

Trabalho Integrado dos componentes de Banco de Dados II, Engenharia de Software I e Programação II

Erick Laufer¹
Fabiano Charnoski²
Gustavo Dalpiaz³
Saulo Benedetti⁴

Resumo

O projeto propõe o desenvolvimento de um sistema web avançado para a agência de eventos "Evento Mania". Além da aplicação de conhecimentos específicos em banco de dados, o projeto incorpora princípios avançados de programação e engenharia de software. Utilizando ferramentas como Github para versionamento do projeto, Trello para fazer os planejamentos, juntamente com Dbeaver, Visual Paradigm, PostgreSQL, Eclipse e Postman, a iniciativa busca criar uma solução robusta, eficiente e escalável. Destaca-se a importância de uma abordagem sólida em engenharia de software, incluindo controle de versionamento, para atender às exigências dinâmicas de uma agência de eventos de destaque.

Palavras-chave: Evento Mania, Github, Trello, Dbeaver, Visual Paradigm, PostgreSQL, Eclipse e Postman

1 INTRODUÇÃO

No contexto dinâmico e desafiador do universo de eventos, a gestão eficiente de informações é crucial para o sucesso e reconhecimento de agências especializadas. Este trabalho tem como propósito a criação de um sistema web avançado dedicado à renomada agência de eventos "Evento Mania", amplamente reconhecida em todo o território brasileiro. O desafio envolve não apenas a aplicação prática de conhecimentos aprofundados em banco de dados, mas também a integração de princípios avançados de programação e engenharia de software.

A implementação do sistema foi conduzida utilizando ferramentas de ponta, como Github para versionamento, Dbeaver e Visual Paradigm para modelagem, Postgres como Sistema de Gerenciamento de Banco de Dados (SGBD), juntamente com o Eclipse como ambiente de desenvolvimento. Essa escolha estratégica visa não apenas atender aos requisitos funcionais, mas também garantir a robustez, eficiência e escalabilidade da solução.

Além disso, para assegurar a qualidade e testabilidade do sistema, foi incorporada a ferramenta Postman para realizar testes de requisições, permitindo uma validação rigorosa das funcionalidades implementadas. A utilização do Postman facilitou a identificação de possíveis falhas e garantiu que o sistema esteja alinhado com as expectativas da agência "Evento Mania".

No âmbito do planejamento e acompanhamento das atividades, foi adotada a ferramenta Trello. Esta plataforma colaborativa proporcionou uma visão clara e organizada do progresso do projeto, permitindo a atribuição de tarefas, o acompanhamento do cronograma e a comunicação eficiente entre os membros da equipe. A integração do Trello no processo de gerenciamento reforça a abordagem metódica adotada para atender às necessidades dinâmicas e desafios complexos enfrentados por agências de destaque no cenário de eventos.

A relevância deste trabalho reside na aplicação prática de conhecimentos adquiridos, proporcionando à agência "Evento Mania" uma ferramenta tecnologicamente avançada para otimizar a gestão de seus eventos. O foco na engenharia de software, aliado ao controle de versionamento, destaca a abordagem metódica adotada para atender às necessidades dinâmicas e desafios complexos enfrentados por agências de destaque no cenário de eventos.

2 MATERIAIS E MÉTODOS

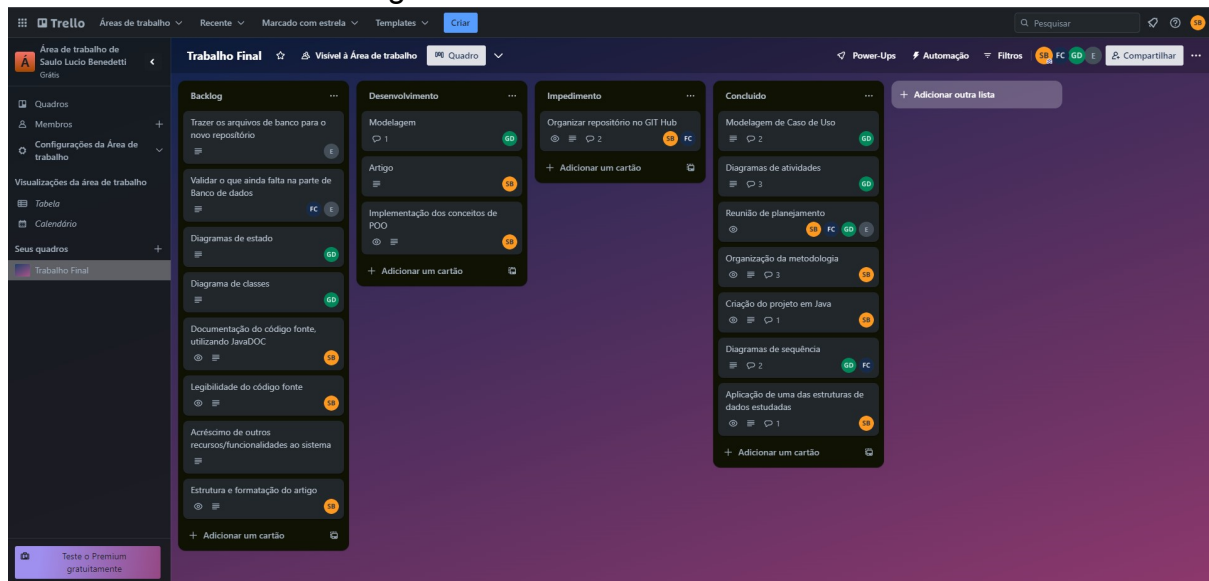
Para o desenvolvimento desse sistema, foi utilizado o banco de dados relacional PostgreSQL, já desenvolvido anteriormente, juntamente com a ferramenta Dbeaver. Na parte de programação foi utilizada a linguagem Java juntamente com a ferramenta Eclipse como ambiente de desenvolvimento para fazer a parte do back-end com Spring Boot e a biblioteca Lombok Java, com o intuito de fazer a conexão com o banco de dados, para fazer os testes das requisições foi utilizado o Postman. Na parte de front-end foi utilizado HTML com Bootstrap e JavaScript. No desenvolvimento dos modelos foram utilizados os requisitos já levantados

