



PROCESSO, TECNOLOGIAS E FERRAMENTAS

SALUTE - Sistema de Alocação e Logística de Uso de Turmas e Espaços

Equipe:

509722 – Daniel Santos Fernandes

510918 – Gabriel Ferreira Cruz Farias

508160 – Lucas Levy de Oliveira Barros

Versão do projeto	Aluno(s)	O que fez?	Data
V1 - Primeira Entrega	Daniel Fernandes, Gabriel Ferreira e Lucas Levy	Definição do Processo, Tecnologias e Ferramentas	28, 29 e 30/08/2023
V1.1. - Primeira entrega	Lucas Levy	Ajustes finais para a primeira entrega	03/09/2023
V2 - Segunda Entrega	Lucas Levy	Adição da modelagem do Banco, ferramenta usada para modelagem	19/09/2023
V2 - Segunda Entrega	Lucas Levy	Adição do Repositório com a configuração inicial do projeto em Java, atualização das tecnologias	26/09/2023
V3 - Status Report	Lucas Levy	Adição da Ferramenta Scene Builder e do FlowChart de Alocação Automática	18/10/2023

1 - Processo

Planejamento Inicial:

Reunião para Obtenção de Ideias:

A reunião inicial teve por objetivo a obtenção de ideias de projetos que poderíamos desenvolver, foi feito um brainstorming, no qual, os membros da equipe dispuseram suas ideias, sem questionamentos.

Após o brainstorming, as ideias foram analisadas, restando apenas as mais bem avaliadas, pelos critérios escolhidos, facilidade de prospecção de clientes, possibilidade de implementação para desktop, dificuldade de desenvolvimento, possibilidade do cliente precisar do sistema, possíveis tecnologias que seriam utilizadas.

Prospecção de Clientes e Levantamento de Requisitos Iniciais:

Nesta fase, procuramos um possível cliente, apresentamos nossa ideia a ele e buscamos fazer perguntas que nos deram uma lista de requisitos iniciais para o projeto e possibilitar descobrir as necessidades dele. Com essas informações, podemos definir os objetivos do projeto.

Requisitos, Stakeholders e Visão do Projeto:

Criação de uma lista de requisitos funcionais e não funcionais, identificados durante a conversa com o cliente, presente em um documento de requisitos.

Criação do Documento de Visão, com o objetivo do projeto, seu escopo positivo(o que iremos desenvolver) e negativo(o que não será atendido). Também, identificamos os stakeholders(partes interessadas) do sistema.

Tecnologias e Ferramentas:

A partir dos requisitos funcionais e não funcionais, buscamos identificar as tecnologias que serão utilizadas no projeto, tais como: Linguagem de programação, frameworks e bibliotecas, ferramentas de modelagem e prototipação, facilitadores de gerenciamento de tarefas, IDEs, banco de dados e etc.

Atribuição de Papéis:

Definir qual papel cada membro terá durante o desenvolvimento do sistema, esse deve ser um evento recorrente dentro do projeto, ocorrendo toda vez que os membros acharem necessário e/ou o projeto entrar em uma nova fase.

Análise e Design:

Protótipo de Baixa Fidelidade:

Após análise dos requisitos iniciais, é feito um protótipo de baixa fidelidade, que servirá de base para a avaliação do cliente, para entendermos se estamos no caminho certo, o que podemos melhorar e etc.

Questionários:

Para entender melhor como um sistema de alocação de salas é visto pelos outros stakeholders, serão criados questionários específicos para cada um deles (professores e alunos), com os dados destes questionários, provavelmente haverá mais algum requisito, ou ponto de melhoria.

Validação e Priorização de Requisitos (Backlog do Produto):

Após termos os requisitos iniciais, questionários e o protótipo de baixa fidelidade, entraremos em contato com o cliente para validação dos requisitos e protótipo de baixa fidelidade, além, da apresentação da compilação da entrevista, assim podem surgir novos requisitos ou o cancelamento de algum outro. Além disso, faremos a priorização dos requisitos. Essa reunião também servirá para a criação do Backlog do Produto (Scrum).

