	Non membre	Membre	Non membre
Fichier normal	Accéder (dossier)	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé
	Déplacer	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé
	Dupliquer	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé
	Importer	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé
	Partager	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé
	Renommer	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé
	Supprimer	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé
	Telecharger	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé
	Voir	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé
Fichier hérité	Accéder (dossier)	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé
	Déplacer	Bug	Bug	Bug	Bug	Bug	Bug	Bug	Bug
	Dupliquer	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé
	Partager	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé
	Renommer	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé
	Supprimer	Autorisé	Interdit	Interdit	Autorisé	Interdit	Interdit	Interdit	Interdit
	Telecharger	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé
		Voir	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé
Droits voulus		Admin			Tuteur			Etudiant	
		Owner (Project)	Membre	Non membre	Owner (Project)	Membre	Non membre	Membre	Non membre
Fichier normal	Accéder (dossier)	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	N/A	Autorisé	N/A
	Déplacer	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	N/A	Autorisé	N/A
	Dupliquer	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	N/A	Autorisé	N/A
	Importer	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	N/A	Autorisé	N/A
	Partager	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	N/A	Autorisé	N/A
	Renommer	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	N/A	Autorisé	N/A
	Supprimer	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	N/A	Autorisé	N/A
	Telecharger	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	N/A	Autorisé	N/A
	Voir	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	Interdit	Autorisé	Interdit
Fichier hérité	Accéder (dossier)	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	N/A	Autorisé	N/A
	Déplacer	Autorisé	Autorisé	Autorisé	Autorisé	Interdit	N/A	Interdit	N/A
	Dupliquer	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	N/A	Autorisé	N/A
	Partager	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	N/A	Autorisé	N/A
	Renommer	Autorisé	Autorisé	Autorisé	Autorisé	Interdit	N/A	Interdit	N/A
	Supprimer	Autorisé	Autorisé	Autorisé	Autorisé	Interdit	N/A	Interdit	N/A
	Telecharger	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	N/A	Autorisé	N/A
		Voir	Autorisé	Autorisé	Autorisé	Autorisé	Autorisé	Interdit	Autorisé

FIGURE 2 – Système de droits des fichiers

4.1.3 Système de tâches (Kanban)

Chaque projet possède un tableau Kanban, qui fonctionnait assez bien de base mais qui a nécessité quelques corrections dans le front-end à cause de problèmes d'actualisation. Les labels sur les tâches ont aussi été modifiés pour que chaque couleur ait un nom unique dans le projet.

4.1.4 Corrections diverses (système de mail, appels serveur...)

De nombreuses fonctionnalités de la plateforme étaient disfonctionnelles à cause d'un manque de mise à jour du code, et difficiles à diagnostiquer à cause de l'absence de tests unitaires.

Parmi celles-ci se trouve le système de mail. La plateforme communique avec un compte SendGrid, qui permet l'utilisation et l'envoi automatique depuis une API de mails formatés. Certains appels

à l'API étaient erronés. De plus, il nous a fallu mettre à jour chacune des templates de mail, afin de retirer toute occurrence du nom de l'entreprise à qui le code a été racheté.

Une autre erreur à laquelle nous avons été confrontés régulièrement est le formatage des appels au serveur. La plateforme ayant beaucoup évolué au cours de son développement, son architecture s'est également modifiée. Certains appels au serveur étaient donc faits d'une manière obsolète, et n'obtenaient plus les bons résultats. Il nous a donc fallu corriger ces requêtes, afin qu'elles soient correctement interprétées par le serveur.

Il y a aussi un bug critique qui fut découvert pendant la hackathon. Lorsqu'un utilisateur possède un accent dans son nom (ex : Amélie), au moment de la création du login la chaîne de caractères est mal interprétée et le login est tronqué au niveau de l'accent (ex : amY). L'architecture fait que lors du transfert du nom du front-end vers le back-end, l'accent est codé avec des caractères classiques, qui sont ensuite interprétés correctement lors de l'ajout en base. Mais cela veut dire que pendant tout le passage en back-end on travaille avec des accents codés. Le login était créé à partir du nom et du prénom de l'utilisateur, qui peuvent donc être temporairement corrompus par des accents codés. La correction de ce souci est donc de détecter la présence d'accents dans les noms, et si il y en a, créer le login à partir de la première partie de l'adresse mail de l'utilisateur (avant le @), sinon, on fait comme avant avec les noms.

4.2 Mise en place de techniques DevOps

4.2.1 CI

Le code de la plateforme est hébergé sur un serveur Gitlab. Nous avons donc exploité les possibilités de CI/CD du serveur pour mettre en place un pipeline automatique lors de l'ajout ou de la modification de code dans le dépôt. Ceci étant mis en place, il peut être utilisé afin de performer des actions automatisées sur le code à chaque ajout.

4.2.2 Sonar

Nous avons alors ajouté à chaque utilisation du pipeline un scanner Sonar, permettant d'analyser le code produit à chaque passage, afin de détecter les erreurs, les vulnérabilités, les duplications...

Ces analyses sont envoyées directement sur le serveur SonarCloud, permettant la visualisation des résultats.

