

Rapport de Stage

Remerciements

Je tiens tout d'abord à remercier chaleureusement Monsieur Cedric Pluta, mon maître de stage chez Sopra Steria, pour son accompagnement, sa patience, et ses conseils précieux tout au long de ces deux mois. Son encadrement attentif et son expertise ont été essentiels pour mon intégration au sein de l'équipe NFS et pour la réussite des missions qui m'ont été confiées.

Je souhaite également adresser mes remerciements à toute l'équipe NFS pour leur accueil chaleureux, leur soutien constant, et les échanges enrichissants qui ont marqué cette expérience. Leur disponibilité et leur collaboration ont grandement facilité mon apprentissage.

Je remercie tout particulièrement Sopra Steria, pour m'avoir offert l'opportunité de réaliser ce stage dans un cadre professionnel exigeant et stimulant. Cette expérience m'a permis de découvrir le fonctionnement d'une grande entreprise innovante, ainsi que l'importance des bonnes pratiques en matière de sécurité et de développement.

Enfin, je tiens à remercier Madame Stéphanie Delpeyroux et Monsieur Philippe Leproux, responsables de l'UE, pour leur suivi attentif et leurs conseils avisés tout au long de cette formation.

Je suis reconnaissante envers toutes ces personnes qui ont contribué à faire de ce stage une étape déterminante dans mon parcours professionnel.

Introduction

Le secteur numérique, en constante évolution, impose aux entreprises de s'adapter aux nouvelles technologies pour offrir des services toujours plus performants. Dans ce contexte dynamique, mon stage chez Sopra Steria a représenté une véritable opportunité d'apprentissage et de mise en pratique de mes compétences dans un environnement professionnel exigeant.

Intégrée au sein du service NFS (Network Facing System), dans le département Énergie et Télécom, j'ai participé à plusieurs projets visant à maintenir et améliorer des applications internes utilisées par Orange Business. Ces outils sont essentiels à la gestion des équipements réseau du client, un enjeu stratégique pour la performance de ses infrastructures.

Ce stage m'a permis d'apprendre de nouvelles technologies comme **Angular**, **Java** et **TypeScript**, tout en consolidant et en mettant en œuvre des compétences que je possédais déjà, notamment en **HTML**, **CSS**, **JavaScript**, ou encore dans l'utilisation d'**environnements virtualisés**. Ma mission principale a porté sur la création d'une page web destinée à visualiser les taux de conformité issus d'audits d'accessibilité, contribuant ainsi à l'amélioration continue des services proposés.

Le présent rapport s'articule autour de trois grandes parties : la première présente l'environnement économique du stage, en décrivant le secteur d'activité de Sopra Steria ainsi que l'entreprise elle-même ; la seconde détaille le cadre du stage, le fonctionnement de l'équipe NFS et les outils mobilisés ; la troisième partie revient sur les missions réalisées, les compétences développées et les leçons tirées de cette expérience professionnelle.

DÉVELOPPEMENT

I. L'environnement économique du stage

A. Le secteur des services numériques (ESN)

Les Entreprises de Services du Numérique (ESN) jouent un rôle central dans la transformation digitale des organisations, en leur fournissant des solutions technologiques adaptées aux enjeux actuels. En France, ce secteur connaît une croissance soutenue, portée par plusieurs facteurs clés :

- **Cybersécurité** : Face à l'augmentation des cyberattaques et aux nouvelles réglementations européennes telles que NIS2 et DORA, les entreprises renforcent leurs dispositifs de sécurité numérique pour protéger leurs données et infrastructures critiques.
- **Intelligence Artificielle (IA)** : L'IA transforme les processus métiers, avec une projection de doublement du marché mondial de l'IA d'ici 2028, atteignant 1 270 milliards de dollars .
- **Formation et compétences** : Le secteur fait face à une pénurie de talents, notamment dans les domaines de l'IA et de la cybersécurité, nécessitant des initiatives de formation pour combler ce déficit.

