Metz Numeric School

Fiche missions
Concepteur Développeur d'Applications

Contexte

Dans le cadre du passage du titre professionnel, **Concepteur Développeur d'Application,** voici des exemples de missions que les apprenants peuvent réaliser durant leur stage en entreprise (d'une durée de 8 semaines) basés sur le référentiel d'évaluation et sur 4 technologies majeures.

C# / ASP.NET Core - **Java / Spring Boot** - **Python / Django ou FastAPI** - **PHP / Symfony ou Laravel**

Missions

1. Installer et configurer son environnement de travail en fonction du projet

- **C#** : Installation de Visual Studio, .NET SDK, SQL Server, Docker, Git.
- **Java** : Installation de IntelliJ, Java 17, Spring Boot, Maven, PostgreSQL, Docker.
- **Python** : Configuration de l'environnement virtuel, Django ou FastAPI, PostgreSQL, Docker Compose.
- **PHP** : Installation de Symfony/Laravel, PHP 8.2, Composer, MySQL, Docker avec volumes.

2. Développer des interfaces utilisateur

- **C#** : Razor Pages ou Blazor pour créer des vues dynamiques d'une application interne.
- **Java** : Thymeleaf dans Spring Boot ou développement d'une interface Angular connectée à l'API.
- **Python** : Templates Django avec Bootstrap ou React connecté à une API Django REST.
- **PHP** : Intégration Twig dans Symfony, Blade pour Laravel, ou création d'un front React + API REST.

3. Développer des composants métier

- **C#** : Implémenter une couche Service pour la logique métier (ex. calcul de prix ou remises).
- **Java** : Services Spring pour gérer la logique métier (commandes, clients, paiements).
- **Python** : Business logic dans Django services ou FastAPI modules.
- **PHP** : Créer des services métiers dans Symfony (avec services configurés) ou Laravel (via Service Providers).

4. Contribuer à la gestion d'un projet informatique

- **Tous** : Suivi des tâches avec Jira, GitHub Projects ou GitLab ; planification des tâches, rédaction de comptes-rendus ; participation aux daily meetings.

5. Analyser les besoins et maquetter une application

- **Tous** : Échange avec les utilisateurs, création de user stories, réalisation de maquettes sur Figma ou Adobe XD, rédaction du dossier de conception.

6. Définir l'architecture logicielle d'une application

- **C#** : Architecture MVC ou Clean Architecture avec DI dans ASP.NET Core.
- **Java** : Architecture en couches (Controller / Service / Repository), patrons de conception, Spring Security.
- **Python** : Architecture Django modulaire ou FastAPI Clean Architecture.
- **PHP** : Architecture Symfony hexagonale (Domain/Infrastructure), ou Laravel (MVC + Services).

7. Concevoir et mettre en place une base de données relationnelle

- **C#** : EF Core pour SQL Server, code-first ou database-first, migration des schémas.
- **Java** : JPA/Hibernate avec PostgreSQL ou MySQL, mapping d'entités.
- **Python** : ORM Django ou SQLAlchemy avec migration via Alembic.
- **PHP** : Doctrine ORM (Symfony) ou Eloquent ORM (Laravel), migration avec CLI.

8. Développer des composants d'accès aux données SQL et NoSQL

- **C#** : Requêtes LINQ, gestion des relations EF, sécurité des accès.
- **Java** : Spring Data JPA ou MongoDB avec Spring Data Mongo.
- **Python** : Django ORM (queryset, manager), ou Pydantic + MongoDB avec motor.

- **PHP** : Repository Symfony avec Doctrine, ou Eloquent ORM dans Laravel, requêtes sécurisées.

9. Préparer et exécuter les plans de tests d'une application

- **C#** : xUnit ou NUnit pour tests unitaires, analyse de couverture, tests de sécurité.
- **Java** : JUnit, Mockito, tests de performance, rapport via JaCoCo.
- **Python** : PyTest ou unittest, tests de charge (Locust), analyse Bandit.
- **PHP** : PHPUnit, tests de sécurité avec Symfony Security Checker ou Psalm.

10. Préparer et documenter le déploiement d'une application

- **C#** : Script de déploiement sur Azure App Service, Dockerfile pour build, documentation YAML Azure DevOps.
- **Java** : Déploiement sur Tomcat, Docker ou Heroku ; configuration CI/CD GitLab.
- **Python** : Déploiement Django sur VPS (Gunicorn + Nginx), avec scripts Fabric ou Ansible.
- **PHP** : Déploiement Symfony/Laravel sur serveur Apache ou Nginx, Docker + documentation, outils type Deployer.

11. Contribuer à la mise en production dans une démarche DevOps

- **C#** : Intégration continue via Azure Pipelines, GitHub Actions, SonarCloud.
- **Java** : CI/CD GitLab avec Maven + Docker, Jenkins pour pipeline automatisé.
- **Python** : GitHub Actions, GitLab CI pour tests, build, déploiement Docker.
- **PHP** : Intégration continue Symfony/Laravel avec GitHub Actions ou GitLab CI, analyse de code avec PHPStan ou Psalm.