**Description :**

Développement d’un générateur d’application web normalisée avec toute la chaine CI/CD,  une série de choix sera proposé au développeur, lui permettant de sélectionner les outils dont son application aura besoin. L’architecture se montant très rapidement, le développeur pourra dès lors se concentrer sur le métier de son application lui faisant gagner un temps précieux dans ses projets.

**Acteurs :**

Utilisateur de l’application.

**Fonctionnalités :**

* Inscription/Authentification.
* Configuration de l’application.
* Front et/ou Back
  + Connexion entre les deux
  + Invocation Rest
  + Authentification
* Front
  + Angular cli
  + Ng boosted, prime-ng
  + Maquette
  + Minimum component (login, home, router)
  + For each bdd table 🡺 x component (menu, menuitem), editable datatable, form crud
* Base de données ?
  + Table, colonne 🡺 entity, repository, service, RestController (Spring Data Rest) générés
  + IHM CRUD
* Containerisation de l’application (Docker). 🡺 Dockerfile, Docker-compose.yml
* Orchestration de l’application (CloudFoundry, Docker Swarm). Manifest.yml, docker-compose.yml
* Intégration continue (Jenkins 🡺 jenkinsfile, Gitlab CI 🡺 .gitlab-ci.yml).
* Déploiement continue (Ansible).

**Backlog de produit :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nom** | **Id user story** | **User Story** | **Priorité** | **Complexité** |
| S’inscrire/s’authentifier | 1 | En tant qu’utilisateur je dois être capable de m’authentifier. Aussi il me faut inscrire afin d’accéder aux différents services fournis par l’application. |  |  |
| Configuration de l’application à générer | 2.1 | En tant qu’utilisateur je veux gérer des applications Front End et / ou Back End sous format .Zip |  |  |
| 2.2 | En tant qu’utilisateur je veux choisir le type d’application (monolithique/micro services) |  |  |
| 2.3 | En tant qu’utilisateur je veux choisir le nom d’application |
| 2.4 | En tant qu’utilisateur je veux choisir le nom du package |
| 2.5 | En tant qu’utilisateur je veux choisir le type d’authentification |
| 2.6 | En tant qu’utilisateur je veux choisir le type de base de données (No SQL / SQL) |
| 2.7 | En tant qu’utilisateur je veux choisir la base de données de production (MySQL, Cassandra, mongo dB...) afin de ….. |
| 2.8 | En tant qu’utilisateur je veux choisir la base de données de développement |
| Génération application Front-End | 3.1 | En tant qu’utilisateur je veux avoir au moins home et login component dans mon application généré |  |  |
| 3.2 | En tant qu’utilisateur je veux avoir pour chaque table demandée un component correspond |  |  |
| Génération application Back-End | 4 | En tant qu’utilisateur je veux avoir les couches persistance, service et web générés dans mon application |  |  |
| Containerisation de l’application | 5 | En tant qu’utilisateur je veux que mon application soit conteneuriser afin d’isoler l’exécution des applications. |  |  |
| Intégration continue | 6 | En tant qu’utilisateur je veuxmon application assure l’intégration continue. |  |  |
| Déploiement continue | 7 | En tant qu’utilisateur je veux que mon application assure le déploiement continu. |  |  |

**Planification des sprints :**

**Sprint 0**

de 03/02/2020 a ../../….

**Préparer de l’environnement**

g

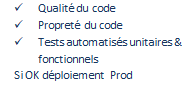
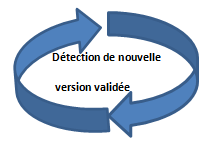
**Auto-formation**

**Production du backlog**

**Etablir l’architecture globale**

**A**

**Architecture globale : Serveur de recette**

 ** **  **Déploiement en recette / Test complet**

**Serveur d’intégration-CI**

 **Déploiement en production**

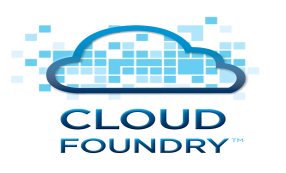
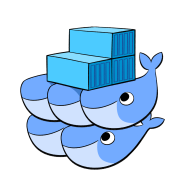
**Docker Push**

**Docker Push**

** Serveur de gestion Serveur de production**

**De configuration logicielle**

**Pull Push Orchestration**



**Développeurs**

**Intégration Continue**

**Déploiement Continu**