Ces dynamiques positionnent les ESN comme des partenaires stratégiques pour accompagner les entreprises dans leur transition numérique, en leur offrant des services allant du conseil à l'intégration de solutions technologiques avancées.

B. L'entreprise dans son secteur : Sopra Steria

Sopra Steria est un acteur majeur européen des services numériques, issu de la fusion en 2015 de deux entreprises françaises : Sopra (fondée en 1968) et Steria (fondée en 1969). Avec 51 000 collaborateurs répartis dans près de 30 pays, le groupe a réalisé un chiffre d'affaires de 5,8 milliards d'euros en 2024 .

Le groupe se positionne comme un partenaire stratégique pour ses clients, en les accompagnant dans leur transformation digitale à travers une offre complète de services : conseil, intégration de systèmes, édition de solutions métiers et services d'infrastructure.

Sopra Steria intervient dans divers secteurs d'activité, notamment :

- **Secteur public** : 24 % du chiffre d'affaires.
- **Services financiers** : 21 %.
- **Aéronautique, défense et sécurité** : 20 %
- **Énergie et télécommunications** : 10 %.

L'entreprise investit également dans des domaines d'avenir tels que l'intelligence artificielle, à travers sa filiale Sopra Steria Next, qui projette une croissance significative du marché de l'IA dans les années à venir .

En intégrant Sopra Steria pour mon stage, j'ai eu l'opportunité de m'immerger dans un environnement dynamique, au cœur des enjeux technologiques actuels, et de contribuer à des projets alignés avec les tendances majeures du secteur numérique.

II. Le cadre du stage

1. Organisation interne de l'entreprise

Sopra Steria adopte une organisation matricielle structurée en départements spécialisés par secteur d'activité, afin de répondre au mieux aux besoins spécifiques de ses clients. Ces départements couvrent des domaines variés tels que l'aéronautique, la défense, les télécommunications, l'énergie, la finance ou encore les administrations publiques.

J'ai effectué mon stage au sein du **département Énergie & Télécom**, et plus précisément dans le **service NFS (Network Facing System)**, qui travaille pour le client **Orange Business**. Ce service est chargé de concevoir, maintenir et faire évoluer plusieurs **applications internes critiques** pour la gestion des infrastructures réseau du client. Ces applications couvrent des domaines variés comme la collecte de données, le diagnostic réseau, la gestion des OS, ou encore la sécurité d'accès aux machines virtuelles.

Parmi les applications gérées par le service NFS, on peut citer :

- **IPCFM (Data Collect)** : application chargée de collecter automatiquement les configurations des routeurs.
- **NFS-IWA / API** : interface utilisateur permettant d'effectuer des diagnostics, des téléchargements ou des envois d'images système.
- **IPToolBox** : ensemble d'outils destinés au diagnostic et à la visualisation de chemins réseau.
- **Secure Gateway** : application permettant d'envoyer des commandes vers des machines virtuelles avec vérification des droits d'accès via une API intermédiaire.
- **OS Repository** : référentiel centralisé pour la gestion et le déploiement des systèmes d'exploitation.

Ces outils sont utilisés exclusivement en interne par les collaborateurs d'Orange Business pour superviser, auditer et intervenir efficacement sur leur réseau.

2. Fonctionnement de l'équipe / département

Le service NFS est réparti sur **trois sites géographiques** : **Rennes, Sophia Antipolis et Montpellier**, ce qui implique un **travail en collaboration à distance** et une coordination importante entre les membres des différentes équipes.

- À **Rennes**, on retrouve principalement deux **Business Analysts**.
- À **Sophia Antipolis**, l'équipe est composée d'un **responsable technique**, d'un **chef de projet**, d'un **Business Analyst** et de **quatre développeurs**.
- À **Montpellier**, où j'ai effectué mon stage, se trouvent **quatre développeurs** ainsi qu'un **Product Owner (PO)**, qui fait le lien entre le client et l'équipe technique.

