



## **OC pizza**

*Système informatique*

Dossier de conception technique

Version 1.0

**Mathieu Corroy**  
Analyste programmeur

**AndroidFuture**

<<https://www.androidfuture.com>>

43 place République, 72000, Le MANS – 02.48.36.87.08 –  
AndroidFuture@gmail.com

S.A.R.L. au capital de 1 000,00 € enregistrée au RCS de Xxx – SIREN 999 999 999 –  
Code APE : 6202A



# TABLE DES MATIERES

## Sommaire

<b>1 - Versions.....</b>	<b>3</b>
<b>2 - Introduction .....</b>	<b>4</b>
2.1 - Objet du document.....	4
2.2 - Besoins du client.....	4
2.2.1 - Enjeux et Objectifs.....	4
<b>3 - Architecture Technique .....</b>	<b>5</b>
3.1 - Diagrammes des classes.....	5
3.2 - Diagrammes des composants .....	7
3.2.1 - Composants.....	8
<b>4 - Architecture de Déploiement .....</b>	<b>9</b>
4.1 - Serveur de Base de données.....	9
4.2 - Serveur .....	9
<b>5 – Modèle Physique de Données.....</b>	<b>10</b>
<b>6 – Architecture logicielle.....</b>	<b>11</b>
4.3 - Principes généraux.....	11
<b>5 - Points particuliers.....</b>	<b>12</b>
5.1 - Environnement de développement.....	12
5.2 - Procédure de packaging / livraison.....	12



# 1 - VERSIONS

Auteur	Date	Description	Version
Mathieu Corroy	14/04/2021	Création du document	1.0



## 2 - INTRODUCTION

### 2.1 - Objet du document

Le présent document constitue le dossier de la conception technique de l'application OC Pizza.

Les éléments du présent dossier découlent :

- De différents diagrammes UML.

### 2.2 - Besoins du client

#### 2.2.1 - Enjeux et Objectifs

Le système mis en place permettra à l'entreprise une meilleure performance et un gain de temps significatif.

L'enjeu étant de faciliter l'achat pour la clientèle ainsi que la gestion des stocks et des recettes pour les employés et le responsable.

Le client aura la possibilité de se connecter au site ou de créer un compte s'il n'en dispose pas. A la création du compte, le client aura la possibilité de saisir ses informations personnelles. Une fois authentifié, il pourra passer commande sur le site, et la payer en ligne ou la modifier avant confirmation définitive de celle-ci s'il le souhaite.

