

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

Úrsula Abreu Ferreira

APOIO AO PLANEJAMENTO 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
Coorientadora: Jordana Sarmenghi Salamon

VITÓRIA
2021

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

Úrsula Abreu Ferreira

**APOIO AO PLANEJAMENTO E ACOMPANHAMENTO DE
DESENVOLVIMENTO INDIVIDUAL EM EQUIPES ÁGEIS**

COMISSÃO EXAMINADORA

Prof. Monalessa Perini Barcellos, D. Sc.

Prof. Rodrigo Laiola Guimarães, Ph. D.

Prof. Paulo Sérgio Santos Júnior, M. Sc.

Vitória, 15 de outubro de 2021

A Nilda, Nivaldo e Willian, que me apoiaram
e me incentivaram a conquistar
meus sonhos.

AGRADECIMENTOS

Aos meus pais, Nilda e Nivaldo, por todo amor, dedicação e por sempre me incentivaram nos momentos mais difíceis a não desistir de estudar e a ir atrás dos meus objetivos. Aos meus avós maternos, por sempre vibrarem com todas as minhas conquistas. Ao meu irmão Willian, por todo amor e carinho, por ser meu amigo, por ser a melhor influência e exemplo que eu poderia ter na vida, por sempre acreditar em mim e no meu potencial e por, mesmo distante, sempre estar presente. Tenho muito orgulho de você e suas conquistas.

A toda a minha família, por todo o amor, carinho e presença nos mais diversos momentos que a vida me proporcionou. Em especial, as minhas primas Carolina e Juliana, que apesar de primas, as considero como irmãs.

Ao longo de todos os anos da graduação eu tive o privilégio de criar amizades que levarei pelo resto da vida. Agradeço à Nicole, por ter embarcado comigo na aventura de sair da casa dos pais e por enfrentarmos juntas o momento de pandemia no mundo. À Beatriz, por ser um exemplo de dedicação e disciplina, por sempre me apoiar e por acreditar em mim quando eu mesmo não acreditava. Ao Gabriel, Felipe, Gilmarllen, Lucas, Lúcio e à Isabel, por todas as disciplinas que fizemos juntos, por todas as idas à cantina do CT, por todas as ajudas e trocas de conhecimento. Ao NCD por ter me dado a oportunidade do primeiro estágio.

Ao Leandro, Lucas, Alan, Matheus, Stefani e Ayoub pela amizade que começou em um jogo e que hoje fazem parte da minha vida, por todas as horas jogando juntos, por toda a diversão e por todo apoio que sempre me deram.

À minha coorientadora, Jordana, por sua amizade, por toda paciência, apoio e dedicação ao longo deste trabalho. À minha orientadora Monalessa pela oportunidade, apoio e compreensão com os problemas de disponibilidade. Aos professores João Paulo, Vitor, Patrícia, Roberta, Rosane e Eliete por serem professores incríveis e que sempre me inspiraram a buscar mais conhecimento.

RESUMO

Com o surgimento das ideias de agilidade pautadas no Manifesto Ágil, muitas empresas começaram a adotar as metodologias ágeis no desenvolvimento de software para lidar com a constante mudança dos requisitos, trazer rapidez, aumentar a qualidade e reduzir os custos do processo de desenvolvimento. Os benefícios gerados pelas metodologias ágeis tornam sua utilização atraente para gestores e diretores de empresas, porém é necessário que o foco da aquisição de recursos humanos não seja somente em pessoas com alto grau de habilidades técnicas para desenvolver os projetos, e sim garantir equipes compostas de profissionais com diversos tipos de habilidades. Assim, a gestão de membros de equipes multidisciplinares em empresas que adotam metodologias ágeis não se limita apenas a garantir a execução de tarefas com qualidade e rapidez. Ainda é necessário conhecer os membros da equipe e ser responsável por acompanhar a carreira de cada um, oferecendo apoio em todo o processo de construção de um plano de desenvolvimento individual. Muitas empresas, como a empresa PicPay, utilizam o PDI (Plano de Desenvolvimento Individual) como forma de evolução e acompanhamento de carreira, mas não existe uma padronização da interação entre líder e membros da equipe, trazendo heterogeneidade ao processo de definição e acompanhamento de um PDI dentro das empresas.

Nesse contexto, este trabalho estrutura um processo de definição e acompanhamento de um PDI com o objetivo de resolver os problemas encontrados e garantir que todos os membros da equipe tenham um PDI que seja efetivo, atenda às necessidades identificadas e seja simples de ser aplicado. Neste trabalho também foi desenvolvida a ferramenta InstantDP para apoiar o processo definido. O processo foi utilizado no desenvolvimento de InstantDP como um recurso de conhecimento sobre o domínio. Um MVP (*Minimum Viable Product*) da ferramenta foi projetado, desenvolvido e disponibilizado para uso, incluindo funcionalidades que apoiam a definição e acompanhamento de PDI.

Palavras-chave: Metodologias Ágeis, Gestão de Pessoas, Plano de Desenvolvimento Individual

LISTA DE FIGURAS

Figura 1 - Diagrama representando o padrão MVC (feito pela autora).	18
Figura 2 - Diagrama representando o padrão MVVM (feito pela autora).	20
Figura 3 - Etapas do processo proposto.	24
Figura 4 - Representação das tarefas da etapa <i>Personal Mapping</i>	24
Figura 5 - Diagrama de casos de uso de InstantDP.	31
Figura 6 - Diagrama de classes de InstantDP.....	33
Figura 7 - Arquitetura da ferramenta InstantDP.....	35
Figura 8 - Diagrama de Classes do CDP da ferramenta.	36
Figura 9 - Tela inicial para definição de um PDI.....	40
Figura 10 - Tela da etapa <i>Personal Mapping</i> , mostrando escolha de uma trilha.....	41
Figura 11 - Tela da etapa <i>Personal Mapping</i> , mostrando o mapeamento de Pontos Fortes e Fracos	42
Figura 12 - Tela de definição de um PDI após conclusão da etapa <i>Personal Mapping</i>	43
Figura 13 - Tela da etapa <i>Skill Mapping</i>	44
Figura 14 - Tela de definição de um PDI após conclusão da etapa <i>Skill Mapping</i>	45
Figura 15 - Tela da etapa <i>Skill Prioritization</i>	46
Figura 16 - Tela de definição de um PDI após conclusão da etapa <i>Skill Prioritization</i>	47
Figura 17 - Tela inicial da etapa <i>Action Mapping</i>	48
Figura 18 - Tela de adição de ações para uma <i>skill</i>	49
Figura 19 - Tela de um PDI em andamento.....	50
Figura 20 - Tela de ações de uma <i>skill</i> priorizada.....	51
Figura 21 - Tela de um PDI em andamento, mostrando algumas métricas	52
Figura 22 - Tela de equipes de um líder.....	53
Figura 23 - Tela do membro de uma equipe que está definindo o PDI.....	54
Figura 24 - Tela do membro de uma equipe que possui o PDI definido	55
Figura 25 - Tela de métricas das equipes por cargo.....	56

LISTA DE TABELAS

Tabela 1 - Requisitos funcionais identificados para o InstantDP.....	28
Tabela 2 - Regras de negócio identificadas para o InstantDP	29
Tabela 3 - Objetivos e sua situação na conclusão da monografia	58

SUMARIO

CAPÍTULO 1	8
1.1 INTRODUÇÃO.....	9
1.2 OBJETIVOS.....	11
1.3 HISTÓRICO DE DESENVOLVIMENTO DO TRABALHO.....	11
1.4 ORGANIZAÇÃO DO TEXTO.....	12
CAPÍTULO 2	14
2.1 DESENVOLVIMENTO ÁGIL DE SOFTWARE.....	14
2.2 GESTÃO DE PESSOAS.....	16
2.3 TECNOLOGIAS ENVOLVIDAS NESTE TRABALHO.....	17
2.3.1 <i>Back-end: padrão MVC e framework NestJS</i>	18
2.3.2 <i>Front-end: padrão MVVM e framework Angular</i>	19
2.4 CONSIDERAÇÕES FINAIS DO CAPÍTULO.....	20
CAPÍTULO 3	22
3.1 PROCESSO DE DEFINIÇÃO E ACOMPANHAMENTO DE PDI.....	22
3.2 CONSIDERAÇÕES FINAIS DO CAPÍTULO.....	26
CAPÍTULO 4	27
4.1 FERRAMENTA INSTANTDP.....	27
4.1.1 <i>Propósito da ferramenta</i>	27
4.1.2 <i>Requisitos</i>	27
4.1.3 <i>Casos de Uso</i>	30
4.1.4 <i>Diagramas de Classes</i>	32
4.2 PROJETO DE SISTEMA.....	33
4.2.1 <i>Arquitetura de Software</i>	34
4.2.1.1 <i>Camada de Lógica de Negócio</i>	35
4.2.1.1.1 <i>Componente de Domínio do Problema</i>	35
4.2.1.1.2 <i>Camada de Interface com o Usuário</i>	36
4.2.1.1.2.1 <i>Componente de Interação Humana</i>	37
4.2.1.1.2.2 <i>Componente de Controle de Interação</i>	38
4.2.1.1.3 <i>Camada de Gerência de Dados</i>	38
4.3 APRESENTAÇÃO DA FERRAMENTA INSTANTDP.....	39
4.4 CONSIDERAÇÕES FINAIS DO CAPÍTULO.....	56
CAPÍTULO 5	57
5.1 CONCLUSÃO.....	57
5.2 TRABALHOS FUTUROS.....	60
REFERÊNCIAS BIBLIOGRÁFICAS	62

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

Agilidade em desenvolvimento de software representa um conjunto de ideias e práticas que surgiram por influência de alguns membros relevantes da comunidade de desenvolvimento. Essas ideias começaram a evoluir a partir do Manifesto Ágil¹, uma série de princípios lançados no ano de 2001 com o objetivo de simplificar e mudar a forma como os projetos de software eram conduzidos (AMORIM, 2015). Com o passar do tempo, essas ideias de agilidade pautadas no Manifesto Ágil tomaram conta do cenário de desenvolvimento de software corporativo. As promessas de rapidez, qualidade e redução de custos tornaram os métodos ágeis bastante atraentes aos olhos de gerentes e diretores (AMORIM, 2015).

Atualmente, muitas empresas, de diversos ramos, estão adotando metodologias ágeis no desenvolvimento de software cujos requisitos mudam constantemente durante o processo de desenvolvimento. Mais do que um simples modismo passageiro, agilidade é uma tendência que representa um conceito mais duradouro que permite a ramificação por diferentes movimentos semelhantes (NAISBITT, 2007).

O PicPay, uma *fintech* criada com o objetivo de facilitar soluções no setor financeiro e mudar a maneira como os usuários se relacionam com o dinheiro, aplica métodos ágeis em seus projetos de desenvolvimento. Em relação à organização das equipes, no PicPay existem as verticais, compostas por tribos que, por sua vez, são compostas por *squads*. As verticais são estruturas que possuem um propósito maior e uma linha de negócio mais abrangente, por exemplo, uma vertical de *banking* tem um propósito de desenvolver todos os produtos com cunho bancário. O propósito de uma vertical é dividido em pequenas partes e essas partes passam a ser de responsabilidade das tribos. Por exemplo, uma tribo de PIX tem o propósito de desenvolver apenas o produto de PIX. As *squads* são equipes multidisciplinares que possuem um propósito menor, por exemplo, no contexto de uma tribo de PIX podemos ter um *squad* com propósito de desenvolver e manter somente a gestão de chaves PIX. Cada *squad* possui pelo

¹ <http://agilemanifesto.org/>

menos um *Tech Manager*, um *Product Manager*, um *Product Designer* e outros membros que compõem a equipe de desenvolvimento.

Gerenciar os membros das equipes ágeis e multidisciplinares não é apenas garantir a execução de tarefas, realização de entregas intermediárias e do produto final. É, também, conhecer os liderados, suas competências, suas ambições e ser responsável pelo acompanhamento da carreira de cada um oferecendo apoio na construção de um plano de desenvolvimento individual. O melhor plano de desenvolvimento individual é aquele que consegue uma interseção entre as necessidades da empresa e as necessidades individuais da pessoa.

Nesse contexto, na empresa PicPay é realizada a implementação de PDI (Plano de Desenvolvimento Individual) para todos os funcionários e a ferramenta Qulture Rocks² é usada para acompanhar as ações definidas entre o líder e os membros da equipe para o seu PDI. Porém, atualmente, essa interação entre líder e membros da equipe não é bem definida e padronizada na empresa, fazendo com que cada líder siga um processo próprio e um modelo de trabalho que avalia ser o melhor.

A autora deste trabalho atua como líder de *squads* no PicPay e, após testar alguns processos e modelos em suas equipes durante um ano, elaborou um processo que tem como objetivo engajar os membros da equipe a montar um PDI que os ajude no seu dia a dia e que olhe não só para o que a equipe precisa em termos de *hard* e *soft skills*, mas também para o que a pessoa julga importante desenvolver para sua carreira.

Originalmente a implementação do processo foi realizada com apoio de diferentes ferramentas, como Google Sheets³, Google Docs⁴, Miro⁵ e Qulture Rocks, porém percebeu-se que o uso concomitante de várias ferramentas trouxe algumas dificuldades. Por exemplo, há dificuldades operacionais, como a necessidade de redundância de dados em ferramentas diferentes, o que contribui para inconsistências nos dados.

Assim, este trabalho propõe a formalização desse processo de definição e acompanhamento de PDI em equipes ágeis e o desenvolvimento de uma ferramenta web para

² <https://qulture.rocks/>

³ <https://www.google.com/sheets/about/>

⁴ <https://www.google.com/docs/about/>

⁵ <https://miro.com/>

apoiar a execução do processo definido, eliminando a necessidade de ter cada etapa apoiada em uma ferramenta diferente e eliminando os problemas trazidos com a heterogeneidade do processo dentro da empresa.

1.2 Objetivos

Este trabalho tem como objetivo geral *descrever o processo de definição e acompanhamento de um PDI em equipes ágeis e desenvolver uma ferramenta que apoie sua execução*. O processo deve proporcionar a empresas que aplicam métodos ágeis em suas equipes a padronização da definição e acompanhamento de um PDI trazendo clareza na evolução de carreira independente de cargo ou nível, mapeando *skills* que necessitam de aprimoramento, definindo ações que ajudem na evolução e validação de cada *skill*, bem como acompanhando todo o andamento do processo e evolução geral de cada pessoa. A ferramenta deve proporcionar apoio à execução do processo definido, contendo funcionalidades de mapeamento das *skills*, definição das ações e acompanhamento. Esse objetivo geral pode ser detalhado nos seguintes objetivos específicos:

- i. Descrever o processo de definição e acompanhamento de um PDI;
- ii. Elaborar a modelagem conceitual de uma ferramenta que apoie a execução do processo definido;
- iii. Definir e projetar uma arquitetura para a ferramenta, levando em consideração padrões, modelos e paradigmas utilizados em projeto de sistemas de software;
- iv. Implementar, testar e disponibilizar para uso um MVP (*Minimum Viable Product*) da ferramenta.

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.

- ii. *Criação do processo de definição e acompanhamento de um PDI*: Consistiu em criar e documentar um processo de definição e acompanhamento de um PDI idealizado para atender equipes ágeis.
- iii. *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*.
- iv. *Levantamento e Análise de Requisitos*: Consistiu na identificação e análise dos requisitos da ferramenta a partir do processo criado, incluindo a elaboração de diagramas da UML (*Unified Modeling Language*), tais como diagrama de casos de uso e diagrama de classes. Após isso, utilizando-se o método prototípico, foram desenvolvidos *mockups* da ferramenta com as definições de design de interface.
- v. *Design, Implementação e Testes*: Nesta etapa ocorreu o projeto de arquitetura e implementação de componentes. Foram feitos testes de usabilidade com os *mockups* da ferramenta ao longo do desenvolvimento do projeto e após a conclusão das funcionalidades.
- vi. *Escrita da Monografia*: Consistiu na escrita desta monografia.

1.4 Organização do Texto

Neste capítulo inicial está contida a introdução do trabalho, assim 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.

Capítulo 3 – Processo de Definição e Acompanhamento de um PDI: Apresenta o processo de definição e acompanhamento de um PDI.

Capítulo 4 – Ferramenta InstantDP: Apresenta a ferramenta, os requisitos da ferramenta desenvolvida para o apoio do processo, a descrição de seu contexto e propósito de uso, seus casos de uso, modelos de classe, arquitetura e detalhes de componentes da arquitetura. Também apresenta os resultados obtidos nos testes de usabilidade e descreve as funcionalidades implementadas da ferramenta, apresentando alguns fluxos possíveis de sua utilização.

Capítulo 5 – Considerações finais: Apresenta as considerações finais do trabalho, incluindo algumas dificuldades encontradas, 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 quatro seções, a saber: Desenvolvimento Ágil de Software (Seção 2.1), Gestão de Pessoas (Seção 2.2), Tecnologias envolvidas nesse trabalho (Seção 2.3) e Considerações finais do capítulo (Seção 2.4).

2.1 Desenvolvimento Ágil de Software

Quando se fala em agilidade, é comum que passe pela cabeça o termo velocidade. Mas, agilidade não significa entregar mais rápido. Ser ágil, seguindo os valores e princípios do Manifesto Ágil, é ter flexibilidade e capacidade de adaptação.

