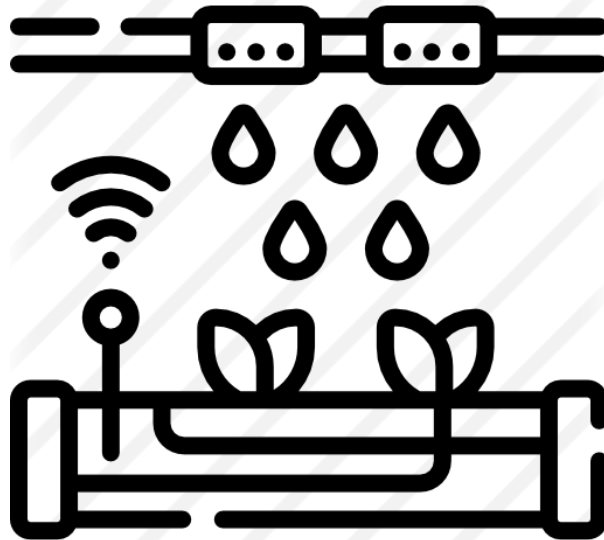


Document de conception



Smart_Irrigation_v0.1

Encadré par :

Pr. S.GENNOUNI

Pr. A.MANSOURI

GSEII2 - 2019-2020

Table des matières

Liste des Figures	5
Liste des Tableaux	6
I. Description générale du projet	7
II. Fonctionnalité entièrement en marche dans la Version v0.1:	8
[Source Code Path : /Group1/Code Source/Serveur/gseii_server]	9
III. Conception et modélisation du partie serveur	10
1. Introduction	10
2. Modélisation fonctionnelle des données	11
3. Choix du formalisme de conception	11
IV. Standards et outils de programmation :	14
1. Choix de la technologie :	14
2. Architectures logiciels et outils de développement :	18
PARTIE CLIENT	22
[Android Source Code Path : /Group1/Code Source/Clients]	22
V. Introduction Générale :	23
VI. Exigences :	23
a. Exigences fonctionnelles établi avec succès :	23
b. Exigences non-fonctionnelles :	23
VII. Contraintes de conception :	23
VIII. Diagramme des cas d'utilisations :	24
IX. Diagramme de séquences :	25
X. Spécifications des cas d'utilisation :	26
XI. Standards et outils de programmation :	27
a. Langage de programmation utilisé :	27
PARTIE MATERIEL	28
[Android Source Code Path : /Group1/Code Source/Hardware]	28

XII. Introduction	29
1. Spécifications des besoins	29
XIII. Méthodologies	29
XIV. Environnement matériel	30
1. Système embarqué	30
➤ <i>Connecter le capteur DS18B20</i>	33
En ligne de commande, taper les lignes suivantes	34
➤ <i>Installer les drivers sur le Pi</i>	34
XV. Environnement logiciel	34
Conclusion Générale :	36

Liste des Figures

FIGURE 1: MODULES DU FRAMEWORK SPRING	16
FIGURE 2: SCHÉMA DU SPRING BOOT	17
FIGURE 3: ARCHITECTURE LOGICIEL DE L'APPLICATION	18
FIGURE 4: DIAGRAMME DES CAS D'UTILISATION	24
FIGURE 5: DIAGRAMME DE SEQUENCES	25
FIGURE 6: FONCTIONNEMENT DU PARTIE HARDWARE	30
FIGURE 7. UNE CARTE RASPBERRY	31

Liste des Tableaux

TABLEAU 1: LISTE DES ROLES DES ACTEURS DU DIAGRAMME DES CAS D'UTILISATION	12
TABLEAU 2: DESCRIPTION DES ENTITES	13
TABLEAU 3: SPECIFICATIONS DES CAS D'UTILISATIONS	26

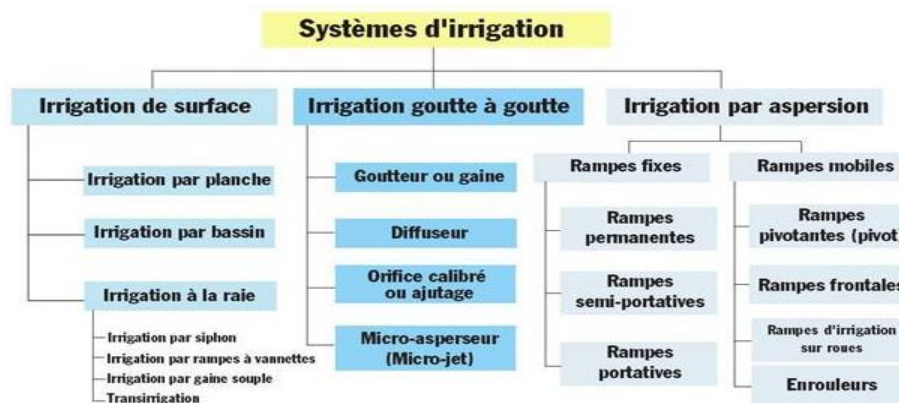
I. Description générale du projet

La protection de l'environnement et l'utilisation rationnelle des ressources naturelles en eau, est un engagement mondial. Dans ce contexte, plusieurs techniques améliorant la consommation de ces ressources de plus en plus rares et convoitées, sont développées, notamment dans le domaine agricole. En effet, il est possible de réduire globalement les doses d'irrigation, en modulant localement les apports selon les besoins des plantes, les réserves en eau et les conditions météorologiques.

Le présent projet consiste alors à concevoir et réaliser un système d'irrigation intelligent. Ce système d'irrigation permet de :

- Réguler automatiquement l'hydratation des plantes grâce à un capteur d'humidité, lié à un système embarqué, afin de réguler l'apport en eau en prenant notamment en compte, les conditions météo, le type de plante surveillée, le type de sol, etc.
- Réguler automatiquement les réserves en eau dans un bassin d'irrigation grâce à un capteur de niveau et une valve de régulation de l'eau, liés à un système embarqué.
- Superviser et commander l'arrosage grâce à une application mobile et une page web.

Ce projet permet de collecter les informations pour surveiller les plantes, intervenir pour la gestion agricole, superviser le système d'irrigation en temps réel et de guidé automatiquement ce processus. Par exemple « activé le système goutte à goutte pendant 5 minutes chaque 2 heures ». La figure suivante illustre les différents systèmes d'irrigation.



