

Desenvolvimento de Software Multiplataforma

1 SOBRE A DISCIPLINA

- DISCIPLINA: Engenharia de Software II
- PROFESSOR: Braz Izaías da Silva Junior
- OBJETIVO: Apresentação resumida do projeto, para primeira avaliação
- Vertente Tecnológica: Arte digital
- EQUIPE: Ana Karoline Santana, Isabela Dantas Fernandes, Luiz Henrique Santos Almeida.

NOME: Isabela Dantas Fernandes

R.A:

2701392423015

2 SOBRE O TRABALHO

1. Motivação (Por quê?) – 20 linhas

Nosso projeto, intitulado AniMate, surge com o objetivo de revolucionar o processo de animação colaborativa. A ideia principal é permitir que diferentes animadores trabalhem simultaneamente em uma única animação, sem a necessidade de esperar que um termine seu trabalho para que o outro comece. Ao invés disso, a animação ocorre de forma dinâmica na mesma página, mas em diferentes camadas. Cada camada pode ser acessada e editada de maneira independente, o que permite a sobreposição de tarefas e a fluidez do trabalho entre os membros da equipe.

Essa abordagem otimiza o processo criativo, garantindo que o trabalho de todos os envolvidos seja integrado de maneira mais rápida e eficiente. Com a possibilidade de visualizar e alterar as camadas em tempo real, o AniMate elimina as limitações de tempo que surgem quando se trabalha de forma sequencial. Isso também reduz os gargalos, facilitando a comunicação entre os animadores e aumentando a produtividade da equipe.

Além disso, o projeto propõe uma interface intuitiva, que possibilita aos animadores, sejam iniciantes ou profissionais, realizar suas contribuições sem a complexidade de ferramentas tradicionais. Com esse modelo, as animações ganham maior dinamismo e a produção se torna mais ágil, sem comprometer

a qualidade. A ideia é transformar a maneira como as animações são feitas, criando um ambiente colaborativo eficiente e inovador.

2.1 Objetivo (O QUE?) – 20 linhas

O objetivo será criar um aplicativo de animação colaborativa onde os animadores podem se juntar em um arquivo para trabalharem nele simultaneamente, facilitando a visualização do progresso de cada um, agilizando a progressão da animação em si, já que, um animador não precisará esperar o próximo terminar tudo da sua etapa para dar continuação.

Juntamente, através de um design inofensivo e intuitivo, o aplicativo deverá ser a porta de entrada para o aprendizado de novos animadores. A intercalação das duas interfaces distintas irá abranger do público iniciante, ao profissional, tornando a plataforma multifuncional.

O programa deverá ter as seguintes funcionalidades:

A timeline para o controle do pacing do produto final, para um controle máximo da duração e posição dos frames na animação final;

Uma canvas, a área desenhável da tela.

Frames para uma separação nítida de cada elemento da animação como por exemplo uma camada para o fundo, uma para o personagem, uma camada para os efeitos etc. Assim tendo uma boa organização do documento e evitando conflitos;

Ferramentas de desenho, elementos essenciais da aplicação que permitiram os animadores realmente desenharem e editarem seus desenhos conforme o necessário. Essas ferramentas são: Pincel, borracha, laço e manipulação, balde de tinta, seletor de cores, botão de desfazer ação e de refazer ação (cada colaborador só pode desfazer suas próprias ações, para não ter conflitos nos processos de cada um);

Ferramentas do time; Anotações, para sugestões visuais sem realmente editar o desenho de outro, um chat para comunicação interna, conexão em tempo real, controle de permissões para o administrador do arquivo, exportações do arquivo, e independência de cada colaborador (eles podem estar em frames e camadas diferentes).

2.2 Metodologia (COMO?) – 20 Linhas

Para facilitar a visualização do projeto e garantir que todos estejam alinhados com o resultado, iniciaremos com a criação do Wire frame e da prototipação no Figma. Essa etapa inicial nos permitirá ajustar detalhes de layout, navegabilidade e experiência do usuário antes de partirmos para o desenvolvimento propriamente dito.

Na parte da programação, o projeto será desenvolvido para web. O front-end será construído com HTML, CSS e Javascript, utilizando as bibliotecas e APIs necessárias para garantir todas as funcionalidades previstas. O back-end, por sua vez, será feito em C++ com o framework Crow, proporcionando performance e simplicidade. Para armazenar os dados, optamos pelo banco de dados MongoDB, já que os arquivos dos projetos exigem um banco de dados capaz de lidar com os frames, camadas e outros componentes de cada arquivo de animação.

A organização do trabalho seguirá a metodologia ágil SCRUM. Teremos sprints bem definidas, cada uma focada em uma etapa específica do desenvolvimento. Começaremos com o planejamento, passando para o Wire frame, as criações das telas e, por fim, pelos testes e ajustes finais.

Para o controle de versão, utilizaremos o GitHub, garantindo que todas as mudanças fiquem registradas e possam ser acompanhadas por toda a equipe. O Git Bash será a ferramenta principal para gerenciar esses commits e branches, mantendo o fluxo de trabalho organizado. Como IDE principal, escolhemos o Visual Studio Code, que oferece integração fácil com o Git, suporte a múltiplas linguagens e extensões que vão agilizar o nosso processo de desenvolvimento.

