

# Launchpad: Guia de Onboarding para Desenvolvedores

**Aluno:** Leandro Marques Venceslau de Souza  
*Departamento de Ciências da Computação*  
*Universidade Federal de Minas Gerais*  
Belo Horizonte, Brasil  
lmvsminerador123@ufmg.br

**Orientador:** Marco Tulio Valente  
*Departamento de Ciências da Computação*  
*Universidade Federal de Minas Gerais*  
Belo Horizonte, Brasil  
mtov@ufmg.br

**Resumo**—Em cenários de substituição de pessoas, o qual é comum ao contexto de criação e manutenção de software [1][2][3], a perda de conhecimento e experiência de contribuidores antigos associada à necessidade de treinamento e inclusão de novos talentos pode impactar significativamente o custo e a duração de um projeto. Casos como esses podem levar a acréscimos de 40% a até 60% no custo de desenvolvimento de um sistema [4]. Como então se adaptar a esses desafios de forma a estabelecer um processo que auxilia a reter talento humano? Este trabalho procura oferecer um método, organizado como guia, cuja função é auxiliar times focados em software a compreender, estruturar e realizar um onboarding especializado para seu contexto.

**Palavras-chave**—Onboarding, Socialização Organizacional, Integração, Desenvolvimento e Manutenção de Software

## I. INTRODUÇÃO

Este trabalho propõe o “Launchpad”, um guia para a socialização organizacional de desenvolvedores. Esse processo, também conhecido como Onboarding, é definido na literatura de gestão [5][6] como - o mecanismo pelo qual indivíduos são ensinados e aprendem as perspectivas culturais, comportamentos e expectativas associadas à uma função. Segundo Talya N. Bauer e Berrin Erdogan [7], Onboarding está diretamente relacionado à produtividade, satisfação com trabalho e criatividade. No contexto de times focados no desenvolvimento e manutenção de software, esse processo e fatores relacionados se tornam ainda mais relevantes devido:

- À natureza intrinsecamente inventiva do desenvolvimento de software, onde times multidisciplinares procuram novas formas de se resolver problemas através de sistemas de informação. A implementação, disponibilização e manutenção de software requer alto grau de colaboração [8]. Equipes formadas por profissionais de diferentes áreas criam partes menores do todo que formam um sistema ou produto. Elas assim o fazem trocando e integrando continuamente informações; código; documentação e experiências. Como resultado, produzem bases de conhecimento extensas e de difícil manutenção [9].
- Ao problema da alta taxa de rotatividade de funcionários nessa área<sup>1</sup>, historicamente reconhecido em artigos ao

longo do tempo [1][2][3]. Essa situação onera o nível de produtividade e rentabilidade de organizações devido aos altos custo da rotatividade de funcionários, a qual pode chegar de 6 a 9 meses do salário de um empregado, segundo dados da SHRM - The Society for Human Resource Management<sup>2</sup>.

Dessa forma, num mundo em que profissionais de software estão cada vez mais móveis devido à diminuição das barreiras de entrada para o emprego nessa área (acesso e entendimento de conhecimento técnico; hardware e conexão à internet), o ato de se “compreender, gerenciar e aprimorar a experiência de integração de novos funcionários se torna uma vantagem competitiva para empresas” [7]. Tais iniciativas auxiliam na melhora de métricas como taxas de rotatividade/retenção de pessoas, níveis de satisfação, assim como indicadores de desempenho e produtividade como o tempo para produtividade - período de tempo médio que um novo contratado leva para se atualizar e contribuir de forma significativa para a organização.

A hipótese principal e motivadora deste trabalho é que - **primeiras impressões importam**. Um funcionário acolhido por um processo de onboarding estruturado, onde é apresentado aos círculos sociais e modus operandi da sua empresa, sendo sujeito a etapas de treinamento com objetivos bem definidos e apoio, têm maiores chances de permanecer numa organização. Isso pois, desde o início, obteve as informações e suporte necessários para não somente exercer sua função como também assimilar os valores e objetivos estratégicos estabelecidos pela empresa e/ou time.

## II. TRABALHOS CORRELATOS

Na primeira parte da disciplina foi feita uma investigação da literatura acerca do processo de Onboarding ou Socialização Organizacional, estruturada na forma de um artigo. Ele contemplou contribuições que vão desde o trabalho seminal feito por John Van Maanen e Edgar H. Schein em 1979, no artigo “*Toward a Theory of Organizational Socialization*” [6], a até trabalhos mais recentes como o de Gaurav G. Sharma e Klaas-Jan Stol, publicado em outubro de 2019, onde os autores propõem um dos poucos modelos teóricos da literatura capaz de identificar atividades ligadas a um processo de onboarding

<sup>1</sup>A look at the Tech Retention Crisis: Causes, Key Trends & Plausible Solutions.

<sup>2</sup>Essential Elements of Employee Retention - SHRM

de sucesso para desenvolvedores de software [3]. Apesar do foco da pesquisa ter sido equipes de desenvolvimento, para além da ciência da computação a literatura científica estudada também abrangeu - administração [10], gestão de recursos humanos [11] e psicologia [5][7][12][13]. Também foram pesquisadas métricas de avaliação e sucesso presentes na literatura, de forma a destacar indicadores de desempenho usados para avaliar a eficácia do processo no Onboarding corporativo. Dentre os indicadores qualitativos estão - auto-eficácia; clareza de função; conhecimento da cultura; integração social [5]. A revisão também identificou, a partir de artigos específicos ao contexto de projetos de código aberto, duas métricas quantitativas aplicáveis ao contexto deste trabalho - velocidade de contribuição e indicadores baseadas em commit [14][15], além de sua aplicação em casos de estudo em empresas como a Google [16] e TAI [17]. Por fim, constatou ampla evidência que o Onboarding planejado e estruturado está relacionado a melhores níveis de satisfação, produtividade, engajamento e comprometimento de funcionários [13], além da diminuição na taxa de rotatividade de empregados no longo prazo [5]. Esse trabalho investigativo e suas conclusões foram insumo para a produção do documento exposto na próxima seção.

### III. “LAUNCHPAD - GUIA DE ONBOARDING PARA DESENVOLVEDORES”

Esta seção expõe os diferentes capítulos que compõem o Launchpad, o qual foi produzido no formato de um documento markdown disponível na plataforma de hospedagem Github [18]. Além disso, como suporte para o pitch que constitui parte das atividades da Monografia, também foi criado uma apresentação de slides que expõe um caso exemplo da aplicação do processo de socialização organizacional proposto<sup>3</sup>.

#### A. Capítulo 1: Compreendendo “Onboarding”

O processo de Onboarding, também conhecido como Socialização Organizacional, é o processo pelo qual novos colaboradores são integrados em uma organização, adquirindo conhecimento, habilidades, perspectivas culturais e integração com os círculos sociais necessários para o desempenho de suas funções de forma eficaz. A importância do Onboarding reside nos benefícios palpáveis que gera às empresas:

- Aceleração da curva de aprendizado, com redução do tempo para produtividade.
- Fortalecimento da cultura organizacional.
- Aumento do engajamento e comprometimento dos colaboradores com a empresa.
- Maior imersão e compreensão sobre o contexto, objetivos e proposta de valor da organização.
- Melhoria dos níveis de satisfação, com aumento do engajamento e comprometimento de novos funcionários com os objetivos estratégicos da empresa.

Porém, esses benefícios pressupõem não somente estabelecer uma base de conhecimento robusta com identificação

prévia de metas relevantes ao contexto do time, como também a elaboração de uma estrutura de execução e reiteração que faça pleno uso desses recursos.

1) *Dimensões do Onboarding*: Como forma de facilitar a visualização geral do processo, o Launchpad divide o Onboarding em dimensões: Integração; Cultura; Imersão no Domínio; Imersão na Tecnologia.

No que tange a duração, o define-se uma janela de tempo para o processo de Onboarding com unidades de tempo em semanas. Sugere-se um período entre 4 e 12 semanas, a ser modificado de acordo com as características e objetivos do time. Por exemplo - para uma equipe responsável pela manutenção e melhoria de um serviço de gerenciamento de tarefas, na qual o escopo de trabalho é bem definido e interno à empresa, um plano de 4 semanas pode ser uma boa opção. Já para outra equipe que lida com uma plataforma de e-commerce complexa e de grande escala, um plano de Onboarding mais extenso, de 12 semanas, pode ser mais aconselhável devido à maior complexidade técnica no que tange à escalabilidade, segurança e interação com clientes externos.

