



Análise e Desenvolvimento de Sistemas

Raphael Joaguarany de Oliveira – RA: 3486950401

ANHANGUERA

Rio de Janeiro, 1 de Maio de 2023

1 - INTRODUÇÃO

O relatório tem em seu objetivo demonstrar um projeto de software referente ao gerenciamento de finanças pessoais, realizando os princípios da metodologia ágil Scrum por meio da utilização da plataforma Trello.

O software foi desenvolvido para ajudar as pessoas a gerenciar suas finanças de forma simples e eficiente, permitindo o controle de despesas e investimentos de maneira integrada. A metodologia ágil Scrum foi utilizada para permitir um desenvolvimento mais ágil e eficiente do software, com a equipe trabalhando em sprints de duas semanas e entregando incrementos funcionais a cada ciclo.

A plataforma Trello foi escolhida como ferramenta de gestão de projetos por sua facilidade de uso e capacidade de integração com outras ferramentas, permitindo que a equipe pudesse acompanhar o progresso do projeto de forma transparente e colaborativa. Neste relatório, apresentaremos o processo de desenvolvimento do software, os desafios enfrentados e as soluções adotadas, além dos resultados alcançados e os próximos passos para a evolução do produto.

2 - MÉTODO

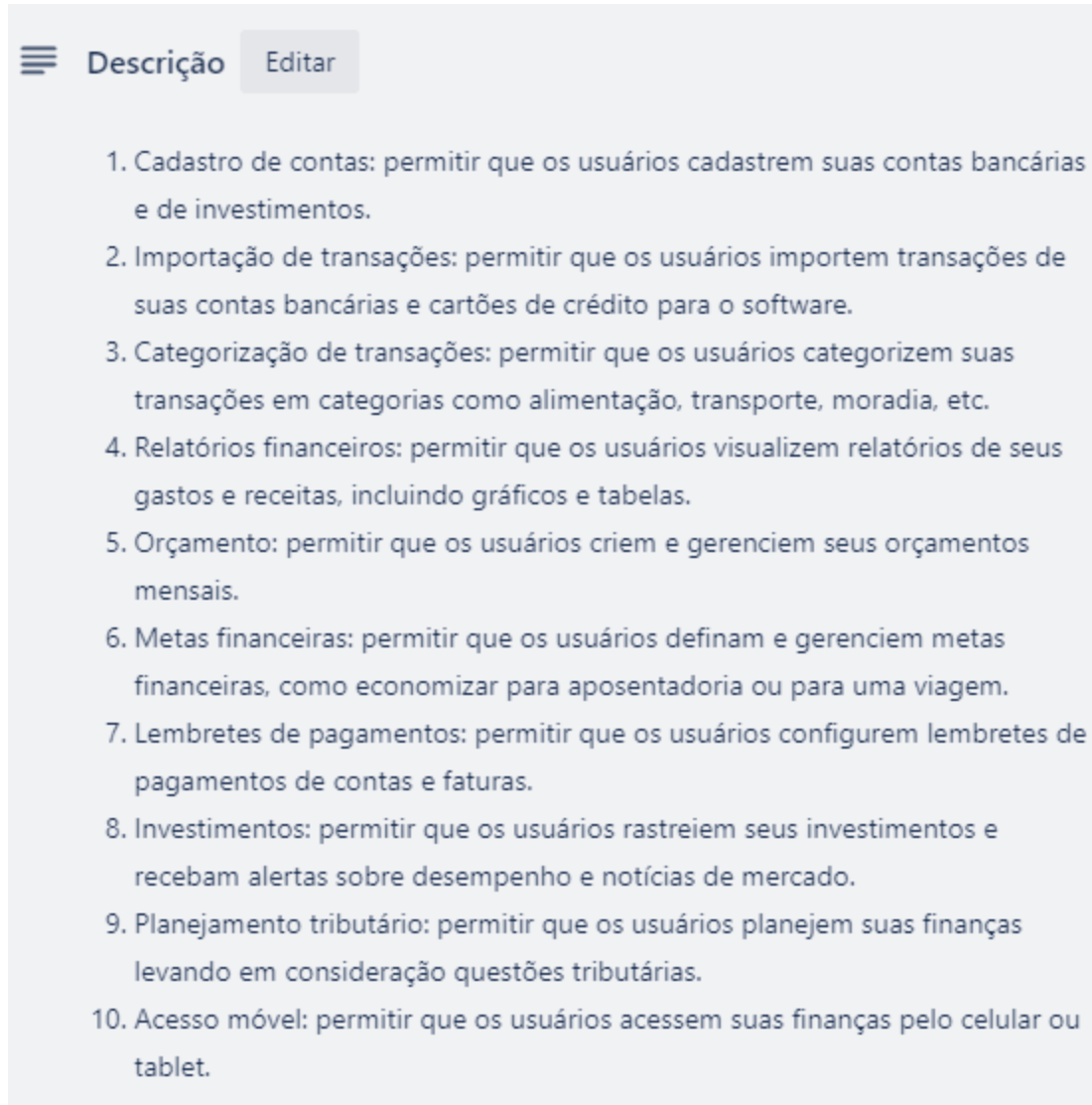
O scrum utilizado é a metodologia ágil com um período de tempo curto, onde nesta situação foi simulado um projeto de aplicação para um software de gerenciamento de finanças.

