



SENAC - Campus Santo Amaro

TADS - Análise Desenvolvimento de Sistemas

**PROJETO INTEDRADOR
DESENVOLVIMENTO DE SISTEMAS
ORIENTADO A OBJETOS**

ALUNOS: GUILHERME SANTOS
LEVI FREITAS
PEDRO MELO
RODRIGO NOGUEIRA

PROFESSOR: MARCOS MONTEIRO

SÃO PAULO
2024

Sumário

1. Componentes da Equipe.....	03
2. Especificação do Sistema.....	05
3. Cenários	08
4. Planejamento (Cronograma).....	09
5. Solução de Software (Requisitos).....	10
6. Casos de Uso.....	11
7. Diagrama de Classes.....	15
8. Modelo de Dados.....	16
9. Conclusão	17

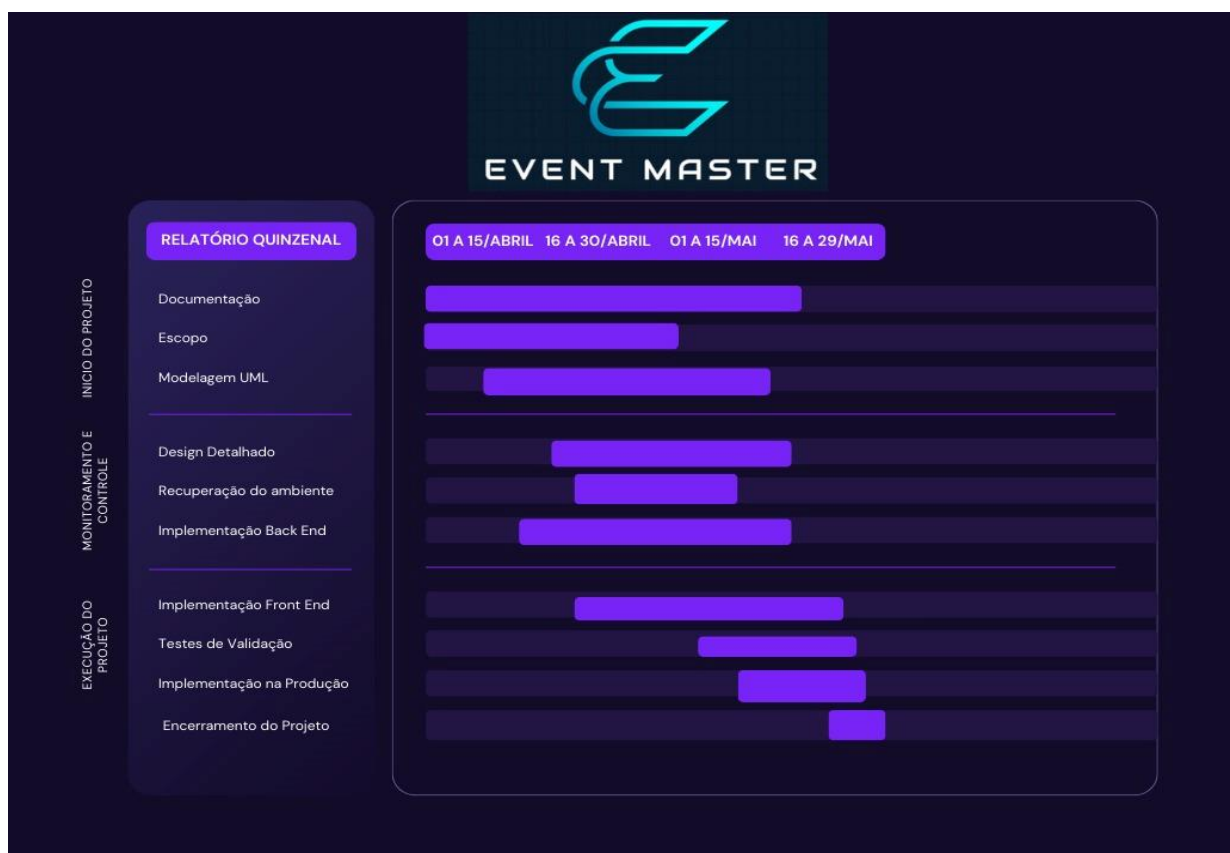
1. Componentes da Equipe

Levi: Gestor do Projeto, elaboração do Escopo e Planejamento, análise de requisitos, coordenação do Projeto. Revisão e aprovação dos entregáveis nos prazos.