L'équipe suit un fonctionnement **agile basé sur le cadre Scrum**, avec des cycles de développement courts appelés **sprints**. Ce modèle permet une meilleure réactivité face aux demandes du client et facilite l'intégration continue des évolutions fonctionnelles.

Les rituels agiles observés incluent :

- **Daily meetings** : réunions journalières de 15 minutes pour faire le point sur les tâches en cours, identifier les obstacles et adapter les priorités.
- **Sprint Planning** : séance de planification à chaque début de sprint pour définir les objectifs et répartir les tâches à partir du backlog.
- **Backlog Refinement** : moment dédié à la clarification et à l'estimation des tâches futures selon les critères **DoR (Definition of Ready)** et **DoD (Definition of Done)**.
- **Demos et Rétrospectives** : à la fin de chaque sprint, une démonstration du travail réalisé est présentée à l'équipe, suivie d'une rétrospective pour améliorer les pratiques collectives.

L'équipe participe également à des **réunions de coordination inter-services**, comme les **IAM (Information Agile Meetings)**, qui permettent de partager l'avancement des projets à l'échelle du service NFS.

3. Culture d'entreprise et outils utilisés

Sopra Steria cultive une **culture d'entreprise centrée sur l'agilité, la collaboration et l'innovation continue**. Les valeurs clés mises en avant sont :

- la **valeur client**,
- l'**esprit d'expérimentation**,
- la **transversalité dans les échanges**,
- le **leadership collaboratif**,
- et une **vision stratégique agile** partagée à tous les niveaux de l'organisation.

Pour faciliter cette organisation, plusieurs outils techniques et collaboratifs sont utilisés quotidiennement :

- **Jira** : plateforme de gestion de projet agile, permettant de suivre l'évolution des tickets, de planifier les sprints et de centraliser les tâches de développement.
- **GitLab** : outil de gestion de versions du code, utilisé aussi pour les pipelines d'intégration et de déploiement continu (**CI/CD**).
- **Poker Planning** : méthode d'estimation collective pour évaluer la charge de travail d'une tâche, utilisée durant les backlog refinements.
- **Docker et Kubernetes** : technologies de conteneurisation et d'orchestration utilisées dans certains projets pour assurer le déploiement automatisé et la scalabilité des applications.

Ces outils et cette organisation agile m'ont permis de m'intégrer rapidement dans l'équipe et de m'adapter au rythme et aux exigences d'un projet professionnel d'envergure. Grâce à l'encadrement des développeurs expérimentés et aux interactions fréquentes avec le Product Owner, j'ai pu comprendre la structure et le cycle de vie des projets, tout en participant activement aux tâches de développement.

III. Les missions réalisées et apports du stage

A. Les travaux effectués

Durant les deux mois que j'ai passés au sein de l'équipe NFS (Network Facing System) chez Sopra Steria, j'ai eu l'occasion de participer à plusieurs missions concrètes réparties sur trois projets internes destinés à Orange Business. Ces missions m'ont permis de mettre en application les connaissances acquises au cours de ma formation, tout en découvrant le fonctionnement d'un environnement professionnel structuré autour de méthodes agiles. En parallèle, j'ai bénéficié d'un accompagnement me permettant de monter en compétence sur des technologies spécifiques, notamment Angular et Java, utilisées au sein des projets. L'équipe fonctionne autour d'un système de tickets que chacun s'attribue selon ses compétences et disponibilités. Pour ma part, en tant que stagiaire avec une expérience encore limitée, on m'a confié des tickets adaptés à mon niveau, essentiellement des tâches qui restaient importantes mais moins critiques ou urgentes. Cette organisation m'a permis de progresser sereinement sans être submergé par la complexité du projet, qui reste très exigeant même pour des développeurs expérimentés. Mon intégration a été facilitée par un accès complet à la documentation, aux environnements de développement, ainsi que par le soutien constant de mon tuteur.