anteriormente junto com a ferramenta Visual paradigm. Para fazer o versionamento do projeto foi utilizado o GitHub. Na execução do planejamento das atividades e das divisões de tarefas foi utilizado a ferramenta Trello.

3 DESENVOLVIMENTO

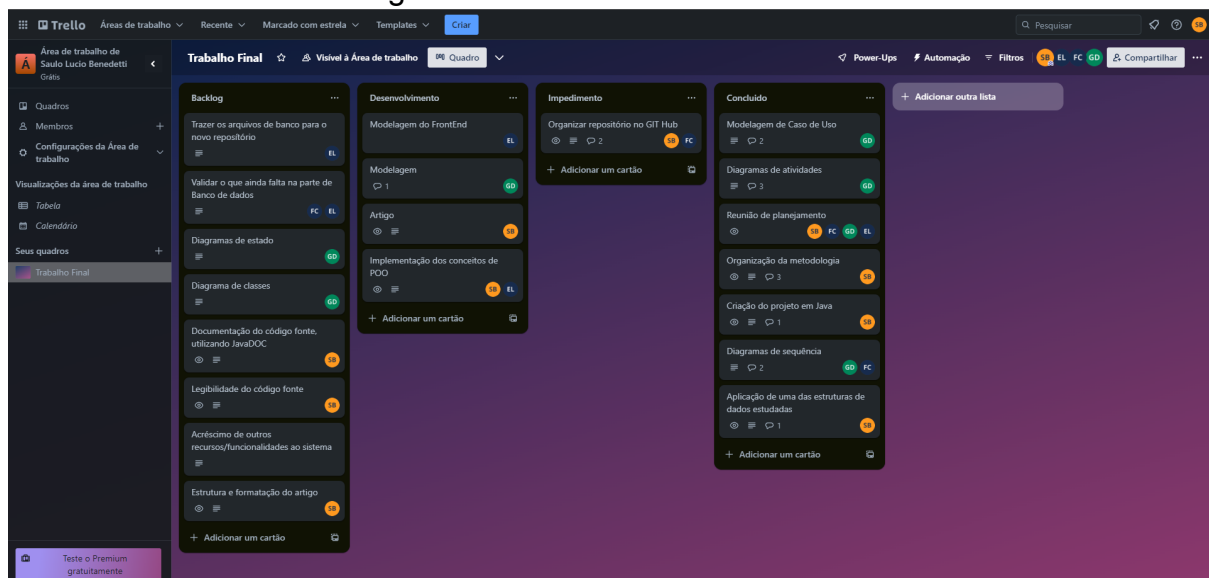
Inicialmente, foi utilizada a ferramenta Trello criando um KANBAN para separar as tarefas que precisavam ser feitas e distribuir entre os integrantes. As figuras 1, 2 e 3 mostram como ficou o KANBAN ao passar do tempo.

Figura 1: KANBAN das atividades.



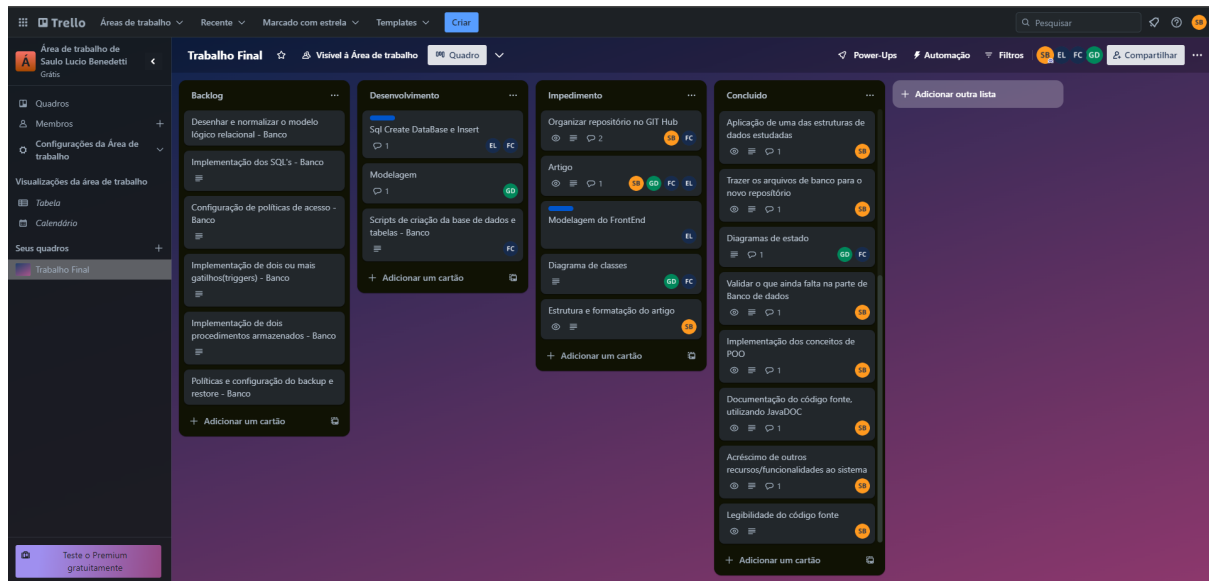
Fonte: Os autores (2023).