Foi solicitado pela empresa um aplicativo de gerenciamento de finanças onde foi adotado para o desenvolvimento do software de gerenciamento de finanças o processo scrum ágil que consiste em um ciclo de desenvolvimento de software iterativo composto por três fases, sendo eles de Planejamento, Execução e Revisão.

No ciclo de planejamento foi utilizado para identificar as funções do software e priorizar em backlog, onde o Product Owner que estava como responsável do projeto, juntamente com a equipe de desenvolvimento, trabalharam para garantir que as funcionalidades mais importantes fossem primeiramente implementadas.

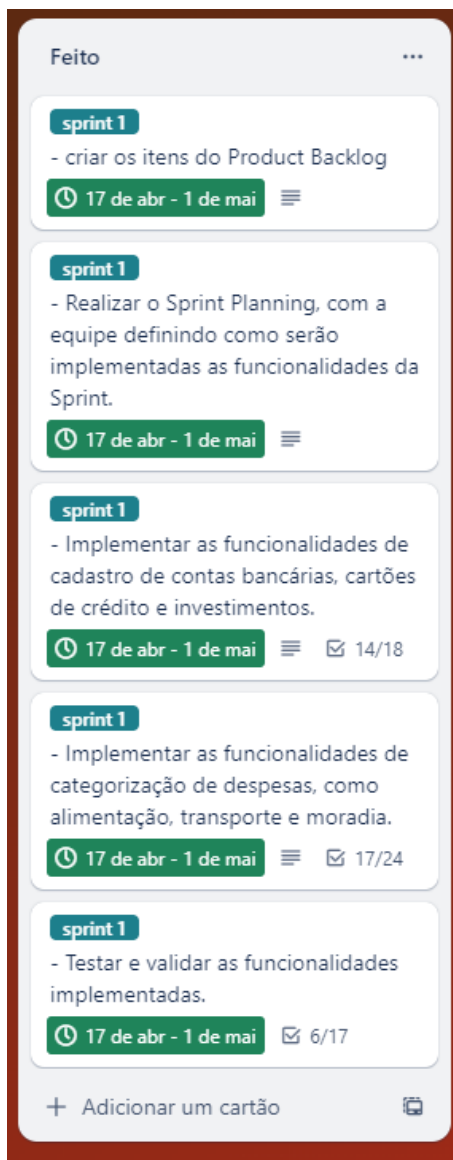
Relatório de Aula Prática - Projeto de Software

Na fase de criação de itens de backlog, foi atribuído a descrição do que foi solicitado a equipe de desenvolvimento, como podemos ver no item abaixo:



Como podemos ver no sprint 1 realizado através da plataforma trello, as tarefas foram realizadas pela equipe no período de 17 de abril até 1 de maio com tarefas realizadas e passando para o próximo sprint, conforme imagem abaixo:

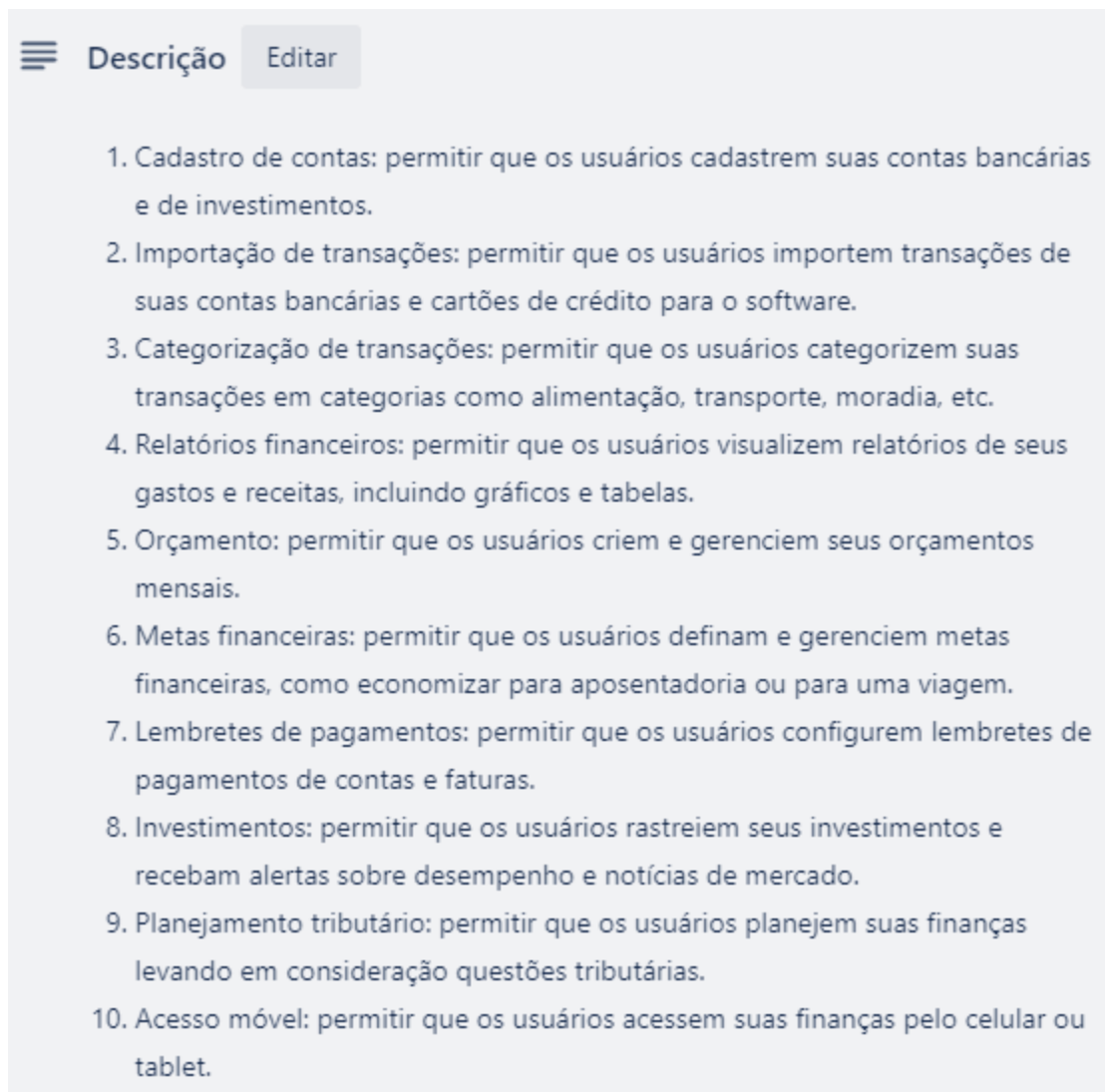
Relatório de Aula Prática - Projeto de Software



Na execução do projeto de desenvolvimento do software, onde podemos ver os itens que estão já em andamento no sprint 2, é onde a equipe precisa se empenhar de forma redobrada pois o tempo de execução é curto e sem brechas para erros, tendo que ser realizadas no período de 2 de maio até o dia 16 de maio e conforme podemos verificar na imagem abaixo, as descrições a serem realizadas a partir dos primeiros sprints dessa segunda fase do projeto que está em

Relatório de Aula Prática - Projeto de Software

andamento:



Observando por inteiro a organização da equipe referente ao sprint 2, será realizada uma Sprint Planning para que seja atribuído junto com a equipe uma revisão do product backlog e lhes apresentar prioridades que devam ser seguidas.

