ESCOLA SENAI

“PROF. DR. EURYCLIDES DE JESUS ZERBINI”

Ana Luiza Cardoso Podavi

Felipe Macedo Dos Santos

Matheus Lena Barros

Sérgio Borges Dos Santos Filho

**Yourself**

Campinas, SP

2024

Ana Luiza Cardoso Podavi

Felipe Macedo Dos Santos

Matheus Lena Barros

Sérgio Borges Dos Santos Filho

**Yourself**

Projeto apresentado à Escola SENAI “Prof. Dr. Euryclides de Jesus Zerbini” para obtenção do certificado de conclusão do Curso Técnico de Desenvolvimento de Sistemas.

Orientadores: Paulo Henrique Pansani e Renato de Mattos Onofre

Campinas, SP

2024

Ana Luiza Cardoso Podavi

Felipe Macedo Dos Santos

Matheus Lena Barros

Sérgio Borges Dos Santos Filho

**Yourself**

Trabalho de conclusão de curso aprovado como requisito parcial para obtenção do grau de técnico, do curso Técnico de Desenvolvimento de Sistemas da Escola SENAI “Prof. Dr. Euryclides de Jesus Zerbini”.

**BANCA EXAMINADORA**

1º Examinador

2º Examinador

3º Examinador

Campinas, SP

2024

**DEDICATÓRIA**

Dedicamos esse projeto a todos os nossos conhecidos que possuem dificuldades em organizar suas vidas acadêmicas, tais quais nos inspiraram a criar uma solução adequada que, além de auxiliar nesse quesito, aprimora a documentação do tempo de concentração e descanso.

**AGRADECIMENTOS**

Agradecemos imensamente aos nossos orientadores, Paulo e Renato, pela confiança depositada em nós, pela orientação paciente e pela dedicação em compartilhar seus conhecimentos. Sua expertise foi fundamental para o sucesso deste trabalho.

**EPÍGRAFE**

**“Se você fizer aquilo que a sua intuição manda,**

**usando bom senso e deixando de lado a vaidade, você**

**terá todas as possibilidades de alcançar seu objetivo”.**  
**Silvio Santos**

**SUMÁRIO**

[1 INTRODUÇÃO 9](#_Toc184215710)

[2 JUSTIFICATIVA 11](#_Toc184215711)

[3 OBJETIVOS 12](#_Toc184215712)

[3.1. Objetivos Gerais 12](#_Toc184215713)

[3.2. Objetivos Específicos 12](#_Toc184215714)

[4 PRODUCT BACKLOG 13](#_Toc184215715)

[5 REQUISITOS NÃO FUNCIONAIS 14](#_Toc184215716)

[6 PREMISSAS 15](#_Toc184215717)

[7 RESTRIÇÕES 16](#_Toc184215718)

[7.1. RESTRIÇÕES PARA FUNCIONAMENTO 16](#_Toc184215719)

[8 ANÁLISE DE RISCOS DE UM PROJETO 17](#_Toc184215720)

[8.1. Nível e Planos de Ação para os Riscos 17](#_Toc184215721)

[8.2. Planos de ação 17](#_Toc184215722)

[9 SPRINTS 19](#_Toc184215723)

[9.1. Primeiro Sprint 19](#_Toc184215724)

[9.1.1. Product Backlog 20](#_Toc184215725)

[9.1.2. Sprint Backlog 20](#_Toc184215726)

[9.1.3. Burn Down Chart 21](#_Toc184215727)

[9.1.4. Diagramas 21](#_Toc184215728)

[9.1.5. Kanban e Retrospectiva 33](#_Toc184215729)

[9.2 Segundo Sprint 39](#_Toc184215730)

[9.2.1 Product Backlog 40](#_Toc184215731)

[9.2.2 Kanban e Retrospectiva 41](#_Toc184215732)

[9.3 Terceiro Sprint 48](#_Toc184215733)

[9.3.1 Product Backlog 49](#_Toc184215734)

[9.3.2 Sprint Backlog 49](#_Toc184215735)

[9.3.3 Plano de Testes 50](#_Toc184215736)

[9.3.4 Kanban e Retrospectiva 53](#_Toc184215737)

[9.4 Quarto Sprint 59](#_Toc184215738)

[9.4.1 Product Backlog 59](#_Toc184215739)

[9.4.2 Sprint Backlog 59](#_Toc184215740)

[9.4.3 Burn Down Chart 59](#_Toc184215741)

[9.4.4 Diagramas 60](#_Toc184215742)

[9.4.5 Kanban e Retrospectiva 60](#_Toc184215743)

[10 Modelo de Dados 61](#_Toc184215744)

[10.1. Diagrama de Entidade e Relacionamento 62](#_Toc184215745)

[10.2. Modelo de Dados do Banco 62](#_Toc184215746)

[10.3. Dicionário de dados 64](#_Toc184215747)

[11 PRINCIPAIS TELAS DO SISTEMA 64](#_Toc184215748)

[12 CONCLUSÃO 72](#_Toc184215749)

[12.1. Escreva os resultados obtidos 72](#_Toc184215750)

[12.2. Constatações 72](#_Toc184215751)

[12.3. Sugestões de possíveis aperfeiçoamentos técnicos 72](#_Toc184215752)

[13 REFERÊNCIAS 73](#_Toc184215753)

[14 GLOSSÁRIO 74](#_Toc184215754)

# INTRODUÇÃO

De acordo com Francesco Cirillo, criador do método Pomodoro, ressalta ser essencial que, mesmo que você não consiga terminar a tarefa em 25 minutos, tire o tempo de descanso — durante esse período, não faça nada que requeira muito esforço mental. Você pode comer ou beber um copo d'água, ou assistir à TV.

A crescente complexidade da vida moderna exige cada vez mais organização e produtividade. Diante desse cenário, a busca por ferramentas e técnicas que auxiliem na gestão do tempo e das tarefas tornou-se uma necessidade.

A gamificação, que consiste na aplicação de elementos de jogos em contextos não lúdicos, emerge como uma promissora alternativa para tornar as atividades cotidianas mais engajadoras e motivacionais. Este trabalho tem como objetivo investigar a potencialidade da gamificação como ferramenta para auxiliar na organização das tarefas diárias, explorando o desenvolvimento e a avaliação de um sistema gamificado que incentive a criação de hábitos mais eficientes e produtivos.

Portanto, a Yourself tem como foco solucionar as problemáticas apresentadas acima, integrando diversas funcionalidades necessárias e não exploradas por outros aplicativos de estudo. Dessa maneira, os estudantes poderão aprender mais facilmente.

A gestão eficaz do tempo e a otimização da produtividade são desafios cada vez mais presentes no cotidiano das pessoas. A técnica Pomodoro, por exemplo, popularizada por Francesco Cirillo, demonstra a importância de intervalos regulares para manter a concentração e evitar a fadiga mental. No entanto, a aplicação de técnicas como essa pode ser desafiadora, especialmente em um mundo repleto de distrações.

A gamificação, ao incorporar elementos lúdicos em contextos não-lúdicos, emerge como uma estratégia promissora para aumentar a motivação e o engajamento dos usuários. Ao transformar tarefas cotidianas em desafios e jogos, a gamificação pode tornar a organização pessoal mais divertida e eficaz.

Este trabalho tem como objetivo desenvolver e avaliar um sistema gamificado, denominado Yourself, que visa auxiliar os usuários a gerenciar suas tarefas diárias de forma mais eficiente. Diferentemente de outros aplicativos, o Yourself se destaca pela sua capacidade de se adaptar às necessidades individuais de cada usuário, oferecendo um plano de estudos personalizado e recursos interativos que tornam o processo de aprendizagem mais prazeroso. Ao integrar elementos da técnica Pomodoro com mecânicas de gamificação, o Yourself busca proporcionar uma experiência mais completa e personalizada, auxiliando os usuários a alcançar seus objetivos de forma mais consistente e satisfatória.

# JUSTIFICATIVA

O crescente desafio enfrentado por estudantes de diversas idades em manter a concentração e a produtividade durante os estudos têm sido amplamente reconhecidos. Com a expansão da educação digital e a multiplicidade de distrações oferecidas pela tecnologia, torna-se cada vez mais difícil para os alunos gerenciarem seu tempo de forma eficaz. Neste contexto, o método Pomodoro surge como uma técnica amplamente reconhecida e eficaz para melhorar a concentração e a eficiência nos estudos, ao dividir o tempo de estudo em intervalos curtos e controlados, seguidos de breves pausas.

O aplicativo "Yourself" se propõe a incorporar essa técnica dentro de uma plataforma acessível e intuitiva, oferecendo suporte aos alunos que buscam maximizar seu rendimento acadêmico. Ao adotar o método Pomodoro, o aplicativo visa auxiliar os usuários a desenvolverem melhores hábitos de estudo, reduzir a procrastinação e, consequentemente, melhorar o desempenho escolar.

