

École polytechnique de Louvain

HaïtiWater

Développement d'une application web pour gérer
la distribution de l'eau en Haïti

Auteurs: **Céline DEKNOP, Adrien HALLET , Sébastien STREBELLE**

Promoteurs: **Kim MENS, Sandra SOARES FRAZAO**

Lecteurs: **Benoît DUHOUX, Olivier CARLIER**

Année académique 2018-2019

Master [120] en sciences informatiques

Table des matières

Abstract	3
1 Introduction	4
2 Contexte	9
2.1 Situation de l'eau à Haïti	9
2.2 Gestion actuelle	12
2.3 Comparaison avec d'autres pays	15
2.4 Comparaison avec des outils existants	17
3 Organisation	18
3.1 Approche de travail	18
3.2 Méthodologie	21
3.3 Répartition des tâches	23
4 Analyse des besoins	27
4.1 Besoins fonctionnels	27
4.2 Besoins non-fonctionnels	27
4.3 Cahier des charges	28
4.4 Structure des données	28
5 Implémentation	29
5.1 Choix technologiques	29
5.2 La hiérarchie dans l'application	37
5.3 Interface utilisateur	39
5.4 Procédure d'utilisation	41
5.5 Client	41
5.6 Serveur	42
6 Validation	43
6.1 Performances	43
6.2 Vérifications automatiques	43
6.3 Vérifications utilisateurs réels	44
7 Améliorations futures	45
7.1 Suite du projet	45
7.2 Défis rencontrés	45
7.3 Propositions	45
8 Conclusion	46
Bibliographie	47

Abstract

Page : 1

Chapitre 1

Introduction

Contexte

Ce mémoire se place dans le cadre d'un projet de développement lancé par l'ONG Protos¹. Protos a pour objectif, entre autres, d'améliorer l'accès à l'eau potable en milieu rural afin d'aider le développement de plusieurs pays du monde. Un des pays dans lesquels Protos s'est engagée est Haïti.

Une succession de crises politiques et catastrophes naturelles ces dernières décennies ont rendu l'accès à l'eau potable, entre autres, particulièrement complexe dans ce pays des Antilles. En 2010, un violent séisme a laissé le pays en ruine, détruisant beaucoup d'infrastructures, y compris de distribution d'eau. Des incertitudes politiques entravent la reconstruction de ces installations et les populations ne sont pas toujours aidées par le pouvoir politique pour assurer la distribution de l'eau, particulièrement dans les zones rurales. C'est une des raisons pour lesquelles l'ONG Protos est active dans le pays.

Protos a contacté l'UCL afin de l'aider dans un projet de développement en Haïti. L'objectif est de réaliser un système logiciel pilote pour la gestion de la distribution d'eau potable en zone rurale. En effet, aucune gestion centralisée organisée par l'Etat n'existe pour les zones rurales. Des réseaux existent, constitués de points de prélèvement d'eau, de conduites de distribution d'eau et de fontaines situées dans les villages, mais la gestion publique de ceux-ci n'est pas opérationnelle.

1. www.protos.ngo

Dès lors, les organismes locaux en charge de ces réseaux en assurent la gestion de manière indépendante, en se s'appuyant comme ils le peuvent sur les moyens et les acteurs locaux.

L'ONG Protos intervient dans ce cadre, pour proposer un appui à ces organismes locaux afin de mieux organiser cette distribution. Grâce à une meilleure organisation, il sera possible d'améliorer le taux de recouvrement des factures liées à la distribution d'eau, ce qui permettra d'assurer des rentrées financières à ces organismes qui sont également en charge de la maintenance physique des réseaux, mais sans recevoir de réels moyens de la part de l'Etat.

Ce projet est prévu sur trois ans, une première année pour la conception d'un prototype, une deuxième année pour finaliser le système tester son déploiement sur place et une troisième année pour le finaliser. Ce mémoire constitue la première partie de ce projet.

Problématiques

La structure hiérarchique des différents acteurs de la distribution de l'eau, gouvernementaux ou non, est assez complexe. Ces acteurs doivent s'organiser, individuellement et en groupes, pour arriver à une gestion efficace des ressources. Dans ce mémoire, nous nous concentrons sur trois axes afin d'améliorer leur coordination.

Communication afin de permettre aux différents acteurs de s'informer à partir de la même plateforme.

Collaboration pour uniformiser le format des données et donner un rôle défini à chacun.

Stockage de manière à rassembler l'information sous forme numérique, permettant une meilleure sauvegarde des données, et à terme de les traiter en large volume de manière statistique.

Motivation

Nous réalisons ce travail dans le cadre de nos études de Master en Sciences Informatiques. Le but de ce travail est de conclure celles-ci, et de mettre tous nos apprentissages en application dans un projet à grande échelle.

De plus, il s'agit d'un projet réel, avec de véritables acteurs, enjeux et un objectif d'utilisation à long terme. Cela entraîne de nouvelles problématiques pour nous, qui ne sont pas abordées dans le reste de notre formation.

Ce mémoire nous permet d'avoir une première expérience de développement d'une application web, partant de rien sauf des attentes de nos clients, et intégrant toutes les parties et les étapes de son développement.

Non contents de parfaire notre formation, nous avons pour objectif d'être utiles à Haïti. Nous espérons que l'application développée dans le cadre de ce mémoire pourra être utilisée sur place et avoir un réel impact positif, aidant la distribution de l'eau à Haïti et son développement.

Objectif

Le but final de ce mémoire est de délivrer une application à l'ONG Protos et aux acteurs de la distribution de l'eau à Haïti. Nous visons un déploiement de cette base applicative sur le terrain, et que les futures équipes de développement pourront travailler à sa maintenance et évolution.

Nous espérons également que cette application leur permettra d'avoir un bon exemple de solution logicielle. Cela pourrait leur aider à voir comment un logiciel peut les aider dans leur gestion, et les aider à bien repenser leurs besoins lors de projets futurs.

Approche

Avant de commencer le développement de l'application, nous avons effectué une longue phase de recherches. Cette phase est séparée en deux.

Durant la première partie, nous avons observé ce qu'il se faisait dans les systèmes existants de la gestion de l'eau en Europe pour avoir une meilleure vue d'ensemble du travail à accomplir et des possibilités. En seconde partie, nous avons analysé les documents fournis par l'ONG Protos afin d'avoir une idée de la solution actuellement déployée en Haïti et des améliorations possibles.

La phase de développement suivante fut l'analyse fonctionnelle et la conception. Nous avons établi nos propositions en terme de fonctionnalités pour l'application, ainsi qu'en terme de fenêtres et contrôles pour les présenter à Protos sous forme d'un cahier des charges.

Après validation de l'interface et des fonctionnalités, nous sommes passés à la phase de réalisation, durant laquelle nous avons implémenté ces fonctionnalités et les systèmes nécessaires à leur bon fonctionnement. Cette phase a été la plus longue étant donné le nombre de fonctionnalités à implémenter et la nécessité de documenter à la fois le code pour les équipes de développement ainsi que l'interface pour nos futurs utilisateurs.

Enfin, la validation. Nous avons présenté l'application et les fonctionnalités que nous avons implémentées à des utilisateurs, afin d'obtenir des retours et de nous assurer que l'application que nous avons développé répondait aux attentes.

Au cours de la phase de réalisation, nous avons pu collaborer avec un stagiaire venant d'Haïti. Cela nous a permis de comprendre plus directement les besoins sur place, complétant ainsi les informations plus théoriques que nous avions obtenues des documents de Protos. Cette vision plus pratique nous a permis d'avoir une meilleure idée de la réalité du terrain.

Contribution

Grâce à notre travail de fin d'études, nous avons permis à l'ONG Protos ainsi qu'aux acteurs de terrain à Haïti d'avoir un premier contact avec le monde du

développement logiciel. Nous espérons que celui-ci les aidera à comprendre les problématiques liées à la création de logiciels informatiques pour leurs projets futurs.

Nous avons également proposé une application pour la gestion de l'eau potable à Haïti. Nous espérons qu'elle pourra servir de base de travail, permettant aux acteurs de terrain d'avoir un jour un outil les aidant dans leur travail quotidien et apportant une aide au développement national.

Nous espérons avoir contribué à l'amélioration de la gestion de l'eau en Haïti, mais ce travail a assurément contribué au développement de nos compétences ; en tant que développeurs, de par l'utilisation de bibliothèques et frameworks peu ou pas connus, en tant qu'analystes grâce au problème réel posé et la nécessité de comprendre les besoins. La coordination du groupe, la planification du travail et la communication avec les clients ont également beaucoup apporté à notre développement professionnel.

Plan

Dans ce document, nous allons exposer le déroulement de notre travail de fin d'études. Tout d'abord, nous allons revenir sur le contexte de ce travail. Nous allons ensuite détailler notre approche pour sa réalisation. Après cela, nous étayerons notre analyse des besoins de l'ONG Protos et des acteurs haïtiens. Ensuite, nous expliquerons toutes les étapes et décisions de notre implémentation. Nous reviendrons sur les techniques que nous avons utilisées afin de valider notre implémentation. Nous clôturerons ce mémoire par une rétrospective sur le travail et une ouverture aux possibilités futures.