Ma toute première mission pendant ce stage a essentiellement consisté à me former et à m'immerger dans l'environnement professionnel et technique de Sopra Steria. En effet, avant même de commencer à coder, j'ai dû suivre plusieurs formations obligatoires liées à la sécurité, qui sont indispensables pour travailler dans cette entreprise. Ces formations donnaient lieu à des certifications que je devais obtenir et transmettre à mon chef de projet avant de pouvoir accéder aux différents outils et ressources.

Par ailleurs, comme je n'avais aucune expérience en Java, un langage central dans ce projet, j'ai suivi une formation spécifique pour acquérir les bases nécessaires. Cela m'a permis de mieux comprendre le code, de pouvoir lire et analyser les scripts, et ainsi commencer à réaliser mes premières tâches. Cette phase d'apprentissage a impliqué beaucoup de recherche personnelle, de lecture de documentation, mais aussi de nombreuses questions posées aux membres de l'équipe. J'ai appris à ne pas hésiter à demander de l'aide pour avancer efficacement.

On m'a aussi clairement expliqué que le projet sur lequel je travaillais était particulièrement complexe, même pour des développeurs expérimentés. Certains d'entre eux avouaient parfois avoir du mal à tout comprendre, ce qui rendait la prise en main difficile. En tant que stagiaire avec seulement deux mois pour montrer ce dont je suis capable, la mission s'annonçait donc très exigeante. Malgré ces défis, j'ai vu cela comme une formidable opportunité d'apprentissage et de progression rapide, et j'ai tout mis en œuvre pour relever ce défi.

1. Mission n°1 – Application I&P : Ajout de contrôles sur les champs “vendor” et “release”

L'objectif principal de cette mission était de renforcer la fiabilité du formulaire permettant l'ajout d'un nouveau *vendor* et d'un nouveau *release*, en empêchant l'utilisateur de soumettre ces champs lorsqu'ils sont vides ou contiennent uniquement des espaces. Pour cela, des validations ont dû être mises en place à la fois côté **back-end** (Java) et **front-end** (Angular).

Réflexion initiale et démarche

Dès le départ, j'avais une idée assez claire de ce que je devais faire : utiliser une condition if pour vérifier si un champ était vide ou invalide. Ce qui était moins évident pour moi, c'était la manière de gérer cette erreur côté back-end. En explorant un peu plus le code, j'ai constaté que l'application utilisait des classes d'exceptions pour signaler les erreurs. Après en avoir discuté avec mon tuteur, j'ai identifié l'exception la plus adaptée, à savoir une

BadRequestException, que j'ai utilisée pour retourner un message explicite à l'utilisateur lorsque les champs “vendor” ou “release” étaient incorrectement remplis.

Côté front-end, la logique était similaire mais un peu plus complexe. Il fallait non seulement empêcher la soumission du formulaire, mais aussi informer visuellement l'utilisateur, avec des bordures rouges autour des champs concernés et un message d'erreur traduit selon la langue de l'interface (français ou anglais). Ne maîtrisant pas totalement Angular au début, j'ai eu besoin d'un accompagnement pour comprendre comment intégrer les contrôles avec les FormGroup et FormControlName, ainsi que pour gérer l'apparence conditionnelle des champs.

Ce que j'ai réalisé

Côté back-end :

- J'ai modifié le code Java afin d'ajouter une vérification : si les champs “vendor” ou “release” sont vides, nuls ou ne contiennent que des espaces (grâce à un trim), une exception de type BadRequest est levée.
- Un message explicite est alors retourné à l'utilisateur pour lui indiquer l'origine du problème.