Rodrigo: Analista e Documentador, colaboração na análise de requisitos e da Modelagem UML, relacionamentos, dos diagramas de classes, de casos de uso e da documentação técnica

Guilherme e Pedro: Desenvolvedores Java, implementação do sistema utilizando Orientação a Objetos, desenvolvimento do Backend e Frontend, realização de testes e validação.

1.1 Cronograma



1.2 EAP (Estrutura Analítica de Projeto)



2. Especificação de Sistema

2.1 Requisitos Funcionais

RF001 - Permitir a busca de Ingressos:

A plataforma deve permitir aos usuários realizarem buscas de ingressos em eventos com base em shows disponíveis, datas, quantidade de espectadores.

RF002 - Cadastro de Eventos:

A plataforma deve permitir que organizadores de eventos cadastrem seus eventos na plataforma, incluindo informações como nome do evento, localização, data e horário, para que possam iniciar o processo de venda de ingressos.

RF003 - Gestão de Tipos de Ingressos:

Capacitar os organizadores a criarem diferentes tipos de ingressos para cada evento, como ingressos regulares e meia-entrada.

RF004 - Processo de Compra Seguro:

Desenvolver um processo de compra de ingressos seguro e intuitivo para os clientes, com opções de pagamento online através de cartão de crédito ou pix.

RF005 - Emissão de Ingressos Eletrônicos:

Emitir ingressos eletrônicos para os clientes após a conclusão da compra, facilitando o controle de acesso e reduzindo a necessidade de filas.

RF006 - Relatórios de Vendas:

Fornecer relatórios detalhados sobre as vendas de ingressos, incluindo dados como número de ingressos vendidos, receita gerada,

padrões de compra, permitindo aos organizadores avaliar o desempenho do evento e tomar decisões estratégicas informadas.

RF007 - Cadastro de Usuários:

Permitir que os usuários se cadastrem na plataforma, fornecendo informações como nome, endereço de e-mail e senha, para criar uma conta pessoal, incluindo seu tipo, organizador ou cliente.

RF008 - Login:

Permitir que os usuários façam login na plataforma utilizando suas credenciais cadastradas, para acessar funcionalidades exclusivas, como criação de eventos (para Organizadores), busca de ingressos, compra e visualização de eventos cadastrados.

2.2 Requisitos não funcionais (NF)

RNF001 - Desempenho Responsivo

Garantir que o sistema seja responsivo e ofereça tempos de resposta rápidos, mesmo durante períodos de pico de tráfego, para proporcionar uma experiência do usuário fluida e sem atrasos perceptíveis.

RNF002 - Segurança dos Dados

Assegurar que todas as informações dos usuários, transações financeiras e dados do evento sejam protegidos por medidas robustas de segurança, como criptografia SSL, para prevenir acesso não autorizado e garantir a confidencialidade e integridade dos dados.

RNF003 - Escalabilidade

Garantir que o sistema seja capaz de lidar com um aumento repentino no número de usuários e eventos cadastrados, sem

comprometer o desempenho ou a disponibilidade do serviço, permitindo que a plataforma cresça de forma eficiente conforme a demanda.

RNF004 - Usabilidade Intuitiva

Projetar uma interface de usuário intuitiva e de fácil utilização, com navegação clara e instruções concisas, para garantir que os usuários possam realizar suas tarefas de forma eficiente e sem a necessidade de treinamento extensivo.

RNF005 - Disponibilidade Contínua

Garantir uma alta disponibilidade do sistema, com tempo de inatividade mínimo, por meio de redundância de servidores, monitoramento constante e prontidão para lidar com falhas de hardware ou software de forma rápida e eficaz.

RNF006 - Manutenção Simples:

Projetar o sistema de forma modular e bem documentada, com código limpo e estruturado, para facilitar a manutenção e atualizações futuras, permitindo que novos recursos sejam adicionados com facilidade e mínima interrupção do serviço.

3. Cenário

3.1 Criação de evento pelo Organizador

O organizador está autenticado no sistema e possui permissão para criar eventos.

1. O Organizador acessa a opção de criar um novo evento no sistema.
2. O Sistema exibe um formulário para preenchimento dos detalhes do evento.
3. O Organizador preenche todos os campos obrigatórios, incluindo nome, descrição, data, localização, preço do ingresso, e capacidade máxima de participantes.
4. O Organizador confirma a criação do evento.
5. O Sistema armazena as informações do evento no banco de dados.

