

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO  
CENTRO TECNOLÓGICO  
DEPARTAMENTO DE INFORMÁTICA  
COLEGIADO DO CURSO DE ENGENHARIA DA COMPUTAÇÃO

Nicole Rizzi Nunes

**EVOLUÇÃO DE INSTANTDP: UMA FERRAMENTA DE  
APOIO À DEFINIÇÃO E ACOMPANHAMENTO DE  
DESENVOLVIMENTO INDIVIDUAL EM EQUIPES ÁGEIS**

Projeto de Graduação apresentado ao Departamento de Informática da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Bacharel em Engenharia da Computação.

Orientadora: Monalessa Perini Barcellos

VITÓRIA  
2022

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO  
CENTRO TECNOLÓGICO  
DEPARTAMENTO DE INFORMÁTICA  
COLEGIADO DO CURSO DE ENGENHARIA DA COMPUTAÇÃO

Nicole Rizzi Nunes

**EVOLUÇÃO DE INSTANTDP: UMA FERRAMENTA DE  
APOIO À DEFINIÇÃO E ACOMPANHAMENTO DE  
DESENVOLVIMENTO INDIVIDUAL EM EQUIPES ÁGEIS**

COMISSÃO EXAMINADORA

---

Prof. Monalessa Perini Barcellos, D. Sc.

---

Prof. Roberta Lima Gomes, Ph. D.

---

Prof. Paulo Sérgio Santos Júnior, M. Sc.

Vitória, 12 de agosto de 2022

*"A verdadeira felicidade e sucesso consistem em gastar nossas energias com um propósito."*

*(William Conper)*

## AGRADECIMENTOS

Agradeço aos meus pais, Fabiola e Ivanildo, por sempre me incentivarem a seguir meus objetivos e vibrarem com as minhas conquistas.

Ao meu marido Luan, por todo amor e companheirismo. Seu apoio foi fundamental para a minha caminhada.

Aos meus amigos Beatriz, Lucio, Lucas, Úrsula que sempre agregam muito à minha vida. Eu não teria conseguido sem vocês.

À minha orientadora Monalessa, por acreditar no meu potencial, pelos ensinamentos e pelo apoio.

Às tutoras que eu tive, Roberta e Patrícia, e aos colegas de PET, por todos os momentos juntos que fizeram a graduação se tornar muito mais leve e divertida. Foi muito bom aprender e trabalhar com vocês.

Aos professores Rosane, Eliete e Vitor, por todo o conteúdo apresentado e por me incentivarem a ser uma pessoa melhor.

Por fim, agradeço a Deus por ter me guiado até aqui.

## RESUMO

As metodologias ágeis proporcionam benefícios que tornam sua utilização atraente para diversas empresas na atualidade. Para isso, é necessário que as equipes sejam compostas por profissionais com diversos tipos de habilidades.

A gestão de membros de equipes multidisciplinares em empresas que adotam metodologias ágeis não se limita a garantir a execução de tarefas com qualidade e rapidez. Também é necessário acompanhar a carreira dos membros, os conhecendo e fornecendo apoio em todo processo de construção de um plano de desenvolvimento individual.

Um dos instrumentos que pode ser utilizado nesse contexto é o Plano de Desenvolvimento Individual (PDI), que consiste em um roteiro de ações que tem como objetivo desenvolver a carreira do colaborador. Muitas empresas, como a PicPay, utilizam o Plano de Desenvolvimento Individual como forma de evolução e acompanhamento de carreira. Porém, muitas vezes isso ocorre sem um processo padrão de definição e acompanhamento do PDI.

Visando apoiar a criação e acompanhamento de PDIs na PicPay, em um trabalho de conclusão de curso de graduação anterior a este foi definido um processo que orienta sobre as atividades a serem realizadas e papéis envolvidos na criação e acompanhamento de um PDI e foi implementado o MVP (*Minimum Viable Product*) da ferramenta *InstantDP*, cujo objetivo é apoiar a execução do processo proposto.

Sendo um MVP, a versão inicialmente produzida da ferramenta apresentava algumas limitações que impediam sua utilização no mundo real. Este trabalho foi realizado com o objetivo de evoluir o MVP de *InstantDP*, a partir do aprimoramento de sua interface e da implementação de novas funcionalidades. Esta monografia apresenta as evoluções realizadas, discute as tecnologias adotadas, fundamentos teóricos e abordagens utilizados para evoluir *InstantDP* e melhorar a experiência de utilização.

**Palavras-chave:** Plano de Desenvolvimento Individual, Carreira, Gestão de Pessoas, Métodos Ágeis.

## LISTA DE FIGURAS

Figura 1 - Representação da arquitetura MVC (FERREIRA, 2021).	18
Figura 2 - Representação da arquitetura MVVM (FERREIRA, 2021).	20
Figura 3 - Tela inicial para definição de um PDI (FERREIRA, 2021).	23
Figura 4 - Tela da etapa Personal Mapping (FERREIRA, 2021).	24
Figura 5 - Tela da etapa Skill Mapping (FERREIRA, 2021).	25
Figura 6 - Tela da etapa Skill Prioritization (FERREIRA, 2021).	26
Figura 7 - Tela inicial da etapa Action Mapping (FERREIRA, 2021).	27
Figura 8 - Tela de adição de ações para uma skill (FERREIRA, 2021).	27
Figura 9 - Diagrama de casos de uso de InstantDP.	30
Figura 10 - Diagrama de classes de InstantDP.	32
Figura 11 - Arquitetura da ferramenta InstantDP (FERREIRA, 2021).	34
Figura 12 - Diagrama de Classes do CDP da ferramenta.	35
Figura 13 - Página inicial de um membro, sem PDI.	38
Figura 14 - Exemplo de feedback ao submeter uma etapa.	39
Figura 15 - Exemplo de validação de formulários, hover no botão e inclusão de labels.	40
Figura 16 - Etapa Skill Prioritization.	40
Figura 17 - Tela de equipes de um líder.	41
Figura 18 - Adicionar um novo membro.	42
Figura 19 - Tela de planos de carreira.	43
Figura 20 - Tela de posições.	43
Figura 21 - Tela de detalhes de uma posição.	44
Figura 22 - Edição de uma posição.	45
Figura 23 - Adicionar uma senioridade.	46
Figura 24 - Edição de uma senioridade.	46
Figura 25 - Exclusão de uma senioridade.	47
Figura 26 - Edição de uma skill.	48
Figura 27 - Exclusão de uma skill.	48
Figura 28 - Adição de uma skill.	49

## **LISTA DE TABELAS**

Tabela 1 - Requisitos funcionais identificados para o InstantDP	28
Tabela 2 - Objetivos e sua situação na conclusão da monografia	52

# SUMÁRIO

<b>CAPÍTULO 1</b>	<b>9</b>
INTRODUÇÃO	9
OBJETIVOS	11
HISTÓRICO DE DESENVOLVIMENTO DO TRABALHO	11
ORGANIZAÇÃO DO TEXTO	12
<b>CAPÍTULO 2</b>	<b>13</b>
DESENVOLVIMENTO ÁGIL DE SOFTWARE	13
GESTÃO DE PESSOAS	15
HEURÍSTICAS DE NIELSEN	16
TECNOLOGIAS ENVOLVIDAS NESTE TRABALHO	17
Front-end: padrão MVVM e framework Angular	19
Considerações Finais do Capítulo	20
<b>CAPÍTULO 3</b>	<b>21</b>
PROCESSO DE DEFINIÇÃO DE PDI NA FERRAMENTA INSTANTDP	21
REQUISITOS	28
CASOS DE USO	29
MODELO ESTRUTURAL	31
PROJETO DE SISTEMA	32
Arquitetura de Software	33
Camada de Lógica de Negócio	34
Componente de Domínio do Problema (CDP)	34
Camada de Interface com o usuário (CIU)	35
Componente de Interação Humana	36
Componente de Controle de Interação	37
Camada de Gerência de Dados	37
FERRAMENTA INSTANTDP APÓS AS MODIFICAÇÕES	37
Testes	49
CONSIDERAÇÕES FINAIS DO CAPÍTULO	49
<b>CAPÍTULO 4</b>	<b>51</b>
CONCLUSÃO	51
Trabalhos futuros	53
<b>REFERÊNCIAS BIBLIOGRÁFICAS</b>	<b>54</b>



# Capítulo 1

## Introdução

---

*Este capítulo apresenta uma breve introdução ao tema do trabalho, seus objetivos, histórico do desenvolvimento e a organização deste documento.*

### 1.1 Introdução

Métodos Ágeis são um conjunto de práticas e técnicas para gestão de projetos e desenvolvimento de software que oferece eficiência, rapidez e flexibilidade. Os métodos ágeis surgiram a partir de um grupo de profissionais que escreveram o Manifesto Ágil<sup>1</sup>, que determina princípios a serem seguidos com o objetivo de simplificar e mudar a forma de conduzir projetos de software (AMORIM, 2015).

As promessas de qualidade, rapidez e redução de custos tornam os métodos ágeis cada vez mais frequentes nas empresas (AMORIM, 2015). Esses métodos incentivam maior trabalho em equipe, foco no cliente, auto-organização e adaptação. Com isso, projetos ágeis são muito dinâmicos e possuem entregas frequentes, atribuindo qualidade e agilidade ao produto final.

Com o passar dos anos, as ideias de agilidade pautadas no Manifesto Ágil tomaram conta do cenário de desenvolvimento de software corporativo e atualmente muitas empresas com projetos dinâmicos adotam os métodos ágeis nas suas equipes de desenvolvimento.

A empresa PicPay, uma *fintech* criada com a missão de tornar a vida das pessoas mais fácil através de soluções financeiras, utiliza métodos ágeis em seus projetos. Na área de tecnologia, a empresa organiza as equipes de forma a existir verticais, compostas por tribos que, por sua vez, são compostas por *squads*. Um *squad* é composto por uma equipe multidisciplinar, focada em um objetivo ou projeto. Como exemplo, há a *squad* de *Design System* que é responsável por propor padrões de *design* e componentes para que o usuário tenha a mesma experiência através de diferentes partes do site ou do app. Uma tribo é um conjunto de *squads* com um propósito em comum como, por exemplo, a tribo Experiência Base que é responsável pela experiência geral do usuário do PicPay. A vertical é um conjunto de tribos que

---

<sup>1</sup> <http://agilemanifesto.org/>

possuem um objetivo maior como, por exemplo, a vertical *Tech Cross* que tem o propósito de desenvolver produtos e soluções para outras equipes de dentro da empresa.

Como citado, uma *squad* é uma equipe multidisciplinar formada de acordo com as necessidades do objetivo ou projeto. Além de outras posições, ela sempre conta com uma equipe técnica, um *Tech Manager* e um *Product Manager*. O *Product Manager* é responsável por desenvolver a estratégia de longo prazo na visão de produto com base nas tendências de mercado e identificação de novas oportunidades. O *Tech Manager*, por sua vez, é responsável por liderar a equipe técnica, gerenciando o fluxo de trabalho, dando *feedbacks* e desenvolvendo junto a cada membro da equipe técnica um Plano de Desenvolvimento Individual (PDI).

O Plano de Desenvolvimento Individual tem como finalidade desenvolver a carreira do profissional, bem como as *skills* (habilidades) necessárias para assumir certas funções (MOREIRA, 2022). O PDI é composto por objetivos e ações para que o colaborador possa desenvolver uma ou mais habilidades. Um bom PDI deve atender tanto às necessidades da empresa quanto as do colaborador.

Atualmente, a PicPay adota a implementação de PDI para todos os colaboradores através da ferramenta Culture Rocks<sup>2</sup>, onde pode-se acompanhar o andamento das ações do PDI de cada colaborador. Porém, a elaboração do PDI não segue um processo padronizado e fica a cargo de cada líder definir um modelo que ache melhor.

Nesse contexto, visando auxiliar na criação e monitoramento de PDIs de maneira sistemática, Ferreira (2021) definiu um processo de criação e acompanhamento de PDI e desenvolveu um MVP (*Minimum Viable Product*) de uma ferramenta denominada *InstantDP*, que apoia a execução do processo para a criação e monitoramento de PDIs em equipes ágeis. Sendo um MVP, a versão desenvolvida por Ferreira (2021) foi resultado de um primeiro ciclo de desenvolvimento e possui algumas limitações quanto à usabilidade e às funcionalidades da ferramenta. Algumas dessas limitações foram foco deste trabalho, particularmente:

- Em relação à usabilidade, a interface apresentada no MVP possuía algumas limitações em relação à apresentação de informações e feedbacks ao usuário. Para implementação de melhorias na usabilidade foram consideradas as Heurísticas de Nielsen (ROSSETTI, 2020), que definem algumas orientações importantes acerca de usabilidade, visando às

---

<sup>2</sup> <https://culture.rocks/>

melhorias na interação do sistema com o usuário (e.g., apresentação do status do sistema, adição de mensagens de erro, etc.).

- Em relação às funcionalidades, o MVP da ferramenta não implementava todos os requisitos e casos de uso definidos em (FERREIRA, 2021). Neste trabalho foram implementados aqueles que não foram tratados em (FERREIRA, 2021).

## 1.2 Objetivos

Este trabalho tem como propósito o aprimoramento da ferramenta *InstantDP* de forma a evoluir o MVP proposto em (FERREIRA, 2021). Esse objetivo pode ser detalhado nos seguintes objetivos específicos:

- i. Identificar e implementar melhorias na usabilidade do sistema, considerando as Heurísticas de Nielsen;
- ii. Identificar e implementar novas funcionalidades de forma a ampliar o atendimento às necessidades relacionadas à criação e acompanhamento de PDIs.

## 1.3 Histórico de Desenvolvimento do Trabalho

Este trabalho foi conduzido de acordo com as seguintes atividades:

- i. *Revisão da Literatura*: O trabalho teve início com uma revisão bibliográfica sobre agilidade e gestão de pessoas, na qual foram lidos materiais (livros, teses, dissertações e artigos científicos) pertinentes ao assunto, incluindo o trabalho de Ferreira (2021).
- ii. *Estudo de Tecnologias*: Nesta etapa ocorreu o estudo de tecnologias relevantes para o desenvolvimento da ferramenta, destacando-se: o *framework Angular*, que utiliza as linguagens *TypeScript*, HTML e SCSS, para o lado do cliente da aplicação, e o *framework NestJS*, que utiliza *TypeScript* como linguagem, para o lado do servidor. Também ocorreu o estudo dos padrões MVVM (*Model-View-ViewModel*), utilizado no *Angular*, e MVC (*Model-View-Controller*), utilizado no *NestJS*.
- iii. *Levantamento e Análise de Requisitos*: Consistiu no entendimento da ferramenta *InstantDP* proposta em (FERREIRA, 2021), na identificação e análise dos requisitos a serem contemplados na evolução do sistema.

- iv. *Design, Implementação e Testes:* Nesta etapa ocorreu o desenvolvimento de *mockups* da ferramenta e desenvolvimento das funcionalidades. Foram feitos testes ao longo do desenvolvimento do projeto e no momento de conclusão das funcionalidades.
- v. *Escrita da Monografia:* Consistiu na escrita desta monografia.

## 1.4 Organização do Texto

Neste capítulo inicial está contida a introdução do trabalho, bem como seus objetivos e seu histórico de desenvolvimento. Além deste capítulo, a monografia possui outros três, a saber:

*Capítulo 2 – Fundamentação Teórica:* Apresenta uma breve fundamentação teórica sobre desenvolvimento ágil de software e gestão de pessoas com destaque para a importância do PDI. Também apresenta as tecnologias que foram utilizadas no desenvolvimento da ferramenta e as heurísticas de Nielsen, utilizadas na melhoria da ferramenta.

*Capítulo 3 – Ferramenta InstantDP:* Apresenta a ferramenta, suas funcionalidades, casos de uso, diagrama de classes e arquitetura. Também apresenta as melhorias feitas considerando-se as Heurísticas de Nielsen e as novas funcionalidades implementadas, apresentando os fluxos de utilização.

*Capítulo 4 – Considerações Finais:* Apresenta as considerações finais do trabalho, incluindo limitações da ferramenta desenvolvida e experiências adquiridas. Além disso, são identificados alguns trabalhos futuros visando à evolução da ferramenta.

## Capítulo 2

### Fundamentação Teórica

---

*Este capítulo apresenta os principais aspectos teóricos que fundamentam este trabalho. Ele está organizado em cinco seções, a saber: Desenvolvimento Ágil de Software (Seção 2.1), Gestão de Pessoas (Seção 2.2), Tecnologias envolvidas nesse trabalho (Seção 2.3), Heurísticas de Nielsen (Seção 2.4) e Considerações Finais do Capítulo (Seção 2.5).*

#### 2.1 Desenvolvimento Ágil de Software

O desenvolvimento em projetos ágeis segue os princípios do Manifesto Ágil (BECK et al., 2001), um documento feito por 17 especialistas, que determina doze princípios:

- Nossa maior prioridade é satisfazer o cliente através da entrega contínua e adiantada de software com valor agregado.
- Mudanças nos requisitos são bem-vindas, mesmo tardiamente no desenvolvimento. Processos ágeis tiram vantagem das mudanças visando vantagem competitiva para o cliente.
- Entregar frequentemente software funcionando, de poucas semanas a poucos meses, com preferência à menor escala de tempo.
- Pessoas de negócio e desenvolvedores devem trabalhar diariamente em conjunto por todo o projeto.
- Construa projetos em torno de indivíduos motivados. Dê a eles o ambiente e o suporte necessário e confie neles para fazer o trabalho.
- O método mais eficiente e eficaz de transmitir informações para e entre uma equipe de desenvolvimento por meio de conversa face a face.
- Software funcionando é a medida primária de progresso.

