

Projeto de *Software* com Métodos Ágeis



Cruzeiro do Sul Virtual
Educação a distância

Material Teórico



Projeto – Parte B

Responsável pelo Conteúdo:

Prof. Me. Artur Marques

Revisão Textual:

Prof.^a Esp. Kelciane da Rocha Campos



- Planejamento e Priorização de Histórias do Usuário;
- Jogo do Planejamento;
- Análise de Requisitos Funcionais, Não Funcionais e Regras de Negócios.



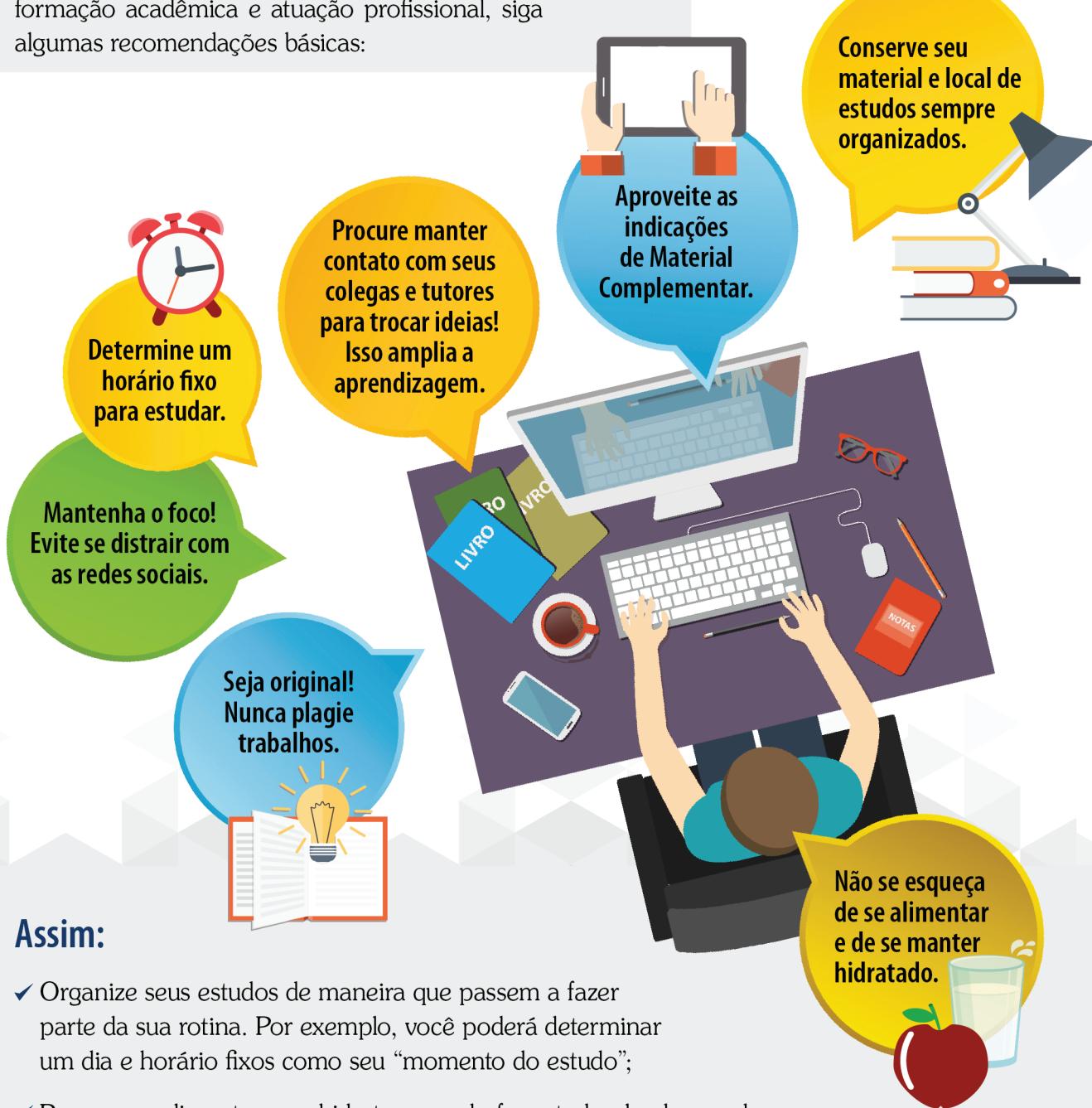
OBJETIVO DE APRENDIZADO

- Planejar as atividades do projeto ágil criando artefatos estruturais.



Orientações de estudo

Para que o conteúdo desta Disciplina seja bem aproveitado e haja maior aplicabilidade na sua formação acadêmica e atuação profissional, siga algumas recomendações básicas:



Assim:

- ✓ Organize seus estudos de maneira que passem a fazer parte da sua rotina. Por exemplo, você poderá determinar um dia e horário fixos como seu “momento do estudo”;
- ✓ Procure se alimentar e se hidratar quando for estudar; lembre-se de que uma alimentação saudável pode proporcionar melhor aproveitamento do estudo;
- ✓ No material de cada Unidade, há leituras indicadas e, entre elas, artigos científicos, livros, vídeos e sites para aprofundar os conhecimentos adquiridos ao longo da Unidade. Além disso, você também encontrará sugestões de conteúdo extra no item **Material Complementar**, que ampliarão sua interpretação e auxiliarão no pleno entendimento dos temas abordados;
- ✓ Após o contato com o conteúdo proposto, participe dos debates mediados em fóruns de discussão, pois irão auxiliar a verificar o quanto você absorveu de conhecimento, além de propiciar o contato com seus colegas e tutores, o que se apresenta como rico espaço de troca de ideias e de aprendizagem.

Planejamento e Priorização de Histórias do Usuário

Normalmente, começamos muitos projetos com um workshop de descoberta. Trata-se de uma oportunidade para que o time se reúna com as partes interessadas do cliente, a fim de responder a três perguntas:

- qual é o problema que estamos tentando resolver?
- para quem estamos resolvendo esse problema?
- como vamos resolver o problema?

Quadro 1

Falamos sobre	<ul style="list-style-type: none"> • objetivos de negócios; • objetivos do projeto; • restrições e riscos potenciais; • critérios de sucesso.
Para quem estamos resolvendo o problema	<ul style="list-style-type: none"> • definir funções de usuário para o aplicativo; • discutir os objetivos do usuário; • identificar os pontos problemáticos do usuário.

Para identificar como vamos resolver o problema, mapeamos os fluxos e tarefas do usuário em uma atividade chamada mapeamento da história do usuário.

O mapeamento é uma técnica de visualização popularizada por Jeff Patton que permite que as equipes de produto mapeiem um aplicativo inteiro em relação às diferentes funções do usuário que o aplicativo deve suportar.

Para Coulter, Fouque e Riethof (2017), a atividade começa identificando ações do usuário de nível superior (ou resultados do usuário), escrevendo-as em notas adesivas e organizando-as em uma linha no topo do mapa da história do usuário. Referimo-nos a essa linha de nível superior como o fluxo narrativo ou a espinha dorsal do mapa da história do usuário.