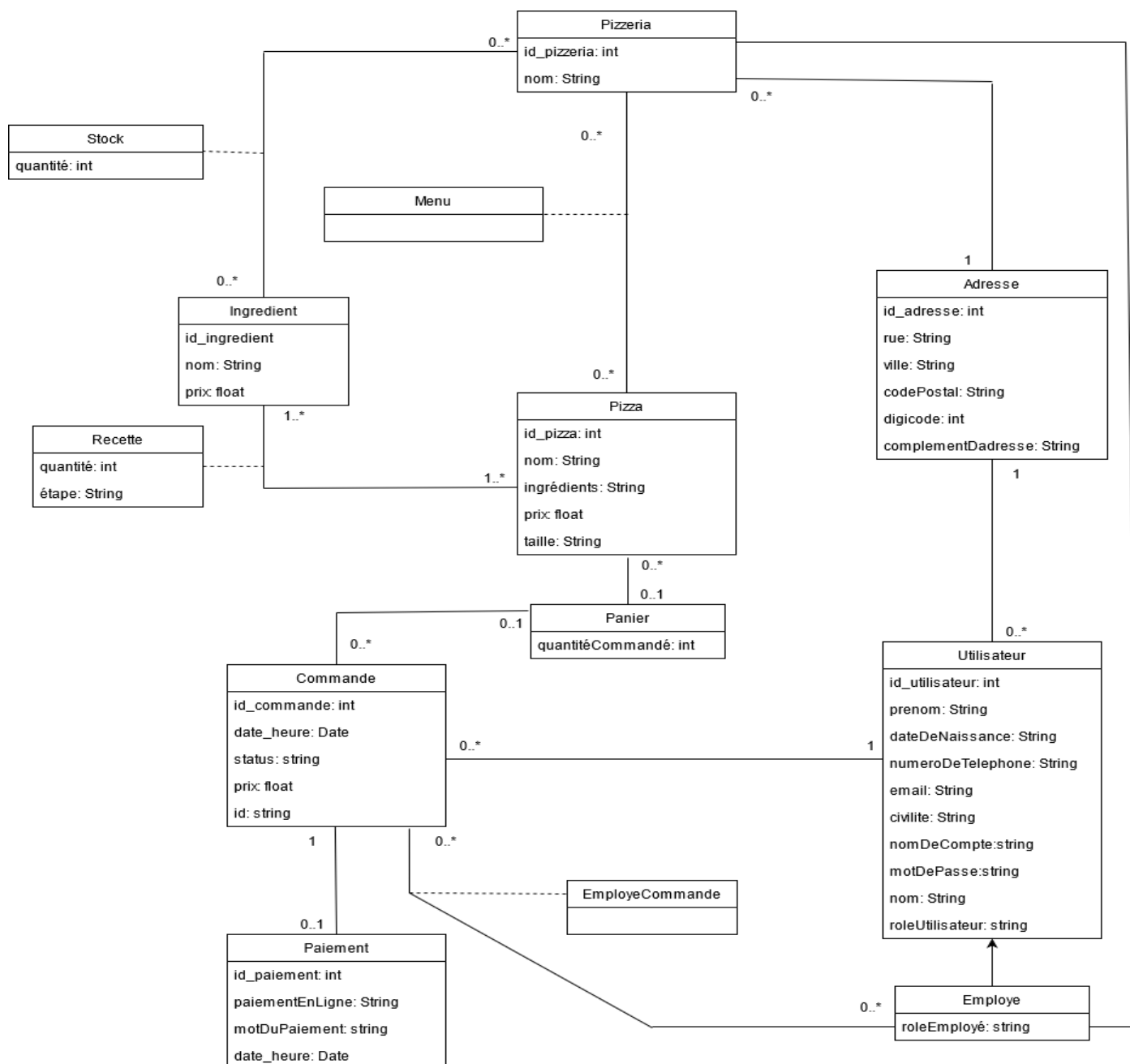
Le livreur pourra récupérer les informations de livraison du client pour lui livrer sa commande dans les meilleurs délais.

Le pizzaiolo aura accès à toutes les commandes passées et le détail de celle-ci via un système de recettes.

Le stock sera mis automatiquement à jour lorsqu'une commande sera passée pour un détail précis sur les éléments restants. Le responsable recevra cependant un mail si les disponibilités des ingrédients sont moindres ou nulles.

## 3 - ARCHITECTURE TECHNIQUE

### 3.1 - Diagrammes des classes





### 3.1.1.1 - Composant

Description, rôle :

Le diagramme de classes est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces des systèmes ainsi que leurs relations.

Une classe est le modèle abstrait d'un objet. Elle définit les attributs et les opérations de cet objet. C'est à partir de ce modèle que seront créés les objets « concrets » possédant des valeurs particulières pour leurs attributs.