- Os processos ágeis promovem desenvolvimento sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter um ritmo constante indefinidamente.
- Contínua atenção à excelência técnica e bom design aumenta a agilidade.
- Simplicidade: a arte de maximizar a quantidade de trabalho não realizado é essencial.
- As melhores arquiteturas, requisitos e designs emergem de equipes auto organizáveis.
- Em intervalos regulares, a equipe reflete sobre como se tornar mais eficaz e então refina e ajusta seu comportamento de acordo.

O gerenciamento ágil de projetos é uma abordagem fundamentada em um conjunto de princípios, cujo objetivo é tornar o processo de gerenciamento de projetos mais simples, flexível e iterativo, de forma a obter melhores resultados em desempenho (tempo, custo e qualidade), menos esforço em gerenciamento e maiores níveis de inovação e agregação de valor para o cliente (AMARAL et al., 2011). O método ágil que mais tem sido utilizado no Brasil é o Scrum, um *framework* para gestão de projetos. No Scrum, o projeto é dividido em ciclos de atividades e as reuniões de alinhamento de tarefas são frequentes, o que proporciona um acompanhamento de perto, dando bastante dinamicidade ao projeto.

Outro método bem conhecido é o Kanban, um sistema definido em um quadro com colunas que permite dividir o trabalho em segmentos ou por status. Nesse método, cada tarefa é alocada em uma coluna, dando visibilidade do seu status a todos. Existem alguns tipos de Kanban que se diferenciam nos tipos de cartões que são utilizados para representar as tarefas (MENDES, 2020).

Existem outros *frameworks*, como por exemplo XP (*Extreme Programming*) e Lean startup, e cada equipe precisa avaliar qual método (ou combinação de métodos) faz mais sentido no contexto do projeto a ser usado.

## 2.2 Gestão de Pessoas

Gestão de pessoas é um conjunto de práticas que visa desenvolver os profissionais da empresa, fazendo com que trabalhem mais engajados e satisfeitos, impactando no resultado final das metas da empresa (GUIMARÃES, 2022).

O trabalho de um líder é construir uma equipe que funcione bem em conjunto, apoiar os membros no alcance de suas metas de carreira e criar processos para realizar o trabalho de forma eficiente e suave. Zhuo (2019) diz que um líder de equipe tem três trabalhos: certificar-se de que sua equipe saiba como é o sucesso e se preocupe em alcançá-lo, entender as pessoas (saber se os membros possuem as habilidades certas, se estão motivados, etc.) e desenvolver grandes processos de equipe.

Segundo Fowler (2005), os métodos ágeis surgem com as principais características de adaptabilidade e foco nas pessoas, enquanto que os métodos tradicionais objetivam a definição de processos e execução dos planos com a tentativa de adequar as pessoas ao processo, frustrando muitas vezes o desempenho do projeto devido as pessoas serem difíceis de prever e quantificar.

Quando métodos ágeis são empregados no desenvolvimento de software, as pessoas têm seu potencial aumentado, gerando maior produtividade e qualidade, agregando assim maior valor ao produto final (PAZ, 2013). Gerir uma equipe vai além de planejar os processos e tarefas. É necessário se preocupar com a motivação, engajamento e evolução de cada um dos membros.

Para gerenciar uma equipe, é necessário conhecer seus liderados, suas competências, seus pontos fracos e suas ambições. Conhecer bem a equipe e entender a melhor forma de trabalho para os seus membros é crucial para que o gestor tome decisões de maneira mais fácil e efetiva (ZHUO, 2019).

O uso de Planos de Desenvolvimento Individual (PDI) é uma abordagem que visa contribuir para a capacitação e crescimento dos membros das equipes. Um PDI adequado permite ao indivíduo enxergar um caminho de crescimento e evolução (ZHUO, 2019). O líder é responsável por direcionar a construção de um PDI que faça sentido tanto para a pessoa quanto para a empresa, sistematizando ações a serem tomadas que sirva de roteiro para que a pessoa atinja objetivos definidos.

Apesar do PDI ser realizado pelo colaborador, também é de responsabilidade do líder acompanhar o andamento do PDI e fornecer *feedbacks* constantes, engajando e motivando a equipe a evoluir.

### 2.3 Heurísticas de Nielsen

Em 1990, Jakob Nielsen desenvolveu 10 heurísticas para orientar a criação de um projeto de uma boa interface e, por consequência, uma ótima experiência de uso para o usuário. Essas heurísticas tornam os produtos e serviços digitais mais humanos e mais agradáveis (ROSSETTI, 2020). São elas:

- **Visibilidade do Status do Sistema:** o sistema deve manter o usuário informado em tempo real por meio de *feedbacks* sobre o que está acontecendo na tela em que ele se encontra. Como exemplo, temos o efeito *hover* (efeito que indica a possibilidade de interagir com o elemento da tela) em elementos clicáveis e mensagens de sinalização quando um processo está em andamento ou é concluído.
- **Correspondência entre o sistema e o mundo real:** o sistema deve usar a mesma linguagem do usuário, ou seja, utilizar palavras, termos, ícones, imagens e conceitos que sejam familiares ao usuário como, por exemplo, utilizar o ícone de lixeira para indicar uma ação de excluir.
- **Controle e liberdade para o usuário:** é importante que o usuário tenha liberdade para realizar as ações que ele deseja. Porém, é necessário que haja uma "saída de emergência" de fácil localização caso a ação seja realizada por engano, permitindo que o usuário retorne ao ponto anterior. O Gmail, por exemplo, dá a opção de desfazer a ação quando um usuário exclui um e-mail.
- **Consistência e padronização:** a interface deve ser previsível e de fácil aprendizagem. O usuário não deve ficar em dúvida sobre o significado de palavras, situações ou ações. Para isso, o sistema deve ter consistência visual, ou seja, manter tamanho de fonte, cores, posição de menus e padrões já consolidados.
- **Prevenção de erros:** ao implementar o sistema, deve-se mapear prováveis erros a serem cometidos, prevenindo-os antes que eles ocorram. Isso pode ser feito, por exemplo, informando quando o usuário digita algo inválido ou dando uma opção de confirmação antes de completar uma ação.



- **Reconhecimento ao invés de memória:** deve-se minimizar a quantidade de informações que o usuário precisa memorizar, por isso, as informações devem ser claras e facilmente identificadas. Para isso, deve-se deixar sempre que possível lembretes de informações que possam ser úteis.
- **Eficiência e flexibilidade de uso:** o sistema deve ser eficiente e flexível para atender as necessidades dos usuários mais experientes, ao mesmo tempo que deve ter as informações detalhadas para atender os usuários mais leigos. A interface deve permitir que a medida que os usuários vão conhecendo a interface consigam interagir de forma mais rápida, adaptando ações frequentes de acordo com o nível em que estão. Um bom exemplo dessa heurística é a existência de atalhos para ações que o usuário faça recorrentemente.
- **Estética e design minimalista:** quanto maior a quantidade de informações, mais tempo o usuário levará para obter a informação mais relevante. Por isso, o design deve ser focado no conteúdo mais importante, deixando informações irrelevantes de lado.
- **Recuperação diante de erros:** os erros devem ser sinalizados de forma clara, indicando o problema ocorrido e sugerindo, quando possível, soluções para que o usuário possa reconhecer a situação e se recuperar dela. Quando um dado é inserido de forma incorreta em um formulário, por exemplo, a ferramenta pode apresentar uma mensagem informando que o dado é inválido.
- **Ajuda e documentação:** é importante que o usuário consiga ter as informações necessárias para utilizar o sistema e tenha resposta em caso de dúvidas. Por isso, sempre que possível, o sistema deve oferecer documentações ou explicações para ajudar o usuário, seja um *tooltip* explicando o que um ícone de ação faz ou até mesmo um FAQ contendo respostas para perguntas frequentes.

## 2.4 Tecnologias Envolvidas neste Trabalho

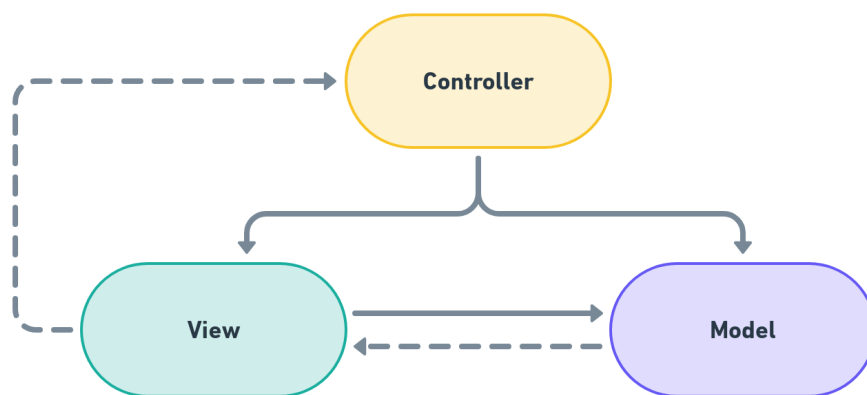
Nesta seção são descritas todas as tecnologias que foram aplicadas para atingir os objetivos deste trabalho. A ferramenta foi desenvolvida como uma aplicação baseada na Web (também conhecida como *WebApp*) (PRESSMAN; LOWE, 2009), seguindo o modelo de *front-end* e *back-end*, onde o *front-end* é a interface com a qual o usuário da ferramenta interage e o *back-end* é uma estrutura que controla e realiza a lógica de negócio da aplicação no servidor.

A ferramenta *InstantDP* foi originalmente implementada utilizando as tecnologias a seguir, as quais foram selecionadas no contexto do trabalho de Ferreira (2021). As tecnologias utilizadas na construção do MVP atendem às necessidades deste trabalho e, portanto, foram mantidas.

#### 2.4.1 Back-end: padrão MVC e *framework* NestJS

NestJS<sup>3</sup> é um *framework* para criar aplicativos `node.js`<sup>4</sup> eficientes e escaláveis do lado do servidor que pode ser construído utilizando as linguagens TypeScript<sup>5</sup> ou JavaScript<sup>6</sup>. Neste trabalho foi utilizado TypeScript.

O NestJS foi utilizado com padrão de arquitetura MVC (*Model-View-Controller*), que consiste em três camadas de funcionalidades definidas, sobre quais o software será construído. A camada *Model* tem a responsabilidade de representar o modelo do negócio e manipulação de dados na aplicação. Essa camada interage diretamente com a camada *Controller*, que processa os dados da *Model* e as envia para a camada *View*, cuja função é exibi-las ao usuário. A figura 1 representa a interação das camadas MVC.



**Figura 1 - Representação da arquitetura MVC (FERREIRA, 2021).**

A implementação do *back-end* com o *framework* NestJS neste trabalho utilizou as seguintes tecnologias:

---

<sup>3</sup> <https://nestjs.com/>

<sup>4</sup> <https://nodejs.org/en/>

<sup>5</sup> <https://www.typescriptlang.org/>

<sup>6</sup> <https://www.javascript.com/>

- **TypeScript:** Linguagem de programação que estende os recursos atuais do JavaScript com a possibilidade de adicionar tipos estáticos predefinidos na sua aplicação, em tempo de compilação.
- **MYSQL<sup>7</sup>:** Sistema de gerenciamento de banco de dados relacional, que utiliza a linguagem de consulta estruturada, o SQL.

#### 2.4.2 Front-end: padrão MVVM e *framework* Angular

O Angular<sup>8</sup>, é um *framework* utilizado para construção de SPAs (*Single Page Applications*) construído em TypeScript que possui uma estrutura baseada em componentes para criação de aplicativos Web escaláveis.

O Angular foi utilizado com padrão de arquitetura MVVM (*Model-View-ViewModel*), que se assemelha em alguns aspectos ao MVC. O MVVM visa estabelecer uma clara separação de responsabilidades em uma aplicação e também consiste em três camadas sobre quais o software será construído. A *Model* é o modelo de domínio da aplicação, ou seja, as classes de negócio que serão utilizadas na aplicação. A *View* tem a responsabilidade de definir a estrutura do que o usuário vê na tela. Por último, a *ViewModel* faz a ponte entre as camadas *Model* e *View* e pode implementar a lógica de validação para garantir a consistência dos dados. A figura 2 representa a interação das camadas MVVM.



**Figura 2 - Representação da arquitetura MVVM (FERREIRA, 2021).**

A implementação do *front-end* com o *framework* Angular neste trabalho utilizou as seguintes linguagens:

<sup>7</sup> <https://www.mysql.com/>

<sup>8</sup> <https://angular.io/>

- **TypeScript:** Linguagem de programação base do Angular que estende os recursos atuais do JavaScript com a possibilidade de adicionar tipos estáticos predefinidos na sua aplicação, em tempo de compilação.
- **HyperText Markup Language (HTML)**<sup>9</sup>: Linguagem de marcação de hipertexto utilizada na construção e organização de informações dentro de páginas web.
- **Sassy Cascading Style Sheets (SCSS)**<sup>10</sup>: É uma linguagem pré-processada de estilo CSS (*Cascading Style Sheets*) utilizada para configurar a apresentação de componentes descritos em HTML.

## 2.5 Considerações Finais do Capítulo

Neste capítulo foram abordados os conceitos teóricos e tecnológicos necessários para o desenvolvimento deste trabalho. Primeiramente, foram apresentados os tópicos relacionados a desenvolvimento ágil de software e gestão de pessoas, que fornecem a base teórica para a construção da ferramenta *InstantDP*. Também foram apresentadas as heurísticas de Nielsen, que foram utilizadas neste trabalho nas melhorias feitas no sistema visando à usabilidade. Por fim, foram discutidas as tecnologias utilizadas no desenvolvimento da ferramenta.

---

<sup>9</sup> <https://html.spec.whatwg.org/>

<sup>10</sup> <https://sass-lang.com/>

## Capítulo 3

### Evolução da Ferramenta InstantDP

---

*Este capítulo apresenta a ferramenta InstantDP e as melhorias realizadas no contexto deste trabalho. Na Seção 3.1 é brevemente apresentado o processo de definição e acompanhamento de PDIs definido por Ferreira (2021) e algumas telas de InstantDP para ilustrar como o processo é apoiado pela ferramenta. Na Seção 3.2 são apresentados os requisitos da ferramenta. Na Seção 3.3 são apresentados os casos de uso. Na Seção 3.4 é apresentado o modelo estrutural, com o diagrama de classes. Na Seção 3.5 é apresentado o projeto de sistema. Na Seção 3.6 são apresentadas as telas da ferramenta após as evoluções realizadas neste trabalho. Na Seção 3.7 é falado sobre os testes de usabilidade. Na seção 3.8 são apresentadas as considerações finais do capítulo.*

#### 3.1 Processo de Definição de PDI na ferramenta *InstantDP*

A ferramenta *InstantDP* foi desenvolvida por Ferreira (2021) com o objetivo de dar apoio à definição e acompanhamento de PDIs, seguindo um processo definido pela mesma autora. Nesta seção são apresentadas algumas das atividades do processo e são ilustradas algumas telas do sistema que apoiam sua execução. O objetivo é apresentar ao leitor uma visão geral do processo para propiciar um melhor entendimento das evoluções realizadas no contexto deste trabalho.

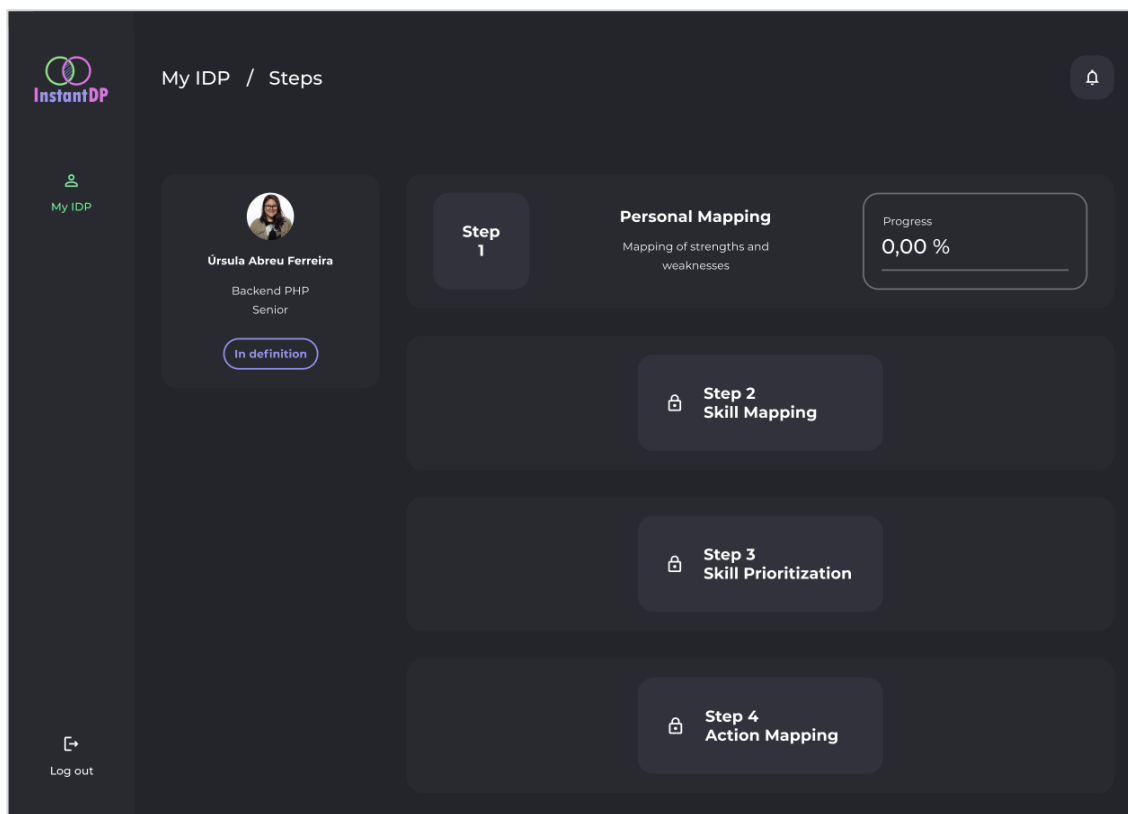
Para explicar o processo de PDI apoiado pela ferramenta *InstantDP*, é necessário, primeiramente, apresentar alguns termos utilizados no contexto do processo (FERREIRA, 2021):

- *Skill*: Uma *skill* é uma habilidade, que pode ser classificada como técnica ou comportamental, e agrupada com outras habilidades de acordo com o contexto. Por exemplo: comunicação pode ser considerada uma *skill* comportamental, enquanto conhecimento em testes unitários pode ser considerada uma *skill* técnica, que poderia ser agrupada com *skills*, como conhecimento em testes de integração, em um grupo de *skills* de testes automatizados. Cada *skill* pode ser classificada em níveis de conhecimento como, por exemplo, iniciante, médio e avançado.

