

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
CAMPUS CAJAZEIRAS
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS**

**EVENTOSERVICE: APLICAÇÃO PARA AUXÍLIO À ORGANIZAÇÃO
DE EVENTOS ACADÊMICOS**

JOSÉ FERREIRA VIEIRA

**Cajazeiras
2021**

JOSÉ FERREIRA VIEIRA

**EVENTOSERVICE: APLICAÇÃO PARA AUXÍLIO À ORGANIZAÇÃO DE EVENTOS
ACADÊMICOS**

Trabalho de Conclusão de Curso apresentado junto ao Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba - Campus Cajazeiras, como requisito à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador

Prof. Me. Francisco Paulo de Freitas Neto.

**Cajazeiras
2021**

V658e

VIEIRA, José Ferreira
EVENTOSERVICE: Aplicação para auxílio à organização
de eventos acadêmicos. José Ferreira Vieira. - Cajazeiras, 2021. 70f..

TCC (PDF)

Orientador: Francisco Paulo de Freitas Neto.

1. Soft 2. nosql. 3. CMS. I. José Ferreira Vieira. II. Título.

CDU: 004.43



Às **16:00** horas do dia **01** do mês de **julho** do ano de **2021**, via Google Meet, compareceu para defesa pública do **Trabalho de Conclusão de Curso**, requisito obrigatório para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas, o(a) aluno(a) **JOSE FERREIRA VIEIRA**, matrícula **201312010100**, tendo como Título do Trabalho **EVENTOSERVICE: APLICAÇÃO PARA AUXÍLIO À ORGANIZAÇÃO DE EVENTOS ACADÊMICOS**. Constituíram a Banca Examinadora os professores **FRANCISCO PAULO DE FREITAS NETO** (orientador), **FABIO GOMES DE ANDRADE** (examinador) e **MICHEL DA SILVA** (examinador).

Após a apresentação e as observações dos membros da Banca Examinadora, ficou definido que o trabalho foi considerado **APROVADO** com nota **90**, com a condição de que o (a) aluno (a) entregue, no prazo máximo de 30 dias, a versão final do trabalho, via processo eletrônico à coordenação de curso. A versão deve conter a ficha catalográfica e atender às sugestões feitas pelos membros da banca. O código fonte desenvolvido no trabalho (caso haja) deve ser enviado para o e-mail da coordenação do curso (cads.cz@ifpb.edu.br).

Cajazeiras-PB, 1 de julho de 2021.

Documento assinado eletronicamente por:

- Michel da Silva, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 01/07/2021 22:56:37.
- Fabio Gomes de Andrade, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 01/07/2021 19:44:27.
- Jose Ferreira Vieira, ALUNO (201312010100) DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - CAJAZEIRAS, em 01/07/2021 19:11:24.
- Francisco Paulo de Freitas Neto, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 01/07/2021 18:22:01.

Este documento foi emitido pelo SUAP em 01/07/2021. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código Verificador: 202633

Código de Autenticação: f2bb217c4a



Dedico todo esforço que tive, para finalizar esse trabalho, ao meu querido pai Francisco Das Chagas Vieira (in memorian) que não pôde vivenciar esse momento, mas que batalhou ao meu lado até a sua partida;

AGRADECIMENTOS

Serei eternamente grato a Deus por tê-lo em meu coração e por ter me proporcionado as maiores alegrias como a conclusão deste trabalho. Sem ele, nada sou;

Agradeço a toda minha família pelo apoio Incondicional de sempre;

Ao meu orientador, Professor Mestre Francisco Paulo de Freitas Neto, que tornou possível a realização deste trabalho;

À Direção, aos docentes, aos técnicos administrativos e aos discentes do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba, Campus Cajazeiras na Cidade de Cajazeiras – PB;

A todos que, direta ou indiretamente, contribuíram para a realização desta pesquisa.

RESUMO

Os eventos acadêmicos são um importante espaço cultural e de evolução da sociedade através da troca de conhecimento entre pesquisadores e interessados em uma determinada área. Diante da necessidade da criação de uma identidade visual(site) a cada novo evento do IFPB Campus Cajazeiras e cobrança de taxa por parte de ferramentas similares no mercado para usar a funcionalidade inscrições pagas, este trabalho vem propor o EventoService, uma ferramenta de auxílio à gestão e divulgação de eventos acadêmicos, com intuito de facilitar o trabalho de organizadores e toda a equipe envolvida na produção. O EventoService oferece aos seus usuários a possibilidade de divulgar seu evento acadêmico de forma ágil e prática sem a necessidade do auxílio de profissionais de tecnologia. O seu diferencial em relação às ferramentas semelhantes do mercado é não cobrar taxa sobre o valor das inscrições. Para tanto, o sistema foi desenvolvido sobre os conceitos de *Content Management System* (CMS), utilizando tecnologias mais abertas e adaptáveis a mudanças, fazendo com que o sistema atinja seu objetivo e ainda assim, esteja aberto a novas mudanças que venham a ser necessárias ao longo do tempo, de acordo com as necessidades dos seus usuários.

Palavras-chave: Eventos acadêmicos, nosql, CMS, Rest.

ABSTRACT

Academic events are an important cultural space and the evolution of society through the exchange of knowledge between researchers and those interested in a particular area. Given the need to create a visual identity (website) at each new event of the IFPB Campus Cajazeiras and charge a fee by similar tools in the market to use the paid registration functionality, this work proposes the EventService, a tool to help management and dissemination of academic events, in order to facilitate the work of organizers and the entire team involved in production. EventService offers its users the possibility of publicizing their academic event in an agile and practical way without the need for the help of technology professionals. Its differential in relation to similar tools on the market is that it does not charge a fee on the registration fee. To this end, the system was developed based on the concepts of *Content Management System* (CMS), using technologies that are more open and adaptable to change, making the system achieve its goal and still be open to new changes that may come to be needed over time, according to the needs of its users.

Keywords: Academic Events, Management, nosql, rest

LISTA DE FIGURAS

Figura 1 – Visão geral do funcionamento de um CMS	18
Figura 2 – Visão geral banco chave valor	22
Figura 3 – Estrutura banco orientado a documento	23
Figura 4 – Visão geral banco orientado a grafo	24
Figura 5 – Fluxo do easyProcess	27
Figura 6 – Diagrama de caso de uso para os atores da aplicação	35
Figura 7 – Modelo arquitetural	36
Figura 8 – Diagrama de componentes	39
Figura 9 – Modelo conceitual	40
Figura 10 – Tela inicial visão web	41
Figura 11 – Tela de apresentação do evento visão web	42
Figura 12 – Tela de apresentação do evento visão web	42
Figura 13 – Tela de login versão WEB	43
Figura 14 – Modal Cadastro de usuário versão WEB	44
Figura 15 – Tela de home visão WEB	45
Figura 16 – Tela para criar evento na aplicação	46
Figura 17 – Tela para o gerenciamento do evento	47
Figura 18 – Tela de principal	48
Figura 19 – Continuação tela de principal	49
Figura 20 – Tela de login	50
Figura 21 – Tela de cadastro	51
Figura 22 – Tela de detalhe do evento	65
Figura 23 – Continua tela de detalhe do evento	66
Figura 24 – Tela de credenciamento no evento	67
Figura 25 – Tela de perfil versão WEB	68
Figura 26 – Tela de perfil versão WEB	68
Figura 27 – Tela de perfil aplicativo	69

LISTA DE QUADROS

Quadro 1 – Definição de Papéis	30
Quadro 2 – Plano de Iteração	30
Quadro 3 – Funcionalidades das ferramentas	31
Quadro 4 – User Stories levantadas	32
Quadro 5 – Alocação de Tarefas - US03 US04 US09 US10	33
Quadro 6 – Teste de aceitação - US03 US04 US09 US10	34
Quadro 7 – Resumo dos usuários	59
Quadro 8 – Definição das funcionalidades	60
Quadro 9 – Alocação de Tarefas - US01 US02	61
Quadro 10 – Teste de aceitação - US01 US92	62
Quadro 11 – Alocação de Tarefas - US05 US06 US07	63
Quadro 12 – Teste de aceitação - US05 US96 US07	64

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface (Interface de Programação de Aplicação)
ADS	Análise e Desenvolvimento de Sistemas
BD	Base de dados
BDR	Base de dados relacional
CMS	Content Management Systems (sistema Gerenciador de conteúdo)
DCU	Diagrama de Caso de Uso
ER	Engenharia de Requisitos
HTTP	Hypertext Transfer Protocol (Protocolo de Transferência de Hipertexto)
HTML	HyperText Markup Language (Linguagem de Marcação de Hipertexto)
URI	Identificador de Recurso Uniforme
IFPB	Instituto Federal de Educação, Ciência e Tecnologia da Paraíba
JSOM	Javascript Object Notation
MVC	Model-View-Controller
MCU	Modelo de Caso de Uso
	MER modelo Entidade Relacionamento
NOSQL	NOT ONY SQL
REST	Representational State Transfer (Transferência de Estado Representacional)
SGBD	Sistema de gerenciamento de banco de dados
SGC	Sistema de Gerenciamento de Conteúdo
SQL	Structured Query Language, (Linguagem de Consulta Estruturada)
	TI Tecnologia da Informação
TCC	Trabalho de Conclusão do Curso
UML	Unified Modeling Language (Linguagem de Modelagem Unificada)
YP	easYProcess

SUMÁRIO

1	INTRODUÇÃO	12
1.1	MOTIVAÇÃO	12
1.2	OBJETIVO GERAL	14
1.3	OBJETIVOS ESPECÍFICOS	14
1.4	METODOLOGIA	15
1.5	ORGANIZAÇÃO DO DOCUMENTO	16
2	FUNDAMENTAÇÃO TEÓRICA	17
2.1	CMS	17
2.2	BANCO DE DADOS NoSQL	18
2.2.1	Escalabilidade	20
2.2.2	Esquema flexível ou ausência de esquema	21
2.2.3	Consistência eventual	21
2.2.4	Modelos de Dados NoSQL	22
2.3	REST	25
2.4	processo de desenvolvimento YP	26
2.4.1	YP	27
3	FERRAMENTA	29
3.1	Etapas do do processo	29
3.1.1	Conversa com cliente	29
3.1.2	Inicialização	29
3.1.3	Definição de Papéis	29
3.1.4	Planejamento de <i>Releases</i>	30
3.2	Ferramentas Relacionadas	30
3.3	Análise	31
3.3.1	User Storie	32
3.3.2	Diagrama de casos de uso	35