Figura 2: KANBAN das atividades.



Fonte: Os autores (2023).

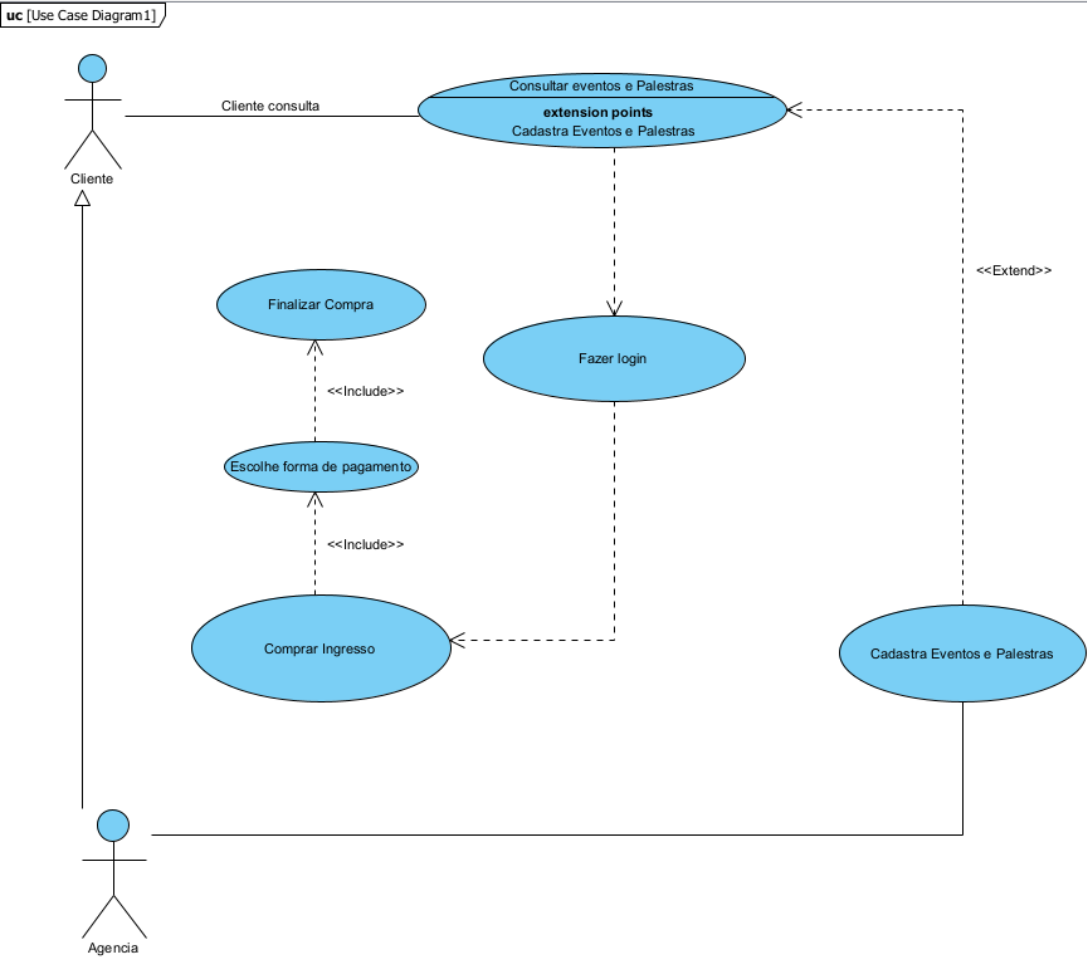
Figura 3: KANBAN das atividades.



Fonte: Os autores (2023).