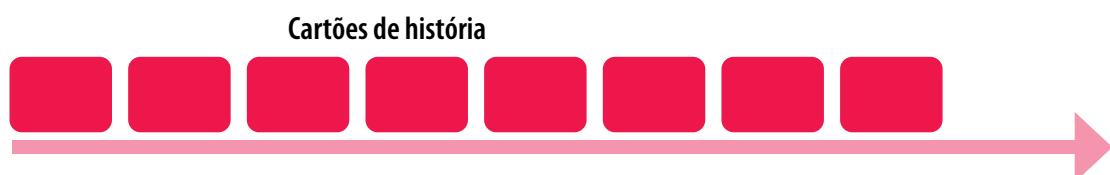


Figura 1 – Fluxo de narrativa

Se você imaginar a construção de um aplicativo como o do nosso exemplo, o fluxo narrativo pode incluir resultados do usuário, como:

- gerenciar minha conta;
- gerenciar minha lista de compras;
- compartilhar indicação de compra para outros clientes.

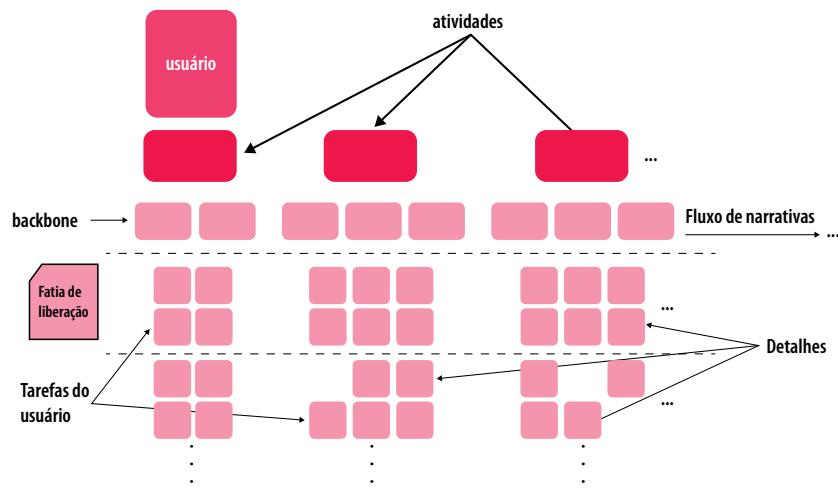


Figura 2 – Organização do mapa de histórias do usuário e seus componentes

Mas é tudo uma questão de aprofundamento, porque depois que as tarefas de alto nível foram identificadas e representadas no fluxo narrativo, passamos a identificar tarefas detalhadas, subtarefas e formas alternativas de realizar uma tarefa. Para distinguir tarefas detalhadas do fluxo narrativo no mapa da história do usuário, nós as escrevemos em notas adesivas (*post-it*, por exemplo) de cores diferentes e as adicionamos ao mapa da história do usuário nas tarefas relevantes de alto nível.

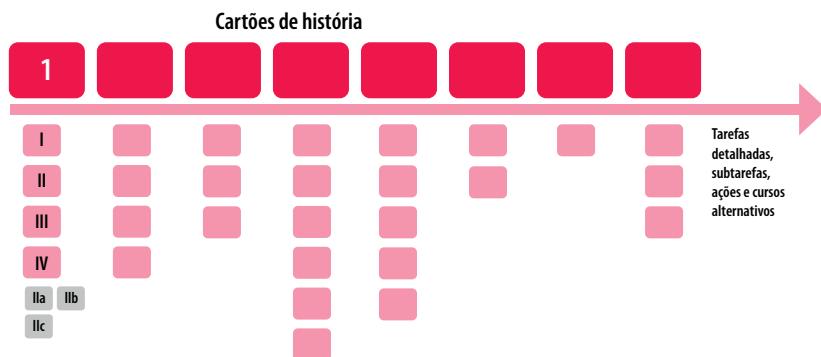


Figura 3 – Aumento de nível de detalhamento

Vamos ampliar o nosso projeto levando em consideração o cartão geral 1, que no nosso caso foi Gerenciar minha conta.

Podemos criar tarefas detalhadas para executarmos, por exemplo:

- I. logar primeira vez;
- II. fazer o cadastro;
- III. editar dados da conta;
- IV. apagar a conta.

Podemos também criar subtarefas:

- IIa) recuperar senha;
- IIb) mudar senha;
- IIc) editar avatar de usuário.

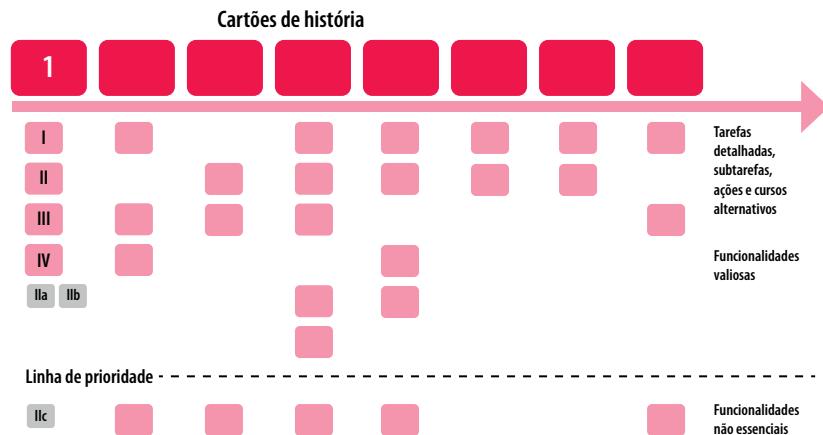


Figura 4 – Atividade de mapeamento da história do usuário

O mapa da história do usuário priorizado também se torna a primeira iteração do *backlog* do projeto. Talvez você não se recorde muito bem do que é *backlog*, então para não restar mais dúvidas:

Um *product backlog* é uma lista de novos recursos, mudanças em recursos existentes, correções de *bugs*, mudanças de infraestrutura ou outras atividades que uma equipe pode entregar para atingir um resultado específico.

O *backlog* do produto é a única fonte autorizada para coisas nas quais uma equipe trabalha. Isso significa que nada é feito que não esteja na lista de pendências do produto. Por outro lado, a presença de um item da carteira de produtos não garante que ele será entregue. Representa uma opção que a equipe tem para entregar um resultado específico, em vez de um compromisso.

Deve ser barato e rápido adicionar um item do *backlog* do produto ao *backlog* do produto, e deve ser igualmente fácil remover um item do *backlog* do produto que não resulta em progresso direto para alcançar o resultado desejado ou permitir o progresso em direção ao resultado.

Os itens do *backlog* do produto assumem uma variedade de formatos, sendo as histórias de usuários o mais comum. A equipe que usa o *backlog* do produto determina o formato que escolheu para usar e olha para os itens do *backlog* como lembretes dos aspectos de uma solução em que eles podem trabalhar.

Os itens do *backlog* do produto variam em tamanho e extensão de detalhes com base em grande parte na rapidez com que uma equipe trabalhará neles. Aqueles nos quais uma equipe trabalhará em breve devem ser pequenos em tamanho e conter detalhes suficientes para que a equipe comece a trabalhar. A equipe pode estabelecer uma definição de pronto para indicar sua concordância sobre as informações que gostaria de ter disponíveis para começar a trabalhar em um item do *backlog* do produto. Os itens do *backlog* do produto que não estão programados para funcionar podem ser bastante amplos e ter poucos detalhes.