3.4	Arquitetura	35
3.4.1	Fron end	36
3.4.2	Back end	37
3.4.3	Camada de persistência	38
3.4.4	Diagrama de componente	38
3.4.5	Implantação	39
3.4.6	Modelo conceitual	40
3.5	Implementação	41
4	CONCLUSÃO	52
4.1	Trabalhos futuros	52
	REFERÊNCIAS	54
	APÊNDICE A – DOCUMENTO DE VISÃO	57
A.1	Objetivo	57
A.2	Abrangência do Projeto	57
A.3	Definições, Acrônimos e Abreviações	57
A.4	Organização do Documento	57
A.5	Descrição do Problema	58
A.6	Resumo dos Envolvidos	58
A.7	Resumo dos Usuários	58
A.8	Descrição da Solução Proposta	59
A.9	Funcionalidades	59
A.10	Restrições do Projeto	60
	APÊNDICE B – ALOCAÇÃO DE TAREFAS E TESTES DE ACEITAÇÃO	61
	APÊNDICE C – TELAS DO SISTEMA	65

1 INTRODUÇÃO

Conforme descrito por Schmidt e Ohira (2002), os eventos de caráter científico são tidos como uma maneira para comunicação oral do conhecimento, já que devido à crescente evolução da ciência, esses tipos de evento tornam-se um meio de divulgação e/ou absorção de novos conhecimentos para um público. Para que a disseminação dos conhecimentos produzidos seja difundida socialmente se faz necessário o uso de diversos canais de comunicação. Meadows e Lemos (1999) descrevem que os canais de comunicação científica podem ser formais (que remete a uma existência duradoura, como periódicos, e livros) e informais (que remetem a uma comunicação efêmera se manifestando sem registros oficiais, como, por exemplo fala predominante em congressos e conferências). Para a realização de um evento, em conformidade ao que afirma Martin (2017), tem-se o antes do evento (envolve idealização, definição de conteúdos, formação da equipe de divulgação), o durante o evento (realização na prática do que foi planejado na etapa anterior) e o depois do evento (apresentação dos resultados obtidos com a realização das etapas anteriores).

Por não se apresentar como atividades triviais, as diversas etapas para realização de um evento científico, algumas plataformas surgiram, com o advento da internet, visando um gerenciamento de atividades destes tipos de eventos. Sendo assim, no momento da condução deste estudo foi possível encontrar algumas opções de software/aplicações/aplicativos que facilitam a gestão de eventos. (Seção 3.2) Porém, em alguns casos as funcionalidades proporcionadas pelas aplicações analisadas são monetizadas (pagas), inviabilizando o uso em alguns casos como pode ser visto no Apêndice A.5. Assim como, diferentes contextos institucionais, principalmente inerentes ao IFPB-Campus Cajazeiras, unidades que são em suma as promotoras destes tipos de eventos, as ferramentas disponíveis no mercado cobrem a maioria das necessidades, mas é interessante ter um sistema que disponibiliza todas as suas funções gratuitamente porque custos com a divulgação e gerenciamento do evento pode inviabilizar a sua produção.

1.1 MOTIVAÇÃO

No início da ciência moderna, os cientistas se comunicavam por meio de cartas e participação em reuniões de entidades científicas para inteirar-se das novas práticas que estavam sendo desenvolvidas na sua área de pesquisa. Com o avanço nas diversas áreas do conhecimento, surgiram as sociedades temáticas, que a partir de então se reúnem em eventos que passam a ser fundamentais para a publicação do

conhecimento produzido por meio da ciência.

Para Carmo e Prado (2005) a ciência é um trabalho social, e sendo assim, precisa ser publicizada, discutida e pensada. A comunidade científica enxergada como produtora e propagadora dos mais recentes conhecimentos científicos, necessita estar frequentemente em busca de informações atualizadas, e para isso, constantemente é necessário usar dos mais diversos meios de comunicação científica para prover o máximo possível a identificação e disseminação do conhecimento já existente.

Meadows e Lemos (1999) agrupam os canais de comunicação científica em dois: formal e informal. Para eles, os canais formais de comunicação são aqueles nos quais se estabelece uma existência longa e dependem essencialmente da visão, onde os livros e periódicos se destacam. Já canais informais são aqueles onde a comunicação é imediata, pois é manifestada principalmente pela fala e muitas das vezes não é registrado de maneira oficial. Conferências e congressos são vistos como arquétipos do diálogo informal.

Targino (2000) fala que canais formais e informais são úteis em diferentes situações no que diz respeito à operacionalização das pesquisas. No entanto, ambos são imprescindíveis à comunicabilidade do conteúdo científico produzido, pois são utilizados em momentos diversos e obedecem a cronologias distintas.

Os eventos científicos estão no grupo da comunicação informal da ciência e possibilitam aos seus participantes o acesso à informação atualizada em seu campo de atuação profissional ou de estudo, como também maior facilidade nas relações e compartilhamento de conhecimento que acontece entre os pesquisadores. “Os participantes de conferências e congressos têm a possibilidade de planejar com antecedência contatos específicos. No entanto, a maioria dos contatos ocorrerão de forma imprevista, contudo serão muito úteis, no decorrer do evento” (MEADOWS; LEMOS, 1999).

Para Schmidt e Ohira (2002), os eventos científicos são uma forma muito eficiente para a comunicação oral do conhecimento, tendo em vista o ritmo crescente da evolução da ciência e, assim, um meio de divulgação e absorção de novos conhecimentos.

Para Marchiori et al. (2006) os eventos ou encontros científicos habitualmente, reúnem profissionais, estudantes, especialistas e demais segmentos interessados em compartilhar e obter conhecimentos sobre uma determinada área. Marchiori afirma que as principais funções de eventos como esses são: possibilitar a troca de experiências; atualização acerca dos progressos recentes de determinada área; sintetizar os avanços

mais recentes de um determinado campo da ciência; publicar novos conhecimentos; e, compartilhar metas e diretrizes para os futuros estudos numa determinada área do saber.

A partir da vivência como estudante do IFPB Cajazeiras e participante nos eventos promovidos pela instituição, vimos que a cada novo evento era necessário criar uma identidade visual para apresentação ao público o que muitas vezes provocava atrasos na divulgação, quando se era necessário que as inscrições para o evento fossem pagas as ferramentas que disponibilizam essas funcionalidades cobravam uma taxa sobre o valor das inscrições o que segundo entrevistas informais com a comissão de evento do campus não é interessante para instituição pagar essas taxas. Diante da importância dos eventos e do contexto observado no IFPB Campus Cajazeiras. Por meio deste estudo, foi construída a ferramenta/aplicação nomeada de *Evento-Service*, que tem como característica fazer uso das possibilidades para promoção da geração de conteúdos acessíveis através da internet para auxiliar no gerenciamento de eventos sem que a instituição tenha custos. É importante frisar que, para diminuir a complexidade do projeto, o controle financeiro é feito através de um gerenciador de pagamentos (mercadopago) ¹ externos à aplicação. Esse tipo de gerenciador de pagamento cobra taxa pelo serviço, de modo que caso a organização do evento queira utilizar a funcionalidade de inscrição paga terá que arcar com as taxas de uso (consultar valor de taxa no site da plataforma do MercadoPago).

Para que a construção da aplicação atenda aos objetivos almejados, a realização deste projeto foi dividida em um conjunto de atividades, seguindo abordagem preconizada por Pressman e Maxim (2016) na engenharia de software, especificamente no processo de entendimento dos requisitos, conforme descrito na (Seção 1.4)

1.2 OBJETIVO GERAL

Desenvolver um Sistema de Gerenciamento de Conteúdo (CMS) para o gerenciamento de eventos com finalidade acadêmica /científica.

1.3 OBJETIVOS ESPECÍFICOS

- Aplicação de análise de software sobre o problema em divulgação de eventos;
- Permitir que o usuário crie e disponibilize conteúdo para um evento científico sem a necessidade de conhecimentos técnicos em informática;

¹ <https://www.mercadopago.com.br>

- Disponibilizar suas funcionalidades por meio de uma API ("Application Programming Interface") possibilitando a integração com outras plataformas.
- Facilitar o acesso do público ao conteúdo acadêmico.

1.4 METODOLOGIA

Esta seção aborda a metodologia usada para o desenvolvimento deste trabalho. Foram realizadas as seguintes atividades:

Entendimento do domínio, etapa em que foi realizada a concepção das funcionalidades da proposta através de entrevistas para coletar informações com *stakeholders* levantando desejos e verificando, problemática existente e a viabilidade da proposta;

Entendimento dos requisitos, embasados no entendimento do domínio, foi levantado um conjunto de funcionalidades para a aplicação e conduzida a modelagem por meio do MCU (modelo de casos de uso) figura 6 e o MER (modelo Entidade Relacionamento) figura 9 para proporcionar um melhor entendimento aos *stakeholders* e assim, negociando as situações em que as possibilidades para a realização de uma funcionalidade solicitada se apresentava distante dos desejos, objetivos pretendidos e documentando todo o processo.

Comparativo entre aplicações: foi realizado um estudo de ferramentas/aplicativos semelhantes à aplicação proposta neste trabalho, de modo a identificar pontos que não atendiam as necessidades do contexto estudado, e assim pensar em propostas com melhorias,

Design inicial da proposta: nesta etapa foi definida a arquitetura (Seção 3.4), tecnologias e ferramentas a serem utilizadas, e a concepção dos protótipos seção 3.5 para as interfaces da aplicação com o usuário.