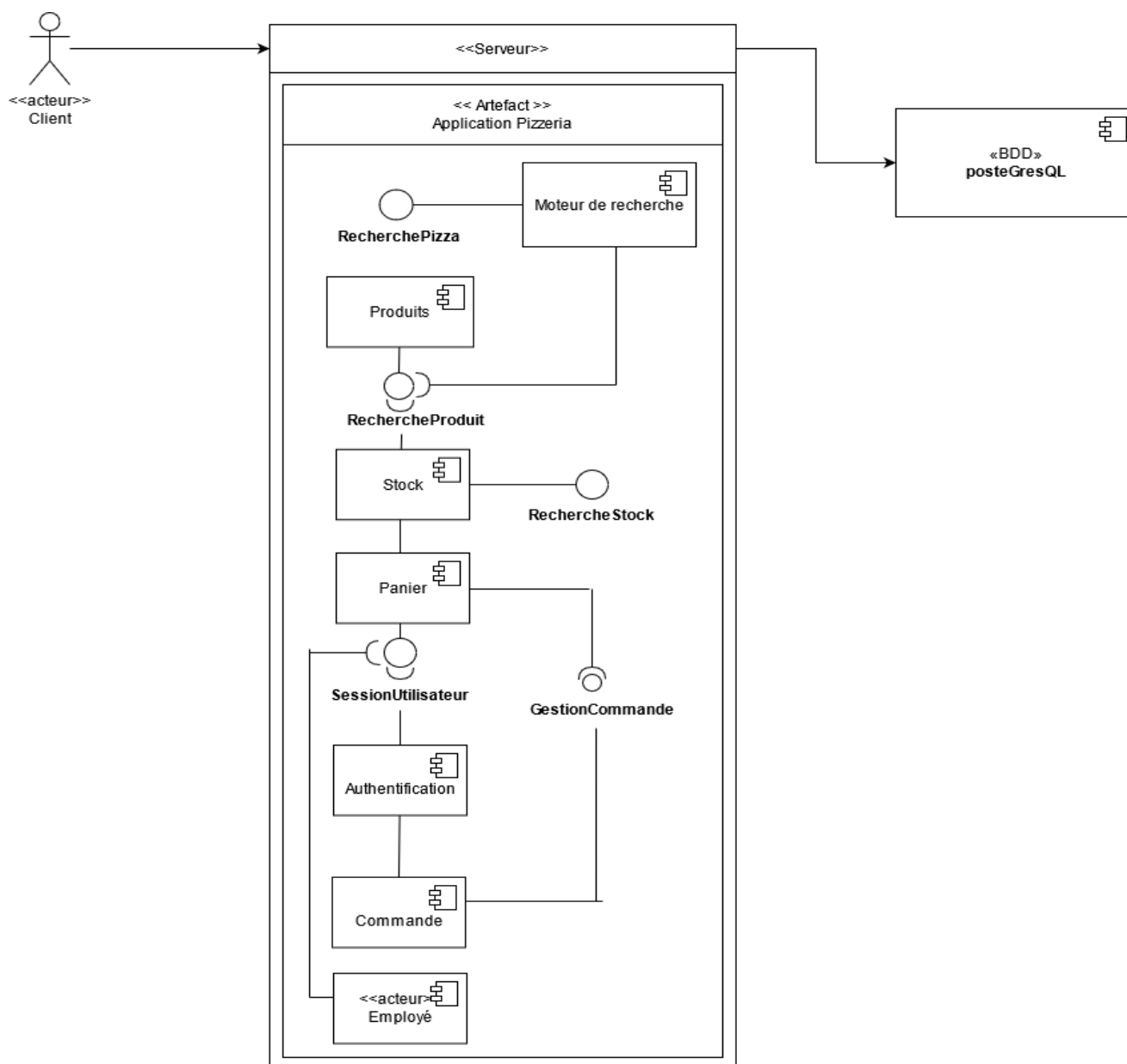
Dans un diagramme de classes, une classe est représentée par une boîte rectangulaire, comprenant plusieurs parties séparées par des traits :

- Dans la partie haute : le **nom** de la classe
- Dans la partie en dessous : la liste des **attributs**.

En UML, la relation entre deux classes s'appelle une **association**.

Les **multiplicités** servent à apposer des contraintes numériques sur l'association, ou plus précisément, combien d'instances d'une classe peuvent être liées à une instance de la classe de l'autre côté de l'association. En d'autres termes, elles bornent les **cardinalités** des instances liées entre elles.