3.2 Compra de Ingresso realizada pelo Cliente

O Cliente realiza o processo de autenticação através do Login. Tendo ingressos disponíveis para o evento desejado procede nos seguintes passos.

1. O Cliente visualiza a lista de eventos disponíveis no sistema.
2. O Cliente seleciona o evento desejado para o qual deseja comprar ingressos.
3. O Sistema exibe informações detalhadas sobre o evento, incluindo descrição, data, localização e preço do ingresso.
4. O Cliente seleciona a opção de comprar ingressos para o evento.
5. O Sistema solicita que o Cliente forneça as informações necessárias, como o número de ingressos desejados e informações de pagamento.
6. O Cliente fornece as informações solicitadas e confirma a compra.
7. O Sistema processa o pagamento de forma segura.
8. Após a confirmação do pagamento, o Sistema emite os ingressos para o Cliente.

4. Planejamento

O trabalho foi iniciado para a disciplina Projeto Integrador a partir das aulas sobre o paradigma da Orientação de Objetos e propôs-se criar uma solução de software para Desktop para atender a necessidade de cadastro de eventos e compra de ingressos. A partir da definição do tema pelo professor em paralelo com as aulas de POO começamos, a partir de abril, criando um cronograma e definindo os papéis de cada integrante.

Em algumas reuniões com os integrantes do grupo definimos algumas classes principais e as relações para criarmos as estruturas em Java. Buscamos ainda nos ater aos enunciados propostos pelo professor, e priorizar os conceitos fundamentais da orientação a objetos, buscando na UML e nos diagramas compreender as melhores formas de implementação na linguagem e nas IDEs que utilizamos.

Encontramos dificuldades para iniciar a codificação e a relação com o banco de dados MySQL. Buscamos ajuda em tutoriais e no material das aulas do semestre, que foi de grande valia para criarmos a lógica, as telas e, principalmente, a implantação dos conceitos fundamentais de OO encapsulamento, herança e polimorfismo.

A criação de cronograma e EAP são ferramentas importantes que nos ajudaram, de certa maneira, na organização do trabalho coletivo, apesar das discordâncias, debates, testes, erros e acertos, contribuíram para não perdemos o foco na entrega do software funcionando corretamente.

5. Solução de Software

A Plataforma de Venda de Ingressos é um sistema desenvolvido para simplificar o processo de organização de eventos, permitindo aos organizadores cadastrarem e vender ingressos de forma eficiente. Além disso, oferece aos clientes uma experiência de compra segura e conveniente, contribuindo para o sucesso e a satisfação tanto dos organizadores quanto dos participantes dos eventos.

No escopo do projeto foi definido que o sistema será desenvolvido na linguagem Java utilizando a biblioteca Swing para criar interfaces gráficas e no paradigma da Orientação a Objetos (POO).

O sistema permitirá a autenticação de usuários, garantindo que apenas usuários autorizados tenham acesso às funcionalidades.

Os eventos poderão ser de diferentes tipos, como conferências, shows, palestras, entre outros.

Os clientes poderão visualizar eventos por categoria, data e localização.

A venda de ingressos será realizada de forma segura, utilizando métodos de pagamento via cartão de crédito ou pix.

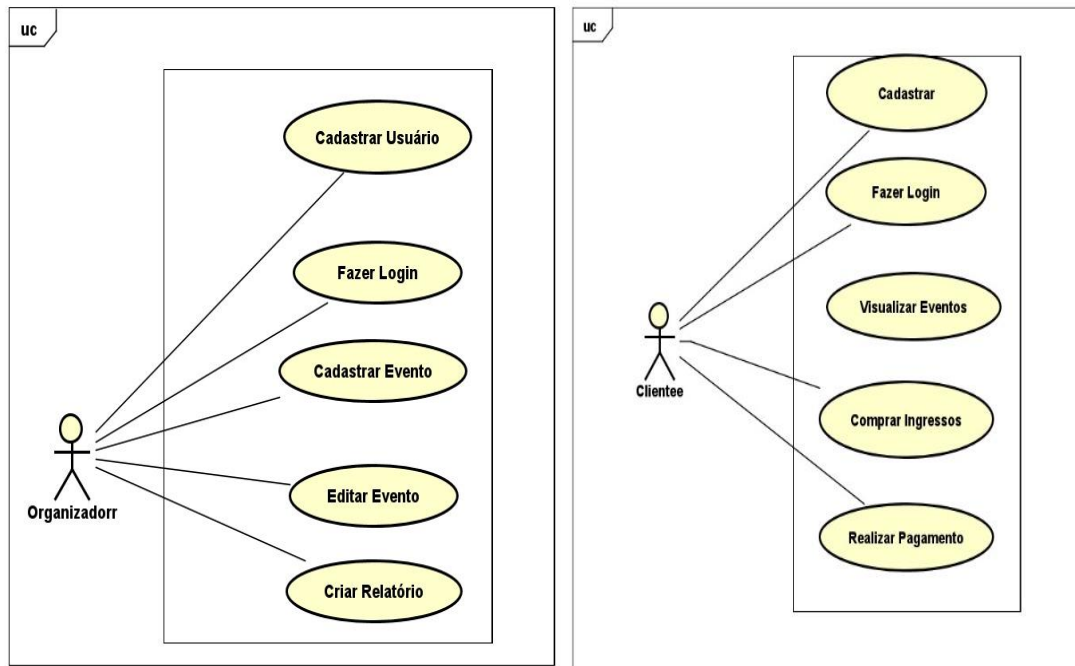
Os organizadores poderão gerar relatórios personalizados sobre vendas de ingressos, participação, entre outros aspectos relevantes.