2.3 Desenvolvimento (PASSO-A-PASSO) – 20 linhas

1. Planejamento inicial.

Nesta fase, o foco é definir claramente os objetivos e escopo do projeto. A equipe vai alinhar a visão geral, como a ideia de permitir que vários animadores trabalhem simultaneamente em camadas diferentes na mesma animação. Serão discutidas também as ferramentas necessárias, a arquitetura do sistema e como o processo de animação colaborativa será implementado. O planejamento também inclui cronogramas e a alocação de recursos para as etapas seguintes.

2. Primeira entrega da base refinada e Definição dos requisitos.

Nessa etapa, a equipe finaliza a definição das funcionalidades principais e os requisitos técnicos que o AniMate precisa ter. Isso inclui a criação de uma base

sólida para o sistema, onde serão detalhadas as especificações do projeto, como a forma como as camadas de animação interagem entre si e o formato colaborativo. A primeira entrega deve fornecer uma visão clara do que o produto será capaz de fazer, com os requisitos bem definidos para que o desenvolvimento seja eficiente.

3. Pesquisa bibliográfica e levantamento de campo.

Esta etapa envolve uma pesquisa sobre as melhores práticas e ferramentas existentes para animações colaborativas. O levantamento de campo também pode incluir entrevistas com animadores, estudos de outros sistemas semelhantes ou análise de softwares que já oferecem funcionalidades de animação em camadas. A pesquisa ajudará a compreender o que é possível aplicar ao projeto, identificar inovações e buscar soluções que facilitem o desenvolvimento da plataforma.

4. Configuração do ambiente de desenvolvimento e Desenvolvimento Front-End.

O ambiente de desenvolvimento será configurado, com a escolha das ferramentas e frameworks mais adequados para o projeto. Em seguida, a equipe inicia o desenvolvimento do front-end, a parte do sistema com a qual os usuários interagirão diretamente. Isso inclui a criação da interface onde os animadores poderão trabalhar nas camadas de animação, ver as alterações em tempo real e navegar facilmente pelo sistema. O design da interface será crucial para a usabilidade do Animate.

5. Desenvolvimento do back-end.

O back-end envolve o desenvolvimento da parte do sistema que lida com o processamento de dados, servidores e lógica de negócios. Para o Animate, isso inclui a criação das funções que suportarão as animações em camadas, como armazenamento de dados, manipulação de animações e a comunicação entre os usuários. O back-end também deve garantir que as camadas possam ser editadas simultaneamente, sem conflitos, e que os dados sejam salvos de forma eficiente e segura.

6. Integração do back-end e front-end.

Após o desenvolvimento do front-end e back-end, a integração entre as duas partes será realizada. Isso significa garantir que a interface que os animadores usarão se comunique corretamente com o sistema que processa os dados. Essa etapa pode envolver ajustes de desempenho e resolver problemas de comunicação entre o cliente (front-end) e o servidor (back-end), garantindo que a animação colaborativa funcione de maneira fluida.

7. Modelagem e implementação do banco de dados.

O banco de dados será modelado para armazenar informações sobre as animações criadas pelos usuários, como as camadas e os dados das interações. Ele precisa ser eficiente, capaz de gerenciar grandes volumes de

dados e suportar acessos simultâneos de vários animadores. Durante esta fase, serão definidas as tabelas e a estrutura que garantem a integridade e a rapidez nas operações.

8. Teste de funcionalidades.

A equipe testará as funcionalidades do Animate para verificar se tudo está funcionando conforme o esperado. Isso envolve testar as interações entre as camadas de animação, a sincronização das edições feitas por diferentes animadores e a estabilidade do sistema. O objetivo é identificar erros e corrigir qualquer problema antes da fase de usabilidade.

9. Teste de usabilidade com alunos.

Durante esta fase, o sistema será testado por alunos ou usuários reais para avaliar como eles interagem com a plataforma. O feedback será crucial para entender se a interface é intuitiva e se a experiência de animação colaborativa é fluida. Serão feitas observações sobre como melhorar o design e a funcionalidade com base no uso real.

10. Implementação de avaliações.

Nessa etapa, o sistema pode incluir funcionalidades para avaliar o desempenho dos animadores, ou até mesmo o processo de animação colaborativa. A equipe pode implementar uma forma de monitorar a eficácia do trabalho em equipe ou permitir que os usuários avaliem o desempenho uns dos outros, promovendo uma melhoria contínua.

11. Ajuste e melhorias com base no feedback.

Após os testes de usabilidade, ajustes serão feitos para resolver qualquer problema identificado pelos usuários. A equipe pode melhorar a interface, corrigir falhas no sistema e otimizar o desempenho com base no feedback recebido. Isso inclui tanto ajustes estéticos quanto melhorias funcionais.

12. Otimização de desempenho.

Finalmente, a última etapa foca na otimização do sistema para garantir que o Animate seja eficiente e rápido, especialmente quando muitos animadores estiverem trabalhando ao mesmo tempo. A performance do sistema será otimizada para reduzir o tempo de resposta, melhorar a estabilidade e garantir que a plataforma possa suportar múltiplos usuários sem comprometer a qualidade da experiência.

