



# Cahier Des Charges Technique

Cette présentation détaille l'organisation et le fonctionnement de la plateforme de livraison de paniers alimentaires, en mettant l'accent sur les périmètres du projet, les besoins fonctionnels et techniques, ainsi que l'environnement technique mis en place.

**Eliot DUBREUIL et Mehdi ERRAES**

**15 Octobre 2025**

# Périmètres du projet

- **Gestion des dépôts**

Création, consultation, modification et suppression (CRUD)

- **Affichage des cartes**

Via Leaflet avec position des dépôts

- **Gestion des tournées**

Affichage de la liste, détail d'une tournée, et visualisation de l'itinéraire entre les dépôts

- **Mise en place des structures**

Pour les adhésions et abonnements (préparées dans la base de données)

- **Interface d'adhésion**

Formulaire d'inscription/connexion prêt

- **Gestion des données**

Via PostgreSQL et exécution de l'application via Docker

# Besoins fonctionnels

- **Consulter la liste des dépôts disponibles**

L'utilisateur doit pouvoir accéder à la liste des dépôts où il peut récupérer ses paniers

- **Visualiser chaque dépôt sur une carte interactive**

L'utilisateur doit pouvoir localiser les dépôts sur une carte Leaflet et obtenir des informations détaillées

- **Consulter la liste des tournées et le détail de chaque tournée**

L'utilisateur doit pouvoir voir les différentes tournées de livraison et accéder aux détails d'une tournée spécifique

- **Afficher l'itinéraire complet d'une tournée**

La carte d'une tournée doit afficher l'ordre de passage des dépôts pour cette tournée

- **S'inscrire et se connecter au système**

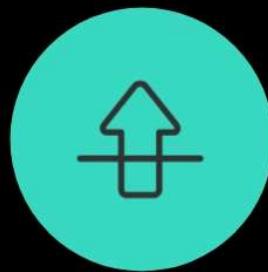
Le système doit proposer un formulaire d'adhésion (inscription/connexion) avec des mots de passe sécurisés

# Besoins non fonctionnels



## Performance

Chargement fluide des cartes et des itinéraires



## Évolutivité

Possibilité d'ajouter facilement de nouvelles fonctionnalités (abonnements, calendrier, gestion des paniers)