Validação da proposta: nesta etapa foi realizada a validação juntos aos *stakeholders* dos casos de uso (Seção: 3.3.2) para os requisitos e dos protótipos para interface da aplicação (Seção 3.5);

Testes: Embasados nos requisitos, foram criados testes de aceitação (Seção 3.3) aplicados ao final de cada iteração.

Elaboração do documento: foi escrito de forma incremental durante todo decorrer do TCC I e TCC II o documento monográfico fundamentado na aplicação

desenvolvida.

1.5 ORGANIZAÇÃO DO DOCUMENTO

Nesta seção apresentamos de forma resumida a organização deste trabalho.

O capítulo 1 apresenta a introdução, motivação do projeto, objetivo geral, específicos e metodologia, possibilitando ao leitor ter o entendimento geral da proposta do trabalho. O capítulo 2 apresenta a literatura que dá fundamentação teórica para a elaboração do projeto. O capítulo 3 apresenta o processo de desenvolvimento empregado na execução do trabalho, as etapas de análise de projeto, detalha a elaboração da proposta de solução, faz uma breve comparação entre ferramentas do gênero e ainda, detalha de forma simplificada a implementação, validação e implantação do sistema. O capítulo 4 traz as considerações finais do trabalho, como também fala sobre funcionalidades e conceitos que poderão ser trabalhos futuros. Por fim, são apresentados os Apêndices A, B e C, que são artefatos gerados no decorrer do desenvolvimento do trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo aborda conceitos teóricos necessários para alcançar os objetivos deste trabalho, visando solucionar as dificuldades apresentadas na motivação. Para tanto, foram escolhidas técnicas, métodos, metodologias e outras ferramentas que foram trabalhadas para a concepção do modelo de solução proposta.

Para fundamentar a preferência por determinada tecnologia, será realizada uma breve explanação sobre as mesmas, de forma a facilitar sua compreensão e funcionamento.

2.1 CMS

Content Management System (CMS) ou Sistema de Gerenciamento de Conteúdo (SGC) é um sistema voltado para criar, gerenciar e publicar conteúdos em sites na *web*. Foi projetado com a finalidade de melhorar a gestão e conteúdos dos sites das organizações.

Para Boiko (2001) a importância de um CMS está em sua capacidade de produção, gestão, compartilhamento e a recuperação de informações corporativas. Os CMS devem ser pensados tendo em mente que a administração do conteúdo precisa ser gerada sob a visão dos usuários e seus objetivos funcionais. Nesse contexto, vários *frameworks* foram desenvolvidos desde o final dos anos 90, mesmo sendo possível o desenvolvimento pela empresa de seu próprio sistema de administração de conteúdo.

No mercado existem diversas soluções de código aberto reconhecidas e muito utilizadas que atendem a variados tipos de demanda e também garante parâmetros de qualidade como disponibilidade, segurança, usabilidade e suporte.

Os CMS precisam ser capazes de controlar variados tipos de artefatos. Em outras palavras, um sistema de gerenciamento de conteúdo é uma aplicação que deverá controlar todos os documentos dentro de um site da *web*. Esses conteúdos podem ser fotos, músicas, textos, vídeos, ou documentos.

Os CMS gerenciando esses aspectos tem como objetivo uma organização das principais funções de manutenção de um portal, pois, um sistema de gerenciamento de conteúdo tem como objetivo principal organizar, favorecer, produzir, organizar e publicar informações em um determinado *website*. Segundo Bárcia (2008), em organizações onde houver a necessidade de criar e administrar um volume grande de dados com

diferentes pessoas na gestão, um CMS deve ser desenvolvido como solução, sobretudo quando a criação e gestão acontece em contextos distintos.

Ainda segundo Bárcia (2008), CMS são softwares que disponibilizam ferramentas para criar e publicar informação. A Figura 1. mostra como é organizada a estrutura de CMS, o primeiro módulo trata o tipo de informação que será armazenada, o segundo, funcionalidades disponibilizadas, depois, tema (cores, tamanho de fonte do texto, posicionamento do conteúdo...), todo esse conjunto resulta em uma página/site para a interação com público.

Figura 1 – Visão geral do funcionamento de um CMS



Fonte: mindtek.com.br (2021)

Um sistema do tipo CMS pode disponibilizar vários perfis de usuário para gerenciamento do *website*, com permissões distintas que podem ser funções simples, como inserção de notícias ou mais complexas, como: mudanças de *layouts* e administração de banco de dados. Em casos avançados, pode ter ainda áreas restritas para ser acessado apenas por perfil autorizado.

2.2 BANCO DE DADOS NOSQL

Um banco de dados é uma coleção de dados relacionados (ELMASRI et al., 2005). O conceito de banco de dados é um termo genérico. Os autores definem de uma maneira própria o conceito. De modo geral, é correto afirmar que no computador um banco de dados é a parte onde são guardadas informações que se relacionam. Somado a isso, os bancos de dados podem ter características, tamanhos e complexidades diferentes, ficando a critério do desenvolvedor determinar qual banco fornece os melhores recursos necessários para a informação que será armazenada.

Segundo Costa (2019), planejar e desenvolver a base de dados é uma das etapas mais importantes no desenvolvimento de um sistema de informação e a cada

novo banco de dados projetado as especificações são diferentes. Essas especificações mudam conforme a complexidade dos dados que serão armazenados. Atualmente, com a evolução da TI (Tecnologia da Informação), até tarefas simples são automatizadas e a quantidade de sistemas com um volume grande de dados não para de aumentar (OLIVEIRA et al., 2018). Segundo Souza (2016) nos últimos anos, novos modelos de banco de dados, chamados NoSQL (Not Only SQL) estão sendo considerados alternativas para a gestão desse volumes de informações, pois gerenciam e armazenam os dados de forma eficiente, possuem alta escalabilidade, disponibilidade e desempenho satisfatório.

Segundo, Carniel et al. (2012), os bancos de dados NoSQL são pensados e desenvolvidos para atender a tarefas específicas das aplicações. A maior parte dos trabalhos que abordam o tema NoSQL, citam 4 tipos de implementações de base de dados com os principais do paradigma NoSQL mas, que armazenam tipos de dados distintos. Santos e SILVA (2013), Sadalage e Fowler (2013), Carniel et al. (2012) e Corbellini et al. (2017), apresentam que os 4 modelos de dados dos bancos NoSQL são : Chave-Valor, Orientado a Colunas, Orientado a Documentos, e Baseado em Grafos. No banco de dados chave-valor, os dados são armazenados como pares chave-valor, e são endereçados por uma chave. Os bancos de dados orientado a colunas definem a estrutura de valores como um conjunto pré-definido de colunas. No banco de dados orientado a documentos, os documentos são conjuntos de atributos e valores, e cada documento contém um identificador. Um banco de dados baseado em grafos utiliza o modelo de grafos para representar o esquema de dados

Existe vários bancos que seguem o paradigma NoSQL cada um com seu modelo específico de dados para atender necessidade distintas, oferecendo assim novas perspectivas, tais como: alta escalabilidade, flexibilidade no modelo de dados, possibilitando estruturas de dados adaptáveis, dinâmicas e com agilidade na implementação, além de criar linguagens de consultas específicas para cada modelo de dados e várias interfaces para o acesso aos dados.

Sadalage e Fowler (2013) fala que os pesquisadores consideram os bancos de dados NoSQL interessantes pela produtividade no desenvolvimento de aplicativos, devido na maioria dos casos se gastar muito tempo copiando e adaptando as informações a base de dados relacional e a possibilidade de se adequar a grandes demandas de dados.

Como a capacidade de crescimento para um banco NoSQL é praticamente ilimitada, é possível replicar facilmente esse tipo de bases de dados. Bancos dessa categoria tornam-se essenciais para grandes volumes de informação exigido pelas

aplicações atuais. Segundo, Rys (2011), o crescimento no uso e evolução do paradigma NoSQL é motivado pela necessidade cada vez maior de se alcançar alta escalabilidade para tratar a grande carga de dados gerados pelos mais variados sistemas e aplicações da *WEB*.

O paradigma de banco de dados NoSQL é um conceito com uma série de características que o torna diferente do Relacional. Algumas dessas características são: esquema flexível ou livre de esquema, escalabilidade e consistência eventual, que podem ser encontradas em diversos modelos desse paradigma, a seguir, serão explicadas, de forma sucinta, estas principais características do conceito NoSQL.

2.2.1 Escalabilidade

Segundo Lóscio et al. (2011), a escalabilidade acontece à medida que o volume de dados da aplicação aumenta muito, levando assim a necessidade de se criar mecanismos que aumentem o desempenho da aplicação para atender a demanda. Como bancos relacionais utilizam-se de características transacionais: *Atomicity, Consistency, Isolation, Durability* (ACID), para através dessa propriedade garantir a integridade dos dados, características como escalabilidade e alta disponibilidade tornam-se custosas. Essa dificuldade abre espaço para bancos NOSQL em ambientes onde requisitos como alta disponibilidade e o uso de sistemas distribuídos são necessários para atender os clientes, favorecer a criação de novas formas, e também prover a integridade dos dados.

Santos e SILVA (2013) afirmam que para resolver, ou ao menos minimizar esse problema, aplica-se ao bancos de dados NoSQL o teorema CAP (Consistency, Availability and Partition Tolerance). A utilização desse teorema exige que a aplicação garanta dois dos seus três aspectos.

1. Consistency (Forte consistência) – Após uma transação a integridade dos dados deve ser mantida.
2. Availability (Alta disponibilidade) – O sistema deve prover o acesso do cliente as suas funcionalidades.
3. Partition Tolerance (Tolerância ao particionamento) – O sistema deve manter suas propriedades mesmo que instalado em servidores diferentes.

Há dois tipos de escalabilidade: horizontal e vertical. Segundo Lóscio et al. (2011), a escalabilidade horizontal acontece com o aumento do número de máquinas

físicas para processamento e armazenamento de dados. Geralmente, são adicionados outros servidores, o que possibilita a criação de um cluster. Quando se utiliza bancos de dados que implementados sob o paradigma relacional, o uso de cluster torna-se impraticável quando se fala de escalabilidade horizontal devido à grande concorrência dos processos.