A sequência de itens do *backlog* do produto em um *backlog* do produto muda à medida que uma equipe ganha um melhor entendimento do resultado e da solução identificada. Essa reordenação dos itens existentes da carteira de produtos, a adição e remoção contínuas de itens da carteira de produtos e o refinamento contínuo dos itens da carteira de produtos conferem à carteira de produtos sua característica dinâmica.

Uma equipe possui seu *backlog* do produto e pode ter uma função específica – *product owner* – com a responsabilidade primária de manter o *backlog* do produto. As principais atividades na manutenção do *backlog* do produto incluem priorizar os itens do *backlog* do produto, decidir quais itens do *backlog* do produto devem ser removidos do *backlog* do produto e facilitar o refinamento do *backlog* do produto.

Uma lista de pendências do produto pode ser uma maneira eficaz de uma equipe comunicar no que está trabalhando e no que planeja trabalhar em seguida. Mapas de histórias e radiadores de informações podem fornecer uma imagem clara de sua lista de pendências para a equipe e as partes interessadas.

O *backlog* do produto pode ser representado em formato físico usando cartões de índice ou notas adesivas, ou pode ser representado em formato eletrônico, como um arquivo de texto, planilha ou uma das muitas ferramentas de gerenciamento de *backlog* que existem. As placas eletrônicas são a melhor opção para uma equipe que tem membros remotos ou coleta muitas informações complementares sobre itens de *backlog* do produto. As placas físicas oferecem a vantagem de tornar o *backlog* do produto continuamente visível e concreto durante as discussões sobre o *backlog* do produto. (AGILE ALLIANCE, 2020, p. 1-2)

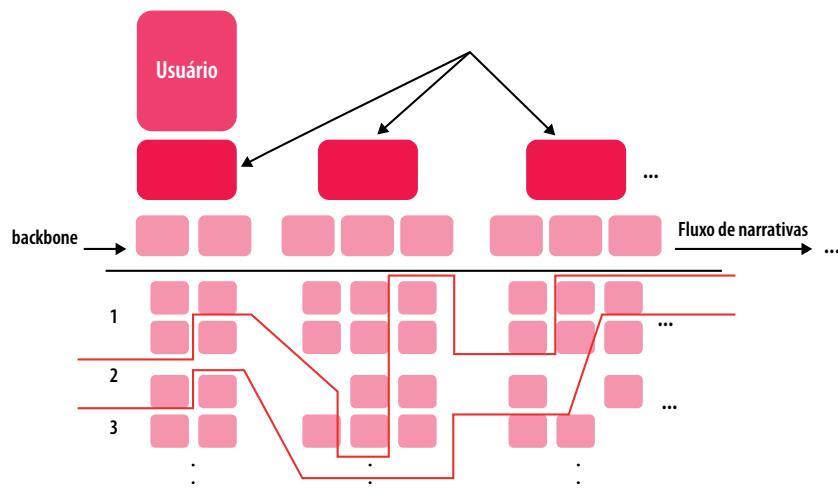


Figura 5 – Fatiamento de histórias por importância e relevância para gerar valor ao negócio

Após o *workshop* de descoberta, traduzimos cada nota adesiva do mapa em uma história de usuário devidamente estruturada e atribuímos critérios de aceitação a cada uma delas. Usamos para tal o nosso cartão padronizado de histórias do usuário de outra unidade da disciplina e você deve se recordar do padrão: como um [tipo de usuário], eu quero [recurso] para que [benefício].

Cada fatia do mapa da história do usuário fornece uma meta e uma lista de tarefas para chegar lá. Mas com a maioria dos produtos, você não será capaz de construir tudo listado em uma fatia em um único *sprint*.

Por fim, o *backlog* do produto e, por extensão, o plano de liberação, evoluem constantemente conforme o projeto avança: as prioridades mudam, o escopo é adicionado ou reduzido conforme o *feedback* é coletado, as histórias são divididas em partes menores etc. Contanto que novos itens do *backlog* sejam estimados e priorizados, o proprietário do produto pode ajustar o plano de lançamento para manter um prazo de lançamento realista.

Jogo do Planejamento

O *Planning Poker* é uma técnica de estimativa ágil que se tornou muito popular nos últimos anos. É baseado em uma técnica de estimativa conhecida como *Wideband Delphi*, que foi criada pela *RAND Corporation* na década de 1968.

A técnica foi aprimorada por James Grenning em 2002. Tornou-se muito mais popular quando foi incluída no livro *Agile Estimating and Planning*, de Mike Cohn. Embora os fundamentos da técnica já existam há muitos anos, os refinamentos de Grenning a tornaram utilizável por equipes ágeis e elas tiraram o máximo proveito.

Há muita mitologia envolvendo o jogo do planejamento, a primeira delas é o chamado *story point*.

Trata-se de um número que reflete a complexidade ou a quantidade de trabalho envolvida no desenvolvimento de uma história de usuário. Por exemplo, uma equipe pode atribuir 1 ponto a uma história de usuário simples (em nosso projeto pode ser, por exemplo: Fazer Login), 2-3 pontos para uma história moderadamente complexa (por exemplo, desenvolver o Carrinho de Compras em nosso projeto) e 4-5 pontos para uma grande história (por exemplo, a funcionalidade de Logística Reversa em caso de devolução de mercadorias), com base na compreensão do trabalho envolvido. Então, diz respeito também a quanto tempo uma história de usuário deve levar para se desenvolver, assumindo zero interrupções.

Além disso, no início de um projeto, há muitos itens pendentes para estimar para que o *Planning Poker* seja eficaz.

Nesses casos, a avaliação de massa relativa funciona bem.

Nunca ouviu falar?! Sem problemas.

Usando essa técnica, a equipe *scrum* primeiro organiza as histórias de usuário em ordem de seu tamanho relativo, do pequeno ao grande nível de esforço e, em seguida, atribui pontos de história a cada um usando uma sequência de Fibonacci modificada (aqui entra o *Planning Poker*).

O resultado é uma lista de pendências inicial totalmente estimada do produto, que permitirá ao proprietário do produto criar um plano de lançamento.

O pôquer de planejamento ou *planning poker*, como é conhecido, é um jogo de estimativa usado por equipes ágeis.

Vários membros da equipe são solicitados a estimar uma história de usuário desenhando uma carta de jogo com uma série de pontos de história e colocando-a voltada para baixo na mesa.

Em seguida, os cartões são virados para cima e se houver discrepâncias – por exemplo, um membro da equipe estimou 1 ponto e outros estimaram 4 ou 5, eles podem discutir e chegar a um consenso.

Vamos fazer isso passo a passo?

Creio que seja o momento ideal de você não ter mais dúvidas a este respeito porque, afinal, estamos desenvolvendo a parte escritural de um projeto.