II. Fonctionnalité entièrement en marche dans la Version v0.1:

Fonctions	Description de la fonction
Fonctionnalités Principales	
<i>Fonction 1</i>	Sauvegarder périodiquement la valeur de Température dans un fichier texte situé au niveau du serveur.
<i>Fonction 2</i>	Le client a la possibilité de consulter : <ul style="list-style-type: none">✓ La dernière valeur de température mesurée✓ Le graphe contenant les température journalières
<i>Fonction 3</i>	Le client a la possibilité d'activer ou de désactiver la pompe a eau depuis son téléphone portable ou bien une session web

Serveur Web

Spring Framework

**[Source Code Path : /Group1/Code
Source/Serveur/gseii_server]**



III. Conception et modélisation du partie serveur

1. Introduction

Dans notre projet on a adapté une architecture dont on a distingué 3 parties une spécialisée dans l'interfaçage avec le système (Front-end) une autre pour l'acquisition des valeurs à travers la Raspberry pi et la dernière partie est celle concertante le serveur qui va lier les trois parties.

Dans ce chapitre on est censé de présenter la partie du Back-End .

Le Back-End, c'est un peu comme la partie immergée de l'iceberg. Elle est invisible pour les visiteurs mais représente une grande partie du développement d'un projet web. Sans elle, le site web reste une coquille vide.

On peut décomposer le Back-End en trois parties essentielles :

- ✓ Un serveur
- ✓ Une application
- ✓ Une base de données (ou l'on stocke les données de l'application)

Concernant la structure on a divisé cette partie en plusieurs parties

-Configuration : [Server/gseii_server/Configuration](#)

Ici ou on met la configuration nécessaire pour la liaison des différentes hôtes par exemple les adresses IP de la Raspberry pi dans le réseau local

-Database : [Server/gseii_server/src/main/java/org/gseii](#)

Ici ou on a quatre packages services ou on a développé le webservices de l'application tout ce qui concerne la communication avec le mobile et le web

Le deuxième package est dao (data acces objet) : est une interface extends de jpaRepository

Le 3eme package est celui des entités ici on définit les tableaux composant la base de données

Le 4eme est celui de métier : composé de 3 fichiers SensorMetier spécifiant les méthodes d'insertion et la récupération des données de la base de données, le fichier Solution ou on definie la JSCh permettant l'exécution à distance sur la plateforme Raspberry

-Sftp : [Server/gseii_server/src/main/java/org/gseii/sftp](#)

La partie concernant la récupération des données sous format JSON à partir du Raspberry pi

En utilisant le Protocol de communication SFTP dans un dossier partageable après on les copient dans un dossier localement pour conserver l'historique des données.

2. Modélisation fonctionnelle des données

Un modèle d'analyse livre une spécification complète des besoins issus des cas d'utilisations et les structures sous une forme qui facilite la compréhension (scénarios) que nous avons présentés dans les descriptions textuelles et les diagrammes de séquences.

L'étude de conception est la phase la plus importante du cycle de développement d'un système informatique. En effet, elle permet de confronter la spécification et l'analyse avec l'implémentation, elle présente le point de convergence des deux aspects :

« Le quoi faire » (analyse) et le « comment faire » (réalisation).

3. Choix du formalisme de conception

Dans la cadre de notre projet, nous avons opté pour le langage UML comme une approche de conception. Ci-dessous, nous présentons ce langage puis nous justifions notre choix.

UML (Unified Modeling Language) est un langage formel et normalisé en termes de modélisation objet. Son indépendance par rapport aux langages de programmation, aux domaines de l'application et aux processus, son caractère polyvalent et sa souplesse ont fait de lui un langage universel. En plus UML est essentiellement un support de communication, qui facilite la représentation et la compréhension de solution objet. Sa notation graphique permet d'exprimer visuellement une solution objet, ce qui facilite la comparaison et l'évaluation des solutions. L'aspect de sa notation, limite l'ambiguïté et les incompréhensions.

a) Identification des acteurs :

Dans notre cas les acteurs sont :

Les utilisateurs qui peuvent utiliser le système et faire des opérations de récupération et de commande.

Un acteur peut être un :

✓ Client

b) Identification des cas d'utilisations :

Nous décrivons pour chaque acteur les cas d'utilisation. On distingue les cas d'utilisation suivants :

Acteur	Rôle
Client	Commander les moteurs à distance à travers l'application mobile ou bien l'application Web Consulter les valeurs périodiquement des capteurs

Tableau 1: Liste des rôles des acteurs du diagramme des cas d'utilisation

c) Représentation des classes :

La modélisation objet est utilisée dans le langage UML pour définir des objets-métiers et l'architecture de l'application. Ces objets sont créés en tant qu'instance de classe et s'interagissent dynamiquement pour offrir le comportement décrit par les cas d'utilisation. La modélisation objet définit le comportement requis par les différentes classes pour assurer la bonne mise en place des cas d'utilisation et des règles de gestion.

Les objets constituent la base de l'architecture des applications, ils peuvent être réutilisés à travers des domaines d'application ou encore être identifiés et dérivés directement des cas d'utilisation ou des domaines d'application, une classe est composée de :

Attributs : représentant des données dont les valeurs représentent l'état de l'objet.

La méthode : il s'agit des opérations applicables aux objets.

Après avoir réalisé le diagramme des cas d'utilisation, nous pouvons dégager les classes ainsi leurs méthodes et leurs attributs qui sont présentés dans le tableau suivant :

N°	Nom Classe	Liste des Attributs	Méthodes
01	Sensor	idSensor	addSensor(string nom)

		actualHour	deleteSensor(short id)
		actualDay	listSensor()
		valuesOfDay	findSensor(short id)
		minOfWeek	saveValue(short id, double value)
		maxOfWeek	findValues(short id)
		avgOfWeek	findMins(short id)
			findMaxs(short id)
			findAvgs(short id)

Tableau 2: Description des entités

Conclusion :

Tout au long de ce chapitre nous avons mené une conception détaillée du système d'information selon une approche objet afin de garantir la fiabilité et l'efficacité de la phase de réalisation de l'application.

Nous avons dressé une liste des acteurs constituant le système en exprimant leurs besoins, puis nous l'avons détaillé en précisant comment les objets et les acteurs doivent collaborer ensemble selon une dimension temporelle.

Finalement, nous avons décrit l'aspect statique avec les diagrammes des classes.

A l'aide de l'étude de notre cas nous avons déterminé l'environnement de développement de notre application qui sera présentée dans le chapitre suivant.

IV. Standards et outils de programmation :

1. Choix de la technologie :

a) Présentation de la technologie java EE:

La technologie Java est à la base de la plupart des applications en réseau, elle est exploitée dans le monde entier pour développer et fournir des applications mobiles et imbriquées, des jeux, du contenu Web et des logiciels d'entreprise. Utilisée par plus de 9 millions de développeurs dans le monde, la technologie Java permet de développer, de déployer et d'utiliser efficacement des applications et des services fascinants.