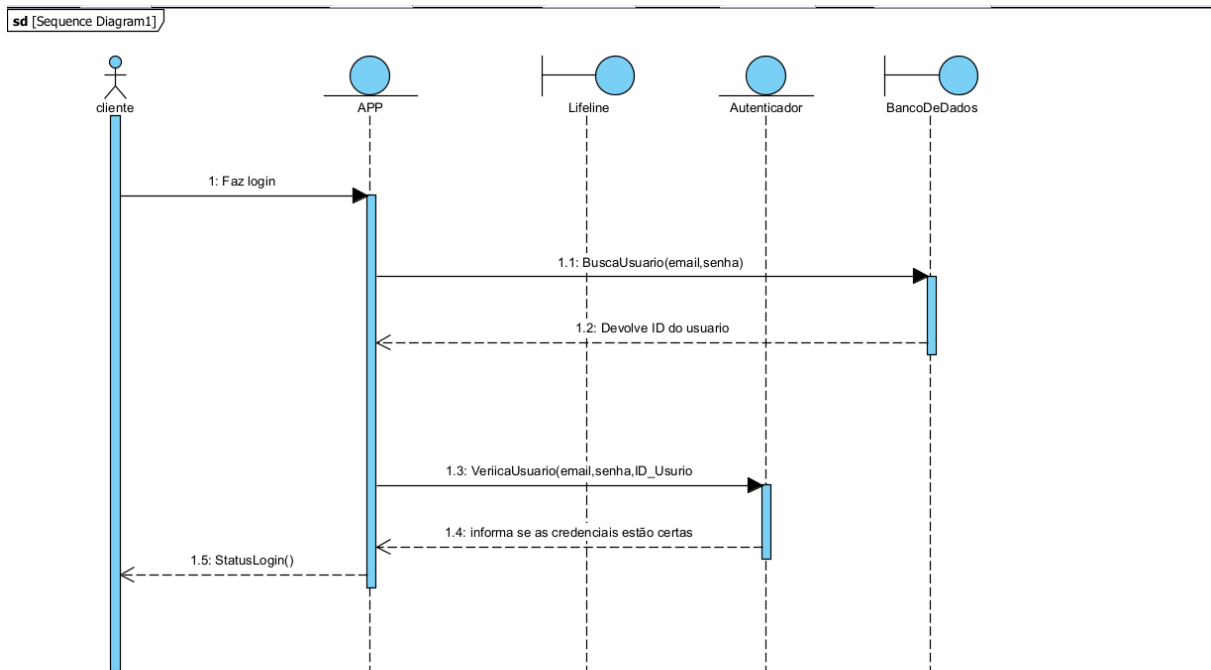
A partir dos requisitos já levantados, foi iniciada a construção do diagrama de casos de uso e dos diagramas de sequência, de atividade, de estado e de classes utilizando a ferramenta Visual Paradigm. Abaixo as figuras 4, 5, 6, 7 e 8 mostram os diagramas.

Figura 4: Diagrama de casos de uso.



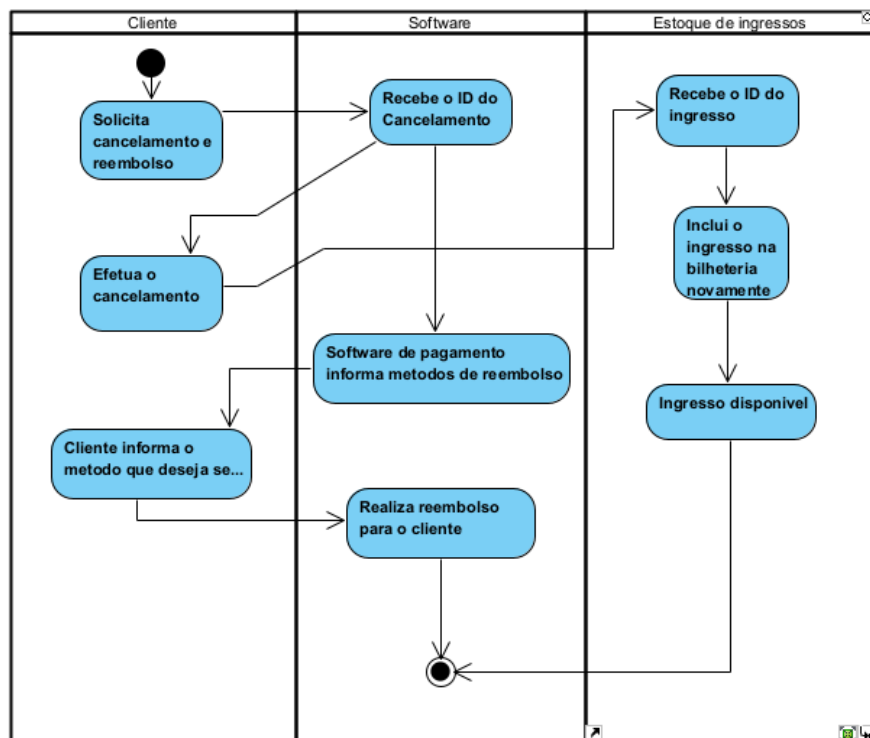
Fonte: Os autores (2023).

Figura 5: Diagrama de sequência.