Chapitre 2

Contexte

2.1 Situation de l'eau à Haïti

Haïti est un pays qui doit actuellement faire face à une situation difficile. Éducation, économie, habitat, etc, sont autant de défis que le pays devra relever. L'eau potable, son assainissement, sa gestion et sa distribution font partie de ces défis, en particulier dans les zones rurales. Les zones d'action de l'ONG Protos sont particulièrement touchées. La majorité des familles s'approvisionne en eau à des fontaines publiques (96% des foyers) et effectue un trajet de plus de 30 minutes (58% des foyers). Il est de plus extrêmement difficile de maintenir et développer le réseau de distribution d'eau potable dans certaines régions en raison de la pauvreté extrême d'une grande partie de l'île. La zone étudiée de *Passe Catabois* (au nord du pays), par exemple, a un taux de recouvrement des factures de 11% seulement. Si la situation est complexe, il est toutefois possible d'identifier des facteurs expliquant l'état, entre autres, des infrastructures d'eau potable.

Problèmes environnementaux

Située dans les Caraïbes au sud-est de la République de Cuba et des États-Unis d'Amérique, la République d'Haïti occupe un territoire de $27.750km^2$ sur la partie ouest de l'île d'Hispaniola. Le pays entier souffre de sa position géographique [5] qui le place dans un risque permanent de catastrophes naturelles. La proximité de failles

tectoniques et de courants aériens et marins favorisant la formation de tempêtes tropicales produisent une succession de cataclysmes. De multiples tempêtes et séismes frappent l'île fréquemment. Le tristement célèbre tremblement de terre du 12 janvier 2010, dont le nombre de victimes est approximé à 250.000 morts et autant de blessés, a, à lui seul, causé des dégâts estimés de 8 à 14 milliards USD. [3]. Ces catastrophes périodiques détruisent les infrastructures et mettent à mal l'économie du pays, empêchant un développement du réseau d'eau potable, entre autres, suffisant pour répondre aux besoins.

Le climat haïtien n'aide pas la situation. Météorologiquement parlant, Haïti affronte une alternance entre saison très chaude et saison moins chaude, les deux provoquant fréquemment des sécheresses. Les différentes régions du pays ont des climats variés, obligeant l'adoption de plusieurs mesures pour sauvegarder la population et rend difficile une politique commune de gestion des ressources. Il convient de noter que l'aridité qui résulte de ce climat est accentuée par les activités humaines. La déforestation massive et rapide du pays, à des fins principalement énergétiques, empêche une rétention correcte des eaux dans les sols. Seuls 2% à 3% du territoire sont occupés par des forêts (venant de plus de 60% il y a un siècle). La sécheresse se trouve ainsi aggravée, et le ruissellement des eaux accélère l'érosion des sols [9]. Les terres sont ainsi progressivement emportées vers la mer. En 2008, un rapport des Nations Unies estimait cette perte à 37 millions de tonnes par an [7]. Les cours d'eaux ont donc un débit accéléré, parfois torrentiel, rendant difficile l'exploitation de l'écoulement naturel et nécessitant plus d'infrastructures humaines.

Problèmes politiques

Pays colonisé français devenu indépendant en 1804, la République d'Haïti est passée au travers de profondes réformes politiques. Le pays oscille politiquement jusqu'à une instabilité contemporaine relative (on dénombre 15 présidents et 22 premiers ministres depuis 1986 alors que le mandat est constitutionnellement valable 5 ans) et des tensions internes continuent de perturber le pays. Cela entraîne une difficulté à adopter des politiques de long terme, ou même une politique commune aux différents gouvernements.

La scène internationale s'est très largement mobilisée pour Haïti et tout particulièrement après le séisme de 2010. De nombreuses organisations humanitaires et non-gouvernementales sont arrivées, mais les aides financières sont soupçonnées de

détournement par ces mêmes organisations [4]. Ces aides sont parfois décriées dans la dépendance qu'elles créent et le manque de vision à long terme. La population s'est également déjà retournée contre les aides et notamment contre les casques bleus venu aider après le séisme. Des soldats népalais des Nations Unies, venus aider après le drame, auraient involontairement déclenché une grave épidémie de choléra en introduisant la maladie en tant que porteurs sains. Le *Center for Disease Control and Prevention (CDC)*, un service du département de la santé des États-Unis d'amérique, a confirmé cette hypothèse et les Nations Unies ont reconnu leur responsabilité en décembre 2016. L'épidémie a causé 10.000 morts pour 800.000 cas dans les Caraïbes, majoritairement en Haïti. Cette maladie a été introduite par des rejets d'eaux usées dans un cours d'eau, par négligence et manque d'installations permettant d'évacuer les eaux usées dans un campement d'aide humanitaire.

La République d'Haïti entretient également des relations conflictuelles avec la République Dominicaine, avec laquelle elle partage l'île d'Hispaniola. Les raisons sont multiples (travailleurs étrangers, tensions ethniques, etc) et un lourd passé pèse sur la relation entre les deux pays.

Nous voyons donc que la situation politique intérieure et internationale haïtienne est un cas complexe. Les diverses tensions et soupçons de corruption ne donnant certainement pas à la problématique de l'eau toute l'attention, ni tous les fonds nécessaires.

Problèmes sociaux

De ces conditions environnementales et politiques émane une population pauvre, voire abandonnée dans les zones rurales où des conseils de village tentent de pallier aux problèmes au quotidien. Là où les villes et principalement la capitale profitent des avantages de la mondialisation, la campagne haïtienne peine à bénéficier de l'aide nationale et repose essentiellement sur les organisations locales et externes au pays. L'éducation y est difficilement accessible et la population est très majoritairement endettée (81% des foyers haïtiens), situation aggravée depuis 2015 lorsque la monnaie locale, la Gourde (HTG) a cessé d'être maintenue à un taux d'échange fixe avec le Dollar (USD). La Gourde continue depuis sa dépréciation (1 Euro vaut 93,5445 Gourdes¹). La société haïtienne est donc pauvre et les inégalités nombreuses. La campagne en particulier, qui semble déjà oubliée financièrement, manque de techniciens et ingénieurs aptes à créer et maintenir un réseau de distribution d'eau potable avancé.

1. Taux de change du marché au 28 mars 2019

Problèmes organisationnels

Le modèle politique et organisationnel d'Haïti se calque sur la théorie occidentale. Cependant les profondes différences résultent en une mise en pratique des concepts qui ne sont pas toujours efficaces. On constate notamment un manque de collaboration entre les entités haïtiennes ou même entre villages. Particulièrement dans les zones rurales, les croyances locales (religions, magie) sont parfois consultées pour des décisions importantes au détriment de la rationalité. La communication entre les différentes parties du pays n'est pas optimale. Les entités gouvernementales sont parfois nommées sans moyens financiers ni même bureaux et se retrouvent désœuvrées. Les fréquentes crises détournent l'attention des problèmes de fond tels que l'éducation, les infrastructures, ...

2.2 Gestion actuelle

Organisation du pays

Comme exposé dans la section précédente, la situation environnementale, politique et sociale, cause une instabilité dans l'organisation d'Haïti. Les présidents changent souvent, et avec eux, les mesures mises en place. Les responsabilités des ministères et leurs sous-entités sont assez peu claires, ou impossibles à mettre en pratique.

Par conséquent, l'ONG Protos est sans doute l'une des entités les plus stables au niveau de l'organisation de la gestion de l'eau en Haïti. Elle supervise plusieurs organisations locales, en apportant son soutien (expérience, logistique, finances), tandis que ce sont les localités elles-mêmes qui tentent de solutionner les problèmes.

Chaque zone est traitée indépendamment (elle dispose de son ONG locale), mais on distingue tout de même trois groupes d'acteurs récurrents :

- Le triangle formé par les autorités locales, les fournisseurs et les usagers
- Le gouvernement national
- Les autres ONG (locales ou non)

Protos se repose beaucoup sur ces entités locales, notamment les services étatiques, comme l'Université d'État d'Haïti (qui aura une place dans ce projet à l'avenir). Également, elle tend à être la plus inclusive possible, et à impliquer un maximum d'acteurs (locaux) dans ses projets. L'idée est que les citoyens instruits servent de bon exemple pour répandre les avancées sociales et/ou technologiques à travers le pays. Selon eux, pour avoir un système fonctionnel, la conscientisation sur la problématique de l'eau devrait partir des habitants eux-même avant de remonter à l'état.

Actuellement, Protos agit dans deux zones du pays : Nord-Est (région dite "Moustiques") et Centre-Est (région dite "Onde Verte"). Celles-ci font partie des plus rurales, et ont été choisies car ce sont celles qui ont le plus besoin d'aide.

Structure organisationnelle

En plus de l'aide fournie par l'ONG Protos, il existe en Haïti plusieurs structures dédiées à la gestion de l'eau. Premièrement, la DINEPA (Direction Nationale de l'Eau Potable et de l'Assainissement), qui a pour mission d'exécuter la politique de l'état dans le secteur de l'eau potable et de l'assainissement autour de 3 grands axes : le développement du secteur au niveau national, la régulation du secteur et le contrôle des acteurs [1]. La DINEPA est sous contrôle direct du Ministère de Travaux Publics, Transports et Communication.

La DINEPA supervise des branches régionales appelées OREPA (Offices Régionaux d'Eau Potable et d'Assainissement), gérant chacune une zone du pays. Elles sont chargées de la prestation du service de distribution de l'eau potable en milieu urbain et rural.

Finalement, il existe des CAEPA (Comités d'Approvisionnement en Eau Potable et d'Assainissement), qui gèrent un réseau de distribution d'eau à un niveau local réduit. Ils sont moins formellement gérés par l'état et sont encore loin d'être en place sur tout le pays, mais ont prouvé leur efficacité. Surtout dans les régions rurales, les habitants se reposent bien plus sur les CAEPA pour leur approvisionnement en eau que d'autres entités privées ou publiques.

