# Cahier des Charges

Sonny Klotz - Jean-Didier Pailleux - Malek Zemni

*Interface de chargement, de contrôle*
*et d'analyse statistique des données*
*pour la constitution d'un graphe de flux*

20/03/2017

# Table des matières

# 1 Motivations du projet

## 1.1 But du projet

### 1.1.1 Contexte du projet

De nos jours, les masses de données collectées sont de plus en plus importantes. L'objectif principal de cette collecte de données est d'en extraire une valeur ajoutée. Or, ces données à l'état brut sont difficilement exploitables dû à leur volume et à leur complexité.
Notre produit correspond au travail indispensable d'analyse de ces données, afin de faciliter leur exploitation.

### 1.1.2 Objectif du projet

Ce projet a pour but de fournir aux utilisateurs une application qui se chargera d'une part de structurer les données, les analyser et les visualiser, et d'autre part de préparer ces données pour un chargement via des API.

## 1.2 Parties prenantes

### 1.2.1 Maître d'ouvrage

Notre interface de contrôle, de chargement, et d'analyse de données est développée pour l'entreprise **DCbrain**.
Le projet a été lancé en collaboration avec l'**UVSQ**.

### 1.2.2 Client

DCbrain est également l'entreprise qui va bénéficier des paquets finaux après leur développement.

### 1.2.3 Autre partie prenante

Les industriels clients de DCbrain sont des parties prenantes indirectes. Notre travail doit pouvoir être utilisé par DCbrain pour renforcer leur application.

## 1.3 Utilisateurs du produit

En premier lieu, l'application va servir aux membres de DCbrain étant donné que leurs clients industriels (Total, ERDF,...) leur fournissent les fichiers CSV, afin qu'ils puissent appliquer des analyses descriptives sur les données dans le but de repérer des anomalies sur les réseaux de ces derniers. Puis en second lieu, DCbrain va déployer l'application pour ses clients, dans ce cas les membres de ces organismes deviendront des utilisateurs.

# 2 Contraintes sur le Projet

## 2.1 Contraintes imposées

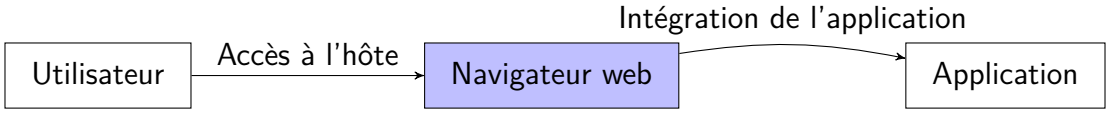
### 2.1.1 Contraintes sur la conception :

| Contrainte | Fiche |
|---|---|
| 1. Le produit doit fournir une application web | **Description :** notre produit sera une application fonctionnant sur un navigateur web, appelée ***applet***.<br>**Justification :** assure une très grande portabilité et fournit à l'utilisateur une interface interactive.<br>**Critère de satisfaction :** on peut lancer l'application sur un navigateur web. |

| | |
|---|---|
| 2. Le produit doit être développé avec un langage de programmation compatible avec l'analyse de données | **Description :** le langage de programmation choisi doit inclure des bibliothèques qui permettent d'analyser les données (taches de data mining et de machine Learning).<br>**Justification :** permettre une analyse de données efficace.<br>**Critère de satisfaction :** on peut effectuer une analyse descriptive de données. |
| 3. Le produit doit fournir une API d'analyse de données en sortie | **Description :** l'application doit intégrer des API d'analyse descriptive de données qui pourront être livrées en sortie au client.<br>**Justification :** permettre une réutilisabilité des fonctionnalités majeures du produit.<br>**Critère de satisfaction :** on peut exporter une API d'analyse de données en sortie. |

### 2.1.2 Environnement de fonctionnement :

Le produit va fournir une application web ou ***applet***. L'environnement technologique de ce genre d'application sont les navigateurs web. Ces navigateurs web jouent le rôle d'interface entre l'utilisateur et l'application.