## 3.2 - Diagrammes des composants





### 3.2.1 - Composants

Description, rôle :

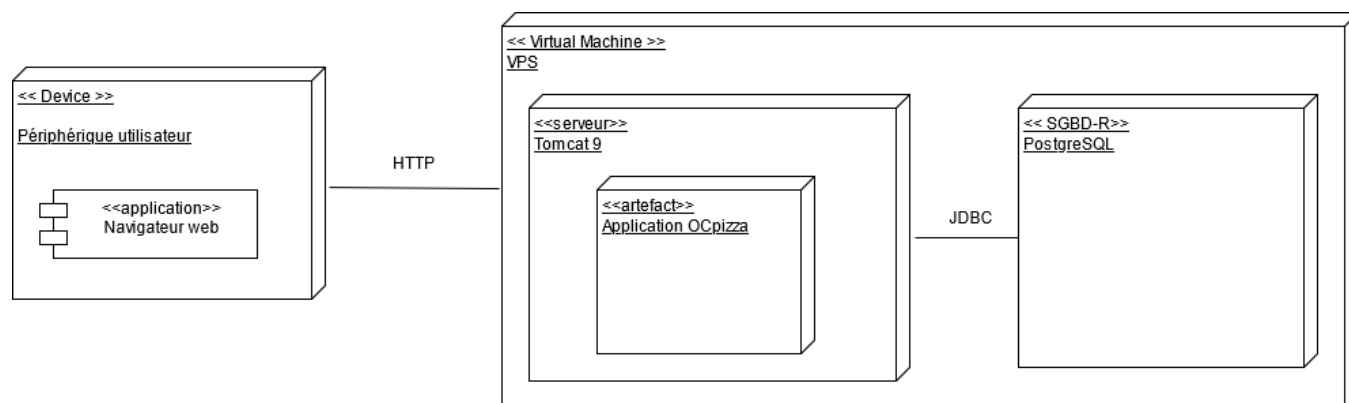
Un diagramme de composants a pour objectif d'illustrer la relation entre les différents composants d'un système. Dans le cadre de l'UML 2.0, le terme « composant » fait référence à un module de classes qui représentent des systèmes ou des sous-systèmes indépendants ayant la capacité de s'interfacer avec le reste du système.

Ils sont d'une valeur inestimable pour la conception du système. Ils peuvent en effet aider à :

- Imaginer la structure physique du système ;
- Faire ressortir les composants du système et à leurs relations ;
- Mettre en évidence le comportement de chaque service vis-à-vis de l'interface



## 4 - ARCHITECTURE DE DEPLOIEMENT



Description, rôle :

Un **diagramme de déploiement** est une vue statique qui sert à représenter l'utilisation de l'infrastructure physique par le système et la manière dont les **composants** du système sont répartis ainsi que leurs relations entre eux. Les éléments utilisés par un **diagramme de déploiement** sont principalement les **nœuds**, les **composants**, les **associations** et les **artefacts**.

