# Cahier des charges technique du Projet Météo

La réalisation de ce document doit reprendre :

* Les objectifs mesurables à atteindre,
* les ressources nécessaires,
* la planification de la mise en oeuvre,
* les outils d’évaluation,
* les méthodes de contrôle.

Il contiendra notamment :

* Une proposition de wireframe,
* L’arborescence générale du site Internet,
* La déclinaison de la charte graphique,
* Les solutions logicielles proposées,
* Les langages informatiques et les librairies utilisées,
* Les solutions d’hébergement,
* Les solutions de paiement (dans le cadre d’un site avec une interface de paiement en ligne),
* Les propositions de maintenance et de mises à jour,
* Les ressources à mobiliser,
* Le plan de formation des utilisateurs et gestionnaires,
* Le planning de réalisation et les délais,
* Le coût de la réalisation et de son recettage.

## Contexte et définition du problème

Dans le cadre de la formation AJC nous avons appris à développer des programmes sur des systèmes embarques utilisant le système d’exploitation Linux. Pour valider cet apprentissage un projet regroupant plusieurs personnes a été détermines. Une station météo devra être développer avec des outils dédiés à l’apprentissage.

## Objectif du projet

Le projet a pour objectif de créer une interface graphique affichant des mesures d’humidité, de pression et de température d’un senseur d’environnement BME280 géré par un Raspberry Pi 3.

L’interface graphique se fait en plein écran sur le Raspberry pi et doit afficher les données météo fournies par le capteur en temps réel, ainsi que des icônes correspondant au temps du moment et à la tendance du moment.

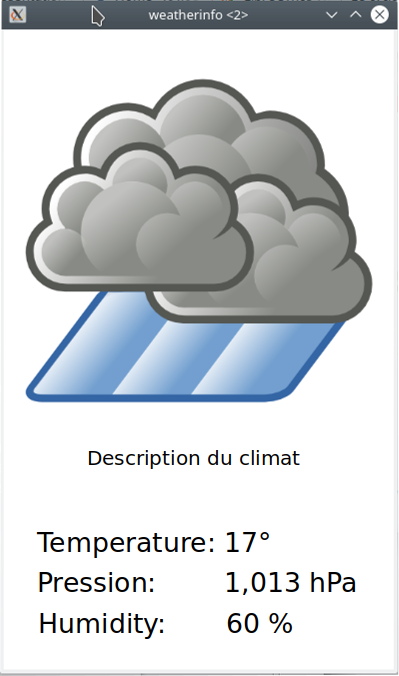
Les calculs pour atteindre ces données devront provenir de l’algorithme Zambretti. La librairie QtQuick sera utilisée pour l’interface graphique. Le programme pourra être déployé automatiquement sous forme de package Debian.

## Périmètre

Comme il s’agit d’un projet dont le but est l’apprentissage nous nous contenterons de développer le programme pour qu’il soit accessible aux personnes de la formations, formateurs y compris.

## Description fonctionnelle des besoins

### Design de l’interface



### Diagramme de classe

### 

## Les critères d’acceptation des produits

La version finale du programme sera testée pour valider sa fiabilité, sa précision.

Le programme sera installer sur un serveur pour le rendre accessible a toutes les personnes de la formations.

## Les contraintes

Il s’agit de notre première gestion de projet. La connaissance élémentaire du développement en C/C++ peut être un frein. Le projet doit être termine le 26 Avril 2019 (Figure 1). Deux pré-soutenances seront réalisées et serviront d’étapes de contrôle de l’avancée du projet. Cependant elles laisseront moins de temps pour le développement du programme.

## La déclinaison de la charte graphique

Simple esthétique

## Les langages informatiques et les librairies utilisées

* Langages
  + C/C++
  + QML
* Libraires :
  + QtQuick

## Les solutions d’hébergement

<https://mentors.debian.net/>

<https://upsilon.cc/~zack/blog/posts/2009/04/howto:_uploading_to_people.d.o_using_dput/>

serveur Tomcat accessible localement

GitHub clone ou chargement du binaire via wget

## Les propositions de maintenance et de mises à jour

Via site hébergeant le paquet Debian.

## Les ressources à mobiliser

Tout le matériel nécessaire au bon fonctionnement du projet a été fourni :

* Capteur BME 280
* Raspberry PI.
* Ecran.
* Fils de connexion.
* Alimentation.

## Logiciel Gestionnaire

Environnement de développement :