O Manifesto Ágil foi elaborado em 2001 por 17 desenvolvedores de software que estavam à procura de melhores formas de conduzir seu trabalho. Ele é composto por quatro valores e doze princípios que norteiam o desenvolvimento em cenários de agilidade. São valores registrados no Manifesto Ágil (BECK et al., 2001):

- Indivíduos e interações mais que processos e ferramentas;
- Software em funcionamento mais que documentação abrangente;
- Colaboração com o cliente mais que negociação de contratos;
- Responder a mudanças mais que seguir um plano.

Os doze princípios do Manifesto Ágil são (BECK et al., 2001):

- 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 à 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 é através 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.

Para auxiliar na implementação dos princípios propostos pelo Manifesto Ágil, diversos *frameworks* e métodos foram desenvolvidos. Um dos mais famosos, o Scrum, é um *framework* que contém processos e técnicas de gestão ágil para o desenvolvimento de projetos de software. No Scrum, as pessoas envolvidas no projeto assumem papéis específicos, cada um com seu propósito bem definido. Essa metodologia proporciona uma visão de ciclos de entregas de atividades e tem como base a realização de reuniões para alinhamentos diários das atividades, levantamento de impedimentos e manutenção de um *backlog* bem estruturado para o projeto.

O Kanban, outro *framework* bastante conhecido e utilizado pela comunidade, é uma das metodologias ágeis criadas para organizar as tarefas de uma empresa, seja com post-its coloridos ou softwares virtuais, deixando visível para todos os envolvidos o status de cada atividade. A divisão destas etapas, geralmente, segue em pelo menos três colunas: *To do*: atividades que não começaram, mas precisam ser feitas; *Doing*: atividades que estão em andamento; *Done*: atividades

que foram finalizadas pela equipe. Além dessas três colunas, as empresas têm como customizar suas etapas, adicionando ou alterando colunas para que façam mais sentido para as equipes.

Existem também outros *frameworks* que apoiam a implementação dos princípios propostos pelo manifesto ágil e cada empresa deve analisar qual metodologia ou combinações de metodologias, no contexto das equipes, consegue apoiar a entrega de melhores resultados e aumento da produtividade.

2.2 Gestão de Pessoas

De forma simples e direta, a gestão de pessoas em uma equipe pode ser definida como o conjunto de atividades para fazer com que uma equipe faça o que precisa ser feito e faça bem-feito. Quando o gestor vai além de dar o trabalho para a equipe e tem uma visão mais ampla e que inspira a equipe a realizar o seu trabalho, ele se torna um líder (ZHUO, 2019).

Em um passo adiante, para que os gestores possam conhecer sua equipe, é preciso reconhecê-la como um sistema complexo e, para isso, existem diferentes abordagens que podem ser utilizadas. Da mesma forma que um piloto deve saber como um avião voa, um gestor deve saber como sistemas sociais funcionam. Os sistemas sociais, assim como os aviões, são sistemas complexos e podem ser divididos em partes menores para o entendimento de cada componente. Assim, um piloto poderia conhecer todas as peças de seu avião, porém este conhecimento não seria suficiente para entender de que forma o conjunto dessas peças faz com que o avião voe. No mesmo sentido, o gestor deve conhecer os diferentes componentes envolvidos no projeto, mas também deve ter a visão do todo para compreender de que forma esse sistema poderá construir os produtos da forma mais adequada. Apesar disso, prever o que acontecerá em um voo ou como o projeto de software progredirá dependerá das condições do ambiente, como fatores externos, internos, pessoas, ferramentas e processos (APPELO, 2010).

Quando métodos ágeis são empregados no desenvolvimento de software, potencializa-se equipes multidisciplinares e autogerenciadas e a realização do trabalho com maior produtividade e qualidade, agregando maior valor ao produto (PAZ, 2013). Gerenciar uma equipe ágil e multidisciplinar vai muito além de fazer os processos acontecerem como planejado. É preciso ser um líder, inspirar e engajar a equipe a fazer parte do produto que está sendo construído. Enquanto líder, deve existir uma constante preocupação e atenção à motivação, ao bem-estar e à evolução dos membros da equipe.

No cenário atual, em que diversas empresas de desenvolvimento estão com vagas abertas em suas equipes, a empresa precisa ter um diferencial que vá além da remuneração (ARAGÃO, 2021). Um dos diferenciais mais atrativos é ter a certeza de que as contribuições nas equipes não serão apenas execução de tarefas e entrega de software, e que haverá um acompanhamento bem estruturado de um PDI adequado que permita ao indivíduo enxergar um caminho de crescimento e evolução (ZHUO, 2019).

Apesar do PDI ser um compromisso dos membros da equipe com o seu desenvolvimento, é responsabilidade do líder direcionar a construção de um plano que faça sentido para a empresa e para a pessoa. O PDI sistematiza diversas ações a serem tomadas para que a pessoa conquiste certo objetivo por meio do desenvolvimento pessoal e profissional, agindo como um roteiro para que a pessoa saia do estado atual e caminhe em direção ao estado que deseja alcançar.

Conhecer os liderados, suas competências, suas ambições, estabelecer uma relação saudável e estável são pontos chave para traçar um plano de evolução de carreira efetivo e conseguir aproveitar o melhor de cada um. Quando o gestor conhece a sua equipe e qual o melhor formato de trabalho, as decisões e a gestão em geral da equipe se tornam mais fáceis e efetivas (ZHUO, 2019).

2.3 Tecnologias Envolvidas neste Trabalho

Nesta seção são apresentadas as tecnologias utilizadas neste trabalho, contemplando suas descrições. 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 parte relativa ao cliente e o *back-end* é a parte relativa ao servidor.

Com o objetivo de utilizar soluções prontas para problemas comuns poupando tempo e esforço, é muito comum a prática de utilização de *frameworks* na construção de *WebApps*. Em Engenharia de Software, um *framework* é uma estrutura que fornece códigos reutilizáveis de forma abstrata, que podem ser utilizados nos mais diversos projetos (RIEHLE, 2000). Neste trabalho foram utilizados *frameworks* tanto para o *front-end*, lado do cliente, quanto para o *back-end*, lado do servidor, e serão apresentados nas seções a seguir.

2.3.1 Back-end: padrão MVC e *framework* NestJS

O padrão de arquitetura MVC (*Model-View-Controller*) foi a base do *back-end* da aplicação. Esse padrão de arquitetura é responsável por contribuir na otimização da velocidade entre as requisições feitas pelo comando dos usuários e é constituído por três camadas, representadas na Figura 1:

- i. Camada *Model*: Também é conhecida como *Business Object Model*. Sua responsabilidade é gerenciar e controlar a forma como os dados se comportam por meio das funções, lógica e regras de negócios estabelecidas;
- ii. Camada *View*: Essa camada é responsável por apresentar as informações de forma visual ao usuário;
- iii. Camada *Controller*: Essa camada é responsável por intermediar as requisições enviadas pelo *View* com as respostas fornecidas pelo *Model*, processando os dados que o usuário informou e repassando para outras camadas.

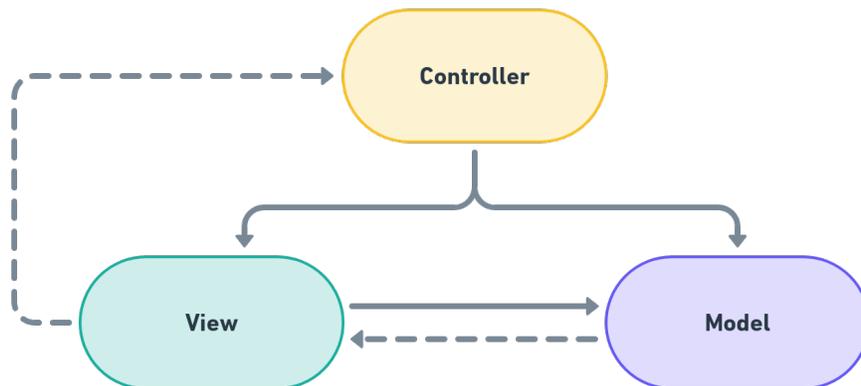


Figura 1 - Diagrama representando o padrão MVC (feito pela autora).

Para a implementação do padrão MVC na ferramenta foi utilizado o *framework* NestJS⁶, um *framework* *node.js* progressivo feito para criar aplicações *back-end* eficientes, escaláveis e confiáveis. É construído com as linguagens *TypeScript*⁷ e *JavaScript*⁸ e neste trabalho foi utilizado apenas com a linguagem *TypeScript*. A escolha de utilização da linguagem *TypeScript* se deu ao fato de trazer a robustez de uma tipagem estática para aplicações *JavaScript*, o que torna o projeto

⁶ <https://nestjs.com/>

⁷ <https://www.typescriptlang.org/>

⁸ <https://www.javascript.com/>

muito mais escalável, conciso e com a possibilidade de identificar bugs ainda em tempo de compilação. Além disso, NestJS possui um poderoso sistema de injeção de dependências que facilita bastante a integração dos módulos com suas dependências, como *controllers* e *services*, de modo que se consiga ter um código limpo, testável e organizado.

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

- **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.
- **MYSQL⁹:** Sistema de gerenciamento de banco de dados relacional, que utiliza a linguagem de consulta estruturada, o SQL.

2.3.2 Front-end: padrão MVVM e *framework* Angular

O padrão de arquitetura MVVM (*Model-View-ViewModel*) foi a base do *front-end* da aplicação. Esse padrão foi apresentado pela primeira vez por Jonh Gossman (Smith, 2009), sendo ele uma variação do padrão PM (*Presentation Model*), e ambos os padrões MVVM e PM são derivados do padrão MVC. O padrão MVVM consiste na divisão do projeto em três camadas de responsabilidades, representadas na Figura 2:

- i. Camada *Model*: Essa camada encapsula a lógica de negócios e os dados.
- ii. Camada *View*: A responsabilidade da camada *View* é definir a aparência ou estrutura que o usuário vê na tela.
- iii. Camada *ViewModel*: Essa camada disponibiliza para a *View* uma lógica de apresentação. A *ViewModel* não tem nenhum conhecimento específico sobre a *View*, ou como ela é implementada, nem o seu tipo. Ela é peça fundamental no MVVM, porque é ela quem vai coordenar as iterações da *View* com o *Model* e pode implementar a lógica de validação para garantir a consistência dos dados.

⁹ <https://www.mysql.com/>



Figura 2 - Diagrama representando o padrão MVVM (feito pela autora).

Para a implementação do padrão MVVM na ferramenta foi utilizado o *framework Angular*¹⁰, um *framework* utilizado para a criação de SPAs (*Single Page Applications*). Quando uma página SPA é acessada pela primeira vez todas as requisições AJAX (*Asynchronous JavaScript and XML*) são processadas para o seu carregamento completo e a aplicação Web interage com o usuário reescrevendo a página atual dinamicamente, sem necessidade de carregar novas páginas, tornando o uso da aplicação mais fluido, sem necessidade de requisições ao servidor em todas as interações.

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

- **TypeScript:** Linguagem de programação base do *Angular*, que estende os recursos atuais do *JavaScript* com a possibilidade de adicionar tipos estáticos pré-definidos na sua aplicação, em tempo de compilação.
- **HyperText Markup Language (HTML)**¹¹: Linguagem de marcação de hipertexto utilizada na construção e organização dos componentes dentro de páginas web.
- **Sassy Cascading Style Sheets (SCSS)**¹²: É um pré-processador da linguagem de estilo CSS (*Cascading Style Sheets*) utilizada para configurar a apresentação de componentes descritos em HTML, com foco em torná-los mais modernos e atrativos.

2.4 Considerações Finais do Capítulo

Este capítulo apresentou fundamentos teóricos e tecnológicos relevantes para o desenvolvimento deste trabalho. No que diz respeito a aspectos teóricos, foram abordados tópicos relacionados a desenvolvimento ágil de software e gestão de pessoas, que fornecem a base teórica para as duas contribuições deste trabalho: processo de definição e acompanhamento

¹⁰ <https://angular.io/>

¹¹ <https://html.spec.whatwg.org/multipage/>

¹² <https://sass-lang.com/>

de um PDI e da ferramenta de apoio ao processo. No âmbito dos aspectos tecnológicos, foram apresentadas as tecnologias envolvidas no processo de construção da ferramenta, descrevendo os *frameworks* que foram utilizados para implementar a arquitetura da ferramenta e linguagens envolvidas no seu desenvolvimento.

Capítulo 3

Processo de Definição e Acompanhamento de PDI e a Ferramenta InstantDP

Este capítulo apresenta o processo de definição e acompanhamento de um PDI. Na Seção 3.1 é apresentado o processo desenvolvido e na Seção 3.2 é apresentada as considerações finais do capítulo.

3.1 Processo de Definição e Acompanhamento de PDI

Como dito anteriormente, uma das contribuições deste trabalho é documentar e apresentar um processo de definição e acompanhamento de um PDI. Este processo foi elaborado, aplicado e validado em vários ciclos pela autora deste trabalho na empresa PicPay. A seguir é apresentada uma breve descrição sobre a construção do processo e sobre o processo em si.

Em empresas em que o processo de definição de um PDI não é padronizado e bem estruturado ocorrem vários problemas, como por exemplo: os líderes definem o processo com base no seu próprio conhecimento, gerando uma desigualdade na forma de definir e acompanhar; como cada líder segue da forma que julga melhor, membros de uma equipe podem ter um apoio melhor ou pior comparado com outras equipes; nem sempre fica claro quais são os passos ou o roteiro a ser seguido para sair de um determinado estado e evoluir para o estado desejado; dentre outros.

Para resolver os problemas citados e garantir que todos os membros da equipe tenham um PDI que seja efetivo, a autora se propôs a estruturar um processo que atenda às necessidades identificadas e seja simples de ser aplicado. Para construir o processo e o fazer chegar a um nível de maturidade suficiente para ser utilizado, ocorreram muitos testes, muitas conversas com outros *Tech Managers*, com *Heads de Tecnologia* e com desenvolvedores. A aplicação do processo nas equipes teve duração de um ano e foi o passo mais rico em termos de *feedbacks*, pois a autora pôde coletar *insights*, sugestões e realizar melhorias no processo para que realmente seja aplicável no contexto de empresas que utilizam metodologias ágeis e que gere o resultado esperado.

Para que o processo seja aplicado, é necessário que a empresa construa um plano de carreira bem estruturado. Isso inclui ter mapeados: todos os cargos da empresa; níveis de

senioridade de cada cargo; grupos de *skills* necessários para cada cargo; e o nível de cada uma dessas *skills* para cada senioridade de um cargo. O processo criado não engloba como a empresa deve construir o seu plano de carreira, uma vez que existem muitos fatores que podem influenciar como a empresa o define e o plano pode evoluir à medida que a empresa cresce e novos fatores influenciam seus cargos e *skills*.

Antes de apresentar as etapas do processo, é importante trazer o significado de alguns termos utilizados durante as etapas:

- *Skills*: Uma *skill* é uma competência técnica ou comportamental que pode ser necessária para um determinado cargo. No contexto do processo, ao mapear os pontos fortes e pontos fracos de uma pessoa, esses pontos são considerados competências, ou seja, *skills*. Por exemplo: se uma pessoa possui um ponto fraco em relação a comunicação e não consegue expor suas ideias em reuniões com muitas pessoas, o processo considera a comunicação como uma *skill* que pode ser trabalhada independente de ser uma *skill* necessária para o cargo da pessoa.
- *Ações*: Ações são a forma de medir se a pessoa cumpriu ou não o que se propôs a desenvolver em um ciclo. As ações precisam ter metas de obtenção de conhecimento e metas de validação do conhecimento adquirido. Para o mesmo exemplo anterior, a pessoa pode definir ações como: leitura de algum livro que a ajude a adquirir uma base teórica para o assunto; e pode definir ações que a desafiem a praticar e aprender na prática como, por exemplo, fazer uma apresentação sobre o assunto estudado para um grupo grande de pessoas.
- *Ciclos*: Quando pensando em evoluir e avançar para um próximo nível de carreira, estamos pensando em *skills* a desenvolver e ações para alcançar esse objetivo. Porém, não é viável e saudável para uma pessoa evoluir todas as *skills* em um mesmo período. Por isso, o PDI é composto por ciclos e um ciclo é um período, normalmente determinado pela empresa, em que uma pessoa trabalha em um grupo limitado de *skills* com o objetivo de trazer foco e uma evolução gradual. Ao fim de cada ciclo, um novo PDI é criado considerando a evolução adquirida no ciclo anterior. O primeiro ciclo de uma pessoa sempre é baseado na sua autoavaliação e os ciclos seguintes contam com *feedbacks* de seus pares e do seu líder.

Considerando que a empresa possua um plano de carreira bem estruturado, 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, que é descrito em seguida.

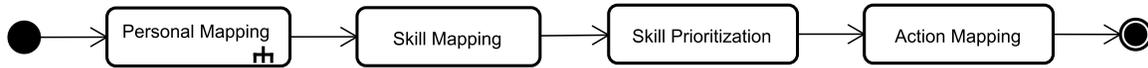


Figura 3 - Etapas do processo proposto.