Le produit doit donc être compatible avec tous les navigateurs web bureau (pas de version mobile exigée) prenant en charge les fonctionnalités des dernières versions ***HTML5*** et ***CSS3***, par exemple ***Google Chrome*** et ***Mozilla Firefox***.

### 2.1.3 Applications partenaires :

Le produit va fournir une API en sortie. Il doit donc prendre en compte de l'environnement d'intégration de cette API, c'est à dire que cette API doit être compatible avec les outils du client, l'entreprise DCbrain.

### 2.1.4 Temps dont disposent les développeurs du projet :

Le produit doit être rendu avant le 26/05/2017.

### 2.1.5 Budget du projet :

La réalisation du produit n'exige pas de ressources financières. Aucun budget n'est donc nécessaire.

## 2.2 Glossaire et conventions de dénomination

**Maître d'ouvrage :** entité porteuse du besoin, définissant l'objectif et les exigences du projet, attendant la réalisation d'un produit, appelé ouvrage.

**Maître d'œuvre :** entité retenue par le maître d'ouvrage pour réaliser l'ouvrage.

**Big data :** ensembles de très gros volumes de données traitées et exploitées pour en tirer des informations.

**Machine Learning :** méthodes automatisables offrant la possibilité à une machine d'évoluer grâce à un processus d'apprentissage à partir des données.

**Réseaux physiques :** réseaux industriels, de fluide, de distribution (par exemple réseau électrique)

**Capteurs IOT :** capteurs *Internet of Things* (internet des objets) déployés sur des réseaux physiques afin d'y collecter des données.

**Graphe de flux :** graphes représentant les données liées au flux du réseau.

**CSV :** *Comma-separated values*, format de fichier ouvert représentant des données tabulaires sous forme de valeurs séparées par des virgules.

**ADD :** *Analyse Descriptive de Données*, fonctionnalité d'analyse de description statistique des données.

**ADD unidimensionnelle :**

**ADD multidimensionnelle :**

**Applet :** application qui s'exécute dans la fenêtre d'un navigateur web.

**API :** *Application Programming Interface*, constituent les paquets utilisables par les développeurs (intégrés), qu'on va livrer au client en plus de l'application elle-même.

**Drag&Drop :** glisser et déposer un fichier dans une fenêtre.

**Structure 1 :**

**Structure 2 :**

# 3 Exigences fonctionnelles

## 3.1 Périmètre de l'ouvrage

Deux parties prenantes vont interagir avec notre système :

— Les industriels : ce sont les détenteurs des réseaux physiques et des données mesurées grâce aux capteurs IoT, clients de DCbrain. Ceux-ci n'ont pas de connaissances en informatique ou en statistique présupposées. Leur besoin va être de contrôler le réseau physique et simuler son évolution.

— DCbrain : notre client souhaite proposer un meilleur service aux industriels. Pour cela, nous offrons une applet simple ainsi que des API pour renforcer leur produit. Les API seront manipuler par des développeurs aptes à comprendre et réutiliser des paquets informatiques documentés.

## 3.2 Périmètre de l'œuvre

### 3.2.1 Diagramme de cas d'utilisation

Le diagramme ci-dessous définit le perimètre d'utilisation de notre travail :
Diag à insérer ici

### 3.2.2 Analyse descriptive de données

DCbrain et les industriels eux-mêmes lancent des ADD sur l'applet à partir d'un fichier CSV au bon format.

Sur une applet s'exécutant dans un navigateur, l'acteur importe son fichier s'il est au bon format - retenter sinon. Il peut ainsi consulter un échantillon du jeu de données brut et deux options sont maintenant disponibles.
L'acteur peut naviguer pour consulter soit une description préliminaire du fichier, soit les données étant mal typées et donc non analysables.
Le déroulement type se poursuit en attribuant un rôle aux colonnes du fichier puis à la sélection de l'une d'elles pour lancer son analyse si l'ensemble de données n'est pas vide.
L'acteur va enfin consulter les résultats de l'ADD et décider de lancer une autre analyse sur le même fichier, importer un autre fichier, ou sauvegarder les résultats en local.