Utilizamos as IDEs Netbeans e VScode para criar os algoritmos e a estrutura MVC do projeto.

A implementação se dá em máquina local em tempo de execução e com acesso e registro no banco de dados MySQL e servidor local.

A implementação de segurança se dá na codificação com a implementação do encapsulamento dos atributos nas classes.

6. Casos de Uso



6.1 Descrição dos Casos de Uso

6.1.1 Criar Evento:

Descrição: Permite que um organizador crie um evento, inserindo todas as informações relevantes.

Ator Principal: Organizador

1. O organizador seleciona a opção de criar um evento.
2. O sistema exibe um formulário para preenchimento dos detalhes do evento, como nome, descrição, data, localização, preço do ingresso etc.
3. O organizador preenche todos os campos obrigatórios.
4. O organizador confirma a criação do evento.
5. O sistema armazena as informações do evento no banco de dados.

6.1.2 Editar Evento:

Descrição: Permite que um organizador edite as informações de um evento já cadastrado.

Ator Principal: Organizador

1. O organizador seleciona o evento que deseja editar.
2. O sistema exibe um formulário pré-preenchido com as informações atuais do evento.
3. O organizador realiza as alterações necessárias nos campos desejados.
4. O organizador confirma as alterações.
5. O sistema atualiza as informações do evento no banco de dados.

6.1.3 Apagar Evento:

Descrição: Permite que um organizador remova um evento cadastrado.

Ator Principal: Organizador

1. O organizador seleciona o evento que deseja apagar.
2. O sistema exibe uma mensagem de confirmação para garantir que o organizador realmente deseja apagar o evento.
3. O organizador confirma a exclusão do evento.
4. O sistema remove todas as informações relacionadas ao evento do banco de dados.

6.1.4 Emitir Relatórios:

Descrição: Permite que um organizador gere relatórios sobre os eventos e vendas de ingressos.

Ator Principal: Organizador

1. O organizador seleciona a opção de emitir relatórios.

2. O sistema exibe opções para tipos de relatórios disponíveis, como vendas por evento, participação, etc.
3. O organizador seleciona o tipo de relatório desejado.
4. O sistema gera o relatório com base nos dados armazenados no banco de dados.
5. O organizador pode visualizar, exportar ou imprimir o relatório conforme necessário.

6.2 Casos de Uso para Clientes

6.2.1. Visualizar Eventos:

Descrição: Permite que um cliente visualize todos os eventos disponíveis na plataforma.

Ator Principal: Cliente

1. O cliente acessa a lista de eventos disponíveis na plataforma.
2. O sistema exibe uma lista de eventos, incluindo detalhes como nome, data, localização e preço do ingresso.
3. O cliente pode navegar pela lista e visualizar os detalhes de cada evento.

6.2.2. Selecionar Evento:

Descrição: Permite que um cliente selecione um evento específico para obter mais informações.

Ator Principal: Cliente

1. O cliente clica em um evento na lista de eventos disponíveis.
2. O sistema exibe informações detalhadas sobre o evento selecionado, incluindo descrição, data, localização e preço do ingresso.
3. O cliente pode decidir se deseja comprar ingressos para o evento ou retornar à lista de eventos.