A **Integração** pode ser entendida como o conjunto de passos que visam promover a rápida e eficaz inserção dos novos colaboradores na organização. Este processo vai além da simples ambientação física, envolvendo também a socialização do indivíduo na cultura organizacional, compartilhamento de informações essenciais sobre a empresa, suas práticas e valores, bem como estabelecimento de vínculos interpessoais. Através da integração, os novos membros obtêm o suporte necessário para se adaptarem ao novo ambiente de trabalho, compreenderem suas funções e responsabilidades.

Como exemplos de passos relacionados à integração, cita-se:

- **Sessões de Boas Vindas**: Momentos focados em introduzir novos colaboradores ao contexto do time, seus valores e objetivos. Visa fornecer informações claras sobre as responsabilidades e expectativas acerca da atuação desses indivíduos. Tais sessões tendem a ter uma estrutura fixa e pré-elaborada.
- **Acolhimento Personalizado**: Alguns times optam por receber novos membros com uma abordagem personalizada, seja ao perfil daqueles que estão chegando ou dos comportamentos que o time deseja promover nessas pessoas. Nesses casos, não se segue uma estrutura pré-definida.
- **Atividades de “Team Building”**: Atos de socialização entre os membros da equipe, como almoços em grupo, jogos colaborativos ou outras atividades que estimulem o relacionamento interpessoal e o espírito de equipe.

No que tange a escopo e tempo, entre as dimensões do Onboarding propostas, Integração é a única que deve ter uma janela de tempo de execução mais estrita. Seu foco é proporcionar uma ótima primeira impressão, uma introdução do novo membro à equipe e seu escopo de trabalho.

A **Cultura** no contexto do Onboarding envolve a transmissão e execução de valores essenciais obtidos a partir da experiência conjunta de um grupo [19]. Através de uma história

<sup>3</sup>Deck de apresentação para o pitch final - Monografia II

compartilhada e de um padrão consistente de liderança, tais valores foram validados como eficazes para o alcance dos objetivos de uma equipe. Uma cultura forte pressupõe a ampla prática e adesão desses valores como forma de gerar coesão e engajamento [10].

Como exemplo, suponha uma equipe de desenvolvimento de software focada num sistema bancário de transações. Nesse caso, é de extrema importância que se garanta a estabilidade e confiabilidade do sistema já que este:

- lida com a integridade dos dados financeiros de seus clientes;
- assegura a continuidade de operações bancárias sem interrupções ou falhas.

Ao longo do tempo, ao promover a qualidade do código a equipe percebeu uma redução de bugs, assim como maior estabilidade do sistema. Através de práticas como revisões de código regulares e testes automatizados, atos estes ativamente encorajados pela liderança do time, o apreço pela qualidade se tornou um valor essencial da equipe.

Uma cultura bem estabelecida torna explícito ao novo funcionário a identidade do grupo - o conjunto de crenças e visão de futuro que desejam promover. Através da assimilação dessa identidade, novos colaboradores são capazes melhor se alinhar com as expectativas organizacionais, demonstrando maior satisfação no trabalho, engajamento, comprometimento e senso de pertencimento. Dessa forma, a cultura se torna uma ferramenta poderosa na formação de uma equipe coesa e no estabelecimento e adesão de uma identidade compartilhada.

Quanto à tempo e escopo, Cultura é uma dimensão que deve ser exercida continuamente durante todo o processo de Onboarding. Ela não é apenas um conjunto de atividades a serem executadas. Ela é uma dimensão ampla pois envolvem valores - sua identificação, exercício e manutenção.

“Domínio de uma aplicação” pode ser entendido como o conjunto de regras de negócio, problemas, processos e conceitos que uma aplicação é projetada para suportar ou solucionar [20]. Já **Imersão no Domínio** é uma etapa do processo de Onboarding que visa promover em novos colaboradores a compreensão do negócio, de seus objetivos e contexto operacional. Isso através de, principalmente, a realização de demandas reais. Tal entendimento paralelo à prática é essencial para que os desenvolvedores possam:

- Criar soluções que não somente funcionem tecnicamente, mas que também agreguem valor real ao negócio.
- Ter um tempo para produtividade menor e, dessa forma, contribuir mais cedo para os propósitos do time.
- Enxergar com mais facilidade o propósito de seu trabalho e como ele contribui para o sucesso da organização, auxiliando, assim, em gerar motivação e engajamento.

A título de exemplo, possíveis atividades a serem consideradas nessa dimensão seriam:

- **Mentoria:** Designar um mentor experiente para novos funcionários. Essa pessoa, através de um acompanhamento individualizado, terá a função de guiar, esclarecer dúvidas e compartilhar conhecimento.

- **Treinamento e Capacitação:** Execução de treinamentos e/ou atividades focadas em capacitar o novo colaborador nas tecnologias e especificidades técnicas do time. Exemplos incluem: Workshops em ferramentas e frameworks relevantes; Sessões de hands-on simulando a resolução de demandas do projeto; Pair programming.
- **Execução de demandas:** Atribuir tarefas, com aumento gradual de complexidade, de forma a promover a participação de novos talentos na execução de melhorias e/ou manutenção do sistema em que o time atua.

No que tange a sua relação com outras dimensões do Onboarding, “Imersão no Domínio” deve andar paralelamente à “Imersão na Tecnologia”. Ambas serão o foco principal do processo após a fase inicial de Integração.

Enquanto a dimensão anterior foca no negócio e nos impactos que a aplicação sob responsabilidade do time tem na organização e stakeholders, a **Imersão na Tecnologia** lida com os artefatos e padrões técnicos utilizados pela aplicação. Em outras palavras, enquanto a “Imersão no Domínio” está focada nos requisitos funcionais e impactos gerados pela aplicação, a “Imersão na Tecnologia” tem como meta gerar compreensão dos requisitos não funcionais, ou seja, dos critérios que especificam como o sistema deve operar para atingir seus objetivos [20]. De forma a ilustrar os possíveis tópicos a serem considerados nessa dimensão do Onboarding tem-se:

- Qual o paradigma de programação utilizado na aplicação? Por qual motivo ele foi escolhido? O que de positivo ele traz?
- No que tange à padrões arquiteturais, qual a forma de estruturação segue a aplicação e quais os trade-offs associados à essa escolha?
- Quais práticas são utilizadas no que tange à testes e critérios de qualidade de código? Quais os tipos de testes praticados, assim como ferramentas e frameworks de suporte? Quais as melhores práticas a serem seguidas para garantir qualidade e segurança para a aplicação?
- Qual o processo utilizado para o Gerenciamento de Configuração de Software (SCM)? Quais os fluxos de trabalho comuns ao projeto?
- Como funciona o processo de revisão de código? Quais os critérios de qualidade e padronização de código seguidos?
- Quais as práticas de DevOps seguidas no projeto? Quais as ferramentas e processos envolvidos nesse contexto?

## B. Capítulo 2: Estabelecendo o Alicerce

O Launchpad foi desenvolvido a partir da crença de que a documentação é um alicerce para o Onboarding. Quanto mais ampla a gama de processos, valores, requerimentos, padrões arquiteturais e funcionalidades chave documentadas, mais robusto e menos dispendioso será a Socialização Organizacional.

Robusto pois a documentação torna tangível e consultável acordos tácitos e decisões relevantes ao contexto de atuação de um time, facilitando não somente o acompanhamento de seu progresso ao longo do tempo como também o entendimento do domínio e cultura da equipe.

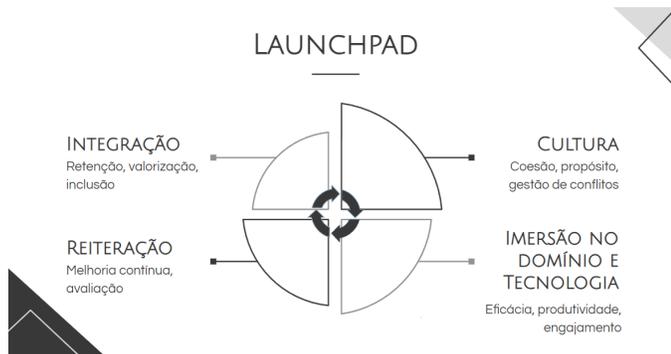


Fig. 1. Diagrama de dimensões.