### 3.2.3 Ecriture de script

Les développeurs de DCbrain utilisent les API pour automatiser des tâches ou bien interfacer le produit avec un autre système.
Il sera nécessaire d'installer sur le bureau de travail les paquets des API pour ensuite pouvoir les importer et les utiliser dans un code.
La démarche d'installation, les spécifications des outils des interfaces et leur description sera détaillée dans une documentation.

### 3.2.4 Maintenance du système

Les développeurs manipulent le code source pour étendre et mettre à jour le système. La démarche sera effectuée lorsque le produit devra être modifer pour des raisons de compatibilité avec le langage et les navigateurs, ou alors pour une extension des fonctionnalités.
Le nouveau modèle est d'abord conçu. On identifie les éléments à modifier et/ou l'on étudie les fonctionnalités del'extension.
Les spécifications seront alors réécrites si nécessaire pour ensuite développer et déployer le système mis à jour.
**Remarque :** A l'issue de la conception du nouveau modèle, une étude des coûts peut être utile pour déterminer s'il faut poursuivre la maintenance.

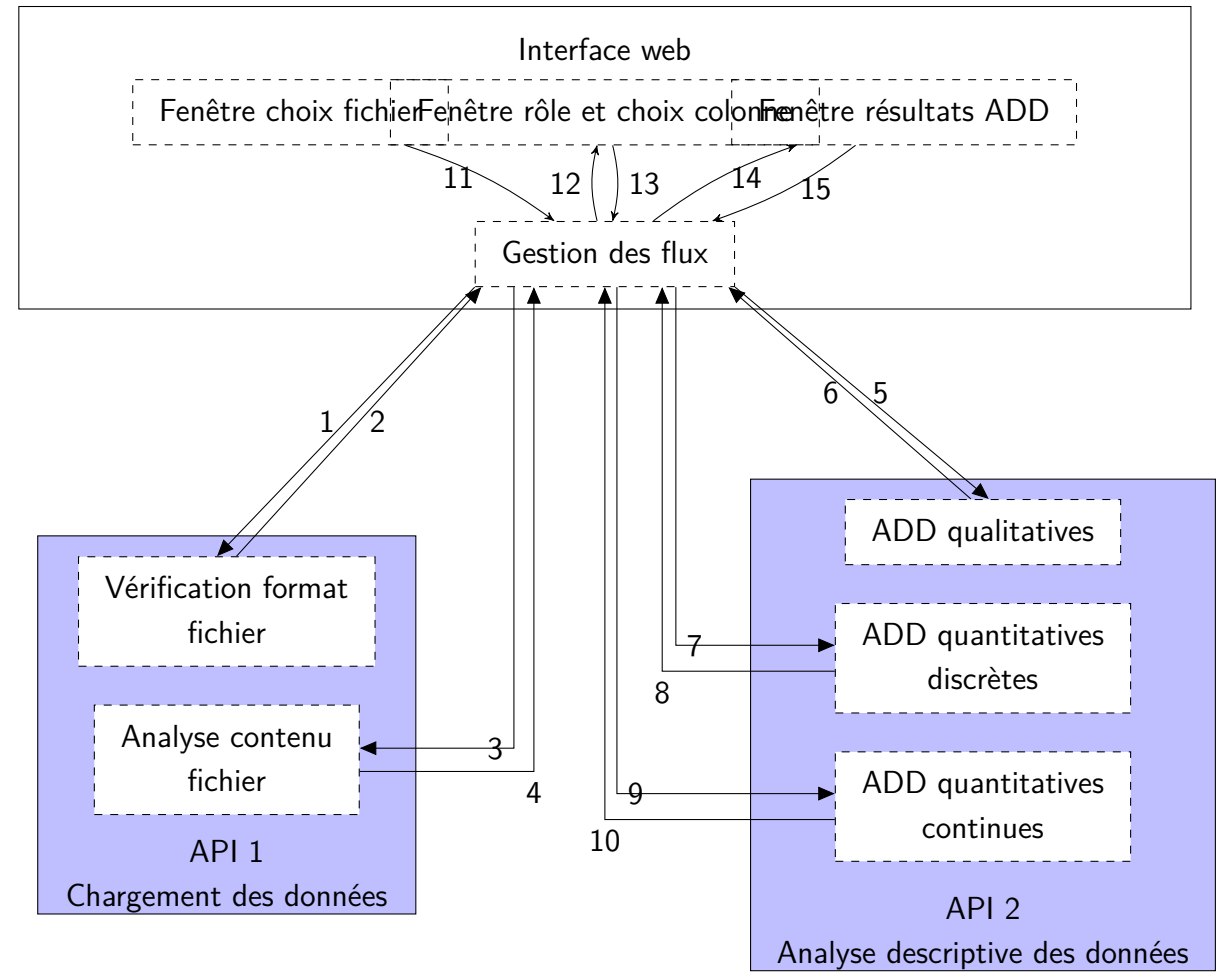



FIGURE 1 – Organigramme des différents modules du logiciel

## 3.3 Présentation de l'organigramme et des fonctionnalités

### 3.3.1 Organigramme et données échangées

**Notes :**
(1) Fichier CSV : lancement de la vérification de son format
(2) Code d'erreur : fichier OK ou ERREUR
(3) Fichier CSV : lancement de l'analyse de son contenu
(4) **structure 1** : contenant les données du fichier, le nombre de lignes et le nombre de colonnes (connus à partir de la taille de la structure)

**structure 2** : contenant 3 informations sur chaque colonne : le type, le rôle et les positions des données erronées ou manquantes
(5) ensemble de données de type qualitatif
(6) erreur ou effectifs, effectifs cumulés, fréquences, fréquences cumulées, diagramme en secteur, histogramme
(7) ensemble de données de type quantitatif discret
(8) erreur ou indicateurs de tendance central, de dispersion et de forme, les anomalies, la distribution des données, un diagramme à moustaches
(9) ensemble de données de type quantitatif continu
(10) même données que (8)
(11) Chemin du fichier CSV importé