### 4.1 - Serveur de Base de données

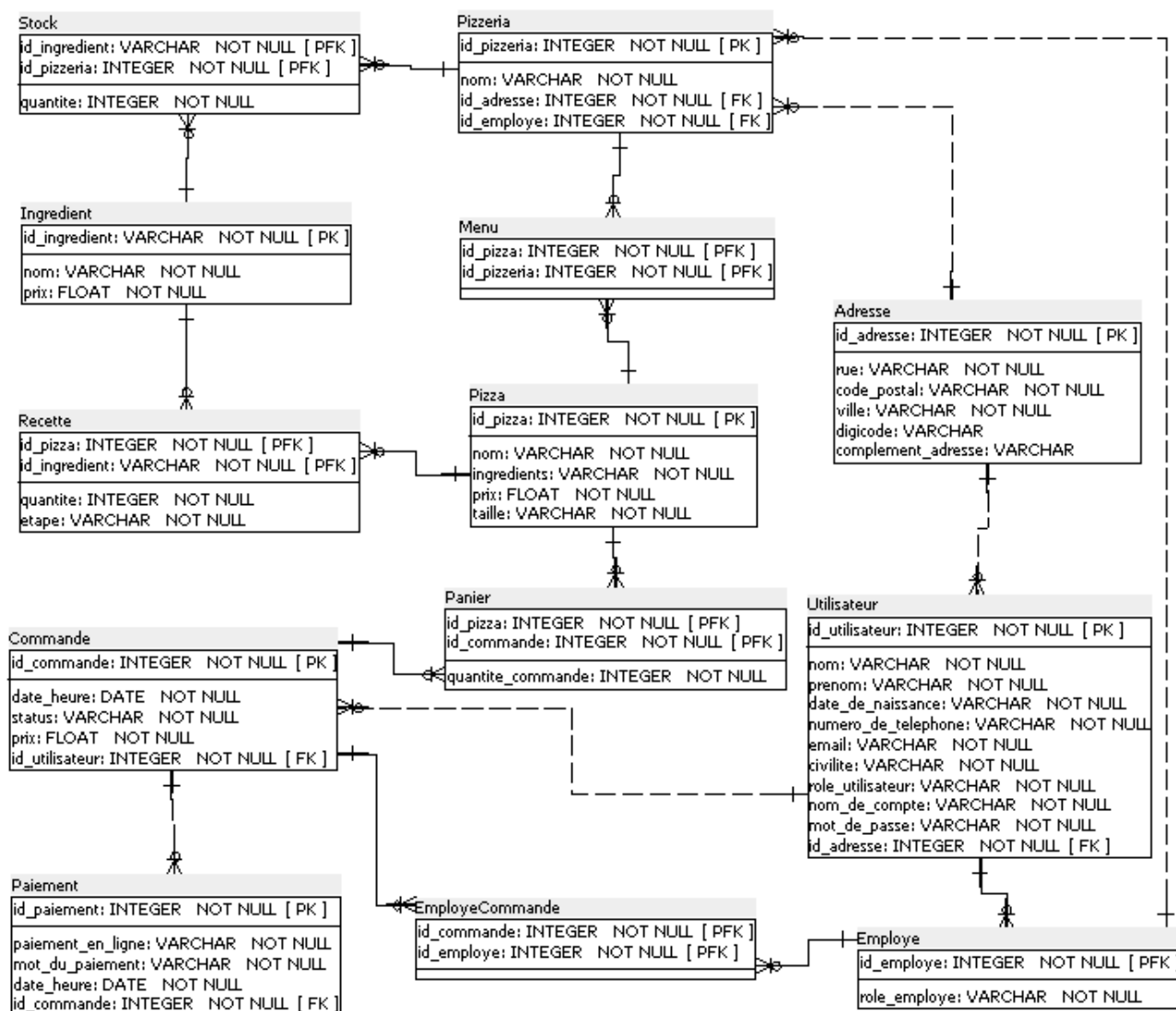
Caractéristiques techniques :

Base de données : PostgreSQL 13.2-1

### 4.2 - Serveur

VPS (Serveur dédié virtuelle) (ex : A2 hosting)

## 5 – MODELE PHYSIQUE DE DONNEES





## 6 – ARCHITECTURE LOGICIELLE

### 4.3 - Principes généraux

Les sources et versions du projet sont gérées par **Git**



## 5 - POINTS PARTICULIERS

### 5.1 - Environnement de développement

- VPS - A2 Hosting
- Web – Tomcat 9
- SGBDR – PostgreSQL 13.2-1

### 5.2 - Procédure de packaging / livraison

#### Déploiement d'une application sous Tomcat

- Tomcat, comme tout moteur de servlet (cela respecte les spécifications) accepte une [archive WAR](#) comme application web. Celle-ci devra bien sûr respecter la [structure d'une webapp](#). Ce fichier doit être déposé dans le répertoire appBase de l'[Host](#) sur lequel l'application est installée. En règle générale, ce répertoire est TOMCAT\_HOME/webapps. Au prochain démarrage de Tomcat, il détectera le nouveau fichier WAR, et [créera automatiquement](#) un [Context](#) correspondant dans le [fichier de configuration](#). L'application sera dès lors accessible. De plus, en règle générale (sauf [configuration contraire](#)), le WAR sera déployé, c'est-à-dire que l'arborescence de fichiers et répertoires qu'il contient sera décompactée dans un répertoire portant le même nom que le WAR.