Especificação dos Casos de Uso:

Após a validação dos requisitos, deverão ser identificados os atores do sistema e criados os casos de uso.

Protótipo de Alta Fidelidade:

Criação do protótipo de alta fidelidade, a partir, dos requisitos e casos de uso, esse protótipo contém um esboço mais detalhado de como será a aplicação, com os fluxos de uso e interações do usuário.

- [Link para o Protótipo](#)

Criação de diagramas:

A partir dos dados obtidos das etapas anteriores, serão criados os diagramas de classes e casos de uso e será feita a modelagem do banco de dados.

Diagrama Lógico (Banco de Dados):

A partir dos requisitos obtidos, foi possível criar o diagrama lógico do banco de dados.

- [Link para o Modelo](#)

FlowChart do algoritmo de alocação automática:

Com o levantamento dos requisitos e a criação do documento que quebra o épico de alocação automática em pequenas tarefas, foi possível criar um FlowChart que procura detalhar o funcionamento desse algoritmo.

- [Link para o Diagrama Online](#)
- [Link para o Diagrama no Repositório](#)

Criação do Repositório no GitHub:

O repositório no GitHub foi criado para fornecer um local centralizado para armazenar e gerenciar o código-fonte do projeto, ele irá facilitar a colaboração entre os membros da equipe, rastrear o histórico de alterações, gerenciar problemas e disponibilizar o código para outras pessoas.

- [Link para o Repositório](#)

Desenvolvimento Iterativo:

Divisão em Sprints:

O processo de desenvolvimento será dividido em sprints com duração de uma semana, isso tornará o feedback mais rápido e tornará o desenvolvimento mais dinâmico e teremos algo para entregar toda quarta durante a aula. Essa é uma das práticas do Scrum

Planejamento da Sprint:

Essa reunião deverá ocorrer toda quarta, às 21:00, para planejar quais task serão feitas durante a semana, levando em conta o peso de cada uma e o que é possível ser entregue dentro do prazo. Ao final, teremos o Backlog da Sprint, e assim, poderemos começar o desenvolvimento.

Revisão da Sprint:

A Revisão da Sprint deverá ocorrer na terça feira, 21:00, nesta reunião, veremos o que pode ser produzido, e como melhorar o processo de desenvolvimento.

Desenvolvimento:

Essa fase está incluída na sprint, o desenvolvedor deverá implementar a funcionalidade, e cuidar para que ela tenha funcionamento correto e não afete outras funcionalidades do código, devendo fazer testes de unidade e de integração, se necessário.

Testes:

Etapa, na qual, o sistema deve ser testado, para verificar se o que foi implementado está funcionando corretamente e atende as necessidades do cliente.

Reunião com o Cliente:

A cada duas ou três semanas, deveremos ter uma reunião com o cliente para entrega do que já foi feito e validação parcial, assim, teremos a certeza de que estamos no caminho certo para o desenvolvimento e os requisitos continuam os mesmo. Pode ser negociada uma mudança de requisitos, pequenos ajustes, algo que não afete tanto o rumo que já foi tomado.

Validação e Entrega:

Validação Final:

Ao final do desenvolvimento e dos testes, entraremos em contato com o cliente, para validação final do sistema, iremos dar o check final e verificar se o sistema atende as necessidades dele.

Entrega:

Após a validação, será feita a instalação do sistema e será feito um treinamento sobre o funcionamento do sistema.

2 - Tecnologias e Ferramentas

Scrum:

O Scrum é uma maneira de organizar o trabalho em projetos, como montar um quebra-cabeça. Ele divide o grande trabalho em pequenas partes chamadas de "sprints", como pedacinhos do quebra-cabeça. Cada sprint tem um tempo certo para ser feito, como um jogo que dura um tempo. Assim, a equipe trabalha em partes menores e pode ver o progresso.