(12) Informations du (4) et ensemble de données contenu dans le fichier CSV
(13) Signal de validation du choix colonne et noms des colonnes.
(14) Envoi des résultats d'analyses de (6), (8) et (10)
(15) Demande d'exportation des résultats de l'ADD, ou analyse d'une autre colonne, ou importation d'un autre fichier

### 3.3.2 Format du fichier .csv

Le format du fichier est établi par DCbrain. Le contenu supporté est décrit par des colonnes nécessaires aux types prédéfinies :

- Timestamp : Jour/mois/année heure :minute :seconde
- Père : Nom du noeud

— Enfant : Nom du noeud

— Mesure (unité) : Valeur

**Remarques :** Le graphe de flux utilisé par DCbrain pour analyser les réseaux de ses clients est orienté. D'où l'utilisation des éléments *Père* - *Enfant* pour repérer de quelle connexion on parle.
Les colonnes (une minimum) *Mesure* qui suivent renseignent sur les données mesurées sur les connexions.

### 3.3.3 Fonctionnalités des modules

— Vérification format fichier :
Ce module va vérifier le fichier fourni en entrée en plusieurs points :

1. l'ouverture du fichier a réussi
2. le fichier est un CSV contenant du texte brut non formaté (pas de mise en forme avec des balises ou autres)
3. le fichier est accessible en lecture

**Critère de satisfaction :** le fichier est bien ouvert, accessible en lecture et ne contient que du texte brut.

— Analyse contenu fichier :
Ce module comprend deux fonctionnalités principales :

1. Lecture du contenu du fichier CSV :
   On initialise une première structure (**structure 1**) pour y sauvegarder le contenu du fichier.
   On lit ligne par ligne des caractères du fichier. A chaque fois qu'on détecte un caractère de séparation (une virgule, un point-virgule ou une tabulation), on stocke les caractères lus (la donnée) dans la structure.
   Cette fonctionnalité fournit la **structure 1** contenant les données du fichier, le nombre de lignes et le nombre de colonnes (connus à partir de la taille de la structure).
   **Critère de satisfaction :** l'ensemble des entrée du fichier sont renseignées dans une structure (donc exploitables)
2. Descriptions préliminaires des données de chaque colonne du fichier CSV :
   On initialise une deuxième structure (**structure 2**) pour y stocker des informations sur chaque colonne.