Além disso, o "Yourself" visa criar uma comunidade de apoio, onde os alunos podem compartilhar suas rotinas, dicas e progressos, promovendo um senso de pertencimento e motivação entre os usuários. Essa comunidade pode servir como uma rede de suporte mútua, onde os estudantes se incentivam e aprendem uns com os outros, tornando o processo de aprendizado mais colaborativo.

Por fim, a gamificação, que recompensa os usuários à medida que atingem metas de estudo, torna o processo mais dinâmico e envolvente. Esse aspecto lúdico pode ser especialmente eficaz para alunos mais jovens, ajudando-os a manter o interesse e o comprometimento com seus estudos.

A implementação desse aplicativo é, portanto, justificada pela necessidade de ferramentas educacionais que combinem eficiência e simplicidade, promovendo um ambiente de estudo mais focado e produtivo.

# OBJETIVOS

Segue abaixo os objetivos do projeto:

# Objetivos Gerais

Criar o projeto o Yourself e desenvolvê-lo com o objetivo de auxiliar as pessoas a se concentrarem nos seus estudos, utilizando a técnica Pomodoro, permitindo que elas tenham mais objetividade e foco na hora de estudar, com a opção de criar uma tarefa e escolher o tempo em que irá realizá-la e o tempo de descanso.

# Objetivos Específicos

* Produzir a documentação em formato ABNT;
* Criar uma aplicação mobile utilizando React Native;
* Utilizar a metodologia Scrum durante o projeto;
* Desenvolver o método de estudo dos usuários;
* Criar o manual do usuário;
* Realizar testes unitários;
* Criar o sistema de armazenamento da aplicação utilizando o banco de dados não relacional Firebase;
* Desenvolver um sistema de gamificação para motivar os usuários a atingir suas metas;
* Avaliar a interface e a experiência do usuário durante o desenvolvimento para identificar e corrigir possíveis problemas;
* Elaborar esboços da interface do usuário para visualizar a estrutura e o layout da aplicação.

# PRODUCT BACKLOG

RF01 - O usuário deve poder se cadastrar, informando email, nome, nome de usuário e senha.

RF02 - O usuário deve poder entrar na plataforma com seu email e senha cadastrados anteriormente.

RF03 - O sistema deve manter uma sessão de usuário ativa por um período de 3 horas após a última interação do usuário.

RF04 - O usuário deve poder cadastrar uma atividade, informando título, nível de dificuldade e categoria.

RF05 - O usuário deve ser capaz de, a qualquer momento, a partir de iniciada, independentemente do temporizador, finalizar uma atividade.

RF06 - O usuário deve poder pausar e despausar o temporizador acoplado à atividade.

RF07 - O usuário deve poder acessar o histórico de suas atividades nos últimos 2 meses.

RF08 - O usuário deve poder editar o tempo de concentração e descanso do temporizador acoplado às atividades.

RF09 O usuário deve ser capaz de excluir uma atividade criada anteriormente.

RF10: Ao finalizar uma atividade, o usuário deve ganhar pontos, os quais serão calculados a partir dos minutos de concentração que foram levados para concluir a tarefa multiplicados pelo nível de dificuldade estabelecido na criação da mesma. A pontuação máxima por atividade deve ser de 450 pontos.

RF11: O usuário deve poder gastar seus pontos adquiridos por meio de atividades em itens colecionáveis na loja do aplicativo. A partir do momento em que um item é comprado, este ficará disponível na tela de perfil do usuário, sendo exibido no formato de grade.

RF12: O usuário deve ter acesso às suas informações cadastradas anteriormente e poder, a partir de verificação de identidade, editá-las.

RF13: O sistema deve permitir que o usuário visualize dados específicos sobre seu desempenho: o total de pontos acumulados e a maior quantidade de dias consecutivos em que pelo menos uma atividade foi realizada.

# REQUISITOS NÃO FUNCIONAIS

RNF01 - O sistema deve ser intuitivo.

RNF02 - O sistema deve possuir um design atrativo.

RNF03 - As respostas das requisições à API devem durar no máximo 300 milissegundos.

RNF04 - O sistema deve apresentar ao usuário um contador visualizando a quantidade de dias consecutivos em que pelo menos uma atividade foi realizada.

RNF05 - O sistema deve apresentar a quantidade atual de pontos do usuário.

RNF06 - O aplicativo deve possuir um guia para o usuário, incluindo um tutorial inicial do uso do sistema.

# PREMISSAS

PRE01 - O projeto será planejado, desenvolvido e testado por 4 alunos, utilizando a metodologia ágil Scrum.

PRE02 - Serão disponibilizados, no mínimo, 4 computadores para o desenvolvimento do projeto.

PRE03 - O desenvolvimento do aplicativo será realizado utilizando React Native, Firebase como banco de dados e Node.js para a criação da API.

PRE04 - O método Pomodoro será uma das principais técnicas implementadas para auxiliar os usuários a manterem o foco durante os estudos.

PRE05 - A interface do aplicativo será avaliada durante o desenvolvimento para garantir uma experiência intuitiva e de fácil utilização.

PRE06 - O sistema de gamificação será desenvolvido para motivar os usuários a atingirem suas metas de estudo de forma eficiente e lúdica.

# RESTRIÇÕES

RES01 - O projeto deverá ser entregue em 4 meses.

RES02 - O aplicativo deve atender aos padrões de qualidade definidos pela equipe.

RES03 - O aplicativo será desenvolvido para dispositivos android.

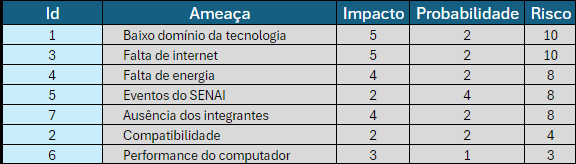
RES04 - O aplicativo é destinado a estudantes universitários que buscam organizar e documentar seus estudos.

RES05 - Apenas 4 integrantes trabalharão no projeto.

# RESTRIÇÕES PARA FUNCIONAMENTO

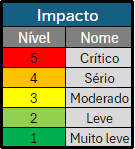
* Software do aplicativo: android 7.0 ou superior;
* Armazenamento: 2 GB ou mais;
* Memória RAM: 2 GB ou mais.

# ANÁLISE DE RISCOS DE UM PROJETO



# 

# Nível e Planos de Ação para os Riscos



# Planos de ação

ID1: Baixo domínio da tecnologia: Isso se resolve estudando e conhecendo mais sobre os assuntos dados ao semestre.

ID2: Compatibilidade: Isso se resolve aplicando o projeto com as mesmas especificações nas máquinas em que serão testadas.

ID3: Falta de internet: Isso é complicado, pois nunca se sabe quando pode acontecer. Mas, se acontecer, é só esperar ou ver no celular vídeos sobre o projeto com dados móveis.

ID4: Falta de energia: A mesma coisa da de cima. Não é possível prever isso, e se acontecer, espera ou vê no celular coisas relacionadas ao projeto.

ID5: Eventos do Senai: Se organize, saiba quando acontecerá.

ID6: Ausência dos integrantes: Você pode avisar o parceiro que faltou para continuar o trabalho em casa.

ID7: Performance do computador: Um computador lento ou com problemas de desempenho pode impactar significativamente a produtividade do grupo. Por isso, você deve apresentar os problemas para os professores ou a direção.

# SPRINTS

Apresenta-se nesta seção a documentação referente ao desenvolvimento do software.

# Primeiro Sprint

A Arch-tech estruturou a lógica dos sprints com base nas horas diárias de trabalho disponíveis, que, neste caso, foram 3 horas por dia. Também foi levada em conta a quantidade de membros da equipe, que era composta por 4 pessoas. Ao multiplicar esses valores, obteve-se o tempo total de trabalho diário, somando 12 horas.

• 3 horas x 4 integrantes = 12 horas diárias

Com esse total, foi calculada a quantidade de dias que o sprint teria, desconsiderando feriados, finais de semana e as aulas de Programação de Dispositivos Móveis. Assim, foi determinado um total de 144 horas de trabalho disponíveis para o primeiro sprint.

• 20 dias úteis → 240 horas no Sprint 1

Com base nisso, estabeleceu-se uma meta de 120 horas de trabalho, reservando 24 horas (16% do sprint) como margem para eventuais contratempos e atrasos. Para atingir essa meta, seriam necessárias aproximadamente 10 horas de trabalho diárias.