Procédures actuelles

On peut remarquer qu'il n'y a pas vraiment de standardisation dans les procédures en place. En effet, un comité de gestion de l'eau existe à Poste-Métier depuis plus de 20 ans, mais il n'est présent que grâce à l'implication du Pasteur Chrisbon Fraïs et à l'aide de Protos. Cet effort s'étant montré efficace, le système établi à Poste-Métier a été appliqué dans d'autres zones, avec plus ou moins de succès.

Le principal problème rencontré par les comités de l'eau est le recouvrement des factures. En effet, on constate que les habitants d'Haïti n'ont pas l'habitude d'avoir à payer pour l'eau, et rechignent à le faire. Premièrement parce qu'ils ne l'ont jamais fait, mais également car la religion a encore une place forte dans la société Haïtienne et l'eau y est considérée comme *don de Dieu*. Protos met en avant qu'une part essentielle du processus de stabilisation du réseau de distribution est l'éducation des habitants, pour qu'ils comprennent qu'ils ne payent pas l'eau elle-même, mais bien tout le service fourni par le réseau pour l'assainissement, le transport et la distribution.

Concrètement, si la zone de Poste Métier a un taux de recouvrement satisfaisant (89%), les autres zones comme Passe Catabois sont bien en deça (11%). Ceci est dû à plusieurs éléments : les habitants sont mieux informés sur la problématique de l'eau, mais les gestionnaires de fontaines ont aussi le pouvoir de couper l'accès à l'eau (à l'aide de cadenas) quand les factures ne sont pas payées. On remarque également que les modes de fonctionnements des différentes zones ne sont pas toujours les mêmes (pas toujours de bureau pour les réunions, règlements plus ou moins stricts).

En pratique, un comité de l'eau (*Komite Dlo* en Créole haïtien) est composé d'un président, un vice-président, un secrétaire, un trésorier et un conseiller, élus par les consommateurs locaux pour cinq ans. Ils se réunissent tous les mois pour parler des affaires courantes, et organisent une fois par an une assemblée générale où sont conviés les usagers du réseau. Ils sont en charge du bon fonctionnement du réseau, de la gestion du budget et des fonds amenés par les utilisateurs, de superviser les *comités fontaines* et de rendre des comptes à la population.

Les comités fontaines, eux, sont composés d'un responsable fontaine, un secrétaire et un trésorier. Il sont en charge du bon fonctionnement et de la qualité du point d'eau, d'apporter les fonds au comité d'eau et de gérer les éventuels conflits entre usagers ou les réparations nécessaires.

Malgré la structure hiérarchique décrite ci-dessus, ces organismes ont encore du mal à effectuer la gestion du réseau au jour le jour. Ils utilisent encore les outils les plus basiques : un crayon et du papier. Cette situation rend l'accès aux informations antérieures difficile voire impossible. Il est également difficile de transmettre ces informations à la hiérarchie supérieure.

L'accès aux fontaines public est accordé après le règlement d'un paiement annuel. Les consommateurs se voient attribuer une carte spécifiant la somme, le lieu et la date. Cette carte est utilisée comme preuve de paiement pour accéder aux fontaines. Cependant, il n'est pas possible pour le responsable fontaine de se souvenir exactement qui a payé ou non et les informations se perdent. De plus, certains abus (distribution non-équitable, prise d'eau sans paiement) ont été constatés.

2.3 Comparaison avec d'autres pays

La situation de l'eau dans d'autres pays, en particulier ceux aux économies fortes et établies, est différente. Le contexte d'Haïti est évidemment particulier, comme nous avons pu le voir dans les sections précédents. Ces circonstances nécessitent des solutions adaptées pour gérer la distribution d'eau. Néanmoins, une analyse de la situation d'autres pays a été utile afin de découvrir d'autres manières de faire, identifier les bonnes pratiques du milieu ainsi que les différences avec la situation haïtienne afin de bénéficier de l'expérience des organismes existants.

Gestion de l'eau en Belgique

Nous avons d'abord étudié la situation de l'eau en Belgique, choix naturel de par la localisation de nos études et du développement de ce mémoire. La première chose importante est que l'eau est une compétence régionale en Belgique, ce qui signifie que chaque région (Flandre, Wallonie et Bruxelles) développe sa propre politique à propos de l'eau.

Par exemple, en Wallonie, la gestion de l'eau est déléguée par le gouvernement wallon à la SPGE, Société Publique de la Gestion de l'Eau. Il s'agit d'une société publique, dont le fonctionnement est décrit par un contrat de gestion établi par le gouvernement. Elle travaille avec différentes intercommunales, sociétés également

publiques, pour s'assurer de la distribution d'eau potable, ainsi que de l'égouttage et l'assainissement des eaux usées. Ces différentes intercommunales sont mandatées par la SPGE via des contrats de gestions sur des parties spécifiques du territoire wallon.

On peut voir que contrairement à Haïti, la gestion de l'eau en Belgique est beaucoup plus liée au secteur public. Ceci n'est pas possible en Haïti, étant donné les problèmes politiques discutés plus tôt dans ce chapitre.

Visite d'un centre opérationnel en France

Afin d'avoir une meilleure vision de ce qu'implique la gestion de l'eau sur le terrain, ainsi que le quotidien d'une société impliquée dans cette gestion, nous sommes allés visiter le groupe Saur², une entreprise de gestion (assainissement, distribution, facturation, etc) de l'eau pour les particuliers et industriels, principalement en France.

Nous avons visité le Centre de Pilotage Opérationnel de Serris. Il s'agit du centre névralgique d'où sont dirigées les activités de Saur pour la région Île-de-France. Ce centre permet un suivi en temps réel des activités sur toute la région par des ordonnanceurs, employés se chargeant de la planification et du suivi du travail des opérateurs de terrain.

Les activités de ce centre sont principalement orientées autour de l'analyse des données reçues à travers tous les capteurs du réseau. Ces analyses sont étudiées par les ordonnanceurs qui déterminent les priorités du système et planifient les journées des opérateurs en conséquence.

Tout le centre fonctionne de manière informatisée. Le système informatique se base sur EPANET (voir section suivante) et propose un véritable ERP³ intégrant toutes les activités du centre, afin d'en augmenter la productivité. Dans ce logiciel, l'eau n'est au final plus qu'une ressource de l'entreprise et l'important est la gestion au quotidien de celle-ci.

Ce système informatique ne détermine pas le réseau de distribution. Celui-ci s'est développé historiquement et le système informatique s'est greffé par dessus. Il permet une collecte des données de ce réseau et de déterminer les opérations à effectuer en fonction de ces données.

2. <https://www.saur.com/>

3. Enterprise Resource Planning, progiciel de gestion intégrée

2.4 Comparaison avec des outils existants

Afin d'avoir une meilleure idée de ce à quoi devrait ressembler un logiciel de gestion de distribution d'eau, nous avons recherché des logiciels existants afin d'identifier leurs similarités, forces et faiblesses.

EPANET est un logiciel libre de droits, développé par l'*Environment Protection Agency* aux États-Unis. Ce logiciel est une véritable référence dans le milieu, et se retrouve souvent utilisé comme base d'autres logiciels. **WaterCAD** s'inspire de suites logicielles connues afin d'apporter une interface moderne et des visuels avancés. **KY Pipe** se base sur EPANET pour proposer une analyse hydraulique avancée et pour permettre de modéliser de nombreux contenus. Beaucoup d'autres logiciels existent, généralement basés sur EPANET ou AutoCAD (un logiciel plus générique).

On peut remarquer que tous ces logiciels proposent une modélisation géographique du réseau de distribution. Celle-ci permet un affichage du réseau sous forme de graphe avec de liens pour les conduites et des nœuds pour les différents points de traitement, ainsi qu'une superposition sur une carte géographique pour une meilleure visualisation. Un autre module commun à tous ces logiciels permet d'effectuer des analyses formelles sur le réseau. Les modèles mathématiques sont utilisés pour poser une évaluation (quantitative, relative, etc) du système. Ces analyses permettent d'étudier les flux, les dépôts, la pression, etc, que ce soit en un point ou sur toute une partie du réseau. Un dernier module commun permet d'obtenir des rapports sur l'état en temps réel du réseau de distribution.

La majeure différence entre ces solutions est le modèle tarifaire. On passe d'un logiciel gratuit à d'autres dont le prix peut s'élever jusqu'à 14.000 \$. Les autres variations sont plus d'ordre visuel, avec des interfaces plus ou moins stylisées ainsi que des représentations du réseau et des données plus ou moins modernes.

Chapitre 3

Organisation

Tout au long de l'année académique, la réalisation du mémoire a été influencée par la composition particulière du groupe. Trois mémorants, deux promoteurs, un client, Protos, et des acteurs de terrain. Autant de points de vue, horaires, désirs et apports à intégrer dans ce travail. Afin de conserver un cap commun et de pouvoir concilier les objectifs de chacun, il a été nécessaire de définir, planifier et attribuer des tâches.

3.1 Approche de travail

Comprendre les problématiques liées au mémoire a été notre première tâche. En tant qu'étudiants en sciences informatiques, notre formation n'inclut pas de cours sur la gestion de l'eau et le contexte haïtien. L'idéal aurait été d'intégrer dans l'équipe des consultants sur les domaines méconnus et d'envisager une visite en Haïti pour rencontrer les acteurs locaux et appréhender la situation.