- Posição: Uma posição é um cargo que uma pessoa exerce. Cada cargo tem níveis de senioridade, de acordo com a experiência. Cada senioridade necessita que a pessoa tenha determinadas *skills* para exercer o cargo.
- Ação: Uma ação é uma atividade ou esforço para obtenção de conhecimento em uma determinada *skill*. Para desenvolver o conhecimento em uma *skill* podem ser necessárias uma ou várias ações, que devem ter quantidade de tempo mensurada e metas de avaliação do conhecimento adquirido.
- Ciclo: Em um PDI, é necessário fazer um planejamento de ações que devem ser realizadas em um determinado prazo, de acordo com as *skills* que são mais relevantes no contexto atual da pessoa. Assim, a empresa pode determinar ciclos, ou seja, períodos em que uma pessoa realiza as ações planejadas e relacionadas a uma ou várias *skills*. Ao fim de cada ciclo, um novo conjunto de ações é definido em um novo PDI, considerando a evolução adquirida no ciclo anterior.
- Plano de carreira: Descreve todas as posições da empresa com suas respectivas senioridades. O plano de carreira também deve conter, para cada senioridade, o conjunto de *skills* necessárias com seus respectivos níveis esperados.

O processo de criação de um PDI se baseia na posição e senioridade em que a pessoa se encontra, para que ela possa comparar as *skills* que ela possui com as necessárias para a função. Para isso, é necessário que a empresa construa um plano de carreira bem estruturado.

Considerando que a empresa já tenha esse plano cadastrado no sistema, o processo pode ser aplicado seguindo quatro etapas que se referem à definição de um ciclo do PDI. A Figura 3 ilustra as etapas do processo na ferramenta (*Personal Mapping, Skill Mapping, Skill Prioritization e Action Mapping*), que é descrito em seguida.



**Figura 3 - Tela inicial para definição de um PDI (FERREIRA, 2021).**

- i. **Personal Mapping:** Na primeira etapa a pessoa deve avaliar sua carreira e qual das trilhas existentes na empresa ela deseja seguir. Independente do ramo da empresa, sempre existe a possibilidade de seguir uma trilha mais técnica como especialista, por exemplo, e uma trilha de gestão. Além de entender a trilha com a qual a pessoa se identifica, essa etapa também tem como objetivo permitir à pessoa identificar e mapear seus pontos fortes e fracos. Por exemplo: uma pessoa pode ocupar a posição de um desenvolvedor *front-end* com senioridade sênior e escolher seguir a trilha de gestão já sabendo que possui pontos fortes em relação a liderança e gestão de tempo, e possui pontos fracos com relação a *feedbacks*. Considerando que a empresa possui a opção de trilha de gestão em seu plano de carreira, esta etapa resultaria nas informações: a trilha escolhida é gestão; os pontos fortes são: liderança e gestão de tempo; e o ponto fraco é *feedback*. A Figura 4 ilustra a tela que apoia a realização desta etapa:

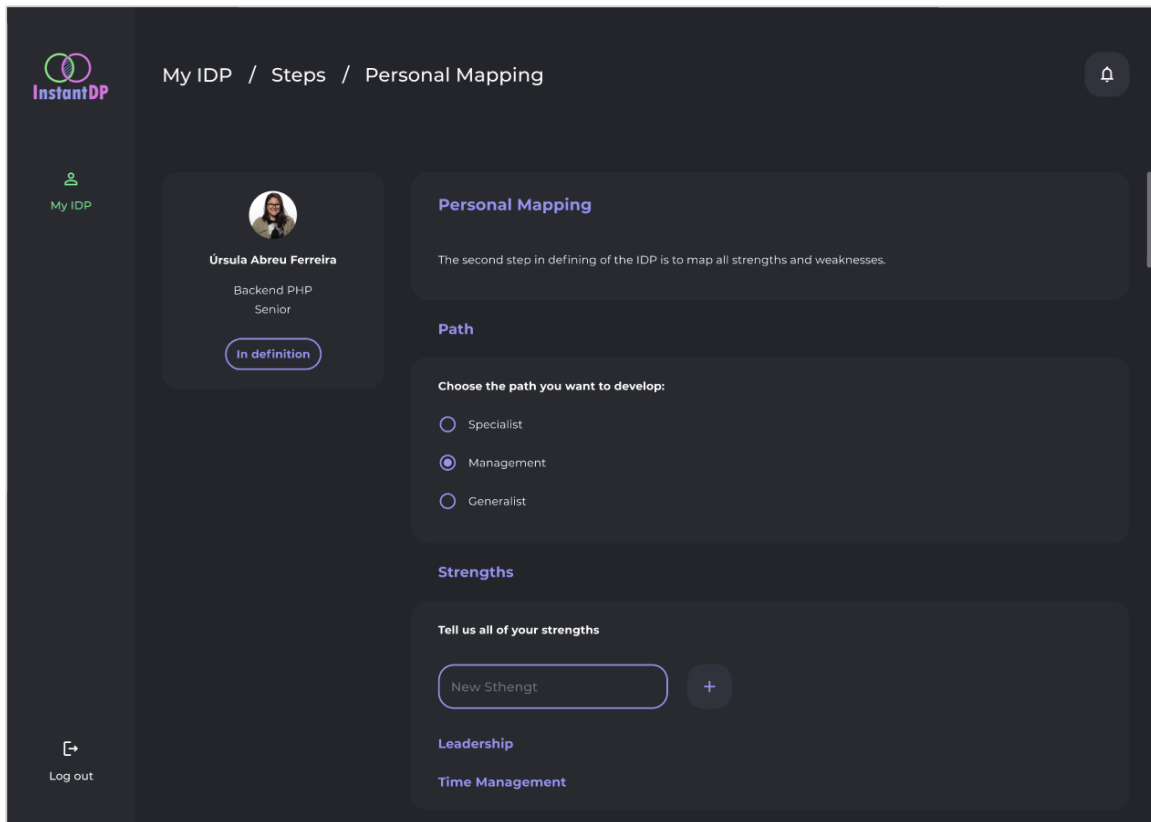


Figura 4 - Tela da etapa *Personal Mapping* (FERREIRA, 2021).

- ii. ***Skill Mapping***: Cada nível de senioridade de uma posição espera que a pessoa tenha um nível de cada *skill* necessária para desempenhar a função. Nesta etapa é feito o mapeamento das *skills* do colaborador em relação às *skills* esperadas para a posição que ele ocupa. Este mapeamento pode ser feito a partir de uma autoavaliação, de uma avaliação de pares e/ou de uma avaliação do líder. Por exemplo: para a mesma pessoa citada na etapa anterior, nesta etapa serão apresentadas as *skills* esperadas para um *front-end* sênior com seus níveis, para que o funcionário faça uma autoavaliação e marque em que nível se encontra. Ao completar essa etapa, é gerada uma lista com as *skills* que se encontram abaixo do esperado para a posição e senioridade em que a pessoa se encontra.



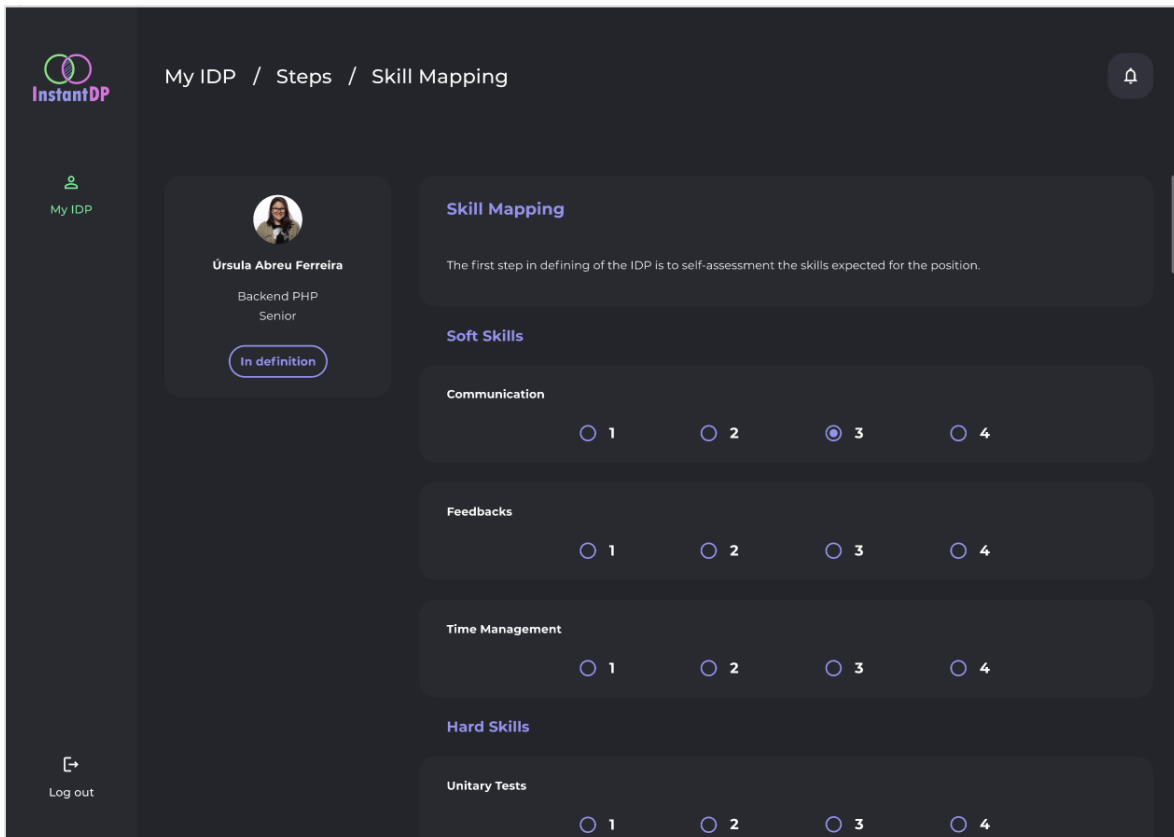


Figura 5 - Tela da etapa *Skill Mapping* (FERREIRA, 2021).

- iii. ***Skill Prioritization***: Com base nos resultados da etapa anterior, essa etapa tem o objetivo de priorizar as *skills* nas quais a pessoa irá focar no próximo ciclo de PDI. É importante que esta etapa seja feita junto com o líder, para que as *skills* sejam priorizadas levando em consideração quais impactam mais no dia a dia de trabalho, quais estão mais longe do esperado e quais demandam maior esforço de aprendizado. Por exemplo: ainda para o caso apresentado anteriormente, se a pessoa se autoavaliar para a *skill* de comunicação com o nível 2, mas no plano de carreira da empresa o nível esperado para um desenvolvedor *front-end* com senioridade sênior é 3, a *skill* de comunicação estaria presente na lista de *skills* e a pessoa pode decidir se essa *skill* será priorizada para o ciclo atual.

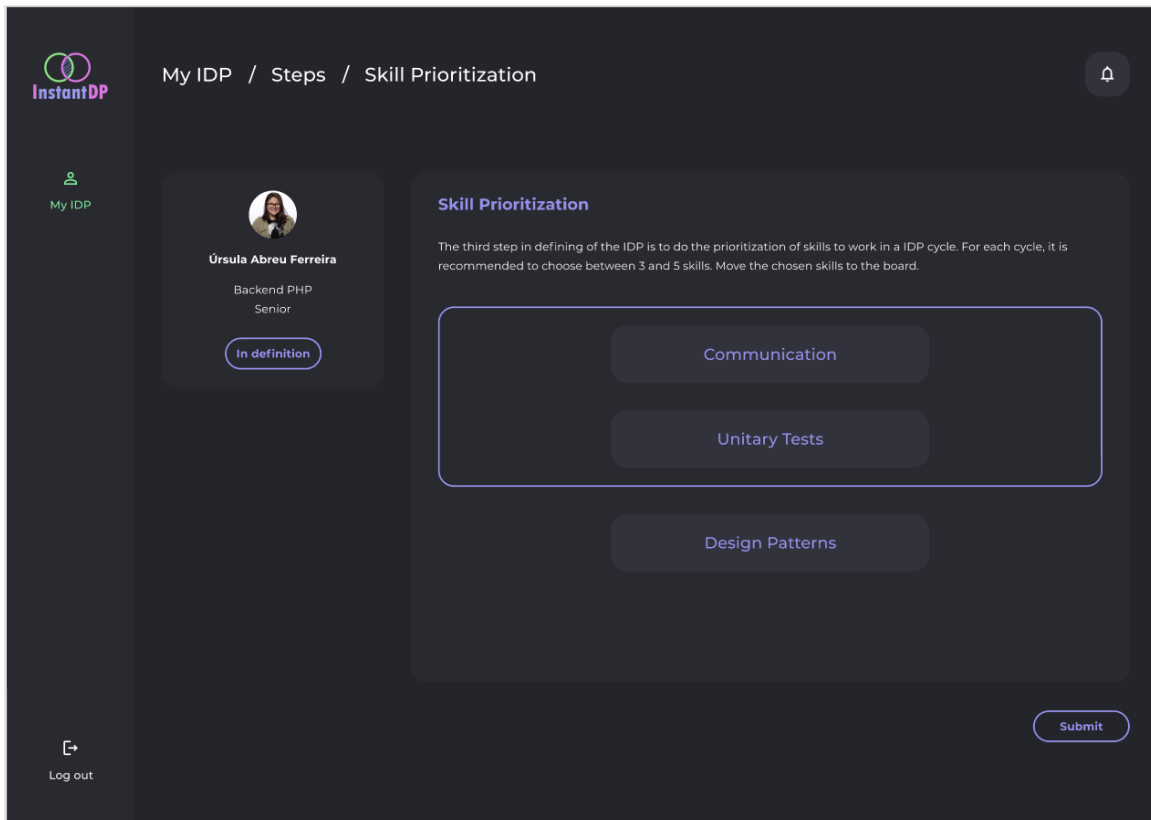


Figura 6 - Tela da etapa *Skill Prioritization* (FERREIRA, 2021).

- iv. **Action Mapping:** A última etapa é a etapa de mapeamento das ações. Tendo definidas quais as *skills* serão priorizadas, é preciso, para cada *skill* priorizada, definir ações com finalidade de desenvolvê-la ou potencializá-la. À medida que a pessoa conclui ações, mais a pessoa se aproxima do estado desejado. Essas ações deverão ser feitas até o final do ciclo para cada *skill*. Por exemplo: continuando o exemplo anterior, se a pessoa escolher priorizar a *skill* de comunicação, nesta etapa a pessoa precisa definir todas as ações para essa *skill*, incluindo ações para adquirir mais conhecimento como leitura de livros e ações de validação como apresentação de um conteúdo em uma reunião da empresa. Ao longo do ciclo o líder deverá fazer o acompanhamento se as ações estão sendo realizadas e se o colaborador está evoluindo nas *skills*. As figuras 7 e 8 ilustram essa etapa:

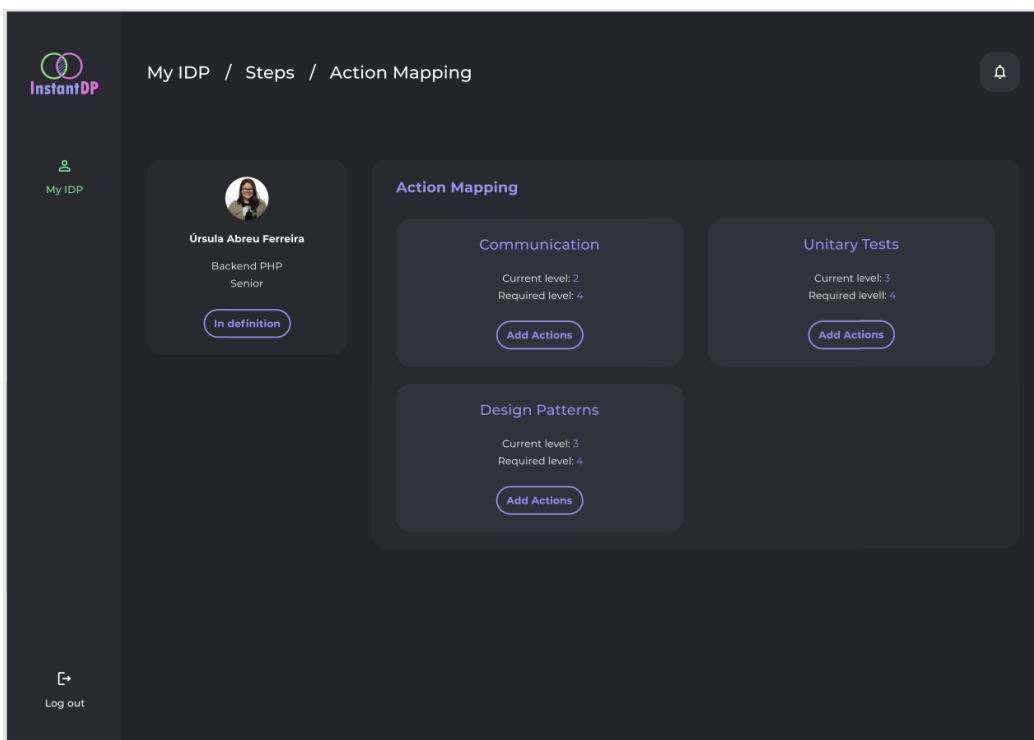


Figura 7 - Tela inicial da etapa *Action Mapping* (FERREIRA, 2021).