4.3 Création de la vitrine

Il nous a été demandé de corriger et compléter une fonctionnalité qui n'était plus proposée par la plateforme depuis de nombreuses versions. Il s'agit d'une vitrine permettant la visualisation simple et rapide des projets passés, en cours ou proposés par un université, aux cotés d'autres informations utiles sur l'université en question. Pour compléter cette vitrine, la création de toute pièce d'une vitrine de projet nous a été confiée, afin d'afficher clairement les informations liées à un projet en particulier.

4.3.1 Vitrine d'université

Comme mentionné plus haut, cette vitrine était pré-existante. Mais le code gérant cette vitrine utilise des méthodes liées à d'anciennes versions de la plateforme, il a donc fallu les corriger. La correction de nombreuses requêtes faites au serveur a également été nécessaire. Il nous a enfin fallu adapter l'affichage des projets, afin de ne montrer que les projets souhaités, en explicitant le nom de ceux qui dépendent d'un projet parent. Enfin, un clic sur un projet affiché renvoie l'utilisateur vers la vitrine dudit projet.

4.3.2 Vitrine de projet

La vitrine de projet est une page qu'il nous a fallu construire de toute pièce. Elle présente les informations utiles et intéressantes concernant le projet choisi. Pour construire cette page, nous avons utilisé un composant AngularJs, afin de ne pas reproduire les erreurs de la vitrine d'université, développée dans un langage qui n'est plus utilisé dans le reste du front-end du site. Accéder aux informations qu'il nous a été demandé de mettre en avant a nécessité des changements dans le système de droits (décrit en section 4.1.2), car certaines informations étaient jusqu'alors considérées comme privées.

Enfin, pour la section des livrables, nous avons pris en charge les fichiers et les liens externes de différentes manières :

- Pour télécharger un fichier qui a été déposé, il est nécessaire de se connecter, sans quoi le livrable est affiché comme soumis, mais ne peut être lu.
- Les liens externes sont accessibles même sans se connecter sur la plateforme.
- Un lien externe pointant vers une vidéo hébergée sur Youtube, Dailymotion ou Vimeo sera interprété, et la vidéo en question sera affichée directement sur la vitrine.
- Un lien externe pointant vers une image sera interprété, et une version réduite de l'image sera affichée directement sur la vitrine, ouvrant l'image dans un nouvel onglet si l'on clique dessus.
- Un lien externe autre sera représenté par un bouton, ouvrant le lien dans un nouvel onglet si l'on clique dessus.

4.4 Autres

Il a été décidé, pour sécuriser les connexions, de sécuriser les échanges en utilisant l'HTTPS. Pour cela, il est nécessaire de générer un certificat, qui permettra aux navigateurs internet d'authentifier le site lorsqu'ils y accèdent. Pour que les navigateurs fassent confiance à un site internet, ce certificat a besoin d'être signé par une autorité de certification. L'autorité mondiale de certification Let's Encrypt propose gratuitement ce genre de certificats. Nous avons donc choisi de recourir à Let's Encrypt pour certifier notre plateforme.

La mise en place de Let's Encrypt est somme toute assez simple. Cependant, la plateforme étant placée dans un conteneur Docker, il nous a fallu agir non seulement sur la configuration du serveur Apache (à l'intérieur du conteneur), mais aussi sur la configuration même du conteneur et de son lien avec la machine hôte, afin de permettre la communication via le port dédié à l'HTTPS.

4.5 Difficultés

4.5.1 Tests unitaires

Le pipeline de CI mis en place permet l'exécution automatique de tests unitaires. Cependant, nous n'avons pas mis en place ces tests, car c'est l'architecture même de la plateforme qui rend difficile la production de ces tests : les frameworks existants ne permettent pas le test simulé de fonctions statiques, or le code complet du back-end fait recours uniquement à des fonctions statiques. Par manque de temps, il a été décidé que nos efforts devraient se tourner d'abord vers la création de la vitrine (voir section 4.3) avant de réfléchir à des moyens de contournement.

5 Gestion de projet

5.1 Rôles des membres

- **Anthony Geourjon** : Chef de projet (Externe)
- **Servan Charlot** : Chef de groupe - Dev lead
- **Zoran Chanet** : DevOps master - Dev

5.2 Méthode de travail

La méthode de travail mise en place était fortement inspirée des méthodes agiles, proche de Kanban et Scrum : les tâches donnaient lieu à des cartes, placées sur un tableau disposant des trois colonnes classiques *ToDo*, *Doing* et *Done*. Chacun, lorsqu'il débutait une tâche, s'assignait la carte correspondante et se chargeait de gérer la position de la carte dans le tableau. Cependant, l'organisation en sprints prévue par la méthode Scrum n'a pas été strictement respectée : chaque fin de sprint donnait lieu à une réunion de retour, mais de nombreuses tâches étaient ajoutées au sprint pendant le sprint, et certaines se reportaient d'un sprint à l'autre.