## Portabilité

Fonctionnement identique sur Windows, macOS et Linux via Docker



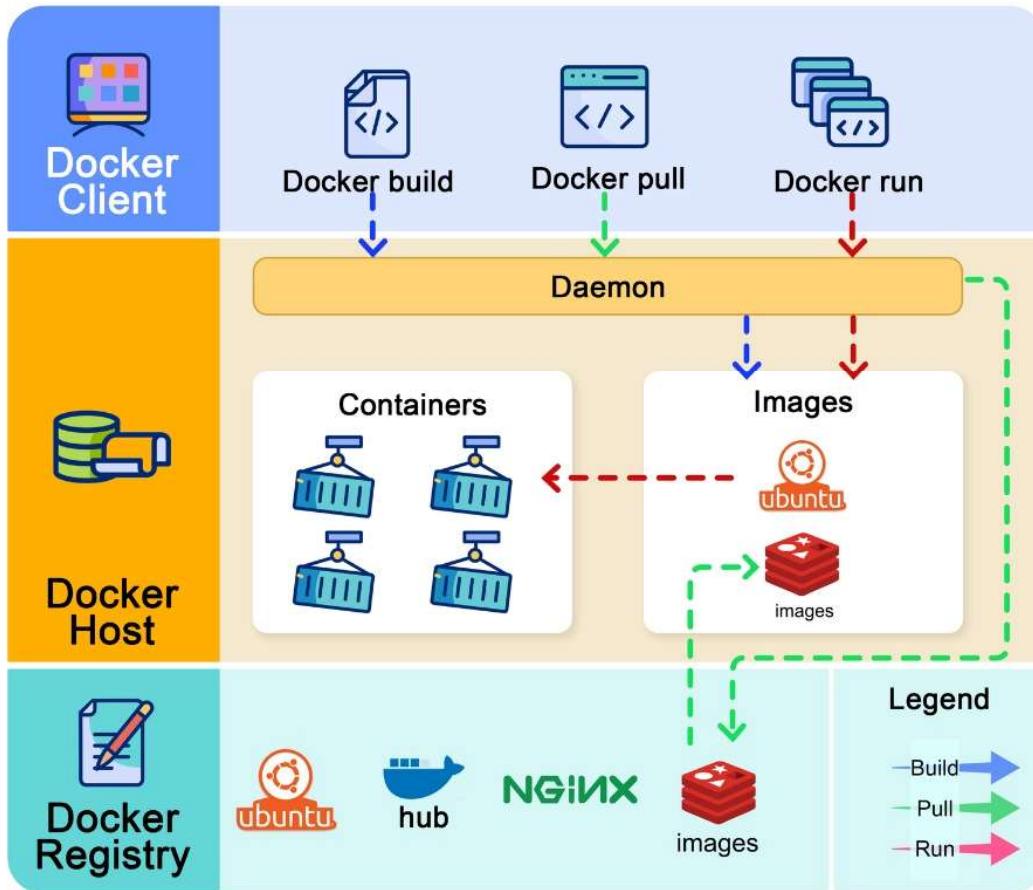
## Simplicité

Code clair, modulaire, facilement maintenable

Cet accent mis sur les besoins non fonctionnels garantit une application performante, évolutive, portable et facile à maintenir, offrant ainsi une meilleure expérience utilisateur et une plateforme solide pour le développement futur.

# How does Docker Work ?

 blog.bytebytego.com



## Architecture générale

L'application est hébergée dans un environnement conteneurisé géré par Docker Compose, permettant une mise en place simple, portable et réproductible du système. Cette architecture composée de deux services interconnectés - Apache-PHP pour l'exécution du site et PostgreSQL pour le stockage des données - assure une infrastructure sécurisée et évolutive.