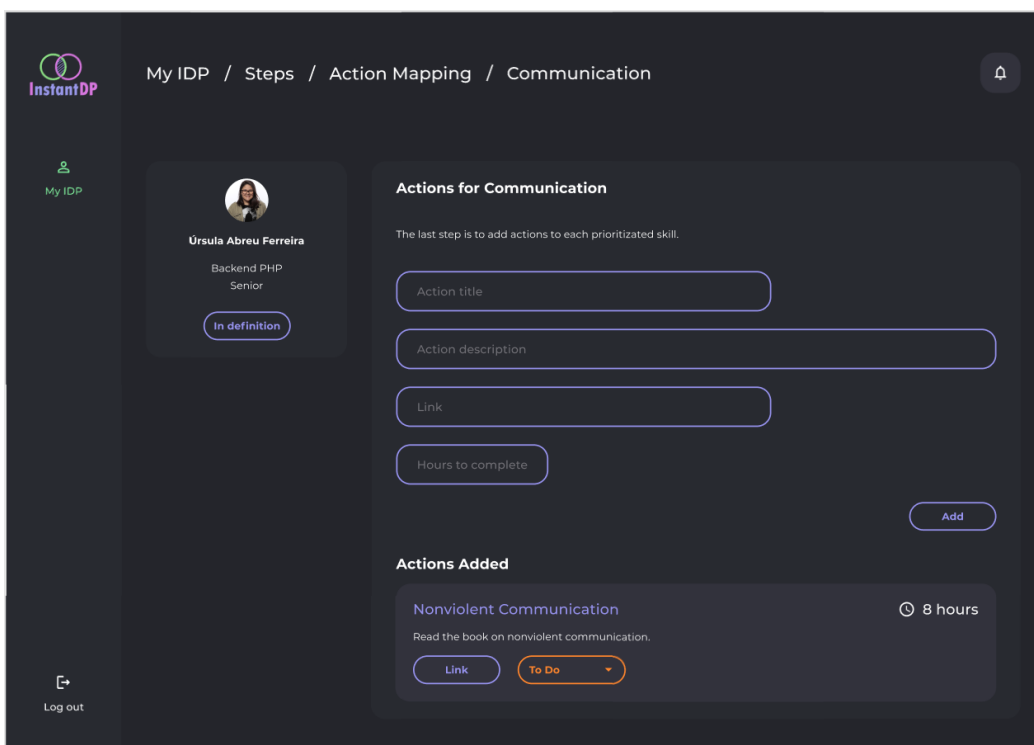


Figura 8 - Tela de adição de ações para uma *skill* (FERREIRA, 2021).

Além do processo de criação de um PDI, a ferramenta possui outras funcionalidades, que serão discutidas mais adiante.

### 3.2 Requisitos

A Tabela 1 apresenta os requisitos funcionais definidos para o sistema em (FERREIRA, 2021). Neste trabalho, foram tratados os requisitos RF02, RF03 e RF04 (destacados em azul na tabela).

**Tabela 1 - Requisitos funcionais identificados para o *InstantDP***

<b>ID</b>	<b>Requisito</b>	<b>Dependência</b>
RF01	Deve ser possível cadastrar <sup>11</sup> uma empresa, com seu nome e e-mail.	
RF02	Deve ser possível cadastrar um plano de carreira de uma empresa, informando o nome, todos os cargos, níveis de senioridade de cada cargo, grupo de <i>skills</i> necessários para cada cargo e o nível de cada uma dessas <i>skills</i> para cada senioridade de um cargo.	RF01
RF03	Deve ser possível cadastrar os membros de uma empresa, informando seu nome, e-mail, cargo, nível de senioridade e papel.	RF02
RF04	Deve ser possível visualizar e cadastrar equipes, informando o nome, líder e os membros da equipe.	RF03
RF05	Deve ser possível cadastrar um ciclo de PDI, informando o membro da empresa, a data de início, data de fechamento e <i>status</i> .	RF03
RF06	Deve ser possível cadastrar um <i>personal mapping</i> , informando o ciclo de PDI, a trilha, pontos fortes e pontos fracos.	RF05
RF07	Deve ser possível cadastrar um <i>skills mapping</i> , informando o ciclo de PDI, as <i>skills</i> e o nível de cada <i>skill</i> .	RF05
RF08	Deve ser possível cadastrar <i>skills</i> priorizadas, informando o ciclo de PDI e as <i>skills</i> priorizadas.	RF06, RF07

<sup>11</sup> O termo *cadastrar* se refere às quatro operações básicas de um CRUD (Create, Read, Update, Delete)

RF09	Deve ser possível cadastrar ações para <i>skills</i> priorizadas, informando o ciclo de PDI, a <i>skill</i> priorizada, o título, descrição, link, duração e <i>status</i> .	RF08
RF10	Deve ser possível acompanhar o progresso de ações, visualizando a quantidade de horas completadas, quantidade de horas ainda a completar, quantidade de ações em cada status e quantidade de ações com <i>status Done</i> por semana.	RF09
RF11	Deve ser possível acompanhar o ciclo de PDI dos membros de uma equipe.	RF04, RF09
RF12	Deve ser possível acompanhar métricas relacionadas a uma equipe, visualizando as <i>skills</i> mais priorizadas e as <i>skills</i> com melhor avaliação de cada cargo.	RF04, RF09

Os requisitos RF02, RF03 e RF04 não foram implementados no MVP desenvolvido por Ferreira (2021). Para utilização da ferramenta era necessário que as informações referentes a esses requisitos já estivessem cadastradas no banco de dados. Esses requisitos foram tratados neste trabalho, possibilitando o uso da ferramenta do começo ao fim.

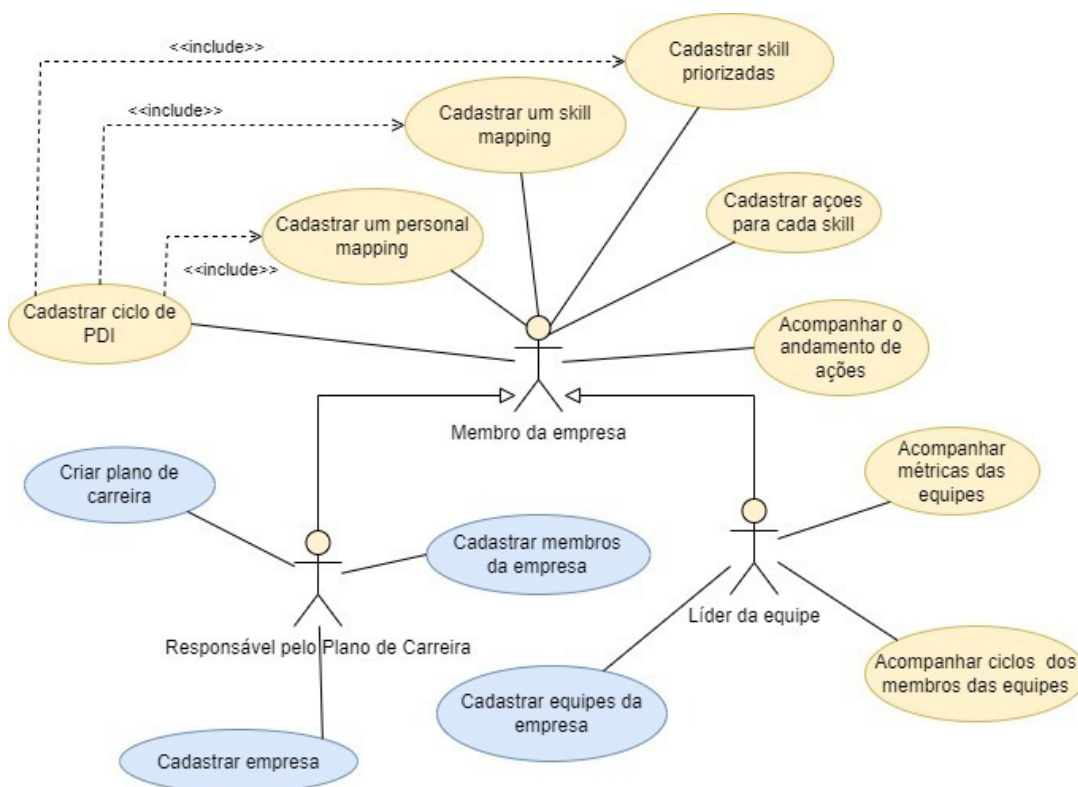
Em adição aos requisitos funcionais, foi definido o requisito não-funcional **usabilidade**. Em (FERREIRA, 2021) o foco foi nos requisitos funcionais, portanto não foram identificados requisitos não-funcionais. Neste trabalho, também foi considerado como requisitos não-funcionais que o sistema deve ter uma boa usabilidade e que isso deve ser alcançado a partir do atendimento às heurísticas de Nielsen.

### 3.3 Casos de uso

Casos de uso representam as funcionalidades do sistema. Por meio dos casos de uso usuários e clientes conseguem visualizar qual a funcionalidade a ser provida pelo sistema por meio de suas descrições textuais, conseguindo reagir mais rapidamente no sentido de refinar, alterar ou rejeitar as funções previstas para o sistema (COCKBURN, 2005). Conforme definido em (FERREIRA, 2021) os seguintes atores interagem com a ferramenta:

- **Responsável pelo Plano de Carreira:** Papel responsável pelo cadastro dos planos de carreira da empresa, contendo as posições com suas respectivas senioridades e *skills*.

- **Membro da Empresa:** Papel atribuído a todas as pessoas da empresa. Todos têm a possibilidade de criar e acompanhar seu próprio PDI.
- **Líder de Equipe:** O líder é responsável por criar e adicionar membros às equipes, acompanhar o PDI de seus liderados e acompanhar suas métricas.



**Figura 9 - Diagrama de casos de uso de *InstantDP*.**

O diagrama da Figura 9 apresenta todos os casos de uso previstos para a ferramenta *InstantDP* em (FERREIRA, 2021). Os objetivos desses casos de uso são manter os recursos que apoiam a realização das principais funcionalidades do sistema, que serão apresentadas a seguir. Entretanto, o MVP não implementava todas as funcionalidades descritas nestes casos de uso. O diagrama da Figura 9 apresenta em amarelo as funcionalidades implementadas no MVP e em azul as funcionalidades implementadas neste trabalho. Vale ressaltar que para atender o novo requisito não-funcional definido, o qual diz respeito à usabilidade da ferramenta, todos os

casos de uso que foram implementados originalmente por Ferreira (2021) - casos de uso indicados em amarelo na Figura 9 - foram aprimorados neste trabalho .

O Membro da Empresa é o ator responsável por todos os casos de uso ligados ao processo de definição e acompanhamento de um PDI, começando por **Cadastrar um Ciclo de PDI**. Para que esse ciclo possa ser realizado, é necessário que o ator siga as quatro etapas citadas na seção 3.1, ou seja, o ator precisa **Cadastrar um *Personal Mapping*** escolhendo a trilha de carreira que deseja seguir e mapeando seus pontos fortes e pontos fracos, **Cadastrar um *Skill Mapping*** avaliando em que nível se encontra em cada *skill* esperada do cargo e senioridade. Dado isso, o ator necessita **Cadastrar Skills Priorizadas** e **Cadastrar Ações para cada Skill**, priorizando as *skills* que serão desenvolvidas e potencializadas no próximo ciclo, e todas as ações que serão feitas para essas *skills*. Além da parte referente ao cadastro do PDI, é de responsabilidade deste ator **Acompanhar Andamento das Ações** para que possa ver sua evolução e se o objetivo está sendo atingido.

O Líder de Equipe é responsável por **Acompanhar Ciclo de Membro de Equipe** e **Acompanhar Métricas de Equipe** visando o desempenho dos membros de suas equipes. Este ator também deve **Cadastrar Equipe da Empresa**, sendo responsável pelo gerenciamento de suas equipes e dos membros que as compõem. É papel do Líder **Cadastrar Membro da Empresa**, cuja responsabilidade é dividida com Responsável pelo Plano de Carreira.

O Responsável pelo Plano de Carreira deve **Cadastrar Empresa** e **Cadastrar Plano de Carreira**, fornecendo todas as informações necessárias sobre as posições existentes na empresa e suas senioridades; quais *skills* são necessárias e em qual nível para cada senioridade, para que a partir do Plano de Carreira, o membro possa criar seu PDI. Vale notar que tanto o Responsável pelo Plano de Carreira quanto o Líder de Equipe são Membro da Empresa, herdando todas as responsabilidades.

### 3.4 Modelo Estrutural

Tomando como base os requisitos, os casos de uso e suas descrições, é possível fazer uma modelagem estrutural, capturando os conceitos e relacionamentos envolvidos no domínio em um diagrama de classes (BARCELLOS, 2018).

O diagrama de classes da ferramenta *InstantDP* exibido na Figura 10 foi criado por Ferreira (2021) de modo a suprir o processo de definição e acompanhamento de um PDI, e neste trabalho não foi necessário fazer modificações.

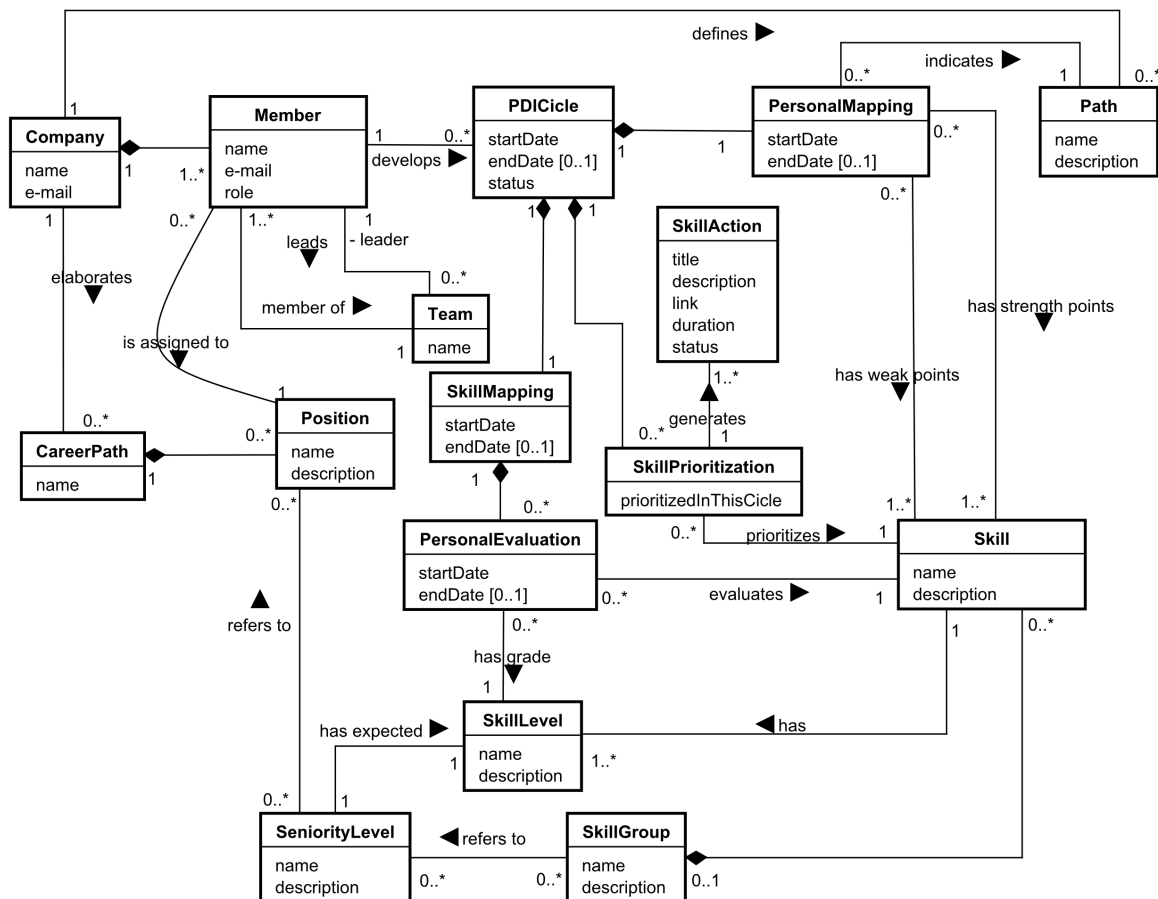


Figura 10 - Diagrama de classes de *InstantDP* (FERREIRA, 2021).