Des ordinateurs portables aux centres de données, des consoles de jeux aux superordinateurs scientifiques, des téléphones portables à Internet, la technologie Java est présente sur tous les fronts !

Des chiffres sur l'utilisation de JAVA :

- ✓ 97 % des bureaux d'entreprise exécutent Java ;
- ✓ 89 % des bureaux (ou ordinateurs) des Etats-Unis exécutent Java ;
- ✓ 9 millions de développeurs Java dans le monde ;
- ✓ Choix n° 1 des développeurs ;
- ✓ Plate-forme de développement n° 1 ;
- ✓ 3 milliards de téléphones mobiles exécutent Java ;
- ✓ 100 % des lecteurs Blu-ray livrés avec Java ;
- ✓ 5 milliards de cartes Java utilisées ;
- ✓ 125 millions de périphériques TV exécutent Java ;
- ✓ Les 5 fabricants d'équipement d'origine principaux fournissent Java ME.

La technologie Java a été testée, ajustée, étendue et mise à l'épreuve par une communauté dédiée de développeurs, d'architectes et de passionnés de Java. Elle a été conçue pour permettre le développement d'applications portables hautes performances sur une large gamme de plates-formes informatiques. Grâce à la mise à disposition d'applications dans des environnements hétérogènes, les entreprises peuvent proposer davantage de services et dynamiser la productivité, la communication et

la collaboration de l'utilisateur final, tout en réduisant considérablement le coût de propriété des applications d'entreprise et grand public. Java est aujourd'hui devenue un outil indispensable qui permet aux développeurs :

- ✓ d'écrire des logiciels sur une plate-forme et de les exécuter sur pratiquement toutes les autres plates-formes,
- ✓ de créer des programmes qui peuvent être exécutés dans un navigateur Web et accéder aux services Web disponibles,
- ✓ de développer des applications côté serveur pour des forums, des magasins et des sondages en ligne, pour le traitement de formulaires HTML, etc.,
- ✓ de combiner des applications ou des services basés sur le langage Java pour créer des applications ou des services très personnalisés,
- ✓ d'écrire des applications puissantes et efficaces pour les téléphones portables, les processeurs à distance, les microcontrôleurs, les modules sans fil, les capteurs, les passerelles, les produits de consommation et tous les autres types de dispositif électronique.

Java EE est un environnement Java indépendant de toute plate-forme qui sert à développer, fabriquer et déployer des applications d'entreprise Web en ligne. Java EE comprend de nombreux composants de Java Standard Edition (Java SE). La plate-forme Java EE est constituée d'un jeu de services, d'API et de protocoles qui permettent de développer des applications Web multinationaux.

Les développeurs pour entreprises ont besoin de Java EE car l'écriture d'applications professionnelles distribuées n'est pas une tâche aisée. Ils ont besoin d'une solution hautement performante qui leur permette de se concentrer exclusivement sur l'écriture de la logique d'entreprise tout en disposant d'une gamme complète de services professionnels tels que des objets distribués transactionnels, des intergiciels orientés vers les messages et des services de répertoire et de nom.

b) Présentation du framework Spring :

Le développement d'une application web directement avec Java EE en respectant les bonnes pratiques de la programmation s'avère une tâche fastidieuse est presque impossible à réaliser vue la contrainte du temps réserver pour le projet.

C'est pourquoi l'utilisation d'un framework va nous faire gagner du temps et nous permettre de développer une application en respectant les normes.

D'abord c'est quoi un framework ?

Un framework est un cadre facilitant le développement, il s'agit généralement d'un ensemble d'API masquant la complexité d'autres API sous-jacents. Ils existent beaucoup de frameworks basés sur le langage Java sur le marché tels que Chrysalis, Echo 2, Echo, Expresso Framework, Jaffa, Jucass, Maverik, Millstone, Spring, MVC, Struts, Tapestry, Turbine, VRaptor, Wicket, ...

Mais Spring reste le plus utilisé par plusieurs sociétés à travers le monde

C'est quoi Spring ?

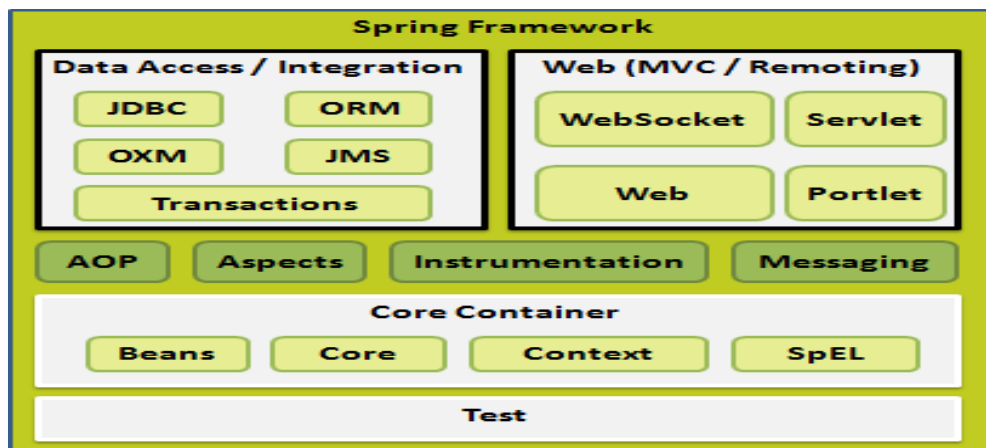


Figure 1: Modules du framework Spring

Spring est considéré comme un conteneur dit « léger ». La raison de ce nommage est expliquée par Erik Gollot dans l'introduction du document « Introduction au framework Spring ».

Spring est effectivement un conteneur dit « léger », c'est-à-dire une infrastructure similaire à un serveur d'applications J2EE. Il prend donc en charge la création d'objets et la mise en relation d'objets par l'intermédiaire d'un fichier de configuration qui décrit les objets à fabriquer et les relations de dépendances entre ces objets. Le gros avantage par rapport aux serveurs d'application est qu'avec Spring, les classes n'ont pas besoin d'implémenter une quelconque interface pour être prises en charge par le framework (au contraire des serveurs d'applications J2EE et des EJBs). C'est en ce sens que Spring est qualifié de conteneur « léger ».

Spring s'appuie principalement sur l'intégration de trois concepts clés :

- ✓ **L'inversion de contrôle est assurée de deux façons différentes: la recherche de dépendances et l'injection de dépendances ;**
- ✓ **La programmation orientée aspect ;**
- ✓ **Une couche d'abstraction.**

Pour l'inversion de contrôle, la recherche de dépendance consiste pour un objet à interroger le conteneur, afin de trouver ses dépendances avec les autres objets. C'est un cas de fonctionnement similaire aux EJBs.