On lit une par une les données de chaque colonne. On vérifie le type de la donnée en le comparant au type attendu. Si le type ne correspond pas, on signale dans la structure que la colonne contient une donnée erronée ou manquante (repérée par sa position dans la colonne). A la fin du parcours d'une colonne, on pourra lui attribuer un rôle/nom.

Cette fonctionnalité fournit donc la **structure 2** contenant 3 informations sur chaque colonne : le type, le rôle et les positions des données erronées ou manquantes.

**Critère de satisfaction :** pour chaque colonne, on connaît son type, son rôle et les éventuels données erronées.

Ce module va donc fournir les deux structures **structure 1** et **structure 2** décrites ci-dessus.

— ADD qualitatives :

1. effectifs, effectifs cumulés
2. fréquence et fréquence cumulée
3. représentations graphiques : diagramme en secteur, histogramme

— ADD quantitatives discrètes :

1. tendance centrale : moyenne, médiane
2. dispersion : quantiles, variance et écart-type -
3. anomalies : boîte à moutaches de Tukey
4. forme (symétrie) : coeff de Pearson ou coeff de Yule
5. forme (aplatissement) : coeff de Fisher
6. représentations graphiques : distribution, cumulatif, boîte à moustaches, ...

— ADD quantitatives continues :

1. découpage en classe selon une précision définie
2. découpage en classe selon une précision par défaut (échelle définie après un parcours des valeurs)
3. représentations graphiques : distribution, boîte à moustaches, ...

— Gestion des flux.

1. Lancement de l'application.
2. Gestion des événements.
3. Gestion des erreurs pour déterminer si oui ou non l'application pour continuer son exécution.
4. Interface entres les différentes fonctionnalités.

— Fenêtre choix fichier.

1. Permet de récupérer un fichier CSV
2. Validation du choix pour passer à la prochaine fenêtre (En renseignant son chemin dans le système de fichier, ou de la manière d'un Drag & Drop).
   **Critère de satisfaction :** Le fichier devra être chargé et devra correspondre à la demande.

— Fenêtre rôle et choix colonne.

1. Affichera le nombre de lignes/colonnes contenu dans le CSV.
   **Critère de satisfaction :** Affiche le bon nombre de lignes et de colonnes.
2. Affichera le titre du fichier.
   **Critère de satisfaction :** Affiche le bon titre du fichier.
3. Affichera un échantillon du contenu du CSV (environ les 1000 premières lignes) avec un système de scroll.
   **Critère de satisfaction :** L'affichage de l'échantillon doit bien se faire sur les 1000 premières valeurs et doivent bien correspondre bien correspondre au contenu du fichier.
4. Affichage des lignes erronés (numéro de la ligne + contenu + type d'erreur).
   **Critère de satisfaction :** Doit bien affiché les lignes contenant les erreurs, le contenu et leur type de façon précise.
5. Mise en place d'un système de navigation sous forme d'onglet (Onglet erreurs, onglet échantillon,...). Cela permettra d'éviter que la fenêtre contienne trop d'informations.
   **Critère de satisfaction :** Le système de navigation doit être lisible et pratique sans trop charger la fenêtre.
6. L'Utilisateur devra sélectionner la colonne avec un clic, puis pourra lancer l'analyse sur celle-ci.
   **Critère de satisfaction :** Sélection de la colonne doit bien se faire + lancement possible après la sélection.

— Fenêtre résultats ADD.

1. La fenêtre affichera les résultats de l'étude qualitative (Médiane, Quantile et anomalie) d'une part et de l'étude quantitative (Histogramme et Diagramme de secteur) d'autre part.
   **Critère de satisfaction :** résultats conformes + diagramme bien dessiné.
2. Une fonctionnalité pour lancer l'exportation des résultats sera disponible (Écriture dans un nouveau fichiers).