Como será utilizada:

A equipe irá se reunir regularmente para falar sobre o trabalho que precisa ser feito. Vamos escolher as tarefas mais importantes para cada sprint, como escolher as peças mais importantes do quebra-cabeça. Ao final de cada sprint, olhar o que conseguimos fazer e decidir o que fazer no próximo sprint.

Trello:

O Trello é uma ferramenta de gerenciamento de projetos baseada em quadros virtuais. Ele permite criar, organizar e acompanhar tarefas e informações de forma visual e colaborativa. Os quadros são divididos em colunas, onde cada coluna representa um estágio do projeto, e os cartões são usados para representar tarefas individuais.

Como será utilizada:

Utilizaremos o Trello para criar uma lista de tarefas que precisam ser feitas, vamos organizar as tarefas em grupos e marcar quando cada tarefa estiver pronta. Assim, todos sabem o que está acontecendo e o que precisa ser feito. No Trello, manteremos nosso board de tasks.

Miro:

O Miro é como uma lousa digital onde você desenha suas ideias, como desenhar em um quadro branco. Ele ajuda a mostrar como algo pode ser feito antes de realmente fazer.

Como será utilizada:

Nós usaremos o Miro para criação do protótipo de baixa fidelidade do sistema. Vamos fazer rascunhos das telas e tentar ver como os requisitos se encaixam. Assim, poderemos visualizar como será um dos pontapés iniciais do sistema.

Git

Git é um sistema de controle de versão amplamente utilizado no desenvolvimento de software que rastreia e gerencia mudanças no código-fonte. Ele permite que várias pessoas

trabalhem simultaneamente em um projeto, controlando o histórico das alterações e facilitando a colaboração.

Como será utilizada:

No projeto, utilizaremos o Git para rastrear todas as alterações feitas no código-fonte. Cada membro da equipe fará "commits" regulares para salvar suas mudanças no repositório Git, o que preserva um histórico completo de todas as edições.

GitHub

GitHub é uma plataforma de hospedagem e colaboração de código-fonte que permite que os desenvolvedores armazenem, compartilhem e colaborem em projetos de software de forma distribuída. Ele fornece recursos para controlar versões, gerenciar problemas (issues) e facilitar a colaboração em equipe.

Como será utilizada:

Utilizaremos o GitHub como um repositório centralizado para hospedar nosso código-fonte. Isso garantirá que todos os membros da equipe tenham acesso ao código atualizado e poderão contribuir para o projeto.

DB Diagram IO

É uma ferramenta de modelagem de banco de dados.

Como será utilizada:

Será utilizada para a criação da modelagem do banco de dados do sistema SALUTE, mais precisamente o Diagrama Lógico.

Figma:

O Figma é uma ferramenta para criação de protótipos, muito poderosa, com ele podemos simular interações de usuários, como ficarão as telas do sistema e os fluxos de interações.

Como será utilizada:

Vamos usar o Figma para criar o protótipo de alta fidelidade do sistema. Fazer cada tela, botão e caixa de texto como se estivessem desenhando um quadro. Isso ajuda a entender como o programa vai parecer para as pessoas.

Astah UML:

Astah é uma ferramenta de modelagem que permite criar diagramas UML (Unified Modeling Language) e outros tipos de diagramas para visualizar e documentar sistemas e processos. Ele é amplamente utilizado por desenvolvedores de software, engenheiros de sistemas e analistas para planejar, projetar e documentar os aspectos estruturais e comportamentais de um sistema.

Como será utilizada:

Utilizaremos para a criação do diagrama de classes, para analisar a interação das entidades e outras classes e para a modelagem do banco de dados.