### 3.5 Projeto de Sistema

A ferramenta *InstantDP* foi desenvolvida com a utilização das tecnologias apresentadas na seção 2.4. A seguir são apresentadas informações gerais da arquitetura e componentes de *InstantDP* definidos em (FERREIRA, 2021), indicando-se algumas alterações realizadas no contexto deste trabalho.



### 3.5.1. Arquitetura de Software

A ferramenta foi desenvolvida seguindo a arquitetura cliente-servidor. Dessa forma, ela foi dividida em dois subsistemas: o subsistema do cliente (*front-end*) e o subsistema do servidor (*back-end*).

O *front-end* foi implementado utilizando Angular, um *framework* que segue o padrão MVVM. Ele é composto pela Camada de Interface com o Usuário (CIU), que trata aspectos de exibição e interação com os usuários e corresponde aos componentes *View* e *ViewModel*. A CIU é composta pelo componente de interação humana (CIH) e pelo componente de controle de interação (CCI), que serão detalhados posteriormente.

O *back-end* foi desenvolvido utilizando NestJS, *framework* que segue o padrão MVC. Ele é composto pela camada lógica de negócio (CLN) correspondente à *Controller* e a Camada de Gerência de Dados (CGD), correspondendo à *Model*. A CGD, por sua vez, trata a persistência dos dados no banco.

Portanto, a arquitetura da ferramenta *InstantDP* combina os padrões MVC e MVVM, de modo que o *Model* e o *Controller* do MVC se complementam com o *ViewModel* e *View* do MVVM. A Figura 11 apresenta a divisão da arquitetura descrita acima e os elementos citados serão apresentados com mais detalhes nas próximas seções.

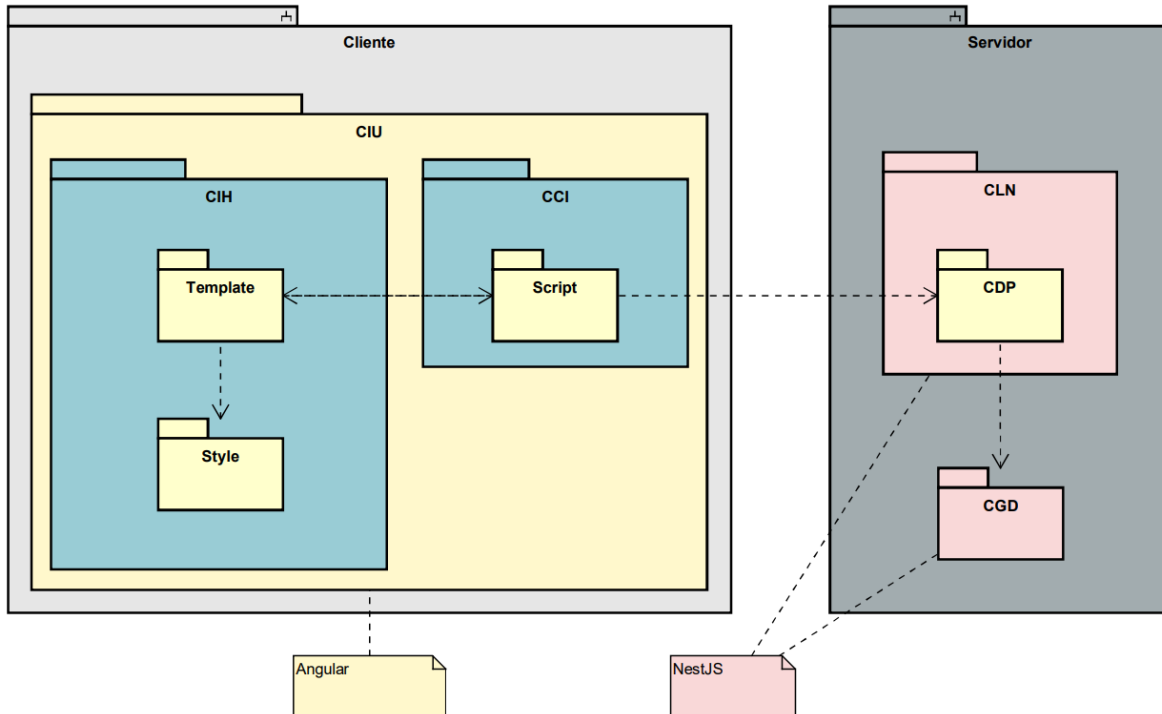


Figura 11 - Arquitetura da ferramenta *InstantDP* (FERREIRA, 2021).

### 3.5.1.1. Camada de Lógica de Negócio

O padrão de desenvolvimento da camada de lógica de negócio escolhida por Ferreira (2021) foi a de Modelo de Domínio. Segundo o padrão Modelo de Domínio, os dados e comportamentos são incorporados no próprio modelo de objetos de domínio. Dessa forma, as requisições da interface são recebidas pelo controlador que, para tratá-las, chama métodos das classes do domínio do problema (BARCELLOS, 2018).

#### 3.5.1.1.1 Componente de Domínio do Problema (CDP)

A Figura 12 apresenta o Componente de Domínio do Problema, modelo conceitual produzido por Ferreira (2021).

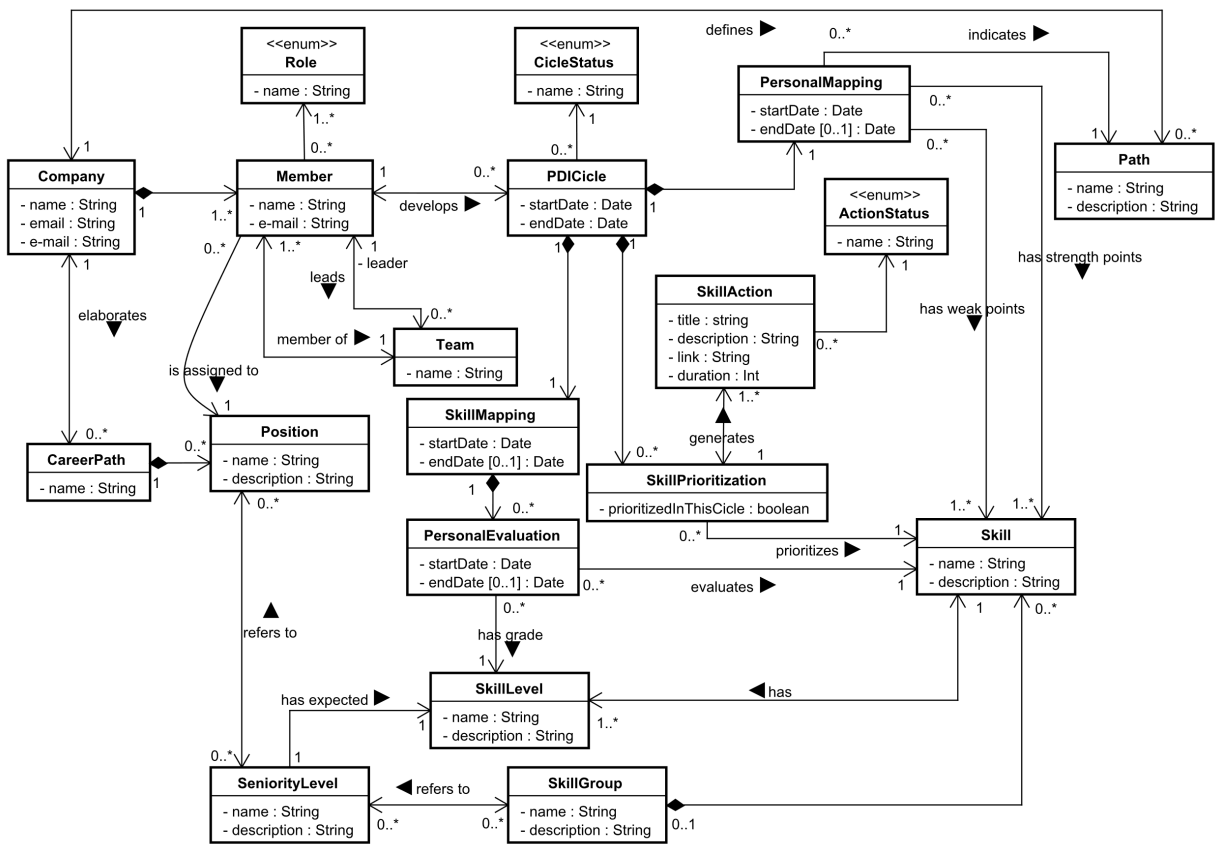


Figura 12 - Diagrama de Classes do CDP da ferramenta (FERREIRA, 2021).

É importante destacar que, apesar do diagrama da Figura 12 contemplar a ferramenta como um todo, a implementação realizada em (FERREIRA, 2021) simplificou algumas relações (por exemplo, as relações apontadas como muitos para muitos, que haviam sido simplificadas em relações um para muitos). Assim, neste trabalho foram necessários ajustes no código para que ele correspondesse ao diagrama da figura.

### 3.5.1.2. Camada de Interface com o usuário (CIU)

A interface com o usuário (IU) é uma parte essencial do projeto, pois ela estabelece uma forma de comunicação entre o sistema e as pessoas que o utilizam (BARCELLOS, 2018). A camada IU envolve duas partes: o Componente de Interação Humana e o Componente de Controle de Interação.

### 3.5.1.2.1 Componente de Interação Humana

O Componente de Interação Humana (CIH) compreende toda a parte visual do sistema como formato de janelas, formulários, relatórios etc. O CIH representa o *View* no modelo MVC e deve ser mantido separado do resto da implementação do sistema.

O CIH compreende os subcomponentes *Template* e *Style*. O *Template* consiste no código HTML, nele são declarados todos os elementos a serem exibidos, seu posicionamento na tela e a chamada de métodos e dados do CCI. O *Style* corresponde ao código de estilização, feito em SCSS.

Neste trabalho foram realizadas algumas melhorias nesse componente da arquitetura da ferramenta. Ao criar um CIH é importante pensar em uma experiência de qualidade ao usuário. Para isso, pode-se seguir algumas *guidelines*. Neste trabalho, decidiu-se utilizar as heurísticas de Nielsen para avaliar a experiência do usuário e identificar pontos de melhoria no CIH.

Ao analisar a interface do MVP da ferramenta, constatou-se que algumas heurísticas não foram atendidas em sua totalidade. As principais heurísticas alvo de melhorias neste trabalho foram: **Visibilidade do Status do Sistema**, **Prevenção de erros** e **Recuperação diante de erros**.

Quando se fala em **Visibilidade do Status do Sistema**, o sistema deve sempre dar *feedbacks* instantâneos ao usuário sobre a tela em que ele se encontra e ações realizadas. As ações realizadas na *InstantDP* como submeter formulários, por exemplo, não apresentavam *feedbacks* mostrando se a ação foi ou não concluída com sucesso. Além disso, alguns elementos clicáveis como botões não apresentavam *hover*.

Ainda no contexto de formulários, os campos não eram validados de forma a mostrar quando o usuário submete um formulário sem preencher todos os campos ou com algum valor inválido. O sistema também não exibia nenhum *feedback* quando alguma requisição apresentava erros. Esses dois problemas são exemplos de como a implementação não segue a heurística de **Recuperação diante de erros**.

O sistema também não atuava na **Prevenção de erros**. Por exemplo, não havia confirmação de ações nem sinalização de que o usuário não finalizou uma tarefa e está saindo da página.

Para atender a essas heurísticas, foram alterados alguns pontos na interface:

- Foi incluído o efeito *hover* nos elementos clicáveis.
- Em todos os formulários foram incluídas validações nos campos, de forma a avisar para o usuário quando não for preenchido um campo obrigatório ou quando um campo for preenchido de forma inválida.
- Ao submeter um formulário, foram adicionados *feedbacks* informando que a ação foi concluída com sucesso ou avisando que houve um erro com mensagem auxiliar para que o usuário possa reconhecer a situação e se recuperar dela.
- Foi adicionado um aviso caso o usuário tente sair da tela no meio de algumas ações.
- Para ações importantes foi incluída uma opção de confirmação antes da ação ser realizada.

#### **3.5.1.2.2 Componente de Controle de Interação**

O Componente de Controle de Interação (CCI) é responsável pela comunicação da interface do usuário com as regras de lógica de negócio. No Angular, ele é composto pelo arquivo Typescript do componente. Nele são implementados todos os métodos que conectam as ações dos usuários no CIH às ações correspondentes do CLN através de uma API (*Application Programming Interface*).

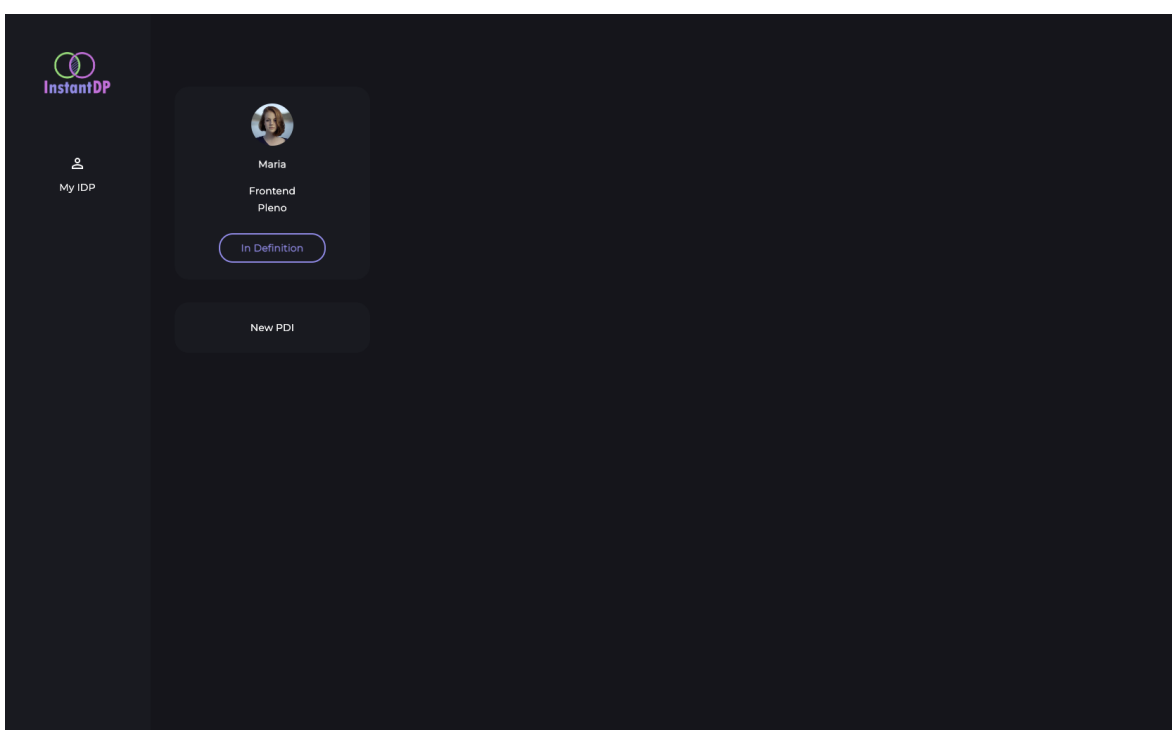
#### **3.5.1.3. Camada de Gerência de Dados**

Para a persistência de dados foi utilizado MySQL, um banco de dados relacional, através do *framework* NestJS. Foi utilizada a biblioteca de integração de banco de dados TypeORM, que é indicada pelo próprio NestJS para implementações com Typescript.

### **3.6 Ferramenta *InstantDP* após as modificações**

Nesta seção serão apresentadas as evoluções realizadas na ferramenta, abordando as telas já existentes (perspectiva de um usuário que atua como Membro da Empresa e de um usuário que atua como Líder de Equipe) bem como a nova perspectiva criada, de Responsável pelo Plano de Carreira.