**Metas do Primeiro Sprint**

Para o Sprint 1, o grupo definiu as seguintes metas:

1. **Design das Telas:**
   * O design das telas seria feito no Figma tanto para Android quanto para web. As telas planejadas incluíam:
     + Tela inicial
     + Tela de login/cadastro
     + Tela home (cadernos)
     + Tela Tarefas
     + Tela de Shop
     + Tela de Profile
2. **Desenvolvimento Web:**
   * Foi planejado codificar todas as telas para a versão web no Visual Code e também desenvolver o backend das mesmas.
3. **Desenvolvimento Android:**
   * Para o Android, o foco foi o desenvolvimento do front-end das telas e a implementação de validações, utilizando o Visual Code.
4. **Banco de Dados:**
   * O planejamento do banco de dados incluiu a criação do modelo conceitual e lógico.
   * Desenvolvimento do dicionário de dados.
   * Implementação dos dados por meio de scripts no MySQL Workbench.

.

# Product Backlog

Não houve alterações no Product Backlog.

# Sprint Backlog

RF01 - O usuário deve poder se cadastrar, informando e-mail, nome, nome de usuário e senha.

RF04 - O usuário deve poder cadastrar uma atividade, informando título, nível de dificuldade e categoria.

# Burn Down Chart

# Diagramas

Apresenta os principais diagramas realizados.



|  |
| --- |
| **NARRATIVA DE CASO DE USO** |
| **Sigla/Nome:**  CSUO1, Cadastrar-se**.**  **Objetivo:** Permitir um registro de um usuário no aplicativo.  **Frequência estimada:** Sempre que um novo usuário querer entrar no aplicativo.  **Ator Principal:** Usuário. |
| **CENARIO PRINCIPAL** |
| 1. O caso de uso começa quando o usuário acessa o aplicativo e escolhe a opção cadastrar. 2. O aplicativo exibe a tela que comporta nome de usuário, e-mail, senha e confirmar senha. 3. O usuário preenche os campos com seus dados. 4. O aplicativo valida os dados para registrá-lo no banco de dados. |
| **CENÁRIO ALTERNATIVO** |
| 1. Caso o usuário não preencha todos os campos obrigatórios (nome de usuário, e-mail, senha e confirmar senha), o sistema exibirá uma mensagem informando que todos os campos são obrigatórios para o cadastro. |
| **CENÁRIO EXCESSÃO** |
| 1. **Dados de usuário já cadastrados.** 2. Caso haja no sistema um cadastro já existente com os dados de cadastro do novo usuário, ele será informado de que o e-mail ou o nome de usuário já estão cadastrados. 3. Não há restrições quanto a quantidade de caracteres na senha. |
| **PRÉ-CONDIÇÕES** |
| 1. O usuário deve ter um e-mail para se cadastrar. 2. O usuário deve criar uma senha com no mínimo uma letra maiúscula e um caractere especial. 3. O usuário deverá fornecer um e-mail que não tenha sido cadastrado anteriormente. |
| **PÓS-CONDIÇÕES** |
| 1. Se o cadastro for realizado com sucesso, o sistema salvará os dados do usuário no banco. 2. Ao clicar no botão de cadastrar, o usuário será redirecionado para a tela de login. |

|  |
| --- |
| **NARRATIVA DE CASO DE USO** |
| **Sigla/Nome:**  CSU03 – Criar Atividade  **Objetivo:**  o objetivo principal é oferecer uma funcionalidade que permita ao usuário organizar seu tempo de trabalho, facilitando a criação de tarefas que serão acompanhadas por ciclos de trabalho e descanso, ajudando a evitar distrações e melhorar a gestão de tempo.  **Frequência estimada:** Sempre que um novo usuário querer entra no aplicativo.  **Ator Principal:** Usuário. |
| **CENARIO PRINCIPAL** |
| 1. **Início do caso de uso:**  * O usuário abre o aplicativo Pomodoro e navega até a tela de criação de atividade.  1. **Tela de criação de atividade:**  * O aplicativo exibe um formulário para o usuário inserir os detalhes da nova atividade.   + Campos disponíveis:     - Título da atividade     - Descrição da atividade (opcional)     - Número de ciclos Pomodoro desejados (padrão é 4 ciclos)     - Tempo de cada ciclo (padrão 25 minutos)     - Intervalo curto (padrão 5 minutos)     - Intervalo longo (padrão 15 minutos após 4 ciclos)  1. **Preenchimento dos dados:**  * O usuário preenche o título da atividade (obrigatório) e, se desejar, uma breve descrição. * O usuário ajusta a quantidade de ciclos ou tempos, se necessário, usando sliders ou campos numéricos.  1. **Validação de dados:**  * O sistema valida se os campos obrigatórios estão preenchidos corretamente. * Caso haja algum erro (por exemplo, título em branco), o sistema exibe uma mensagem de alerta solicitando a correção.  1. **Confirmação:**  * O usuário pressiona o botão “Salvar” para confirmar a criação da atividade.  1. **Gravação da atividade:**  * O sistema grava a nova atividade no banco de dados, associada à conta do usuário.  1. **Feedback:**  * O sistema exibe uma mensagem confirmando que a atividade foi criada com sucesso e já está disponível na lista de atividades do usuário.  1. **Fim do caso de uso:**  * O usuário pode escolher iniciar a atividade imediatamente ou voltar para a lista de atividades. |
| **CENÁRIO ALTERNATIVO** |
| 1. **Fluxo Alternativo 1 – Dados incompletos:** 2. Se o usuário tentar salvar a atividade sem preencher os campos obrigatórios, o sistema exibe uma mensagem de erro e solicita o preenchimento correto dos campos. 3. **Fluxo Alternativo 2 – Cancelamento da criação:** 4. O usuário pode optar por cancelar a criação da atividade a qualquer momento antes de salvar, retornando à tela anterior sem que os dados sejam gravados**.** |
| **CENÁRIO EXCESSÃO** |
| 1. **Falha na validação de dados obrigatórios**    * O sistema detecta que campos obrigatórios, como "Título", estão vazios ou preenchidos incorretamente.    * O sistema exibe uma mensagem de erro, solicitando o preenchimento correto.    * O usuário corrige os dados e tenta novamente. 2. **Formato inválido de dados (ex: tempo de ciclos)**    * O sistema identifica que o valor inserido para o tempo de ciclos não é válido (ex: número negativo).    * Exibe um alerta, pedindo a inserção de um valor válido.    * O usuário corrige e tenta salvar novamente. 3. **Falha na conexão com o banco de dados**    * O sistema não consegue acessar o banco de dados para salvar a atividade.    * O sistema exibe uma mensagem de erro e tenta reconectar automaticamente ou solicita ao usuário tentar mais tarde.    * O usuário pode salvar localmente e tentar novamente quando a conexão for restabelecida. 4. **Sessão expirada**    * O sistema detecta que a sessão do usuário expirou.    * Exibe uma mensagem pedindo que o usuário faça login novamente.    * Após o login, o usuário pode retornar ao ponto onde estava, se os dados foram armazenados temporariamente. 5. **Cancelamento pelo usuário**    * O usuário decide cancelar a criação da atividade.    * O sistema exibe uma caixa de confirmação para confirmar o cancelamento.    * Se confirmado, o sistema descarta os dados e retorna à tela principal. |
| **PRÉ-CONDIÇÕES** |
| 1. O usuário deve estar autenticado no aplicativo. 2. O aplicativo deve estar conectado ao banco de dados. |
| **PÓS-CONDIÇÕES** |
| 1. Uma nova atividade é criada e armazenada no banco de dados. 2. O usuário pode iniciar a contagem de tempo da atividade. |