Segundo, BONFIM e LIANG (2014), a escalabilidade vertical consiste em aumentar o poder de processamento e armazenamento das máquinas, ou seja, quando há a necessidade de mais memória, ou a troca do processador por um mais potente, entre outros fatores. Sadalage e Fowler (2013), dizem que escalabilidade vertical acontece quando há uma melhoria de hardware de um único servidor. Essa estratégia de escalabilidade é vista como inviável, devido ao alto custo a cada troca da máquina ou aumento do processamento o que impossibilita este método de escalabilidade.

2.2.2 Esquema flexível ou ausência de esquema

Lóscio et al. (2011) afirmam que características como esquema flexível ou ausência de esquema é bem evidente nos bancos de dados implementados no paradigma NoSQL. Para Saccol (2001), devido o enorme crescimento de dados semi estruturados produzido pelos mais variados tipos de plataformas existentes, surgiu uma categoria de Banco de dados nova, criada com o objetivo de dar suporte a consultas em ambientes onde o modelo não tem esquema definido. Esta propriedade é fundamental quando se deseja garantir uma alta escalabilidade, vale destacar a não garantia da integridade dos dados, o que é um ponto negativo.

2.2.3 Consistência eventual

Tomando por referência o teorema CAP, pode se afirmar que não é possível se manter a consistência sempre devido à distribuição dos dados em diferentes pontos que podem estar separados geograficamente.

Dada a impossibilidade de ser consistente o tempo todo, os bancos NoSQL trabalham com o conceito de consistência eventual, que refere-se ao fato do banco não ser consistente o tempo todo, como ocorre nos bancos relacionais.

Isso não significa que o banco abandonará a consistência, mas sim, que os bancos trabalham com o conceito de janela de inconsistência, tornando-se consistente após certo tempo. A consistência eventual é necessária para garantir outras propriedades, como disponibilidade, que em alguns domínios é essencial.

2.2.4 Modelos de Dados NoSQL

Dentre os vários tipos de banco de dados NoSQL, segundo, Lóscio et al. (2011), não existe um modelo melhor, e sim um modelo que se adequa melhor para cada problema, pois cada tipo de banco de dados NoSQL foi pensado para uso em um contexto específico. Ainda segundo Lóscio et al. (2011), os principais tipos de modelos de dados NoSQL podem ser classificados em: orientado a documentos, chave-valor, orientado a colunas e orientado a grafos.

Banco de dados chave valor são compostos de uma tabela hash, estrutura de dados que associa uma chave única a um valor. A chave será usada para pesquisa do valor posteriormente. Como é mostrado na figura 2, cada value(valor) é referenciado pelo uma chave única key(chave). Essa estrutura é bastante conhecida pois, oferece uma grande eficiência por conta da sua facilidade de implementação e velocidade na busca de dados, seu acesso aos dados é ágil pois, a busca é feita através das chaves, aspecto que dá maior rapidez e propicia aumentar a capacidade o que o torna uma boa alternativa para sistemas onde a escalabilidade é essencial. Um par chave-valor é uma estrutura onde a informação pode ser facilmente acessadaTiwari (2011).

Figura 2 – Visão geral banco chave valor

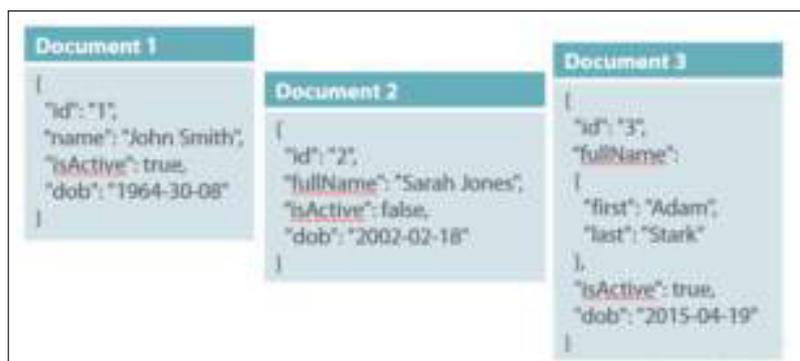
Key	Value
K1	AAA,BBB,CCC
K2	AAA,BBB
K3	AAA,DDD
K4	AAA,2,01/01/2015
K5	3,ZZZ,5623

Fonte: db4beginners (2021a)

Segundo, Cattell (2011), os bancos de dados orientado a documentos são bem mais poderosos que o BDs chave valor, pois suportam dados com maior grau de complexidade. No modelo orientado a documentos, o armazenamento de documentos é feito geralmente no formato *JavaScript Object Notation* (JSON). Sadalage e Fowler (2013), definem bancos de documento como sendo banco de dados que armazena e

recupera documentos que na grande maioria estão nos formatos *XML*, *JSON*, *BSON* dentre outros. Esses tipos de documentos são estruturas de dados no modelo de árvores hierárquicas e que se auto descrevem, formado por coleções, mapas e valores escalares, como podemos ver na figura 3 o documento é composto por uma chave única e vários atributos com seus respectivos valores.

Figura 3 – Estrutura banco orientado a documento

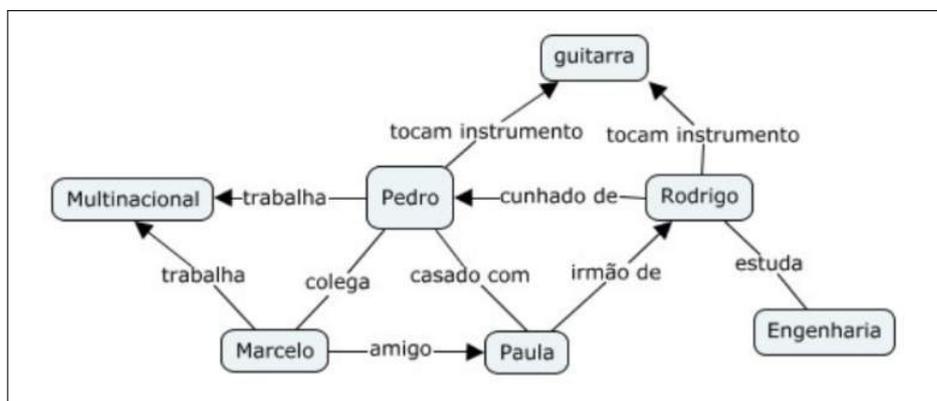


Fonte: db4beginners (2021c)

Nayak et al. (2013), afirmam que bancos de dados orientados a documentos têm ótima escalabilidade horizontal e desempenho. Os dados armazenados em um base de dados desse tipo são semelhantes a registros em bancos relacionais. Contudo, são mais flexíveis, pois são estruturas de dados com menos esquemas e regras para o armazenamento dos dados.

Segundo Higa (2016), bancos de dados orientados a grafos são uma das várias categorias de bancos de dados NoSQL. O principal aspecto que o diferencia do modelo relacional é a representação evidente das relações entre os dados, por meio de um modelo com vértices nomeados de nós e linhas, chamadas de relação como pode ser visto na Figura 4.

Figura 4 – Visão geral banco orientado a grafo



Fonte: db4beginners (2021b)

Segundo, Santos e SILVA (2013), o desenvolvimento de banco de dados orientado a grafo está fortemente ligado à criação do paradigma NOSQL. Os bancos em grafo foram pensados para casos onde seja preciso armazenar um volume grande de dados complexos, além de contar com uma estrutura com nível de acoplamento baixo e ser muito suscetível a mudanças nas operações de criação, remoção, atualização e leitura. O paradigma orientado a grafos foi pensado como um solução do problema de performance que acontece no modelo relacional quando é necessário consulta de dados com muitos relacionamentos entre tabelas. O modelo destaca-se por conta da velocidade da consulta, que se mantém independentemente do número de nós do grafo.

Segundo Pereira et al. (2013), os bancos de dados em famílias de colunas agrupam os dados em famílias de colunas. As linhas são organizadas de acordo com o seu contexto, característica que torna esse modelo de banco de dados mais complexo que o tipo chave-valor. Diferentemente dos tipos relacionais que têm seu conceito de orientação a registros, ele usa orientação a atributos ou colunas, operações de escrita e leitura são atômicas. Na execução dessas operações são levado em consideração todos os valores da linha. Pokorny (2013), fala que esse tipo de banco de dados é formado por uma composição entre linha, coluna e as linhas e colunas são identificadas por chaves.

Mesmo não estando ligado ao conceito dos bancos do modelo relacional, este modelo de banco armazena os dados em famílias de colunas. Nesse conceito, linhas e colunas são organizadas e ordenadas de acordo com seu contexto, daí vem a

semelhança com o paradigma do relacional. Por suas características, esse modelo torna-se bem mais propício e apresenta melhor desempenho para sistemas com grande processamento de dados distribuídos.

Segundo Kyle (2011) O mongoDB um banco de dados orientado a documentos, desenvolvido em linguagem C++. suas principais características são escalabilidade horizontal e a alta disponibilidade A sua flexibilidade de esquema é um ponto que favorece a utilização Por parte dos usuários Souza (2016).

Com base nos autores citados percebe-se que não há um banco de dados melhor ou pior, e sim o mais adequado para cada aplicação. No presente trabalho foi utilizado o banco MongoDB para se obter flexibilidade no esquema de dados e a possibilidade de escalonamento horizontal com mais facilidade. Isso porque dependendo do tipo de evento que está sendo realizado as informações podem ser diferentes. Nessas situações, torna-se interessante ter um banco com esquema flexível de dados, além da escalabilidade horizontal diminuir os custos de manutenção do sistema caso ele passe a ser utilizado por muitos usuários.