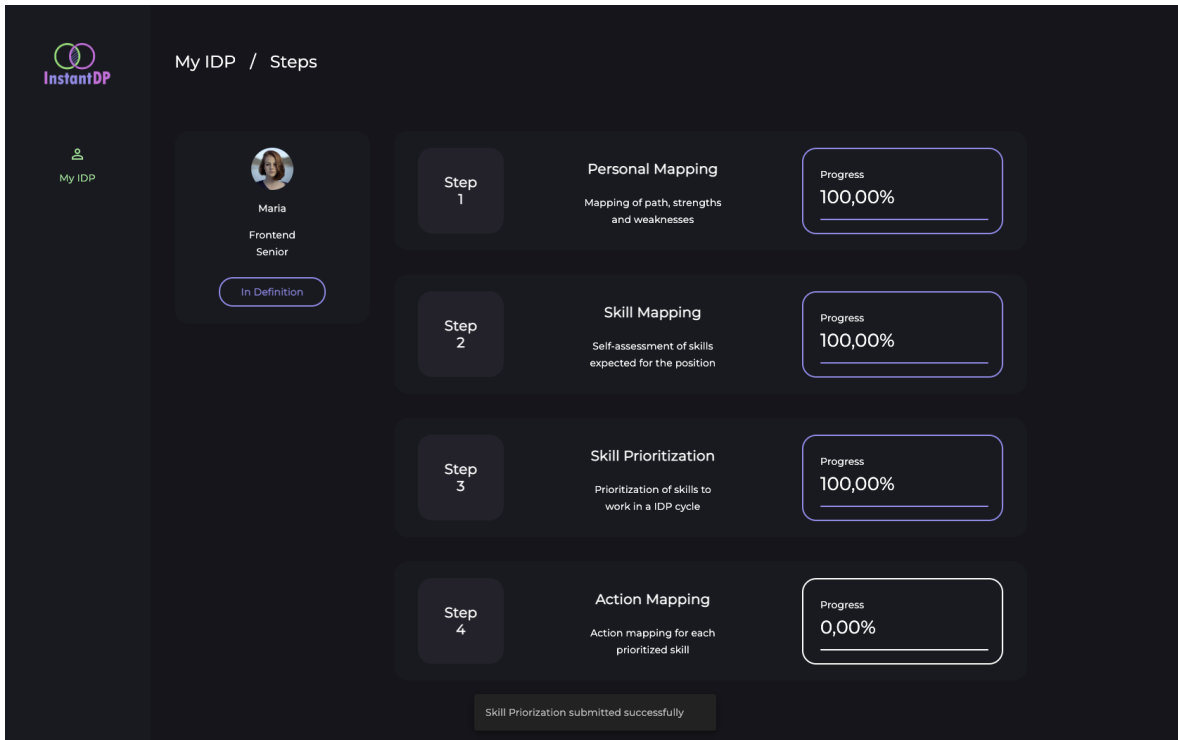
Na perspectiva de um Membro da Empresa, a primeira tela é a tela inicial de definição de um PDI. Nesta tela é possível visualizar em qual etapa da definição do PDI a pessoa se encontra e caso não haja ainda um processo de PDI em definição, mostra-se um botão para criar um PDI (Figura 13).



**Figura 13 - Página inicial de um membro, sem PDI.**

Para seguir as Heurísticas de Nielsen e pontos de melhoria apontados na seção 3.5.1.2.1, foram feitas algumas modificações no fluxo de criação de um PDI:

- Foram adicionados feedbacks ao submeter um formulário (Figura 14).



**Figura 14 - Exemplo de *feedback* ao submeter uma etapa.**

- Foram adicionadas validações nos formulários e efeito *hover* nos itens clicáveis, como mostra a figura 15, onde o usuário não preencheu campos obrigatórios e por isso os campos inválidos apresentam uma mensagem avisando que o campo está inválido, e o botão está com o efeito do *hover*. Além disso, os campos de formulário constavam apenas *placeholder*, que some quando o usuário começa a digitar, fazendo com que possa gerar algum tipo de confusão. Por isso, foram adicionadas *labels* aos campos dos formulários, para que ao começar a digitar, o usuário não perca a informação à qual o campo se refere (Figura 15).

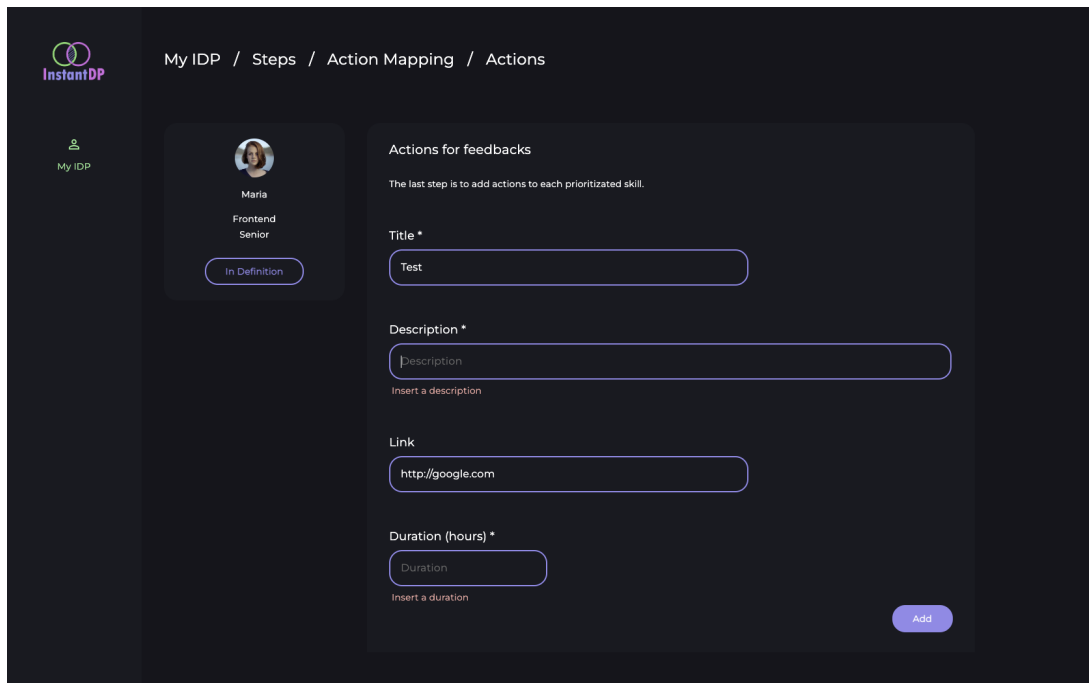


Figura 15 - Exemplo de validação de formulários, *hover* no botão e inclusão de *labels*.

Quanto à apresentação das informações ao usuário, foram adicionadas informações do nível atual e do nível esperado na etapa de *Skill Priorization*, de forma a usuário ter mais informações para decidir qual *skill* priorizar (Figura 16).

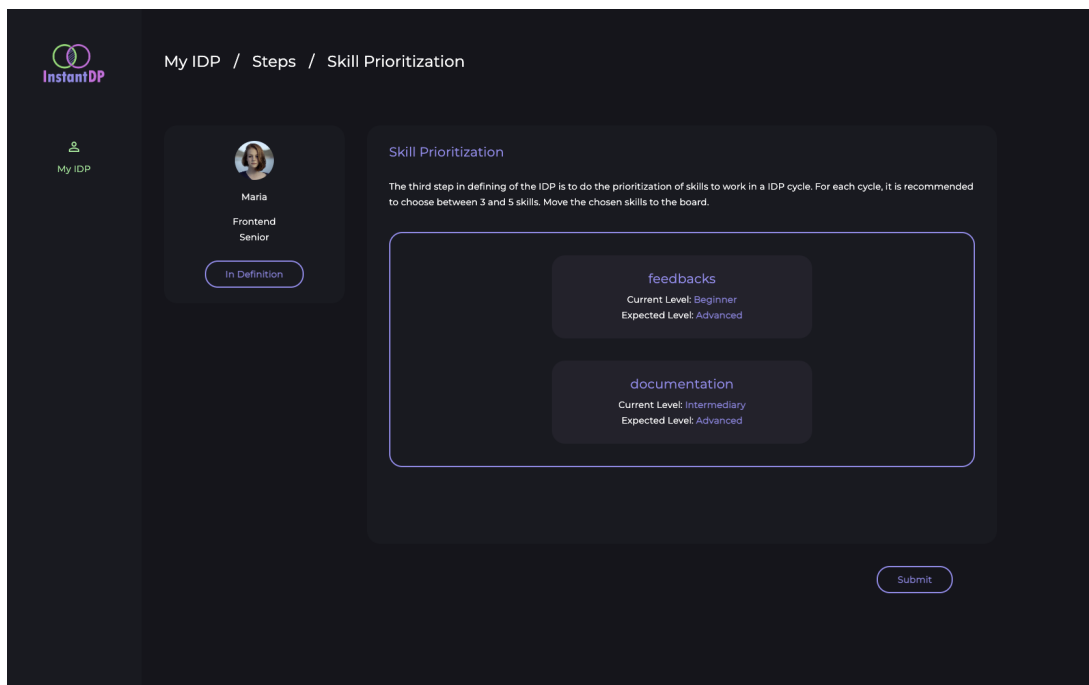
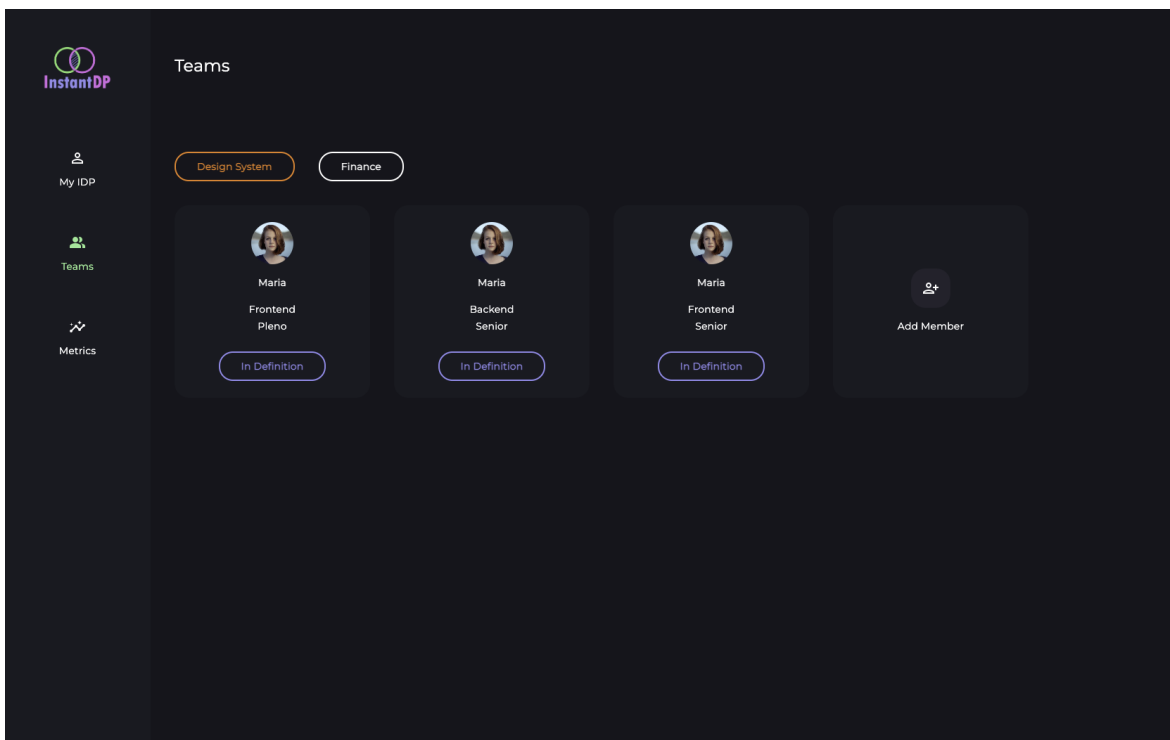


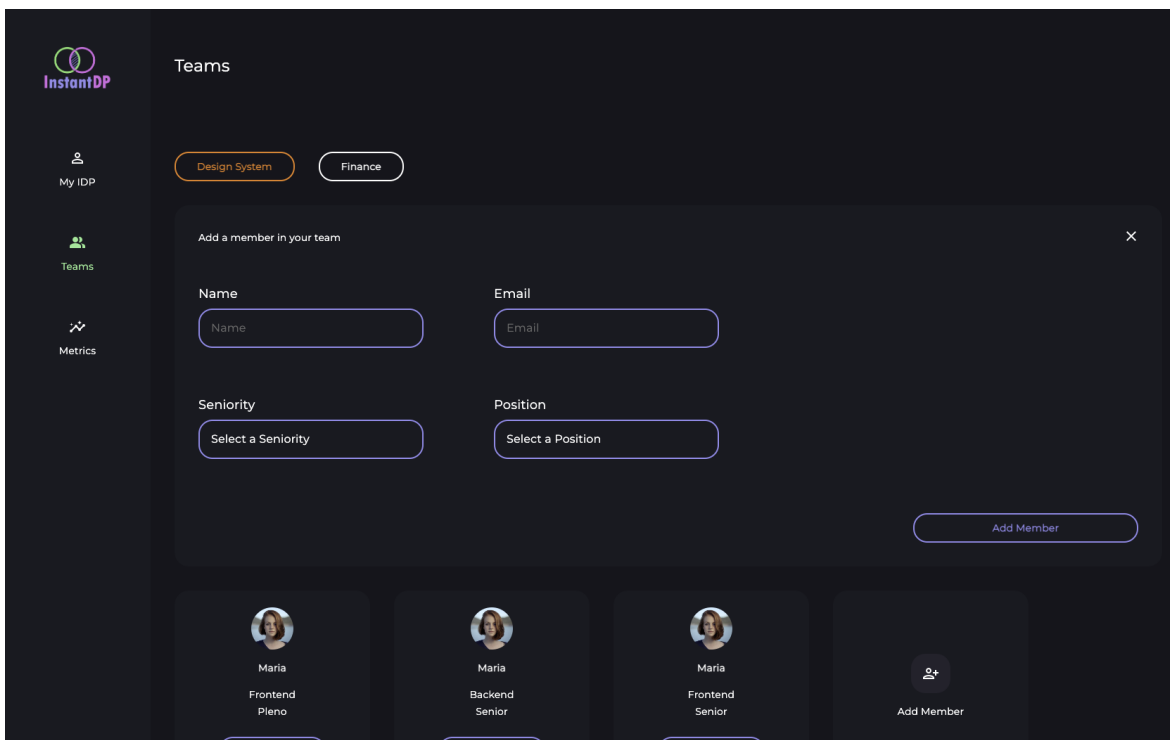
Figura 16 - Etapa Skill Priorization.



Na perspectiva de um usuário que atua como Líder de Equipe, além do menu lateral conter a opção *My IDP*, existem também as opções *Teams* e *Metrics*. Na opção *Metrics*, é possível realizar o acompanhamento das métricas de uma equipe, e não houve alteração nesse fluxo. Clicando na opção *Teams* o líder pode visualizar todas as equipes nas quais exerce o papel de líder com seus respectivos membros (Figura 17). Nesta tela, foi implementado o fluxo de adicionar um novo membro à equipe (Figura 18).



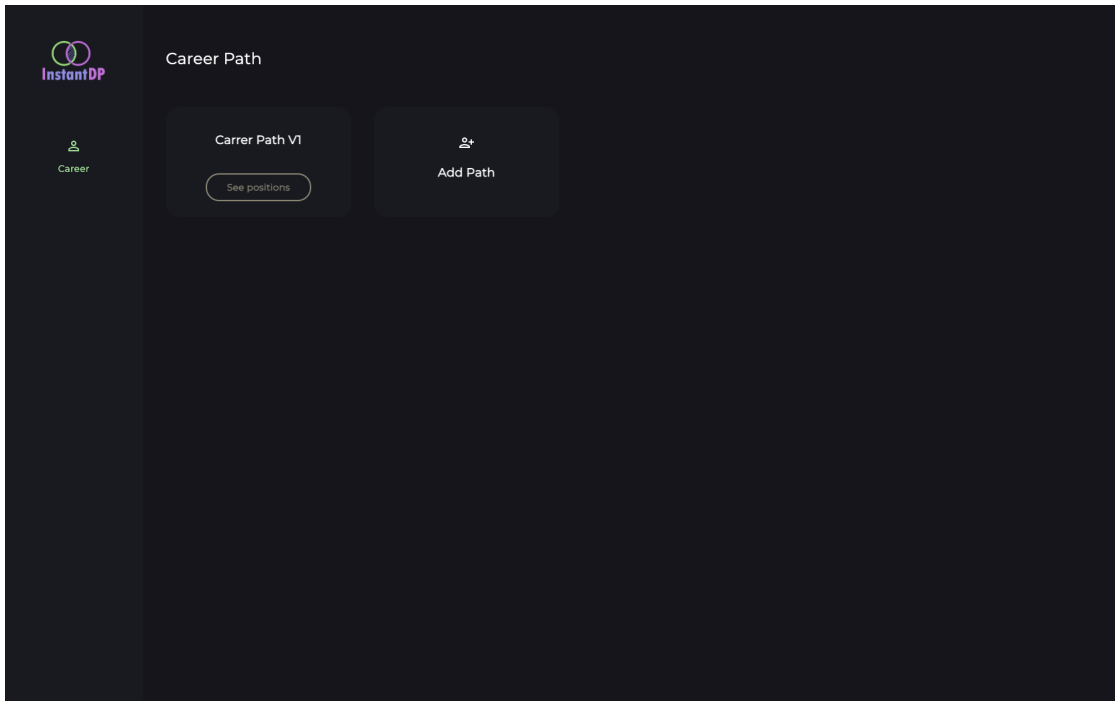
**Figura 17 - Tela de equipes de um líder.**



**Figura 18 - Adicionar um novo membro.**

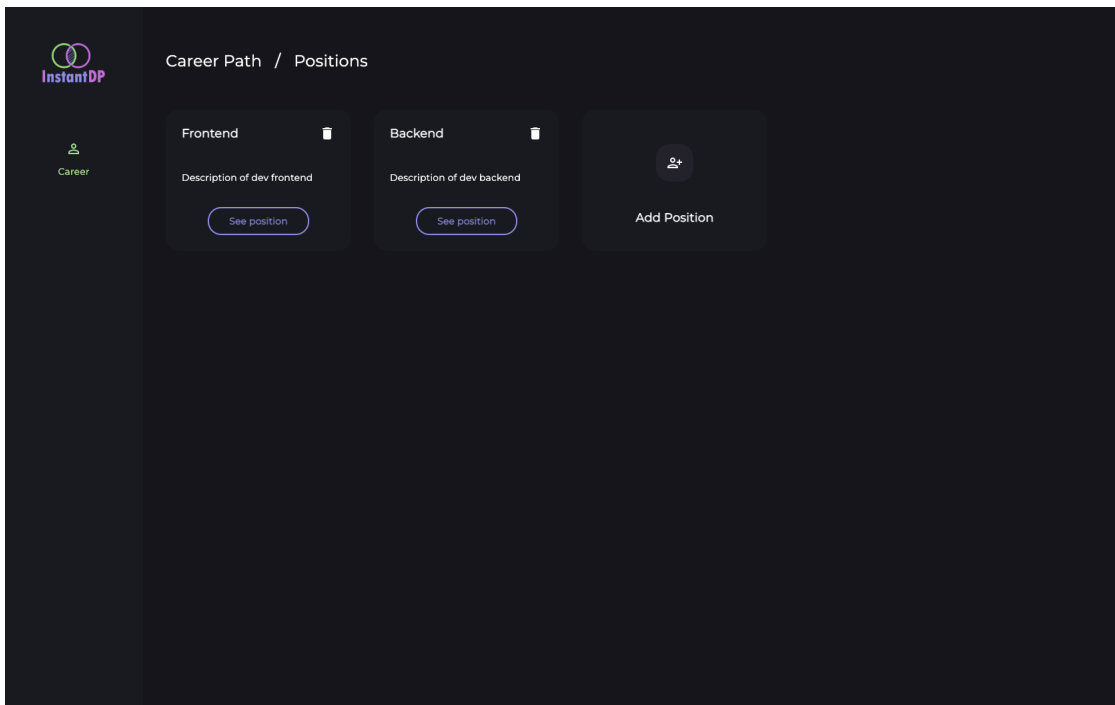
Para que toda a criação e acompanhamento dos PDIs sejam feitos, é necessário que a empresa contenha um plano de carreira bem estruturado, que mostre ao funcionário o que é esperado dele.