Pour des raisons évidentes de budget et de temps, l'option de se rendre sur place et engager plus de partenaires a immédiatement été écartée. Pour comprendre la situation et relever les défis posés par le contexte, nous avons pu compter sur l'expertise de nos promoteurs et les 700 pages de documentation qui nous ont été fournies en préambule. Cette première documentation, orientée sur le contexte, a apporté des éclaircissements sur les défis de la gestion de l'eau potable en Haïti (ci-avant présentés en section 2.1).

Tandis que les besoins non-fonctionnels de l'application étaient établis par ce contexte et les vues de Protos, le contenu fonctionnel s'est basé sur les outils actuellement en place. Nous avons vu en section 2.2 que des solutions locales ont été mises en place. En essayant de tirer le meilleur de toutes ces solutions, nous avons établi un cahier des charges (voir chapitre 4).

L'arrivée en Belgique de plusieurs acteurs haïtiens impliqués dans la gestion de l'eau potable a modifié notre approche. A partir de janvier 2019, des réunions régulières avec des utilisateurs potentiels de l'application ont été mises en place. Ce changement d'approche a permis d'orienter le développement vers un but bien plus précis qu'auparavant, nous permettant de bénéficier de nouvelles expertises et nouveaux points de vue.

Planification

L'organisation d'un groupe, même réduit à trois personnes, est nécessaire pour orienter le travail dans un but commun. La planification de la réalisation du mémoire, qui est constitué en vaste majorité de l'application web produite, s'est effectuée sur trois plans :

Mensuel Au départ du mémoire, un plan général a été dressé. Au début, les inconnues (e.g., étendue du projet, fonctionnalités à implémenter) ont empêché de définir avec précision les productions et les dates butoires. Avec l'avancée du mémoire, nous avons pu estimer l'état et notre vitesse de travail. Les réunions avec Protos pour présenter nos deux plus grandes échéances, à savoir les versions alpha et beta de l'application web, ont été l'occasion de refaire le point et reformuler les objectifs.

Hebdomadaire Chaque semaine en alternance, nous rencontrions un de nos promoteurs. Dans la continuité de la planification mensuelle, ces réunions permettaient de présenter les productions, rapporter les tâches en cours et leur avancement, et enfin de projeter le futur du travail. Ces présentations étaient également l'occasion de rassembler les productions et d'avoir, chaque semaine, des documents ou démonstrations pouvant être comprises par des acteurs étrangers à l'équipe de développement. C'était donc autant un médium d'information que l'occasion de nous recentrer sur le travail, sans compter les précieux retours de nos promoteurs.

Quotidien Après une période d'expérimentation avec plusieurs outils de suivi, la gestion au quotidien des tâches à accomplir s'est ancrée sur Trello¹, un tableau en ligne composé de cartes amovibles réparties dans des colonnes. Dans notre organisation, une carte représente une tâche plus ou moins atomique semblable à un *récit utilisateur*². Nous utilisons les colonnes pour classer les cartes selon l'état de la tâche associée. Nous avons ainsi défini les colonnes : *à faire*, *à faire urgemment*, *en cours*, *problématique* et *fini*. Nous avons également utilisé des messageries instantanées (e.g. Slack³) et canaux vocaux pour organiser les tâches.

Etablir un plan et s'y tenir n'était cependant pas suffisant. Le projet a connu de nombreuses modifications dans ses objectifs et contraintes. Différents facteurs viennent expliquer ces modifications :

- La présentation de prototypes aux partenaires et aux promoteurs ont souvent amené de nouvelles idées pour l'application, ou suggéré la modification de fonctionnalités existantes.
- L'arrivée de nouveaux acteurs, principalement aux mois de février et mars. Plusieurs haïtiens ont pu apporter des éclaircissements sur la situation qui ont modifié l'application et l'orientation de son développement.
- Certaines fonctionnalités se sont révélées plus complexes à implémenter que prévu.

La figure 3.1 présente une vue simplifiée du diagramme de Gantt final sur lequel les modules implémentés (voir section 4.3) sont placés sur une ligne temporelle en dessous de laquelle les grandes phases du mémoire ont été placées, ainsi que les deux versions majeures, *alpha* et *beta*, présentées à tous les acteurs du projet.

Les phases d'*analyse* et *architecture* se sont également clôturées par des réunions de taille plus importante afin qu'un maximum de personnes puissent donner leur avis sur les artefacts (cahier des charges, ébauches d'interface graphique, etc). C'est notamment après ces réunions que les changements de planification étaient décidés pour répondre aux nouveaux besoins du client.

Le diagramme en figure 3.1 est donc la version finale. Comparé au plan initial, on constate une augmentation de la phase d'implémentation en raison de l'augmentation de la charge de travail, les modules *historique* et *finances* étaient en effet

1. <https://trello.com>

2. Un récit utilisateur est une phrase ou texte court décrivant une fonctionnalité à développer

3. <https://slack.com/intl/fr-be/>

imprévus. De nombreuses autres fonctionnalités ajoutées à posteriori ont allongé cette phase. Il était également prévu de passer plus de temps sur l'analyse et la conception du système. Or le cahier des charges et les ébauches n'étaient pas assez concrets pour avoir des retours précis sur l'outil. Devant ce fait et l'ampleur du projet, nous avons démarré l'implémentation plus tôt que prévu, et elle s'est également allongée sur le mois d'avril (pour un total d'environ 6 semaines supplémentaires).

Ces imprévus n'ont heureusement pas eu d'impact sur la production. Le démarrage anticipé de l'analyse en août, l'ajout d'une semaine de travail imprévue durant la session d'examen de janvier ainsi qu'une zone tampon de deux semaines prévue dans l'horaire initial ont permis d'accomplir la totalité des tâches initialement prévues, ainsi qu'une part appréciable des suggestions apportées au cours de l'année.

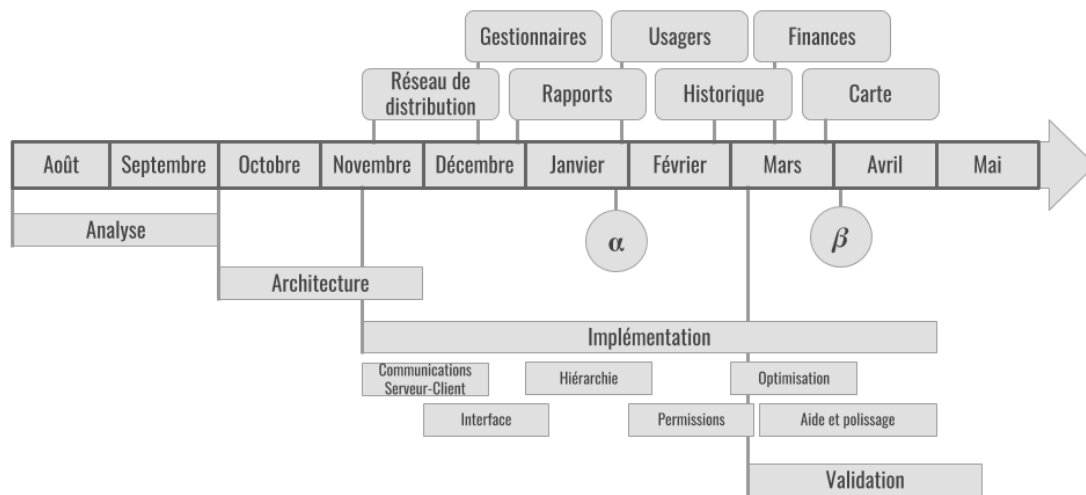


FIGURE 3.1 – Représentation simplifiée du diagramme de Gantt final

3.2 Méthodologie

Garantir l'avancement sans variation indésirable de tout projet se fait par le biais d'une méthodologie. Il est important que cette méthode de travail soit connue

de chacun pour garantir un rôle fixe et un suivi efficace. Cette notion de méthode est une caractéristique bien connue de la gestion de projet, le but étant de partir d'un besoin et d'apporter une solution informatisée. La méthode lie ces deux états et définit un système de communication, échéances, tâches, etc. De nombreuses méthodes déjà établies existent et sont prêtes à l'emploi par n'importe quelle équipe.

Agile

Dans la section précédente, nous avons évoqué les changements constants dans la planification et les tâches. C'est donc tout naturellement que nous avons opté pour une *méthode agile*, en opposition aux méthodes traditionnelles dites "waterfall" qui se montrent plus rigides et réfractaires au changement.

Nous n'avons pas suivi une méthode agile fixe, préférant une version plus relaxée, adaptée aux variations de disponibilité inhérentes à notre statut d'étudiant. On peut cependant relever des grands principes inspirés de méthodes existantes.

Développement dirigé par les fonctionnalités *Feature Driven Development*

en anglais, ce type de développement focalise les ressources sur la création. Tout d'abord création d'un prototype général et d'une liste de fonctionnalités. Ensuite création de celles-ci, ajoutées à l'application en impliquant le client à chaque fois qu'une fonctionnalité est créée. C'est également le client qui définit la priorité des fonctionnalités. Ce dernier point fut très utile dans notre mémoire où les acteurs haïtiens ont souhaité orienter le développement vers l'aspect financier de l'application, en opposition à Protos qui accentuait principalement le réseau de distribution.