|  |
| --- |
| **NARRATIVA DE CASO DE USO** |
| **Sigla/Nome:**  CSU02-Login**.**  **Objetivo:** Permitir que um usuário entre no site.  **Frequência estimada:** Todas as vezes que for necessário acessar o sistema.  **Ator Principal:** Usuário. |
| **CENARIO PRINCIPAL** |
| 1. **Abertura do Aplicativo:** O caso de uso inicia quando o usuário abre o aplicativo de Pomodoro. 2. **Tela de Login:** O sistema exibe a tela de login, contendo campos para inserção de **e-mail** e **senha**, além de um botão para **login**. 3. **Inserção das Credenciais:** O usuário insere o e-mail e a senha cadastrados no aplicativo. 4. **Submissão:** O usuário clica no botão de "Login". 5. **Validação:** O sistema de autenticação verifica se as credenciais inseridas são válidas. 6. **Acesso Autorizado:** Com as credenciais validadas, o sistema autoriza o login e carrega o ambiente personalizado do usuário, que inclui suas configurações de Pomodoro e o histórico de sessões anteriores. 7. **Redirecionamento:** O usuário é redirecionado para a tela principal do aplicativo, onde ele pode iniciar ou configurar uma sessão de Pomodoro. |
| **CENÁRIO ALTERNATIVO** |
| **Recuperação de Senha:**   1. Caso o usuário tenha esquecido a senha, ele clica na opção "Esqueci minha senha". 2. O sistema solicita o e-mail cadastrado. 3. O sistema envia um e-mail com instruções para redefinir a senha. 4. Após redefinir a senha, o usuário pode realizar o login normalmente. |
| **CENÁRIO EXCESSÃO** |
| 1. **Inserção de Credenciais:** O usuário insere um e-mail e/ou senha que não correspondem aos dados cadastrados. 2. **Tentativa de Autenticação:** O usuário clica no botão de login, e o sistema tenta validar as credenciais. 3. **Erro de Autenticação:** O sistema detecta que o e-mail ou senha não são válidos. 4. O sistema exibe uma mensagem de erro: "E-mail ou senha incorretos. Tente novamente." 5. **Novo Login:** O usuário retorna à tela de login, podendo tentar novamente com credenciais corretas. |
| **PRÉ-CONDIÇÕES** |
| 1. O usuário já criou uma conta no aplicativo de Pomodoro. 2. O usuário possui conexão à internet. |
| **PÓS-CONDIÇÕES** |
| 1. O usuário está autenticado no sistema e pode utilizar o aplicativo para gerenciar suas sessões de Pomodoro. |

|  |
| --- |
| **NARRATIVA DE CASO DE USO** |
| **Sigla/Nome:**  CSUO5, Consultar atividades.  **Objetivo:** Permitir que um usuário consulte suas atividades.  **Frequência estimada:** Sempre que um usuário finalizar uma atividade.  **Ator Principal:** Usuário. |
| **CENARIO PRINCIPAL** |
| 1. O caso de uso começa quando o usuário acessa o histórico de atividades. 2. O aplicativo exibe na tela as últimas atividades finalizadas pelo usuário. 3. Caso o usuário clique em ver mais, mostrará todas as atividades realizadas por ele(a). |
| **CENÁRIO ALTERNATIVO** |
| 1. Caso o usuário entre no histórico sem estar cadastrado, aparecerá uma mensagem informado que é necessário login para ver o histórico. |
| **CENÁRIO EXCESSÃO** |
| 1. **Nenhuma atividade cadastrada.** 2. Caso o usuário não tenha nenhuma atividade realizada, aparecerá uma mensagem informando que não há atividades realizadas. 3. Não há restrições quanto a quantidade de atividades que possam ser realizadas. |
| **PRÉ-CONDIÇÕES** |
| 1. O usuário deve estar logado. 2. O usuário deve ter realizado pelo menos uma atividade. |
| **PÓS-CONDIÇÕES** |
| 1. As atividades permanecem inalteradas, sem impacto em suas propriedades ou status. |

|  |
| --- |
| **NARRATIVA DE CASO DE USO** |
| **Sigla/Nome:**  CSU06 – Excluir Atividade  **Objetivo:**  Permitir que o usuário exclua a atividade caso não for o que esperava  **Frequência estimada:** Sempre que um novo usuário querer entra no aplicativo.  **Ator Principal:** Usuário. |
| **CENARIO PRINCIPAL** |
| 1. **Acesso à Lista de Atividades:** O usuário abre o aplicativo Pomodoro e navega até a seção de atividades, onde visualiza uma lista de suas tarefas programadas. 2. **Seleção da Atividade:** O usuário identifica uma atividade que deseja excluir. Essa atividade pode ser uma tarefa que já foi concluída ou que não é mais relevante. 3. **Ação de Exclusão:** O usuário toca no ícone de "excluir" (geralmente representado por uma lixeira) ao lado da atividade selecionada. 4. **Confirmação de Exclusão:** O aplicativo exibe uma janela de confirmação perguntando se o usuário tem certeza de que deseja excluir a atividade. O usuário pode optar por "Sim" ou "Não".   Se o usuário escolher "Sim":   * A atividade é removida da lista. * O aplicativo exibe uma mensagem de sucesso confirmando a exclusão. * A lista de atividades é atualizada automaticamente para refletir a mudança. * Se o usuário escolher "Não": * A janela de confirmação é fechada e o usuário retorna à lista de atividades sem alterações.  1. **Feedback:** O usuário recebe um feedback visual (como uma notificação) indicando que a atividade foi excluída com sucesso, reforçando a sensação de controle sobre suas tarefas. |
| **CENÁRIO ALTERNATIVO** |
| 1. **Atividade Não Encontrada:** Caso o usuário tente excluir uma atividade que não existe (por exemplo, já foi excluída anteriormente), o aplicativo informa que a atividade não foi encontrada e sugere a atualização da lista. |
| **CENÁRIO EXCESSÃO** |
| 1. **Atividade Não Encontrada:**  * Se o usuário tenta excluir uma atividade que já foi removida ou não existe, o aplicativo exibe uma mensagem de erro: "Atividade não encontrada." * O aplicativo sugere que o usuário atualize a lista de atividades.  1. **Erro de Conexão:**  * Se houver problemas de conexão com o servidor ao tentar excluir a atividade |
| **PRÉ-CONDIÇÕES** |
| 1. O usuário está autenticado no aplicativo e possui atividades registradas em sua lista. |
| **PÓS-CONDIÇÕES** |
| 1. A atividade é removida da lista do usuário, e o aplicativo reflete a nova lista de tarefas. |

|  |
| --- |
| **NARRATIVA DE CASO DE USO** |
| **Sigla/Nome:**  CSU07 – Editar Perfil  **Objetivo:**  Permitir que o usuário acesse seu perfil e modifica-lo  **Frequência estimada:** Sempre que um novo usuário querer entra no aplicativo.  **Ator Principal:** Usuário. |
| **CENARIO PRINCIPAL** |
| 1. **Usuário acessa a área de perfil:**  * O caso de uso começa quando o usuário clica no ícone de perfil no canto inferior da tela Home.  1. **Sistema exibe informações atuais do perfil:**  * O sistema exibe as informações de nome, apelido.  1. **Usuário seleciona a opção de editar perfil:**  * O usuário clica no botão “Configurações”.  1. **Usuário modifica o nome ou apelido:**  * O usuário pode alterar os campos de nome e apelido, inserindo os novos valores desejados.  1. **Usuário opta por alterar a senha:**   Se desejar, o usuário pode clicar em “Alterar Senha”. O sistema solicita a senha atual e a nova senha. O usuário insere:   * Senha Atual (para verificação) * Nova Senha * Confirmação da Nova Senha  1. **Usuário confirma as alterações:**  * O usuário clica no botão “Salvar” para confirmar as mudanças.  1. **Sistema valida as informações:**  * O sistema verifica: * Se os campos de nome e apelido estão dentro dos limites de caracteres. * Se a senha atual está correta. * Se a nova senha atende aos requisitos (e.g., mínimo de caracteres, combinação de números e letras).  1. **Sistema salva as alterações:**  * Se todas as informações forem válidas, o sistema atualiza o perfil do usuário no banco de dados.  1. **Sistema exibe mensagem de confirmação:**  * O sistema mostra uma mensagem de sucesso indicando que as alterações foram realizadas com sucesso.  1. **Usuário visualiza o perfil atualizado:**  * O aplicativo retorna à tela de perfil, exibindo as mudanças feitas. |
| **CENÁRIO ALTERNATIVO** |
| 1. **Erro na validação da senha:**   Se o usuário inserir a senha atual incorretamente ou a nova senha não atender aos requisitos, o sistema exibirá uma mensagem de erro e solicitará correção antes de permitir o salvamento. |
| **CENÁRIO EXCESSÃO** |
| 1. **Falha na Conexão com o Servidor:**  * Causa: A conexão com o servidor falha durante a tentativa de salvar as alterações. * Resposta do Sistema: O sistema exibe uma mensagem de erro: “Falha ao salvar as alterações. Verifique sua conexão e tente novamente.” * Ação do Usuário: O usuário deve verificar sua conexão à internet e tentar novamente quando a conexão estiver estável.  1. **Erro no Servidor:**  * Causa: Ocorre um erro interno no servidor durante o processo de atualização de dados. * Resposta do Sistema: O sistema exibe uma mensagem de erro: “Erro no sistema. Não foi possível concluir a alteração. Tente novamente mais tarde.” * Ação do Usuário: O usuário deve esperar e tentar novamente em outro momento ou entrar em contato com o suporte técnico.  1. **Sessão Expirada:**  * Causa: O tempo de sessão do usuário expira enquanto ele está modificando o perfil. * Resposta do Sistema: O sistema redireciona o usuário para a tela de login e exibe uma mensagem: “Sua sessão expirou. Por favor, faça login novamente para continuar.” * Ação do Usuário: O usuário deve fazer login novamente para continuar as alterações.  1. **Senha Atual Inválida:**  * Causa: O usuário insere a senha atual incorretamente ao tentar alterar a senha. * Resposta do Sistema: O sistema exibe uma mensagem de erro: "Senha atual incorreta. Por favor, tente novamente." * Ação do Usuário: O usuário deve inserir a senha correta ou cancelar a alteração de senha. |
| **PRÉ-CONDIÇÕES** |
| 1. O usuário deve estar autenticado no sistema. 2. O usuário já possui um perfil previamente criado. |
| **PÓS-CONDIÇÕES** |
| 1. O perfil do usuário é atualizado com o novo nome, apelido e/ou senha. 2. O usuário pode fazer login com a nova senha (se alterada). |