#### Documentation:

- <http://www-igm.univ-mlv.fr/~dr/XPOSE2003/tomcat/tomcat.php?rub=17>

#### Autodéploiement de Tomcat

Si, dans le [fichier de configuration](#) de Tomcat, dans une balise [Host](#), vous avez défini l'attribut **autoDeploy** à *true* (ou que vous ne l'avez pas spécifié, *true* étant sa valeur par défaut), le déploiement automatiquement d'applications aura lieu. Il se comporte de la manière suivante :

- Les archives WAR (Web Application Archives) non déployées sont tous les fichiers \*.war du répertoire **appBase** (voir configuration de [Host](#)) pour lesquels il n'existe pas de répertoire portant le même nom dans ce même répertoire. Exemple : si un fichier "**monappli.war**" existe, mais qu'il n'y a pas de répertoire "**monappli**" dans **appBase**, l'application est considérée comme non déployée. Dans ce cas, le WAR sera automatiquement déployé, dans un répertoire portant le même nom. Attention toutefois, vous pouvez empêcher cette



procédure en définissant le paramètre **unpackWARs** à *false* (voir configuration de **Host**).

Donc, si vous souhaitez mettre à jour un WAR, n'oubliez pas de supprimer le répertoire associé pour bénéficier du déploiement automatique au prochain démarrage du serveur.

- En ce qui concerne les applications non compactées (un répertoire est présent dans **appBase**, sans fichier WAR portant le même nom), ils se verront affecter un **Context** automatique dans le fichier **server.xml**. Ce contexte sera configuré en fonction du **DefaultContext** éventuellement spécifié pour le **Host** associé. Le chemin pour ce contexte (c'est-à-dire son URL relative) sera le nom du répertoire. C'est-à-dire que, si Tomcat trouve un répertoire nommé **monAppli** contenant un fichier **web.xml** (une webapp donc), il lui fournira automatiquement le contexte **/monAppli**. On pourra ainsi accéder à cette application.
- Dans le répertoire **appBase** d'un **Host**, on peut aussi trouver des fichiers de configuration de Context. C'est-à-dire des fichiers XML contenant comme élément de base un **Context**, et tous les sous-éléments que l'on peut souhaiter. Il s'agit en fait ici de délocaliser les informations de contexte d'une application. Ce qui s'avère intéressant, car ces fichiers peuvent être pris en compte (et les applications correspondantes déployées) sans que le serveur redémarre. Alors que toute modification du fichier **server.xml** nécessite un redémarrage de Tomcat pour être effective.

Notons enfin que ce déploiement automatique peut également intervenir pendant l'exécution de Tomcat. En paramétrant **liveDeploy** à *true* dans un **Host**, les applications web (WAR, répertoires) ajoutées pendant l'exécution de Tomcat seront elles aussi déployées automatiquement.

#### Documentation:

- <http://www-igm.univ-mlv.fr/~dr/XPOSE2003/tomcat/tomcat.php?rub=6>

#### Base de données

##### Commande de création de base de données:

- `$ CREATE DATABASE nom_base OWNER nom_role;`

##### Créer une base de données avec pgAdmin4 :

#### AndroidFuture

<<https://www.androidfuture.com>>

43 place République, 72000, Le MANS – 02.48.36.87.08 –

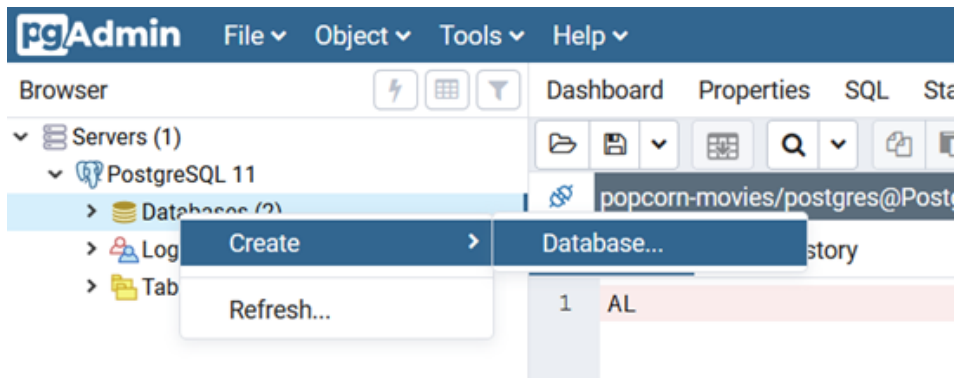
AndroidFuture@gmail.com

S.A.R.L. au capital de 1 000,00 € enregistrée au RCS de Xxxx – SIREN 999 999 999 –