2.3 REST

Segundo, Fielding e Taylor (2000), a *Web* é um projeto bem sucedido pelo fato de sua arquitetura de *software* conseguir entregar a infraestrutura base para um sistema hipermídia distribuído em escala de Internet. A *Web* evoluiu de maneira interativa e a definição de seus padrões arquiteturais passou por inúmeras intervenções.

Ainda de acordo com Fielding e Taylor (2000), pode-se entender *REST (Representational State Transfe)* por um estilo arquitetônico utilizado para orientar a criação e desenvolvimento de uma aplicação de hipermídia disponibilizada em larga escala na *Web*.

Segundo, Garland (2010), pode-se definir um estilo arquitetônico como sendo uma especialização de elementos com tipos e relações, acompanhada de uma série de restrições sobre como eles poderão ser usados pelo público alvo. Esta definição também pode ser encontrada na página oficial do Software *Engineering Institute (SEI)*. Quanto ao estilo arquitetural do REST, Pautasso, Zimmermann e Leymann Pautasso et al. (2008), define que pode se ver ele como sendo uma entidade abstrata, onde os conceitos têm sido utilizados como forma de explicar a grande escalabilidade que o protocolo HTTP é capaz de entregar.

Comumente na literatura encontra-se o termo *Restful*. Este termo é geralmente

usado para referenciar sistemas que implementam os conceitos REST. Segundo, (RICHARDSON; RUBY, 2008), *restful* é parecido com o que se designa “orientação a objetos”. Um sistema pode ser implementado com linguagem OO, no entanto, isso não significa que sua arquitetura é orientada a objetos. Ao termo Restful se aplica a mesma lógica.

O REST pode ser implementado utilizando outro protocolo, não apenas com o HTTP. Nesta seção, serão apresentadas as principais características do REST com foco na interpretação atual, usada para definir “*Restful Web services*”. Outras informações sobre o REST podem ser encontradas também em (FIELDING; TAYLOR, 2000).

O REST define quatro princípios (restrições) de interface (FIELDING; TAYLOR, 2000). São eles: identificação dos recursos, manipulação de recursos por meio de representações de mensagens auto descritivas e Conceito de *Hypertext As The Engine Of Application State (HATEOAS)*. Estas restrições são a base do REST.

O REST é implementado usando o protocolo HTTP e as URI (Identificador de Recurso Uniforme) identificam os recursos. Um *Restful Web Service* apresenta um conjunto de recursos e eles são usados pelos clientes para interagir. Em “*Cool URIs don’t change*”, criado em 1998 por *Tim Berners-Lee*, mostra sua preocupação no tocante ao uso dos URIs e sua manutenção para que eles continuem fiéis à sua criação, não levando em consideração o tempo que devam existir. Também o mesmo documento pode ser prova desta preocupação e compromisso, permanece até nos dias atuais com a mesma URI.

Basicamente, a manipulação dos recursos é feita utilizando um conjunto fixo de quatro operações definidas pelos métodos do HTTP. (POST, GET, PUT e DELETE) formam estas operações. Este mapeamento consegue cumprir a meta “restrição” de interface uniforme do REST.

2.4 PROCESSO DE DESENVOLVIMENTO YP

Segundo, Humphrey (1989), “Um processo de desenvolvimento de software é um conjunto de ferramentas, métodos e práticas usados para construir um produto de software”. Em Dubinsky e Hazzan (2003), é dito que “Processos de engenharia de software são difíceis de serem implantados, até mesmo na academia”.

É na academia onde encontra-se o ambiente propício para teoria, e sim aliar teoria à prática a fim de propiciar ao estudante uma visão genérica do trabalho com Engenharia de Software. Para esse propósito, faz-se necessário que a aplicação dos

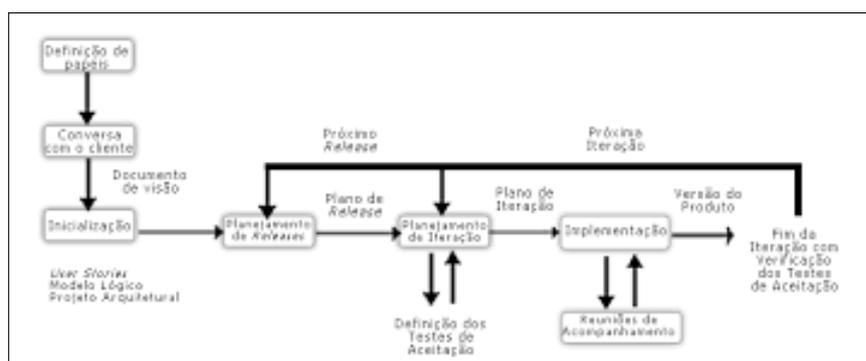
conhecimentos obtidos ao longo do curso seja eficaz na elaboração e construção de um produto. O sucesso é mais facilmente alcançado quando se pode fazer uso de um processo de desenvolvimento de software adequado. Garcia et al. (2004)

2.4.1 YP

Segundo Garcia et al. (2004) o easYProcess (YP), é um processo de software simplificado que se apoia em práticas do XP. O objetivo do YP é auxiliar tanto a gestão do desenvolvimento de aplicações em disciplinas de engenharia de software como também na aprendizagem dos conceitos desta disciplina na graduação, e ainda pode ser utilizado em projetos pequenos e médios.

O fluxo básico do YP está ilustrado na figura 5.

Figura 5 – Fluxo do easyProcess



Fonte: GARCIA et al., 2004

Quando utiliza-se o YP o primeiro passo é a definição de papéis, o YP recomenda os seguintes papéis: cliente, usuário, testador, desenvolvedor e gerente, podendo uma mesma pessoa desempenhar mais de um papel dentro do processo,

Na fase seguinte acontece uma conversa com o cliente. É nessa etapa onde são levantadas informações sobre o escopo do projeto.

Na etapa de Inicialização o cliente define as *User Stories* e é criado o projeto arquitetural e o modelo lógico de dados, este último apenas se necessário. O cliente deve definir prioridades para as *User Stories* e a equipe define um tempo estimado

para a implementação de cada uma delas. Com base nessa estimativa, é possível então avaliar a viabilidade de implementação do projeto no escopo e tempo previstos Garcia et al. (2004).

Depois de Inicialização dá-se início ao Planejamento, fase que se divide em dois planos, o de release e o de interação. Os dois possuem tempo fixo, podendo haver variação de escopo permitida. No ambiente acadêmico, são sugeridos três releases, com duas iterações de duas semanas, por semestre letivo Garcia et al. (2004).

A execução do processo deve ser supervisionada pelo gerente por meio da Reunião de Acompanhamento semanal, onde o objetivo é recolher e analisar métricas.

3 FERRAMENTA

Por meio deste estudo, foi construída a ferramenta/aplicação nomeada de “EventoService”, que tem como característica fazer uso das possibilidades para promoção da geração de conteúdos acessíveis através da WEB para auxiliar no gerenciamento de eventos.

O YP foi adotado como processo de desenvolvimento de software desse projeto, por que como abordado Seção 2.4 ele é simples e adaptativo .

3.1 ETAPAS DO DO PROCESSO

3.1.1 Conversa com cliente

Nessa etapa foi criado o Documento de Visão. O objetivo é que ele sirva de ferramenta auxiliar para evitar problemas custosos. Ele mostra os problemas que precisam ser solucionados, as necessidades dos envolvidos, a abrangência do projeto e as aplicabilidades esperadas para o sistema. Segundo Filho (2014), o objetivo desse documento é evidenciar as necessidades e funções comuns do sistema, definindo os requisitos de alto nível em termos de necessidades dos usuários finais. O documento de visão produzido durante esta etapa pode ser visto no Apêndice A.

3.1.2 Inicialização

Como indicado pelo YP, nesta etapa foram definidas as Users Stories e suas estimativas de prazo para implementação. As Users Stories levantadas podem ser vistas no Quadro 4, o quadro traz o identificador da Users Stories e uma descrição simplificada. Na Seção 3.3 uma dessas pode ser vista em detalhes, as demais estão disponíveis no Apêndice - B. *User Stories* e Testes de Aceitação.

3.1.3 Definição de Papéis

Nessa etapa foi definido a função de cada um dos envolvidos no sistema como pode ser visto no Quadro 1:

Quadro 1 – Definição de Papéis

Papel	Stakeholder
Cliente	Francisco Paulo de Freitas Neto
Usuário	Francisco Paulo de Freitas Neto
Gerente	José Ferreira Vieira
Desenvolvedor	José Ferreira Vieira
Testador	José Ferreira Vieira

Fonte: Elaborado pelo autor

3.1.4 Planejamento de *Releases*

Durante esta etapa foram definidas as *Releases*, juntamente com a data estimada para o término de seu desenvolvimento. Esses dados podem ser vistos no Quadro 2 Plano de Iteração.

Quadro 2 – Plano de Iteração

RELEASE	User Stories	Data de entrega
01	US 01 e US 02	10/05/2018
02	US 03 e US 04	10/06/2018
03	US 05 e US 06	10/07/2018
04	US 07 e US 08	30/04/2021
05	US 01 e US 02	30/04/2021

Fonte: Elaborado pelo autor

3.2 FERRAMENTAS RELACIONADAS

Ao analisar algumas ferramentas/aplicações/aplicativos em busca de compreender as funcionalidades disponibilizadas pelas mesmas e as necessidades dos stakeholders do contexto estudado, constata-se que algumas das necessidades dos atores não eram atendidas pela versão gratuita dessas plataformas, exemplo: credenciamento digital e inscrições pagas sem cobrança de taxa.

