|  |
| --- |
| **OC PIZZA**  **Documentez votre système de gestion de pizzeria**  Dossier de conception technique  Version 1.0 |
| **Auteur**  AUBRUN Eric  *Analalyste-programmeur* |

Table des matières

1 - Versions 4

2 - Introduction 5

2.1 - Objet du document 5

2.2 - Références 5

3 - Architecture Technique 6

3.1 - Composants généraux 6

3.1.1 - Package A 6

3.1.1.1 - Composant X 6

3.1.1.2 - Composant Y 6

3.1.2 - Package B 6

3.1.2.1 - Composant Z 6

3.2 - Application Web 6

3.2.1 - Composants X 6

3.2.2 - Composants Y et Z 6

3.3 - Application XXX... 6

4 - Architecture de Déploiement 7

4.1 - Serveur de Base de données 7

4.2 - Serveur XXX 7

5 - Architecture logicielle 8

5.1 - Principes généraux 8

5.1.1 - Les couches 8

5.1.2 - Les modules 8

5.1.3 - Structure des sources 8

5.2 - Application Web 9

5.3 - Application Xxx 9

6 - Points particuliers 10

6.1 - Gestion des logs 10

6.2 - Fichiers de configuration 10

6.2.1 - Application web 10

6.2.1.1 - Datasources 10

6.2.1.2 - Fichier xxx.yyy 10

6.2.2 - Application Xxx 10

6.3 - Ressources 10

6.4 - Environnement de développement 10

6.5 - Procédure de packaging / livraison 10

6.6 - XXX 10

7 - Glossaire 11

# Versions

|  |  |  |  |
| --- | --- | --- | --- |
| Auteur | Date | Description | Version |
| Eric AUBRUN | 01/05/2021 | Création du document | XXX |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Objet du document

Le présent document constitue le dossier de conception technique de l'application XXX…

Objectif du document…

Les éléments du présents dossiers découlent :

* de …
* ...

Le présent document constitue le dossier de conception technique de l'application **OC PIZZA**. Il est destiné à l’attention des développeurs, mainteneurs et de l’équipe technique d’OC PIZZA.

L’objectif de l’analyse technique étant de lister les contraintes spécifiques dont les développeurs vont devoir tenir compte pour coder l’application, le présent document présentera les langages et les conventions de développement, les conventions de développement, l’architecture logicielle et de déploiement ainsi que les logs, le monitoring propres à l’application.

Les éléments du présent dossier découlent des documents suivants :

* Mise en place d’un nouveau système informatique pour l’ensemble des pizzerias du groupe OC Pizza
* Concevez la solution technique d’un système de gestion de pizzeria

## Références

Pour de plus amples informations, se référer également aux éléments suivants :

1. **PDOCPizza\_02\_fonctionnelle.pdf – 1.0** : Le Dossier de conception technique de l'application
2. **PDOCPizza\_03\_exploitation.pdf – 1.0** : Le Dossier d'exploitation de l’application
3. **PDOCPizza\_04\_livraison.pdf – 1.0** : Le PV de livraison finale

# Architecture Technique

## Composants généraux

### Package A

#### Composant X

Description et rôle/objectif...

#### Composant Y

### Package B

#### Composant Z

## Application Web

La pile logicielle est la suivante :

* Application **J2EE** (JDK version 1.8) / **PHP** (version) / **Python**…
* Serveur d'application **JOnAS 5.2.4 / ...**

Diagramme UML de Composants

### Composants X

Description et rôle/objectif...

### Composants Y et Z

## Application XXX...

# Architecture de Déploiement

Diagramme UML de déploiement

Explication / commentaires si nécessaires...

## Serveur de Base de données

Description

Caractéristiques techniques (ex: Serveur Linux Debian Jessie + PostgreSQL 9.6…)

Informations importantes / points particuliers

## Serveur XXX

...

# Architecture logicielle

## Principes généraux

Les sources et versions du projet sont gérées par **Git**, les dépendances et le packaging par **Apache Maven / Grunt / ...**.

...

### Les couches

L'architecture applicative est la suivante :

* unecouche **business** : responsable de la logique métier du composant
* unecouche **model** : implémentation du modèle des objets métiers
* …
  + …
* ...

### Les modules

Ex: modules Maven dans le cas d’application multi-module...

### Structure des sources

La structuration des répertoires du projet suit la logique suivante :

* les répertoires sources sont crées de façon à respecter la philosophie Maven (à savoir : « convention plutôt que configuration »)

racine  
 ├─ *pom.xml*  
 ├─ <moduleX>  
 │ ├─ *pom.xml*  
 │ └─ src  
 │ ├─ main  
 │ │ ├─ java  
 │ │ └─ resources  
 │ └─ test  
 │ ├─ java  
 │ └─ resources  
 ├─ <moduleY>  
 │ ├─ *pom.xml*  
 │ └─ src  
 │ ├─ main  
 │ │ ├─ java  
 │ │ └─ resources  
 │ └─ test  
 │ ├─ java  
 │ └─ ressources  
 └─ src  
 └─ lib

* ...

## Application Web

…

Si besoin, diagramme UML de composants pour monter les différents modules et leur inter-dépendances

## Application Xxx

…

# Points particuliers

## Gestion des logs

…

## Fichiers de configuration

### Application web

...

#### Datasources

...

#### Fichier xxx.yyy

...

### Application Xxx

...

## Ressources

...

## Environnement de développement

## Procédure de packaging / livraison

## XXX

…

# Glossaire

|  |  |
| --- | --- |
|  |  |
|  |  |