Estas são as seguintes regras básicas:

1. Cada participante recebe um baralho de cartas de estimativa representando uma sequência de números. As sequências mais populares envolvem a duplicação de cada cartão (0, $\frac{1}{2}$, 1, 2, 4, 8, 16, etc.) ou a sequência de Fibonacci (1, 2, 3, 5, 8, 13, 21, etc.). A sequência de Fibonacci é geralmente mais popular.
2. O moderador apresenta uma história de usuário por vez para a equipe. São as histórias daquela *sprint* ou no máximo 2 *sprints*.
3. O *Product Owner* responde a quaisquer perguntas que a equipe possa ter sobre a história.
4. Cada participante seleciona em particular um cartão que representa sua estimativa do “tamanho” da história do usuário. Normalmente, o tamanho representa um valor que leva em consideração o tempo, risco, complexidade e quaisquer outros fatores relevantes. Claro que todos os participantes desenvolvedores devem ter ciência de desenvolvimento, o que é simples e o que é complexo, não se trata de um jogo de adivinhação e nem chute.
5. Quando todos estão prontos com um “orçamento” (sim, você está estimando, orçando quando lança uma carta), todos os cartões são apresentados simultaneamente.
6. Se houver consenso sobre um determinado número, o tamanho é registrado e a equipe avança para a próxima história.
7. No caso (muito provável) de que as estimativas sejam diferentes, os estimadores alto e baixo defendem suas estimativas para o resto da equipe.
8. O grupo debate brevemente os argumentos.
9. Uma nova rodada de estimativa é feita (etapas 4 e 5).
10. Continue até que o consenso seja alcançado e o moderador registre a estimativa.
11. Repita para todas as histórias. (HARTMAN, 2009, p. 1)

Porque esse “jogo” se tornou tão útil:

- promove a colaboração envolvendo toda a equipe;
- cria uma estimativa de consenso em vez de ter uma única pessoa conduzindo a estimativa;
- expõe problemas no início por meio da discussão de cada história de usuário.

Aqueles que realmente poderiam fazer o trabalho são os que votam. Muitas vezes, as equipes ágeis fazem com que todos votem, mesmo que não tenham ideia do trabalho envolvido na história. Eu particularmente sou extremamente contra isso. Vota quem pode fazer o trabalho de verdade e, sim, tem que ser mais que um, os motivos são óbvios e seguem os princípios dos times ágeis autogerenciáveis e com experiência. Além disso, há uma velha máxima na área de tecnologia da informação e diz respeito a: *backup*, redundância e contingência. Nunca se esqueça disso, você pode ficar “na mão” de uma hora para outra.

Quando houver empate na votação entre dois tamanhos consecutivos, basta escolher o tamanho maior e seguir em frente.

Lembre-se de que os tamanhos consecutivos podem ser 5 e 8 se você estiver usando a sequência de Fibonacci para dimensionamento (1, 2, 3, 5, 8, 13...). Ninguém deve reclamar de usar o número mais alto e uma divisão igual geralmente leva muito tempo para ser resolvida. Geralmente também resulta no número mais alto, pelo menos na minha experiência, então vamos usar esses fatos a nosso favor e seguir em frente.

Pare as discussões de implementação antes que elas se aprofundem demais; afinal, as equipes costumam ir até os detalhes técnicos quando estão discutindo uma história de usuário. Isso é bom até certo ponto, mas deve ser severamente limitado. Não mais do que uma discussão de um minuto. As pessoas que fazem o dimensionamento devem determinar em sua mente a solução mais simples possível e escolher um tamanho com base nesse cenário.

Use algum tipo de cronômetro para limitar a discussão; normalmente se limita as discussões a não mais do que 2 minutos, se der 1 minuto é melhor. Trata-se de questão de você criar uma cultura de ser eficaz.

Também é importante limitar o número de rodadas quando não se tem consenso; normalmente eu limito nos projetos até a terceira rodada, de tal forma que se exceder eu escolho o maior tamanho e sigo em frente para a próxima história.

Jogo de planejamento não é um amontoado de adultos brincando com cartas, trata-se de uma técnica mundialmente utilizada para estimativa de desenvolvimento; otimize o tempo da sua equipe de desenvolvimento, dou uma boa prática aqui; faça com que a pessoa que está criando as histórias de usuário (pode ser o *product owner* ou outro) se reúna com os líderes de controle de qualidade e desenvolvimento antes de ir jogar o *planning poker* com o *time scrum*; escrevo isso para garantir que as histórias de usuário tenham respostas para as perguntas mais óbvias que a equipe fará durante esse processo.

Por fim, é importante lembrarmos e mantermos a linha de base de uma história de usuário, ela deve ser e se manter consistente durante e para cada iteração (*sprint*). Então, o que quer que a equipe escolha como linha de base precisa ser consistente de *sprint* para *sprint*. Se um dia ideal tiver tamanho 1, todas as *sprints* precisam usar esse ponto de referência. Se uma determinada história de usuário for de tamanho 1 ou 3, então isso precisa permanecer consistente nas *sprints*.

Ao final você deve sair com uma lista de história de usuários mais bem estruturada e com as observações que o time de desenvolvimento colocar; se traduzir os *story points* em horas é melhor, veja:

StoryCard N° 2 Como consumidor, eu gostaria de uma descrição do produto, para eu saber especificadamente as características do produto. Observações: - Características gerais e características técnicas. Restrições: - Nas características gerais, no máximo 3 linhas de texto. - Nas características técnicas, até 10 linhas de texto.	Tarefas - Fazer breve texto sobre características básicas do produto; - Fazer texto sobre características técnicas do produto; - Criar "aba" para características básicas (Cor, o que vem no kit, sabor); - Criar "aba" para características técnicas (Tipo de grão, material dos produtos (cerâmica, plástico, etc.), embalagem; Horas: 2/4
StoryCard N° 3 Como consumidor, gostaria da opção de embalagens de presentes para que eu possa comprar para alguém. Observações: - Imagem da embalagem para presentes. Restrições: - Uma embalagem para presente por produto/kit.	Tarefas - Criar checkbox escrito: embalagem para presente? No momento do carrinho; - Mostrar valor adicional da embalagem para presente no momento da seleção da checkbox (pode ser um alert); - Ao clicar em efetuar pagamento, perguntar para o cliente (caso ele não tenha selecionado), se é embalagem para presente; Horas: 2

Figura 6 – Histórias do usuário após o *Planning Poker* do e-commerce do café gourmet

Fonte: Acervo do conteudista

Análise de Requisitos Funcionais, Não Funcionais e Regras de Negócios

A engenharia de requisitos é um campo de especialização aplicável em várias situações e processos. É útil até mesmo no desenvolvimento ágil de produtos além de *software* e pode ajudar a fornecer mais substância ao framework *Scrum*.

No *Scrum*, o *Product Backlog* é um conjunto dinâmico de requisitos, com o *Product Owner* responsável por seu conteúdo e gerenciamento. Isso deve ser feito continuamente e em colaboração com as partes interessadas e a Equipe de Desenvolvimento.

Para obter uma visão comum dos requisitos, alguma forma de documentação é útil, seja uma Especificação de Requisitos de Software (ERS) ou uma coleção de histórias de usuários (em nosso caso, as histórias do usuário é o que utilizamos como forma de requisitos ágeis).

Em qualquer uma das formas, é bom ser específico e usar regras como “a voz ativa”. A experiência mostra que é pelo menos tão importante conversar sobre esses requisitos em conjunto para realmente compartilhar o entendimento.

A comunicação face a face é algo fortemente adotado na comunidade ágil, onde esse entendimento compartilhado recebe uma ênfase ainda maior. Embora o caminho da engenharia de requisitos possa variar nas abordagens tradicionais e ágeis, o objetivo final permanece o mesmo, e é importante manter esse objetivo em mente para não se perder ao aplicar técnicas específicas.