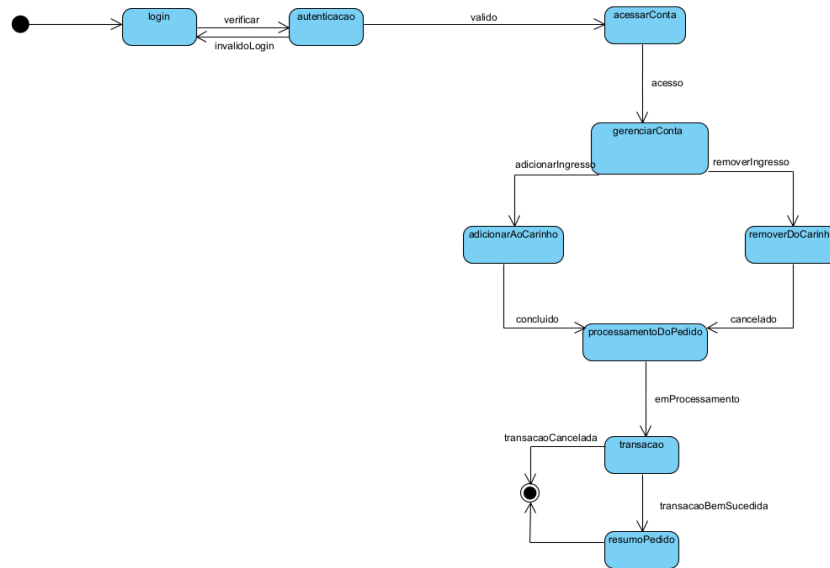
Fonte: Os autores (2023).

Figura 6: Diagrama de atividade.



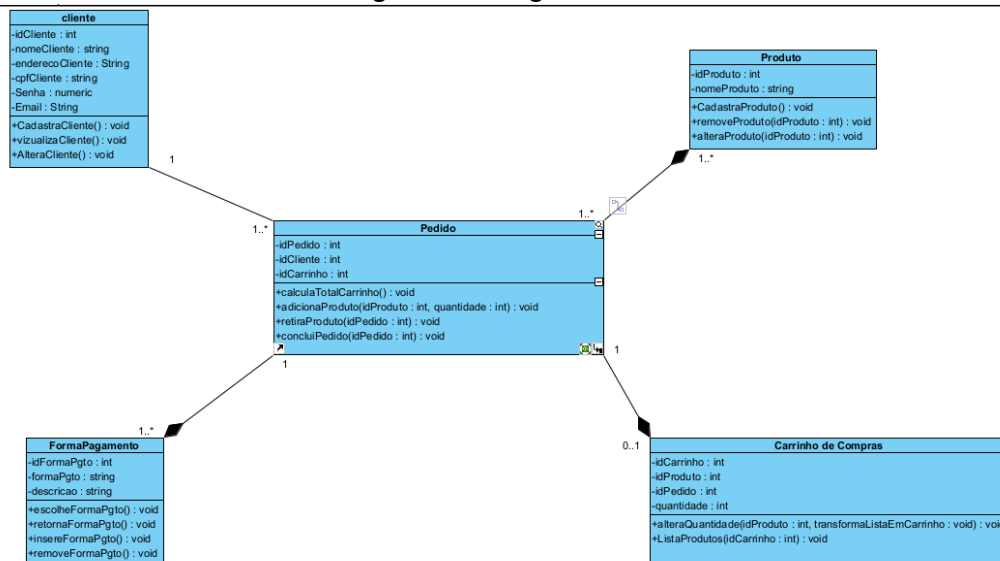
Fonte: Os autores (2023).

Figura 7: Diagrama de estado.



Fonte: Os autores (2023).

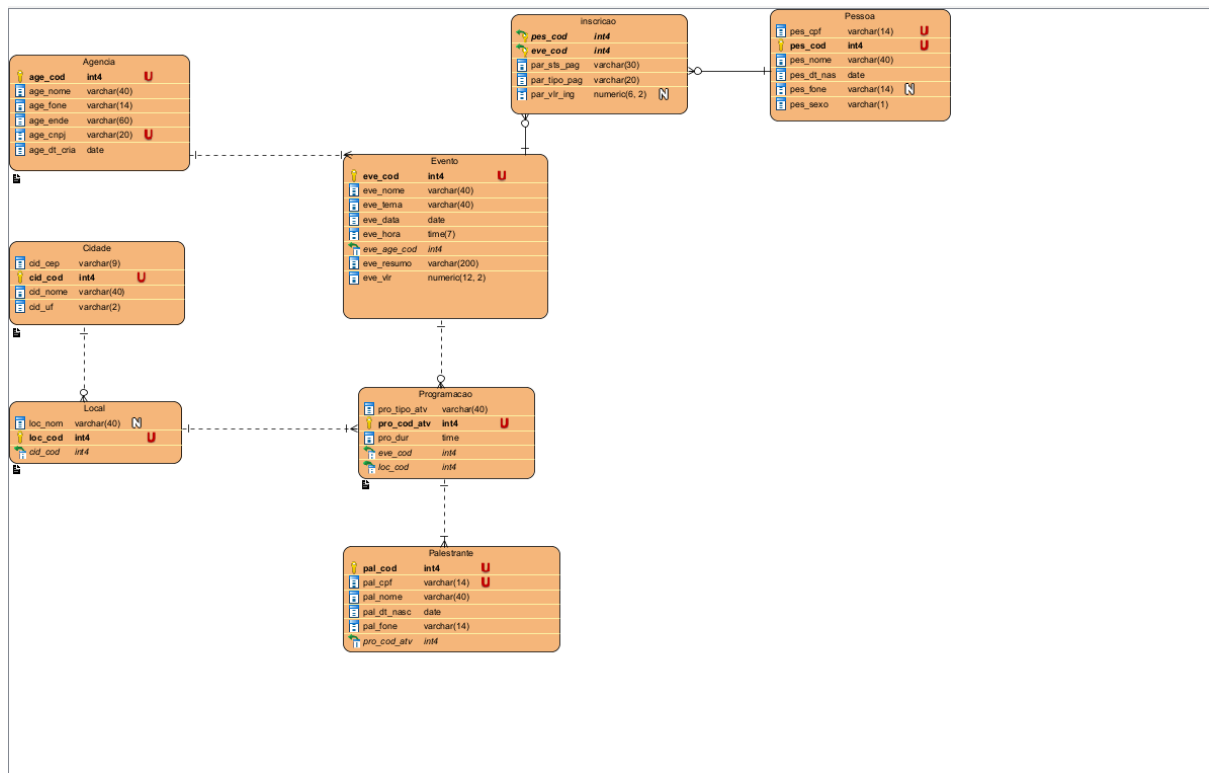
Figura 8: Diagrama de classes.