Na nova perspectiva criada, de Responsável pelo Plano de Carreira, a tela inicial mostra os planos de carreira existentes na empresa (Figura 19). A partir desta tela, é possível ver e editar as informações dos planos de carreira da empresa, bem como criar um novo plano.



**Figura 19 - Tela de planos de carreira.**

Ao clicar em ver um plano de carreira, o usuário é levado para a tela que contém todas as posições daquele plano de carreira (Figura 20). Nesta tela, é possível adicionar, excluir ou ver os detalhes de uma posição.



**Figura 20 - Tela de posições.**

Ao clicar no botão de visualizar posição, o usuário é redirecionado para uma tela (Figura 21) onde consegue ver e, se necessário, editar as informações da posição escolhida. Nesta tela é possível visualizar as informações básicas da posição, como nome e descrição, e as senioridades desta posição. Ao selecionar uma senioridade, são mostrados nome, descrição e os grupos de *skill*. Dentro de um grupo de *skill*, mostram-se as *skills*, cada uma com seu nome, descrição e nível esperado para a senioridade visualizada.

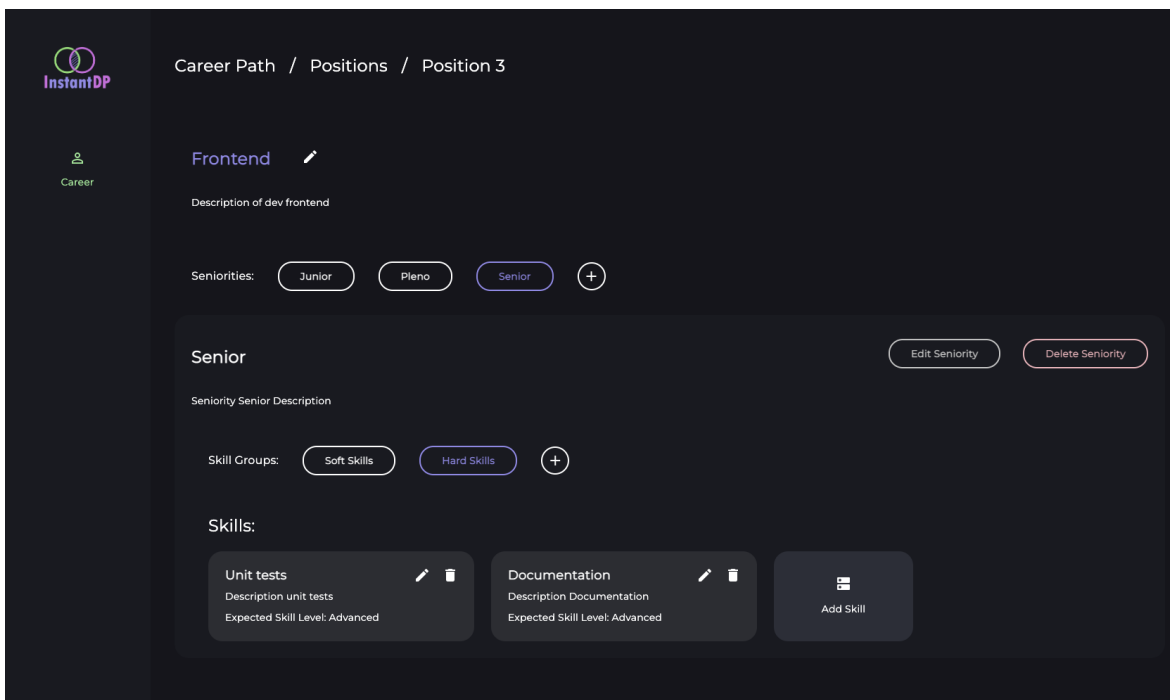
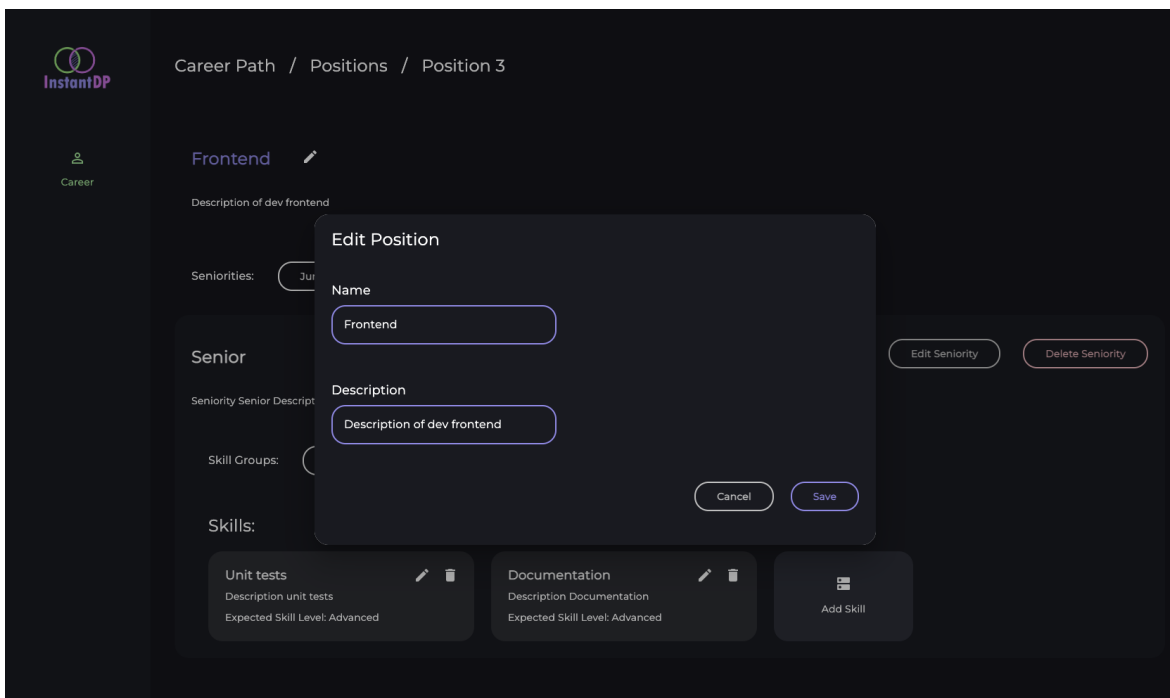


Figura 21 - Tela de detalhes de uma posição

Para que o plano de carreira seja editado conforme necessidade, sem criar um formulário muito extenso, optou-se por implementar uma tela onde as informações estão dispostas em seções diferentes, permitindo serem editadas separadamente.

Clicando no ícone de edição ao lado do nome da posição, é possível editar as informações básicas da posição como nome e descrição, como mostra a figura 22.



**Figura 22 - Edição de uma posição.**

As senioridades contidas na posição podem ser exibidas clicando no botão que contém o nome da senioridade (Figura 21). Ao lado das senioridades existentes, é apresentado um botão +, onde é possível adicionar uma nova senioridade (Figura 23). Também é possível editar uma senioridade clicando no botão *Edit Seniority* (Figura 24) ou ainda excluir a senioridade clicando em *Delete Seniority* (Figura 25).

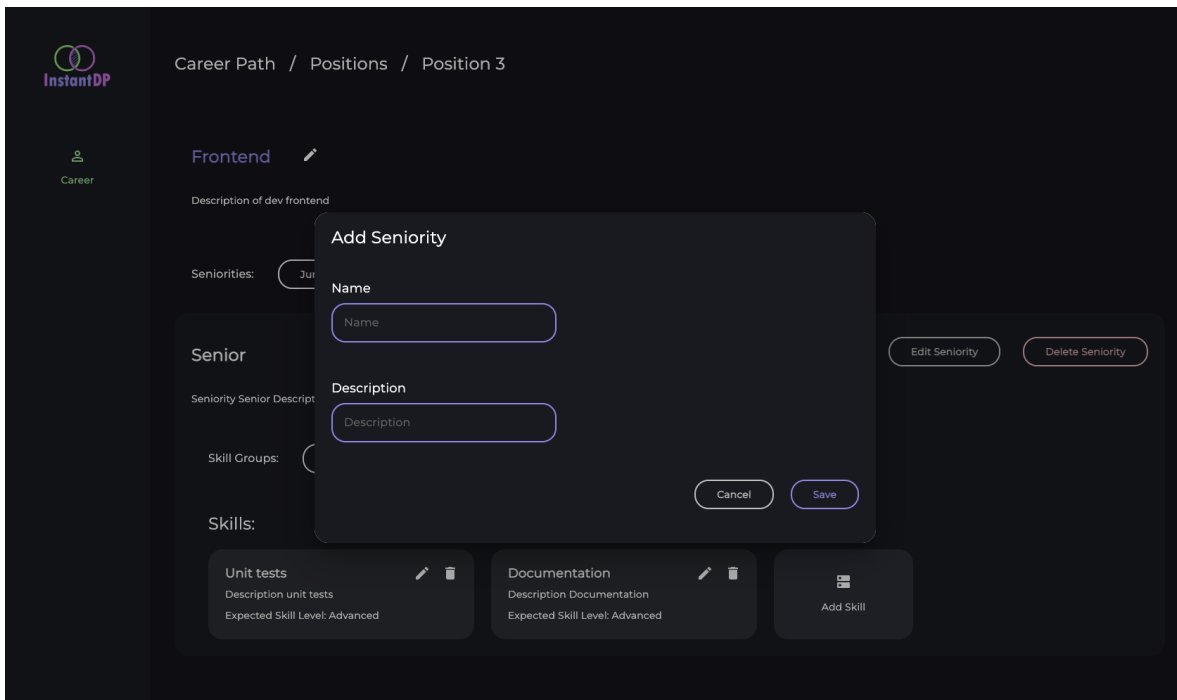


Figura 23 - Adicionar uma senioridade.

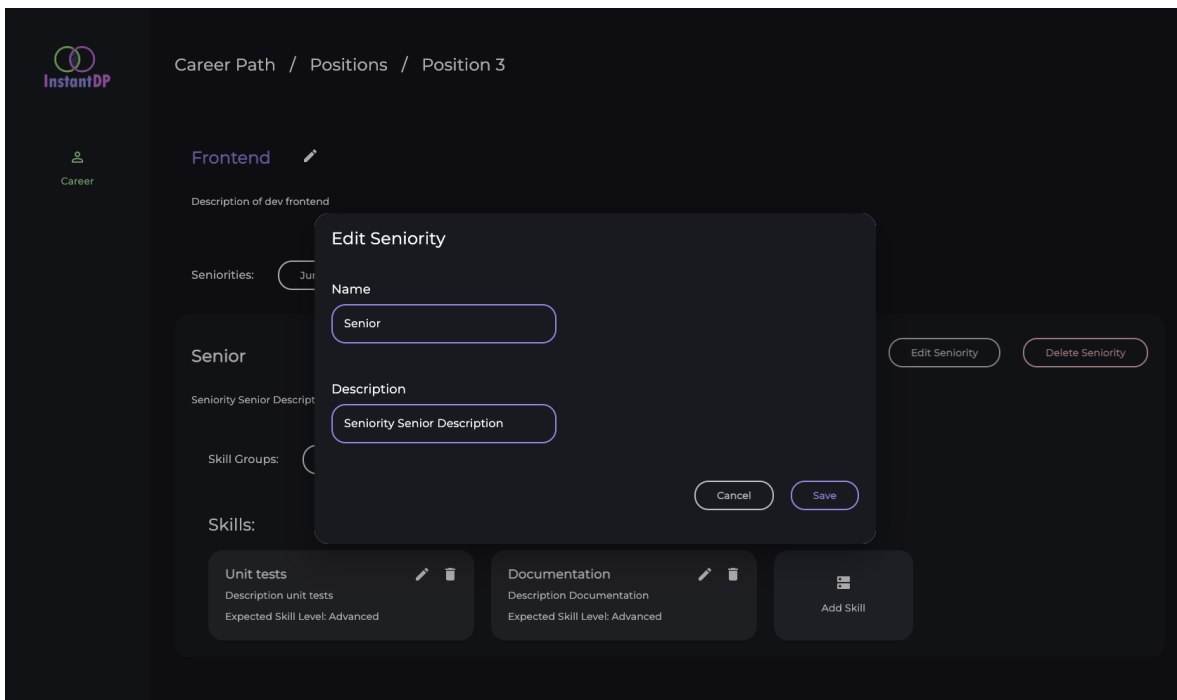
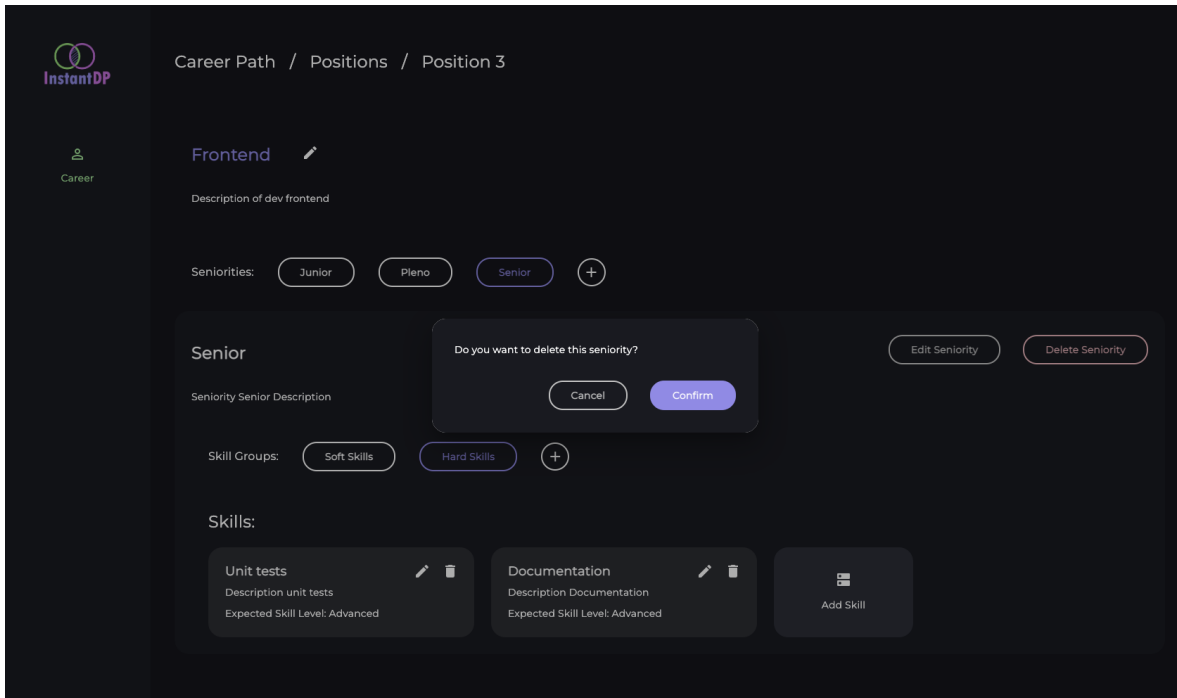


Figura 24 - Edição de uma senioridade.



**Figura 25 - Exclusão de uma senioridade.**

Para fazer a exclusão de uma senioridade, é necessário que não tenham membros nesta senioridade. O mesmo vale para posições. Todas as ações de exclusão possuem um modal de confirmação, para evitar que o usuário faça uma ação indevida, seguindo as Heurísticas de Nielsen.