As formas genéricas de pensar dentro da engenharia de requisitos são muito semelhantes no trabalho tradicional e ágil, como vemos a seguir:

- do problema à solução;
- do alto nível ao detalhe;
- consideração das consequências técnicas e financeiras na determinação do escopo e priorização dos requisitos.

No *Agile*, você cumpre os requisitos *just-in-time* apenas o suficiente. Os detalhes dos requisitos serão tratados durante a *Sprint*, ou pouco antes da *Sprint*, em sessões de refinamento.

Ao pegar um pequeno conjunto de necessidades das partes interessadas e traduzi-las em uma solução de trabalho, um *feedback* rápido é obtido. O *feedback* frequente torna possível aprender mais sobre os requisitos e adaptar o processo.

As técnicas de requisitos ágeis suportam a ideia de um esforço conjunto em torno de um conjunto limitado de requisitos, em contraste com a situação tradicional em que um especialista normalmente trabalha em um documento de requisitos abrangente.

As equipes ágeis preferem capturar os requisitos de uma forma que suportem fortemente a interação e a flexibilidade desejada. Exemplos bem conhecidos são:

- **Histórias de usuários em notas adesivas ou fichas:** normalmente é como tudo começa. O dono do produto distribui fichas para os usuários escreverem suas histórias naquele texto padrão que já vimos na unidade anterior;
- **Mapas de histórias:** é a técnica que abordei nesta unidade, onde criamos uma espinha dorsal de histórias para cada usuário que focamos, tentando construir uma jornada desse usuário, a qual quebramos em tarefas, subtarefas e designamos depois quais recortes utilizaremos para construirmos essas funcionalidades;
- **Painéis de visão:** ajudam a descrever, visualizar e validar sua visão e estratégia de produto. Eles capturam o grupo-alvo, as necessidades do usuário, os principais recursos do produto e as metas de negócios, também descrevem os principais elementos do seu modelo de negócios, incluindo concorrentes, fontes de receita, fatores de custo e canais. Não abordei esse tipo de artefato, pois ele se parece muito com o Canvas de Negócio adaptado e tem viés de desenvolvimento de produtos. Claro que pode ser adaptado para desenvolver um produto de *software*, mas as duas formas anteriores que descrevi são absolutamente eficientes e eficazes para a descrição de produto e consequente descrição de requisitos.

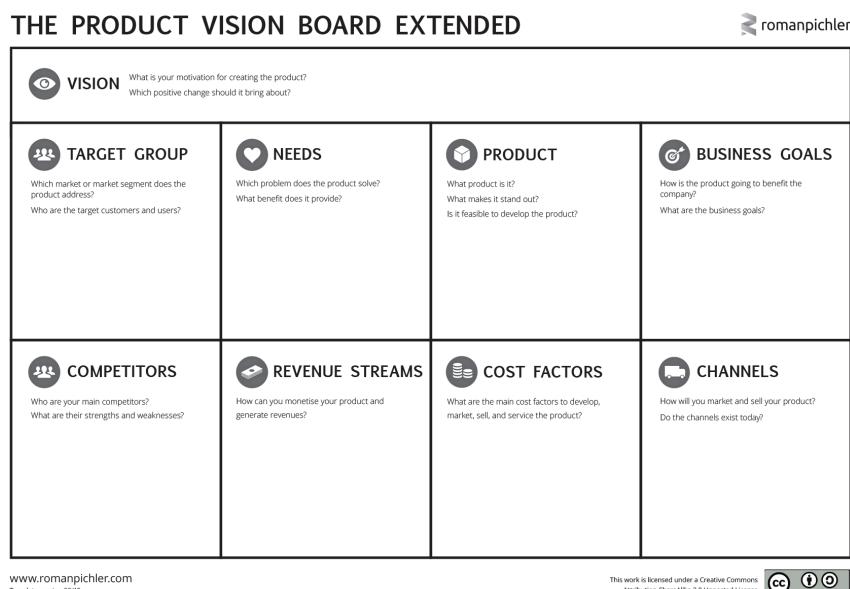


Figura 7 – Exemplo de um painel de visão utilizado para desenvolver um produto

Fonte: CHAGAS, 2017

Em tempo, muitos alunos e profissionais em início de jornada ainda têm dúvidas severas sobre o papel do dono do produto em um time scrum e do próprio analista de negócios nessas fases de análise de requisitos e de descoberta de requisitos no projeto ágil; vejamos como resolvemos isso:

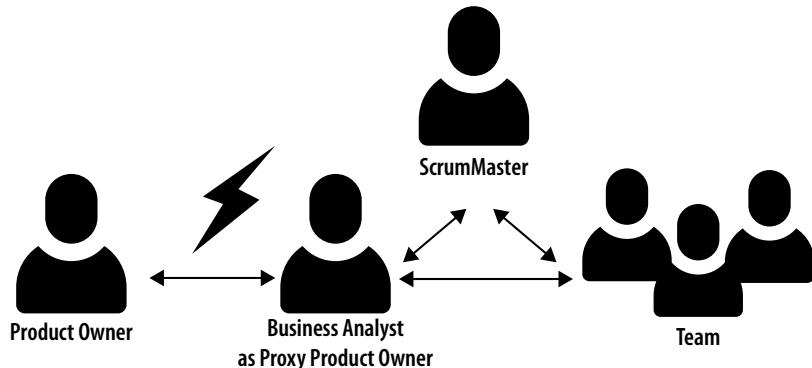


Figura 8 – “O *Product Owner* é aquele que determina as metas e conduz o time na entrega constante de valor”

Fonte: COHN, 2009

Em um ambiente ágil, os aspectos funcionais e não funcionais são capturados juntos, analisando-se cada requisito individual (normalmente uma história de usuário separada), por exemplo, com a ajuda das 7 dimensões do produto de Ellen Gottesdiener e Mary Gorman.



Figura 9 – Onde estão os Requisitos Funcionais e Não Funcionais numa sessão de descoberta de requisitos ágil

Fonte: Adaptada de GONÇALVES, 2019

Isso leva a um melhor entendimento entre as partes envolvidas, detalhes adicionais como critérios de aceitação e frequentemente a novos requisitos (divisão de histórias de usuários).



As 7 dimensões do produto. Disponível em: <https://bit.ly/3syFUPU>

Vale a pena conhecermos essa técnica emergente de Elicitação e análise ágil de requisitos.

Então, para criar um produto de alto valor, as sete dimensões do produto são usadas para cada opção de produto identificada. Os parceiros, divididos em três grupos: negócio, cliente e tecnologia, são identificados e cada grupo identifica o valor buscado na opção de produto. Um plano é criado para fornecer a opção de produto e conversas estruturadas são

usadas para descoberta colaborativa contínua, independentemente do tamanho inicial da necessidade do produto, que pode ser expressa de várias maneiras (recurso, caso de uso ou história – por exemplo, usando conversas estruturadas), e estas alternativas são exploradas.

Vamos dar uma olhada em cada uma das sete dimensões do produto que são usadas para esclarecer cada opção de produto; deixei a imagem acima como referência para as definições:

- **Usuário:** como os usuários irão interagir com o produto?
- **Interface:** como o produto se conectará a usuários, sistemas e dispositivos?
- **Ação:** quais recursos o produto oferece aos usuários?
- **Dados:** que tipo de dados e informações são necessários e onde são armazenados?
- **Controle:** que tipo de restrições o produto impõe?
- **Ambiente:** a que tipo de propriedades físicas e plataformas de tecnologia o produto deve estar em conformidade?
- **Atributo de qualidade:** quais atributos de qualidade restringem e controlam o produto?