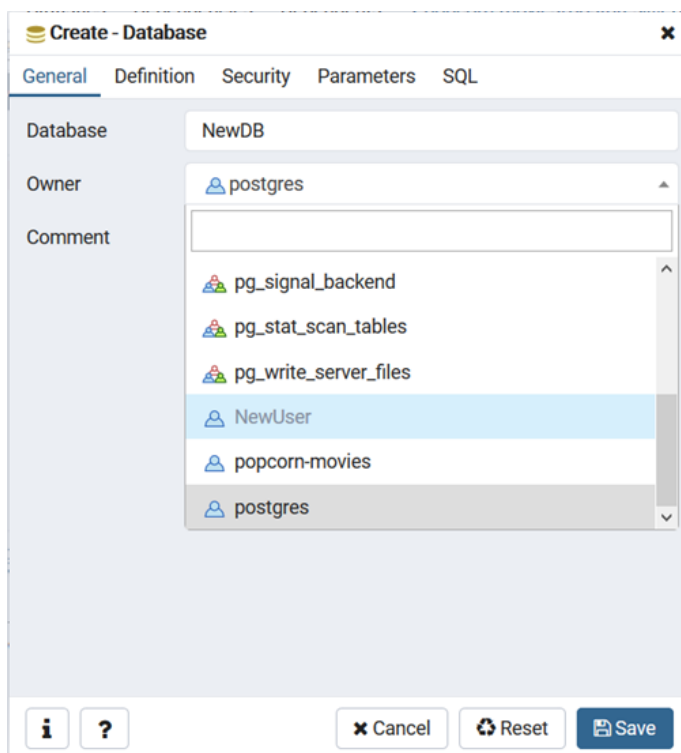
Code APE : 6202A



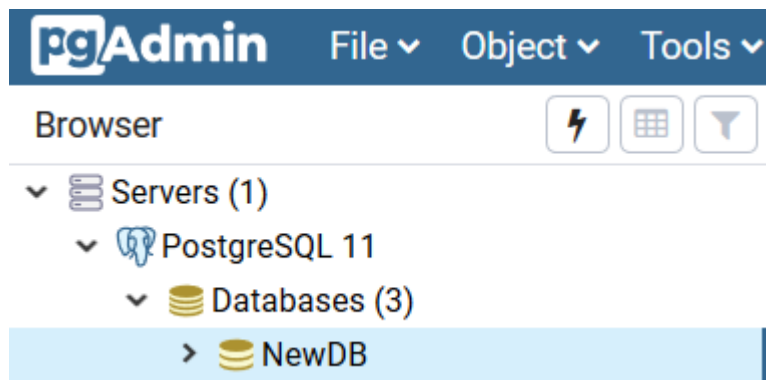
- Faites un clic droit sur « Databases » puis sélectionnez « Create > Database ».



- Renseignez-le nom de votre base de données ainsi que le compte propriétaire.



- Cliquez ensuite sur « Save ».
- Si tout s'est bien passé, la base de données va apparaître dans la liste des bases de données.



- Vous disposez à présent d'une base de données qui a pour propriétaire l'utilisateur souhaité.

#### Documentation:

- <https://apero-tech.fr/creer-un-utilisateur-et-une-base-de-donnees-postgresql-dont-il-sera-propretaire-via-linterface-pgadmin-4-sous-windows/#:~:text=III.-,Cr%C3%A9ation%20de%20la%20base%20de%20donn%C3%A9es,Cliquez%20ensuite%20sur%20%C2%AB%20Save%20%C2%BB.>