Côté front-end :

- J'ai intégré des conditions dans le fichier TypeScript afin de bloquer l'envoi du formulaire si l'un des deux champs est vide ou invalide.
- J'ai affiché un message d'erreur, avec une gestion multilingue en l'ajoutant dans les fichiers fr.json et en.json, puis en appelant la clé de traduction correspondante dans le code.
- Visuellement, j'ai rendu les champs invalides reconnaissables grâce à une bordure rouge. Pour cela, j'ai encapsulé les champs dans une balise form et utilisé FormControlName avec FormGroup pour gérer les validations Angular.

Problèmes rencontrés

- **Instabilité du backend en local** : Lors des premiers tests, le backend plantait parfois au lancement. J'ai appris à identifier le problème et à relancer manuellement le service via **Gradle**.
- **Navigation dans le code** : Le projet étant très vaste, je me perdais souvent dans les dossiers. Au départ, j'ouvrais les fichiers un par un de manière intuitive, ce qui n'était pas très efficace. Mon tuteur m'a alors montré comment utiliser les fonctions de recherche de **VS Code** et **IntelliJ** (par exemple en cherchant “vendor”) et comment

naviguer rapidement dans le code avec **Ctrl + clic** pour suivre les définitions de classes.

- **Validation Angular** : Je n'étais pas à l'aise avec les mécanismes de validation d'Angular. On m'a montré comment gérer l'invalidité d'un champ avec un exemple concret, que j'ai ensuite pu adapter pour les autres champs.
- **Problème avec Git dans VS Code** : Contrairement à IntelliJ où je pouvais effectuer mes commits sans souci, l'interface de VS Code ne me permettait pas de le faire correctement. J'ai donc appris à utiliser le **terminal** pour effectuer mes commits manuellement, ce qui m'a permis de mieux comprendre le fonctionnement de Git en ligne de commande.

Apports de cette mission

Cette première mission m'a permis d'aborder concrètement la **validation des données côté client et serveur**, tout en me familiarisant avec une architecture **Angular/Java**. J'ai amélioré mes compétences techniques, notamment en débogage, en navigation dans une base de code complexe, en gestion de la traduction d'interface, et en utilisation d'outils comme **IntelliJ**, **VS Code**, **Gradle**, **Yarn** et **Git**. J'ai aussi pris conscience de l'importance des bonnes pratiques en matière d'expérience utilisateur, avec des messages clairs et des retours visuels efficaces.

2. Mission n°2 – Application IWA : Affichage du nombre de résultats dans la pagination

Sur l'application de collecte de données IWA, la pagination affichait le nombre total de résultats uniquement si ceux-ci étaient supérieurs à 50. Ma mission a donc consisté à modifier ce comportement afin que le nombre de résultats s'affiche systématiquement, même lorsqu'il est inférieur à 50.

Ce que j'ai réalisé :

Dans un premier temps, je pensais simplement devoir ajouter un chiffre statique sous le tableau, qui afficherait le total des résultats trouvés. Mais en analysant l'application, j'ai découvert qu'il existait déjà un module de pagination Angular, utilisé automatiquement lorsqu'il y avait plus de 50 éléments. Or, ce module affichait déjà le nombre total de résultats de manière automatique.

Avec mon tuteur, nous avons donc fait le choix d'afficher ce module tout le temps, même lorsque le nombre de résultats était inférieur à 50. Cela m'a évité de devoir recoder un système d'affichage manuel du total. En revanche, comme les flèches de navigation n'avaient pas d'utilité dans ce cas, j'ai modifié le module pour que ces flèches ne s'affichent pas lorsque les résultats tenaient sur une seule page.

En cours d'exploration du code, j'ai également identifié un bug : seuls 20 résultats s'affichaient à la première recherche, peu importe le nombre réel d'éléments trouvés. Après avoir analysé le code du composant et du service Angular, j'ai repéré une limite imposée à 20 résultats. J'ai donc modifié cette valeur pour qu'elle corresponde à la pagination attendue, à savoir 50 éléments par page.