Alors que l'injection de dépendances peut être effectuée de trois manières possibles:

- ✓ L'injection de dépendance via le constructeur.
- ✓ L'injection de dépendance via les modificateurs (setters).
- ✓ L'injection de dépendance via une interface.
- ✓ Les deux premières sont les plus utilisées par Spring.

En ce qui concerne la couche d'abstraction, elle permet d'intégrer d'autres frameworks et bibliothèques avec une plus grande facilité. Cela se fait par l'apport ou non de couches d'abstraction spécifiques à des frameworks particuliers. Il est ainsi possible d'intégrer un module d'envoi de mails plus facilement.

Ce framework, grâce à sa couche d'abstraction, ne concurrence pas d'autres frameworks dans une couche spécifique d'un modèle architectural Modèle-Vue-Contrôleur mais s'avère un framework multi-couches pouvant s'insérer au niveau de toutes les couches; modèle, vue et contrôleur. Ainsi il permet d'intégrer **Hibernate** ou **iBATIS** pour la couche de persistance ou encore Struts et JavaServer Faces pour la couche présentation.

Mais malheureusement Spring est connu pour sa configuration qui peut s'avérer complexe et fastidieuse. Il n'était pas rare de passer plusieurs jours sur la configuration d'un projet Spring notamment pour des personnes novices dans l'utilisation du framework. Fort de ce constat, les équipes de Spring décidèrent de travailler sur un projet permettant de faciliter le développement d'application Spring pour les développeurs. C'est ainsi qu'est né **Spring Boot**.

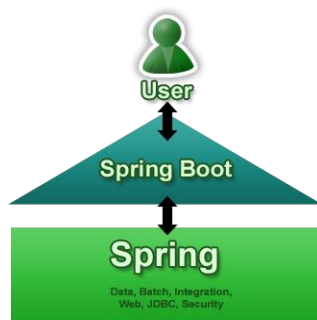


Figure 2: Schéma du Spring Boot

Spring Boot est un projet ou un micro framework qui a notamment pour but de faciliter la configuration d'un projet Spring et de réduire le temps alloué au démarrage d'un projet.

Tout est déjà configuré pour le développeur Spring, il ne lui reste qu'à se concentrer sur le code métier.

2. Architectures logiciels et outils de développement :

a) Architecture logiciel de l'application :

Notre application est basée sur un modèle en couche comme le montre la figure suivante:

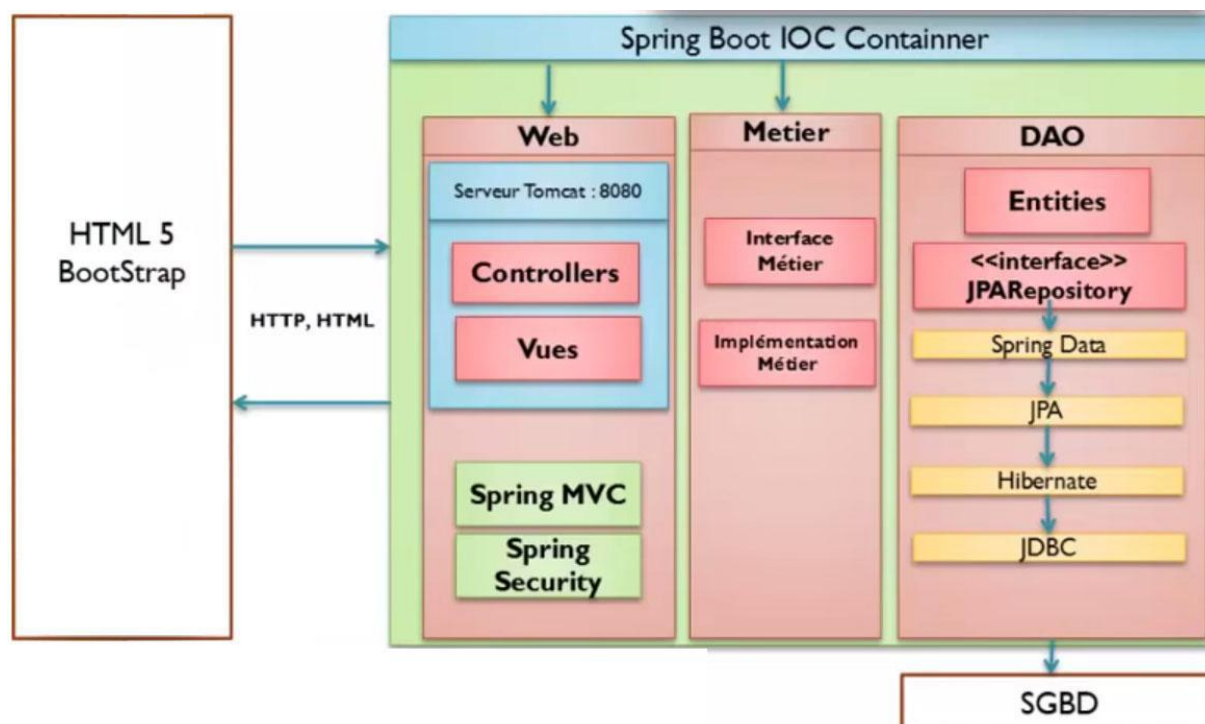


Figure 3: Architecture logiciel de l'application

En effet, c'est l'architecture à trois niveaux ou architecture à trois couches qu'est l'application du modèle plus général multi-tiers.

Son nom provient de l'anglais 'tier' signifiant étage ou niveau. Il s'agit d'un modèle logique d'architecture applicative qui vise à modéliser une application comme un empilement de trois couches logicielles (étages, niveaux, tiers ou strates) dont le rôle est clairement défini:

La présentation des données (**couche WEB**) : correspondant à l'affichage, la restitution sur le poste de travail, le dialogue avec l'utilisateur ;

Le traitement métier des données (**couche Métier**) : correspondant à la mise en œuvre de l'ensemble des règles de gestion et de la logique applicative ;

L'accès aux données persistantes (**couche DAO –Data Access Objet**) : correspondant aux données qui sont destinées à être conservées sur la durée, voire de manière définitive.

b) Présentation des modules Spring utilisés :

✓ Spring Data JPA:

Spring Data facilite l'écriture des couches d'accès aux données et tente d'offrir une abstraction commune pour l'accès aux données quel que soient les sources de données sous-jacentes, tout en prenant en compte les spécificités de celles-ci

Spring Data JPA offre une couche d'abstraction supplémentaire par rapport à JPA et se charge de l'implémentation des fonctionnalités les plus courantes des DAO.