5.3 Gestion des risques

Risque	Solution envisagée
Panne matérielle (ex : poste de travail)	Le code est conservé et versionné sur Gricad GitLab
Panne du serveur Gricad	Le code est également conservé localement
Une tâche prend plus de temps que prévu	Travailler au FabLab pour contacter le chef de projet
Comprendre le code est chronophage	Faire des rapports réguliers à l'équipe pour expliquer les découvertes sur l'organisation du code

5.4 Planning prévisionnel et effectif

- **20-22 février 2019** : Hack Ton Campus, la plateforme EDCampus sera utilisée pour la gestion des défis.
- **Avril 2019** : Projet pilote, un projet mêlant une entreprise, l'UGA et des étudiants servira de test à la plateforme.
- **Septembre 2019** : Mise en production globale.

6 Outils

De nombreux outils ont été utilisés pour la gestion de projet. Notre board Kanban était sur Jira, puis sur Gricad une fois que les étudiants de l'UHA n'étaient plus là. Nous utilisons Slack pour la discussion instantanée avec Anthony. Nous utilisons Google Drive pour stocker les fichiers et document à partager. Enfin, nous utilisons git et Gricad pour la gestion du code, avec Gitlab CI/CD et Sonar(cloud) pour l'intégration continue.

7 Métriques logiciels

7.1 Lignes de code

Résultat de l'analyse Cloc :


```
24101 text files .
17513 unique files .
5060 files ignored.
```

```
http://cloc.sourceforge.net v 1.60
T=191.43 s (77.7 files/s, 12184.7 lines/s)
```

Language	files	blank	comment	code
PHP	6993	93533	241148	799095
Javascript	6457	95135	201288	465337
HTML	725	152296	1611	151072
CSS	266	7417	2827	74990
LESS	74	1455	1597	9366
C#	65	2191	2811	8934
XSD	35	375	0	4303
ActionScript	37	941	212	3251
XML	66	108	35	2999
SASS	18	226	72	2604
YAML	70	138	81	1321
ASP.Net	2	69	11	958
make	12	111	14	409
Python	8	132	87	390
Bourne Shell	16	63	48	287
Bourne Again Shell	17	52	47	195
MSBuild scripts	1	0	10	192
Perl	4	41	25	189
Ant	3	27	8	176
Smarty	2	2	0	104
Maven	1	5	4	25
XAML	2	1	0	13
CoffeeScript	3	4	0	11
DOS Batch	1	0	0	4
SUM:	14878	354322	451936	1526225

7.2 Langages

Les langages utilisés sont majoritairement le PHP en back-end et l'AngularJS/HTML/CSS en front-end.

7.3 Performance

Nous n'avons pas de moyens pour quantifier les performances de l'application, mais à la vue du code du back-end (beaucoup d'appels en base assez lourds) et de l'agencement de la base (aucune clé étrangère), on se rend compte que cette version est très peu performante.

7.4 Temps ingénieur

Nous avons travaillé 16,5 jours (par personne), conformément aux créneaux de projet.

8 Conclusion

EDCampus est une plateforme intéressante par le nombre de fonctionnalités qu'elle propose. À terme, elle pourrait prétendre à remplacer en les rassemblant de nombreuses plateformes utilisées aujourd'hui pour la gestion de projets universitaires, telles que moodle, trello, AIR Imag... mais l'état actuel du projet ne permet pas de créer une base stable ni solide pour ce but. Le code est trop vérolé et approximatif pour que la construction de nouvelles fonctionnalités se fasse sans encombre, et la majorité des fonctionnalités accessibles à ce jour ne fonctionne qu'en surface. De plus, une trop grande partie du code et de la base de donnée est actuellement inutilisée, et alourdit le projet sans raison. Afin de mener à bien le développement de cette plateforme, il serait nécessaire de prendre le temps, avant de continuer à la développer, de l'étudier, l'épurer et la reproduire à partir d'une base plus saine, permettant un développement fiable et la mise en place de technologies DevOps plus poussées.

Ce projet nous a permis de manipuler la vieille manière de faire du web, avec du PHP et de l'AngularJS. Cela nous a aussi donné une expérience dans l'exercice de reprendre du code en très mauvais état, rempli de bugs et dénué de documentation ou de logique.

9 Glossaire

- Back-end : Application coté serveur.
- CD : Continuous Delivery, livraison continue.
- CI : Continuous Integration, intégration continue.
- Dev : Développeur/Développement.
- DevOps : Développement et Exploitation (Operation).
- Framework : Cadre de développement, imposant une architecture connue/normée.
- Front-end : Application coté client.
- HTTP : HyperText Transfer Protocol, protocole de transfert d'hypertexte, utilisé pour accéder aux pages web.
- HTTPS : HTTP Secure, surcouche de sécurité au protocole HTTP.
- Pipeline : Espace dans lequel des actions automatisées peuvent être mises en place sur le code poussé sur le dépôt.
- Statique : Se dit d'une fonction qui peut être appelée sans nécessiter d'instance d'un objet.
- Template : Modèle.
- UGA : Université Grenoble-Alpes.
- UHA : Université de Haute-Alsace.

Références

- [1] Wiki Air EDCampus. https://air.imag.fr/index.php/PROJET-INF05_1819_EDCampus.