|  |
| --- |
| **NARRATIVA DE CASO DE USO** |
| **Sigla/Nome:**  CSUO8, editar temporizador Pomodoro**.**  **Objetivo:** Permitir que o usuário ajuste as durações das sessões de trabalho (Pomodoro).  **Frequência estimada:** Sempre que um novo usuário querer entrar no aplicativo.  **Ator Principal:** Usuário. |
| **CENARIO PRINCIPAL** |
| 1. **Usuário acessa as configurações do temporizador:**  * O caso de uso começa quando o usuário clica no ícone de configurações na tela Home do aplicativo e navega até a seção “Pomodoro”.  1. **Sistema exibe as durações atuais:**  * O sistema apresenta as durações atuais para: * Sessão de Concentração (Pomodoro) * Pausa de Descanso  1. **Usuário seleciona a opção de editar temporizador:**  * O usuário clica nas opções de “Sessão de Concentração” e “Pausa de Descanso” para ajustar as durações.  1. **Usuário modifica as durações:**  * O usuário insere ou seleciona novos valores para: * Duração da Sessão de Concentração. * Duração da Pausa de Descanso.  1. **• Usuário confirma as alterações:**  * Após modificar os tempos desejados, o usuário clica no botão "Salvar" ou "Confirmar".  1. **Sistema valida os novos valores:**  * O sistema verifica se os valores inseridos são válidos e numéricos (não há restrição de mínimo ou máximo).  1. **Sistema salva as alterações:**  * O sistema atualiza as configurações de temporizador no perfil do usuário e armazena os novos valores.  1. **Sistema exibe confirmação:**  * O sistema mostra uma mensagem de sucesso indicando que as alterações foram salvas.  1. **Usuário visualiza as novas durações:**  * O aplicativo retorna à tela de configurações, exibindo as novas durações personalizadas. |
| **CENÁRIO ALTERNATIVO** |
| 1. **Erro na validação dos tempos:**   Se o usuário inserir valores fora do intervalo permitido, o sistema exibe uma mensagem de erro.   * O usuário tenta salvar uma sessão de Pomodoro com menos de 10 minutos. * O sistema exibe uma mensagem: “A duração mínima para uma sessão de Pomodoro é de 10 minutos. Por favor, insira um valor válido.” * O usuário deve corrigir o valor e tentar novamente. |
| **CENÁRIO EXCESSÃO** |
| 1. **Falha na Conexão com o Servidor:**  * Causa: A conexão com o servidor falha durante a tentativa de salvar as novas durações. * Resposta do Sistema: O sistema exibe uma mensagem de erro: “Falha ao salvar as alterações. Verifique sua conexão e tente novamente.” * Ação do Usuário: O usuário deve verificar sua conexão à internet e tentar novamente quando a conexão estiver estável.  1. **Erro no Servidor:**  * Causa: Ocorre um erro interno no servidor durante o processo de atualização das configurações. * Resposta do Sistema: O sistema exibe uma mensagem de erro: “Erro no sistema. Não foi possível concluir a alteração. Tente novamente mais tarde.” * Ação do Usuário: O usuário deve esperar e tentar novamente em outro momento ou entrar em contato com o suporte técnico.  1. **Sessão Expirada:**  * Causa: O tempo de sessão do usuário expira enquanto ele está modificando as configurações do temporizador. * Resposta do Sistema: O sistema redireciona o usuário para a tela de login e exibe uma mensagem: “Sua sessão expirou. Por favor, faça login novamente para continuar.” * Ação do Usuário: O usuário deve fazer login novamente para continuar as alterações. |
| **PRÉ-CONDIÇÕES** |
| * + - 1. O usuário deve estar autenticado no sistema.       2. O usuário deve ter um perfil ativo. |
| **PÓS-CONDIÇÕES** |
| 1. O temporizador Pomodoro é atualizado com os novos tempos configurados pelo usuário. 2. As próximas sessões utilizarão os tempos personalizados. |

# Kanban e Retrospectiva

A Arch-Tech desenvolveu um cronograma e um método de organização focados nos objetivos e no tempo do projeto. Nesse contexto, as tarefas foram divididas em categorias específicas e gerais. O tempo e os critérios de cada unidade curricular também foram organizados por Sprint, considerando sua relevância.

Seguindo essa abordagem, o kanban foi estruturado com as prioridades definidas da seguinte forma: foram separados os requisitos para cada unidade curricular, especificando o que deve ser realizado em banco de dados, Android e na parte web.

Além disso, foi criada uma seção para acompanhar o andamento de cada tarefa, proporcionando uma visão clara do progresso da equipe. O grupo também incluiu

uma área dedicada ao que entrará na fase de testes, e, por fim, após passar por

todas as etapas, os requisitos serão movidos para a categoria de concluído.

**Dia 0:**



**Semana 1:**



Na primeira semana foi colocado todas as tarefas em andamento e algumas tarefas foram concluídas, como concluir a estrutura geral sendo o design, modelar o banco e etc., começamos a criar o logo do app e desenvolver os requisitos não funcionais

**Semana 2:**



Tudo que foi colocado em andamento na Primeira semana foi concluída

Concluindo o design faltando a documentação.

**Semana 3:**



Nessa semana concluímos diversas telas importantes e começamos a documentação

**Semana 4:**



Foi feito tudo que foi colocado em andamento, assim concluindo o primeiro Sprint.

**Retrospectiva:**

O grupo analisou os acontecimentos do sprint e chegaram as seguintes conclusões:

**Coisas que deram certo:**

Conseguimos consolidar a ideia do projeto, estabelecer as metas e objetivos, e chegamos a uma conclusão das tecnologias que vão ser utilizadas no projeto.

**O que deu errado:**

Deu errado Modelagem do banco, o primeiro levantamento, tivemos que mudar o design,

**O que pode melhorar:**

A organização, planejamento em relação a tarefas

## Segundo Sprint

A Arch-tech organizou a lógica dos sprints baseada na quantidade de horas de trabalho disponíveis por dia, que, no caso, foram 3 horas. Além disso, foi considerada a quantidade de integrantes presentes na equipe, que eram 4. Assim, multiplicando os valores, foi possível calcular o tempo aproximado de trabalho diário, que somava 12 horas.

* 3 horas X 4 pessoas = 12 horas por dia

Após a contagem de horas, foi realizado o cálculo de dias que o sprint teria, desconsiderando feriados, finais de semana e as aulas de Programação de Dispositivos Móveis. A partir disso, foi encontrado um total de 240 horas.

* 20 dias uteis → 240 horas no Sprint 2

Dessa forma, foi definida uma meta de trabalho de 230 horas, disponibilizando 20 horas (12,5% do sprint) para possíveis imprevistos e atrasos. Portanto, para uma boa execução do sprint seria necessárias 10 horas de trabalho diárias.

**Metas do Segundo Sprint**

Para esse Sprint o grupo separou as seguintes metas a serem cumpridas:

1. **Construir os Componentes das Telas Do projetos.**
2. **Fazer as Rotas Essenciais da Api**

# Product Backlog

Não houve alterações no Product Backlog.

# Sprint Backlog

RF01 - O usuário deve poder se cadastrar, informando e-mail, nome, nome de usuário e senha.

RF04 - O usuário deve poder cadastrar uma atividade, informando título, nível de dificuldade e categoria.

# Burn Down Chart:

**Diagrama de Classe:**

# Kanban e Retrospectiva

A Arch-Tech elaborou um cronograma e um método de organização alinhados aos objetivos do projeto. Nesse contexto, as atividades foram categorizadas em tarefas específicas e gerais. O tempo e os critérios de cada unidade curricular foram organizados em Sprints, considerando sua importância.

Com base nessa abordagem, as prioridades no Kanban foram estruturadas da seguinte forma: foram identificadas as necessidades nas áreas de documentação, front-end. e back-end. Também foi criado um espaço para acompanhar o progresso de cada uma, proporcionando uma visão clara do avanço da equipe. Ao final de cada fase, os requisitos são movidos para a seção de concluídos.