Menos dispendioso pois, independente do contexto, times de desenvolvimento de software lidam com a geração e manutenção de conhecimento. Conhecimento pode ser transitório, seja porque não é mais aplicável ao contexto devido à mudanças do próprio negócio ou por forças externas (ex: mudanças de normas e leis), seja pela rotatividade de funcionários comum ao cenário de implementação e manutenção de software. Além disso, transmiti-lo é um ato que requer tempo e dedicação. Uma documentação organizada para consulta sobre demanda facilita o gerenciamento de conhecimento. Ela reduz os impactos negativos da saída de funcionários ao tornar consultável partes de seu saber. Ela facilita a transmissão de conhecimento de forma assíncrona, diminuindo assim a necessidade de acompanhamento direto de novos funcionários por parte de membros ativos no time, os quais, muitas das vezes, precisam conciliar múltiplas responsabilidades dentro do tempo limitado que possuem para a realização de suas atividades diárias.

Dito isso, bases de conhecimento para times de desenvolvimento são geralmente extensas e de difícil manutenção. Dessa forma, é importante que a experiência prática do Onboarding seja apenas uma parte do todo, idealmente apresentando aos novos talentos uma parte específica na qual terão maiores chances de trabalhar nas etapas iniciais de sua atuação. Acredita-se que uma experiência introdutória focada leva aos seguintes efeitos desejados como resultados do Onboarding - satisfação; engajamento e diminuição do tempo para produtividade.

As seções deste capítulo - Fundamentação; Mapa de Dependências; Documentação Técnica; Banco de Tarefas - definem a estrutura do Alicerce: o documento base para o processo de Socialização Organizacional proposto pelo Launchpad. Sua meta é não somente explicitar e facilitar a tarefa de identificação do contexto, objetivos, aspectos sócio comportamentais e processos internos relevantes à novos membros da equipe, como também prover os insumos necessários à realização de um processo de Onboarding de sucesso.

1) *Fundamentação*: É importante que se entenda o cenário no qual uma equipe atua antes de se iniciar a execução do processo de Onboarding, já que é este meio que delimita o que é relevante para o sucesso do time e, conseqüentemente, a

tomada de decisões e delimitação de metas para a Socialização Organizacional. As necessidades de um time focado no design de interfaces web interativas são diferentes das de um time focado na análise de dados. Enquanto o primeiro tem como prioridades a criação de layouts responsivos e a otimização da experiência do usuário, o segundo tem como foco a transformação de dados brutos em informações e visualizações relevantes à tomada de decisões de negócio.

Dessa forma, é crucial que se torne explícito, além de documentado, a conjuntura na qual um time está inserido. Para auxiliar nesse processo, propõe-se como exercício inicial de delimitação de escopo e exploração de contexto a resposta do seguinte modelo de perguntas, mesmo que de forma parcial:

- 1) Como um time de desenvolvimento de software, nossa área de foco é "...". Atuamos na "...".
- 2) Nesse contexto, temos como objetivos "...".
- 3) Esses objetivos agregam valor ao negócio pois "...", o que está alinhado com os seguintes objetivos estratégicos da organização "...".
- 4) Nos esforçamos para alcançá-los levando em consideração os valores-chave da empresa "...".
- 5) Como um time, nossas crenças fundamentais são que "...".

A seguir expõe-se um exemplo de resposta baseado nesse modelo:

- 1) Como um time de desenvolvimento de software, nossa área de foco é **Front-end**. Atuamos na **criação e manutenção de interfaces web focadas em e-commerce, baseado no design system desenvolvido pela empresa**.
- 2) Nesse contexto, temos como objetivos **aprimorar a usabilidade, acessibilidade e experiência de utilização desses sistemas, garantindo consistência visual e funcional com os padrões de design estabelecidos**.
- 3) Esses objetivos agregam valor ao negócio pois **contribuem para aumentar o engajamento dos usuários, melhorar as taxas de conversão e fortalecer a identidade das nossas marcas online**. Isso está alinhado com os seguintes objetivos estratégicos da organização - **aumentar sua presença em mídias digitais e potencializar vendas online**.
- 4) Nos esforçamos para alcançar esses objetivos levando em consideração os valores-chave da empresa: **foco no cliente, excelência e inovação**.
- 5) Como um time, nossas crenças fundamentais são que **a colaboração e a comunicação eficaz são essenciais para o sucesso, e que o aprendizado contínuo junto à adaptabilidade são cruciais em um ambiente de constante evolução tecnológica**.

2) *Mapa de Dependências*: O Mapa de Dependências tem como função detalhar as necessidades logísticas, legais, tecnológicas e de acesso que novos colaboradores devem atender para começar a contribuir. Ele está dividido nas seções: Com-

pliance; Acessos necessários e procedimentos de requisição; Referências para consulta sob demanda.

**Compliance** é o conjunto de materiais e recursos destinados a assegurar que novos colaboradores compreendam e sigam as políticas, regulamentações e normas da organização. Tais documentos podem tomar diversas formas, desde contratos formais a slides e organogramas. Vale ressaltar que nem todo time de desenvolvimento de software precisa, necessariamente, se preocupar em mapear os pontos levantados nesta sessão. Isso pois, em alguns casos, a organização da qual o time faz parte é a responsável pelo mapeamento, execução e monitoramento de critérios de compliance. Esse conjunto de documentos, considerando o contexto de desenvolvimento, tem como pontos a serem considerados:

- **Obrigações legais e contratuais:**

- Código de conduta: Documento que destaca expectativas de comportamento, ética no trabalho assim como diretrizes para resolução de conflitos.
- Políticas de segurança: Instruções quanto à segurança da informação, incluindo práticas recomendadas para senhas, treinamentos periódicos em cibersegurança, melhores práticas no uso de dispositivos pessoais e do trabalho.
- Política de dados e privacidade: Registro das leis de segurança e privacidade dos dados a serem seguidas no exercício do trabalho de desenvolvimento. Se o escopo de trabalho do time envolve o Brasil e dados de cidadãos brasileiros, por exemplo, nesse caso a equipe precisa garantir que está seguindo a Lei Geral de Proteção de Dados Pessoais <sup>4</sup>.

- **Recursos necessários à produção:**

- Hardware: Documento, lista ou inventário delimitando os dispositivos físicos necessários ao exercício do trabalho, assim como suas respectivas especificidades técnicas. Itens a serem considerados incluem, mas não estão limitados a: Computador; Notebook; Monitores; Processadores; Dispositivo de Armazenamento e/ou Backup.
- Software e licenças: Lista os softwares necessários, incluindo ferramentas específicas para compliance e segurança, como antivírus, firewalls e sistemas de monitoramento. Dentre exemplos cita-se: Licenças de sistemas operacionais, ferramentas de desenvolvimento (IDEs, editores de texto), software de gestão de projetos (Jira, Trello), e ferramentas de colaboração (Slack, Microsoft Teams).
- Suporte técnico e manutenção: Delimitação dos contatos para a resolução de problemas técnicos e de manutenção de equipamentos. Essa esfera de documentos também pode incluir procedimentos e regras a serem seguidas no que tange à atualização de software e hardware (ex: frequência desejada; versões suportadas; etc).

Nome da Ferramenta	Utilidade da Ferramenta	Papel e Permissões	Procedimento de Requisição	Responsáveis pela Aprovação
GitHub	Plataforma de controle de versão	Desenvolvedor: Leitura e Escrita	Preenchimento de formulário	Pessoa X - <a href="mailto:email@email.com">email@email.com</a>
Jenkins	Serviços de CI/CD	Administrador: Projeto X	E-mail para responsável	Pessoa Y - <a href="mailto:email@email.com">email@email.com</a>
Jira	Gerenciamento de Projetos	Usuário: Leitura	Solicitação via portal TI	Pessoa Z - <a href="mailto:email@email.com">email@email.com</a>

Fig. 2. Caso exemplo - Tabela de Acessos

- Ergonomia: Conjunto de regras e equipamentos necessários à garantia de conforto e saúde dos membros da equipe. Sua função é prevenir lesões ocupacionais. Isso inclui a escolha de cadeiras ergonômicas, mesas ajustáveis, suportes para monitores, teclados e mouses ergonômicos, iluminação e ventilação adequada, além da organização do espaço de trabalho para reduzir o esforço físico e mental dos colaboradores.