Ainda nessa tela, é possível visualizar todas as *skills* necessárias para uma determinada senioridade. Cada *skill* contém níveis como por exemplo, iniciante, intermediário e avançado, e um nível esperado para o funcionário que está na senioridade em questão. Os níveis são importantes para a construção do PDI e para avaliar se o funcionário está de acordo com a função em que se encontra. Para cada *skill*, é possível realizar a edição (Figura 26) e sua exclusão (Figura 27) através dos ícones presentes no card da *skill*. Também é possível adicionar uma nova *skill* clicando no card *Add Skill*, como mostra a Figura 28.

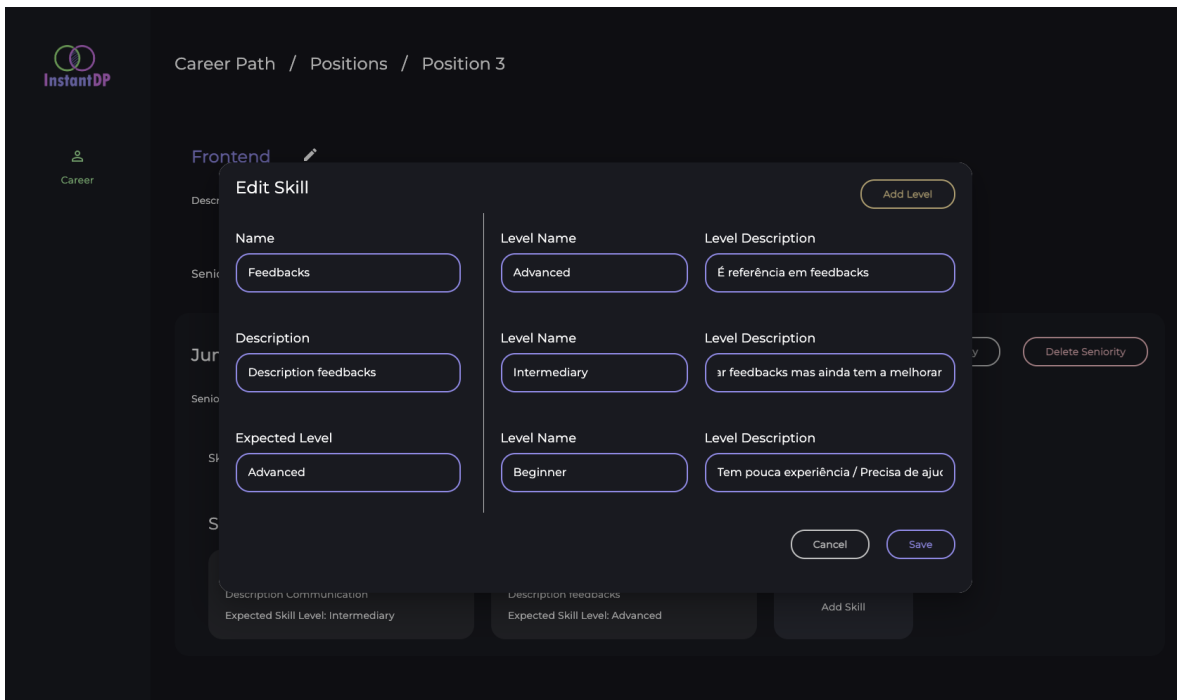


Figura 26 - Edição de uma *skill*.

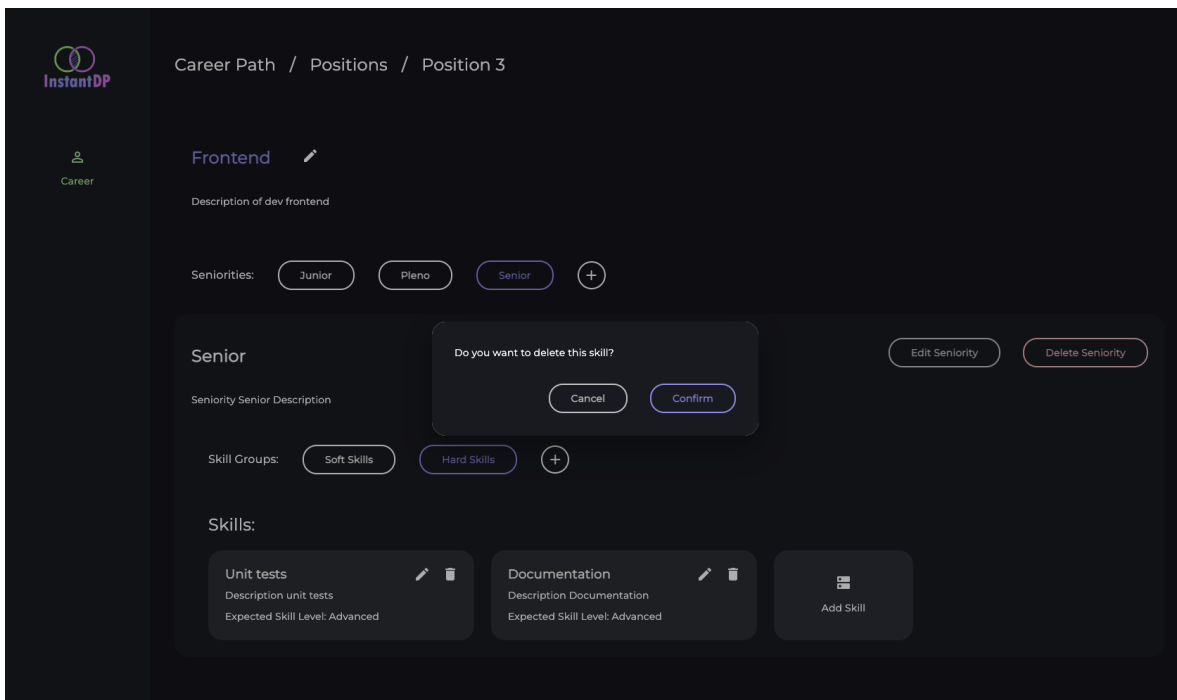


Figura 27 - Exclusão de uma *skill*.



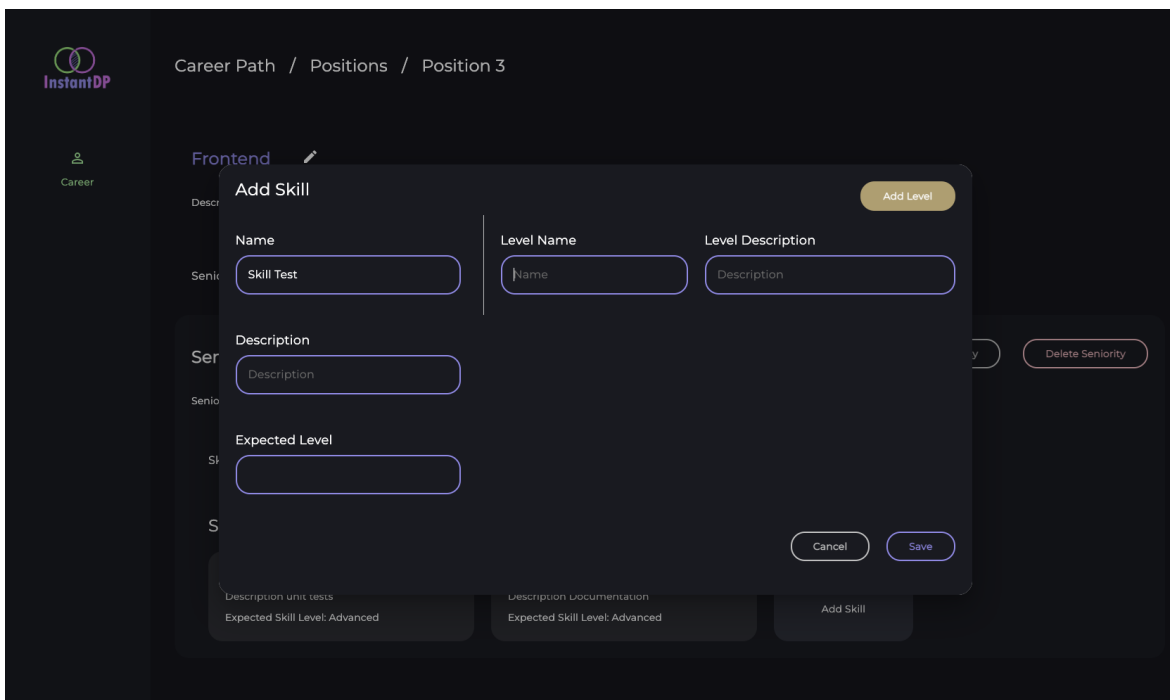


Figura 28 - Adição de uma *skill*.

### 3.7 Testes

Para avaliar a ferramenta, foram realizados testes de usabilidade. Testes de usabilidade têm como objetivo observar usuários reais usando a interface para descobrir problemas e pontos de melhorias.

Ao longo do desenvolvimento do trabalho os testes de usabilidade foram realizados com a participação de colaboradores do PicPay. Os testes tiveram foco na utilização do sistema e verificação do atendimento aos requisitos funcionais implementados considerando a nova perspectiva criada - Responsável pelo Plano de Carreira - e nos pontos que foram melhorados para a criação de um PDI.

Os testes foram essenciais para a percepção de possíveis melhorias nos fluxos adicionados e na validação da ferramenta em relação às Heurísticas de Nielsen.

### 3.8 Considerações Finais do Capítulo

Este capítulo apresentou a ferramenta *InstantDP*, incluindo algumas das definições feitas em (FERREIRA, 2021) e as melhorias implementadas neste trabalho. Além das melhorias baseadas nas Heurísticas de Nielsen, foram implementadas as funcionalidades mais

importantes indicadas como trabalhos futuros por Ferreira (2021), para viabilizar o uso da ferramenta na prática. Vale destacar que diferente do MVP desenvolvido em (FERREIRA, 2021), onde era necessário incluir dados diretamente no banco ou via ferramentas como Postman<sup>12</sup>, agora todas as requisições conseguem ser feitas através do *front-end*. No próximo capítulo são apresentadas as considerações finais do trabalho.

---

<sup>12</sup> <https://www.postman.com/>

# Capítulo 4

## Considerações Finais

---

*Neste capítulo são realizadas as considerações finais deste trabalho, sendo apresentadas suas principais contribuições e perspectivas de trabalhos futuros.*

### 4.1 Conclusão

A definição e institucionalização de um processo de planejamento e acompanhamento de desenvolvimento individual em equipes ágeis é um desafio para as empresas. A criação da ferramenta *InstantDP* em 2021 visou contribuir para a definição, padronização e apoio automatizado ao processo, porém a ferramenta apresentava limitações que impediam sua utilização em um ambiente real.

Durante o desenvolvimento deste trabalho foram feitas correções na arquitetura e implementados requisitos funcionais e não-funcionais visando à viabilização de uso da ferramenta. Para isso, este trabalho baseou-se na melhoria da interface de acordo com as heurísticas de Nielsen e em trabalhos futuros definidos em (FERREIRA, 2021).

Ao longo do desenvolvimento deste projeto, os conhecimentos adquiridos ao decorrer do curso foram fundamentais para o sucesso do desenvolvimento do trabalho, podendo-se destacar:

- Engenharia de Software: nessa disciplina foi tratado todo conhecimento necessário para o levantamento e análise de requisitos e modelagem de casos de uso, diagramas de classes e de projeto de sistema. Estes conhecimentos facilitaram a organização do tempo para desenvolvimento por prover previsibilidade quanto às funcionalidades necessárias para a ferramenta.
- Interface Humano-Computador: nesta disciplina foram apresentados os conceitos teóricos e práticos para o desenvolvimento de uma boa interface, como as heurísticas de Nielsen.
- Banco de Dados: esta disciplina deu a base para os conhecimentos em relação a modelagem e utilização de persistência em bancos de dados relacionais, como o utilizado nesse sistema.

- Programação II e III: estas disciplinas deram a base de lógica de programação e os paradigmas programação Orientada a Objetos e Estruturada, utilizadas na implementação deste trabalho.

No Capítulo 1 da monografia foram apresentados os objetivos a serem alcançados no desenvolvimento deste trabalho. Na tabela abaixo, são apresentados novamente cada um dos objetivos, assim como a indicação de atendimento e os resultados alcançados que evidenciam o atendimento dos mesmos.

**Tabela 2 - Objetivos e sua situação na conclusão da monografia**

<b>Objetivo</b>	<b>Status</b>	<b>Resultado</b>
Identificar e implementar melhorias na usabilidade do sistema, considerando as Heurísticas de Nielsen	Atendido	Melhorias na usabilidade descritas e implementadas na ferramenta (vide seção 3.2.1.2.1 e seção 3.6).
Identificar e implementar novas de funcionalidades de forma a ampliar o atendimento às necessidades relacionadas à criação e acompanhamento de PDIs	Atendido	Ferramenta <i>InstantDP</i> implementada (vide Seção 3.6).

Por fim, conclui-se que o desenvolvimento deste projeto final de graduação foi bem sucedido. Foi possível evoluir o MVP de forma a tratar os principais trabalhos futuros indicados em (FERREIRA, 2021). Vale destacar que apesar da evolução construída, a ferramenta ainda possui limitações que precisam ser tratadas para viabilizar seu uso em empresas e apresenta várias oportunidades de melhorias

Entre os desafios no desenvolvimento deste trabalho, destaca-se o aprendizado da tecnologia utilizada no backend (NestJS) e o fluxo de criação e edição de um plano de carreira, por se tratar de um fluxo complexo que precisava fazer sentido para o usuário da ferramenta e na implementação. Além disso, também foi necessário o aprendizado sobre gestão de pessoas, que foi extremamente enriquecedor.

## 4.2 Trabalhos futuros

A ferramenta *InstantDP* foi evoluída neste trabalho, mas há, ainda, diversas oportunidades para melhorias e criação de novas funcionalidades. Algumas melhorias foram identificadas por Ferreira (2021) na criação do MVP, outras pela autora deste trabalho ao decorrer do desenvolvimento. As melhorias identificadas são:

- Implementar telas de *login* e *logout*
- Implementar visualização de um ciclo de PDI finalizado, uma vez que hoje a ferramenta só contempla o caso de ciclo em desenvolvimento.
- Criar uma funcionalidade com métricas gerais da empresa.
- Criar funcionalidade de *feedbacks* por meio da etapa de *Skill Mapping* para pares e líder, com finalidade de validar o ciclo concluído e trazer insumos para a construção de novos ciclos.
- Criar uma visualização em que o membro da empresa saiba exatamente onde está no plano de carreira e o caminho que ainda precisa desenvolver.
- Criar funcionalidade de notificações, possibilitando que os líderes recebam notificações quando os membros de suas equipes completam as ações definidas.
- Salvar o histórico de cargos dos membros da empresa, com finalidade de visualizar a evolução dos membros.
- Incluir botões de ajuda em algumas telas da ferramenta com alguns FAQs que ajudem os usuários com dúvidas comuns e relevantes.
- Incluir funcionalidade que auxilie na validação do conhecimento do membro e/ou verificação do atendimento aos critérios de aceitação de uma *skill*, para avaliar se a referida *skill* foi realmente desenvolvida pelo membro.

## Referências Bibliográficas

AMORIM, Gabriel: Como conciliar agilidade e arquitetura de software. Disponível em: <<https://www.devmedia.com.br/como-conciliar-agilidade-e-arquitetura-de-software/33281>>.

Acesso em maio de 2022.

BARCELLOS, M.P: Engenharia de Software: Notas de aula. 2018. Disponível em: <<https://nemo.inf.ufes.br/wp-content/uploads/Monalessa/EngSoftware/NotasDeAula-EngSw-EngComp-v2018.pdf>>. Acesso em junho de 2022.

BECK, K. S. K. S. J. E. A.: Manifesto for agile software development. Disponível em: <<http://agilemanifesto.org/>>. Acesso em maio de 2022.

COCKBURN, A., Escrevendo Casos de Uso Eficazes: Um guia prático para desenvolvedores de software, Porto Alegre: Bookman, 2005.

FERREIRA, Ú. A., “Apoio ao Planejamento e Acompanhamento de Desenvolvimento Individual em Equipes Ágeis”, Projeto de Graduação, Colegiado do Curso Engenharia da Computação, Universidade Federal do Espírito Santo, Vitória-ES, Brasil, 2021.

FOWLER, M. The new methodology, 2005. Disponível em: <<https://martinfowler.com/articles/newMethodology.html>>. Acesso em julho de 2022.

GUIMARÃES, Bruna: Gestão de Pessoas: o que é, principais processos, objetivos e seus pilares. Disponível em: <<https://www.gupy.io/blog/gestao-de-pessoas>>. Acesso em junho de 2022.

MOREIRA, Karine: PDI: Guia completo para te ajudar a colocar esse plano em prática. Disponível em: <<https://mereio.com/blog/pdi/>>. Acesso em junho de 2022.

MENDES, Guilherme: O que é Kanban? Conheça os principais tipos e como utilizá-los. <<https://www.fm2s.com.br/tipos-de-kanban/>>. Acesso em maio de 2022.

NAISBITT, John: O Líder do Futuro. 11 conceitos essenciais para ter clareza num mundo confuso e se antecipar às novas tendências. Rio de Janeiro: Sextante, 2007.

ROSSETTI, Micaela: HEURÍSTICAS DE NIELSEN. Disponível em:  
<<https://softdesign.com.br/blog/heuristicas-de-nielsen>> Acesso em maio de 2021.