- i. **Personal Mapping:** A primeira etapa para a definição de um bom PDI é a pessoa praticar o autoconhecimento, incluindo conhecer a trilha que a pessoa deseja seguir em sua carreira. Independente do ramo da empresa, sempre existe a possibilidade de seguir uma trilha mais técnica ou especialista e uma trilha de gestão. Além de entender a trilha com a qual a pessoa se identifica, esta etapa também tem como objetivo mapear os pontos fortes e fracos que a pessoa já consegue identificar em si. A execução desta etapa consiste na identificação desses pontos. Por exemplo: uma pessoa pode ocupar o cargo de um desenvolvedor *back-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*. Como esta etapa é complexa, sendo composta por mais de uma tarefa, a etapa foi detalhada na Figura 4:

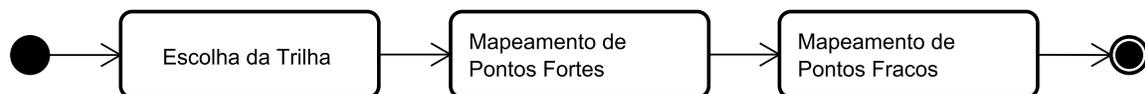


Figura 4 - Representação das tarefas da etapa Personal Mapping.

- ii. **Skills Mapping:** Esta etapa tem como objetivo avaliar o estado atual da pessoa. Quando a pessoa é contratada, ela é contratada para um determinado cargo com um determinado nível, de acordo com experiências passadas, testes de conhecimento, etc. Com o plano de carreira bem definido, podemos saber todas as *skills* e níveis de cada *skill* que são

esperados em cada cargo da empresa. O tipo de mapeamento das *skills* que necessitam de desenvolvimento ou aprimoramento depende do ciclo: o primeiro ciclo ocorre quando a pessoa é nova na empresa ou quando a empresa está começando a aplicar o processo. Nesses cenários, o mapeamento das *skills* ocorre por meio de uma autoavaliação de acordo com cada nível possível em cada *skill* esperada para o cargo e senioridade da pessoa, que foram definidos no plano de carreira da empresa. Para os próximos ciclos, além da autoavaliação, conta-se com a avaliação dos pares e do líder da pessoa. Ao final da avaliação, é gerada uma lista de *skills* em que o nível atual da pessoa está abaixo do nível esperado para o cargo e senioridade. Essa lista de *skills* é utilizada na próxima etapa. Por exemplo: para a mesma pessoa citada na etapa anterior e considerando que é o primeiro ciclo dela, no plano de carreira da empresa pode ter mapeado que para um desenvolvedor *back-end* com senioridade sênior a *skill* de testes unitários é uma das esperadas e essa *skill* tem como possíveis níveis: 1, para nível inicial; 2, para nível intermediário; e 3 para nível avançado. Então, nesta etapa, a pessoa se autoavaliaria na *skill* de testes unitários escolhendo um dos possíveis níveis, assim como para as demais *skills* necessárias para um desenvolvedor *back-end* com senioridade sênior.

- iii. ***Skill Prioritization:*** Após o mapeamento dos pontos fortes, pontos fracos e a geração da lista de *skills* que a pessoa precisa desenvolver ou potencializar, tem-se a etapa de priorização. Esta etapa tem como objetivo priorizar as *skills* com o intuito de ser o foco do próximo ciclo de um PDI. Para ajudar na priorização, é importante que a pessoa, juntamente com o líder, explore melhor: *skills* com maior impacto no dia a dia da pessoa e da equipe; *skills* necessárias para um projeto a curto prazo; e *skills* que necessitam de maior esforço, devido à diferença entre nível esperado e nível atual ser muito grande. Também é importante saber escolher a quantidade de *skills* a trabalhar em um ciclo, uma vez que não existe uma quantidade ideal, ou seja, varia muito de acordo com a dificuldade das *skills*, a diferença de nível e o perfil da pessoa. Novamente se torna importante o autoconhecimento e a aproximação do líder com todos os membros da equipe, para conseguir coerentemente orientar os membros da equipe a priorizar uma quantidade de *skills* que encaixe com o perfil e objetivo de carreira. Por exemplo: ainda para o caso apresentado anteriormente, se a pessoa se autoavaliar para a *skill* de testes unitários com o nível 2, mas no plano de carreira da empresa o nível esperado para um desenvolvedor

back-end com senioridade sênior é 3, a *skill* de testes unitários estaria presente na lista de *skills* e a pessoa pode decidir se essa *skill* será priorizada para o ciclo atual.

- iv. **Action Mapping:** A última etapa é a etapa de mapeamento das ações. Para cada *skill* priorizada na etapa anterior é preciso definir ações com finalidade de desenvolvê-la ou potencializá-la. Essa etapa é fundamental para todo o acompanhamento do PDI e da evolução da pessoa. À medida que a pessoa conclui ações, mais a pessoa se aproxima do estado desejado. Por exemplo: continuando o exemplo anterior, se a pessoa escolher priorizar a *skill* de testes unitários, nesta etapa a pessoa precisa definir todas as ações para essa *skill*, incluindo ações para adquirir mais conhecimento e ações de validação, como a aplicação dos conhecimentos em um projeto específico.

Nos testes realizados com o processo definido foram utilizadas diversas ferramentas como: Google Sheets, Google Docs, Miro e Qulture Rocks. Cada etapa do processo era desenvolvida em uma ferramenta diferente: a etapa de **Personal Mapping** era realizada em um arquivo no Google Docs; as etapas de **Skill Mapping** e **Skill Prioritization** eram realizadas utilizando várias planilhas no Google Sheets com a adição de fórmulas para a geração de listas de *skills* priorizadas; e a etapa de **Action Mapping** era realizada utilizando as ferramentas Miro e Qulture Rocks, com o objetivo de trazer um resultado visual para os membros da equipe e documentar as ações mapeadas na ferramenta oficial da empresa. Ao utilizar diversas ferramentas, se tornou difícil manter a consistência dos dados. Para resolver esse problema e para garantir uma experiência consistente e de fácil utilização, este trabalho também traz como contribuição uma ferramenta de apoio à execução do processo. Essa ferramenta será detalhada nas seções seguintes.

3.2 Considerações Finais do Capítulo

Este capítulo apresentou o processo de definição e acompanhamento de um PDI. No próximo capítulo, será apresentada a ferramenta InstantDP desenvolvida para dar apoio à execução do processo definido.

Capítulo 4

Ferramenta InstantDP

Este capítulo apresenta os principais resultados produzidos ao longo do desenvolvimento de InstantDP (Individual Development Plan), uma ferramenta de apoio à execução do processo de definição e acompanhamento de PDI. Na Seção 4.1 é apresentado o propósito de InstantDP, assim como os resultados produzidos durante a especificação e análise de requisitos da ferramenta. Na Seção 4.2 são apresentados resultados produzidos durante a fase de projeto de sistema da ferramenta. Na Seção 4.3 são apresentadas algumas telas da ferramenta. Por fim, a Seção 4.4 apresenta as considerações finais do capítulo.

4.1 Ferramenta InstantDP

O levantamento de requisitos envolve um conjunto de atividades que deve permitir a comunicação, priorização, negociação e a colaboração com todos os interessados relevantes. Deve prover, ainda, uma base para o aparecimento, descoberta e identificação de requisitos, como parte de um processo altamente interativo (AURUM; WOHLIN, 2005). Nesta seção, são apresentados resultados produzidos durante a fase de Levantamento, Análise e Especificação de Requisitos, incluindo: o propósito de InstantDP, os requisitos identificados, seus casos de usos e o diagrama de classes da ferramenta.

4.1.1 Propósito da ferramenta

Como foi apresentado na seção anterior, a utilização do processo com auxílio de muitas ferramentas gera dois problemas grandes: dificuldade de garantir a consistência dos dados e esforço para realizar o fluxo de definição e acompanhamento, devido à necessidade de se usar mais de uma ferramenta. Diante disso, propõe-se InstantDP, que tem como objetivo apoiar o processo de definição e acompanhamento de um PDI, permitindo a criação e acompanhamento de PDI tanto pela pessoa à qual o PDI se destina quanto pelo líder da pessoa.

4.1.2 Requisitos

Com base no processo definido anteriormente e visando atender ao propósito da ferramenta descrito na seção anterior, foram levantados os seguintes requisitos funcionais e regras de negócio:

Tabela 1 - Requisitos funcionais identificados para o InstantDP

ID	Requisito	Dependência
RF01	Deve ser possível cadastrar ¹³ 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 skills necessários para cada cargo e o nível de cada uma dessas skills 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 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 skills 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
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

¹³ O termo cadastrar se refere às quatro operações básicas de um CRUD (Create, Read, Update, Delete)

Tabela 1 - Requisitos funcionais identificados para o InstantDP (cont.)

ID	Requisito	Dependência
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 skills mais priorizadas e as skills com melhor avaliação de cada cargo.	RF04, RF09

Tabela 2 - Regras de negócio identificadas para o InstantDP

ID	Regra de Negócio	Requisitos Relacionados
RN01	Os membros de uma empresa devem assumir um papel, esses papéis podem ser: <i>admin</i> , caso o membro seja o responsável pela criação da empresa na ferramenta e pelo plano de carreira; <i>leader</i> , caso o membro seja líder de uma ou mais equipes; ou <i>user</i> , para os demais casos.	RF03
RN02	Ao inserir um novo membro de uma empresa é obrigatório que o novo membro assuma o papel de <i>user</i> , podendo ainda adicionar outros papéis.	RF03
RN03	Apenas membros de uma empresa que assumem o papel <i>leader</i> podem cadastrar equipes.	RF04
RN04	Apenas membros de uma empresa que assumem o papel <i>leader</i> podem acompanhar o ciclo de PDI dos membros das equipes que lideram.	RF11
RN05	Apenas membros de uma empresa que assumem o papel <i>leader</i> podem acompanhar as métricas das equipes em que também assumem o papel <i>leader</i> .	RF12
RN06	Os ciclos de PDI possuem um <i>status</i> que pode ser: <i>In Definition</i> , caso esteja em uma das etapas de definição do PDI; <i>In Progress</i> , caso todas as <i>skills</i> priorizadas possuam pelo menos uma ação definida; ou <i>Done</i> , caso o ciclo seja concluído.	RF05
RN07	Ao cadastrar um novo ciclo de PDI é obrigatório que assuma o <i>status In Definition</i> .	RF05
RN08	O <i>status</i> de um ciclo de PDI só pode ser alterado para <i>Done</i> caso todas as ações de todas as <i>skills</i> priorizadas estejam com <i>status Done</i> .	RF05

Tabela 2 - Regras de negócio identificadas para o InstantDP (cont.)

ID	Regra de Negócio	Requisitos Relacionados
RN09	É obrigatório que seja priorizada pelo menos uma <i>skill</i> a ser trabalhada no ciclo.	RF08
RN10	As ações definidas para uma <i>skill</i> possuem um <i>status</i> que pode ser: <i>To Do</i> , caso não tenha iniciado a ação; <i>In Progress</i> , caso tenha iniciado a ação; ou <i>Done</i> , caso tenha concluído a ação.	RF09
RN11	Ao cadastrar uma nova ação para uma <i>skill</i> , ela deve obrigatoriamente assumir o <i>status To Do</i> .	RF09
RN12	É obrigatório que seja definida pela menos uma ação para cada <i>skill</i> priorizada.	RF09
RN13	Ao cadastrar um novo membro de uma empresa, um ciclo de PDI deve ser criado com o <i>status In Definition</i> .	RF03

4.1.3 Casos de Uso

Os casos de uso têm como propósito capturar e descrever as funcionalidades que um sistema deve prover para seus usuários (atores) (BARCELLOS, 2018). Com base na identificação dos *stakeholders* envolvidos no processo de definição e acompanhamento de um PDI, foram definidos os seguintes atores para a ferramenta InstantDP:

- **Responsável pelo Plano de Carreira:** Papel responsável pela criação da empresa na ferramenta e cadastrar o plano de carreira.
- **Líder de Equipe:** O líder de equipe tem como função criar as equipes, adicionar os membros da equipe, acompanhar o ciclo de PDI de cada membro de suas equipes e acompanhar as métricas das equipes. Ao acompanhar o ciclo de PDI e as métricas, o líder de equipe também é responsável por garantir que os membros das equipes estão trabalhando nas ações definidas.
- **Membro da Empresa:** Qualquer membro da empresa tem como função criar o seu próprio PDI, precisando passar por todas as etapas de definição e podem acompanhar o andamento das ações definidas.

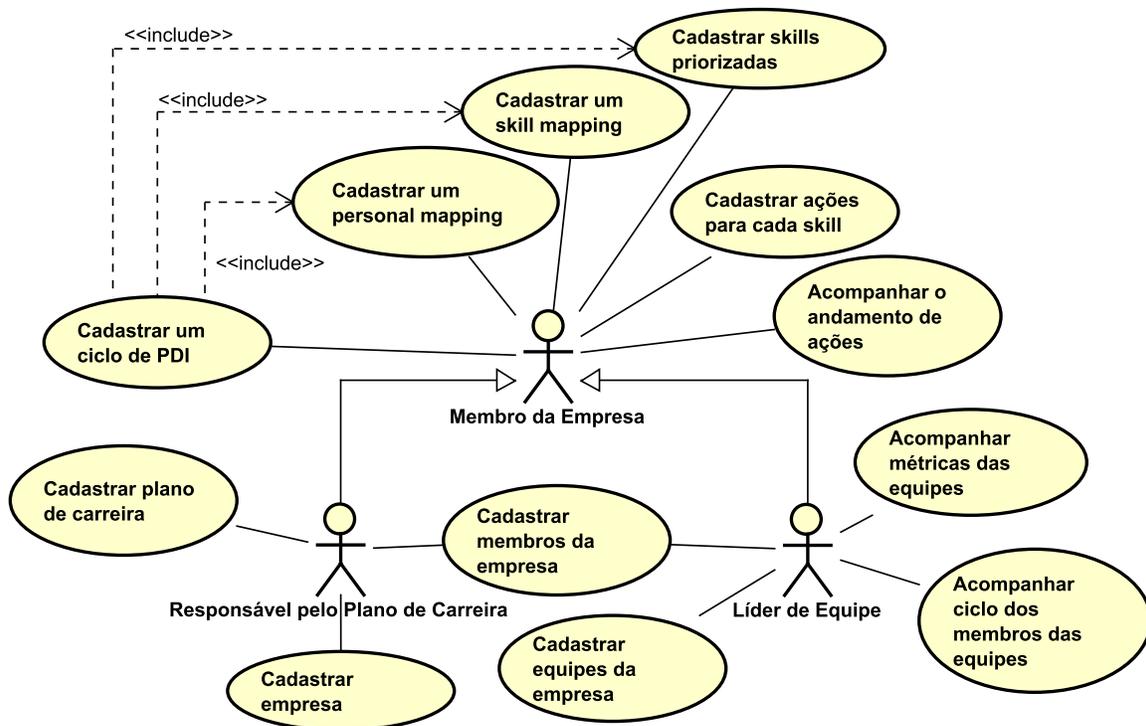


Figura 5 - Diagrama de casos de uso de InstantDP.

Assim como nos requisitos funcionais, os casos de uso cujos títulos começam com “Cadastrar” se referem às quatro operações básicas de um CRUD (*Create, Read, Update, Delete*). Devido ao curto período para o desenvolvimento, disponibilização da ferramenta para o uso e por ser um MVP, alguns requisitos apresentados possuem toda a implementação das operações CRUD do lado do servidor, o *back-end*, mas não possuem implementação do lado do cliente, o *front-end*. Os objetivos desses casos de uso são manter os recursos que apoiam a realização das principais funcionalidades do sistema, que serão apresentadas a seguir.

O caso de uso **Cadastrar Empresa** é realizado pelo Responsável pelo Plano de Carreira da empresa e tem como objetivo gerenciar o cadastro da empresa na ferramenta, incluindo informações como nome da empresa e o e-mail. O Responsável pelo Plano de Carreira também realiza outros dois casos de uso: o caso de uso **Cadastrar Plano de Carreira**, que é fundamental para toda a construção de um PDI, contendo todos os cargos, níveis de senioridade de cada cargo, grupos de *skills* necessários para cada cargo e o nível de cada *skill* para cada senioridade de um cargo; e o caso de uso **Cadastrar Membro da Empresa**, cuja responsabilidade é dividida com o Líder de Equipe e tem como objetivo cadastrar os membros da empresa na ferramenta.

O Líder de Equipe deve **Cadastrar Equipe da Empresa**, sendo responsável por gerenciar as suas equipes e principalmente os membros que compõem as equipes. Este ator também é responsável por **Acompanhar Ciclo de Membro de Equipe** e **Acompanhar Métricas de Equipe**, visando extrair informações importantes com relação ao desempenho dos membros de suas equipes e entender os pontos de melhoria de cada cargo de suas equipes.

O Membro da Empresa é o ator que possui a maior quantidade de responsabilidades. Vale notar que tanto o Responsável pelo Plano de Carreira quanto o Líder de Equipe são Membro da Empresa, herdando todas as responsabilidades. Este ator é responsável por todos os casos de uso ligados ao processo de definição e acompanhamento de um PDI e, ao iniciar o processo, a primeira responsabilidade é a de **Cadastrar um Ciclo de PDI**. Após a criação de um ciclo de PDI, o ator necessita **Cadastrar um Personal Mapping**, detalhando a escolha de trilha de carreira que deseja seguir e mapeando seus pontos fortes e pontos fracos. O próximo passo é **Cadastrar um Skill Mapping**, realizando uma avaliação para cada *skill* esperada do cargo e senioridade. Com o *Personal Mapping* e o *Skill Mapping* criados, é responsabilidade deste ator **Cadastrar Skills Priorizadas** e **Cadastrar Ações para cada Skill**, selecionando todas as *skills* que deseja desenvolver ou potencializar em um ciclo de PDI e definindo todas as ações que serão executadas para atingir o objetivo. Também é responsabilidade deste ator **Acompanhar Andamento das Ações**, sendo a parte fundamental para o fechamento do ciclo de PDI, realizando as ações e acompanhando as métricas de evolução.