Itérations courtes Une itération est une étape indépendante du développement. Chaque itération a sa propre planification et rétrospective. En méthode agile, les itérations sont courtes. Puisque nous voyions nos promoteurs toutes les semaines en alternance, nos itérations se sont naturellement calquées sur ce rythme. Nous avons évoqué l'intérêt de recentrer le développement en section 3.1. C'est également une excellente manière de détecter rapidement les risques (tant au niveau de la planification que de l'application produite) et d'agir en conséquence.

Livraison continue A partir de février, un acteur haïtien arrivé en Belgique a utilisé intensivement l'application. Grâce à ses retours, nous avons pu modifier l'application. Durant trois mois, nous avons déployé de nouvelles

versions de l'application toutes les deux ou trois itérations afin qu'il puisse essayer les fonctionnalités ajoutées. Cela a nécessité un peu plus de travail, pour déployer l'application et garantir l'intégrité des données entre les différentes versions. Effort récompensé par des retours fréquents de la part d'un utilisateur impliqué dont le point de vue extérieur s'est avéré extrêmement utile.

Phases du mémoire

Si le développement (implémentation) de l'application a suivi ces principes agiles, on notera que le déroulement d'ensemble du mémoire est divisé en grandes phases. Ce principe est mis en avant dans la figure 3.1. Cette caractéristique est traditionnellement indicatrice d'un développement en cascade où les différentes étapes du développement logiciel sont délimitées en temps et responsabilités (analyse, conception, implémentation, test, déploiement).

Les raisons sont multiples. On citera notamment notre méconnaissance des contextes de la gestion de l'eau potable et d'Haïti pour lesquels il a fallu se documenter pendant une période assez longue ainsi que la quantité de documents initiaux qui ont forcé à travailler durant ces premiers mois sur la compréhension et l'ébauche d'une solution.

Les disponibilités des acteurs externes au développement (membres de Protos, Haïtiens, chercheurs externes) ont aussi été un facteur. Chacun ayant une mission déterminée, nous ne pouvions avoir de contacts trop fréquents (aussi dûs à la distance et au décalage rendant difficile l'organisation de réunions). Nous avons donc dû aussi adapter les itérations aux disponibilités, variables, de chacun. Ceci explique aussi partiellement pourquoi le développement s'est autant accéléré lorsque des acteurs haïtiens sont venus en Belgique et nous ont offert des retours fréquents, ont participé à différentes réunions. La disponibilité du client est très importante dans le développement agile afin de pouvoir évaluer le travail accompli. En l'absence de membres extérieurs, ce sont nos promoteurs qui ont parfois joué ce rôle de client.

3.3 Répartition des tâches

Dès le début, nous avons cherché à tirer avantage de nos centres d'intérêts et capacités différents. Nous avons eu la chance d'être une équipe complémentaire et d'avoir un mémoire intéressant au point que toutes les tâches étaient attribuées sans problème et rapidement. Dans cette section, nous discutons de cette attribution systématique et de ce qu'elle implique pour ce travail.

Analyse

La phase d'analyse principale, d'août à octobre, avait pour but de comprendre la situation et le rôle de l'application. Afin que toute l'équipe avance vers le même objectif, il a été nécessaire que chacun soit impliqué dans l'analyse. En débutant par une recherche libre, nous avons cherché à nous documenter sur le contexte haïtien, les solutions existantes et les entreprises qui appliquent ces solutions. L'apport de plusieurs centaines de pages de documentation par Protos et nos promoteurs a nécessité une certaine organisation. Demander à chacun de nous, mémorants, de tout lire nous a semblé improductif, mais il était nécessaire que les informations circulent. Nous avons donc adopté un système de file pour les documents. Chaque nouveau document est placé dans cette file et pris en charge par un de nous trois (selon disponibilité et affinité). Au fil de la lecture, un document résumant le contenu et relevant les informations utiles pour le mémoire est produit. Le résumé terminé et mis à disposition, il est lu et discuté en groupe. Un autre objectif de ce système est de pouvoir bénéficier de documents raccourcis pour rechercher des informations, notamment dans la rédaction de ce travail.

Précisons cependant que même si ce que nous considérons être comme la phase d'analyse principale était de août à octobre, les changements d'exigences et l'apport de nouvelles informations au cours de l'année ont induit des phases d'analyse complémentaires, principalement au démarrage l'implémentation de chaque module. A nouveau, nous nous sommes tous trois impliqués dans ces phases afin que le développement aille dans un sens unique.

Conception et implémentation

Contrairement à l'analyse, la conception et l'implémentation ont connu une très forte séparation des tâches entre nous trois, chacun ayant un champ de travail défini.

Adrien responsable de la gestion du projet (planification, présentations intermédiaires, communication), de l'analyse fonctionnelle et du *front-end* (le client) : interface graphique, fonctionnalités utilisateur.

Céline responsable du *back-end* (le serveur) : base de données, traitement des informations et calculs, gestion des performances.

Sébastien responsable du *service worker* (l'accès hors-ligne à l'application), des tests et renfort au *back-end* sur le module financier et la vérification des permissions.

Pour chacune de ces préoccupations, le responsable désigné s'est chargé de la conception, des schémas, de la réalisation (implémentation) et de la documentation. Une communication constante et présentation de l'avancement des tâches à chaque membre de l'équipe se sont implicitement imposées, toujours dans l'optique d'orienter le développement dans le même sens.

L'inconvénient principal de ce type d'organisation et la compartementalisation qui nuit à l'apprentissage individuel puisque tout le monde ne travaille pas avec la totalité des technologies utilisées. Cela pose un risque d'arrêt, même temporaire, du développement si un des membres se trouve dans l'incapacité d'assumer sa charge de travail. En entreprise, ce risque serait à corriger au plus vite en impliquant plus d'acteurs dans chaque branche du développement, ou en impliquant tous les acteurs dans la totalité du code (par exemple via un système de rotation).

Nous avons tout de même effectué ce choix pour la puissance de travail qu'il permet et les facilités de développement et d'organisation apportées. Chacun de nous a développé une connaissance accrue de la technologie utilisée dans sa partie, permettant une augmentation drastique de l'efficacité qui s'est particulièrement ressentie dans la seconde partie (second quadrimestre) du travail. Organisationnellement parlant, savoir directement à qui s'adresser en cas de problème ou demande de fonctionnalité a été également particulièrement pratique.

Rédaction

La rédaction du présent texte a à nouveau rassemblé l'équipe. En tant que document résumant et justifiant le travail accompli, il était naturel d'être tous inclus dans son élaboration. Nous avons considéré les différentes sections en tant qu'entités individuelles, permettant un déroulement de la rédaction simplifié, étape par étape. A des granularités différentes (parfois allant jusqu'à établir le plan pour chaque sous-section), nous avons dressé une liste du contenu à intégrer. La section était ensuite attribuée au mémorant responsable du sujet discuté.

Une fois le premier jet établi, une relecture permettait à chacun d'apporter les modifications désirées, parfois discutées, pour obtenir un texte dans lequel chacun de nous peut s'identifier. Ce processus était répété pour chaque section et chaque jet, encouragés et orientés par les commentaires de nos promoteurs.

Au quotidien

La planification générale avait une précision au maximum hebdomadaire. Pour le travail effectué au jour le jour, nous nous sommes entièrement reposé sur Trello avec un système de colonnes décrivant l'état et la priorité de chaque tâche comme exposé ci-avant. Autre héritage des méthodologies agiles, ces cartes sont des tâches à réaliser pour avancer le développement. Dans notre cas, les tâches étaient souvent atomiques ou ne reflétaient qu'une seule unité de développement (e.g., ajouter un champ dans un formulaire).

Un système de description de la tâche, assignation de membre(s) à une carte et étiquetage, ajouté au principe des différentes colonnes indiquant la priorité et l'état de chaque carte, a permis de gérer efficacement le travail.

Quant à l'ajout des cartes, on relève deux possibilités :

- Au début des itérations, lorsqu'on planifiait le sprint à venir.
- Durant une itération, lorsqu'un membre de l'équipe avait besoin d'une fonctionnalité qu'un autre membre devait implémenter, le premier membre ajoutait une carte en assignant le second. Par exemple, si le développement du client nécessitait l'implémentation d'une fonctionnalité du serveur, la carte était ajoutée en assignant directement la personne responsable.

La colonne de travail prioritaire s'est avérée une très bonne idée. Puisque nos fonctionnalités sont dépendantes à la fois du client et du serveur, cela permettait d'identifier clairement les besoins. Nous pouvions ainsi débiter une séance de développement à n'importe quel moment en sachant ce dont les autres avaient besoin.

Chapitre 4

Analyse des besoins

Total des pages : 6 à 10

4.1 Besoins fonctionnels

Pages : 1 à 2

Gestion des données

Simplification des procédures

4.2 Besoins non-fonctionnels

Pages : 1 à 2

Sécurité des données

Connexions lentes et peu fiables

4.3 Cahier des charges

Pages : 2 à 3

Complet en annexe ?

4.4 Structure des données

Pages : 2 à 3

Complet en annexe ?

Chapitre 5

Implémentation

Total des pages : 16 à 22

5.1 Choix technologiques

Pages : 3 à 4

Web

Établie dès le premier jour, l'exigence du support web pour ce travail a formé la totalité du mémoire. Nous n'avons pas eu de réel choix à ce niveau, il nous est néanmoins possible de trouver des avantages à ce type de développement :

- L'application ne nécessite aucune installation. Il n'y a aucun prérequis d'installation et la seule nécessité est de disposer d'un navigateur internet (préférentiellement majeur et relativement récent). C'est une facilité pour l'utilisateur, qui ne pourrait avoir plus de facilités à atteindre l'application puisqu'il suffit d'accéder au site internet. L'avantage est également étendu au développement qui ne doit pas se préoccuper du système d'exploitation de l'utilisateur.