* Qt Creator /Kdevelop
* Visual Studio

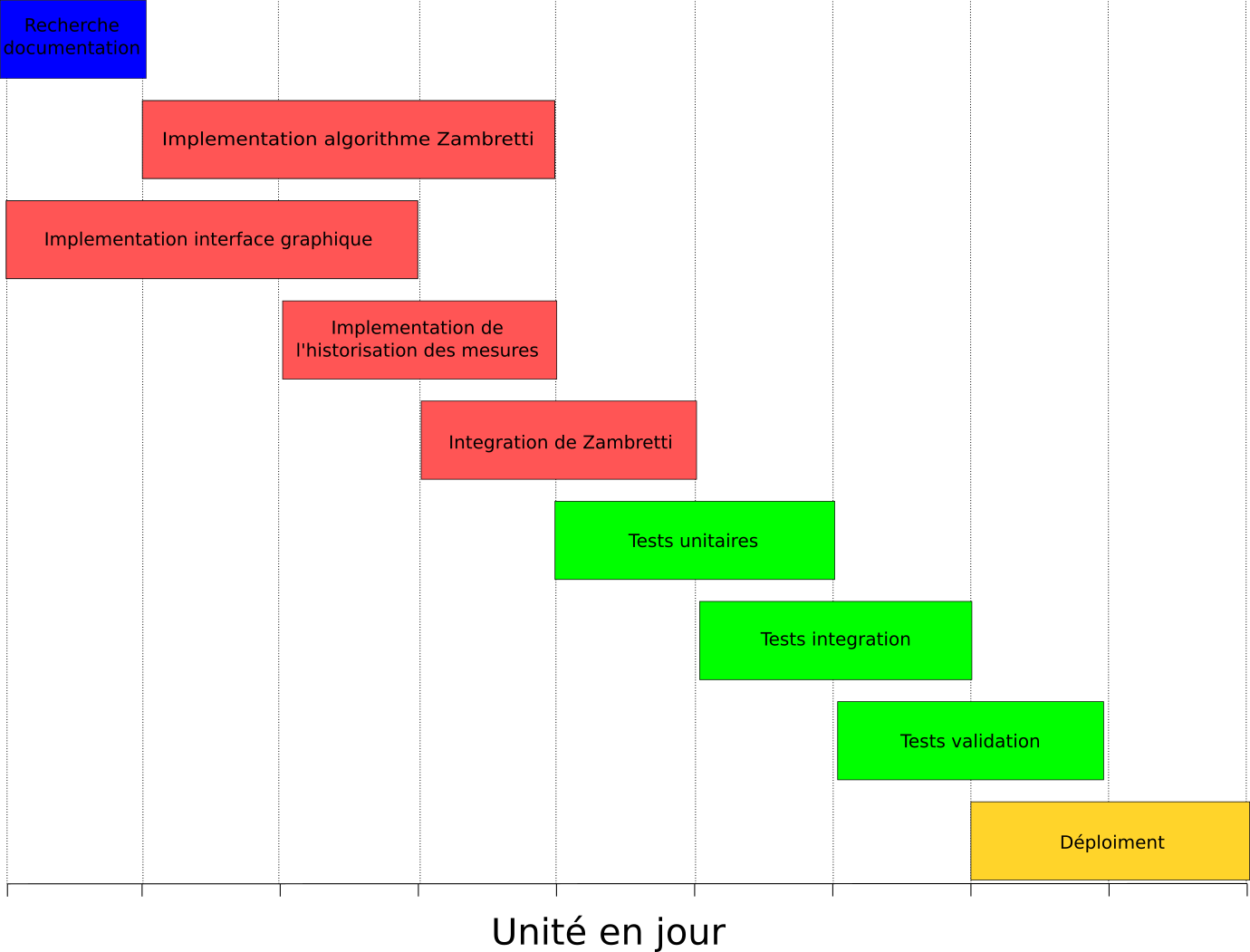
Système de gestion de version :

* Git

Gestion de projet :

* Trello a été utilisé comme gestionnaire de suivit de l’avance du projet.

## Le planning de réalisation et les délais

  
Figure 1: Diagramme de Gantt

**Recherche documentation sur Zambretti :**

- Recherche sur l’algorithme de Zambretti

- Recherche sur la doc du capteur

- Mise ne forme du problème : Identification des moyens calculatoire et algorithmiques à mettre en œuvre pour traiter le problème (conversions appropriées des données du capteur, identifications des étapes de calculs, algorithme principal et premiers sous-algorithmes)

**Création de l’interface graphique :**

- Utilisation d’un nombre de Zambretti de test comme entrée du programme (Le nombre correspond au tableau de Zambretti dans le cas où la partie de programme calculant ce nombre peut fournir un résultat, et sera -1 si le programme ne permet pas de renvoyer le nombre de Zambretti)

- Création de l’interface qui permettra de produire l’affichage voulu.

- Affichage de l’icône et du texte correspondant au nombre de Zambretti.

**Implémentation de l’algorithme Zambretti :**

- Utilisation de données de test comment entrée du programme (Pression en mbar).

- Production du code fournissant le nombre de Zambretti en sortie.

**Création de l’historisation :**

L’historisation doit permettre de garder en mémoire la dernière valeur de moyenne de Pression, ainsi qu’un code d’erreur (dans le cas où la pression ne peut être calculée).

**Intégration de Zambretti dans l’interface :**

- transfert des données de sortie capteurs vers les entrées de Zambretti et de l’interface

- transfert des données de sortie de Zambretti vers l’entrée de l’interface

**Tests unitaires :** à définir

**Tests d’intégration :** à définir

**Tests de validation :** à déf

**Déploiement :** Création d’un paquet Debian.

On pourra aussi éventuellement rendre le paquet accessible par internet.

**Règles de gestion de projet :**

* Séance mise au point 10 minutes journalière évoquant les avancées, les problèmes rencontrés, les solutions à mettre en œuvre.
* Assigner l’importance des fonctionnalités et les priorités dans le traitement de ces fonctionnalités par des scores (cf. diagramme de Gantt)
* Utilisation de Git pour gérer les versions du programme.

## Hypothèses

Il pourrait y avoir de nouvelles fonctionnalités pour le traitement des mesures météorologique. Par exemple la collecte de métriques météo, serveur web, Prometheus et Grafana. Si les critères essentiels du projet sont réalisés dans les meilleurs délais nous pourrions ajouter les fonctionnalités que procurent ces deux logiciels a notre programme. Calculer d’autres données à partir des données récupérées utilisant éventuellement les coordonnées GPS de la localisation du capteur.

## Bibliographie

Qt : https://www.qt.io/

Github : [**https://github.com/**](https://github.com/)

Icones weather :

https://www.deviantart.com/darkobra/art/Tango-Weather-Icon-Pack-98024429

Informations sur l’algorithme de Zambretti

<https://web.archive.org/web/20110610213848/>http://www.meteormetrics.com/zambretti.htm

Définition des paramètres météos :

http://www.meteofrance.fr/prevoir-le-temps/observer-le-temps/parametres-observes