4.1.4 Diagramas de Classes

As classes de um modelo representam a expressão inicial de aspectos estruturais do sistema. O paradigma de desenvolvimento orientado a objetos tem como atividade crucial na modelagem conceitual estrutural a identificação das classes do sistema (BARCELLOS, 2018). Como o processo de definição e acompanhamento de um PDI foi utilizado como base para o entendimento do domínio da aplicação da ferramenta, algumas classes da ferramenta foram identificadas a partir da definição do processo. A Figura 4 apresenta o diagrama de classes desenvolvido para a ferramenta InstantDP.

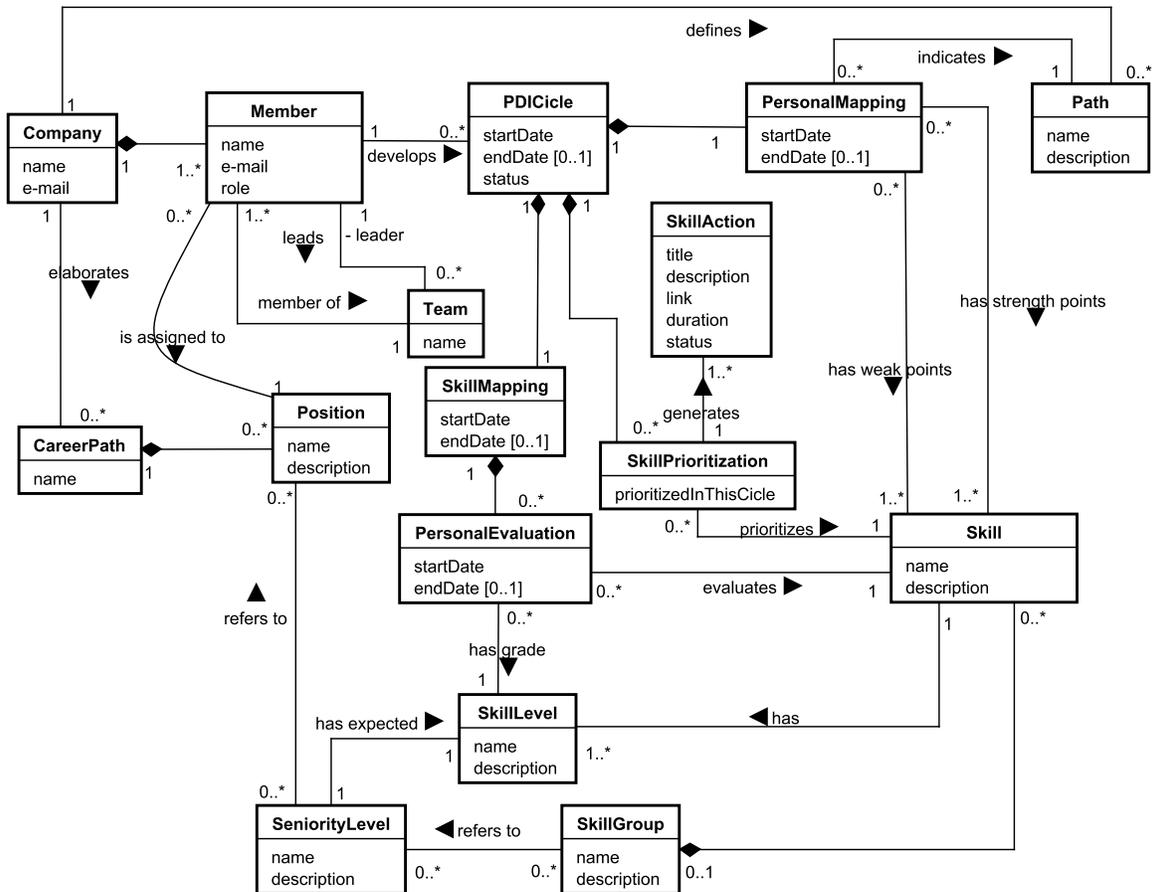


Figura 6 - Diagrama de classes de InstantDP

Complementando o modelo, existe uma restrição que visa garantir a integridade dos dados: na relação entre **Personal Evaluation** e **Skill Level**, o *Skill Level* indicado para a *Skill* sendo avaliada na *Personal Evaluation* deve ser um dos *Skill Levels* relacionados à referida *Skill*;

4.2 Projeto de Sistema

O objetivo da fase de projeto de sistema é produzir uma solução para o problema identificado e modelado durante o levantamento e análise de requisitos, incorporando a tecnologia aos requisitos e projetando o que será construído na implementação (BARCELLOS, 2018). A ferramenta InstantDP foi desenvolvida com a utilização das tecnologias apresentadas na seção 2.3. A seção a seguir apresenta a arquitetura de InstantDP.

4.2.1 Arquitetura de Software

Arquitetura de software de um sistema computacional é a estrutura (ou estruturas) do sistema que compreende elementos de software, propriedades externamente visíveis desses elementos e os relacionamentos entre elas (BASS, CLEMENTS e KAZMAN, 2003). A ferramenta InstantDP foi desenvolvida seguindo a arquitetura cliente-servidor. Para o projeto de arquitetura da ferramenta, optou-se por dividi-la em dois subsistemas: o subsistema do servidor (*back-end*), executado no servidor e o subsistema do cliente (*front-end*), executado no cliente.

O *back-end* foi implementado utilizando o *framework* NestJS, este *framework* segue o padrão MVC (*Model-View-Controller*), sendo composto pela camada lógica de negócio (CLN), correspondente aos componentes *Controller*, e a camada de gerência de dados (CGD), correspondendo aos componentes *Model*. A CLN tem a responsabilidade das regras de negócio da ferramenta e foi desenvolvida seguindo o padrão Modelo de Domínio, sendo composta pelo Componente de Domínio de Problema (CDP), que será detalhado posteriormente. A CGD é responsável por tratar a persistência dos dados no banco de dados.

O *front-end* foi implementado utilizando o *framework* Angular, que segue o padrão MVVM (*Model-View-ViewModel*), sendo composto pela camada de interface com o usuário (CIU), tratando aspectos de exibição e interação com os usuários da ferramenta, correspondendo aos componentes *View* e *ViewModel*. A CIU se divide em dois componentes que serão detalhados posteriormente: componente de interação humana (CIH) e componente de controle de interação (CCI).

A arquitetura de InstantDP combina os padrões MVC e MVVM, onde o *Model* e o *Controller* do MVC se complementam com o *ViewModel* e *View* do MVVM, permitindo possibilidades mais ricas de interação com a ferramenta. A Figura 5 apresenta um diagrama representando a arquitetura de InstantDP organizada segundo os elementos descritos anteriormente.

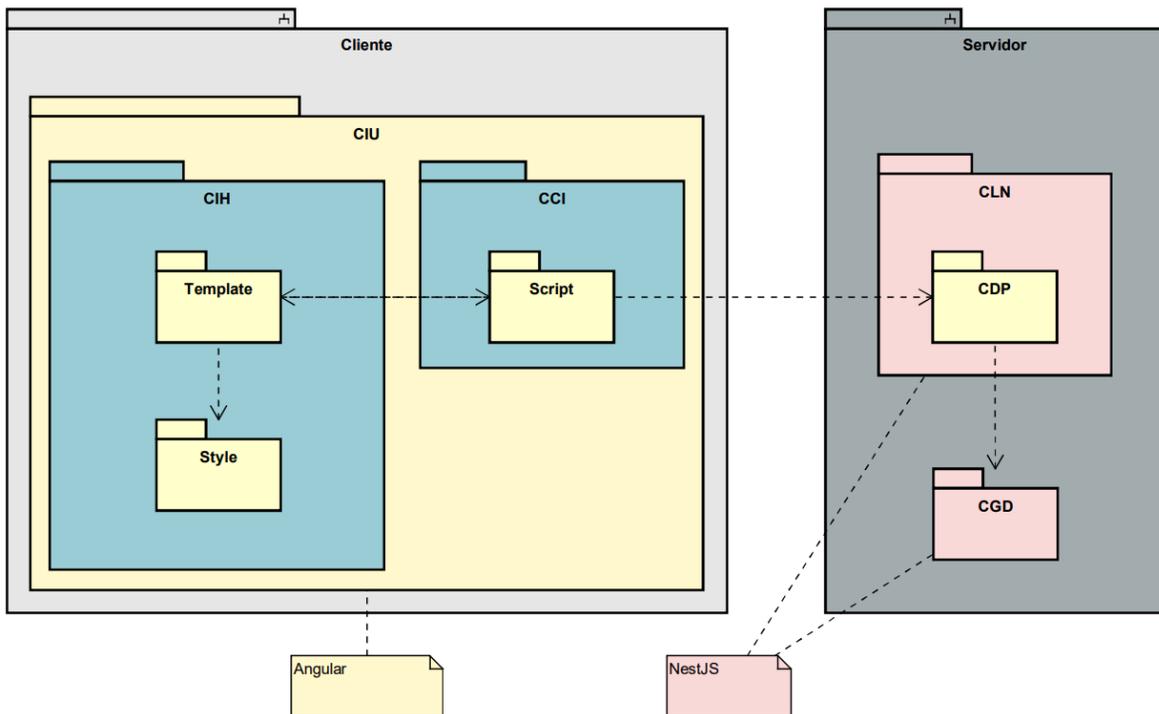


Figura 7 - Arquitetura da ferramenta InstantDP.

Nas próximas subseções são apresentados mais detalhes de cada camada, com fragmentos de diagramas do projeto.

4.2.1.1 Camada de Lógica de Negócio

Com base no escopo e nas funcionalidades necessárias para a ferramenta, o padrão de desenvolvimento da camada de lógica de negócio escolhido foi o de Modelo de Domínio. Nesse padrão, as responsabilidades são distribuídas nos objetos de domínio do problema e a lógica de aplicação é pulverizada nesses objetos, sendo que cada objeto tem uma parte de lógica que é relevante a ele (BARCELLOS, 2018).

4.2.1.1.1 Componente de Domínio do Problema

Essa camada é composta pelo Componente de Domínio do Problema, que são modelos conceituais estruturais produzidos na fase de análise do projeto. A Figura 6 apresenta o diagrama de classes produzido nesta fase.

componente de botão que pode ser reutilizado em várias telas, quanto podendo ser uma tela completa. Para cada componente são gerados três arquivos. Para o exemplo de um componente de botão com nome *button*, teríamos os arquivos: *button.component.html* e *button.component.scss*, representando o *Template* e o *Style*, respectivamente, que implementam o Componente de Interação Humana (CIH); e o arquivo *button.component.ts* representando o *Script*, que implementa o Componente de Controle de Interação (CCI). As seções seguintes detalham mais cada componente.

4.2.1.2.1 Componente de Interação Humana

Toda parte visual de um sistema que compreende os objetos gráficos utilizados para a interação com o usuário faz parte do Componente de Interação Humana (CIH). O CIH é a implementação da interface do sistema e, considerando que alterações na interface de um sistema é constante, visando sempre à evolução do sistema, deve ser mantida separada do resto da implementação do sistema.

No início do desenvolvimento deste componente vários protótipos da interface das telas da ferramenta foram desenhados em folhas de papel, para mapear as possibilidades de utilização da ferramenta. Os protótipos desenvolvidos em papel foram utilizados na disciplina “Interface Humano-Computador”, com a finalidade de aprimorar a usabilidade da ferramenta.

No contexto da referida disciplina, utilizando-se a ferramenta StoryboardThat¹⁴, foram criados *storyboards*, que são organizadores gráficos como uma série de ilustrações ou imagens em sequência com o propósito de visualizar previamente os elementos interativos, para validar se a interface idealizada é capaz de realizar as principais tarefas propostas: definição de um PDI, acompanhamento de um PDI e acompanhamento das métricas geradas.

Após o desenvolvimento dos *storyboards* e com os feedbacks colhidos tanto do professor da disciplina, quanto dos alunos da disciplina, vários ajustes foram realizados na interface idealizada inicialmente. A interface também passou por um processo de prototipação na ferramenta Figma¹⁵, onde foram desenhadas todas as telas essenciais para completar as principais tarefas propostas e assim realizar testes de usabilidade para validar a usabilidade do protótipo desenvolvido. Os testes de usabilidade têm como objetivo observar usuários reais usando a interface para descobrir problemas e pontos de melhorias. Assim, com os resultados dos testes

¹⁴ <https://www.storyboardthat.com/pt>

¹⁵ <https://www.figma.com/design/>

de usabilidade foi possível melhorar a interface da ferramenta podendo desenvolver uma ferramenta que apoie o processo desenvolvido e que também tenha uma boa usabilidade para seus usuários.

Para facilitar tanto as escolhas no desenho da interface quanto na implementação da ferramenta, foi utilizado o Material Design¹⁶, o *design system* criado pelo Google, que provê tanto a documentação dos componentes quanto a possibilidade de utilização dentro do *framework* Angular.

O CIH é composto pelos subcomponentes *Template* e *Style*. O *Template* consiste em código HTML no qual são declarados os elementos a serem exibidos na tela, definindo principalmente as chamadas de métodos e dados fornecidos pelo CCI. O *Style* consiste nos códigos de estilização do *Template*, que foram desenvolvidos em SCSS. Apesar da utilização do Material Design, várias modificações visuais foram feitas utilizando SCSS, mas não foram realizadas modificação de comportamento dos componentes.

4.2.1.2.2 Componente de Controle de Interação

O CCI é o componente responsável pela comunicação entre as regras de lógica de negócio e a interface do usuário. No Angular, ele é composto pelo subcomponente *Script*, que implementa todos os métodos referentes a um componente ou tela. Nele está contida a estrutura que abstrai os dados da CGD, os métodos que conectam as ações dos usuários no CIH às ações correspondentes do CLN através de uma API, atualizando o que é exibido no *Template* quando necessário.

4.2.1.3 Camada de Gerência de Dados

A persistência de dados da aplicação foi implementada utilizando um banco de dados relacional (MySQL), acessado através do *framework* NestJS. Para a implementação da ferramenta foi escolhida a utilização da biblioteca de integração de banco de dados TypeORM. Essa escolha se deu por ser o *Object Relational Mapper* (ORM) mais maduro disponível para TypeScript e, por ser escrito em TypeScript, ele se integra bem com a estrutura NestJS. O NestJS ainda oferece recursos adicionais específicos do NestJS, como injeção de modelo / repositório, capacidade de teste e configuração assíncrona para tornar o acesso ao banco de dados escolhido ainda mais fácil.

¹⁶ <https://material.io/>

4.3 Apresentação da Ferramenta InstantDP

Nesta seção serão apresentadas as telas da ferramenta, abordando a perspectiva de um usuário que atua como Membro da Empresa e a perspectiva de um usuário que atua como Líder de Equipe. Como abordado na seção 3.2.3, devido ao curto período para o desenvolvimento, disponibilização da ferramenta para o uso e por ser um MVP, os casos de uso específicos para usuários que atuam como Responsável pelo Plano de Carreira, o cadastro de membros da empresa e cadastro de equipes não foram desenvolvidos no *front-end* da ferramenta. Os casos de uso citados são essenciais para a utilização da ferramenta, assim a execução dos casos de uso ocorreu utilizando a ferramenta Postman¹⁷, realizando requisições para o *back-end* da ferramenta, possibilitando a realização desses cadastros. Todas as telas da ferramenta possuem um menu de navegação, mostrando ao usuário onde ele se encontra na ferramenta e possibilitando a navegação. Vale destacar que foram utilizados dados fictícios para a demonstração da ferramenta e suas possibilidades.

Iniciando pela perspectiva de um usuário que atua como Membro da Empresa, ou seja, todos os membros de uma empresa podem passar por esse fluxo, a primeira tela é a tela inicial de definição de um PDI (Figura 7). Nesta tela é possível que o usuário veja as quatro etapas de definição de um PDI, trazendo a ideia de ser uma construção gradual. Como algumas etapas dependem de etapas anteriores, o usuário só pode iniciar pela etapa *Personal Mapping*, estando as demais etapas bloqueadas. Além das etapas, é possível que o usuário veja suas informações como nome, cargo e senioridade.