Após isso, a equipe irá efetuar uma definição de sprint goal com base nos itens selecionados para a sprint e finalização com o comprometimento da equipe em concluir o trabalho planejado para a sprint, conforme verificamos nas imagens abaixo:




Descrição

Editar

1. Preparação: Antes da reunião de Sprint Planning, é importante que a equipe tenha revisado e priorizado o Product Backlog. Eles também devem ter uma compreensão clara das capacidades e limitações do sistema e das necessidades do usuário. Além disso, a equipe deve ter uma estimativa da sua capacidade de trabalho para a Sprint, com base em sua velocidade anterior.
2. Revisão do Product Backlog: Na primeira parte da reunião de Sprint Planning, a equipe de desenvolvimento revisará o Product Backlog com o Product Owner. O Product Owner apresentará as prioridades do Product Backlog, explicando o valor que cada item traz para o usuário final e para o negócio. A equipe pode fazer perguntas e discutir os critérios de aceitação para cada item. O objetivo desta fase é ter certeza de que a equipe tem uma compreensão clara do que está sendo solicitado.
3. Seleção de itens do Product Backlog: Após a revisão do Product Backlog, a equipe selecionará os itens do Product Backlog que serão incluídos na Sprint. Eles devem levar em consideração a prioridade, a complexidade e a capacidade estimada da equipe. A equipe deve ter em mente que a sua capacidade de trabalho pode ser afetada por feriados, férias ou outras interrupções durante a Sprint.
4. Definição do Sprint Goal: Com base nos itens selecionados para a Sprint, a equipe define um objetivo claro e conciso para a Sprint. Esse objetivo é o Sprint Goal e deve ser alcançável dentro da Sprint. Ele deve comunicar a intenção da Sprint para toda a equipe e ajudar a manter o foco no trabalho durante a Sprint.
5. Planejamento da implementação: Nesta fase, a equipe planeja a implementação dos itens selecionados para a Sprint. Eles podem criar tarefas para cada item, estimar o tempo necessário para completá-las e discutir as dependências entre elas. O objetivo é criar um plano para alcançar o Sprint Goal. Durante esta fase, a equipe pode identificar riscos ou problemas que possam surgir e discutir estratégias para minimizá-los.
6. Comprometimento da equipe: Finalmente, a equipe compromete-se a concluir o trabalho planejado durante a Sprint. Eles devem estar confiantes de que o objetivo é alcançável e que o plano de implementação é viável. Eles também devem concordar em atualizar regularmente o progresso e revisar o Sprint Backlog conforme necessário durante a Sprint.

Relatório de Aula Prática - Projeto de Software

Partindo para a sprint 3, podemos observar a implementação das funcionalidades de planejamento financeiro, como orçamentos e previsões de gastos e receitas, onde serão tratadas discussões com a equipe sobre a análise de requisitos, design, testes, revisões e ajustes e por fim documentações, conforme verificamos abaixo um prazo para essa sprint de 16 de maio à 30 de maio:

 Descrição Editar

Análise de requisitos:

- Entender claramente os requisitos do usuário e do negócio.
- Avaliar as limitações técnicas do sistema.

Design:

- Criar um design que atenda aos requisitos e também seja escalável, robusto e fácil de manter.
- Decidir quais tecnologias serão usadas.
- Criar um plano de teste.

Desenvolvimento:

- Implementar as funcionalidades de acordo com o design criado.
- Seguir as melhores práticas de codificação.
- Criar testes unitários.

Testes:

- Realizar testes funcionais, de integração e de desempenho para garantir que tudo funcione conforme o esperado.
- Testar a entrada e validação de dados para contas bancárias, cartões de crédito e investimentos.
- Verificar se os dados são armazenados corretamente.

Revisão e ajustes:

- Revisar os resultados dos testes.
- Fazer ajustes necessários para garantir que todas as funcionalidades estão funcionando corretamente e que os requisitos estão sendo atendidos.