Respondemos a essas perguntas através de sessões de descoberta com pensamento visual e linguagem visual para ajudar as partes interessadas a chegarem a um consenso sobre os requisitos de produto apropriados e como criar o nível certo de detalhes em suas histórias de usuário.

Por isso utilizamos um **Canvas** nas unidades anteriores para o nosso projeto. O melhor de tudo isso é que usamos ferramentas de baixa tecnologia, esboços, diagramas, notas adesivas coloridas e, o mais importante, conversas estruturadas, as equipes se reúnem de forma colaborativa para chegar a um consenso sobre os requisitos de produto de maior valor.

Então, uma boa pergunta para você fazer caso seja o “analista de negócios” ou o “dono do produto” poderia ser: quanto tempo devemos gastar em descobertas?

Bom, há 2 problemas aqui:

- primeiro, pressupõe-se que você faça a descoberta primeiro e depois a entrega, o que não é verdade. Você deve descobrir e entregar o tempo todo. Requisitos ágeis são descobertos o tempo todo. Mude seu “mindset”, lembre-se de que só fazemos um documento de requisitos completos do sistema quando criamos um DER – Documento de Especificação de Requisitos; nele, como se fosse uma cláusula pétreia, os requisitos postos não mudam.... (e foi nessa de não mudar requisitos que a crise do software veio e muita gente “cabeça dura” perdeu o emprego);
- em segundo lugar, isso implica que existe uma resposta que é verdadeira em todas as situações. Bom, como sabemos, não existe. Você precisa fazer descobertas suficientes para mitigar os riscos e as pessoas que desenvolveram devem se sentir confortáveis. Empatia é fundamental em ágil, colocar-se no lugar do outro é mandatório.

Pense nisso; escrever código é uma maneira perfeitamente válida de aprender, se for a maneira mais rápida de aprender. Na maioria das vezes não é.

Minha experiência nesses mais de 30 anos na profissão tem demonstrado que alunos, mesmo de últimos anos ou semestres iniciam a escrita de código muito cedo e

acabam usando a escrita de código como descoberta de requisitos e, pior, descoberta de arquitetura, o que leva a um projeto, independentemente de ser ágil ou não, a uma colcha de retalhos completamente ininteligível. Isso não é projeto ágil, senhoras e senhores, é “*go crazy horses*”.

Então, descubra um requisito, defina um resultado claro e pergunte: qual é o melhor caminho para o resultado desejado?

Precisamos descobrir quais necessidades, pontos fracos, desejos, vontades ou o que impacta esse resultado.

Depois pergunte: quais dessas oportunidades, se as abordássemos, teria o maior impacto sobre o resultado desejado?

A partir daí, precisamos descobrir qual solução melhor atenderá à oportunidade-alvo que escolhemos. Percebeu que você está trabalhando sua história de usuário, por exemplo?

A descoberta não é um processo linear. É um processo complexo e confuso que está acontecendo ao mesmo tempo e é assim mesmo! Como a descoberta não é linear, sempre que aprendemos novas informações, precisamos rever nossas decisões anteriores e perguntar se essas novas informações mudam alguma coisa.

Por isso é importante o princípio de adiar a tomada de decisão ao máximo, para que o pensamento e as oportunidades de como realizar sejam feitas de forma assertiva no momento ideal; bem, isso, além de ágil, é um princípio *LEAN*.

Talvez uma das atitudes que você deverá tomar será melhorar suas perguntas, já que a descoberta não é linear. Em vez de perguntar: “quantas descobertas precisamos fazer?”, as equipes devem fazer descobertas contínuas e perguntar: “como podemos integrar rapidamente as novas informações que estamos aprendendo sem perder o ritmo?”.

Independentemente disso, precisamos aprender a sermos críticos com a análise de nossas ideias, isso significa que você precisará manter o estado de dúvida e realizar uma investigação sistemática e prolongada, afinal estes são os fundamentos do pensamento.

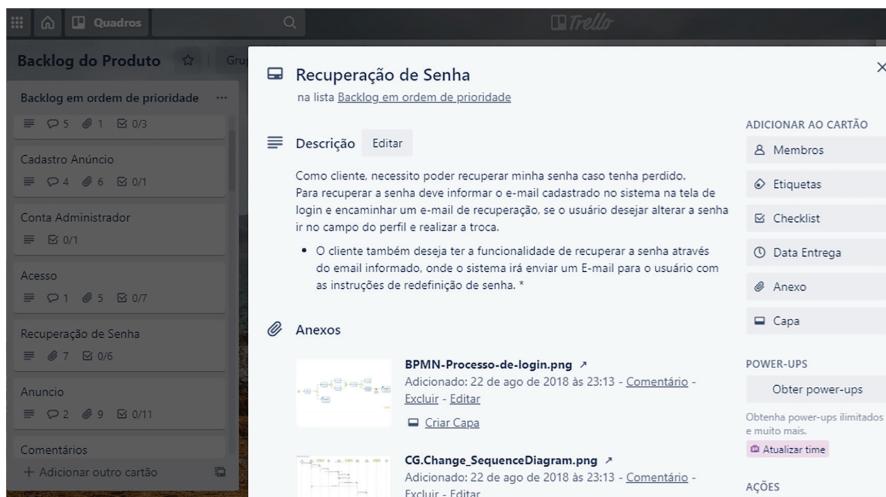


Figura 10 – *Trello* em ação para auxiliar nos requisitos a partir da história do usuário; administrar usuários, uma tarefa relacionada à Recuperação de senha

Fonte: Acervo do conteudista

Veja na Figura anterior que a ferramenta *TRELLO*, por exemplo, pode nos ajudar a organizar as descobertas de requisitos, melhorar seu entendimento, criar artefatos e deixar anexo para quem os ler ter uma visão mais clara. Além de que, nesse nosso projeto, ele é um cartão de história do usuário, percebe?!

Conforme vamos descobrindo como esse requisito vai se transformar em uma entrega, vamos agregando os componentes; veja os anexos do mesmo cartão (com os artefatos que enriquecem a informação).

Anexos

- BPMN-Processo-de-login.png ↗ Adicionado: 22 de ago de 2018 às 23:13 - [Comentário](#) - [Excluir](#) - [Editar](#)
[Criar Capa](#)
- CG.Change_SequenceDiagram.png ↗ Adicionado: 22 de ago de 2018 às 23:13 - [Comentário](#) - [Excluir](#) - [Editar](#)
[Criar Capa](#)
- Tabela - Regras de negocio.xlsx ↗ Adicionado: 22 de ago de 2018 às 23:04 - [Comentário](#) - [Excluir](#) - [Editar](#)
- StoryTelling.docx ↗ Adicionado: 22 de ago de 2018 às 23:04 - [Comentário](#) - [Excluir](#) - [Editar](#)
- tela senha.JPG ↗ Adicionado: 20 de ago de 2018 às 22:47 - [Comentário](#) - [Excluir](#) - [Editar](#)
[Criar Capa](#)
- tela senha.JPG ↗ Adicionado: 20 de ago de 2018 às 22:47 - [Comentário](#) - [Excluir](#) - [Editar](#)
[Criar Capa](#)
- alterar senha.png ↗ Adicionado: 20 de ago de 2018 às 21:08 - [Comentário](#) - [Excluir](#) - [Editar](#)
[Criar Capa](#)