- Le déploiement des mises à jour et la manière dont l'application fonctionne en général est uniforme à travers les utilisateurs et les conflits de version, bien que devant toujours être gérés notamment par le biais du cache (voir section ??), sont moindres que sur une application locale. D'éventuels conflits de données peuvent être réglés directement via le serveur et les outils de développement disponibles à l'équipe, ce qui est plus rapide et moins complexe que de créer des automatisations pour gérer ces mises à jour individuellement pour chaque utilisateur.

— Ne présentant pas que des avantages, le développement web apporte son lot de problématiques nouvelles :

- Utiliser une application en ligne nécessite une connexion internet, et même si cette dernière n'est pas requise en permanence (voir section 5.5), il est impératif qu'elle soit présente ponctuellement pour charger l'application et envoyer/recevoir les données lors de son utilisation.
- La diversité des navigateurs internet (Firefox, Chrome, Edge, etc) entraîne la nécessité de gérer ces environnements différents pour que l'application fonctionne sur un maximum de navigateurs et de manière similaire.

Choix du framework - Django

Un framework est une infrastructure logicielle dans laquelle l'application est développée. On peut argumenter, avec raison, qu'un framework ajoute une certaine quantité d'outils qui ont pour effet de complexifier l'application et d'en augmenter la taille. Dans une application aussi large que la nôtre cependant, l'apport bénéfique qu'ont ces outils supplante largement ces inconvénients. Parmi ces apports bénéfique on citera surtout :

- L'environnement de travail d'un framework impose certaines contraintes qui répandent des bonnes pratiques. Par exemple une structure des fichiers du code source qui permet à long terme une meilleure maintenance, surtout dans de larges applications.
- Les outils fournis sont testés, utilisés et éprouvés. Il n'est pas nécessaire de devoir créer nous-même des fonctionnalités déjà présentes dans un framework, et même si recréer ces fonctionnalités est une expérience enrichissante, elle reste moins efficace. De plus, les fonctionnalités existantes sont de meilleure qualité, offrent de meilleures performances et surtout ne sont pas à maintenir par l'équipe de développement de l'application, mais par l'organisme en charge de la publication du framework. Ces avantages seront d'autant plus importants que le framework est populaire et maintenu à jour.

- Un framework aura beaucoup de fonctionnalités par défaut qui rendront le développement plus aisé. Messages d'erreur, outils de mesures, etc.
- La documentation et l'aide sont de meilleure qualité. Les framework ont généralement une communauté et une équipe de développement dont les capacités et l'expérience permettent une production de documentation rendant très facile la compréhension du projet et la recherche d'aide. Avantages qu'il est impossible d'égaliser dans une équipe de trois personnes sur dix mois.

La table 5.1 reprend une liste non-exhaustive de frameworks qui ont été envisagés pour ce projet. Afin de pouvoir développer une application compatible, nous avons évalué les possibilités et arrêté un choix durant la phase d'analyse du développement. Ci-après sont exposées les motivations de ce choix.

Langage	Framework	C	M	Pe	Po	T	Total
JavaScript	Express	M	M	B	B	M	3.5
	MeteorJS	B	B	M	N	B	3.5
	NodeJS	M	M	M	B	N	3.5
Python	Django	M	M	B	B	M	3.5
	Flask	B	B	B	B	B	5
	Pyramid	B	B	B	N	B	4
Ruby	Rails	N	M	B	B	B	3

TABLE 5.1 – Comparatif de quelques frameworks connus

Critères - [B]on, [M]oyen, [N]égatif

C Complexité - Difficulté d'utilisation / apprentissage.

Évaluer la complexité est hautement subjective en fonction des préférences et habitudes du développeur. Nous pouvons néanmoins comparer les langages utilisés et la structure du framework.

Une mesure de la complexité se base sur l'imbrication des blocs de code. Un code comportant beaucoup de structures imbriquées sera moins lisible, et donc plus complexe. Les résultats¹ en figure 5.1 montrent que cette mesure favorise le langage Python dans nos options.

M Maintenabilité - Aisance et possibilités de maintenance.

L'aisance avec laquelle un code est maintenu à jour est inversement lié à la complexité. A cela, nous pouvons ajouter les fréquences de mise à jour des frameworks. La structure du framework impacte également la maintenabilité ; moins on modifie de fichiers pour implémenter une fonctionnalité, plus facile sera la maintenance à moyen et long terme.

1. Étude de Seerene ©, 2015

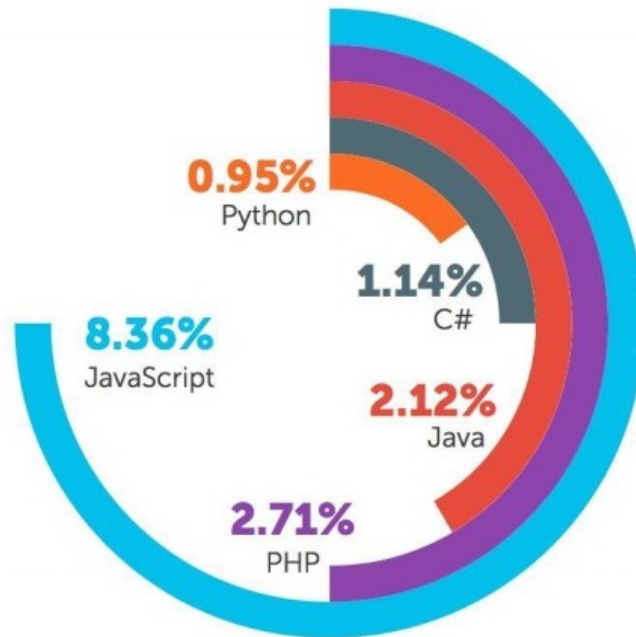


FIGURE 5.1 – Pourcentage de code imbriqué

Pe Performance - Capacités du framework.

Bien que moins importantes que dans une application critique (e.g. : aérospatial, banque), les performances sont mesurées en termes de temps de réponse pour une requête du client au serveur.

Po Popularité - Popularité du langage à travers le monde.

La popularité est sans doute la plus importante des mesures de ce comparatif. Un langage populaire dispose de plus de tutoriels et aide en ligne que d'obscurs langages / frameworks peu ou pas utilisés.

La figure 5.2² montre une évolution de la popularité des frameworks les plus courants.

T Taille - Espace de stockage requis

Un framework léger sera plus intéressant pour ce projet. D'une part pour la complexité, car moins il est nécessaire de modifier/ajouter des fichiers, plus l'utilisation est aisée. D'autre part car le serveur de destination nous est inconnu et nous ne savons donc pas quel espace de stockage sera mis à disposition.

2. <https://hotframeworks.com/>

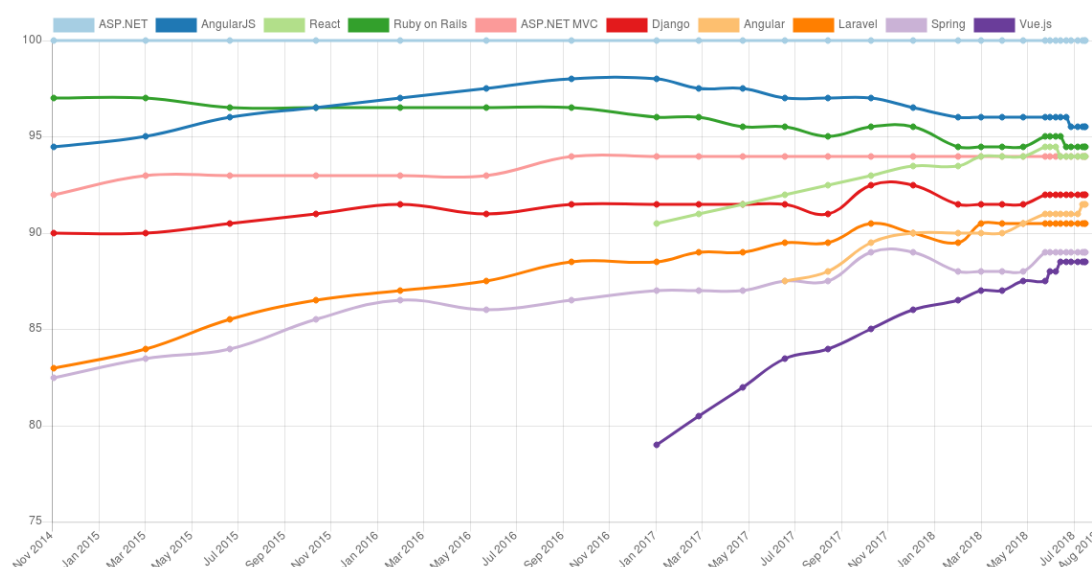


FIGURE 5.2 – Popularité des frameworks de novembre 2014 à août 2018

Django

Le choix d'un framework repose sur de nombreux critères dont plusieurs sont subjectifs. En l'absence d'une ligne directrice, le choix se fait souvent selon l'expérience et les désirs de l'équipe. Notre comparatif terminé, il est devenu clair que nous favorisons un framework utilisant le langage de programmation Python.

Flask a été envisagé, cependant les facilités offertes par Django dans la gestion de la base de données, particulièrement pour les données géographiques, a été le facteur décisif. Parmi les sites populaires utilisant Django, on citera Instagram, DropBox, Disqus³.