A segunda seção do Mapa de Dependências - **Acessos e procedimentos de requisição** - lida com o fato de que, independente do porte de um time e natureza do sistema com que trabalham, sempre haverá uma suíte de ferramentas internas e externas utilizadas no exercício das funções da equipe. Dessa forma, faz-se necessário um catálogo dessas ferramentas, assim como do nível de permissão e formas apropriadas para requisição de acesso.

No que tange à permissões, esse item também pode ser entendido como os diferentes conjuntos de ações ou operações específicas que determinados papéis dentro de um sistema podem realizar. Considerando o momento de Onboarding onde funcionários estão aprendendo um novo cenário, pode ser mais seguro dar a essas pessoas um papel com nível de permissões limitada, no intuito de reduzir possíveis problemas (ex: deleção de dados; modificação indevida de configurações críticas; acesso não autorizado a informações sensíveis). Assim, num cenário ideal esse quesito deve estar de acordo com as designações de compliance levantadas no item anterior, sejam elas estipuladas pelo próprio time através de melhores práticas de segurança como o princípio do menor privilégio, obrigações contratuais para com o cliente ou políticas da organização na qual o time está inserido.

O catálogo dessas informações pode tomar diferentes formas - planilhas, tabelas e diagramas. A nível de simplificação, sugere-se ao menos uma tabela com as seguintes colunas: nome da ferramenta, utilidade, papel e permissões, procedimento de requisição, responsáveis pela aprovação.

A terceira e última seção do Mapa de Dependências - **Referências para consulta sob demanda** - procura estabelecer medidas para facilitar a navegação da base de conhecimentos do time por novos colaboradores. Reiterando, novos colaboradores devem ser apresentados a um filtro do todo, sendo esse idealmente alinhado com o que essas pessoas terão maiores chances de atuar nos momentos iniciais de sua carreira na equipe. Assim, sugere-se a criação e/ou separação dos seguintes artefatos num local único, de forma a facilitar sua consulta e atualização contínua: processos; links relevantes no

<sup>4</sup>LGPD ; L'13709

Nome	Papel na Equipe	Contato	País/Localização	Timezone
Maria Silva	Desenvolvedora Backend	<a href="mailto:maria.silva@email.com">maria.silva@email.com</a>	Brasil	GMT-3
Julia Santos	Desenvolvedora Front-end	<a href="mailto:julia.santos@email.com">julia.santos@email.com</a>	Brasil	GMT-3
João Souza	Novo Colaborador - Desenvolvedor Front-end	<a href="mailto:joao.souza@email.com">joao.souza@email.com</a>	Portugal	GMT+1
Ana Pereira	Analista de QA	<a href="mailto:ana.pereira@email.com">ana.pereira@email.com</a>	EUA (Califórnia)	GMT-7
Carlos Lima	Engenheiro DevOps	<a href="mailto:carlos.lima@email.com">carlos.lima@email.com</a>	Índia	GMT+5:30
Luisa Fernandes	Designer UX/UI	<a href="mailto:luisa.fernandes@email.com">luisa.fernandes@email.com</a>	Austrália (Sydney)	GMT+10

Fig. 3. Caso Exemplo - Tabela de Membros

dia a dia; tabela de membros.

A título de exemplo, pense em uma equipe internacional focada na criação e manutenção de um e-commerce web. Houve uma recente contratação de um desenvolvedor Front-end que irá substituir o atual membro do time. Dentre as possíveis referências a serem mapeadas, poderíamos considerar:

- **Processos**

- SCM
- Atualização de ambiente
- Requisição de deployments

- **Links relevantes no dia a dia**

- Ambientes (dev; produção)
- Repositórios
- Templates de design

3) *Documentação Técnica*: Esta seção do Alicerce tem como foco o desenvolvimento e/ou separação dos documentos técnicos relevantes ao contexto de uma pessoa sendo submetida ao processo de Onboarding. O grau de abrangência e granularidade dessa documentação depende principalmente de dois fatores: do contexto em que o time está inserido; das partes específicas do todo que compõem uma aplicação ou projeto nas quais um recém-contratado terá mais chances de atuar nas fases iniciais de sua atuação.

A principal ferramenta que auxilia nesse processo de filtragem, de escolha do que deve compor a documentação técnica destinada a recém-integrados, é a Fundamentação do Alicerce. Fazendo alusão a exemplos expostos anteriormente, um time com aplicação voltada para a análise de dados poderia dar ênfase a itens de documentação como:

- Padrões de modelagem de dados, com uma descrição detalhada dos esquemas de banco de dados, tabelas principais e suas relações, incluindo exemplos de consultas comuns utilizadas.
- Pipelines de Dados: Explicação dos processos de ETL (Extração, Transformação e Carga), com diagramas de fluxo de dados e exemplos de scripts de transformação.
- Guias de uso para ferramentas relevantes ao processo de análise e formulação de informações a partir dos dados obtidos (bibliotecas, frameworks, etc), incluindo tutoriais e casos exemplo, quando possível.

Já outro time focado na manutenção e extensão de um e-commerce poderia enfatizar itens como:

- Documentação sobre a arquitetura de microsserviços, incluindo diagramas de interação entre eles.

- Exemplos de fluxos de chamadas de API relevantes à realização de tarefas como o processamento de pedidos e manipulação de inventário.
- Guias detalhados sobre como integrar e testar diferentes gateways de pagamento, com exemplos de configuração e código para a realização de transações seguras.

O Launchpad, como um guia, tem caráter propositivo. Dessa forma abaixo encontram-se alguns dos tópicos, comuns ao ofício de desenvolvimento e manutenção de software, que podem ser considerados na elaboração da documentação técnica que compõe o Alicerce.

- **Paradigma de programação:**

- Programação Orientada a Objetos (POO)?
- Programação Funcional?
- Programação Procedural?

- **Padrões Arquiteturais:**

- Microsserviços?
- Event-driven architecture?
- Arquitetura de portas e adaptadores (hexagonal)?

- **Testes e Qualidade de Código:**

- Tipos de testes praticados:
  - \* Unitários,
  - \* Integração,
  - \* End-to-end, etc.
- Ferramentas e frameworks.
- Melhores práticas a serem reforçadas durante o processo de escrita e manutenção de testes.
- Estabelecer a necessidade ou não de cobertura de testes, assim como o valor de cobertura do código almejado.

- **Gerenciamento de Configuração de Software (SCM):**

- Sistema de controle de versão utilizado (Git; Subversion; Mercurial; etc).
- Fluxos de trabalho (Workflows) comuns: Git Flow, GitHub Flow, GitLab Flow.
- Práticas recomendadas para commits, branches, merges, resoluções de conflitos, etc.

- **Revisão de Código:**

- Processos de code review: Pull Requests, Merge Requests.
- Ferramenta de revisão de código.
- Critérios de qualidade a serem considerados:
  - \* Legibilidade,
  - \* Eficiência,
  - \* Segurança,
  - \* Padrões de codificação.

- **Tipos de repositórios utilizados:**

- Monorepo
- Polirepo

- **Práticas de DevOps:**

- Integração Contínua (CI)
- Entrega Contínua (CD)

### C. Banco de tarefas

Artefato final do Alicerce, o banco de tarefas tem como objetivo prover os insumos necessários à imersão do recém-chegado no domínio em que o time atua, assim como nas tecnologias utilizadas no projeto. Sua elaboração é feita através da separação de encargos, idealmente pré-refinados, que foram identificados com um nível de complexidade condizente com o perfil de senioridade do novo colaborador. De forma a facilitar a atribuição futura dessas atividades, seu nível de complexidade deve ser expresso de alguma maneira. Por simplicidade, recomenda-se uma prática comum, com o uso da sequência de Fibonacci (1,2,3,5,8,...). Tarefas candidatas incluem, mas não estão limitadas a:

- **Resolução de débitos técnicos através da refatoração:**

- Reduzir duplicidade de código: tratamento de código duplicado através da criação de métodos ou classes utilitárias que encapsulam a lógica repetida.
- Modularização e simplificação de métodos: dividir funções longas em partes menores, idealmente especializadas na resolução de uma única parte do problema.
- Redução de acoplamento: identificar e remover dependências desnecessárias entre classes e módulos utilizando, por exemplo, práticas como injeção de dependências ou até mesmo a substituição de instâncias diretas de classes por interfaces mais fáceis de serem substituídas sob demanda.