No Quadro 3 pode-se encontrar o comparativo das principais aplicações analisadas que são possíveis de serem utilizadas na criação de eventos acadêmico-científicos, e as respectivas funcionalidades comparando-as com as funcionalidades desejadas pelos usuários do contexto analisado ofertado pela aplicação proposta. Vale ressaltar que a comparação foi feita apenas com as funcionalidades oferecidas no plano gra-

tuito das ferramentas concorrentes, As aplicações analisadas foram: Even3¹, Doity², congressy.³

Quadro 3 – Funcionalidades das ferramentas

Funcionalidade	Even3	Doity	congressy	EventoService
Site do evento	Sim	Sim	Sim	sim
inscrições no evento	Sim	Sim	Sim	sim
Inscrições pagas sem taxa	Não	Não	Não	sim
Realizar credenciamento	Sim	Sim	Não	sim
Site do evento	Sim	Sim	Sim	sim
Integração com outras plataformas	Sim	Sim	Sim	sim

Fonte: Elaborado pelo autor ⁴

Em sua versão paga, essas ferramentas disponibilizam uma série de funcionalidades adicionais. Até a data 14 junho de 2021 o valor cobrado é de 10% das inscrições (se as inscrições foram gratuitas não há cobrança de taxa).

As funcionalidades do EventoService são totalmente gratuitas, cabendo a organização do evento apenas a taxa de gerenciamento de pagamentos do mercado pago que até a data da escrita desse trabalho 14 de junho de 2021 varia entre 0,99 e 4,99 dependendo do meio de pagamento escolhido (cartão, boleto, pix).

3.3 ANÁLISE

A Engenharia de requisitos (ER) é vista como a etapa mais crítica do desenvolvimento de software, Pois é nesse momento do processo que se faz a análise, documentação e verificação das funcionalidades e restrições de software Sommerville (2011). Entretanto, garantir que todo o processo seja realizado com sucesso, não é uma tarefa simples de ser feita, em especial quando se aplica a ER em sistemas complexos e com regras de negócio que mudam constantemente (BARJIS, 2008). Os artefatos criados na fase de análise do sistema serão apresentados nas próximas seções deste capítulo,

¹ <https://www.even3.com.br>

² <https://doity.com.br>

³ <https://www.congressy.com>

3.3.1 User Storie

As *User Stories* levantadas podem ser vistas no Quadro 4. Elas foram divididas em menores tarefas, deixando a cargo do cliente definir os testes de aceitação referentes à cada *User Story*.

Quadro 4 – User Stories levantadas

Identificação	Definição
US01	O sistema deve disponibilizar uma página de cadastro e autenticação para os usuários.
US02	O sistema deve proporcionar o gerenciamento de perfil para os usuários.
US03	O sistema deve disponibilizar uma página para criar evento.
US04	O sistema deve disponibilizar uma página para gerenciamento dos eventos.
US05	O sistema deve disponibilizar uma página de acompanhamento dos eventos.
US06	O sistema deve disponibilizar uma página de inscrição no evento.
US07	O sistema deve realizar gerenciamento do credenciamento dos participantes.
US08	O sistema deve gerar a lista de presença para o evento e atividades.
US09	O sistema deve gerar a relatório ou fim do evento.
US10	O sistema deve permitir a definição da forma de inscrição (pago ou gratuita).

Fonte: Elaborado pelo autor

Para facilitar a organização, foi feito uso de um quadro de Alocação de Tarefas (AT). Nesse quadro são especificadas informações sobre o desenvolvimento das *User Stories* como: a tarefas, responsáveis, estimativas de tempo, tempo real consumido e status da tarefa. A especificação das tarefas executadas durante o desenvolvimento das *User Stories* US03, US04, US09 e US10. pode ser visto mo Quadro 5, as tarefas das outras *User Stories* são detalhadas no Apêndice B.

Quadro 5 – Alocação de Tarefas - US03 US04 US09 US10

Tarefa	Descrição	Responsável	Estimativa de tempo (Horas)	Tempo gasto (horas)
T1	implementar script para salvar o evento.	José Ferreira	4	5
T2	Implementa script de validação do evento.	José Ferreira	3	1
T3	Implementar script para editar o evento.	José Ferreira	1	1
T4	Implementar script para excluir o evento.	José Ferreira	3	3
T5	Criar interface com usuário para as funcionalidades.	José Ferreira	9	11
T6	Implementar script para convidar colaborador.	José Ferreira	2	1
T7	Implementar script para convidar participante.	José Ferreira	3	3
T8	Implementar script para criação de atividade	José Ferreira	5	6
T9	criar script de configuração das inscrições do evento	José Ferreira	6	5
T10	criar script de configuração de inscrições pagas	José Ferreira	19	15
T10	Implementar script para geração de relatórios	José Ferreira	12	10
T10	Implementar script para geração de lista de presença	José Ferreira	12	10

Fonte: Elaborado pelo autor

O Quadro 6 mostra os testes de aceitação produzidos durante o andamento das US03, US04, US09 e US10, os demais testes são detalhadas no no Apêndice B.

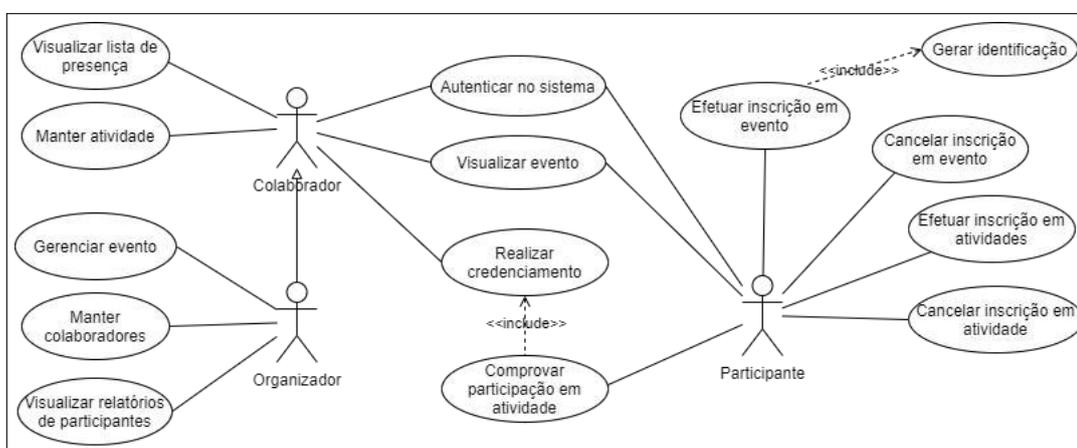
Quadro 6 – Teste de aceitação - US03 US04 US09 US10

Tarefa	Teste de aceitação	Status
TA1	Dado que estou na página de cadastro de evento, quando eu preencher o formulário com os dados obrigatórios de forma correta , o evento deve ser salvo e informado uma mensagem de sucesso além de me direcionar para página de configuração do evento.	Finalizado.
TA2	Dado que estou na página de cadastro de evento, quando eu preencher o formulário com os dados obrigatórios de forma incorreta , o evento não deve ser salvo e uma mensagem de erro deve ser exibida.	Finalizado.
TA3	Dado que estou acessando a página de edição do evento, quando eu alterar alguma informação do formulário com os dados válidos, o evento deve ser atualizado e uma mensagem de sucesso deve ser exibida.	Finalizado.
TA4	Dado que estou na página de edição do evento, quando selecionar a opção excluir o evento, um diálogo de confirmação deve ser exibido. Caso confirme, o evento deve ser excluído..	Finalizado..
TA5	Dado que estou na página de edição do evento, quando selecionar a opção excluir o evento, um diálogo de confirmação deve ser exibido. Caso não confirme o evento não deve ser excluído.	Finalizado.
TA6	O usuário devidamente logado quando clica em um evento que ele é um organizador, deve ser redirecionado para o painel administrativo onde pode configurar diversas informações sobre o evento.	Finalizado.
TA7	No painel colaboradores: o organizador pode convidar colaboradores inserido nome e email da pessoa convidada. Quando o organizador clicar em enviar o convite deve ser enviado por e-mail para pessoa indicada	Finalizado.
TA8	No painel participantes: o organizador pode convidar participantes inserido nome e email da pessoa convidada. Quando o organizador clicar em enviar o convite deve ser enviado por e-mail para pessoa indicada	Finalizado.
TA9	No painel atividade o organizador pode criar novas atividades para o evento. inserir os dados corretamente e clicar em salvar, a atividade será vinculadas ao evento, se os dados forem digitados incorretamente deve ser exibida uma mensagem de erro. O organizador e colaboradores também poderão ver a lista de presença dos	Finalizado.
TA10	No painel inscrições: o organizador Pode configurar Quais são os dados que serão solicitados para o participante no ato da inscrição	Finalizado.
TA11	No painel pagamentos: o organizador vai poder configurar se quer ou não cobrar ingressos e quais serão os valores e prazos definidos	Finalizado.
TA12	Ao final do evento o sistema disponibilizará aos organizadores um relatório sobre o evento	Finalizado.

3.3.2 Diagrama de casos de uso

Os casos de uso descrevem as regras de negócios e, por isso, eles são excelentes para entender o que o sistema pode ou não fazer, de acordo com a perspectiva do usuário, (SANTOS, 2015) A Figura 6 apresenta os casos de uso para os atores envolvidos na utilização da aplicação (Organizador, Colaborador e Participante). Foram levantados 14 casos de usos, eles fazem uma descrição breve das funcionalidades que foram disponibilizadas para os atores do sistema, o detalhamento desses casos de uso pode ser encontrado no Quadro 4.