[Mostrar menos anexos.](#)

[Adicionar um anexo](#)

Capa

POWER-UPS

Obtenha power-ups ilimitados e muito mais.
[Obter power-ups](#)

AÇÕES

→ Mover
Copiar
Criar modelo
Seguir
Arquivar
Compartilhar

Figura 11 – Artefatos (inseridos no *Trello*) de design, negócio e arquitetura enriquecendo o requisito ágil

Fonte: Acervo do conteudista

Vejamos a sequência oferecida para essa funcionalidade:

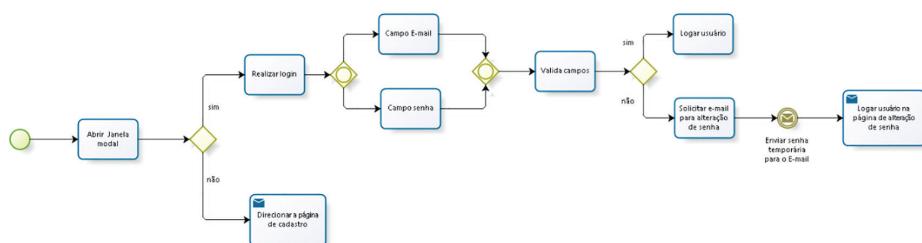


Figura 12 – O processo de alteração de senha

Fonte: Acervo do conteudista

Aqui logo abaixo, as regras de negócios associadas a Alteração de Senha.

Tabela 1 – Regras de negócio associadas a Alteração de senha

Plataforma		
RN1	Plataforma Web	Deverá ser compatível com <i>desktop</i>
RN2	Plataforma Web	Deverá ser compatível com <i>mobile</i>
RN3	Usuários	Poderá ser acessada por pessoas logadas
RN4	Usuários	Poderá ser acessada por pessoas não logadas
Segurança		
RN5	Segurança	Deverá ter segurança de dados pessoais
RN6	Segurança	Deverá possuir certificado SSL
Filtro – Busca		
RN7	Campo de pesquisa	O campo de pesquisa deverá buscar os produtos pelo nome do produto
Filtro de ordenação		
RN8	Filtro	O site deverá ter filtros de produtos
RN9	Filtro	O filtro poderá ser separado por consoles
RN10	Filtro	O filtro poderá ser separado por acessórios
RN11	Filtro	O filtro poderá ser separado por jogos
RN12	Filtro	Poderá ser filtrado também por localidade do anunciante
RN13	Filtro	Poderá ser filtrado também por tipo de console (ps3, ps4, xbox 360, xbox one...)
RN14	Ordenado	Poderá ser ordenado também por ordem alfabética
RN15	Ordenado	Poderá ser ordenado também por valor mínimo
RN16	Ordenado	Poderá ser ordenado também por valor máximo
Exibição da página de lista de produtos		
RN17	Lista de produtos	Todas as páginas de listas de produtos do site devem ser limitadas a 23 produtos por página
RN18	Lista de produtos	Na página de produto as páginas se iniciam a partir de 28 anúncios
RN19	Lista de produtos	Deverá conter foto do produto
RN20	Lista de produtos	Deverá conter breve descrição do item
RN21	Lista de produtos	Deverá conter os tipos disponíveis de negociação (compra e/ou troca)
RN23	Lista de produtos	Deverá conter preço do produto

Veja a seguir a jornada do cliente levando em consideração as regras de negócios atribuídas a ele num estilo narrativo e integrado incluindo a menção de tabelas, é uma forma bastante interessante de abordar o tema e organizar de forma lógica e funcional o desafio do entendimento dos requisitos encaixando tudo em suas partes.

A necessidade do cliente

(RN1-RN2) o cliente precisa de uma plataforma web, com páginas responsivas, que possam ser acessadas através de qualquer navegador, de forma que ele possa acessar o website através de qualquer dispositivo, inclusive móvel, e em qualquer lugar do mundo, onde ele esteja conectado à internet.

(RN5) o cliente precisa de itens de segurança da informação que sejam visíveis no navegador para o cliente, para segurança de dados no que tange às compras, trocas e vendas realizadas através do website (RN6). O cliente necessita de um certificado SSL que ateste a segurança da plataforma publicamente para que, qualquer usuário possa identificar que está em ambiente seguro, mesmo não entendendo muito sobre computadores.

O cliente precisa de transparência em seus negócios, como a opção de “status dos seus pedidos”, informando a ele, em tempo real, seja ele vendedor ou comprador, o status em que se encontra sua negociação. Co parte da transparência nas negociações, (RN28) o cliente solicita que seja possível, em qualquer momento, entrar em contato com a outra parte, através da página de mensagens disponibilizado no website quando a negociação é iniciada.

(RN7) O cliente necessita buscar os produtos através de uma página de pesquisa, onde ele possa encontrar qualquer produto anunciado no website pelo seu nome. Haverá filtros de categorias (RN8-RN11) e parâmetros de ordenação e filtros específicos (RN12-RN16). (RN17) Na página de busca, os resultados encontrados, devem ser limitados a 30 anúncios por página, (RN18) iniciando as paginações a partir de 28 anúncios. (RN19) Os produtos exibidos na lista devem conter fotos, (RN20) breve descrição do item, (RN21). Tipos disponíveis de negociação, (RN12) e preço do produto. O cliente deseja ser capaz de ir a uma página que tenha todos os detalhes do produto escolhido.

Na tela de detalhe do produto, o cliente deseja ter a opção de prosseguir com a negociação ou voltar para a página anterior. O cliente deseja que o meio de negociação do produto, seja compra, troca, esteja visível na página, conforme opções de negociação cadastradas previamente no anúncio.

(RN24) A partir da tela de detalhe do produto, o cliente deseja ser capaz de prosseguir com a negociação, onde será possível prosseguir de duas formas:

- **Fluxo de compras:** (RN25) O cliente será redirecionando para a área de confirmação de dados financeiros, onde poderá escolher a opção de pagamento, que foi previamente cadastrada no anúncio podendo ser limitada, pois cada vendedor poderá escolher quais formas irá disponibilizar. Ao escolher a forma de pagamento e confirmar os dados, (RN26) o cliente poderá interagir com o vendedor através de uma página de mensagens que será disponibilizado pelo site, para decidirem a forma de entrega;
- **Fluxo de trocas:** (RN26) O cliente irá interagir com o vendedor através de uma página de mensagens que será disponibilizado pelo site, onde irá combinar com o vendedor sobre a negociação que deseja. (RN132-RN133) Para uma melhor negociação entre os usuários só será possível só será possível finalizar a troca com a confirmação de ambos, sendo o anunciante o primeiro a confirmar. (RN27). O que for combinado entre o vendedor e o comprador, deve ser por conta própria de ambos, pois o site não se responsabilizará por quaisquer reclamações ou não conformidade dos itens anunciados.

(RN28) O cliente precisa de uma página que tenha as informações do vendedor do qual está anunciando o item que o cliente está interessado. Nesta página, o cliente deseja ser capaz de visualizar os seguintes itens do vendedor:

- (RN29) Quantidade de vendas realizadas pelo vendedor desde o início;
- (RN30) Quantidade de vendas realizadas pelo vendedor nos últimos 3 meses;
- (RN31) Reputação de vendas, através de um intervalo de 0 a 5, sendo 0 ruim 5 bom;
- (RN32) Outros produtos anunciados pelo vendedor.