**Remarque :** En ce qui concerne les critères de satisfaction de l'ADD, puisqu'il s'agit d'un domaine scientifique, le système devra fournir des résultats conformes aux techniques de ce domaine.

# 4 Exigences non fonctionnelles

## 4.1 Interface utilisateur du produit

### 4.1.1 Exigences d'apparence

Le produit devra adopter une apparence simple, agréable et devra être facile à comprendre dès sa première prise en main. Le maitre d'ouvrage a émis le souhait d'avoir de l'Anglais comme langue utilisée pour l'affichage textuel.

### 4.1.2 Exigences de style

Mots clés : Interface graphique + intéractive.

## 4.2 Utilisabilité

Le produit devra être simple d'utilisation et facile à comprendre dès sa première prise en main, pour ne pas soumettre une formation concernant la manipulation du produit aux futurs utilisateurs.

## 4.3 Exigences de performance

Les exigences de performance du futur produit concerne la latence acceptable que devra avoir l'application. Ici la durée de réponse de l'affichage d'un CSV, ou d'une requête concernant l'analyse de données venant de l'utilisateur ne devra pas être trop longue, le client a suggérer que cette attente devra être de l'ordre de la minute. En ce qui concerne le temps de passage d'une fenêtre à l'autre (choix de(s) colonne(s) pour l'exécution d'une analyse statistique, affichage de la fenêtre d'erreurs, lors du choix du méthode d'affichage des résultats,..), l'application devra répondre de manière fluide.

## 4.4 Précision et exactitude

Lors de l'analyse des données, les résultats sur la Variance et la Moyenne doivent être calculée de façon très précise car elles vont servir de base dans d'autres calculs statistiques. Les autres valeurs calculées peuvent être les plus précises possible mais ces valeurs vont servir pour l'interprétation afin mieux comprendre ce qui se passe sur le réseau.

## 4.5 Maintenabilité du projet

Le produit doit pouvoir être maintenu par ses utilisateurs finaux ou par des développeurs qui ne sont pas les développeurs d'origines, dans le cas où DCbrain souhaiterait ajouter de nouvelles méthodes d'analyses descriptives ou de nouveaux procédés pour afficher les résultats de manière graphique afin de satisfaire les exigences de leurs clients.

Le produit devra permettre l'insertion d'éventuels API supplémentaires, tel que l'ADD multidimensionnelle qui utilisera l'API de l'ADD unidimensionnel dans le but de fournir des descriptions statistiques plus poussées pour obtenir une meilleur vue d'ensemble sur les données collectées. Mais encore l'insertion de technique d'analyse de graphe dans le but d'obtenir des informations sur ces derniers.

## 4.6 Sécurité

### 4.6.1 Accès au système

Le produit final étant une application web, l'accès au système se fera à partir d'un navigateur web pour les utilisateurs. L'accès au produit sera défini grâce à un mécanisme d'adressage (path) dans un système de gestion de fichiers.

### 4.6.2 Intégrité

Le produit manipulera des données fiables. Les données ne représenteront pas de risques pour l'environnement et l'utilisateur du produit final.

### 4.6.3 Protection des données à caractères personnel

Le produit ne devant pas faire appel aux données à caractères personnel des utilisateurs, ne devra donc aucunement altérer ou supprimer de telles données.

# 5 Autres aspects du projet

## 5.1 Question ouvertes

La question de l'esthétisme et la présentation des résultats est laissée ouverte puisqu'il n'y a pas de réponse arrêtée à ce sujet.
Le tout est d'avoir une approche prenant en compte les utilisateurs pour savoir quelles vues faciliteront l'interprétation des analyses.