¹⁷ <https://www.postman.com/>

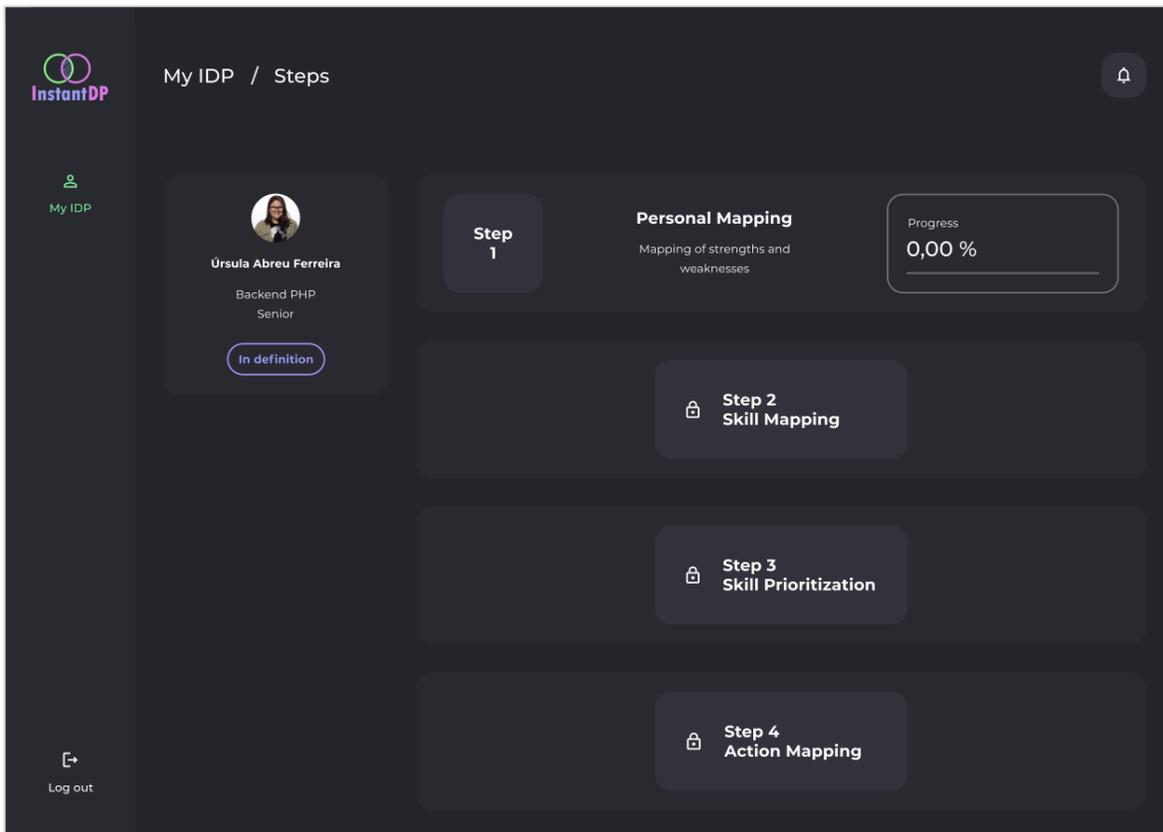


Figura 9 - Tela inicial para definição de um PDI

Ao iniciar a etapa *Personal Mapping*, clicando no respectivo *card*, o usuário se depara com a tela de construção do *Personal Mapping*. Nesta tela o usuário tem a possibilidade de escolher qual trilha deseja seguir em sua carreira, com base nas trilhas disponíveis na empresa (Figuras 8). A escolha da trilha é uma decisão importante para a carreira do usuário e deixar clara essa escolha facilita ao líder do usuário guiar e trabalhar *skills* mais relevantes para a trilha escolhida.

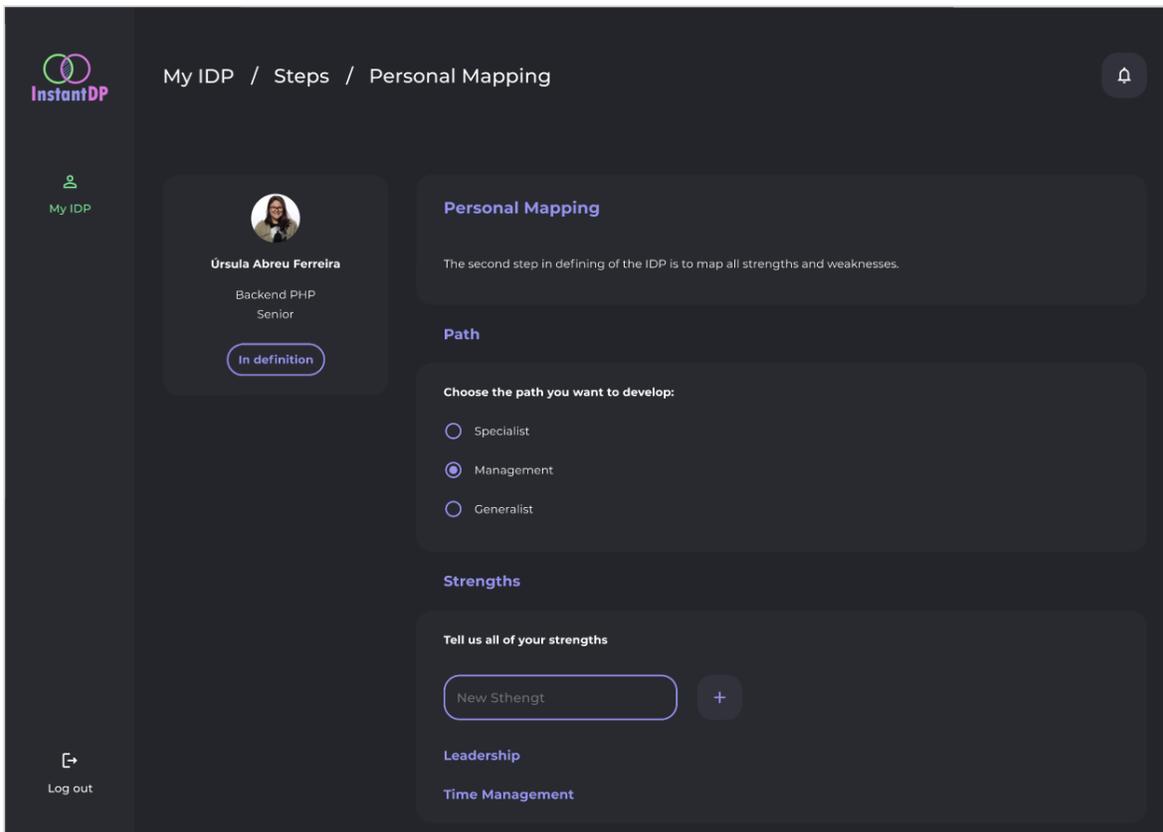


Figura 10 - Tela da etapa *Personal Mapping*, mostrando escolha de uma trilha

Nesta tela também é possível que o usuário mapeie todos os seus pontos fortes e pontos fracos (Figura 9). Ao contrário do que muitas pessoas pensam, não são só os pontos fracos que devem ser desenvolvidos, mas também os pontos fortes, que são seus diferenciais e, dessa forma, potencializá-los também traz resultados positivos. Estes pontos fortes e fracos serão utilizados posteriormente na etapa de *Skill Prioritization*. Ao finalizar a escolha de trilha e o mapeamento dos pontos fortes e fracos, o usuário tem a opção de submeter suas respostas, clicando no botão *Submit*, que conclui esta etapa.

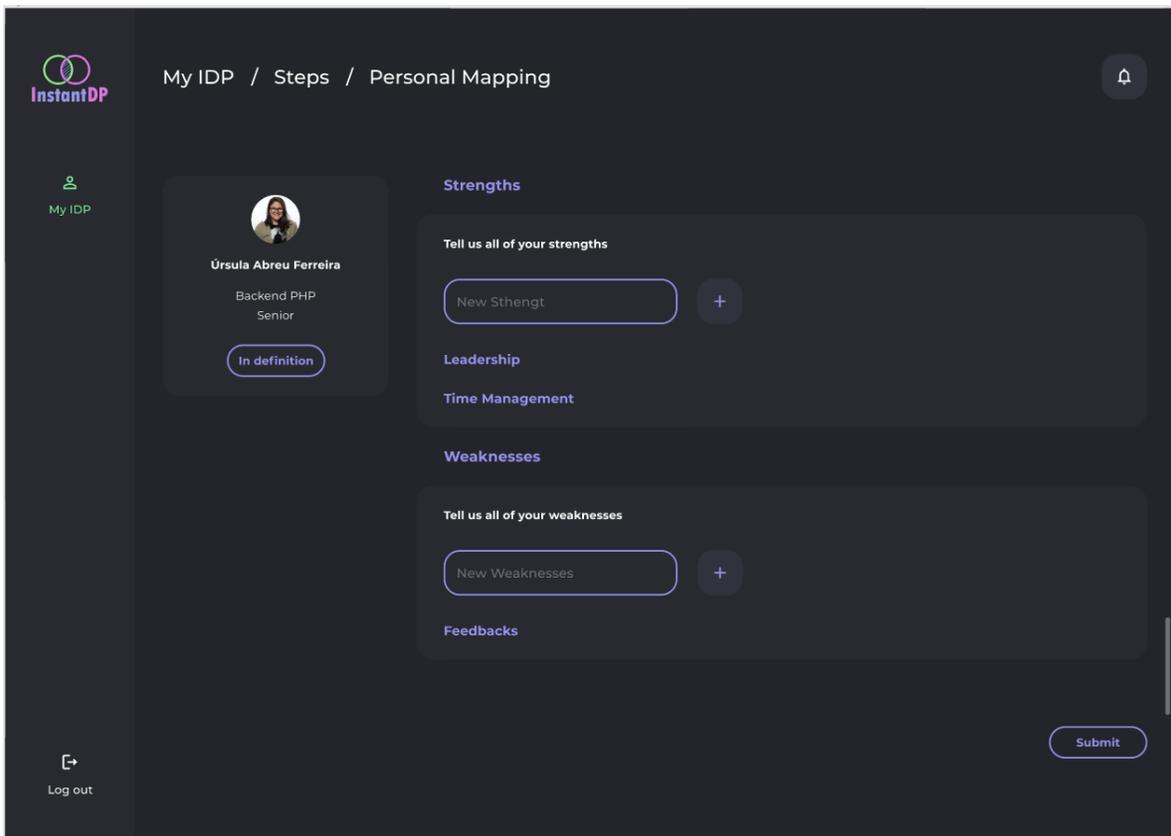


Figura 11 - Tela da etapa *Personal Mapping*, mostrando o mapeamento de Pontos Fortes e Fracos

Ao finalizar a etapa de *Personal Mapping*, o usuário retorna para a tela que apresenta todas as etapas. Nela é apresentado seu progresso na execução do processo e é feita a liberação da próxima etapa (Figura 10), ficando livre a sua escolha de continuar o processo ou retornar a ela em outro momento.

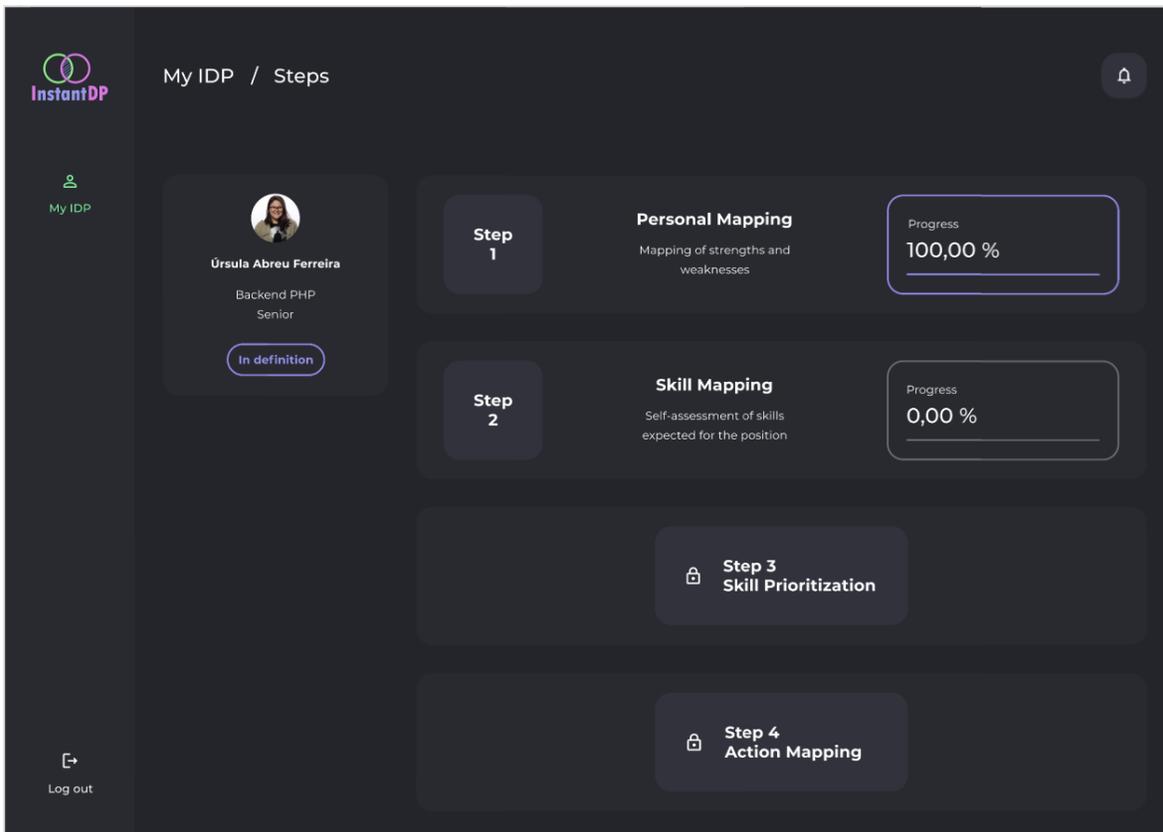


Figura 12 - Tela de definição de um PDI após conclusão da etapa *Personal Mapping*

Ao iniciar a etapa *Skill Mapping*, clicando no respectivo *card* exibido na Figura 10, o usuário se depara com a tela apresentada na Figura 11. Baseada no cargo e senioridade do usuário, a ferramenta traz todos os grupos de *skills* relacionados. Por exemplo: grupo de *hard skills* e grupo de *soft skills* esperadas pela empresa, juntamente com os possíveis níveis de cada *skill*, que foram definidas no plano de carreira da empresa.

Na etapa de *Skill Mapping* é importante que o usuário pratique o autoconhecimento. Por se tratar da construção de um primeiro ciclo e não existir ainda a possibilidade de *feedback* de pares e líder, o usuário que definirá os níveis das *skills* apresentadas de acordo com a sua percepção do dia a dia e com *feedbacks* recebidos diretamente de pessoas próximas. Após avaliar todas as *skills*, assim como na etapa anterior, o usuário tem a opção de submeter suas respostas, ao clicar no botão *Submit*, concluindo esta etapa.

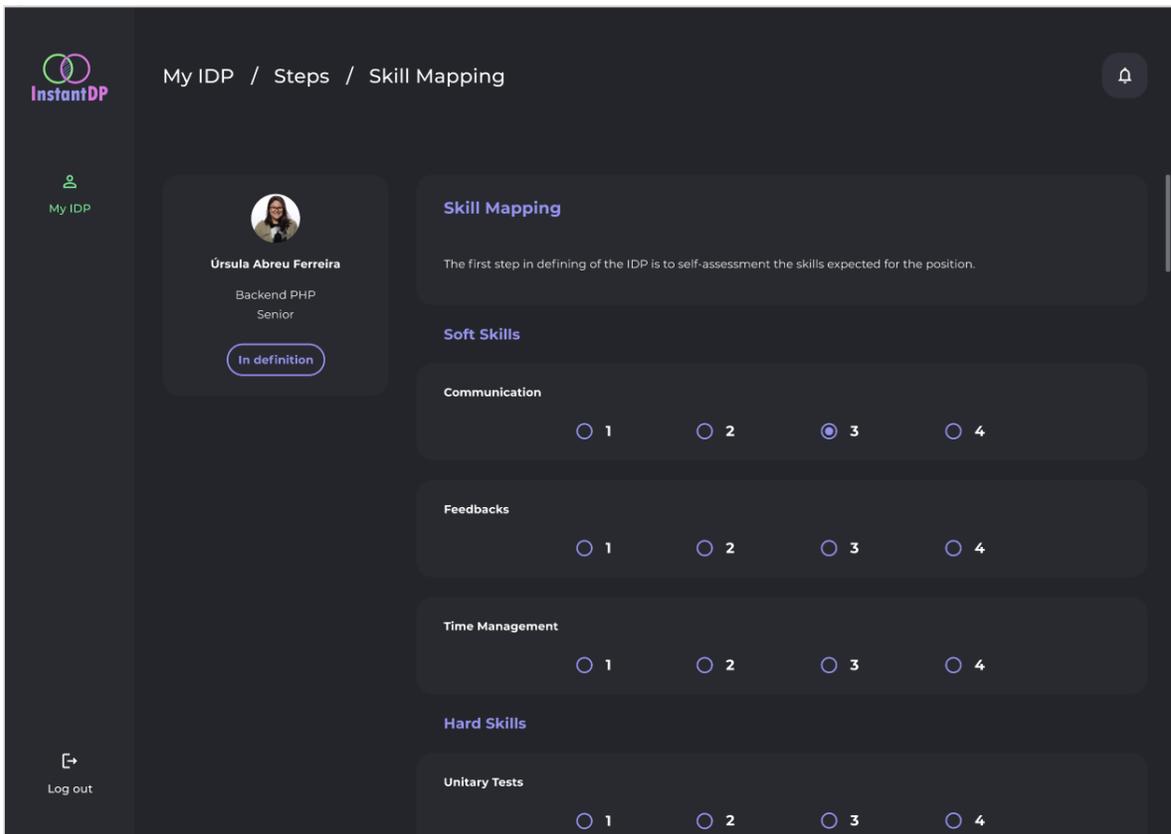


Figura 13 - Tela da etapa *Skill Mapping*

Ao finalizar a etapa de *Skill Mapping*, o usuário é levado para a tela que apresenta todas as etapas e o progresso da execução do processo com liberação da próxima etapa (Figura 12).