Python

Le langage Python est un langage de programmation populaire principalement grâce à sa syntaxe légère permettant un code compréhensible et généralement plus court (voir figure 5.1). Il n'est pas le plus utilisé, ce sont PHP (79% d'utilisation),

3. <https://djangostars.com/blog/10-popular-sites-made-on-django/>

ASP.NET (11.3%) et Java (4%) qui constituent le trio de tête [8]. Toutefois, nous pensons que Python s'adapte parfaitement aux besoins de l'application demandée. C'est un langage relativement simple et compréhensible dont la popularité croît ces dernières années. Les bibliothèques logicielles et la documentation, toutes deux disponibles à foison sur internet, en font un langage particulièrement ouvert aux débutants. Un dernier argument, et non des moindres, est l'expérience déjà acquise dans la programmation en Python durant nos années d'études.

PostgreSQL et PostGIS

Si l'utilisation du langage Python était imposée dès lors que Django fut choisi en tant qu'environnement logiciel, le système de gestion de base de données, dont le rôle est de lier l'application web et la base de données, a pu être choisi un peu plus librement.

Nous avons tout de suite écarté les systèmes non-relationnels. Bien que présentant des avantages, principalement dans l'agilité avec laquelle la structure des données peut évoluer, ces derniers ne sont pas adaptés à une application comme celle développée dans le cadre de ce mémoire. En effet les données ont de fortes dépendances, les éléments du réseau étant rattachés à des gestionnaires faisant parties de zones géographiques elle-même gérées, etc. Les outils offerts par les bases de données relationnelles pour joindre différents groupes de données (par exemple pour croiser les consommateurs avec leurs paiements afin d'obtenir un détail financier), sont tout simplement mieux adaptés au fonctionnement de l'application.

Restent donc les systèmes de gestion de base de données (SGBD) relationnels, dans lesquels nous avons opté pour PostgreSQL, classé quatrième système de gestion de bases de données le plus utilisé au monde⁴. La première motivation de ce choix est la recommandation du framework Django, lequel s'adapte mieux à ce SGBD via son système de *mapping objet-relationnel* (voire section 5.6). La seconde motivation est l'existence de l'extension PostGIS [2], une extension du SGBD permettant de traiter aisément les données géographiques.

4. <https://db-engines.com/en/ranking>, 1 mai 2019

Bootstrap

Bien que l'utilisation de frameworks *front-end* soit de plus en plus répandue, nous n'en avons pas intégré dans le client de l'application. Les choix sont pourtant nombreux et tant Angular, React que Vue ont été envisagés. Nous avons décidé de n'en utiliser aucun pour ne pas ajouter de technologie supplémentaire et pour limiter le nombre de dépendances du projet afin d'en garantir la perrenité sur le long terme.

Néanmoins, la version 3 (plus documentée, éprouvée et ayant de meilleures compatibilités avec les autres frameworks et bibliothèques que la version 4) de Bootstrap est présente dans l'application. Bootstrap est une bibliothèque implémentant une large quantité de fonctionnalités que nous utilisons majoritairement pour agencer visuellement les blocs de l'interface graphique en fonction de la taille de l'affichage disponible, afficher des composants avancés (multisélection, sélection de date, menus) et styliser les composants de l'interface.

Bootstrap ajoute l'avantage non-négligeable d'implémenter un bon nombre de fonctionnalités nous permettant d'accélérer le développement, tout en étant une bibliothèque assez connue que pour être extensivement testée et documentée, la rendant une meilleure alternative à une création personnalisée qui aurait sans doute été moins performante.

Le désavantage majeur est son poids. Accompagnée de nombreuses fonctionnalités qui n'ont pas été utilisées dans ce projet, Bootstrap est lourde (approximativement 400 KB), d'autant plus en considérant la lenteur des connexions haïtiennes, cependant il est tout à fait possible de nettoyer la bibliothèque (voir section 7.1) pour n'en conserver que les fonctionnalités utilisées, et ainsi optimiser les téléchargements.

DataTables

DataTables [6] est la seconde plus grosse bibliothèque du projet, gérant l'affichage des données sous forme de tables. A nouveau sélectionnée de par son ancienneté permettant une documentation et des tutoriels efficaces, DataTables était très loin d'être la seule option. Nous avons envisagé, et débuté la première version avec, la bibliothèque *Django-Tables2* qui causait un ralentissement du chargement des pages contenant les données, et une perte de contrôle de l'information au niveau

du serveur car les tables étaient directement rattachées aux modèles de la base de données.

Dans notre volonté d'offrir une API pour communiquer avec le serveur (voir section 5.6), nous avons migré vers DataTables. En utilisant la technologie AJAX (Asynchronous Javascript And XML) de DataTables, nous pouvons exploiter pleinement l'asynchronicité du navigateur et charger une page de l'application en totale indépendance des données ce qui augmente la vitesse de chargement des contrôles, permettant d'utiliser directement une fonctionnalité de l'application sans attendre le téléchargement des données (par exemple pour ajouter un élément).

Les capacités offertes par DataTables sur le contrôle des données nous permettent également de gérer les données à l'envoi et/ou à la réception et de contrôler leur affichage, tout en bénéficiant des fonctionnalités implémentées par la bibliothèque (champ de recherche, tri dynamique, pagination).

Outre son poids conséquent auquel nous pouvons à nouveau opposer la solution évoquée dans la bibliothèque Bootstrap, DataTables requiert d'implémenter les fonctionnalités de l'API et de gérer le transfert et traitement des requêtes en envoi et réponse du client et du serveur, ce qui représente une charge de travail non-négligeable.

Chart.JS

La bibliothèque Chart.JS est utilisée pour générer les représentations visuelles de l'application. Les bibliothèques permettant de créer des graphiques sont nombreuses et Chart.JS a de nombreux concurrents qui offrent parfois plus de fonctionnalités. Chart.JS a une bonne réputation dans ce domaine et est réputée pour ne pas être trop complexe à utiliser, toujours dans le but de respecter les exigences technologiques. De plus, Chart.JS génère des canvas là où un de ses concurrents majeurs, la bibliothèque D3.JS, génère des images vectorielles ou des éléments du domaine, plus gourmands en performances en particulier avec de larges ensembles de données.

5.2 La hiérarchie dans l'application

Le bon fonctionnement de l'application repose sur le principe hiérarchique de la récolte des informations venant de la saisie de données par les utilisateurs de l'application puisque le manque de financement et l'état actuel du réseau ne permet pas une récolte en temps réel des données via des capteurs. Une représentation de la structure hiérarchique vous est présentée en figure 5.3.

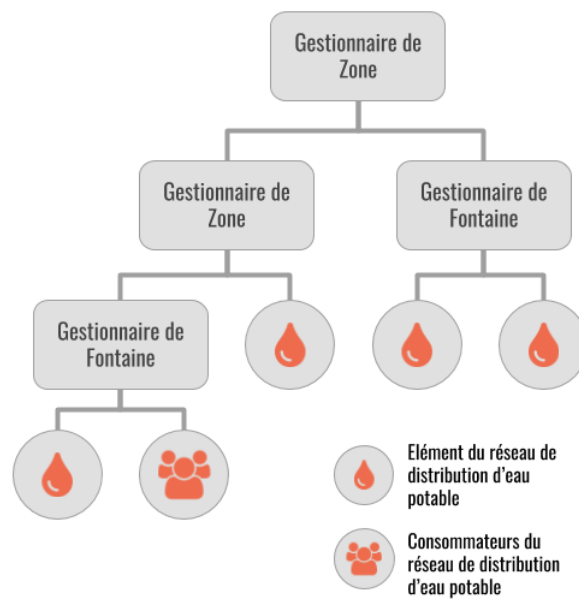


FIGURE 5.3 – Représentation du principe hiérarchique des utilisateurs, éléments et consommateurs

Structure

Rappelons que l'application a trois concepts clefs :

Zone

Une zone est un partitionnement des données de l'application. A la base de l'application se situe la zone racine dans laquelle toutes les données sont enregistrées. Une zone peut être composée d'autres zones. Une zone parente rassemble la totalité des données des zones filles. En revanche une zone fille ne rassemble que les données qui lui sont rattachées et ne voit pas le contenu des zones soeurs. On peut ainsi se servir des zones pour

diviser géographiquement le réseau de distribution d'eau selon le principe de la hiérarchie haïtienne existante (DINEPA, OREPA, CAEPA) exposée en section 2.2. Chacun a donc ses propres responsabilités et ne vient pas empiéter sur le territoire de l'autre.

Elément du réseau

Un élément du réseau peut être une conduite, un réservoir, une source, une fontaine, un kiosque ou une prise individuelle. Ces six éléments constituent le réseau de distribution d'eau potable haïtien et sont enregistrés dans une zone.

Consommateur

Un consommateur est un habitant qui utilise le réseau de distribution d'eau potable. Dès son enregistrement dans l'application, le consommateur est rattaché à un élément de sortie (fontaine, kiosque ou prise individuelle) du réseau, ce qui permet de gérer sa facturation et, de manière plus générale, de consulter la densité des consommateurs.