## 5.2 Tâche à faire

### 5.2.1 Étapes

Le développement du produit se décompose en deux étapes :

— Spécifications
— Développement de l'application
— Compte-rendu

### 5.2.2 Phases de développement

— Build

— Développement des fonctionnalités

— Développement des tests

— Ecriture de la documentation

— Exécution des tests et correction éventuelle du code (debug)

**Remarque** : Ces étapes ne sont pas effféctuées séparément, il sera avantageux de les réaliser en même temps. L'intensité de travail sur chacune des phases va varier tout le long du projet.

## 5.3 Contrôle de la finalisation

La qualité de l'applet sera étudiée selon plusieurs niveaux :

1. Le test des fonctionnalités du système.
2. La vérification de la satisfaction des exigences.
3. Une mesure de la conformité du produit avec les spécifications.

## 5.4 Estimation des coûts

### 5.4.1 Tableau des coûts

| Module | Nombre de lignes | Justification |
|---|---|---|
| Gestion des flux | 15 | Mise en forme du main et appel de l'application |
| Fenêtre choix fichier | 10 + 20 | Fonctions pour : Drag& Drop + Système de fichiers |
| Fenêtre rôle et choix colonne | 5 + 20 + 10 + 10 + 20 | Communication avec le module application + Affichage de l'interface+ lecteurs des valeurs + affichage des valeurs |
| Fenêtre résultats ADD | 10 + 3* 30 | Envoie d'informations au module application + Construction des graphe pour l'ADD pour les 3 types d'analyse |

| | | |
|---|---|---|
| ADD qualitatives | 20 + 20*3 | Application des formules pour les calculs de fréquences et d'effectifs + calcul des valeurs pour la construction de 3 graphes |
| ADD quantitatives discrètes | 60 + 20*2 | Application des formules attaché à l'analyse quantitative discret + calcul des valeurs pour la construction de 2 graphes |
| ADD quantitatives continu | 20 + 10 + 10 + 5 + 20*2 | Parcours + choix précision classe d'intervalle + écriture + communication avec les modules+ calcul des valeurs pour la construction de 2 graphes |
| Vérification format fichier | 30 | Ouverture fichier + vérification si ouverture en lecture + présence de texte formaté ou non |
| Analyse contenu fichier | 20 + 5 + 25 + 10 | Recopie et vérification + initialisation de la structure+ Parcours du fichiers avec condition + Fonction pour donner nom et type de colonne |
| **Coût Total** | **565** | **Estimation totale du coût** |

### 5.4.2 Tableau répartition des tâches

| Module | Malek | Sonny | Jean-Didier | Total |
|---|---|---|---|---|
| Gestion des flux | | | x | 1 |
| Fenêtre choix fenêtre | | | x | 1 |
| Fenêtre rôle et choix colonne | x | | | 1 |
| Fenêtre résultats ADD | | x | | 1 |
| ADD qualitatives | | | x | 1 |
| ADD quantitatives discrètes | | x | | 1 |
| ADD quantitative continues | | x | | 1 |

| | | | | |
|---|---|---|---|---|
| Vérification format fichier | x | | | 1 |
| Analyse contenu fichier | x | | | 1 |

## 5.5 Documentation utilisateur et formation

Le produit final ne nécessitera pas de formation pour apprendre sa manipulation. Mais une documentation utilisateur sera fournie en même temps que le produit, dans lequel sera détaillé les consignes d'installation et d'utilisations.

# 6 Conclusion

D'un point de vue technique, le choix des langages de programmation utilisés pour le développement du produit est justifié par les contraintes et les exigences définies précédemment. La contrainte 1 (fournir une application web) et la contrainte 2 (compatibilité avec ADD) nous permettent de nous fixer sur le choix du langage : ***Python***.
D'une part, ce langage est orienté pour le développement d'applications web robustes et distribuées, déployées et exécutées sur un serveur d'applications. D'autre part, ce langage qui est basé sur Java est doté de plusieurs modules d'analyse de données, dont l'outil ***Weka*** par exemple.
De plus, l'applet Java doit être intégrée dans une page web pour être exécutée : on va donc avoir recours à des langages de balisage comme ***HTML*** pour la présentation et ***CSS*** pour la mise en forme.