NetBeans:

O NetBeans é um ambiente de desenvolvimento integrado (IDE) que simplifica a criação de aplicativos de software. Ele oferece uma plataforma amigável para escrever, depurar e testar código em diversas linguagens de programação, incluindo Java. Com recursos como sugestões de código e depuração visual, o NetBeans auxilia desenvolvedores a construir programas de maneira eficiente e produtiva.

Como será utilizada:

No projeto, utilizaremos o NetBeans como nossa ferramenta principal de desenvolvimento. Nele, vamos escrever o código do programa em Java, criando as instruções necessárias para que o sistema funcione corretamente. Também, vamos aproveitar recursos como: sugestões de código e depuração para garantir a qualidade e eficácia do nosso software.

Maven:

O Maven é uma ferramenta de automação de construção (build) amplamente utilizada no desenvolvimento de software. Ele auxilia na construção, gerenciamento de dependências e na organização de projetos de software de maneira eficiente.

Como será utilizada:

No projeto, usaremos o Maven para gerenciar as dependências do software. Isso incluirá bibliotecas Java necessárias, como o SQLite, JavaFX, Aspose.Cells e Apache PDFBox. Maven automaticamente baixará e manterá essas dependências atualizadas, simplificando o processo de integração dessas bibliotecas no projeto. Ele também será usado para automatizar a compilação do código-fonte Java do nosso sistema, padronização do projeto e ciclo de vida.

Java:

Java é uma linguagem de programação versátil amplamente usada para desenvolver software que pode ser executado em diferentes tipos de dispositivos.

Como será utilizada:

No nosso sistema, utilizaremos Java como a linguagem principal para escrever o código que faz o programa funcionar. Com Java, criaremos a lógica por trás das interações do usuário e a manipulação dos dados, permitindo que nosso sistema atenda às necessidades da faculdade.

JavaFX:

JavaFX é uma plataforma que estende o Java para a criação de interfaces gráficas interativas e visualmente atraentes em aplicativos de software.

Como será utilizada:

Integramos o JavaFX ao nosso sistema para desenvolver a interface visual que os usuários verão e interagirão. Com o JavaFX, criaremos botões, janelas e elementos de design, tornando a experiência do usuário mais amigável e interativa.

Scene Builder:

Scene builder é uma ferramenta para criação de interfaces gráficas com JavaFX e Java.

Como será utilizada:

Vamos utilizar para criar de forma mais rápida e fácil a interface do sistema.

iReport:

O iReport é uma ferramenta de geração de relatórios que permite criar layouts personalizados para apresentar informações de maneira visualmente atraente.

Como será utilizada:

No sistema, utilizaremos o iReport para criar relatórios detalhados com os dados do sistema, como informações de turmas, alocação de salas e estatísticas. Com o iReport, poderemos formatar os relatórios de maneira profissional e personalizada, tornando a visualização dos resultados mais clara e compreensível.

Aspose.Cells for Java e Apache PDFBox:

O Aspose.Cells for Java e o Apache PDFBox são bibliotecas que permitem manipular arquivos Excel e PDF em aplicativos Java, respectivamente.

Como será utilizada:

Integraremos o Aspose.Cells e o Apache PDFBox ao nosso sistema para abrir, editar e criar arquivos Excel e PDF. Usaremos o Aspose.Cells para lidar com planilhas e informações do Excel, e o Apache PDFBox para manipular arquivos PDF, permitindo que o sistema carregue e processe esses tipos de documentos de maneira eficiente e precisa.

SQLite:

O SQLite é um sistema de gerenciamento de banco de dados relacional leve e incorporado. Ele é amplamente utilizado para armazenar dados localmente em aplicativos de software, sendo uma opção popular para sistemas que não requerem um servidor de banco de dados separado.

Como será utilizada:

Utilizaremos o SQLite para armazenar dados localmente. Isso incluirá informações como configurações do sistema, histórico de alocação de salas e outras informações que precisam ser mantidas e acessadas pelo próprio sistema.