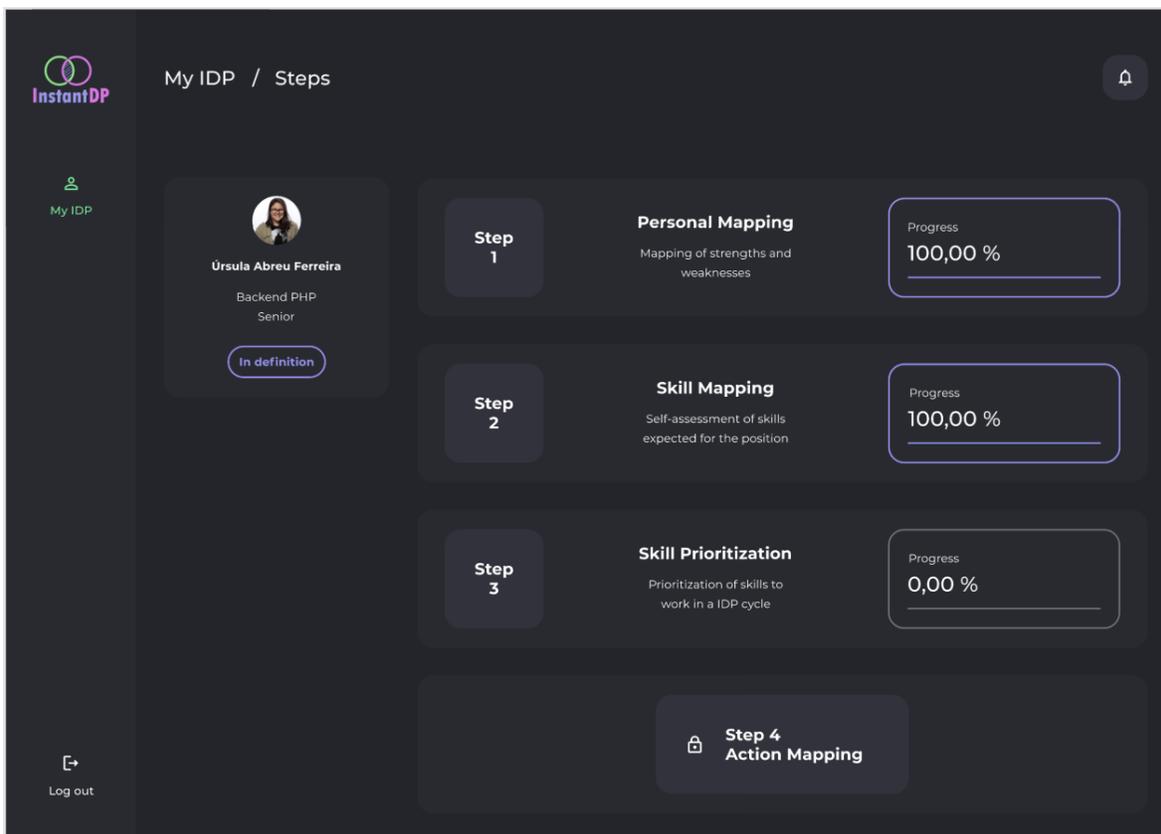


Figura 14 - Tela de definição de um PDI após conclusão da etapa *Skill Mapping*

Ao iniciar a etapa *Skill Prioritization*, clicando no respectivo *card*, o usuário é direcionado para a tela apresentada na Figura 13. A ferramenta faz um cruzamento da avaliação realizada pelo usuário na etapa de *Skill Mapping*, com os níveis esperados para o cargo e senioridade do usuário definidos no plano de carreira, trazendo uma lista de todas as *skills* em que o usuário se avaliou abaixo do nível esperado. Além disso, também estão na lista os pontos fortes e fracos mapeados na etapa *Personal Mapping*, para incentivar o usuário que trabalhe tanto nas *skills* de interesse para a empresa, quanto nas *skills* de interesse próprio.

A tela apresentada na Figura 13 traz a lista das *skills* e um quadro de priorização, na qual o usuário pode arrastar para o quadro todas as *skills* que entender, sozinho ou juntamente com o líder, ser importante para trabalhar no momento. O líder pode auxiliar neste momento, por exemplo, na escolha de *skills* que possam ajudar a curto prazo em algum projeto priorizado para a equipe. Após escolher todas as *skills* que serão priorizadas, assim como na etapa anterior, o usuário tem a opção de submeter suas respostas, ao clicar no botão *Submit*, concluindo esta etapa.

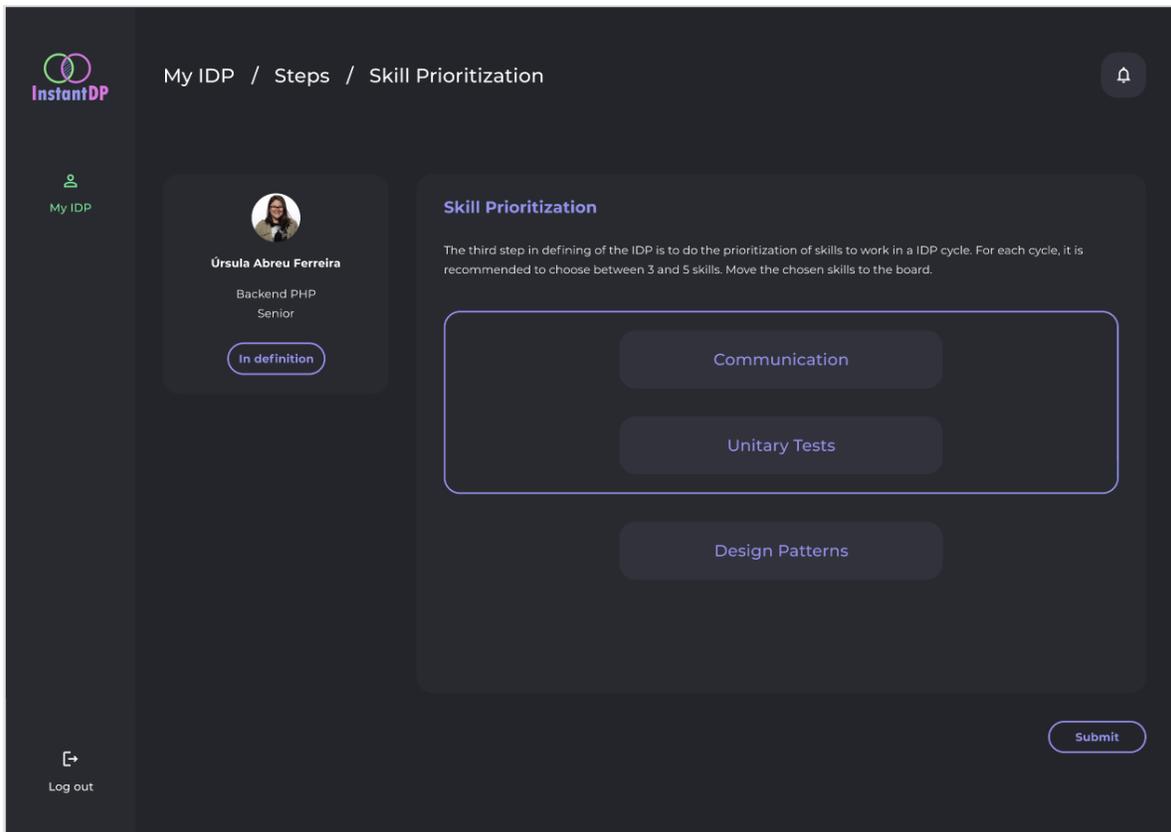


Figura 15 - Tela da etapa *Skill Prioritization*

Ao finalizar a etapa de *Skill Prioritization*, o usuário, novamente, é levado à tela de visualização geral da execução do processo, com liberação da próxima etapa (Figura 14).

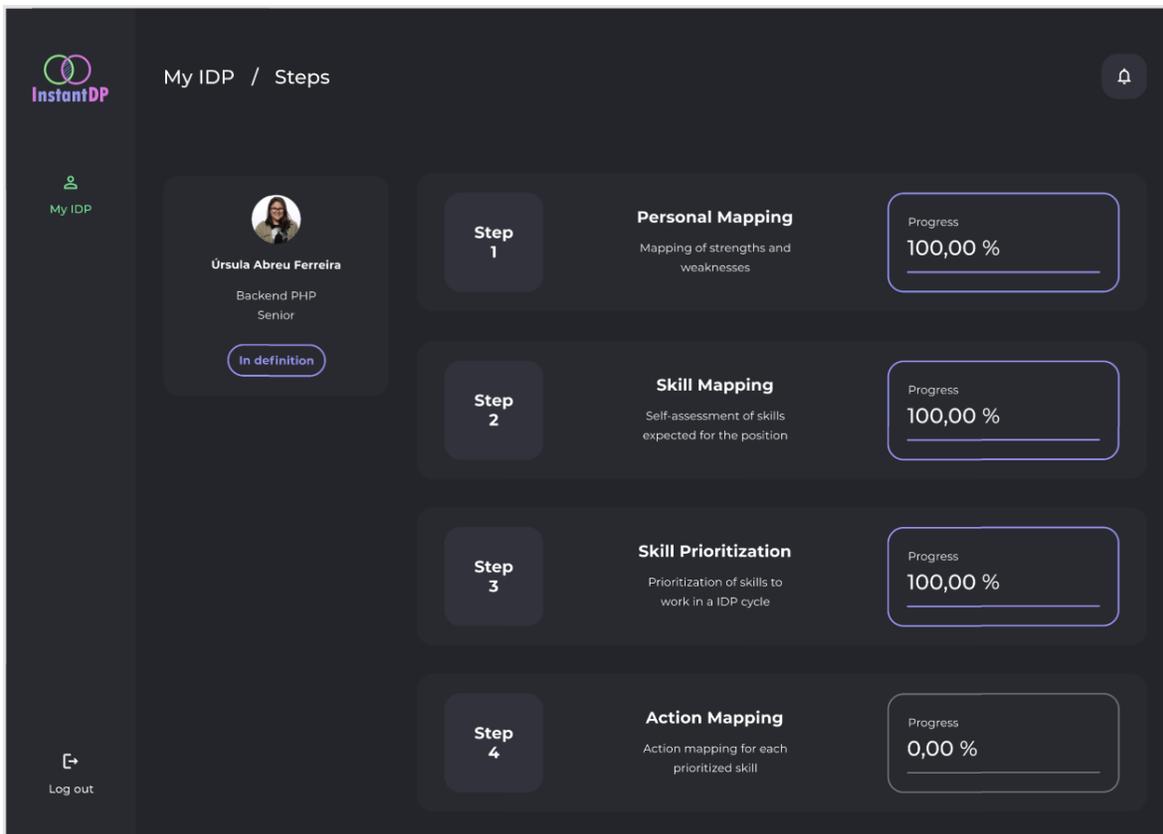


Figura 16 - Tela de definição de um PDI após conclusão da etapa *Skill Prioritization*

Ao iniciar a etapa *Action Mapping*, clicando no respectivo *card*, o usuário é levado para a tela ilustrada na Figura 15, na qual são apresentadas todas as *skills* priorizadas na etapa *Skill Prioritization* com a possibilidade de adicionar ações para cada *skill*, mostrando também o nível atual do usuário e o nível esperado.

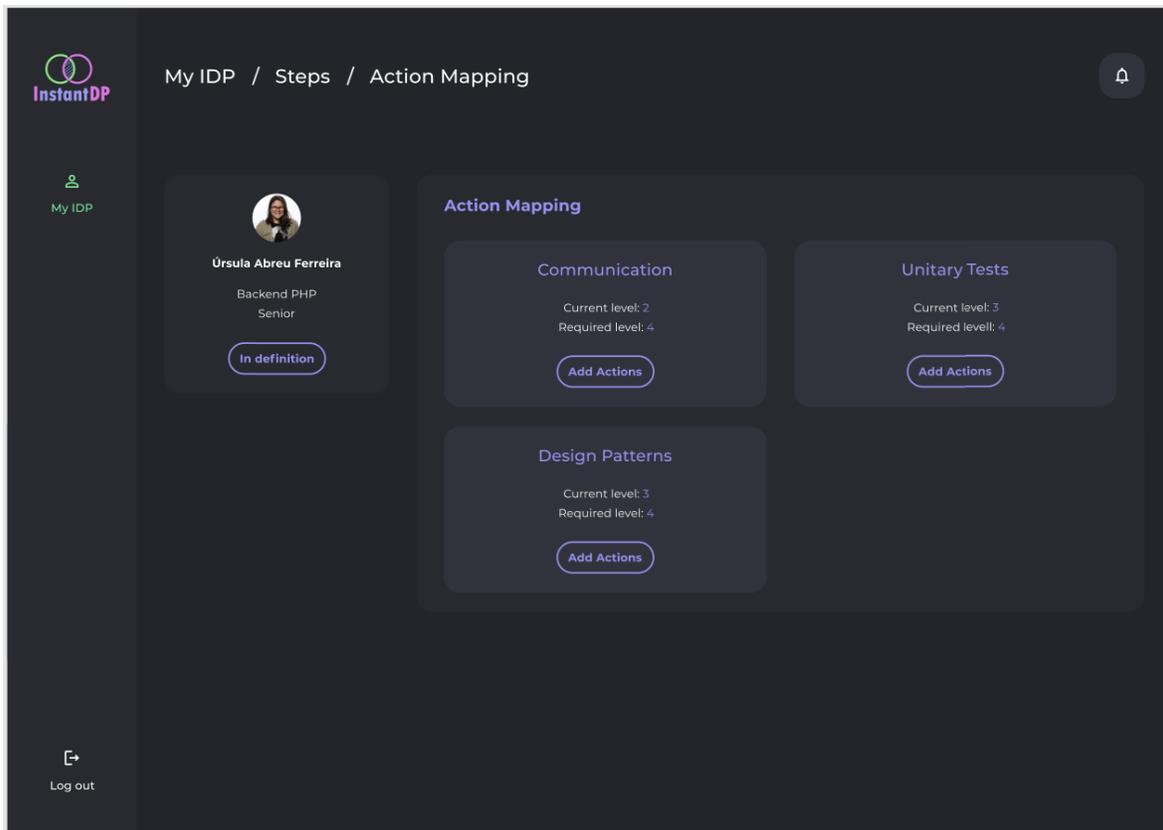


Figura 17 - Tela inicial da etapa *Action Mapping*

Para adicionar ações para as *skills*, é necessário que o usuário clique no botão *Add Actions*. A Figura 16 mostra como funciona a adição de ações. Na figura também é possível ver uma ação já adicionada, sendo criada com o *status To Do*. A definição de ações é o ponto chave para garantir a evolução em uma *skill*, sendo necessário que o usuário adicione ações tanto para adquirir conhecimento quanto ações para validar e aplicar o conhecimento no seu dia a dia na equipe. Após adicionar pelo menos uma ação para cada *skill* priorizada, o usuário conclui todo o processo de definição de um PDI.

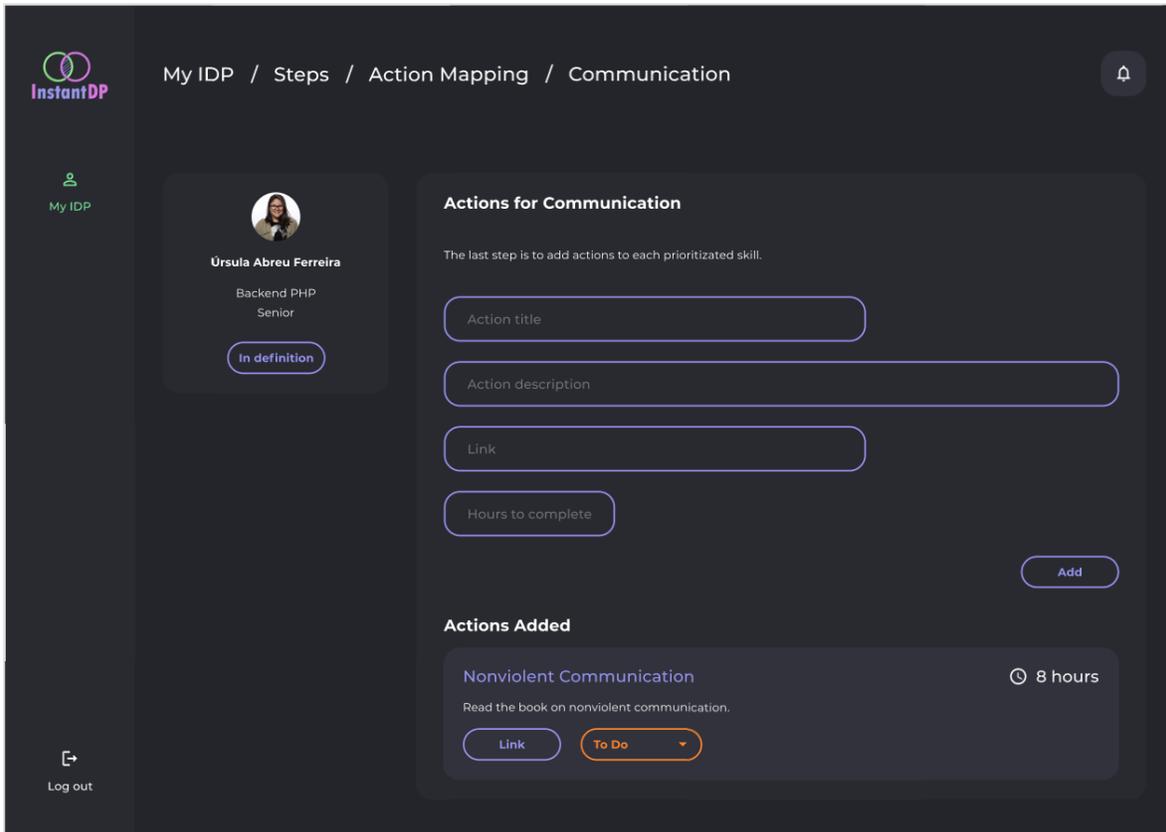


Figura 18 - Tela de adição de ações para uma *skill*

Ao concluir todas as etapas do processo, o ciclo de PDI do usuário passa do *status In Definition* para o *status In Progress* e são apresentadas informações sobre a trilha que o usuário deseja seguir, seus pontos fortes e fracos, todas as *skills* prioritizadas e um acompanhamento de métricas (Figura 17).