Dans notre projet, on se concentre sur l'essentiel : l'écriture des requêtes, comme c'est indiqué dans la figure ci-dessous:

Pour l'intégrer, il suffit d'ajouter une dépendance dans Maven :

```
<dependency>  
  <groupId>org.springframework.boot</groupId>  
  <artifactId>spring-boot-starter-data-jpa</artifactId>  
</dependency>
```

Spring Data JPA utilise Hibernate qui est un framework ORM (Object-Relational-Mapping) pour assurer le requêtage en base des données. Le concept d'ORM (Object-Relational-Mapping) est un concept de persistance permettant de transformer de manière transparente un objet en données au sein d'un SGBDR et vice versa.

✓ Conteneur Spring IOC :

Le conteneur IOC est au cœur du framework Spring, il est responsable d'instancier, configurer et assembler les objets, gérer leur cycle de vie complet de la création jusqu'à la destruction. Il utilise l'injection de dépendance (DI) pour gérer les éléments qui composent une application.

Le conteneur reçoit les instructions sur les objets à instancier, configuré, assemblé par la lecture des métadonnées de configuration fournis. Les métadonnées de configuration peuvent être représentées soit par XML, annotations Java, ou un code Java. Le conteneur Spring IOC rend l'utilisation des

classes Java POJO et les métadonnées de configuration pour produire un système ou une application entièrement configurée et exécutable.

Il existe deux types de conteneurs IOC :

- **BeanFactory**: Interface qui existe dans le package `org.springframework.beans.factory.BeanFactory`
- **ApplicationContext** : Interface qui existe dans le package `org.springframework.context.ApplicationContext`

BeanFactory et ApplicationContext sont des interfaces qui agissent comme des conteneurs IoC.

L'interface ApplicationContext est construite au-dessus de l'interface BeanFactory. Il ajoute quelques fonctionnalités supplémentaires que BeanFactory tels que l'intégration simple avec l'AOP de Spring, la couche application contexte spécifique (par exemple WebApplicationContext) pour les applications Web. Il est donc préférable d'utiliser ApplicationContext que BeanFactory.

c) Outils de développement utilisés :

✓ **Eclipse IDE for java EE :**

Eclipse est un environnement de développement intégré (Integrated Development Environment) dont le but est de fournir une plate-forme modulaire pour permettre de réaliser des développements informatiques.

✓ **Maven :**

Maven est un outil d'automatisation de construction utilisé principalement pour des projets Java. Maven aborde deux aspects grands aspects: il décrit comment le logiciel est construit, et deuxièmement, il décrit ses dépendances.

✓ **MySQL :**

MySQL est un serveur de bases de données relationnelles Open Source, il stocke les données dans des tables séparées plutôt que de tout rassembler dans une seule table. Cela améliore la rapidité et la souplesse de l'ensemble.

Les tables sont reliées par des relations définies, qui rendent possible la combinaison de données entre plusieurs tables durant une requête.

Le SQL dans "MySQL" signifie "Structured Query Language" : le langage standard pour les traitements de bases de données.

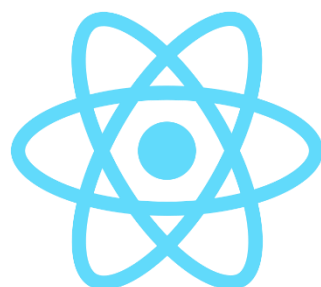
d) Conclusion

Dans ce chapitre, au départ, nous avons mis en avant les outils de développement utilisés pour la programmation de l'application par la suite nous avons présenté les modules et architectures du framework Spring et nous avons clôturé par la présentation des écrans de l'application réalisée.

Partie Client

Clients Mobile/Web

**[Android Source Code Path : /Group1/Code
Source/Clients]**



V. Introduction Générale :

L'application mobile et l'interface web sont conçue à être des clients, c'est-à-dire aucun traitement se fait au sein de ces derniers.

A l'aide d'une connexion internet l'utilisateur peut accéder à plusieurs fonctionnalités que cette application Web/Mobile offre.

VI. Exigences :

a. Exigences fonctionnelles établi avec succès :

Les clients Mobile/Web se mis à plusieurs exigences fonctionnelles :

- ✓ Contrôle des périodes d'irrigation manuellement d'après application ;
- ✓ Affichages des statistiques recueillis par les capteurs sous forme de graphs ;
- ✓ Affichage des dernières valeurs recueillis par les capteurs ;

b. Exigences non-fonctionnelles :

- ✓ Disponibilité de statistiques temps réel ;
 - ✓ Intégrité des données ;
 - ✓ Temps de réponse <2s ;
 - ✓ Front-End à améliorer ;
-

VII. Contraintes de conception :

L'application se mis à une contrainte principale :

- ✓ La communication entre l'app mobile/Web et le Serveur se fait par le web service REST ;