Figura 6 – Diagrama de caso de uso para os atores da aplicação



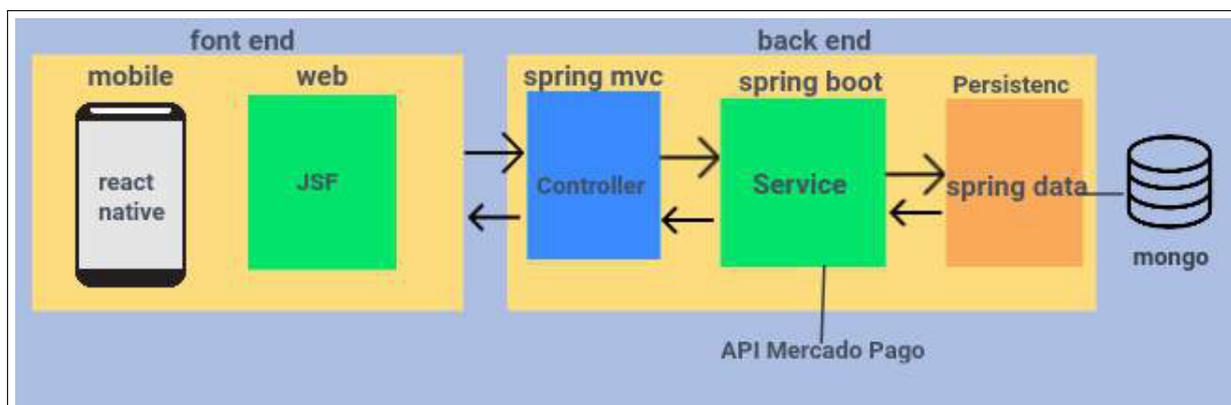
Fonte: Elaborada pelo autor

3.4 ARQUITETURA

A aplicação é organizada em camadas horizontais, sendo cada camada responsável por desempenhar um papel específico no sistema. São elas: visão, serviços e persistência de dados. Nas seções 3.4.1, 3.4.2 e 3.4.3 são abordados aspectos de cada uma delas. Segundo Sommerville (2007), o modelo em camadas organiza o sistema em uma estrutura que cada componente desempenha sua função independente dos demais, fornecendo um conjunto de serviços, essa abordagem apoia o desenvolvimento incremental de sistemas.

Conforme uma camada é desenvolvida, serviços disponibilizados pela mesma podem ser liberados para os usuários. Na Figura 7, pode ser vista a representação do modelo arquitetural usado para a implementação do EventService.

Figura 7 – Modelo arquitetural



Fonte: Elaborada pelo autor

3.4.1 Fron end

Essa camada permite ao usuário interagir diretamente com o sistema. É composta por páginas JSF (*JavaServer Faces*) e HTML (*HyperText Markup Language*), que foram estilizadas com o uso de CSS3. Para agilizar o processo de desenvolvimento dessas páginas, foram usados os *frameworks Bootstrap* na versão 4 e *Primefaces*. Com isso, a velocidade na criação das páginas aumentou devido à existência de dezenas de componentes já estilizados e estruturalmente organizados, que puderam ser reaproveitados. Alguns efeitos visuais da página foram implementados com o uso de JavaScript.

Todo o conteúdo dinâmico dessas páginas foi implementado usando JSF em sua versão 2.2 e para o conteúdo estático foi usado HTML. Esta camada pode interagir diretamente (e apenas) com sua camada adjacente, a camada de back end que será discutida na seção 3.4.2. O estabelecimento desta comunicação é feita utilizando o *Spring rest MVC*, através de controladores. Toda a configuração necessária para a inicialização desta camada é feita através do módulo *Spring Boot*.

Segundo Lemon e Rossi (1995) um Framework pode ser visto como um projeto genérico em um domínio que pode ser adaptado a aplicações específicas, servindo como um molde para a construção de aplicações.

O Spring Boot, o Spring rest MVC, assim como o Spring Data fazem parte do

Spring Framework⁵. Trata-se de um poderoso framework de código aberto desenvolvido em Java, para suporte a injeção de dependências, gerenciamento de transações, aplicações web, manipulação de dados, troca de mensagens e testes.

O Spring Boot é responsável por toda a configuração da aplicação. Ele foi escolhido por ser ideal para construir aplicações Spring prontos para a produção, tornando a tarefa de configuração a mais simples possível.

O Spring MVC é o framework MVC nativo do Spring framework que permite para desenvolver aplicações web em Java, seguindo o padrão MVC de arquitetura. A junção do Spring Boot e Spring MVC em um projeto torna o desenvolvimento de uma aplicação web em Java bem mais fácil, pois não há mais a necessidade de o desenvolvedor ficar se preocupando com arquivos de configuração seja Java ou XML.

O sistema também conta com uma versão mobile criada com React native⁶. Desenvolvido pelo Facebook em 2015, o React Native é um Framework para criação de aplicativos móveis multiplataforma. O React native foi criado com base no React, framework JS para desenvolvimento web. O React Native torna possível o desenvolvimento de aplicações móveis multiplataforma (Android e iOS) usando apenas Javascript, diferente de outros frameworks com esta mesma finalidade (Córdova, por exemplo). O código desenvolvido com o React Native é transpilado para linguagem nativa do sistema (android ou IOS), o que deixa o app mais performático.

3.4.2 Back end

O back end contém todas as funcionalidades responsáveis por gerenciar toda a parte do domínio de negócio da aplicação. Assim, é primordial que ela seja isolada das demais camadas, pois ela pode ser reutilizável em aplicações que usam a mesma linguagem, mas que utilizam tecnologias diferentes. Com isso, ela desempenha as funcionalidades núcleo da aplicação, como cadastro de usuário e criação de eventos. Essa camada contém as entidades de domínio de negócio utilizadas na aplicação. As entidades são manipuladas de acordo com a lógica de negócio e são repassadas para as camadas adjacentes, também contém todos os serviços, onde é responsável por executar a lógica de negócio da aplicação e fazer a comunicação com serviços externos como a API do MercadoPago⁷ uma solução de comércio eletrônico para transações comerciais através de pagamentos online.

⁵ <https://spring.io/projects/spring-framework>

⁶ <https://reactnative.dev/>

⁷ <https://www.mercadopago.com.br/developers/pt/guides>

3.4.3 Camada de persistência

A camada de persistência Está incorporada ao back and e ela a responsável pela gravação dos dados de forma permanente na base de dados. A persistência das informações é realizada através do Framework Spring Data mongo ⁸. Como forma de persistência será utilizado o banco de dados nosql mongodb. A comunicação entre as camadas será realizada através de *API REST* e os dados serão trafegados entre as camadas no formato JSON.

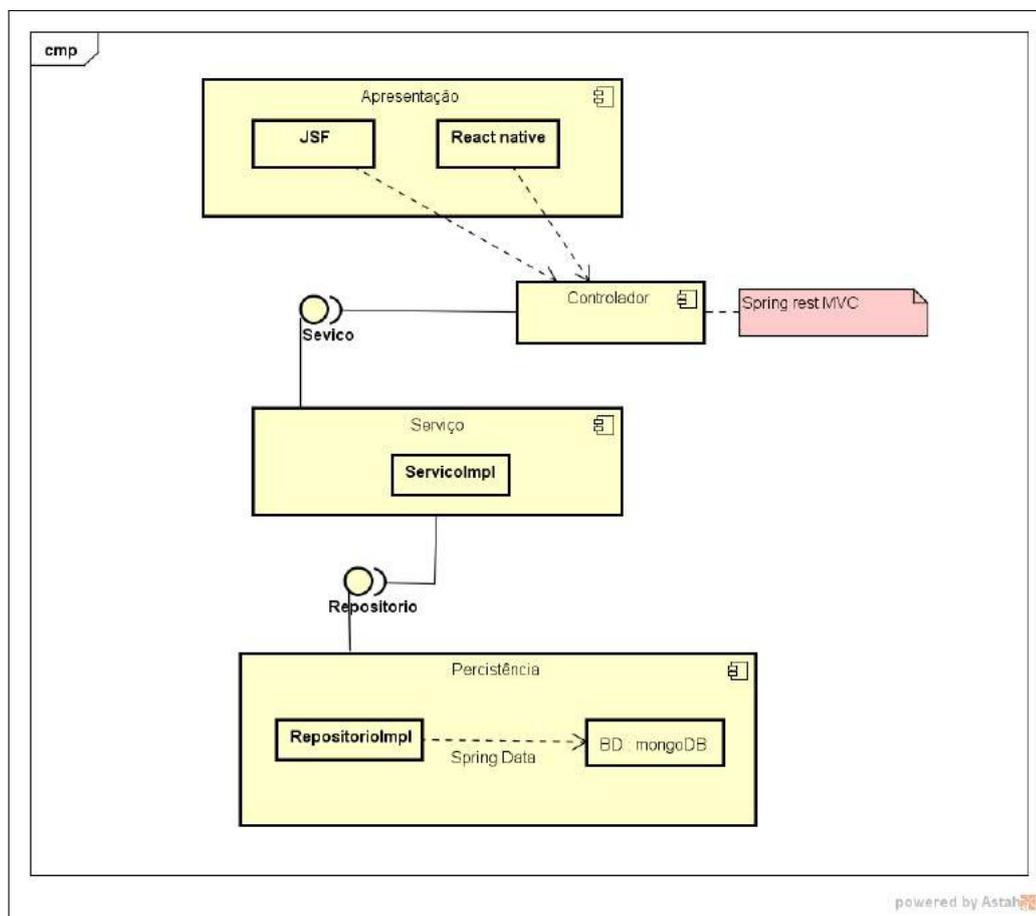
3.4.4 Diagrama de componente

A UML (Unified Modeling Language), é uma linguagem de modelagem amplamente usada, oferecendo uma série de convenções diagramáticas úteis para auxiliar a esboçar e documentar sistemas (ALMEIDA, 2006).

O diagrama de componentes faz parte das especificações da UML, por ser estrutural, é usado para representar a arquitetura de uma aplicação. A figura 8 apresenta o diagrama de componentes das funcionalidades que tratam dos eventos: basicamente o fluxo inicia na camada de apresentação(front-end) responsável por interagir com o usuário e o front repassa os dados em formato json para a camada de serviço no back-end. O serviço recebe esses dados através do controlador que chama os métodos da interface serviço implementado pela classe *servicoiml* que valida as regras de negócio. Caso os dados estejam válidos repassa para a interface repositório que faz comunicação com banco de dados, se não retorna uma mensagem de erro para o controlador.

⁸ <https://spring.io/projects/spring-data-mongodb>

Figura 8 – Diagrama de componentes



Fonte: Elaborada pelo autor

3.4.5 Implantação

Para realizar o teste, o sistema foi implantado através de contêineres Docker⁹ versão 20.10.2 ou superiores com auxílio da extensão compose¹⁰ versão 1.27.3 ou superiores, vale ressaltar que o docker também pode ser usado em ambiente de produção, o que torna possível a implantação do sistema sem a necessidade de um servidor java.