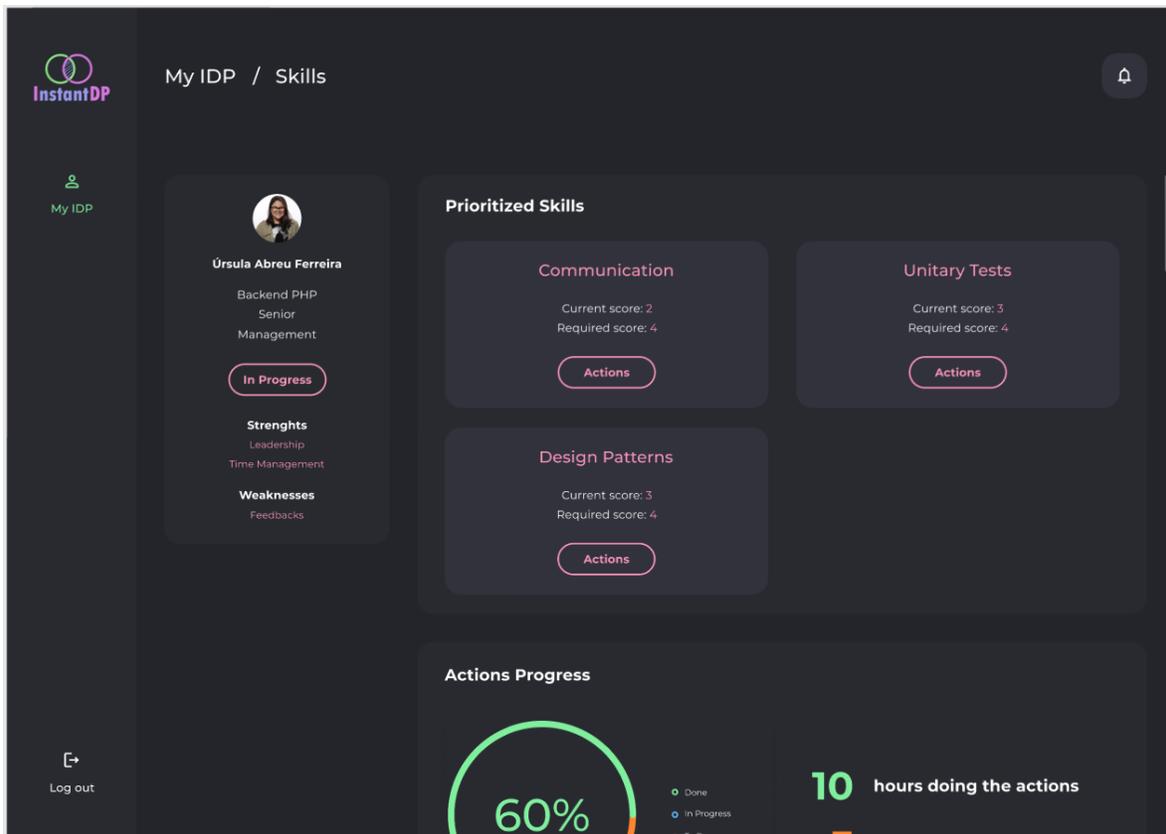


Figura 19 - Tela de um PDI em andamento

Ao clicar no botão *Actions* de uma das *skills* priorizadas, é possível que o usuário visualize todas as ações mapeadas e algumas informações relevantes sobre elas (Figura 18). Na tela de ações de uma *skill* é possível ver o progresso do usuário e a lista de ações mapeadas. A ferramenta possibilita que o usuário altere o *status* de suas ações para *In Progress*, quando está trabalhando nesta ação, ou para *Done*, quando finaliza a ação. Também é possível ver a duração em horas da ação, possibilitando que o usuário tome a decisão de qual ação trabalhar no momento baseado no tempo que o usuário possui livre.

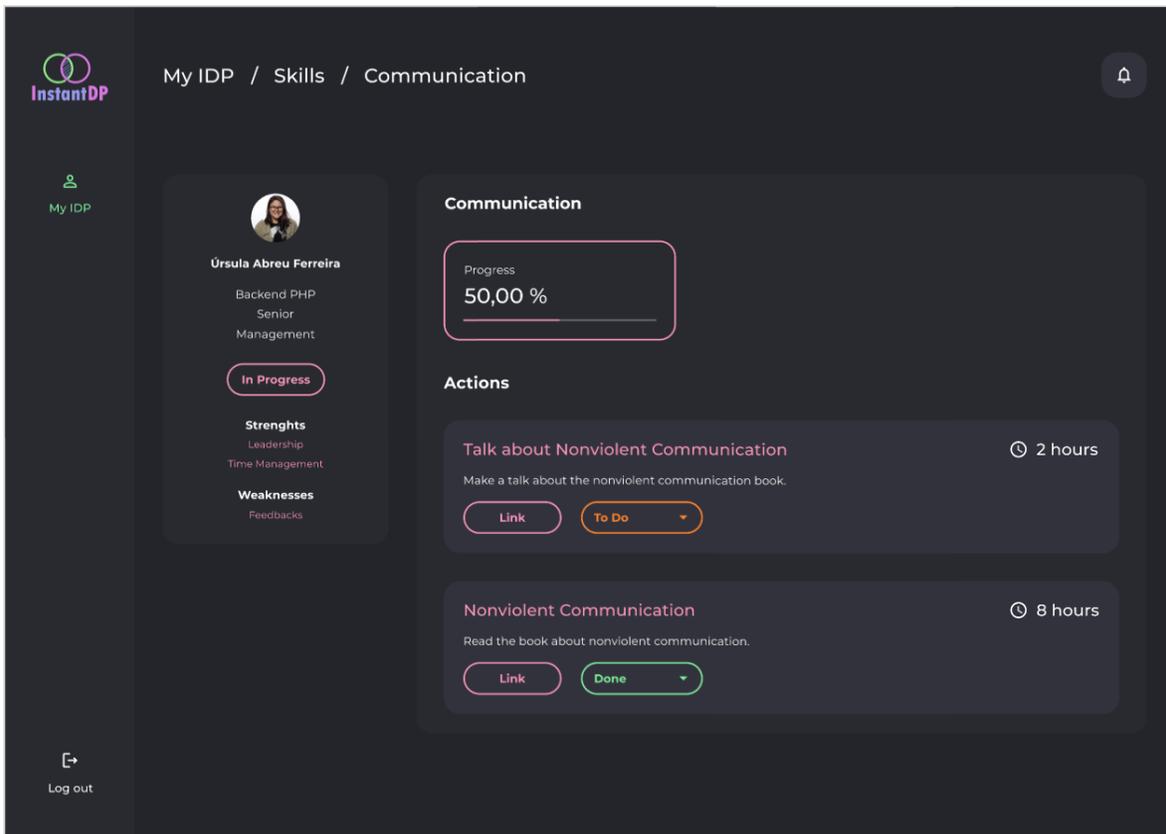


Figura 20 - Tela de ações de uma *skill* priorizada

Por fim, é possível que o usuário acompanhe suas métricas na tela inicial do seu ciclo (Figura 19). A tela apresenta o percentual de ações em cada status, a quantidade de horas já dedicadas realizando ações, a quantidade de horas restantes e um acompanhamento semanal de quantas ações foram concluídas. O intuito dessas informações é trazer para o usuário a visão do quanto ele está dedicando de tempo na realização das ações e, conseqüentemente, na sua evolução de carreira.

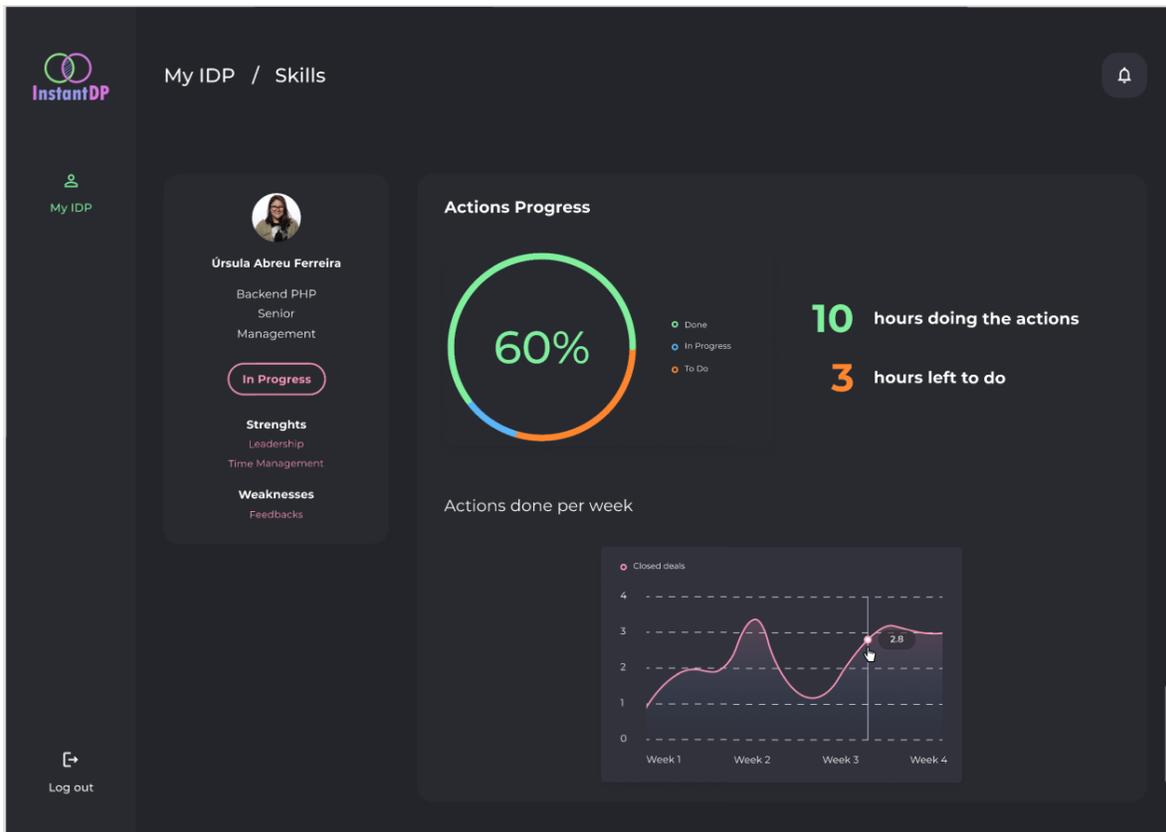


Figura 21 - Tela de um PDI em andamento, mostrando algumas métricas

Na perspectiva de um usuário que atua como Líder de Equipe, além do menu lateral conter a opção *My IDP*, existem também as opções *Team* e *Metrics*. Clicando na opção *Team*, o usuário pode visualizar todas as equipes nas quais exerce o papel de líder e, selecionando uma delas, consegue visualizar todos os membros, com informações sobre o nome, cargo e status do ciclo de PDI (Figura 20).

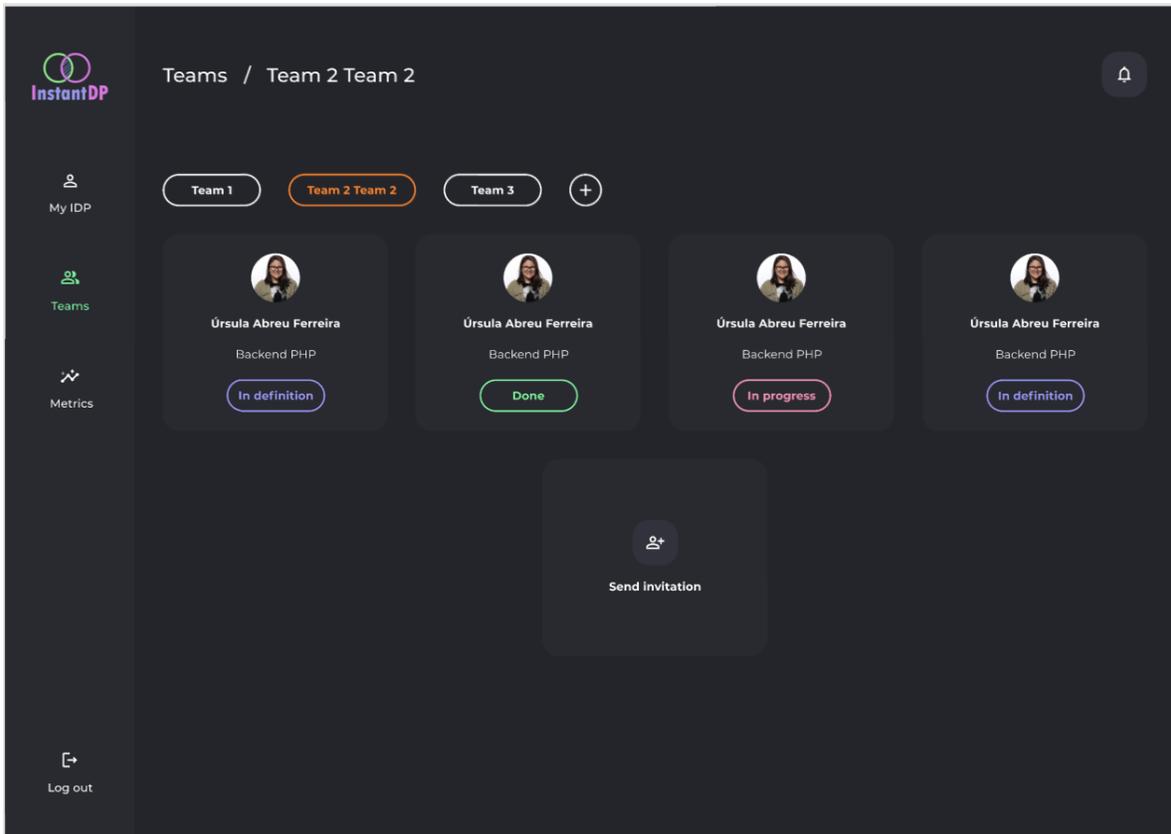


Figura 22 - Tela de equipes de um líder

O usuário pode, ao clicar no *card* de um membro da equipe, visualizar seu perfil e ciclo de PDI. Esta funcionalidade permite que o usuário acompanhe o andamento da definição de um PDI (Figura 21) ou das ações em andamento para PDIs definidos (Figura 22). Este tipo de acompanhamento pode gerar *insights* e *feedbacks* para os membros das equipes, podendo o usuário analisar o tempo que está dedicando na realização das ações, por exemplo.