VIII. Diagramme des cas d'utilisations :

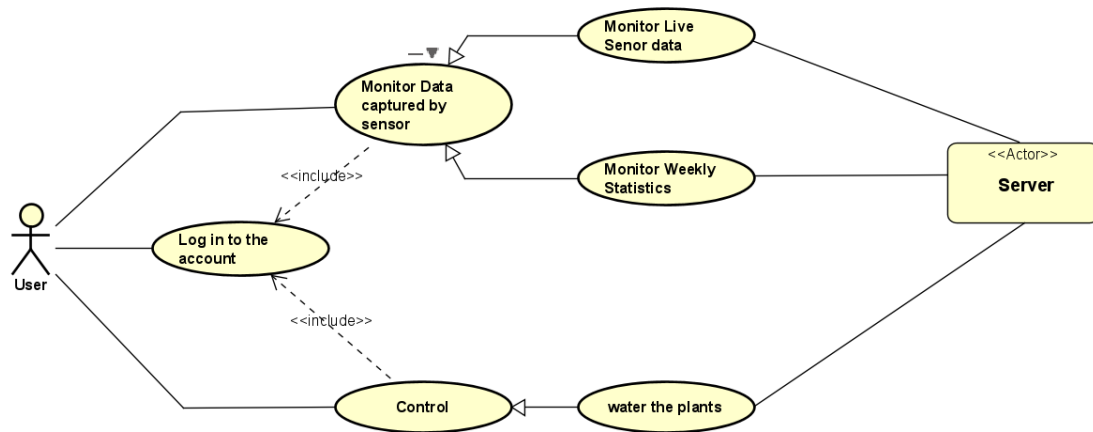


Figure 4: Diagramme des cas d'utilisation

IX. Diagramme de séquences :

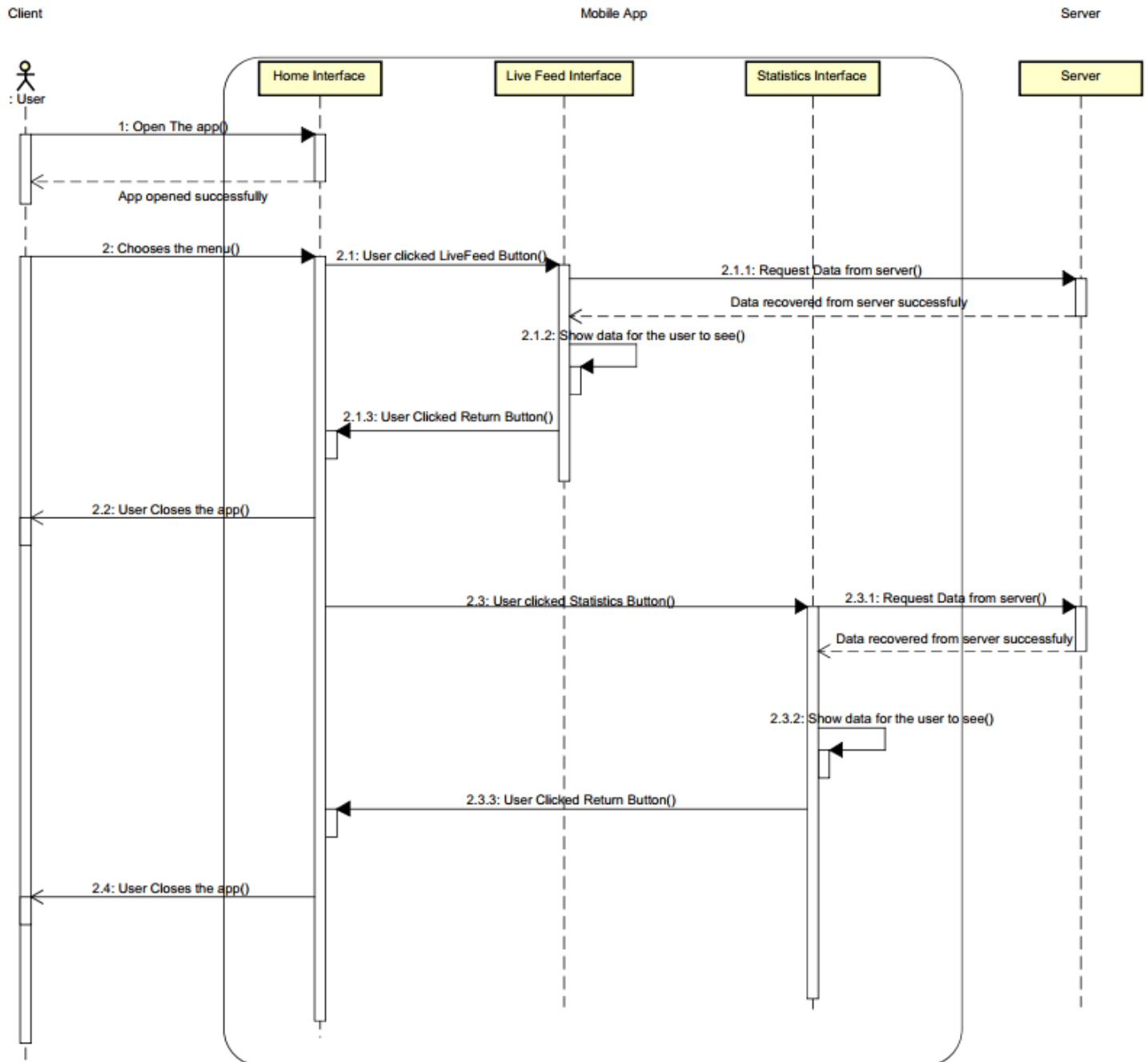


Figure 5: Diagramme de séquences

X. Spécifications des cas d'utilisation :

Acteur(s)	Utilisateur Application Web/Mobile Serveur		
Exigence(s) spécial(s)			
Exigence spéciale numéro 1	Android Smartphone (Android Version supérieur à 8.0) Navigateur Internet		
Précondition(s)			
Précondition numéro 1	Connexion Internet disponible.		
Flux d'évènements			
Flux de base	1.Utilisateur ouvre application 2.Le menu principal s'affiche		
Flux alternatif 1 : Class <u>livemeasurement</u> <u>.java</u>	1.Utilisateur choisi d'afficher la dernière température acquit par le capteur 2.La température s'affiche 3.Utilisateur peut revenir au menu principale		
Flux alternatif 2 : Class <u>Statistics.java</u>	1.Utilisateur choisi d'afficher les statistiques sous forme de graph 2. Graph s'affiche 3.Utilisateur peut revenir au menu principale		
Historique des versions			
Date (04-01-2020)	V0.1	La 1ere version avec des Fonctions de base	Auteur(s) : Equipe Smart Irrigation

Tableau 3: Spécifications des cas d'utilisations

XI. Standards et outils de programmation :

a. Langage de programmation utilisé :



Java c'est un langage de programmation orienté objet ;

- Version utilisée : Java 8, v231



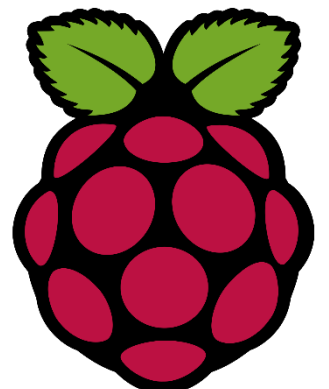
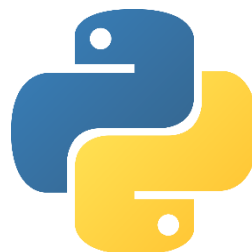
Android Studio c'est l'environnement de développement intégré (IDE) officiel du système d'exploitation Android de Google, basé sur le logiciel IntelliJ IDEA de JetBrains et conçu spécifiquement pour le développement Android ;

- Version utilisée : v3.5.3

Partie Matériel

Raspberry Pie + Python

[Android Source Code Path : /Group1/Code
Source/Hardware]



XII. Introduction

Dans ce chapitre, nous allons nous intéresser à la partie conception qui est une étape primordiale dans le processus de développement en raison des décisions prises dans cette étape et leurs impacts sur les performances de la solution. Nous allons présenter dans un premier temps l'aspect conceptuel de notre solution. Ensuite, nous allons représenter l'environnement matériel et logiciel utilisé. Enfin, nous allons discuter le choix du langage de développement logiciel.