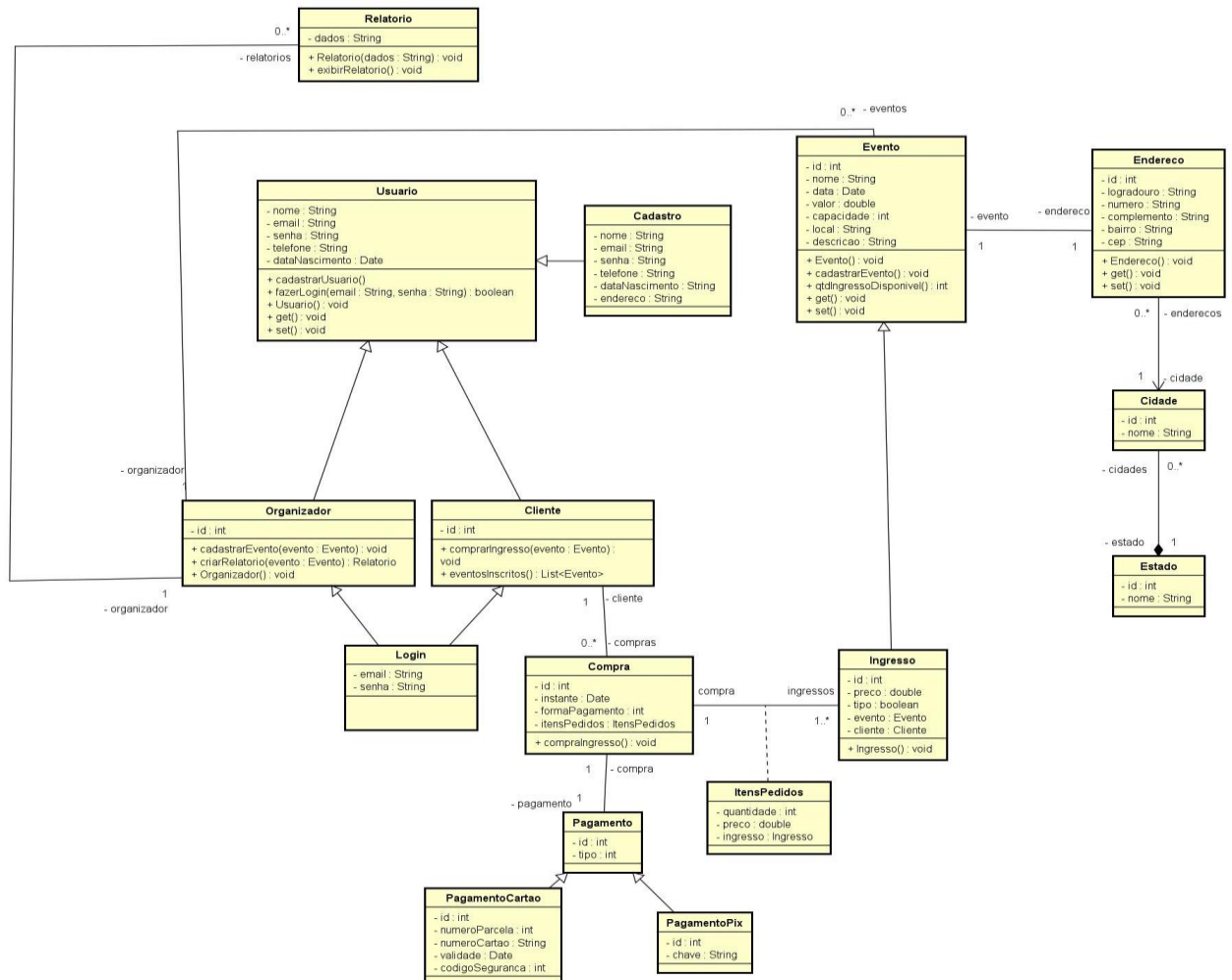
6.2.3. Comprar Ingresso:

Descrição: Permite que um cliente compre ingressos para um evento selecionado.