Nesta Sprint, o grupo optou por uma organização diferente em relação à primeira. Assim, as tarefas foram divididas por semanas, o que impactou a forma como o Kanban será apresentado.

**Dia 0:**

****

****

**Semana 01:**

****

Na primeira semana nós já havíamos finalizado as premissas e restrições do projeto, elaboramos os componentes de tela começamos a estilizar as telas de cadastro e login, o processo para cadastrar o usuário e o diagrama de classes.

**Semana 02:**

****Em nossa segunda semana de projeto no segundo sprint, foi finalizada a tela de login, a tela de cadastro estava em processo, e iniciamos as telas de atividade ativa e histórico de atividade.

**Semana 03:**

 Na terceira semana foi iniciado o carregamento do histórico do app, componente de tela do perfil, e a documentação dos Sprints

**Semana 04:**



Semana quatro, sprint dois já finalizado.

**Retrospectiva**

O grupo analisou os acontecimentos do sprint e chegaram as seguintes conclusões:

**Coisas que deram certo:**

* Organização **do Tempo:** A equipe calculou corretamente o total de horas disponíveis, somando 240 horas, o que permitiu uma boa gestão do tempo.
* Divisão **de Tarefas:** As metas foram claramente definidas, como a construção dos componentes das telas e a configuração das rotas, o que facilitou o acompanhamento do progresso.
* Documentação**:** O diagrama de classes e a documentação dos Sprints foram finalizados, proporcionando uma base sólida para o desenvolvimento.

**Coisas que deram errado:**

* Imprevistos **Técnicos:** Durante o Sprint, a equipe enfrentou problemas de compatibilidade entre os sistemas operacionais utilizados nas máquinas, o que atrasou algumas atividades.
* Falta **de Conectividade:** Algumas sessões de trabalho foram impactadas pela instabilidade da internet, dificultando o acesso a recursos online e a comunicação em tempo real.
* Atrasos **nas Entregas:** Algumas tarefas, como a finalização da tela de cadastro, não foram concluídas no prazo estabelecido, o que causou um efeito cascata nas entregas subsequentes.

**O que pode melhorar:**

* Preparação **para Imprevistos:** Criar um plano de contingência mais robusto para lidar com problemas de conexão e energia, como ter materiais offline e opções de trabalho em equipe que não dependam da internet.
* Treinamento **e Suporte:** Oferecer sessões de capacitação sobre as ferramentas e tecnologias que serão utilizadas, para minimizar o baixo domínio da tecnologia.
* Revisão **das Metas:** Revisar as metas do Sprint para torná-las mais realistas, considerando a carga de trabalho e o tempo disponível. Isso pode ajudar a evitar atrasos e frustrações.
* Feedback **Contínuo:** Implementar sessões de feedback mais frequentes durante o Sprint, permitindo ajustes imediatos e uma melhor adaptação às dificuldades encontradas.

# Terceiro Sprint

A Arch-tech organizou a lógica dos sprints com base nas horas de trabalho disponíveis por dia, que foram estipuladas em 3 horas. Além disso, foi considerada a quantidade de integrantes da equipe, totalizando 4 pessoas. Com isso, ao multiplicar os valores, foi possível estimar o tempo total de trabalho diário em 12 horas.

3 horas x 4 pessoas = 12 horas por dia (NO SENAI).

Após o levantamento das horas, foi feito o cálculo dos dias de trabalho no sprint, desconsiderando feriados, finais de semana e as aulas de Programação de Dispositivos Móveis. O resultado foi um total de 240 horas.

20 dias úteis → 240 horas no Sprint 3.

Com base nisso, foi estabelecida uma meta de trabalho de 150 horas, reservando 90 horas (37,5% do sprint) para possíveis imprevistos e atrasos. Portanto, para uma boa execução do sprint, seriam necessárias cerca de 10 horas de trabalho diárias.

Para este sprint, o grupo definiu as seguintes metas: desenvolver as telas, como a loja, perfil e atividades já prontas. Isso inclui desde a parte de design, codificação, responsividade, até a integração com o modelo MVC e conexão das funções ao banco de dados. Além disso, foi planejado finalizar toda a parte de back-end (lógica) das classes flashcard, arquivo e pomodoro.

**Metas do Terceiro Sprint**

Para esse Sprint o grupo separou as seguintes metas a serem cumpridas:

* **Começar a fazer os testes**
* **E consumir as rotas**

# Product Backlog

* Não houve alterações no Product Backlog.

# Sprint Backlog

* RF11: O usuário deve poder gastar seus pontos adquiridos por meio de atividades em itens colecionáveis na loja do aplicativo. A partir do momento em que um item é comprado, este ficará disponível na tela de perfil do usuário, sendo exibido no formato de grade.
* RF12: O usuário deve ter acesso às suas informações cadastradas anteriormente e poder, a partir de verificação de identidade, editá-las.
* RF13: O sistema deve permitir que o usuário visualize dados específicos sobre seu desempenho: o total de pontos acumulados e a maior quantidade de dias consecutivos em que pelo menos uma atividade foi realizada.

# Burn Down Chart:

**Diagrama de Classe:**

* + 1. Plano de Testes

O plano de teste é uma maneira de encontrar defeitos e bugs no sistema para serem futuramente corrigidos. Mesmo que durante a programação sempre foram realizados alguns, ainda sim no plano são realizados testes finais passo a passo.

|  |  |  |
| --- | --- | --- |
| Plano de Testes de Login | | |
| PASSOS | RESULTADO ESPERADO | RESULTADO OBTIDO |
| digitar um login cadastrado no banco, mas com a senha errada. | o sistema deverá mostrar a mensagem de erro ao logar, pois o nome e a senha devem estar iguais ao efetuar o cadastro. | O sistema mostra a mensagem de erro “senha incorreta”. |
| Digitar login e senha que não foram cadastrados no banco. | o sistema deverá mostrar a mensagem de erro ao logar, pois o nome e a senha devem estar iguais ao efetuar o cadastro. | O sistema mostra a mensagem de erro “usuário não cadastrado”. |
| Deixar o campo de senha vazio. | o sistema deverá mostrar a mensagem de erro ao logar, pois o nome e a senha devem estar preenchidos. | O sistema mostra que é necessário preencher o campo de senha. |
| Deixar o campo de login vazio. | o sistema deverá mostrar a mensagem de erro ao logar, pois o nome e a senha devem estar preenchidos. | O sistema mostra que é necessário preencher o campo de login. |
| digitar uma senha cadastrada no banco, mas com o login errado. | o sistema deverá mostrar a mensagem de erro ao logar, pois o nome e a senha devem estar iguais ao efetuar o cadastro | O sistema mostra a mensagem de erro “usuário não cadastrado”. |
| digitar um login e senha correspondentes cadastrados no banco. | o sistema deverá entrar na tela inicial do usuário. | O sistema entra na página inicial do usuário. |

|  |  |  |
| --- | --- | --- |
| Plano de Testes de Cadastro | | |
| PASSOS | RESULTADO ESPERADO | RESULTADO OBTIDO |
| Deixar os campos Email, nome, apelido e senha vazios. | O sistema deve retornar status 400 e mensagem de erro: "Todos os campos são obrigatórios". | O sistema retorna status 400 e mensagem de erro ("Todos os campos são obrigatórios"). |
| Digitar uma senha com menos de 6 caracteres. | O sistema deve retornar status 400 e mensagem de erro: "A senha deve ter pelo menos 6 caracteres". | O sistema retorna status 400 e mensagem de erro (A senha deve ter pelo menos 6 caracteres). |
| Usar um apelido já existente. | O sistema deve retornar status 400 e mensagem de erro: "Apelido já existe". | O sistema retorna status 400 e mensagem de erro (Apelido já existe). |
| Usar um Email já cadastrado. | O sistema deve retornar status 400 e mensagem de erro: "Usuário já cadastrado". | O sistema retorna status 400 e mensagem de erro (Usuário já cadastrado). |
| Preencher todos os campos corretamente. | O sistema deve chamar o próximo middleware, permitindo o cadastro do usuário. | O sistema chama o próximo middleware. |
| Deixar o campo apelido vazio, mas preencher os outros campos. | O sistema deve retornar status 400 e mensagem de erro: "Todos os campos são obrigatórios". | O sistema retorna status 400 e mensagem de erro (Todos os campos são obrigatórios). |
| Deixar o campo senha vazio, mas preencher os outros campos. | O sistema deve retornar status 400 e mensagem de erro: "Todos os campos são obrigatórios". | O sistema retorna status 400 e mensagem de erro (Todos os campos são obrigatórios.). |