- **Revisão e melhoria da suíte de testes implementada:**

- Em projetos que fazem uso de testes, esses são um valioso mapa dos requisitos de negócio e problemas técnicos que o domínio da aplicação procura solucionar. Dessa forma, trabalhar na manutenção e criação de testes tende a oferecer uma visão abrangente das funcionalidades do sistema, auxiliando novos colaboradores a compreender o contexto em que estão sendo inseridos.

- **Contribuições para a documentação do time:**

- Fazer com que recém chegados sigam a documentação vigente e, caso necessário, proponham alterações, é uma ótima atividade a ser considerada. Ela não somente apresenta o novo membro aos processos disponíveis, como também auxilia na manutenção e melhoria da base de conhecimento do time.
- Em um caso de estudo realizado em 2010 na Google [16], por exemplo, foi constatado que novos engenheiros de software são encorajados a contribuir com entradas no glossário oficial da organização, cujo objetivo é auxiliar funcionários a entender jargões e contextos típicos da empresa, como também no “Manual do Desenvolvedor Google”, o qual contém explicações e links para recursos especializados, como guias de estilo.

- **Aprendizagem entre pares:**

- “Codelabs”: tutoriais práticos, ministrados por membros já estabelecidos no time, nos quais o aluno segue instruções passo a passo para a resolução de uma determinada tarefa comum ao contexto de atuação do time, utilizando para tal as tecnologias empregadas no projeto.
- “Pair Programming”: diferente de “Codelabs”, o “Pair programming” não é uma atividade pré-estruturada. Ela consiste no trabalho conjunto entre o recém-chegado e um desenvolvedor mais experiente na resolução de demandas. Isso não só ajuda no aprendizado de melhores práticas e padrões de codificação do projeto, mas também facilita a transferência de conhecimento tácito, promovendo maior integração na equipe.
- Revisões de código: a participação ativa nesse processo, tanto como autor quanto como revisor, auxilia na compreensão do código-base e cultura de codificação do time.

### D. Capítulo 3: Envolvimento dos Atores

Neste capítulo tanto se identifica quanto se descreve as diferentes figuras envolvidas no processo de Socialização Organizacional de desenvolvedores, assim como suas respectivas responsabilidades e encargos coletivos para com pessoas sendo submetidas ao processo de Onboarding.

1) *Anfitrião de Integração*: Sua função é garantir que o novo colaborador tenha uma recepção calorosa e informativa, promovendo uma primeira impressão positiva e facilitando a adaptação inicial. O escopo de suas ações deve ser tanto garantir quanto repassar a novos funcionários aquilo que precisam para iniciar seu trabalho no time.

#### **Responsabilidades:**

- **Boas-Vindas**: Realizar sessões de boas-vindas, seja 1:1 ou com demais membros da equipe, procurando repassar de forma resumida seu contexto, valores e objetivos.
- **Orientação Inicial**: Fornecer informações básicas sobre as políticas da empresa e/ou equipe, regras de segurança e procedimentos operacionais de acordo com os itens de Compliance vigentes.
- **Coordenação de Atividades**: Organizar e coordenar atividades de integração, como dinâmicas em grupo e eventos de team building.
- **Mediar Conflitos**: Atuar como mediador em potenciais conflitos que possam surgir durante o período de adaptação, ajudando a resolver desentendimentos e facilitando a comunicação entre o novo colaborador e a equipe.

2) *Mentor*: Pessoa experiente do time que irá acompanhar pessoalmente o progresso do novo colaborador, seja durante períodos específicos ou todo o processo de Onboarding. Idealmente, deve ser uma figura exemplar no que tange tanto ao seu comprometimento com os valores essenciais da equipe quanto ao seu conhecimento do domínio e tecnologia aplicados no time.

#### **Responsabilidades:**

- **Orientação Técnica:** Auxiliar no entendimento das ferramentas e tecnologias utilizadas pela equipe.
- **Acompanhamento Individualizado:** Oferecer suporte contínuo ao novo colaborador, procurando entender assim como solucionar dificuldades enfrentadas por recém-chegados.
- **Sugestão ou atribuição de incumbências:** Como a pessoa que irá acompanhar de perto o progresso do novo integrante, o mentor estará mais apto a auxiliar na escolha de tarefas a serem repassadas a esses indivíduos.
- **Feedback Regular:** Fornecer feedback construtivo e orientar sobre o desempenho do novo colaborador.
- **Facilitar a Integração do recém chegado nos círculos sociais da equipe:** Ser a ponte entre o novo membro e a equipe, facilitando a construção de relações interpessoais.

3) *Parceiro:* Todo membro do time pode ser considerado um parceiro, sendo que a função deste papel é prover suporte secundário à pessoa sendo submetida ao processo de Onboarding. Elas podem ou não ter responsabilidades semelhantes à de mentores, mas sua incumbência e tempo de dedicação direta para com recém-chegados é sempre limitado. Apesar de serem figuras de apoio, dando auxílio momentâneo quando necessário, elas têm papel importante em atividades que envolvem aprendizado entre pares.

#### **Responsabilidades:**

- **Apoio de curto prazo:** Fornecer suporte em períodos curtos e específicos, ajudando o novo colaborador a resolver dúvidas imediatas e problemas pontuais.
- **Treinamento específico:** Conduzir sessões de treinamento. Exemplo: workshops focados nas tecnologias e processos específicos do domínio em que o time atua.

4) *Gerenciamento de expectativas:* Independente do papel exercido por membros do time que auxiliam no processo de Onboarding, é importante que tenham em mente o seguinte dever, partilhado entre todos envolvidos no processo - Gerenciamento de expectativas. Inicialmente, cabe ao Mentor, junto à liderança do time, definir expectativas realistas nos seguintes pontos:

#### **• Escopo do trabalho:**

- Estabelecer que o nível de complexidade e impacto esperado irá aumentar de forma gradual e de acordo com a familiarização da pessoa com o sistema.
- Clarificar quais partes do projeto o novo colaborador estará mais envolvido inicialmente, como módulos específicos ou tipos de problemas a serem resolvidos.

#### **• Desempenho:**

- **Velocidade de Aprendizado:** Espera-se que novos colaboradores levem um tempo para entender o contexto, sistema e processos internos. O processo de aprendizado leva tempo e isso é normal e desejável.
- **Volume de Trabalho Inicial:** Nos primeiros meses, o foco estará mais na aprendizagem e menos na entrega de grandes volumes de trabalho.

- **Qualidade vs. Quantidade:** A qualidade do trabalho entregue é mais importante que a quantidade durante o período de Onboarding.

#### **• Comunicação:**

- Qual etiqueta e/ou boas práticas de comunicação devem ser respeitadas.
- Qual o nível de integração social desejado.
- Quais reuniões e dinâmicas de equipe são vistas como importantes, assim como grau esperado de participação nesses momentos.

#### **• Hierarquia:**

- Estrutura hierárquica a ser respeitada (caso exista), assim como o papel de cada figura envolvida.
- Quando problemas devem ser escalados e quais as linhas de comunicação propícias nesses casos.

Após esse momento inicial, cabe aos atores envolvidos no processo de Onboarding transmitir essas informações no intuito de se reduzir o estresse comum ao contexto de incerteza e adequação ao novo em que recém-chegados se encontram. Estudos [9][10] destacam que essa transparência é um fator relevante para evitar equívocos comuns como os descritos abaixo:

- “Devo fazer tudo sozinho para que meu gerente tenha uma boa impressão de mim.”
- “Devo ser a pessoa que resolve qualquer bug/demanda que encontrar. Além disso, eu devo começar a resolvê-lo imediatamente, mesmo que não tenha tempo para isso.”

Transparência quanto às expectativas ameniza incertezas e estresse à medida que auxilia na compreensão do que se é esperado. Dentre as diferentes formas de repasse de expectativas a serem praticadas, tem-se:

- Discussões abertas em momentos propícios (reuniões “one on one”).
- Revisões periódicas do progresso da pessoa sendo introduzida à organização.
- Documentação de planos de curto e longo prazo quanto a resultados estratégicos esperados do time.

Tais práticas geram transparência e comunicação clara de forma a estabelecer um ambiente propício à aprendizagem. O resultado é uma assimilação mais coesa, contribuindo, em última análise, para uma sensação de segurança psicológica e redução de incertezas.

#### *E. Capítulo 4: Execução*

Neste capítulo são expostas diretrizes e objetivos almejados pela socialização organizacional proposta no Launchpad. Sua execução está separada em três momentos, os quais têm relação direta com as “Dimensões do Onboarding” expostas no capítulo 1: Integração, Imersão no domínio e tecnologia, Cultura.

