

Plan de tests

JUMEAU NUMÉRIQUE DANS L'INDUSTRIE DU FUTUR **L2G1-2021**



Plan de tests

PROJET JUMEAU NUMÉRIQUE

Identification du document

| |
|---|
| Référence du document : L2G1.T |
| Version du document : 1.1 |
| Date du document : 08/04/2022 |
| Auteurs : Wattrelos Tigran Boka Ricardo Abou Assaf Mahyr-Florian Daumont-Ouk Ilan' |

Sommaire

| | |
|--------------------------|----------|
| Introduction | 3 |
| Tests à effectuer | 5 |
| Test unitaire : | 5 |
| Test d'intégration : | 6 |
| Test fonctionnel : | 8 |

Introduction

L'objectif principal du plan de tests est de démontrer l'ordre dans lequel chaque composant de notre système sera complété, testé individuellement, et intégré avec les autres composants du système. Il est nécessaire de détecter le plus rapidement possible les problèmes logiciels de manière à pouvoir les corriger à moindre coût. De plus, les tests sont également réalisés pour développer la confiance et la connaissance de notre produit livré.

Chaque membre de l'équipe devra être impliqué dans le processus des tests. La planification des tests, ainsi que le suivi et le contrôle des tests font partie de notre projet.

Le plan de tests se compose de 3 éléments principaux :

- Test unitaire : permet de vérifier le comportement de chaque fonction, méthodes etc. du digital twin correspondant de manière satisfaisante à ce qui a été défini lors de la phase de conception détaillée.
- Test d'intégration : permet de vérifier que les différents modules, paquetages etc. du digital twin fonctionnent correctement.
- Test fonctionnel : permet de vérifier que le digital twin recouvre bien le périmètre fonctionnel ayant été défini lors de la phase de la spécification. Le périmètre fonctionnel est défini dans le cahier des charges

Tests à effectuer

Test unitaire :

| Test 1 |
|--|
| Objectif |
| Vérifier que le script "test_recup_capteurs.py" permet d'avoir accès aux données des capteurs |
| Prérequis du test |
| <ul style="list-style-type: none">- Connaître la température et l'humidité du moment.- Le Raspberry est connecté en ssh à une machine permettant de lancer des commandes via un terminal- La carte mère est connectée au Sense Hat (carte capteur)- Le script python est installé dans le raspberry |
| Description du test |
| <ol style="list-style-type: none">1) Exécuter le script "test_recup_capteurs.py"2) Les données des capteurs affichés sur le terminal sont cohérentes avec celles du milieu où le test s'effectue3) Souffler sur le capteur4) L'humidité affichée sur le terminal augmente5) Approcher le Raspberry Pi vers une source de chaleur6) La température affichée sur le terminal augmente7) Faire descendre d'un étage le Raspberry Pi8) La pression affichée sur le terminal diminue |

Test d'intégration :

| Test 2 |
|---|
| Objectif |
| S'assurer de l'envoi des données des capteurs de la carte mère vers le serveur |
| Prérequis du test |
| <ul style="list-style-type: none">- Le MQTT est configuré entre : l'envoyeur de message (carte mère) et le receveur (serveur)- Le Raspberry est connecté en ssh à une machine permettant de lancer des commandes via un terminal |
| Description du test |
| <ol style="list-style-type: none">1) Exécuter la fonction "lien_serv_rasp.py" sur le raspberry pi2) Exécuter la fonction "lien_serv_rasp.py" sur le serveur3) Un message s'affiche sur le terminal du serveur indiquant l'heure d'envoi et "message_test_raspberry" |

| Test 3 |
|--|
| Objectif |
| Vérifier l'accès à la base de données pour que InfluxDB stocke les données |
| Prérequis du test |
| <ul style="list-style-type: none"> - InfluxDB est installé et actif sur le serveur - La carte mère est connectée au serveur et envoie des données |
| Description du test |
| <ol style="list-style-type: none"> 1) Se connecter à InfluxDB depuis le serveur Ubuntu de la machine virtuelle 2) Regarder et vérifier dans l'interface InfluxDB que les données des 3 capteurs sont envoyées vers la base de données grâce au script "RecoitCapteurs.java" 3) S'assurer du bon fonctionnement de l'archivage des données |

| Test 4 |
|--|
| Objectif |
| S'assurer du lien entre la base de données et le tableau de bord |
| Prérequis du test |
| <ul style="list-style-type: none"> - InfluxDB et Grafana fonctionnent indépendamment et effectuent leurs tâches - InfluxDB et Grafana ont été installé et configuré conformément au manuel d'installation, puis se connecter à Grafana |
| Description du test |
| <ol style="list-style-type: none"> 1) Les données affichées sur le tableau de bord sont les mêmes que celles de la base de données 2) Le temps d'envoi des nouvelles données est inférieur à 1 seconde |

Test fonctionnel :

| Test final |
|---|
| Objectif |
| S'assurer de la finalité du projet |
| Prérequis du test |
| <ul style="list-style-type: none">- Le Raspberry Pi est alimenté, allumé et connecté à internet- La couche logicielle est installée- Tous les tests antérieurs ont été validés |
| Description du test |
| <ol style="list-style-type: none">1) Lancer Grafana2) Des données sont affichées en temps réel3) Les archives des données des capteurs sont visibles.4) La manipulation des capteurs engendre des changements notables et représentatifs de la réalité |