1. Spécifications des besoins

L'objectif principal est de développer et concevoir un système qui soit :

- ✓ **Efficace** : le système doit transmettre les données sans interruptions ou coupures.
 - ✓ **Souple** : il doit être facile à manipuler sans connaissance particulière en informatique ou en électronique.
 - ✓ **Extensible** : il doit être conçu de telle sorte que nous pouvons le modifier sans besoin de reprendre tout le travail.
 - ✓ **Flexible** : le système peut être adapté à d'autres cas et situations d'application qui font partie du « smart irrigation ».
-

XIII. Méthodologies

Pour procéder à la conception de l'application, nous l'avons modélisé en utilisant le logiciel de création de diagrammes Microsoft Visio.

Microsoft Visio

C'est un logiciel de diagrammes et de synoptique pour Windows qui fait partie de la suite bureautique Microsoft Office mais se vend séparément.

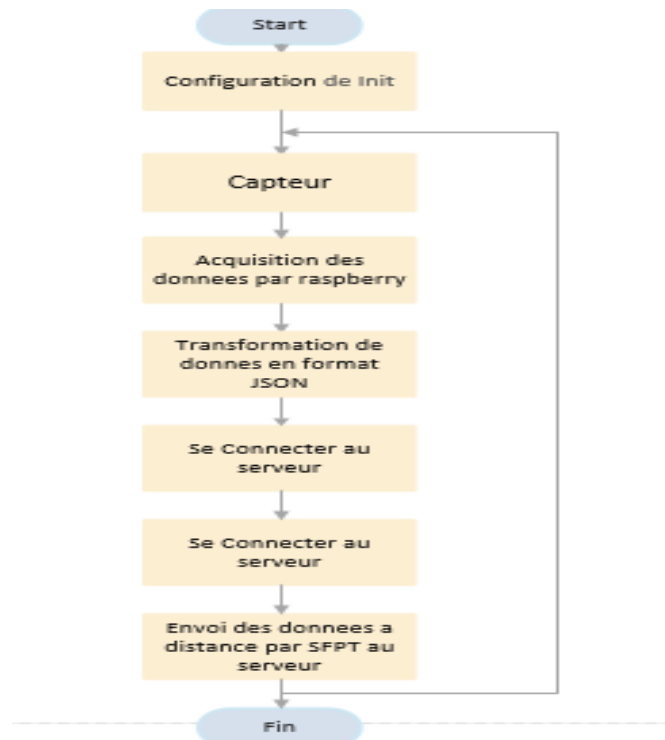


Figure 6: Fonctionnement du partie Hardware

Après la configuration du fichier init pour que le code s'exécute à chaque démarrage du Raspberry pi ,

la care pi fait l'acquisition des données à chaque période à travers un capteur de température puis les transforme en fichiers JSON après elle les envoi au serveur par le protocole SFTP .

XIV. Environnement matériel

1. Système embarqué

Un système embarqué est un système électronique complexe qui intègre du logiciel et du matériel conçus ensemble afin de fournir des fonctionnalités données. Il contient généralement un ou plusieurs microprocesseurs destinés à exécuter un ensemble de programmes définis lors de la conception et stockés dans des mémoires. Le système matériel et l'application logiciel sont intimement liés et immergés dans le matériel et ne sont pas aussi facilement discernables comme dans un environnement de travail classique de type ordinateur de bureau PC (Personal Computer).

a) Materiel utilisé :

✓ Carte Raspberry

La Raspberry Pi modèle B est un mini-ordinateur à processeur ARM. Cet ordinateur, qui a la taille d'une carte de crédit, permet l'exécution de plusieurs variantes du système d'exploitation libreGNU/Linux et des logiciels compatibles. Il est fourni d'une seule carte mère, sans boîtier, sans alimentation, sans clavier, sans souris ni écran dans l'objectif de diminuer les coûts et de permettre l'utilisation de matériel de récupération.

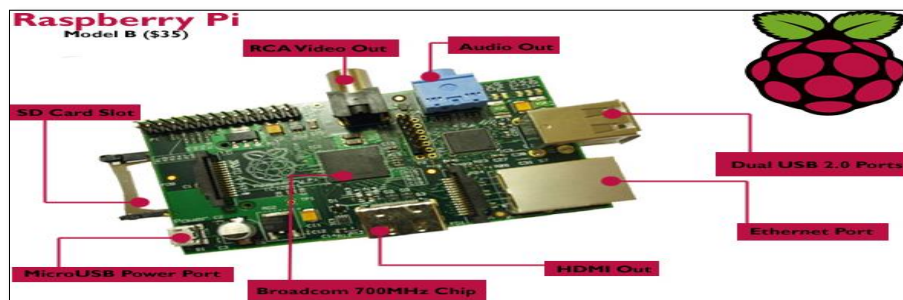


Figure 7. Une carte Raspberry

Ce mini-ordinateur est destiné à encourager l'apprentissage de la programmation informatique. Il contient deux ports USB et un port RJ45 et intègre un puissant processeur ARM cadencé à 700 MHz et une mémoire vive de 512Mo.

✓ Choix de la plateforme

Il est simple et difficile de choisir entre les plates-formes existantes. Il est simple de choisir s'il s'agit de voir le coût comme paramètre de sélection par exemple. Mais il est difficile de choisir si l'application à développer impose des contraintes surtout le temps-réel.

Dans le dernier cas, plusieurs paramètres entrent en jeu : taille mémoire, rapidité du processeur, système d'exploitation, ...

- Raspberry Pi est petit...

De la **taille d'une carte de crédit**, une carte Raspberry Pi est parfaitement adaptée pour être

Intégrée dans un robot. Pas besoin d'ajouter de la place supplémentaire, ou une charge

Additionnelle de 2 kg pour rendre votre robot intelligent.

- Pas cher – moins de 40€

Oui, vous avez bien lu : moins de 40€ pour un ordinateur complet. **Raspberry Pi est**

vraiment accessible. De ce fait, acheter une carte pour votre robot ne vous ruinera pas, et n'augmentera pas de façon considérable le prix final du robot. Vous pourrez bénéficier de nombreuses fonctionnalités très utiles, pour un prix plus que raisonnable.

- **Raspberry Pi est puissant**

Malgré sa petite taille, ne sous-estimez pas les capacités d'une carte Raspberry Pi ! Celle-ci est aussi puissante qu'un smartphone (pas les dernières modèles). La dernière version en date, Raspberry Pi 3, comprend un **processeur 4 coeurs ainsi que 1GB de RAM**. Vous pouvez y installer un vrai système d'exploitation, comme par exemple Raspbian, Ubuntu ou Windows IoT, de quoi faire tourner la plupart des applications que vous utilisez sur votre ordinateur portable.