Fonte: Os autores (2023).

Em consequência da utilização do banco de dados já desenvolvido, apenas foram feitos ajustes para deixar com maior legibilidade. Dessa forma também houve o ajuste no diagrama do banco de dados, conforme mostra a figura 9.

Figura 9: Diagrama do banco de dados.

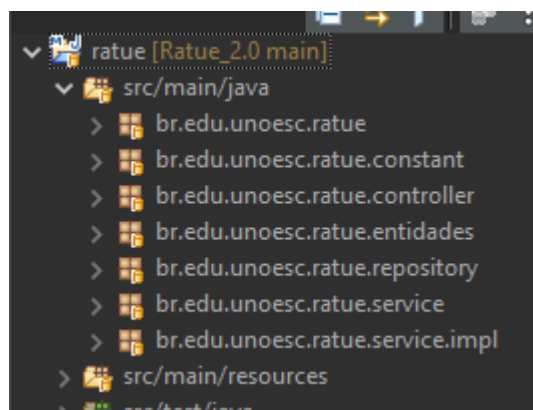


Fonte: Os autores (2023).

Paralelamente a isso iniciou a parte de banco de dados, fazendo ajustes na nomenclatura de uma tabela e ajustando o tipo de uma coluna, também ajustando o script de insert do banco de dados, além disso foram criadas políticas de acesso, foram implementados dois gatilhos, implementados dois procedimentos armazenados para controle das regras de negócio do sistema e foram criados políticas e configuração do backup e restore.

Após isso se deu início ao desenvolvimento do back-end no Eclipse em Java com o Spring Boot junto com Lombok. A estrutura do projeto foi feita da seguinte forma, foi criado “packages” para fazer a divisão das classes e interfaces, deixando o projeto com mais legibilidade. A figura 10 mostra essa divisão.

Figura 10: Organização dos pacotes.



Fonte: Os autores (2023).

No pacote `br.edu.unoesc.ratue` se encontra a classe principal do projeto, no pacote `br.edu.unoesc.ratue.constant` foi criada uma classe `StringConstant` que possui todas as Strings constantes utilizadas no projeto, no pacote `br.edu.unoesc.ratue.controller` foi criada a classe `PessoaController` controladora para a entidade `Pessoa`, no pacote `br.edu.unoesc.ratue.entidades` foi criada 5 classes representando entidades do banco de dados, no pacote `br.edu.unoesc.ratue.repository` foi criada a interface `PessoaRepository` que estende a classe `JpaRepository`, no pacote `br.edu.unoesc.ratue.service` foi criada a interface `PessoaService`, no pacote `br.edu.unoesc.ratue.service.impl` foi criada a classe `PessoaServiceImpl` que faz a implementação das funcionalidades da interface `PessoaService`. Para fazer a conexão com o banco de dados houve a configuração do arquivo `application.properties` como mostra a figura 11.

Figura 11: Configuração do arquivo `application.properties`.

```
1 server.servlet.context-path =/ratue
2 server.address=0.0.0.0
3 server.port=8000
4 spring.h2.console.enabled=true
5 spring.h2.console.path=/h2-console
6 spring.datasource.url= jdbc:postgresql://localhost:5432/ratue
7 spring.datasource.username=postgres
8 spring.datasource.password=postgres
9 spring.jpa.hibernate.ddl-auto=none
10 spring.jpa.properties.hibernate.jdbc.lob.non_contextual_creation=true
11
```

Fonte: Os autores (2023).

Foram desenvolvidas 5 requisições para a entidade `Pessoa`, uma para salvar uma pessoa no banco, uma para deletar uma pessoa pelo seu código, uma para atualizar os dados da pessoa, uma para buscar a pessoa pelo nome e outra que busca todas as pessoas. Foi adicionado um tratamento de Exceptions para caso aconteça alguma exceção ao criar, deletar ou atualizar a pessoa, o sistema não pare. Para testar essas requisições foi utilizado a ferramenta Postman.