# Technologies utilisées

## Front-end

HTML, CSS, JavaScript,  
Bootstrap, AOS (animations),  
Font Awesome

## Cartographie

Leaflet, Leaflet Routine  
Machine

## Back-end

PHP

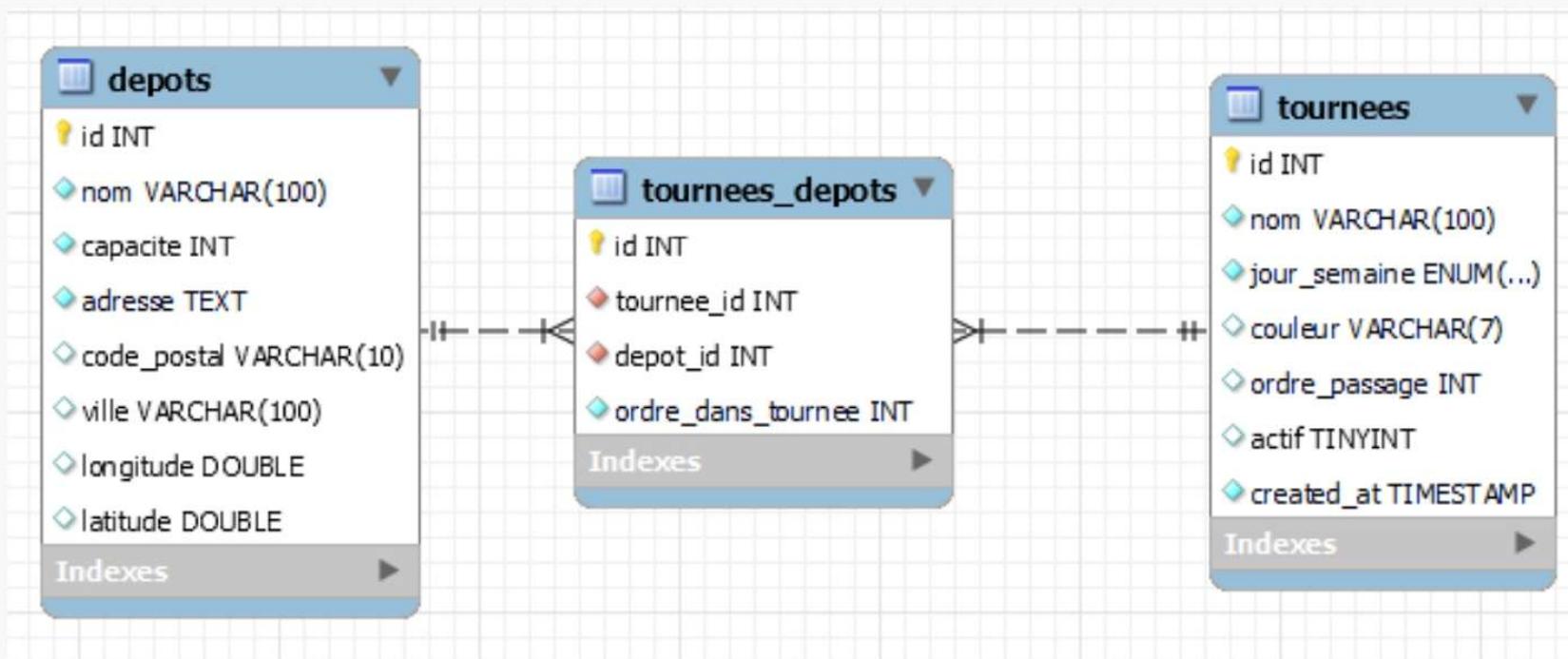
## Base de données

PostgreSQL

## Infrastructure

Docker

# Modèle relationnel



# Solutions

## La solution d'hébergement

L'application est entièrement conteneurisée avec Docker Compose, permettant une mise en place simple, portable et reproductible du système. Cette infrastructure sécurisée et évolutive se compose de deux services interconnectés : Apache-PHP pour l'exécution du site et PostgreSQL pour le stockage des données.

## La solution pour les tests

Le système doit proposer des tests unitaires et d'intégration pour garantir la fiabilité et la stabilité de l'application. Ces tests automatisés couvriront les fonctionnalités clés, les interactions avec la base de données et les aspects de sécurité, comme la protection contre les injections et le XSS.

# Organisation du projet



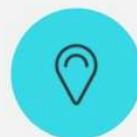
## Code source PHP, fichiers HTML/CSS/JS

Le répertoire App/apache-php/ contient le code source de l'application, ainsi que les fichiers front-end (HTML, CSS, JavaScript).



## Scripts SQL d'initialisation de la base

Le répertoire App/db-init/ contient les scripts SQL permettant de créer les tables de la base de données et d'insérer les données de démonstration.



## Fichier d'orchestration Docker

Le fichier App/docker-compose.yml définit l'orchestration des services Docker (apache-php et bd-postgre) pour le lancement de l'application.



## Configuration de la connexion à la base

Le fichier App/config/db.php contient la configuration de la connexion à la base de données, en utilisant les variables d'environnement définies dans le fichier .env à la racine du projet.

Cette organisation des répertoires et fichiers permet de structurer le projet de manière modulaire et facilite la maintenance et l'évolutivité de l'application.

# Variables d'environnement

Nom de la variable	Valeur
DB_USER	utilisateur_bdd
DB_PASS	mot_de_passe_bdd
DB_NAME	nom_base_donnees

# Présentation de l'équipe



**ERRAES Mehdi**

Développeur Front-end



**DUBREUIL Eliot**

Développeur Back-end

**En résumé, ce cahier des charges technique détaille l'organisation et le fonctionnement d'une plateforme de livraison de paniers alimentaires. Il met l'accent sur les périmètres du projet, les besoins fonctionnels et techniques, ainsi que l'environnement technique mis en place. Cette présentation fournit une vue d'ensemble du projet et servira de base pour la suite du développement.**