Documentação:

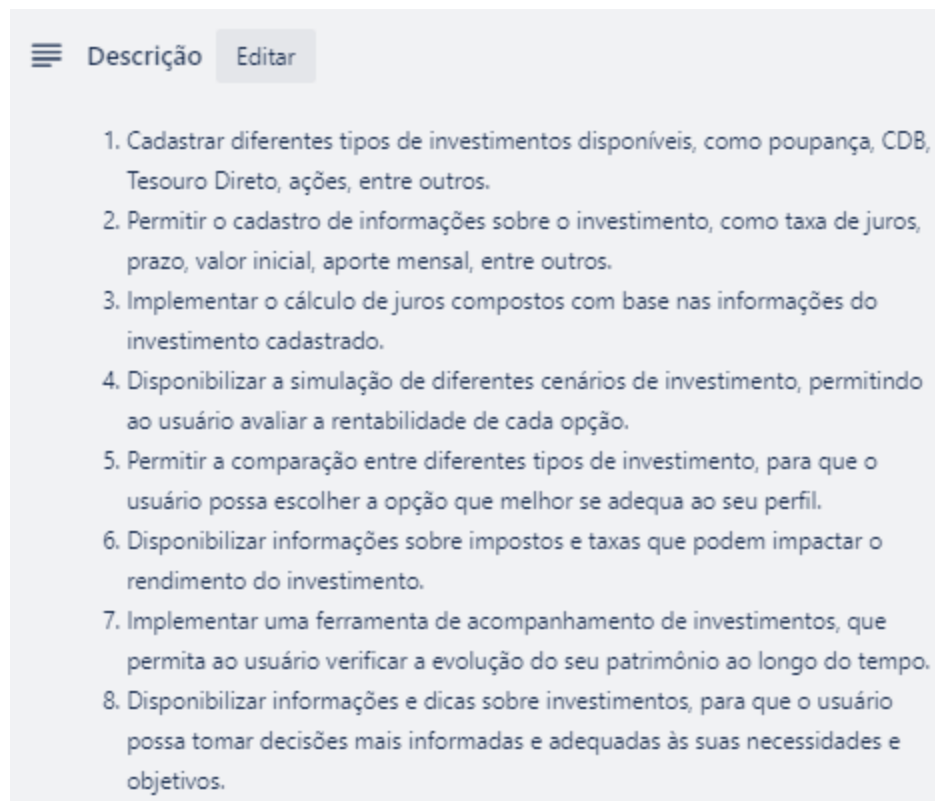
- Criar documentação técnica para desenvolvedores.
- Criar documentação do usuário para explicar como usar as novas funcionalidades.

Implantação:

- Implantar as funcionalidades no ambiente de produção.
- Garantir que a nova funcionalidade esteja disponível para os usuários finais e que todas as configurações estejam corretas.
- Realizar testes adicionais para garantir que tudo esteja funcionando corretamente em produção.


Relatório de Aula Prática - Projeto de Software

Ainda seguindo pelo sprint 3, verificamos sobre implementar as funcionalidades de simulação de investimentos e cálculo de juros compostos, onde a equipe irá implementar diversos tipos de funções necessárias e estipuladas para essa sprint em sua aba que precisam ser iniciadas no processo de desenvolvimento, conforme verificamos na imagem abaixo:



Nesta fase do desenvolvimento, a equipe irá efetuar testes e validações de todas as funcionalidades que foram implementadas até o momento e por fim verificar se o software está em conformidade com as normas de segurança.