A parte do front-end foi desenvolvida utilizando HTML, Bootstrap e JavaScript. A figura 12, 13 e 14 mostra o resultado final.

Figura 12: Página inicial do site.

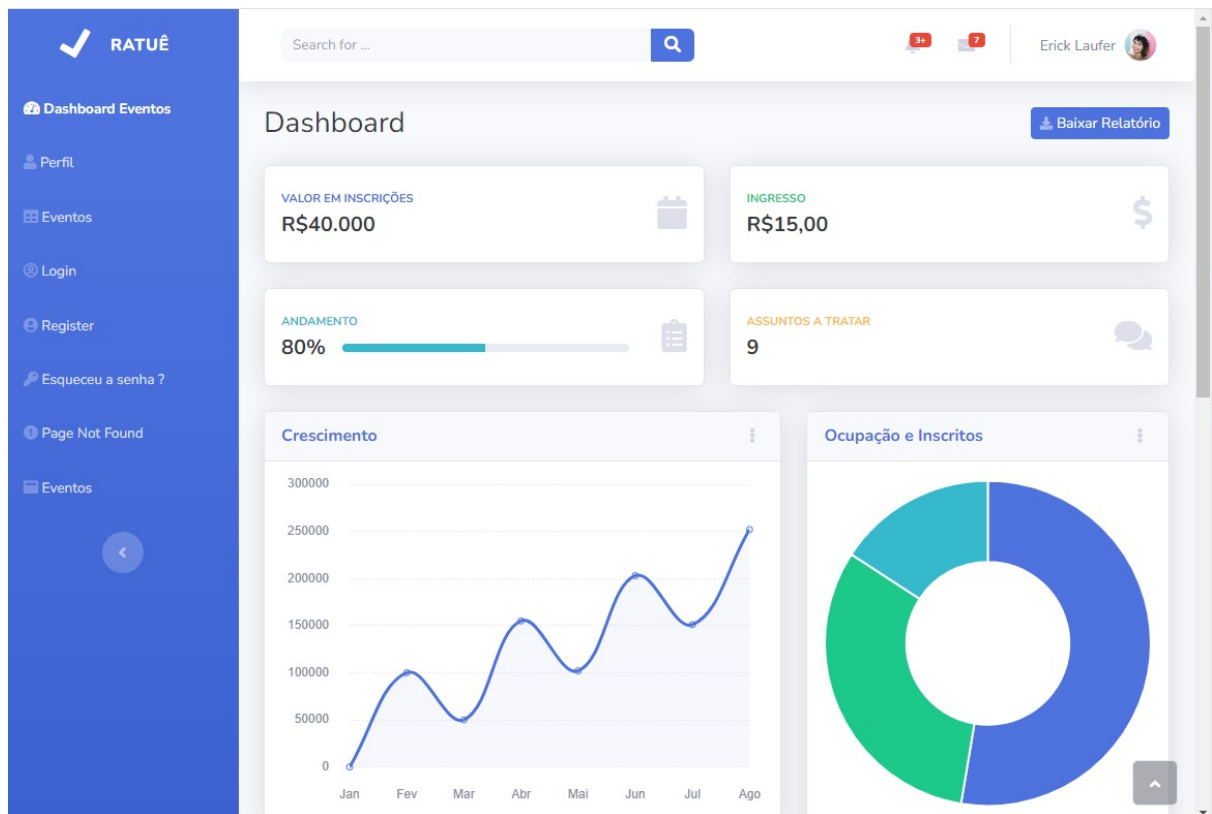
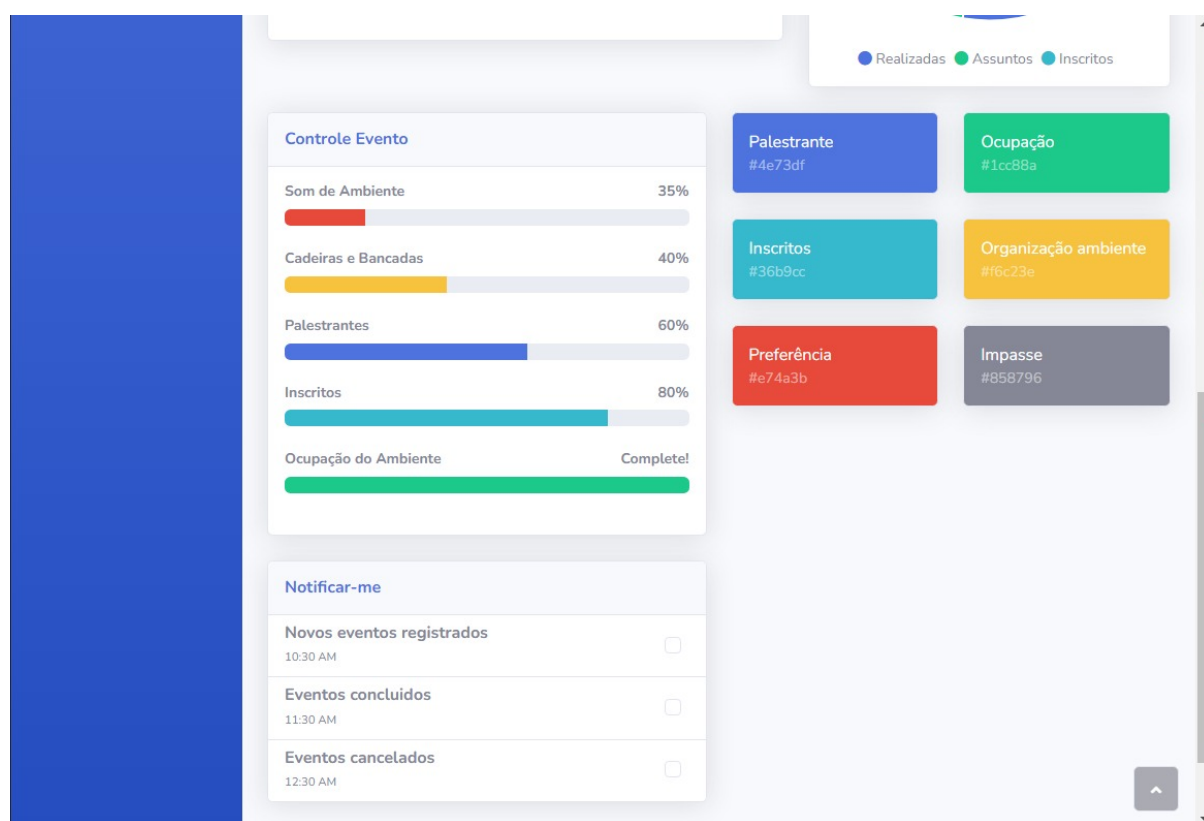