(RN33) O cliente precisa ter a facilidade de entrar em contato conosco, por isso, é necessária uma página de contato, onde contenha os dados como no (RN34) telefone

e (RN35) *e-mail*. Estes dados devem estar disponíveis para que ele possa entrar em contato conosco.

(RN36) O cliente precisa de um formulário, na tela de contato, onde possa informar (RN37) nome completo, (RN38) *e-mail* (RN39) telefone (RN40) a mensagem que deseja e deve selecionar uma categoria com (RN41-RN42) temas previamente definidos por nós (RN43). Esse formulário será recebido pela nossa área de atendimento ao cliente que dará o retorno ao cliente através do *e-mail* informado no formulário.

(RN44) O cliente deseja acessar o *website* de forma anônima, ou seja, sem *login*. Desta forma, ele deve operar o sistema com acesso apenas às seguintes páginas:

- (RN45) Busca dos produtos anunciados;
- (RN46) Detalhe do produto;
- (RN47) Dados do vendedor;
- (RN48) Contato do *site*;
- (RN49) Cadastro do usuário;
- (RN50) *Login*.

(RN51) Quaisquer páginas que o usuário tente acessar, que esteja fora desta lista, o *website* deve exibir uma janela (modal) no qual o cliente e terá a (RN52) opção de fazer o *login* ou (RN53) efetuar um cadastro de usuário.

(RN54) O cliente deseja ter acesso a uma página de cadastro de usuário, na qual poderá cadastrar: dados pessoais (RN55-RN60), (RN61-RN70) dados financeiros, (RN71-RN77) endereços de entrega e (RN78) perfil de utilização da ferramenta.

O cliente deseja, no cadastro de usuário, ter a opção de selecionar um perfil *premium* onde será cobrado um valor e reais, por funções que os usuários de perfil gratuito, perfil simples, não podem ter, tais como:

- (RN83) Produtos poderão ser anunciado com destaque, o que significa, maior exposição nos resultados das buscas feita pelos clientes. Estes produtos estão sempre no topo dos resultados;
- (RN81) Sem limite máximo de anúncios mensais, durante a vigência do perfil *premium*. (RN79) Diferente da conta gratuita que será limitada a 3 anúncios mensais;
- (RN82) Duração do anuncio ilimitada, durante a vigência do perfil *premium*. (RN80) diferente da conta gratuita que irá expirar o anuncio dentro de 30 dias.

(RN84) O perfil *premium* terá vigência limitada dependendo do plano que contratou. Assim que o tempo de vigência exceder, o perfil será automaticamente trocado para perfil grátis.

O cliente deseja ter acesso à função de *login*, onde possa informar seu usuário e senha para ter acesso ao sistema. O cliente também deseja ter a funcionalidade de recuperar a senha através do *e-mail* informado, onde o sistema irá enviar um *e-mail* para o usuário com as instruções de redefinição de senha.

Fonte: Documento de Regras de Negócio para a disciplina elaborado pelo Autor

Veja, declaramos a jornada inteira do cliente para que você possa ver como pode criar um documento desse. Além disso, veja como seria o mockup para esse item de Alterar a senha.

Senha

Senha atual:	<input type="text"/>
Senha nova:	<input type="text"/>
Senha nova novamente:	<input type="text"/>

Cancelar **Salvar alterações**

Figura 13 – *Mockup* da tela da IHC – interface humano-computador para a Alteração de senha

Por fim, veja as observações que você pode colocar no cartão dessa atividade do *Trello*, visando deixar as coisas claras para o desenvolvimento.

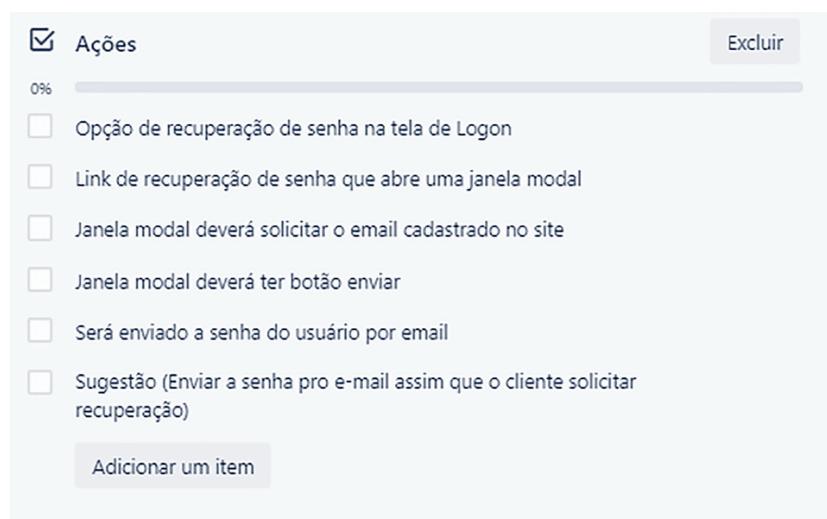


Figura 14 – Ações esperadas para a funcionalidade de Alterar senha

Fonte: Acervo do conteudista

Aqui temos uma amostra do que você pode fazer com a evolução de um *card* do *Trello*, desde sua humilde origem como história do usuário até as instruções de *design*, arquitetura de interface e de dados, passando por diagramas de UML. Claro que veremos isso em profundidade em outras unidades desta disciplina.

Material Complementar

Indicações para saber mais sobre os assuntos abordados nesta Unidade:

Vídeos

User Story Mapping – Miguel Fernandes

<https://youtu.be/qoPOXzhzESE>

Creando un mapa de historias de usuario

<https://youtu.be/kVzEBsUyEfg>

Inception – Jornada do usuário

<https://youtu.be/54oPgmNsZRw>

Inception – Desenhando o processo

<https://youtu.be/a0plT8heXhY>

Referências

AGILE ALLIANCE. **Product backlog.** Disponível em: <[https://www.agilealliance.org/glossary/backlog/#q=~\(infinite~false~filters~\(postType~\(~'page~'post~'aa_book~'aa_event_session~'aa_experience_report~'aa_glossary~'aa_research_paper~'aa_video\)~tags~\(~'backlog\)\)~searchTerm~'~sort~false~sortDirection~'asc~page~1\)}>](https://www.agilealliance.org/glossary/backlog/#q=~(infinite~false~filters~(postType~(~'page~'post~'aa_book~'aa_event_session~'aa_experience_report~'aa_glossary~'aa_research_paper~'aa_video)~tags~(~'backlog))~searchTerm~'~sort~false~sortDirection~'asc~page~1)}>)>. Acesso em: 25/08/2020.

COULTER, B.; FOUQUE, C.; RIETHOF, D. K. **From user story mapping to high-level release plan.** 2017. Disponível em: <<https://www.caktusgroup.com/blog/2017/07/31/user-story-mapping-high-level-release-plan/>>. Acesso em: 25/08/2020.

HARTMAN, B. **An introduction to planning poker.** 2009. Disponível em: <<https://dzone.com/articles/introduction-planning-poker>>. Acesso em: 25/08/2020.



Cruzeiro do Sul
Educacional