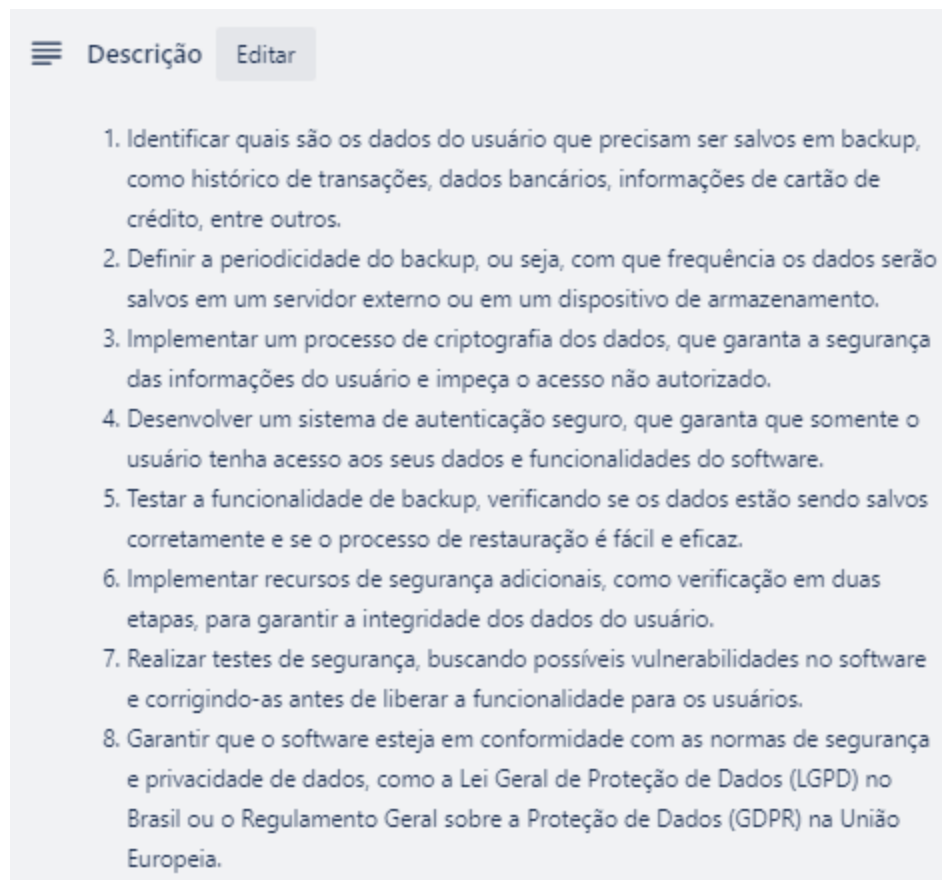
Relatório de Aula Prática - Projeto de Software

 Descrição Editar

1. Identificar as funcionalidades que foram implementadas e definir os critérios de aceitação para cada uma delas.
2. Criar casos de teste para cada funcionalidade, com cenários diversos que permitam verificar o funcionamento correto do sistema.
3. Realizar testes manuais e automatizados em cada funcionalidade, utilizando os casos de teste criados.
4. Registrar os resultados dos testes e documentar possíveis falhas encontradas.
5. Corrigir as falhas identificadas e realizar novos testes para garantir que as correções foram bem-sucedidas.
6. Verificar se as funcionalidades implementadas atendem aos requisitos definidos e se estão funcionando de acordo com o esperado.
7. Realizar testes de integração para garantir que todas as funcionalidades trabalhem corretamente juntas.
8. Realizar testes de performance para verificar se o software é capaz de suportar a carga esperada de usuários.
9. Validar as funcionalidades implementadas com usuários finais, buscando feedbacks para possíveis melhorias e ajustes.
10. Verificar se o software está em conformidade com as normas de segurança e privacidade de dados.

Nesta última fase do desenvolvimento, onde encontramos a sprint 4 do projeto, é onde a equipe irá implementar as integrações com serviços de investimentos e bancos para sincronizar dados de contas e transações e também as funcionalidades do backup juntamente com a segurança dos dados de todos os usuários, conforme podemos verificar em sua descrição de sprint:

Relatório de Aula Prática - Projeto de Software



The image shows a screenshot of a software development tool interface. At the top, there is a navigation bar with a hamburger menu icon on the left, followed by the text "Descrição" and a button labeled "Editar". Below this, there is a list of eight tasks, each numbered from 1 to 8. The tasks are written in a light blue font on a light gray background.