Figura 13: Página de estatísticas dos eventos.



Fonte: Os autores (2023).

Figura 14: Página de cadastro de eventos.

Palestrante	Assunto	Local	Data	Início	Valor de inscrição
Airi Satou	Accountant	Tokyo	33	2008/11/28	\$162,700
Angelica Ramos	Chief Executive Officer(CEO)	London	47	2009/10/09	\$1,200,000
Ashton Cox	Junior Technical Author	San Francisco	66	2009/01/12	\$86,000
Bradley Greer	Software Engineer	London	41	2012/10/13	\$132,000
Brenden Wagner	Software Engineer	San Francisco	28	2011/06/07	\$206,850
Brielle Williamson	Integration Specialist	New York	61	2012/12/02	\$372,000
Bruno Nash	Software Engineer	London	38	2011/05/03	\$163,500
Caesar Vance	Pre-Sales Support	New York	21	2011/12/12	\$106,450
Cara Stevens	Sales Assistant	New York	46	2011/12/06	\$145,600
Cedric Kelly	Senior JavaScript Developer	Edinburgh	22	2012/03/29	\$433,060

Fonte: Os autores (2023).

4 CONCLUSÃO

O desenvolvimento deste projeto resultou em um sistema web avançado, alinhado com os objetivos propostos para a agência de eventos "Evento Mania". A integração dos conhecimentos em Banco de Dados II, Engenharia de Software I e Programação II foi essencial para o sucesso da iniciativa, proporcionando uma solução tecnologicamente avançada e adaptável às demandas dinâmicas do setor de eventos. A escolha estratégica de ferramentas como GitHub para versionamento, Dbeaver e Visual Paradigm para modelagem, PostgreSQL como SGBD e Eclipse para o ambiente de desenvolvimento foi fundamental para garantir a robustez e eficiência da solução. O uso de Java no back-end, juntamente com Spring Boot e Lombok, contribuiu para uma implementação coesa e de fácil manutenção, enquanto o front-end foi desenvolvido utilizando HTML com Bootstrap e JavaScript, proporcionando uma experiência de usuário intuitiva e moderna.

4 REFERÊNCIAS

ALVES, Roberson J. F. **Apostila de Banco de Dados**. São Miguel do Oeste: Unoesc, 2023. Material didático em PDF.

BARBOSA, Otilia D. **Spring Boot API**. São Miguel do Oeste: Unoesc, 2023. Material didático em PDF.

BARBOSA, Otilia D. **Parte 3 - Anotações, JavaDoc e JUnit**. São Miguel do Oeste: Unoesc, 2023. Material didático em PDF.

BOOTSTRAP. Bootstrap: **The world's most popular framework for building responsive, mobile-first sites**. Disponível em: <https://getbootstrap.com/>. Acesso em: 14 de novembro de 2023.

PETRY, Franciele C. **Modelagem UML: DIAGRAMA DE CLASSES**. 2023. 26 slides. Apresentação de slides.

PETRY, Franciele C. **Modelagem UML: DIAGRAMA DE ATIVIDADE E DE ESTADOS**. 2023. 35 slides. Apresentação de slides.

PETRY, Franciele C. **Modelagem UML: DIAGRAMA DE SEQUÊNCIA**. 2023. 34 slides. Apresentação de slides.

PETRY, Franciele C. **ANÁLISE E PROJETO OO COM UML: INTRODUÇÃO E CASOS DE USO**. 2023. 34 slides. Apresentação de slides.