##### *1) Integração:*

- **Objetivos** - Retenção; valorização; acolhimento.
- **Principal Ator** - Anfitrião de integração.
- **Timing** - Executado no início do processo, com tempo de execução e passos definidos.

Fase inicial do processo, a integração também é a mais direta uma vez que envolve objetivos bem definidos e padronizáveis. Nela o anfitrião de integração deve repassar ao novo colaborador tudo aquilo que foi definido como necessário para que essa pessoa comece a produzir e se ajustar ao time. O principal artefato do Alicerce que auxilia nessa identificação é o Mapa de dependências. Dentre as três fases propostas ela deve ser a mais curta, recomendando-se não mais que uma semana para sua realização.

O resultado almejado é que, ao final deste momento, o novo integrante tenha ciência:

- De quem são seus colegas de trabalho - nome; contato; horas de trabalho; área de especialidade; etc.
- Quem será seu mentor e quais são os parceiros que prestarão suporte durante o processo de Onboarding.
- De quais os ritos ou reuniões ela deve participar, assim como seus dias de execução e horários.
- De canais de suporte disponíveis a elas, quais pessoas ela pode contatar em caso de problemas sejam pessoais ou no âmbito do trabalho.
- Dos itens de compliance relevantes ao time, como obrigações legais e contratuais, além dos recursos necessários à produção.
- De como realizar o setup de seu ambiente de trabalho.

Quanto a etapas desta fase, recomenda-se:

- 1) Boas-vindas onde o anfitrião apresenta tanto o time quanto seu contexto de operação, valores e objetivos.
- 2) Sessão de setup de ambiente onde, seja com auxílio de documentação do processo ou integrantes do time, o novo colaborador é auxiliado na tarefa de configurar seu ambiente de trabalho para produção. Isso inclui, mas não está limitado a configurações de:
  - e-mails e/ou notificações
  - SCM e repositórios de código
  - ferramentas de desenvolvimento (IDE, plugins, etc.)
  - bancos de dados e servidores
- 3) Introdução do novo colaborador à sua pessoa mentora, mediada pelo anfitrião.
- 4) Revisão dos itens de compliance, incluindo um overview das políticas de segurança, privacidade de dados e outras normas relevantes ao time e à empresa.
- 5) Introdução e/ou repasse dos processos e procedimentos da empresa aplicáveis ao contexto de trabalho do time. Por exemplo:
  - Metodologias empregadas (scrum; kanban; etc)
  - Políticas de comunicação interna
  - Fluxos de trabalho e aprovação de tarefas, desde a atribuição até a conclusão.

Paralelo à execução dessas etapas, cabe ao anfitrião organizar a integração do recém-chegado nos processos internos. Ele deve, por exemplo, incluir a recente admissão no calendário de ritos e reuniões; garantir/requisitar acessos a softwares e hardwares necessários; manter atualizado o mapa de dependências; etc.

## 2) *Imersão no domínio e tecnologia:*

- **Objetivos** - Eficácia; redução do tempo para produtividade; engajamento.
- **Principais atores** - Mentor e Parceiro.
- **Timing** - Executado após a fase de integração, com tempo de execução e passos maleáveis.

A imersão no domínio e tecnologia é o foco principal do processo de Onboarding proposto no Launchpad e, dessa forma, é aquele que irá demandar mais, tanto do ponto de vista de tempo quanto de dedicação dos atores envolvidos. Esta etapa está relacionada a processos que auxiliam na introdução, com rapidez e propósito, de novos colaboradores ao contexto do time e demandas que tipicamente irão realizar na equipe.

Um dos quesitos chaves para a organização dessa fase é a ênfase naquilo que é necessário para que um recém-contratado ou recém-chegado comece a trabalhar. O principal auxílio para se identificar esses conhecimentos vitais é o Alicerce, com destaque para a seção de Fundamentação. Uma equipe, uma vez munida dessas informações, estará apta a planejar uma etapa de imersão personalizada aos desafios técnicos e de negócio específicos de seu contexto.

A maior unidade de tempo utilizada no Launchpad são semanas e encoraja-se que atividades sejam planejadas com isso em mente. Dependendo da preferência do time, tarefas de imersão podem ser planejadas com antecedência pelo mentor e/ou parceiros, ou até mesmo de semana em semana de acordo com o progresso da pessoa sendo integrada. Caso tenha elaborado um banco de tarefas, procure fazer uso dos números de complexidade associados a cada atividade, estabelecendo um número máximo a ser contemplado ao longo de sete dias.

Outro ponto a se ter em mente ao planejar atividades é a necessidade de variação. É tentador realizar, principalmente nos períodos iniciais de Onboarding, a exposição exaustiva de informações acerca do projeto, desde seus padrões técnicos a regras de negócio. No entanto, abordagens passo a passo tendem a ser mais efetivas do ponto de vista de aprendizado e motivação, principalmente quando se intercala teoria e prática. Para facilitar o ato de planejar atividades variadas, o Launchpad sugere o uso de um banco de tarefas, cujas atividades podem ser separadas nas seguintes categorias.

- **Incumbências:** Primeira categoria de atividades a serem realizadas na fase de imersão na tecnologia e domínio, as incumbências são tarefas que devem, ao final de sua realização, gerar algum tipo de resultado tangível - documentação; funcionalidade; refinamento; etc. Incumbências têm caráter variado, podendo incluir atividades teóricas como a leitura de documentos técnicos e suíte de testes, além de práticas como a correção de bugs mapeados e refatoração de sessões do código.
- **Aprendizagem entre pares:** Segunda categoria de atividades, essas tarefas focam na colaboração e aprendizado direto com o mentor e parceiros. O intuito é promover uma troca de conhecimento constante entre os envolvidos, além de introduzir o novo colaborador às pessoas com quem irá trabalhar. O processo de Onboarding é uma

atividade necessariamente humana e, para que seja bem sucedida, necessita de interação entre atores e recrutas. Assim, recomenda-se que esses indivíduos sejam encorajados a levantar perguntas sempre que necessário. O diálogo deve acontecer de forma aberta seja em canais de discussão internos, ritos, treinamentos ou reuniões 1:1. Idealmente, ao final do processo de Onboarding o colaborador deve ter interagido e aprendido com todos os membros da equipe.

- **Entrega de ponta a ponta:** Terceira categoria de atividades, entregas ponta a ponta são, necessariamente, tarefas práticas. Elas envolvem a resolução de demandas do time do início até a liberação em produção, incluindo ou não o suporte de parceiros e mentor. Dessa forma, variáveis relacionadas a entregas ponta a ponta são um indicativo relevante em relação ao nível de imersão na tecnologia e domínio de uma pessoa sendo submetida ao processo de Onboarding. Dentre essas variáveis temos:
  - Nível de suporte requisitado de pares
  - Nível de complexidade da tarefa entregue
  - Velocidade de entrega
  - Métricas de qualidade relacionadas a essa entrega
    - \* Cobertura de testes
    - \* Regressões causadas pela entrega (caso ocorram)
    - \* Feedback do cliente ou usuário final
- **Atividades paralelas:** Quarta e última categoria, atividades paralelas têm triplo intuito - a prestação de apoio, acompanhamento do progresso e promoção de segurança psicológica ao recém-chegado. O principal exemplo são reuniões 1:1 entre mentor e orientando, onde deve ser praticado o feedback contínuo, construtivo e propositivo. Cabe ao mentor identificar pontos fortes do orientando e sugerir melhorias onde necessário. O mentor também pode ajudar o orientando a identificar áreas de desenvolvimento dentro do projeto e seus recursos, propondo, por exemplo, treinamentos, recomendação de cursos, livros, ou a participação em projetos específicos que ofereçam aprendizado relevante.

Como atividades paralelas estão relacionadas a acompanhamento, idealmente devem acontecer de forma periódica (ex: uma vez a cada semana ou a cada duas semanas) e interferir o menos possível com as demais atividades a serem realizadas pelo orientando. Ademais, por simplicidade, pode-se definir que um planejamento de tarefas está variado caso inclua, a cada semana, ao menos duas categorias distintas entre as supracitadas.