Problèmes rencontrés :

Au début, j'ai mal compris la tâche : je pensais devoir uniquement afficher un petit chiffre sous le tableau. Ce n'est qu'en étudiant le fonctionnement de l'application que j'ai compris que la logique d'affichage dépendait du module de pagination existant. Cela m'a demandé

de chercher où était intégré ce module et de comprendre comment il était activé ou masqué selon les cas.

La structure du code n'était pas évidente à suivre, car plusieurs fichiers interagissaient pour gérer la pagination. J'ai donc dû me familiariser avec les différentes couches Angular impliquées (service, composant, HTML). De plus, la limitation à 20 résultats n'était pas documentée, j'ai dû la retrouver moi-même en analysant les conditions dans les différents fichiers liés à la recherche.

Apports :

Cette mission m'a permis de :

- Approfondir ma compréhension de la pagination dynamique dans une application Angular.
- Améliorer ma capacité à lire et analyser du code que je ne connais pas.
- Apprendre à corriger un bug existant dans un environnement professionnel structuré.
- Adapter mes idées initiales aux outils déjà présents dans le projet, ce qui m'a fait gagner du temps et éviter de dupliquer une logique existante.

3. Mission n°3 – Application I&P : Création d'une page dédiée à l'accessibilité (RGAA)

Objectif :

Créer une page web dédiée à la restitution des résultats de l'audit RGAA, ainsi qu'ajouter un lien vers cette page dans le pied de page (footer) de l'application.

Ma réflexion initiale :

Au départ, j'ai réfléchi à comment créer cette page d'accessibilité RGAA dans l'application. Je pensais devoir simplement ajouter un lien dans le pied de page qui redirigerait vers une nouvelle page, mais j'ai vite compris que cela impliquait aussi de gérer le routage complet de cette nouvelle page dans Angular.

Je pensais aussi qu'il faudrait écrire un script pour extraire automatiquement les données RGAA depuis un fichier JSON ou une autre source, ce qui m'a fait chercher une solution complexe au début.

En explorant le projet, j'ai découvert qu'il existait déjà un module d'audit qui générerait un fichier JSON avec toutes les données nécessaires. Cela a simplifié ma tâche : il ne restait plus qu'à récupérer ce fichier et afficher les données.

Ensuite, j'ai dû réfléchir à la meilleure façon de rendre ces informations claires et accessibles, en intégrant des tableaux et des graphiques, tout en veillant à ce que le lien dans le footer fonctionne dynamiquement dans tous les environnements.

Ce que j'ai réalisé :

J'ai commencé par configurer une nouvelle route /accessibility dans le fichier app.route.ts d'Angular, en créant un fichier spécifique accessibility.route.ts pour gérer les sous-routes, ce qui permettait d'accéder à la page via /accessibility ou /accessibility/rgaa.

Puis, j'ai développé un composant Angular qui affiche un tableau récapitulatif des résultats d'audit et des graphiques circulaires illustrant visuellement les taux de conformité. Ces données étaient chargées dynamiquement à partir d'un fichier JSON statique.

Pour le lien dans le pied de page, j'ai importé RouterLink dans le composant footer, ce qui permettait d'ajouter un lien dynamique vers la page RGAA. J'ai utilisé window.location.origin pour rendre les URLs adaptables selon l'environnement de déploiement.

Concernant la stylisation, j'ai utilisé du SCSS afin de rendre la page claire, accessible et visuellement cohérente, notamment pour la présentation des graphiques et l'organisation générale du contenu.

Problèmes rencontrés :

- Une mauvaise compréhension initiale du ticket : j'ai d'abord cru devoir écrire un script pour extraire et traiter les données, alors qu'elles étaient déjà disponibles dans un fichier JSON généré par un outil d'audit existant. Cette erreur m'a appris l'importance de clarifier les attentes avec l'équipe dès le début.
- Gestion des branches Git : j'ai travaillé sur deux tickets dans la même branche, ce qui a créé des conflits. Pour corriger cela, nous avons utilisé git stash afin de déplacer mes modifications sur une nouvelle branche propre. Cette expérience m'a montré l'importance de créer une nouvelle branche par ticket et de faire des commits fréquents.
- Stylisation complexe : pour rendre la page claire et accessible, j'ai dû approfondir mes compétences CSS, notamment pour afficher correctement les graphiques et assurer une bonne lisibilité.