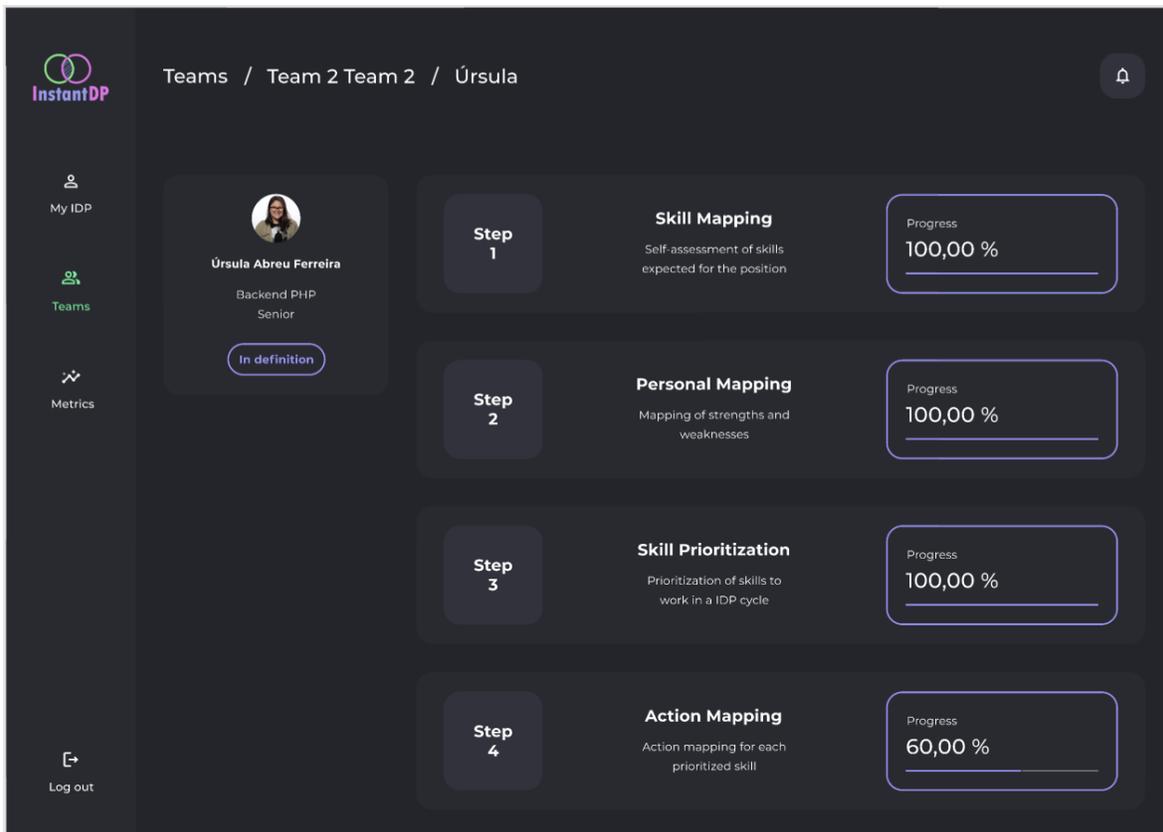


Figura 23 - Tela do membro de uma equipe que está definindo o PDI

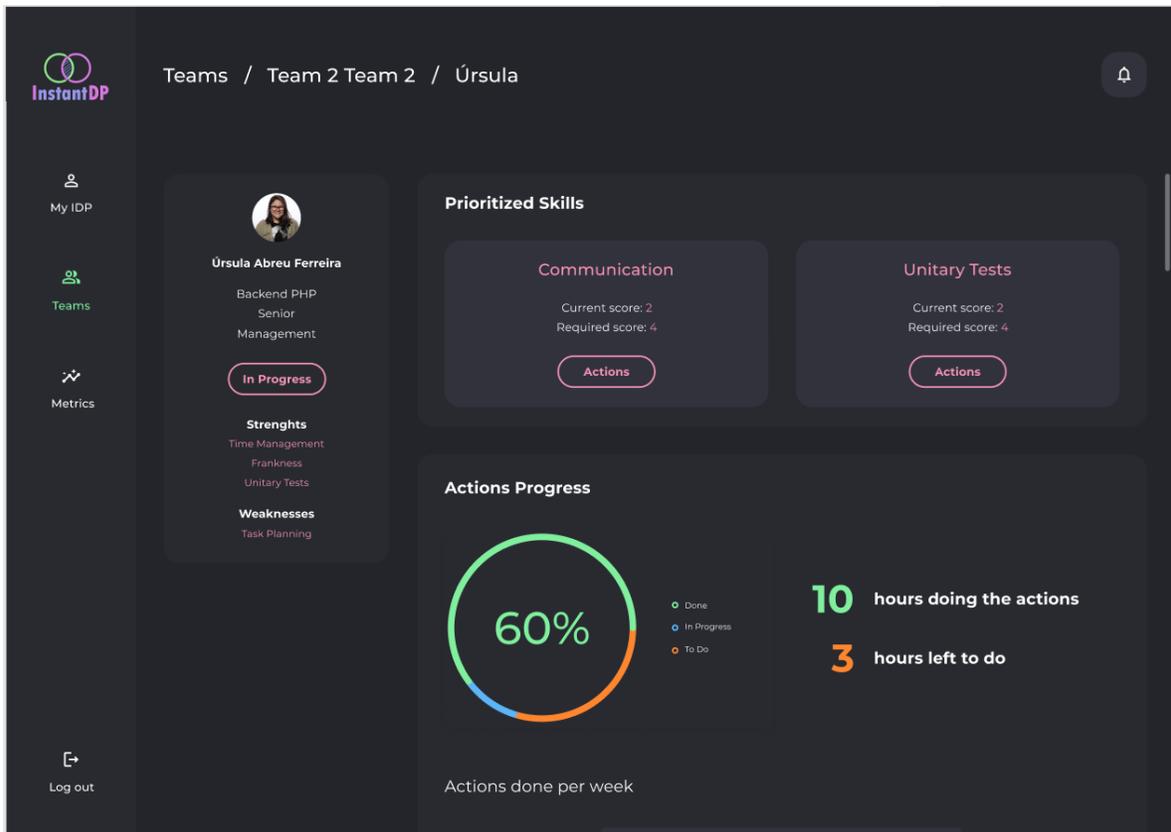


Figura 24 - Tela do membro de uma equipe que possui o PDI definido

Clicando na opção *Metrics* do menu lateral, o usuário poderá acompanhar as métricas de cada uma das suas equipes. Nesta tela (Figura 23), o usuário tem a possibilidade de escolher de qual equipe deseja visualizar as métricas e escolher o cargo. A ferramenta traz, por cargo, as três *skills* mais priorizadas e as três *skills* nas quais os membros possuem maior nível. A informação das *skills* mais priorizadas traz para o usuário quais são as *skills* nas quais a equipe possui um déficit maior e pode auxiliar na contratação de mais pessoas que possuam experiência e conhecimento nessas *skills* para completar a equipe. Conhecer as *skills* que os membros possuem maior nível ajuda a mapear as fortalezas da equipe, podendo a equipe se tornar referência dentro da empresa.

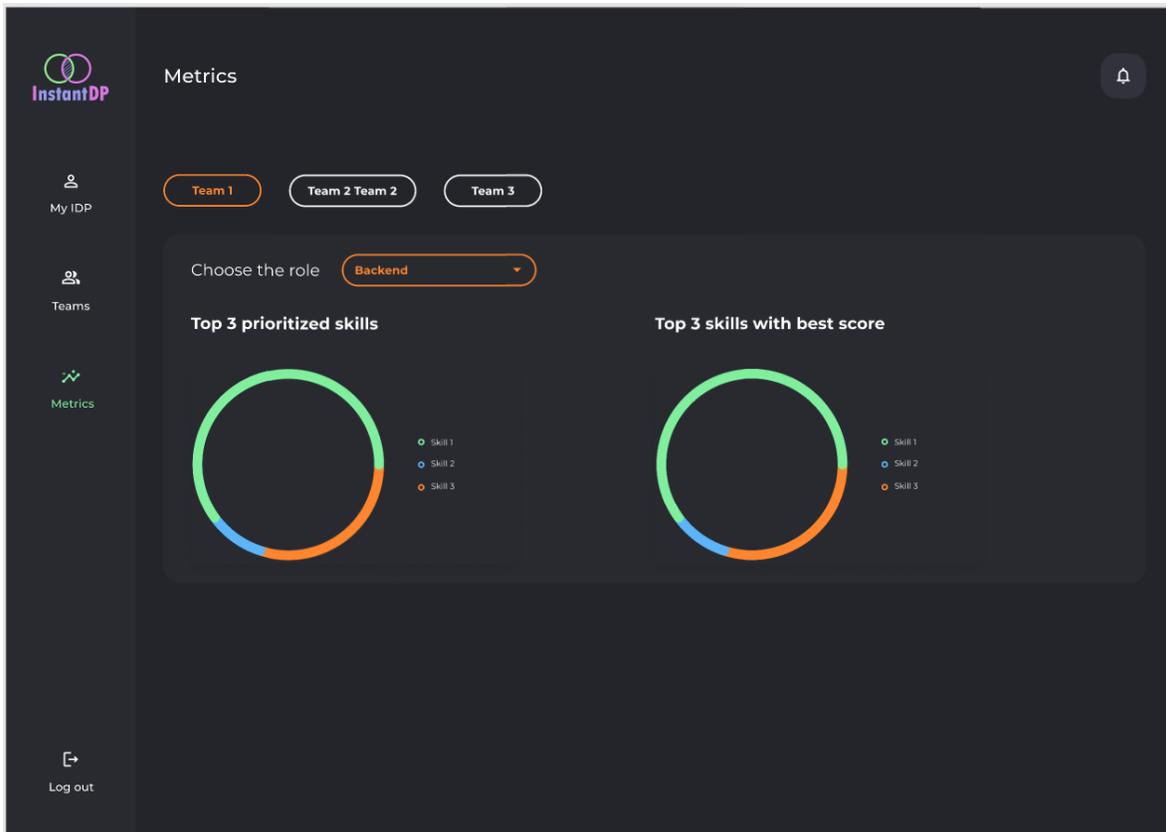


Figura 25 - Tela de métricas das equipes por cargo

4.4 Considerações Finais do Capítulo

Este capítulo apresentou a ferramenta InstantDP desenvolvida para dar apoio à execução do processo definido. Foram apresentados seus requisitos e sua modelagem conceitual, seguindo com detalhes do projeto de sistema e finalizando com a apresentação das telas da ferramenta ilustrando seu uso. No próximo capítulo, são apresentadas as considerações finais do trabalho.

Capítulo 5

Considerações Finais

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

5.1 Conclusão

Este trabalho propôs duas contribuições: um processo de definição e acompanhamento de um PDI, desenvolvido com o propósito de estruturar um processo que possa atender necessidades de empresas que aplicam metodologias ágeis e que se preocupam com a evolução de carreira dos membros da empresa e, ao mesmo tempo, seja simples de ser aplicado; e uma ferramenta (InstantDP), que tem como propósito apoiar a execução do processo definido, permitindo que seus usuários definam e acompanhem os ciclos de PDI.

Ao longo do desenvolvimento do projeto, foram praticadas inúmeras habilidades desenvolvidas durante a graduação em diferentes disciplinas. Os conhecimentos mais utilizados foram obtidos através das seguintes disciplinas:

- *Programação II e III*: foram as disciplinas que deram toda a base de conhecimentos essenciais para a implementação da ferramenta InstantDP, no que diz respeito às linguagens em si. Nessas disciplinas foram abordados assuntos relacionados a lógica de programação e paradigmas de programação orientada a objeto e estruturada, sendo utilizadas fortemente durante a implementação da ferramenta desenvolvida. Esses conhecimentos, no contexto das disciplinas, foram aplicados em linguagens diferentes das linguagens utilizadas na implementação deste trabalho, porém conhecer outras linguagens acelera a curva de aprendizado de novas linguagens.
- *Engenharia de Software*: nessa disciplina foi tratado conhecimento necessário para o levantamento e análise de requisitos e modelagem de casos de uso, diagramas de classes e de projeto de sistema. Esse conteúdo tornou as tomadas de decisões de implementação mais claras e precisas, além de facilitar a organização do tempo

para desenvolvimento, por prover previsibilidade quanto às funcionalidades necessárias para a ferramenta.

- *Banco de Dados*: essa disciplina proveu toda a base de conhecimento com relação à estruturação, modelagem e utilização de persistência em bancos de dados relacionais, tornando a utilização na ferramenta mais simples.
- *Interface Humano-Computador*: essa disciplina foi fundamental para o desenvolvimento dos protótipos e realização de testes de usabilidade, provendo apoio teórico e prático e possibilitando o desenvolvimento de uma interface testada e validada por usuários reais.

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 3 - Objetivos e sua situação na conclusão da monografia

Objetivo	Status	Resultado
Descrever o processo de definição e acompanhamento de um PDI.	Atendido	Processo de definição e acompanhamento de um PDI (vide seção 3.1).
Elaborar a modelagem conceitual de uma ferramenta que apoie a execução do processo definido.	Atendido	Requisitos, modelos de casos de uso e de classes da ferramenta InstantDP (vide seção 3.2).
Definir e projetar uma arquitetura para a ferramenta, levando em consideração padrões, modelos e paradigmas utilizados em projeto de sistemas de software.	Atendido	Projeto da arquitetura e projeto detalhado da ferramenta InstantDP (vide Seção Erro! Fonte de referência não encontrada.).

Tabela 3 - Objetivos e sua situação na conclusão da monografia (cont.)

Objetivo	Status	Resultado
Implementar, testar e disponibilizar para uso um MVP (<i>Minimum Viable Product</i>) da ferramenta.	Atendido	Ferramenta InstantDP (vide Seção Erro! Fonte de referência não encontrada.).

Entre os desafios para o desenvolvimento desse trabalho, tanto para a estruturação do processo quanto para o desenvolvimento da ferramenta, destaca-se a dificuldade de criar um processo que possa atender diferentes empresas que apliquem metodologias ágeis e garantir que seja um processo simples, trazendo resultado para a empresa e para os membros da empresa. O processo proposto foi definido pela autora durante sua atuação no PicPay e foi utilizado na prática, mostrando-se viável. Outra dificuldade relevante foi desenvolver uma ferramenta que englobe todas as etapas do processo, buscando-se boa usabilidade e simplicidade.

Como abordado em capítulos anteriores, o MVP da ferramenta InstantDP, do lado do cliente (*front-end*), possui algumas limitações como a ausência de tela de *login* e *logout* e CRUD de alguns elementos da ferramenta. Tais limitações ocorreram devido a restrições de tempo para a conclusão do trabalho e, por ser um MVP, foram priorizadas as funcionalidades ligadas ao objetivo principal do processo, a definição e o acompanhamento de um PDI. Desta forma, a ferramenta foi disponibilizada apenas de maneira local e as limitações da ferramenta podem ser desenvolvidas em trabalhos futuros, melhorando a ferramenta para o seu uso na prática.

Vale destacar que o processo foi testado em várias equipes em que a autora deste trabalho atuou como líder em um período de um ano, possibilitando validar que o processo é simples e coeso o suficiente para ser utilizado em situações similares em empresas que aplicam metodologias ágeis. O uso do processo pela autora mostrou resultados positivos com relação ao desenvolvimento pessoal e profissional dos membros das equipes. Os testes realizados também possibilitaram levantar melhorias para o processo, chegando à versão apresentada neste trabalho.

5.2 Trabalhos Futuros

Como trabalhos futuros, os seguintes pontos foram identificados como melhorias na ferramenta:

- Implementar o CRUD, do lado do cliente da ferramenta, para:
 - Empresas;
 - Membros da empresa;
 - Plano de carreira, contemplando: todos os cargos, níveis de senioridade de cada cargo, *skills* necessários para cada cargo e o nível de cada uma dessas *skills* para cada senioridade de um cargo;
 - Equipes;
 - Mais ciclos de PDI, visto que a ferramenta só implementa um.
- Implementar telas de *login* e *logout*.
- Implementar melhorias de usabilidade na ferramenta, principalmente com melhores formas de exibição em caso de erros aos usuários.
- Melhorar as descrições das etapas de definição de um PDI, para tornar mais simples o entendimento e objetivo de cada etapa.
- Implementar visualização de um ciclo de PDI finalizado, uma vez que hoje a ferramenta só contempla o caso de ciclo em desenvolvimento.
- Implementar a possibilidade de criar uma ação para uma *skill* priorizada, após a conclusão das etapas de definição.
- Adicionar uma tela ao final de cada etapa do processo com finalidade de visualizar as escolhas e conectar com a próxima etapa.
- Na etapa de *Skill Prioritization*, adicionar mais informações sobre as *skills* para auxiliar na escolha das *skills*, por exemplo: se é uma *skill* gerada a partir da etapa de *Personal Mapping* ou *Skill Mapping* e qual o *level* atual e qual o esperado para *skills* geradas na etapa de *Skill Mapping*.
- Melhorar as métricas individuais e métricas das equipes, trazendo mais informações relevantes e que possam ser utilizadas para tomadas de decisão.
- Criar uma tela de *Home*, personalizando a tela inicial da ferramenta e trazendo conteúdos e informações relevantes para o contexto de plano de carreira.

- 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 aos usuários com dúvidas comuns e relevantes.
- Adicionar integração com ferramentas como o Jira¹⁸, trazendo as métricas de entregas dos membros das equipes e das equipes para a ferramenta para, por exemplo, validar as ações que possuem aplicação em projetos e tarefas do time.

Além disso, como trabalho futuro, uma vez que a ferramenta seja viabilizada para uso, pretende-se disponibilizá-la para uso por outras pessoas, em outras organizações, a fim de avaliar se o processo e a ferramenta são adequados para cenários diferentes dos que foram considerados até o momento, por exemplo em equipes tradicionais ou híbridas.

¹⁸ <https://www.atlassian.com/br/software/jira>

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 2021.
- APPELO, Jurgen: Management 3.0: Leading Agile Developers, Developing Agile Leaders. Prentice Hall, 2010.
- ARAGÃO, Marina: PicPay cresce na pandemia e tem 450 vagas de emprego abertas. Disponível em: <https://economia.estadao.com.br/noticias/geral,picpay-cresce-na-pandemia-e-tem-450-vagas-de-emprego-abertas,70003601383>. Acesso em setembro de 2021.
- AURUM, A, WOHLIN, C., Engineering and Managing Software Requirements, Springer-Verlag, 2005
- 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 setembro de 2021.
- BASS, L., CLEMENTS, P., KAZMAN, R., Software Architecture in Practice, Second edition, Addison Wesley, 2003
- BECK, K. S. K. S. J. E. A.: Manifesto for agile software development. Disponível em: <<http://agilemanifesto.org/>>. Acesso em maio de 2021.
- 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.
- PAZ, Daniel: Gestão de equipes ágeis. Disponível em: <<https://www.devmedia.com.br/gestao-de-equipes-ageis/27505>>. Acesso em maio de 2021.
- PRESSMAN, R. S.; LOWE, D. Web engineering: a practitioner's approach. [S.l.]: McGraw-Hill Education, 2009. v. 1.
- RIEHLE, D.: Framework Design: A Role Modeling Approach. Ph.D. Thesis, No. 13509. Zurich, Switzerland, ETH Zurich, 2000.

ZHUO, Julie: The Making of a Manager: What to Do When Everyone Looks to You. Portfolio, 2019.