|  |  |  |
| --- | --- | --- |
| Plano de Testes de Criação de Atividade | | |
| PASSOS | RESULTADO ESPERADO | RESULTADO OBTIDO |
| Deixar os campos **título da lição**, **dificuldade**, **categoria** e **descrição** vazios. | O sistema deve retornar a mensagem de erro: "Por favor, preencha o campo título". | O sistema retornou a mensagem de erro: "Por favor, preencha o campo título". |
| Preencher apenas o campo **Título da atividade** e deixar os outros campos vazios. | O sistema deve retornar a mensagem de erro: "Por favor, preencha o campo categoria". | O sistema retorna a mensagem de erro (A senha deve ter pelo menos 6 caracteres). |
| Selecionar uma **dificuldade** válida (por exemplo, clicando no botão para selecionar "Fácil", "Normal" ou "Difícil") e deixar os outros campos vazios. | O sistema deve retornar a mensagem de erro: "Por favor, preencha o campo título". | O sistema retornou a mensagem de erro: "Por favor, preencha o campo título". |
| Preencher os campos **título da atividade categoria**, **descrição**, mas não clicar no botão de **dificuldade**. | O sistema deve retornar a mensagem de erro: "Por favor, preencha o campo dificuldade". | O sistema retornou a mensagem de erro: "Por favor, preencha o campo dificuldade". |
| Preencher os campos **nome da lição**, **dificuldade**, **categoria**, mas deixar o campo **descrição** vazio. | O sistema deve permitir o cadastro, mas com descrição em branco. | O sistema chamou a tela de pomodoro. |
| Preencher todos os campos corretamente: **nome da lição**, **dificuldade**, **categoria**, **descrição**. | O sistema deve cadastrar a atividade e, passar para a tela de pomodoro. | O sistema cadastrou a atividade e passou para a tela de pomodoro. |

|  |  |  |
| --- | --- | --- |
| Plano de Testes de Consulta de Atividade | | |
| PASSOS | RESULTADO ESPERADO | RESULTADO OBTIDO |
| Consultar as atividades com o usuário logado. | O sistema exibe apenas as atividades pertencentes ao usuário logado. | O sistema exibiu corretamente as atividades do usuário logado. |
| Consultar as atividades sem estar logado no sistema. | O sistema não permite a consulta, exibindo uma mensagem de erro indicando que é necessário fazer login. | O sistema exibiu a mensagem: "É necessário fazer login para consultar as atividades." |
| Consultar atividades de mais de 2 meses atrás. | O sistema exibe apenas atividades dos últimos 2 meses, ignorando as mais antigas. | O sistema exibiu corretamente apenas atividades criadas nos últimos 2 meses.. |
| Tentar consultar as atividades de outro usuário. | O sistema não permite a consulta, exibindo apenas atividades do usuário logado. | O sistema exibiu apenas as atividades pertencentes ao usuário logado, sem acesso às de outros usuários. |

* + 1. Kanban e Retrospectiva

A Arch-Tech desenvolveu um cronograma e um método de organização focados nos objetivos e no tempo do projeto. Nesse contexto, as tarefas foram divididas em categorias específicas e gerais. O tempo e os critérios de cada unidade curricular também foram organizados por Sprint, considerando sua relevância.

Seguindo essa abordagem, o kanban foi estruturado com as prioridades definidas da seguinte forma: foram separados os requisitos para cada unidade curricular, especificando o que deve ser realizado em banco de dados, Android e na parte web.

Além disso, foi criada uma seção para acompanhar o andamento de cada tarefa, proporcionando uma visão clara do progresso da equipe. O grupo também incluiu

uma área dedicada ao que entrará na fase de testes, e, por fim, após passar por

todas as etapas, os requisitos serão movidos para a categoria de concluído.

**Semana 1**

Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

Descrição gerada automaticamente

Na primeira semana focamos no Node e em criar as narrativas de caso de uso

**Semana 2**

Imagem de texto no celular

Descrição gerada automaticamente com confiança média

Na segunda semana terminou o node e a loja

**Semana 3**

Tela de celular com publicação numa rede social

Descrição gerada automaticamente

Na terceira terminou as narrativas e começou a consumir as rotas

**Semana 4**

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Na quarta foi começada os planos de testes

**Semana 5**

Interface gráfica do usuário, Aplicativo, chat ou mensagem de texto

Descrição gerada automaticamente

Na quinta terminou os planos de testes e de consumir as rotas

**Retrospectiva**

A Arch-Tech elaborou um cronograma e um método de organização alinhados aos objetivos do projeto. Nesse contexto, as atividades foram categorizadas em tarefas específicas e gerais. O tempo e os critérios de cada unidade curricular foram organizados em Sprints, considerando sua importância.

Com base nessa abordagem, as prioridades no Kanban foram estruturadas da seguinte forma: foram identificadas as necessidades nas áreas de documentação, front-end e back-end. Também foi criado um espaço para acompanhar o progresso de cada uma, proporcionando uma visão clara do avanço da equipe. Ao final de cada fase, os requisitos são movidos para a seção de concluídos.

Nesta terceira Sprint, o foco foi na integração e consumo das rotas das telas, com o objetivo de garantir que a comunicação entre o front-end e o back-end fosse eficiente e sem falhas. O grupo manteve a divisão das tarefas por semanas, impactando a forma como o Kanban é apresentado, de modo a refletir claramente o progresso das implementações**.**

# Quarto Sprint

No quarto sprint, a Arch-tech estruturou a organização das atividades levando em consideração as horas de trabalho disponíveis por dia, definidas como 3 horas. Também foi levado em conta o número de integrantes da equipe, totalizando 4 pessoas. Dessa forma, ao multiplicar as horas diárias pelo número de membros, obteve-se um total de 12 horas de trabalho diário.

3 horas x 4 pessoas = 12 horas por dia (no SENAI).

Com base nisso, foi calculado o total de horas do sprint, desconsiderando feriados, finais de semana e as aulas de Programação de Dispositivos Móveis. O total resultante foi de 240 horas de trabalho.

20 dias úteis → 240 horas no Sprint 4.

Para otimizar a execução do sprint, foi estabelecida uma meta de 150 horas de trabalho, deixando 90 horas (37,5% do sprint) para possíveis imprevistos e atrasos. Assim, para garantir um bom andamento das atividades, seria necessário dedicar cerca de 10 horas de trabalho por dia.

Nesse último sprint o grupo decidiu focar na conclusão do app e na parte do final do back, apenas finalizando o trabalho para entrega-lo. Portanto o grupo colocou como prioridade finalizar o back, o editar do perfil criar um timer funcional.

# Product Backlog

# Sprint Backlog

# Burn Down Chart

# Diagramas

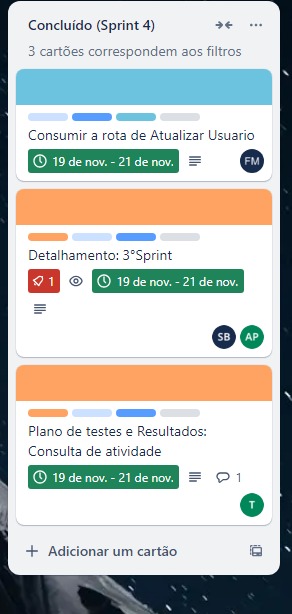
|  |  |  |
| --- | --- | --- |
| Plano de Testes de Editar Perfil | | |
| PASSOS | RESULTADO ESPERADO | RESULTADO OBTIDO |
| 1. Deixar os campos Nome, Apelido, Nova senha e confirmar senha vazios. | O sistema deve retornar mensagem de erro: " Preencha, pelo menos, um campo!". | O sistema retorna ("Preencha, pelo menos, um campo!"). |
| 2. Preencher o campo "Nova senha" sem as especificações ditas | O sistema deve retornar mensagem de erro: " Sua senha deve conter: • pelo menos 8 caracteres • pelo menos uma letra maiúscula • pelo menos um número • pelo menos um caractere especial (!@#$%^&\*(),.?":{}|<>)” | O sistema retorna mensagem de erro " Sua senha deve conter: • pelo menos 8 caracteres • pelo menos uma letra maiúscula • pelo menos um número • pelo menos um caractere especial (!@#$%^&\*(),.?":{}|<>)” |
| 4. Preencher "Confirmar senha" com valor diferente de "Nova senha". | O sistema deve retornar mensagem de erro: "As senhas não coincidem". | O sistema retorna status 400 e mensagem de erro (As senhas não coincidem). |
| 5. Preencher todos os campos corretamente. | O sistema deve chamar a próxima tela, atualizando os dados do usuário e mandando ele ir fazer login de novo. | O sistema chama a próxima tela. atualizando os dados do usuário e mandando ele ir fazer login de novo. |
| 7. Deixar o campo "Confirmar senha" vazio, mas preencher os outros campos. | O sistema deve retornar mensagem de erro: "senha não coincidem". | O sistema retornou mensagem de erro: "senha não coincidem". |

# Kanban e Retrospectiva

No quarto sprint, a Arch-tech seguiu uma abordagem similar à do sprint anterior, utilizando um método de organização baseado no tempo disponível e na distribuição de tarefas entre os membros da equipe. A equipe levou em consideração que cada integrante trabalharia 3 horas por dia, totalizando 12 horas diárias de trabalho. Considerando o total de 240 horas de trabalho no sprint, o grupo estabeleceu uma meta de 150 horas, reservando 90 horas para imprevistos e atrasos. O foco principal do sprint foi a conclusão do aplicativo, com ênfase na finalização do back-end e na implementação de funcionalidades como o editor de perfil e a criação de um timer funcional.