- **Beaucoup d'entrées/sorties**

Une carte Raspberry Pi n'est pas seulement un ordinateur. Avec **40 pins GPIO**, vous pouvez facilement connecter votre carte avec de nombreux capteurs et composants électroniques. En utilisant des protocoles comme **serial, i2c, ou encore spi**, vous êtes assuré de pouvoir utiliser un maximum de composants électroniques. Ceci est un point important, car cela rend Raspberry Pi **compatible avec la plupart des appareils et composants** que vous trouverez pour vos projets robotiques.

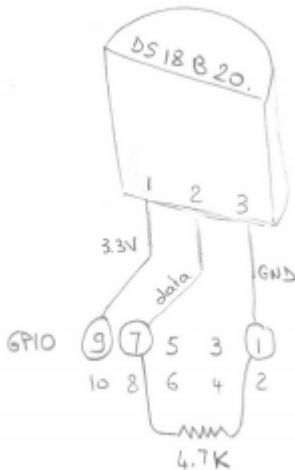
- **Une multitude d'appareils connectés**
- **Très facile d'utilisation par rapport aux technologies industrielles**

Tout le monde peut utiliser une carte Raspberry Pi. Tout ce que vous avez à faire, c'est de télécharger un système d'exploitation, de l'écrire sur une carte microSD, de connecter votre Raspberry Pi à un écran, et c'est tout !

capteur DS18B20

Le capteur DS18B20 est un capteur à coût réduit (10 € les 5 capteurs étanches avec câble de 1 mètre ou 8 € les 10 capteurs de base), qui utilise le protocole 1-wire. On peut connecter plusieurs capteurs DS18B20 en série sur un même port du Raspberry Pi. Nous allons voir ici comment connecter un seul thermomètre et en écrire la valeur dans un fichier txt par un script Shell ou PHP.

➤ Connecter le capteur DS18B20



Le capteur peut mesurer des températures entre -55°C et $+125^{\circ}\text{C}$ et reste précis à 0.5°C sur l'intervalle -10°C à $+85^{\circ}\text{C}$. Chaque capteur a un numéro de série unique (64 bits) et peut donc être identifié précisément.

Pour **une seule sonde**, on câble comme dans le schéma de droite.

On peut mettre **plusieurs capteurs en série** (en théorie autant qu'on veut, en pratique il semble qu'il vaille mieux se limiter à 10 capteurs). Dans ce cas, on connecte une seule résistance 4.7 K et plusieurs sondes selon le schéma ci-dessous.

Paramétrer le Pi pour lire la / les températures

Editer `/boot/config.txt` pour qu'il contienne la ligne suivante puis redémarrer le Pi :

```
dtoverlay=w1-gpio
```

Tester le capteur

En ligne de commande, taper les lignes suivantes

```
sudo modprobe w1-gpio
sudo modprobe w1-therm
cd /sys/bus/w1/devices
ls
cd 28-xxxx (changer pour correspondre aux numéros de série qui s'affichent)
cat w1_slave
```

Si le capteur est correctement câblé, on voit apparaître quelque chose qui ressemble à ce qui

suit, où `t=20187` signifie que la température est 20.187°C :

```
43 01 4b 46 7f ff 0d 10 bd : crc=bd YES
43 01 4b 46 7f ff 0d 10 bd t=20187
```

S'il y a plus d'un capteur de température, `ls` dans `/sys/bus/w1/devices` affichera plusieurs répertoires.

➤ Installer les drivers sur le Pi

`modprobe w1-gpio` et `modprobe w1-therm` sont les drivers pour les capteurs de

température. Ils doivent démarrer lorsque le Pi démarre. Pour cela, ajouter les lignes suivantes à `/etc/modules` :

```
w1-gpio
w1-therm
```

On redémarre le Pi et les drivers sont maintenant chargés.

XV. Environnement logiciel

Python est un langage de programmation interprété, multi-paradigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage

dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions ; il est ainsi similaire à Perl, Ruby, Scheme, Smalltalk et Tcl.

Le langage Python est placé sous une licence libre proche de la licence BSD6 et fonctionne sur la plupart des plates-formes informatiques, des smartphones aux ordinateurs centraux⁷, de Windows à Unix avec notamment GNU/Linux en passant par macOS, ou encore Android, iOS, et peut aussi être traduit en Java ou .NET. Il est conçu pour optimiser la productivité des programmeurs en offrant des outils de haut niveau et une syntaxe simple à utiliser.

Pourquoi Python est-il si populaire ?

Un certain nombre de raisons sont à l'origine de ce phénomène.

1. Simplicité

Python est un langage orienté objet simple à utiliser et à mettre en oeuvre, il est donc logique qu'un grand nombre de développeurs affluent pour programmer avec. De plus, le code Python est propre et facile à comprendre.

2. Polyvalence

Python fonctionne sur plusieurs types de systèmes et de plateformes, cela fait partie des points sur lesquels il se distingue des autres langages de programmation. Sa force réside également dans le fait qu'il peut être utilisé pour réaliser diverses typologies de projets, pour n'en citer que quelques uns :

- développement web, développement de logiciel
- très souvent utilisé dans la Data Science
- pour les opérations de systèmes
- coder des jeux vidéo avancés
- certains l'ont même expérimenté avec l'Intelligence Artificielle

Il n'y a pas de limites aux applications Python.

3. Possibilités web

Python dispose d'un code propre et d'une énorme documentation. Cela aide les développeurs à créer et à personnaliser leur code facilement et efficacement. La syntaxe facilite le processus de révision du code. Python possède également plusieurs GUI frameworks comme par exemple, Django, qui permettent la création d'applications graphiques, la réutilisation de code et aussi la réalisation d'applications multi-plateformes.

4. Promis à un bel avenir

Python domine les concepts émergents tels que la Data Science, l'Intelligence Artificielle et le Machine Learning. Ces technologies deviennent de plus en plus importantes pour les industries qui génèrent beaucoup de données et qui oeuvrent abondamment sur l'automatisation. Python a prouvé qu'il était capable de répondre à ces nouvelles problématiques, c'est pourquoi il est de plus en plus populaire auprès des développeurs.

Le protocole utilisée :

SFTP signifie SSH File Transfer Protocol ou Secure File Transfer Protocol. Comme l'indique la première définition, SFTP fait partie de SSH ou Secure Shell. Il s'agit d'un remplaçant sûr pour l'établissement d'une session de terminal sur des machines UNIX. SFTP est le composant de ce protocole SSH qui assure le transfert de fichiers.

Conclusion Générale :

Les 2 premières fonctionnalités demandées par l'encadrant sont achevées avec succès et marchent parfaitement, en plus d'une fonctionnalité supplémentaire qui la commande.