Nas fases iniciais do Onboarding, espera-se que a categoria de atividades realizadas pelo novo colaborador esteja focada na aprendizagem entre pares, com entregas esporádicas de incumbências. Com o acréscimo de conforto, confiança e conhecimento do projeto e suas peculiaridades, é comum que atividades de aprendizagem entre pares ocorram com menor frequência e incumbências sejam entregues com agilidade. Assim, abre-se caminho para tarefas mais envolvidas como entregas ponta a ponta, as quais tendem a demandar mais

do colaborador uma vez que necessitam de um conhecimento holístico das tecnologias empregadas e do domínio específico à aplicação do time. Ao final do processo, é esperado que a pessoa esteja integrada e capaz de realizar, com consistência e menos suporte, entregas dessa natureza.

### 3) *Cultura:*

- **Objetivos** - Coesão; propósito.
- **Principais atores** - Todo o time.
- **Timing** - Executada durante todo o processo, sendo uma dimensão transversal às demais.

Como exposto no capítulo 2, cultura é uma dimensão ampla expressada principalmente pelos valores ou crenças essenciais obtidas a partir da experiência conjunta do grupo. O Onboarding deve ser utilizado como oportunidade tanto para manter quanto para transmitir esses valores de uma forma mais tangível, através de exemplos, histórias, processos e interações com os membros da equipe. Dessa forma, cultura é uma fase transversal, devendo ser exercida continuamente nas fases de integração e imersão no domínio e tecnologia.

Para times especializados na manutenção e/ou desenvolvimento de software, pode-se resumir tais valores como atividades, processos e padrões comportamentais e técnicos (design; arquitetura; testes) verificados como importantes para o alcance dos objetivos da equipe. Cabe ao time como um todo a identificação dessas crenças. Apesar da Fundamentação do Alicerce ser um ponto de partida para essa tarefa de identificação, ela é, necessariamente, uma construção coletiva e introspectiva. Desse modo, caso os valores essenciais de um time não estejam mapeados, recomenda-se a realização, com regularidade, de ritos de auto-avaliação com a inclusão de novos colaboradores. A título de exemplo:

- **Retrospectiva:** Reunião geralmente realizada ao final de ciclos de entrega, onde a equipe reflete sobre o que funcionou bem, o que não funcionou, e o que pode ser melhorado. Ao se discutir abertamente esses tópicos o time pode identificar padrões comportamentais e técnicos que contribuíram para o sucesso, mapeando-os como valores essenciais.
- **Reuniões de revisão de técnica** são momentos em que o time discute e avalia as decisões arquiteturais e tecnológicas do sistema. Elas podem envolver a análise de diagramas de arquitetura, revisão de princípios arquiteturais e discussão de novas propostas ou até mesmo a revisão conjunta do código implementado por colegas.

A partir do momento em que as crenças essenciais do time são registradas, faz-se necessário a criação de mecanismos para assegurar sua adesão e manutenção por novos membros da equipe. Tais mecanismos incluem mas não estão limitados a:

- **Diretrizes:** Documentos que estabelecem claramente as expectativas, padrões e melhores práticas a serem seguidas pela equipe. Estas diretrizes devem ser facilmente acessíveis e regularmente atualizadas para refletir as mudanças no projeto, tecnologias utilizadas e stakeholders. Diretrizes podem tomar o formato de, por

exemplo, um treinamento que promova comunicação respeitosa e inclusiva ou até mesmo um manual de estilo de código que define padrões de codificação, práticas de revisão de código, e melhores práticas a serem seguidas na hora de se criar e manter testes.

- **Automações:** Ferramentas e scripts que garantem que os padrões e práticas da equipe sejam seguidos automaticamente. Isso pode incluir práticas de DevOps, testes automatizados e verificações de conformidade e qualidade do código. A título de exemplo, cita-se pipeline de CI/CD configurado para executar testes automatizados e análise de qualidade de código a cada merge, garantindo que apenas código que atenda aos padrões de qualidade seja integrado ao projeto.

Os atores envolvidos no processo de Onboarding devem ser os principais proponentes da cultura inerente ao time, expressando-as e reforçando-as aos novos colaboradores durante seu acompanhamento, seja no contexto individual ou coletivo. É desejado que, ao final do processo, a pessoa recém integrada tenha ciência e pratique os valores essenciais do time, sabendo inclusive os mecanismos implementados para garantir sua aplicação e preservação.

#### *F. Capítulo 5: Iteração e Melhoria Contínua*

O Launchpad foi produzido tendo em mente que instituições que praticam o Onboarding Institucionalizado tendem a obter maior sucesso [13]. Essa categoria de socialização organizacional são programas estruturados onde recém-chegados recebem orientação e mentoria formal do grupo em que estão sendo inseridos. Como todo processo, o Onboarding proposto neste guia deve ser avaliado e ajustado periodicamente para garantir sua efetividade. A avaliação de sua eficácia pode ser feita através de diferentes métricas, tanto qualitativas como quantitativas.

1) *Métricas qualitativas:* São aquelas coletadas de forma semi-estruturada, envolvendo um maior grau de subjetividade entre os envolvidos no processo de socialização organizacional. São obtidas a partir do acompanhamento feito por mentores a seus orientandos, feedbacks do time e recém-chegado, além de documentos escritos como formulários. A partir desses artefatos, o time pode extrair informações que são indicativas de um processo de Onboarding bem sucedido [5]:

- **Auto-eficácia:** Grau com que novos colaboradores se sentem confiantes e confortáveis em exercer suas funções.
- **Clareza de função:** Representa quão bem os novos funcionários entendem sua função e expectativas relacionadas ao seu trabalho.
- **Integração social:** Representa até que ponto os novos funcionários se sentem socialmente confortáveis e aceitos pelos seus colegas de trabalho.
- **Conhecimento e prática da cultura:** Está relacionada ao entendimento da cultura exclusiva ao time - suas políticas, objetivos estratégicos, crenças fundamentais e valor do serviço ou produtos entregues às partes interessadas.

Essas métricas estão relacionadas a fatores como satisfação e motivação, grau de comprometimento com o time e seus objetivos, potencialização do desempenho individual do novo funcionário, assim como menores taxas de rotatividade.

2) *Métricas quantitativas:* São aquelas coletadas de forma estruturada, seja periodicamente com a participação de input humano ou automaticamente por ferramentas, como resultado do trabalho feito pela pessoa sendo submetida ao processo de Onboarding. Possíveis métricas desta categoria irão variar em importância devido à grande diferença de demandas e anseios de cada time. Recomenda-se que as métricas quantitativas estejam ligadas, principalmente, à entregas ponta a ponta, já que essa categoria de atividades é indicativa do conhecimento do novo colaborador acerca do domínio, processos e tecnologias intrínsecas ao projeto. Dentre os possíveis dados a serem coletados, destaca-se:

- **Métricas baseadas em commit:** Podem ser usadas para avaliar o nível de proficiência alcançado pelo colaborador e para medir o grau de conhecimento baseado na autoria e no interesse em partes específicas do projeto. Exemplos incluem:
  - Número e/ou frequência de commits.
  - Natureza dos commits (por exemplo: resolução de bugs, refatoração, features, automações).
- **Volume de entrega:** a taxa na qual os novos desenvolvedores fazem contribuições para o projeto ao longo do período de Onboarding. A mediação envolve o rastreamento da frequência e do volume de contribuições, as quais podem ser auferidas a partir de:
  - commits
  - comentários em revisão de código
  - criação ou manutenção de documentação do projeto
- **Nível de complexidade atrelado à tarefa:** como exposto anteriormente, é desejável que atividades, sejam elas mapeadas no banco de tarefas ou não, tenham uma numeração que indique seu nível de complexidade.

Pesos podem ser associados a cada tipo de dado, de forma a diferenciar sua relevância para o projeto. Um time de desenvolvimento focado em inovação e entrega rápida de novas funcionalidades pode dar mais ênfase à velocidade da entrega e à complexidade técnica das tarefas realizadas. Em contrapartida, um time dedicado à manutenção de sistemas críticos pode focar mais na qualidade dos commits, regressões causadas pela entrega, e participação em revisões de código.

3) *Avaliação e Reiteração:* Reiterar com base em dados, avaliações e feedback é fundamental para garantir que o Onboarding continue eficaz e alinhado com os objetivos do time. Este é o momento de observar o progresso final do novo membro, identificar possíveis dificuldades e coletar insights a partir das métricas disponíveis, no intuito de aprimorar o processo. Após avaliação chega a etapa crucial de transformar insights em ações a serem aplicadas no próximo ciclo:

- 1) Atualize o plano de onboarding com base nas informações obtidas. Ajuste itens como o conteúdo, a duração das atividades, as ferramentas e a forma

como o conhecimento é compartilhado. Tente tornar seu Onboarding o mais especializado possível para o contexto do time e, novamente, seguindo o princípio de **ênfatar aquilo que é necessário**.