Essa abordagem permite que o Animate evolua de maneira organizada, com uma base sólida de planejamento e execução, garantindo que as funcionalidades sejam bem implementadas e testadas ao longo do processo.

2.4 Resultados esperados (O QUE VOCÊ ESPERA ALCANÇAR) – 20 linhas

Ao término do projeto, espera-se que diversos conhecimentos teóricos sejam aplicados na prática, proporcionando uma compreensão mais aprofundada dos conceitos abordados ao longo do desenvolvimento. A proposta inclui a criação de uma plataforma dinâmica utilizando HTML, CSS e outras tecnologias complementares, garantindo que os recursos empregados atendam plenamente às necessidades do *AniMate*. Além disso, busca-se implementar um sistema seguro e eficiente para o armazenamento contínuo das artes produzidas e dos dados dos usuários, assegurando a integridade e acessibilidade das informações.

Outro aspecto relevante do projeto envolve a criação de ferramentas voltadas à produção de animações, permitindo que os usuários tenham acesso a funcionalidades específicas para facilitar o desenvolvimento de seus trabalhos. Esse processo contribuirá não apenas para a experiência do usuário, mas também para o aprimoramento das habilidades técnicas envolvidas na concepção e implementação dessas soluções.

A conclusão do projeto proporcionará maior experiência em todas as etapas do desenvolvimento, desde a estruturação inicial até a implementação e testes da aplicação. Além disso, espera-se o aperfeiçoamento das habilidades essenciais para a atuação na área, como a elaboração colaborativa, a capacidade de adaptação a desafios técnicos e a constante revisão e aprimoramento das ideias ao longo do processo.

Por fim, a familiarização com uma nova linguagem de programação e a implementação de bibliotecas específicas contribuirão significativamente para o desenvolvimento técnico da equipe, ampliando a bagagem prática e teórica adquirida ao longo do projeto. Esses conhecimentos serão fundamentais para a construção de soluções inovadoras e para a inserção no mercado de tecnologia e desenvolvimento de software.

Na parte financeira, os planos de vendas seriam em diferentes tipos de pacotes. Os possíveis planos sendo: um plano estudantil que pode ser vendido como uma ferramenta de aprendizado a instituições de ensino ou empresas que possuem parcerias com instituições de ensino que buscam iniciar novos talentos na área; um plano para equipes independentes que buscam uma forma de facilitar o seu trabalho em equipe nos projetos e um plano para empresas em si.

2. Cronograma (PASSO-A-PASSO + TEMPO) – 20 linhas

Planejamento inicial – 12/02 - 19/02

Primeira entrega da base refinada e Definição dos requisitos – 26/02 - 19/03

Pesquisa bibliográfica e levantamento de campo – 23/04 - 30/04

Configuração do ambiente de desenvolvimento e Desenvolvimento front-end – 30/04 - 07/05

Desenvolvimento do back-end – 07/05 - 14/05

Integração do back-end e front-end – 14/05 - 21/05

Modelagem e implementação do banco de dados – 21/05 - 28/05

Teste de funcionalidades – 28/05 - 04/06

Teste de usabilidade com alunos – 04/06 - 11/06

Implementação de avaliações - 11/06 - 18/06

Ajuste e melhorias com base no feedback – 18/06 - 25/06

Otimização de desempenho – 25/06 - 02/07

2.5 Referências bibliográficas (de onde tivemos nossa inspiração) – 20 linhas

A ideia inicial do *AniMate* surgiu da necessidade de uma plataforma que reunisse todos os instrumentos considerados essenciais para um ambiente de criação ideal e completo. Durante o desenvolvimento, foram identificadas algumas alternativas que serviram de referência para a concepção do projeto, possibilitando a análise de possíveis melhorias.

MAGMA. Magma, 2025. Disponível em: <<https://magma.com/>>

Acesso em: 18 fev. 2025.

FLIPACLIP. FlipaClip, 2025. Disponível em: <<https://flipaclip.com/pt/>>

Acesso em: 18 fev. 2025.

GARTIC. Gartic, 2025. Disponível em: <<https://gartic.com.br/>>

Acesso em: 18 fev. 2025.

CANVA. Canva, 2025. Disponível em: <<https://www.canva.com/>>

Acesso em: 18 fev. 2025.

GOOGLE DOCS. Google, 2025. Disponível em:

<<https://docs.google.com/document/u/0/?hl=pt-BR>>

Acesso em: 18 fev. 2025.

CRUNCHYROLL: Inside Anime. Behind the Scenes of Dr. STONE | The Making of an Anime. YouTube, 13 de setembro de 2019. Disponível em:

< <https://www.youtube.com/watch?v=ykEp0wlchiY> >

Acesso em: 18 fev. 2025.

Riyan. Making of little Witch Academia – Studio Trigger. Youtube, 8 de dezembro de 2015. Disponível em:

< <https://www.youtube.com/watch?v=Ba55wWnEn3s> >

Acesso em: 18 fev. 2025.

--