Com base nisso, o grupo organizou o kanban para o quarto sprint da seguinte maneira:

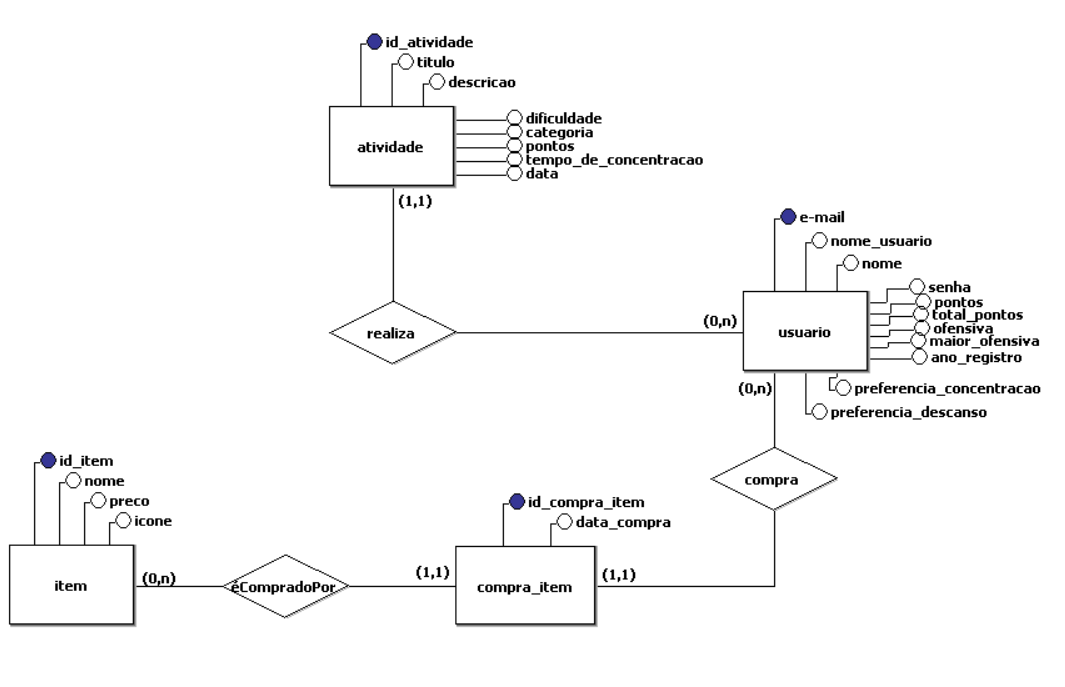
**Semana 1**



# Modelo de Dados

Esta parte do planejamento traz informações necessárias para a construção de um banco de dados para o Sistema de Gerenciamento de Acessos.

# Diagrama de Entidade e Relacionamento



# Modelo de Dados do Banco

Usuários: Array

[

{

anoRegistro: Int

apelido: String

maiorOfensiva: Int

nome: String

ofesiva: Int

preferenciaConcentração: Int

preferenciaDescanso: Int

TotalPontos: Int

email: String

}

]

Itens: Array

[

{

Id: Int

Nome:string

Preco: Int

Icone: string

}

]

Atividade: Array

[

{

Categoria: String

Titulo: String

Data: timestamp

Descrição: String

Dificuldade: Int

Email: String

Id: String

Pontos: Int

tempoConcentração: Int

}

]

# Dicionário de dados

|  |  |  |
| --- | --- | --- |
| **Coleção: usuarios** | | |
| **Descrição: Armazena as informações dos usuários. Cada documento é identificado pelo email do usuário.** | | |
| Campo | Tipo de Dados | Descrição |
| anoRegistro | number | Ano em que o usuário foi registrado. |
| apelido | string | Apelido do usuário. |
| atividades | array (string) | IDs das atividades associadas ao usuário. |
| email | string | Email do usuário (ID do documento). |
| itens | array (string) | IDs dos itens possuídos pelo usuário. |
| maiorOfensiva | number | Maior ofensiva alcançada pelo usuário. |
| nome | string | Nome completo do usuário. |
| ofensiva | number | Ofensiva atual do usuário. |
| pontos | number | Pontuação atual do usuário. |
| preferenciaConcentracao | number | Tempo de concentração preferido (em minutos). |
| preferenciaDescanso | number | Tempo de descanso preferido (em minutos). |
| senhaHash | string | Hash da senha do usuário. |
| totalPontos | number | Pontuação total acumulada pelo usuário. |
| ultimaAtividade | string | ID da última atividade realizada pelo usuário. |

|  |  |  |
| --- | --- | --- |
| **Coleção: itens** | | |
| **Descrição: Armazena os itens disponíveis no sistema. Cada documento é identificado por um ID aleatório gerado pelo Firestore.** | | |
| Campo | Tipo de Dados | Descrição |
| icone | string | URL ou referência ao ícone do item. |
| id | string | ID único do item. |
| nome | string | Nome do item. |
| preco | number | Preço do item em pontos. |

|  |  |  |
| --- | --- | --- |
| **Coleção: atividades** | | |
| **Descrição: Armazena as atividades realizadas ou disponíveis no sistema. Cada documento é identificado por um ID aleatório gerado pelo Firestore.** | | |
| Campo | Tipo de Dados | Descrição |
| categoria | string | Categoria da atividade. |
| data | timestamp | Data e hora da realização ou criação da atividade. |
| descricao | string | Descrição da atividade. |
| dificuldade | number | Nível de dificuldade da atividade. |
| email | string | Email do usuário responsável pela atividade. |
| id | string | ID único da atividade. |
| pontos | number | Pontos atribuídos pela realização da atividade. |
| tempoConcentracao | number | Tempo de concentração necessário (em minutos). |
| titulo | string | Título da atividade. |

# PRINCIPAIS TELAS DO SISTEMA

* Tela de cadastro



* Tela de login



* Tela inicial



* Tela de nova atividade



* Tela de atividade ativa



* Tela da loja



* Tela de perfil



* Tela de atualização do perfil



* Tela de edição do temporizador



# CONCLUSÃO

# Escreva os resultados obtidos

Resultados obtidos

# Constatações

Constatações

# Sugestões de possíveis aperfeiçoamentos técnicos

Sugestões

# REFERÊNCIAS

**O que é o 'Método Pomodoro' de gestão de tempo que conquistou fãs ao redor do mundo por melhorar produtividade disponível em:** <<https://www.bbc.com/portuguese/geral-55563393>>

[PomodoroTechnique.pdf](https://edisciplinas.usp.br/pluginfile.php/4684649/mod_folder/content/0/PomodoroTechnique.pdf)

[Técnica Pomodoro - o que é e como funciona - Brasil Escola](https://brasilescola.uol.com.br/dicas-de-estudo/tecnica-pomodoro-que-e-e-como-funciona.htm)

# GLOSSÁRIO

**\*Pomodoro** -A **Técnica Pomodoro** é um [método de gerenciamento de tempo](https://pt.wikipedia.org/wiki/Gerenciamento_de_tempo) desenvolvido por Francesco Cirillo no final dos anos 1980. A técnica consiste na utilização de um [cronômetro](https://pt.wikipedia.org/wiki/Cron%C3%B4metro) para dividir o trabalho em períodos de 25 minutos, separados por breves intervalos.

\***Kanban -** Kanban é um sistema visual de gestão de trabalho que utiliza cartões ou sinais visuais para controlar e gerenciar o fluxo de trabalho12345. Ele representa o processo de produção e permite uma gestão organizada e compreensível para todos os envolvidos. O método visa otimizar os fluxos de trabalho e identificar possíveis gargalos no processo.

**Scrum** - Metodologia ágil de gerenciamento de projetos que utiliza iterações chamadas "sprints" para desenvolvimento incremental. Inclui papéis como Product Owner, Scrum Master e equipe de desenvolvimento, além de reuniões diárias e retrospectivas.