- 2) Defina ações específicas para resolver as dificuldades e desafios encontrados. Se possível, procure separar as ações de acordo com as dimensões do Onboarding e suas respectivas fases de execução. Isso auxilia em definir escopo e atores responsáveis em executar as ações levantadas.

Exemplos práticos incluem:

- **Insight A:**

- A auto-eficácia neste ciclo não foi satisfatória. Novos colaboradores têm dificuldades de aprender e utilizar as tecnologias empregadas no projeto.

- **Ações:**

- **Imersão no Domínio e Tecnologia:**

- \* Definir um plano de treinamento específico para essa tecnologia, com incumbências periódicas a serem entregues ao longo do Onboarding.
- \* Promover novas atividades de aprendizagem entre pares ministradas por mentores ou parceiros, como "Codelabs" e "Workshops".
- \* Investigar e sugerir recursos internos ou externos à equipe para colaboradores que serão integrados no próximo ciclo.

- **Insight B:**

- O conhecimento e prática da cultura foi insuficiente. Novos membros estão tendo dificuldades em se integrar ao time, gerando menos adesão e aplicação dos valores essenciais inerentes à equipe.

- **Ações:**

- **Integração:**

- \* Promover eventos de "team-building" com mais frequência, expressando a importância de membros atuais do time em interagir com recém-chegados.

- **Cultura:**

- \* Criar canais de comunicação onde haverá interação aberta e sem protocolos entre novos e atuais membros do time.
- \* Incluir na avaliação de performance de mentores o desempenho positivo de seus orientandos como diferencial, de forma a incentivar um acompanhamento mais próximo.

- **Insight C:**

- Neste ciclo, novos colaboradores fizeram um número menor de entregas ponta a ponta. O tempo até a realização do primeiro commit aumentou devido ao moroso processo de setup do ambiente e liberação de acessos. Além disso, verificou-se um aumento no nível de complexidade das tarefas entregues.

- **Ações:**

- **Integração:**

- \* Tornar automatizado o processo de conceder permissões e acessos.
- \* Implementar ou aprimorar documentação base para setup do ambiente.

- **Imersão no Domínio e Tecnologia:**

- \* Revisar o banco de tarefas, garantindo que atividades complexas sejam quebradas em partes menores sempre que possível.

#### IV. CONCLUSÃO

O Launchpad foi desenvolvido como um guia. Seu objetivo é definir diretrizes que ajudem na melhoria ou organização de um processo de Onboarding, visando auxiliar equipes na retenção de talento humano. O trabalho de revisão da literatura científica que fundamentou este documento encontrou amplas evidências que processos de socialização organizacional bem estruturados e personalizados estão associados a níveis mais altos de satisfação, produtividade, engajamento e comprometimento dos funcionários, além de uma redução na taxa de rotatividade de empregados, especialmente a longo prazo.

Reitera-se a hipótese inicial de que **primeiras impressões importam**. A chave para o sucesso do processo de Onboarding proposto reside na sua adaptação e personalização contínua para atender às necessidades específicas de cada time e projeto. Combinando documentação focada, envolvimento dos atores com separação clara de responsabilidades, aliada à compreensão do que configura Onboarding, é possível refinar e aprimorar cada etapa da socialização organizacional, garantindo que ela continue a ser uma ferramenta valiosa para equipes focadas no desenvolvimento e manutenção de Sistemas de Informação.

Quanto a trabalhos futuros, foi identificado que a maior parte das contribuições da literatura científica que lidam com a socialização organizacional, como esta, são empíricas. Dessa forma, acredita-se que há espaço para mais trabalhos teóricos com comprovação de dados, como o modelo de Modelo de Gaurav G. Sharma & Klaas-Jan Stol [3]. Essa afirmação leva em consideração a presença de métricas mensuráveis comuns ao desenvolvimento e manutenção de software, como velocidade de contribuição e indicadores baseadas em commit [14]. Tendo em vista a prevalência de ferramentas de versionamento como o Git em projetos e empresas [15], é plausível a implementação e exploração da viabilidade dessas e outras métricas quantitativas em modelos teóricos aplicados ao Onboarding corporativo.

#### REFERENCES

- [1] Steven G Westlund and John C Hannon. Retaining talent: Assessing job satisfaction facets most significantly related to software developer turnover intentions. *Journal of Information Technology Management*, 19(4):1–15, 2008.
- [2] Julian M Bass, Sarah Beecham, Mohammed Abdur Razzak, and John Noll. Employee retention and turnover in global software development: Comparing in-house offshoring and offshore outsourcing. In *Proceedings of the 13th International Conference on Global Software Engineering*, pages 82–91, 2018.

- [3] Gaurav G Sharma and Klaas-Jan Stol. Exploring onboarding success, organizational fit, and turnover intention of software professionals. *Journal of Systems and Software*, 159:110442, 2020.
- [4] Tarek K. Abdel-Hamid. A study of staff turnover, acquisition, and assimilation and their impact on software development cost and schedule. *Journal of Management Information Systems*, 6(1):21–40, 1989.
- [5] Talya N Bauer. Maximizing success. *SHRM Foundation's Effective practice guidelines series*, 2010.
- [6] John Eastin Van Maanen and Edgar Henry Schein. Toward a theory of organizational socialization. 1977.
- [7] Talya N Bauer and Berrin Erdogan. Organizational socialization: The effective onboarding of new employees. 2011.
- [8] Dave Farley and Jez Humble. *Continuous delivery: a handbook for building, deploying, testing and releasing software*. Addison-Wesley Professional, 2010.
- [9] Jezreel Mejía, Isaac Rodríguez-Maldonado, Héctor Girón-Bobadilla, and Mirna Muñoz. Knowledge management in software process improvement: A systematic literature review. In *2019 14th Iberian Conference on Information Systems and Technologies (CISTI)*, pages 1–7. IEEE, 2019.
- [10] Fernando C Prestes Motta and Isabella F Gouveia VASCONCELOS. A cultura organizacional. *MOTTA, Fernando C. Prestes. Teoria geral da administração*, 3(1):1–22, 2002.
- [11] Howard J Klein, Jinyan Fan, and Kristopher J Preacher. The effects of early socialization experiences on content mastery and outcomes: A mediational approach. *Journal of Vocational Behavior*, 68(1):96–115, 2006.
- [12] M Ronald Buckley, Donald B Fedor, John G Veres, Danielle S Wiese, and Shawn M Carraher. Investigating newcomer expectations and job-related outcomes. *Journal of Applied Psychology*, 83(3):452, 1998.
- [13] Talya N Bauer, Todd Bodner, Berrin Erdogan, Donald M Truxillo, and Jennifer S Tucker. Newcomer adjustment during organizational socialization: a meta-analytic review of antecedents, outcomes, and methods. *Journal of applied psychology*, 92(3):707, 2007.
- [14] Fabian Fagerholm, Alejandro S. Guinea, Jürgen Münch, and Jay Borenstein. The role of mentoring and project characteristics for onboarding in open source software projects. In *Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, ESEM '14*, New York, NY, USA, 2014. Association for Computing Machinery.
- [15] Casey Casalnuovo, Bogdan Vasilescu, Premkumar Devanbu, and Vladimir Filkov. Developer onboarding in github: The role of prior social links and language experience. In *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering, ESEC/FSE 2015*, page 817–828, New York, NY, USA, 2015. Association for Computing Machinery.
- [16] Maggie Johnson and Max Senges. Learning to be a programmer in a complex organization: A case study on practice-based learning during the onboarding process at google. *Journal of Workplace Learning*, 22(3):180–194, 2010.
- [17] Shreya Kumar, Charles Wallace, and Michael Young. Mentoring trajectories in an evolving agile workplace. In *Proceedings of the 38th International Conference on Software Engineering Companion*, pages 142–151, 2016.
- [18] Leandro Marques Venceslau de Souza. [Launchpad: Guia de Onboarding para Desenvolvedores](#). 2024.
- [19] Edgar H Schein. *Organizational culture.*, volume 45. American Psychological Association, 1990.
- [20] Marco Tulio Valente. Engenharia de software moderna. *Princípios e Práticas para Desenvolvimento de Software com Produtividade*, 1(24), 2020.