Deux types d'utilisateurs sont amenés à se connecter à l'application :

Gestionnaire de Fontaine

Ce gestionnaire est un noeud terminal dans un chemin de la hiérarchie des utilisateurs. Son rôle est simple ; au plus proche du consommateur, il est chargé de gérer (et donc assigné à) un ou plusieurs éléments de sortie du réseau, c'est-à-dire qu'il s'occupe du bon fonctionnement de la distribution aux consommateurs. Il a quatre charges relatives aux éléments du réseau de distribution d'eau qui lui sont assignés :

1. Rapporter l'utilisation, sous forme des volumes distribués.
2. Signaler les problèmes avec la pompe ou avec la qualité de l'eau.
3. Ajouter, modifier ou supprimer les consommateurs.
4. Rapporter les paiements des consommateurs.

Le gestionnaire de fontaine est donc une entité locale similaires aux *comités fontaines* vus en section 2.2.

Gestionnaire de Zone

Le gestionnaire de zone est un rôle plus complexe. Il est un administrateur local ou global, dépendamment de sa position dans la hiérarchie. Un gestionnaire de zone peut superviser d'autres gestionnaires, de zone ou de fontaine, et ce en nombre illimité. Un gestionnaire de zone peut effectuer les mêmes actions qu'un gestionnaire de fontaine, mais ses charges sont plutôt orientées dans l'administration de sa zone :

1. Ajouter, modifier ou supprimer des zones.
2. Ajouter, modifier ou supprimer des gestionnaires.

3. Assigner des gestionnaires de zone à une zone, et des gestionnaires de fontaine à un ou plusieurs éléments de sortie du réseau.
4. Gérer ses gestionnaires et leurs actions via l'historique.
5. Utiliser les données récoltées par ses gestionnaires pour gérer le réseau de distribution.

Permissions

Bien entendu, une hiérarchie implique un système de permissions. Dans notre application, ce dernier se calque exactement sur la structure exposée ci-avant. En toute normalité, les utilisateurs ne peuvent effectuer que les actions qui sont assignées à leur rôle. L'interface propose des contrôles réduits aux gestionnaires de fontaine, en opposition aux gestionnaires de zone qui doivent assumer leurs responsabilités supplémentaires.

De plus, les données sont fortement liées à la hiérarchie et empêchent une quelconque utilisation, ajout, modification ou suppression par un utilisateur non-autorisé. Lorsqu'une donnée, peu importe son type (une zone, un consommateur, un rapport mensuel, etc), est ajoutée dans l'application, elle n'est visible que par l'utilisateur qui l'a créée et les utilisateurs qui lui sont supérieurs dans la hiérarchie (utilisateurs parents).

En somme, plus un utilisateur est haut dans la hiérarchie, plus il a de pouvoir sur les données de l'application, la zone racine étant donc la somme de toutes les données, faisant des gestionnaires de cette zone les utilisateurs ayant le plus d'impact sur le système.

5.3 Interface utilisateur

Partie visible de notre application, l'interface graphique permet d'utiliser les fonctionnalités d'HaïtiWater. En réalité, l'interface graphique n'est qu'un moyen de faciliter l'utilisation de l'application et de rendre les données accessibles et lisibles par un humain. L'application est entièrement contrôlée par l'API qui requiert les données à afficher ou modifier. Somme toute, n'importe quelle autre interface (application locale, mobile, etc) pourrait envoyer les mêmes requêtes. Cela permet

donc de décentraliser la logique de l'application dans le serveur tandis que l'interface permet de présenter l'information à un humain (tableaux, graphiques), le guider dans son utilisation et faciliter la saisie de données via les formulaires. Non content de permettre, dans le futur, la création de nouvelles interfaces, cette séparation des rôles améliorer le développement et la maintenance de l'application, nous y reviendrons en section 5.6.

Se voulant simple, l'interface est composée de blocs de contrôle permettant de gérer les données dans une zone de travail commune à toute les pages permettant de naviguer à travers l'application.

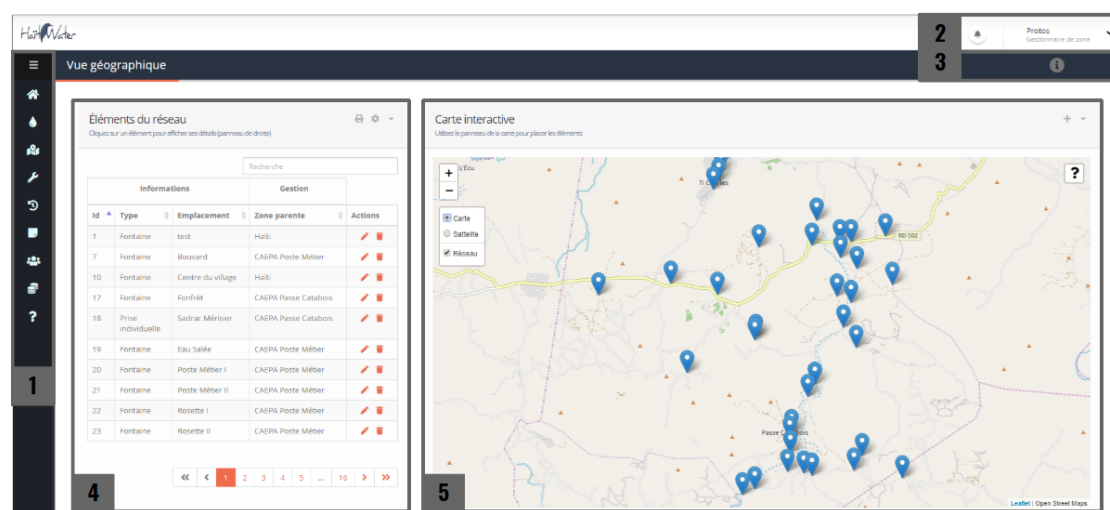


FIGURE 5.4 – Capture d'écran du module géographique d'HaitiWater

Une capture d'écran de l'application vous est présentée en figure 5.4. Les contrôles communs incluent :

1. Le menu déroulant permettant d'accéder aux modules de l'application.
2. La zone utilisateur qui affiche ses notifications et le nom de compte qui lie à la déconnexion ou modification des informations personnelles dans son menu déroulant.
3. L'aide rapide, permettant de déclencher une visite virtuelle de la page ouverte, visite textuelle qui présente brièvement les contrôles et leur utilité.

Les contrôles spécifiques à cette page sont :

4. La table est un bloc de contrôle récurrent dans notre application. Ici listant les différents éléments du réseau de distribution, chaque table possède un champ de recherche, un système de tri et des contrôles permettant d'ajouter, supprimer et modifier les éléments contenus. Remarquez que pour augmenter la lisibilité, les tables disposent d'une pagination permettant de limiter l'affichage à 10 (valeur par défaut), 25, 50 éléments ou de tout afficher.
5. Un bloc de contrôle plus complexe, ici la carte interactive. Sur celle-ci, il est possible de placer des marqueurs (fontaines, kiosques, prises individuelles, sources et réservoirs) ou des lignes polygonales (conduites). Supportant les habituels contrôles de zoom (molette ou boutons) et déplacement (en cliquer/glisser), il est également possible de basculer entre une vue cartographique d'OpenStreetMap⁵ et une vue satellite de Google⁶

Référence en annexe ?

5.4 Procédure d'utilisation

Pages : 2 à 3

Référence en annexe ?

5.5 Client

Mise en cache

Pages : 4 à 5

5. <https://www.openstreetmap.org/>

6. <https://www.google.com/maps>

Modularité et responsiveness

Gabarits

Accessibilité hors-ligne

...

5.6 Serveur

Pages : 4 à 5

Authentification

Requêtes

...

Chapitre 6

Validation

Total des pages : 8 à 11

6.1 Performances

Pages : 3 à 4

Temps

Poids

6.2 Vérifications automatiques

Pages : 2 à 3

Tests unitaires

Tests fonctionnels

6.3 Vérifications utilisateurs réels

Pages : 3 à 4

Méthodologie

Résultats obtenus

Modifications apportées

Chapitre 7

Améliorations futures

Total des pages : 4 à 7

7.1 Suite du projet

Pages : 2 à 3

7.2 Défis rencontrés

Pages : 1 à 2

7.3 Propositions

Pages : 1 à 2

Chapitre 8

Conclusion

Pages : 1 à 2

Bibliographie

Bibliographie

- [1] Mission de la dinepa. Site internet du gouvernement haïtien.
- [2] PostGIS Steering Committee. Postgis.
- [3] Andrew Powell et Oscar Becerra Eduardo A. Cavallo. Estimating the direct economic damage of the earthquake in haiti. *IDB Working Paper Series*, (IDB-WP-163), February 2010.
- [4] Greeet Schaumans et Katrien Van Hooydonk. Analyse commune de contexte haïti, Octobre 2015.
- [5] Evens Emmanuel et Per Lindskog. Regards sur la situation des ressources en eau de la république d'haïti, Octobre 2015.
- [6] SpryMedia Ltd. Datatables.
- [7] Comission Économique pour l'Amérique Latine et les Caraïbes. Impact socioéconomique de la dégradation des terres en haïti et interventions pour la réhabilitation du milieu cultivé, Janvier 2008.
- [8] Web Technology Surveys. Historical yearly trends in the usage of server-side programming languages for websites, 2019. Consulté le 30 avril 2019.
- [9] Vereda Williams. A case study of the desertification of haiti. *Journal of Sustainable Development*, 4, Juin 2011.

Pages : 2 à 3

Annexe A

Cahier des charges complet

Pages : beaucoup

Annexe B

Base de données

Pages : beaucoup

Annexe C

Wireframes

Pages : beaucoup

Annexe D

Diagrammes d'activité

Pages : beaucoup

Annexe E

Documents de validation

Pages : 10 à 20

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain

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