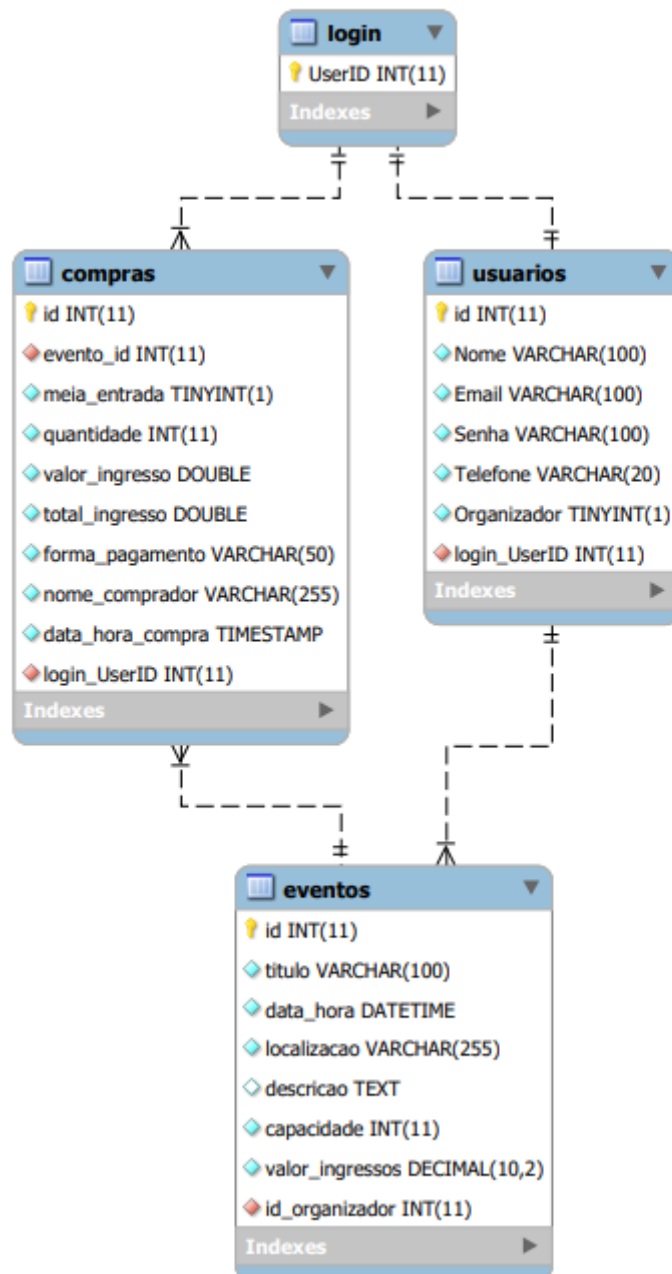
Ator Principal: Cliente

1. O cliente seleciona a opção de comprar ingressos para o evento desejado.
2. O sistema solicita que o cliente faça login em sua conta ou crie uma nova conta, se ainda não tiver uma.
3. O cliente fornece as informações necessárias, como número de ingressos desejados e informações de pagamento.
4. O sistema processa o pagamento de forma segura.
5. Após a confirmação do pagamento, o sistema emite os ingressos para o cliente, que pode ser em forma de código QR ou outro formato apropriado.
6. O sistema atualiza a quantidade de ingressos disponíveis para o evento no banco de dados.

7. Diagrama de Classes



8. Modelo de Dados



9. Conclusão

A realização deste projeto e sua conclusão nos faz refletir sobre a importância da prática para desenvolver um software e também ressaltamos a experiência valiosa adquirida ao enfrentar os desafios e imprevistos ao longo do processo de desenvolvimento neste terceiro semestre, a qual já nos eleva o nível de dificuldade e complexidade. Apesar dos contratempos, como problemas técnicos inesperados e ajustes de escopo, a determinação e o comprometimento da equipe resultaram em um projeto bem-sucedido. A capacidade de superar esses obstáculos não apenas fortaleceu as habilidades técnicas e de gestão do grupo, mas também promoveu o crescimento pessoal e profissional de cada membro da equipe.

Ao implementar as melhores práticas de Desenvolvimento Orientado a Objetos, pudemos experimentar os percalços nos caminhos na área de Engenharia de software e das dificuldades com a implementação dos pilares da programação orientada a objetos.

Esta experiência inicial demonstrou os desafios de aplicar conceitos teóricos em um ambiente prático, no caso, o software para Desktop feito em Java. Com a ajuda de toda a equipe, dos materiais de aula e do esclarecimento de dúvidas, combinada com a orientação e suporte dos professores e mentores, foi fundamental para o término do projeto. Essa experiência proporcionou uma base para os futuros projetos acadêmicos e profissionais que teremos, equipando-os com as habilidades e o conhecimento necessários para enfrentar desafios similares no mundo real.

Enfim, o projeto nos ajudou a refletir e na prática realizar a implementação das regras de negócio, na relação entre as classes e a instanciação de objetos, aplicar os conhecimentos em banco de dados vistos no semestre passado e a praticar a linguagem java e também a criação de interfaces.