1. Identificar quais são os dados do usuário que precisam ser salvos em backup, como histórico de transações, dados bancários, informações de cartão de crédito, entre outros.
2. Definir a periodicidade do backup, ou seja, com que frequência os dados serão salvos em um servidor externo ou em um dispositivo de armazenamento.
3. Implementar um processo de criptografia dos dados, que garanta a segurança das informações do usuário e impeça o acesso não autorizado.
4. Desenvolver um sistema de autenticação seguro, que garanta que somente o usuário tenha acesso aos seus dados e funcionalidades do software.
5. Testar a funcionalidade de backup, verificando se os dados estão sendo salvos corretamente e se o processo de restauração é fácil e eficaz.
6. Implementar recursos de segurança adicionais, como verificação em duas etapas, para garantir a integridade dos dados do usuário.
7. Realizar testes de segurança, buscando possíveis vulnerabilidades no software e corrigindo-as antes de liberar a funcionalidade para os usuários.
8. Garantir que o software esteja em conformidade com as normas de segurança e privacidade de dados, como a Lei Geral de Proteção de Dados (LGPD) no Brasil ou o Regulamento Geral sobre a Proteção de Dados (GDPR) na União Europeia.

a última fase do sprint conta com prazo de conclusão a ser entregue do dia 13 de junho à 27 de junho, os mesmo se encontram em uma backlog da sprint conforme imagem abaixo:

Relatório de Aula Prática - Projeto de Software



Relatório de Aula Prática - Projeto de Software

Esta fase da sprint 4 conta com o prazo final para que seja efetuada a finalização do desenvolvimento, testes finais para que o produto seja aceito e a entrega do software de gerenciamento de finanças.

Na imagem abaixo podemos verificar um checklist onde os desenvolvedores podem finalizar funções já testadas e realizadas pelos mesmos e analisadas pelo Product Owner.

Descrição

Editar

1. Revisar o código-fonte e corrigir possíveis erros e bugs;
2. Realizar testes de integração para verificar se todas as funcionalidades estão operando corretamente em conjunto;
3. Executar testes de aceitação para validar se o software atende aos requisitos definidos e se atende às expectativas do cliente;
4. Garantir que a interface do usuário seja intuitiva e fácil de usar, realizando testes de usabilidade;
5. Implementar a documentação necessária, incluindo manuais de usuário e guias de instalação;
6. Realizar ajustes finais com base no feedback do cliente e da equipe de teste;
7. Realizar um teste de desempenho para garantir que o software possa lidar com um grande volume de dados e usuários;
8. Certificar-se de que a segurança do software esteja adequada, garantindo que as informações dos usuários estejam protegidas;
9. Realizar uma análise de risco de segurança e implementar medidas de segurança para proteger os dados dos usuários, como backups regulares e criptografia;
10. Realizar o deploy do software em um ambiente de produção, garantindo que todas as configurações estejam corretas e que o software esteja acessível aos usuários finais.

☒ Checklist

Excluir

0%

☐ Revisar o código-fonte e corrigir possíveis erros e bugs;

☐ Realizar testes de integração para verificar se todas as funcionalidades estão operando corretamente em conjunto;

☐ Executar testes de aceitação para validar se o software atende aos requisitos definidos e se atende às expectativas do cliente;

☐ Garantir que a interface do usuário seja intuitiva e fácil de usar, realizando testes de usabilidade;

☐ Implementar a documentação necessária, incluindo manuais de usuário e guias de instalação;

☐ Realizar ajustes finais com base no feedback do cliente e da equipe de teste;

☐ Realizar um teste de desempenho para garantir que o software possa lidar com um grande volume de dados e usuários;

☐ Certificar-se de que a segurança do software esteja adequada, garantindo que as informações dos usuários estejam protegidas;

☐ Realizar uma análise de risco de segurança e implementar medidas de segurança para proteger os dados dos usuários, como backups regulares e criptografia;

☐ Realizar o deploy do software em um ambiente de produção, garantindo que todas as configurações estejam corretas e que o software esteja acessível aos usuários finais.

Adicionar um item

3 - CONCLUSÃO

O método que foi utilizado para a realização deste projeto é o mais indicado em situações como está, prazo curto para a entrega do projeto, com essa metodologia é permitido que seja criadas soluções simples e rápidas para sua conclusão.

Este processo scrum vem se mostrando cada vez mais eficiente para um resultado final do software de gerenciamento de finanças de forma altamente funcional e eficiente e que seja finalizado dentro do prazo estabelecido e um produto de qualidade, atendendo as expectativas do cliente.

REFERÊNCIAS

TRELLO , Disponível em

<<https://trello.com/>> Acesso em 01 de Maio de 2023

Link para acesso ao projeto:

<https://trello.com/b/4nZVjOuJ/projeto-de-gerenciamento-de-finan%C3%A7as>