Apports :

Cette mission m'a permis de :

- Gagner en autonomie sur la création complète d'un composant Angular, de la configuration des routes au rendu des données.
- Comprendre plus en profondeur le fonctionnement du routage Angular et du chargement dynamique de données JSON.
- Mieux gérer la communication avec l'équipe afin de clarifier les attentes d'un ticket.
- Apprendre l'importance d'une bonne gestion des branches Git et d'une organisation rigoureuse.

B. Les apports du stage

Ce stage m'a permis de développer de nombreuses compétences, tant sur le plan technique que personnel.

Sur le plan **technique**, j'ai acquis une maîtrise plus approfondie du framework Angular, notamment dans la gestion des routes, des formulaires, des composants et de la communication entre services. J'ai également consolidé mes bases en Java côté back-end, en comprenant mieux la gestion des exceptions, les contrôles de validation et la structure d'un projet Java d'envergure. L'utilisation d'outils professionnels tels que Git, IntelliJ IDEA, Visual Studio Code, Gradle, ou encore GitLab CI m'a permis de me familiariser avec des workflows utilisés quotidiennement dans le monde de l'entreprise.

Sur le plan **personnel**, j'ai appris à m'adapter à un environnement complexe, à gagner en autonomie et à prendre des initiatives. J'ai également pu constater l'importance du travail en équipe, de la communication, et de la rigueur dans la gestion des versions et des tâches.

Les difficultés que j'ai rencontrées – erreurs techniques, incompréhension de tickets, gestion de branches – m'ont permis de progresser en me confrontant à des situations concrètes, et en cherchant à les résoudre par moi-même avant de solliciter l'aide de mes collègues.

Enfin, ce stage m'a offert une immersion réelle dans le monde professionnel, et une meilleure compréhension des enjeux liés à la qualité du code, à l'accessibilité numérique, et à la sécurité des applications. Il a renforcé ma motivation à évoluer dans le domaine du développement web et à continuer à me former sur les technologies modernes.

Conclusion

Ce stage chez Sopra Steria a été une expérience extrêmement enrichissante, tant sur le plan professionnel que personnel. Intégrée au sein du service NFS (Network Facing System), j'ai pu évoluer dans un environnement agile et collaboratif, au contact de projets concrets à forte valeur ajoutée pour le client Orange Business.

Sur le plan professionnel, ce stage m'a permis de consolider mes compétences techniques en développement full stack, en particulier avec Angular pour le front-end, mais surtout avec Java côté back-end, qui m'a particulièrement intéressée. J'ai pu appréhender la gestion d'un projet complet, de la création d'une interface à la mise en place des règles métier, tout en comprenant les enjeux liés à la qualité du code, à la sécurité, et à l'intégration continue. J'ai aussi développé ma capacité à travailler en équipe, à m'adapter à une organisation agile, et à communiquer efficacement au sein d'une équipe technique.

Sur le plan personnel, cette immersion m'a confirmé mon attrait pour le développement full stack, et plus particulièrement pour la logique métier et la structuration côté serveur. Cette expérience m'a aussi appris l'importance de la rigueur, de l'autonomie et de la clarté dans la gestion des tâches.

Ce stage m'a confortée dans mon projet professionnel : je souhaite désormais approfondir mes compétences en back-end et en architecture logicielle, tout en continuant à évoluer dans des environnements full stack. À l'avenir, je me projette dans un poste de développeuse full stack, avec une spécialisation progressive vers des rôles d'architecte ou de cheffe de projet technique, où je pourrais concilier expertise technique et coordination d'équipe.