Quanto ao processo de instalação, basta gerar os arquivos .jar do módulo back-end e war do módulo front-end. Depois execute o comando: 'docker-compose up'.

Para testar o app mobile é necessário ter o nodejs¹¹ instalado versão 12 ou

⁹ <https://docker.com/>

¹⁰ <https://docs.docker.com/compose/>

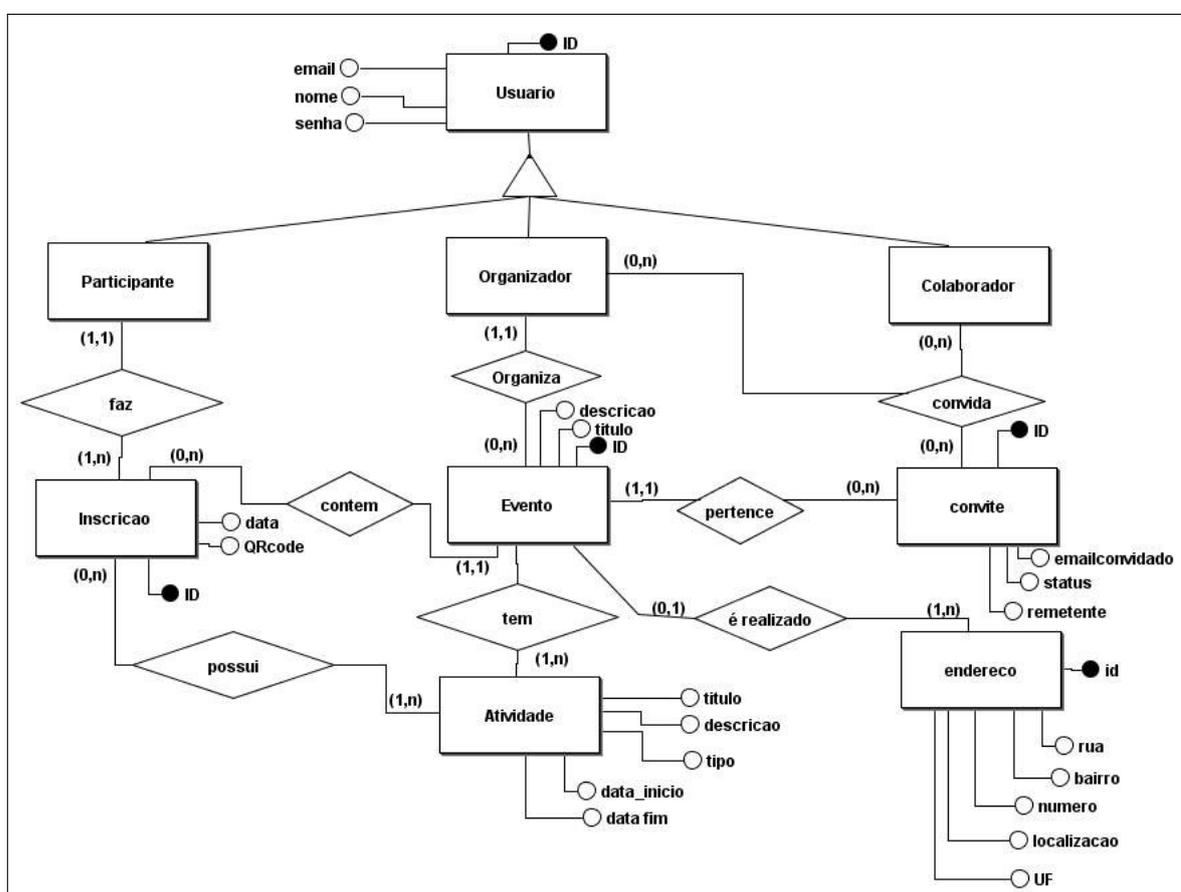
¹¹ <https://nodejs.org/en/>

superiores, e um emulador android ou celular com expor instalado, para executar digite o comando 'expor start' dentro do módulo app (Para executar o app é necessário que o back-end esteja em funcionamento).O código-fonte está disponível no github ¹²

3.4.6 Modelo conceitual

Segundo Souza (2011), um modelo conceitual bem feito é essencial para o sucesso do projeto do banco de dados da aplicação. Mesmo utilizando o mongoDB que é um banco de dados de esquema flexível foi criado um modelo conceitual para o projeto porque esses conceitos não incluem detalhes de implementação, são mais fáceis de entender e auxiliam a comunicação entre os projetistas e os usuários Souza (2011).

Figura 9 – Modelo conceitual



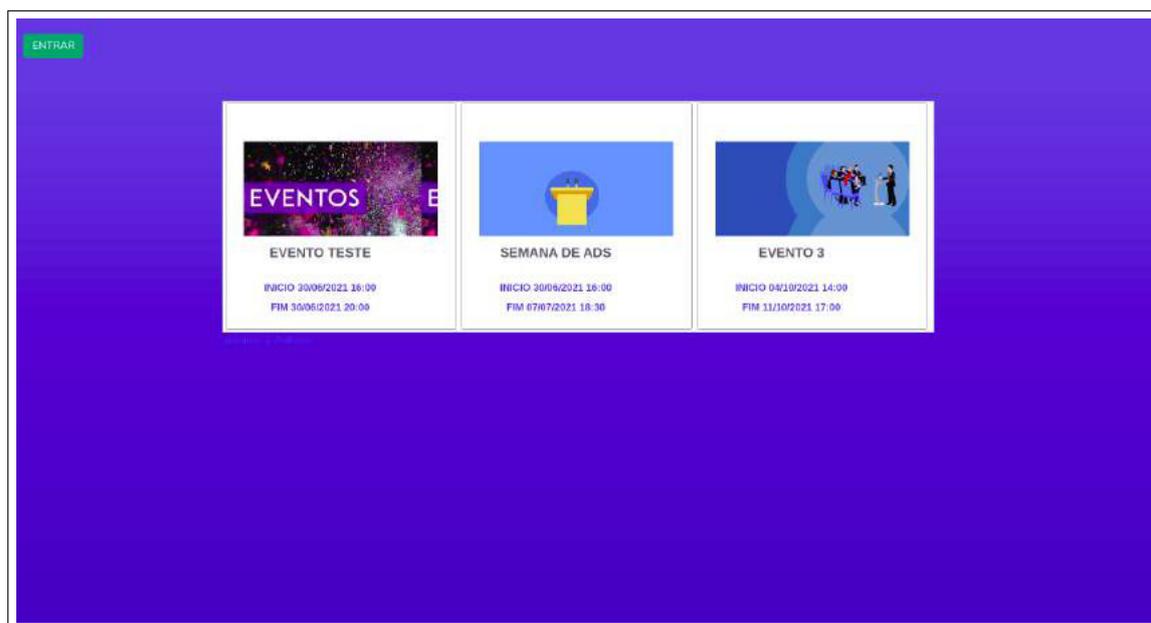
Fonte: Elaborada pelo autor

¹² <https://github.com/joseferreira01/tccll>

3.5 IMPLEMENTAÇÃO

Como pode ser visto na figura 10 ao acessar o sistema o usuário pode ver os eventos que estão disponíveis na plataforma, nessa tela ele pode ver informações como banner, título, data de início e término do evento.

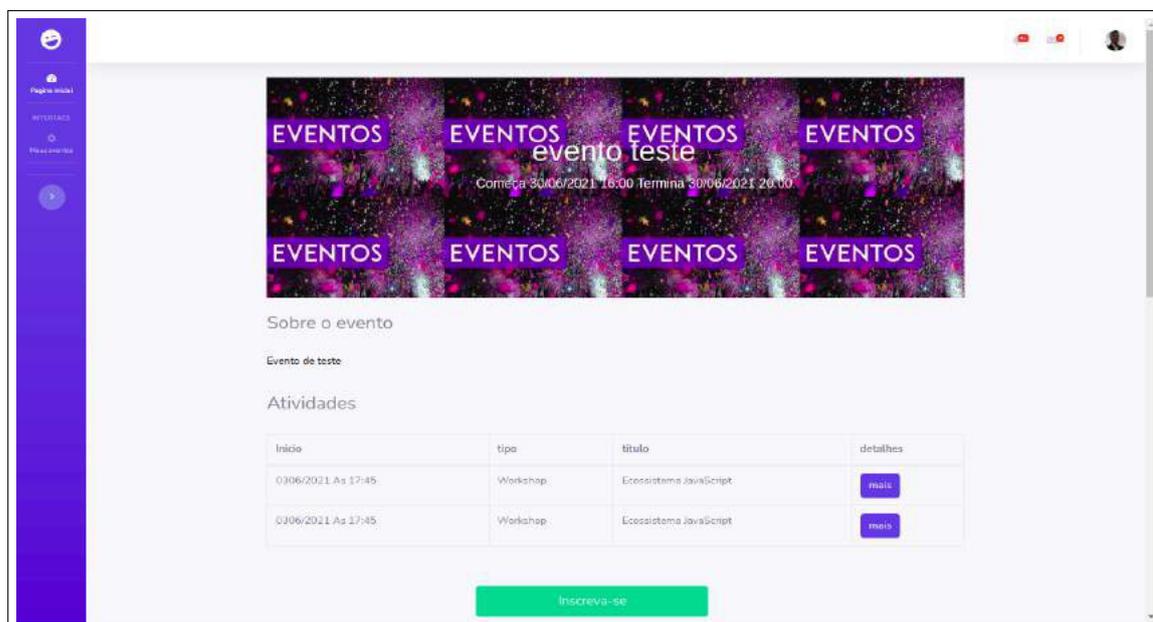
Figura 10 – Tela inicial visão web



Fonte: Elaborada pelo autor

Ao clicar em um desses eventos ele é direcionado para a página de apresentação, Figuras 11 onde ele pode ver informações como banner, data de início e término, atividades e um botão para inscrição no evento.

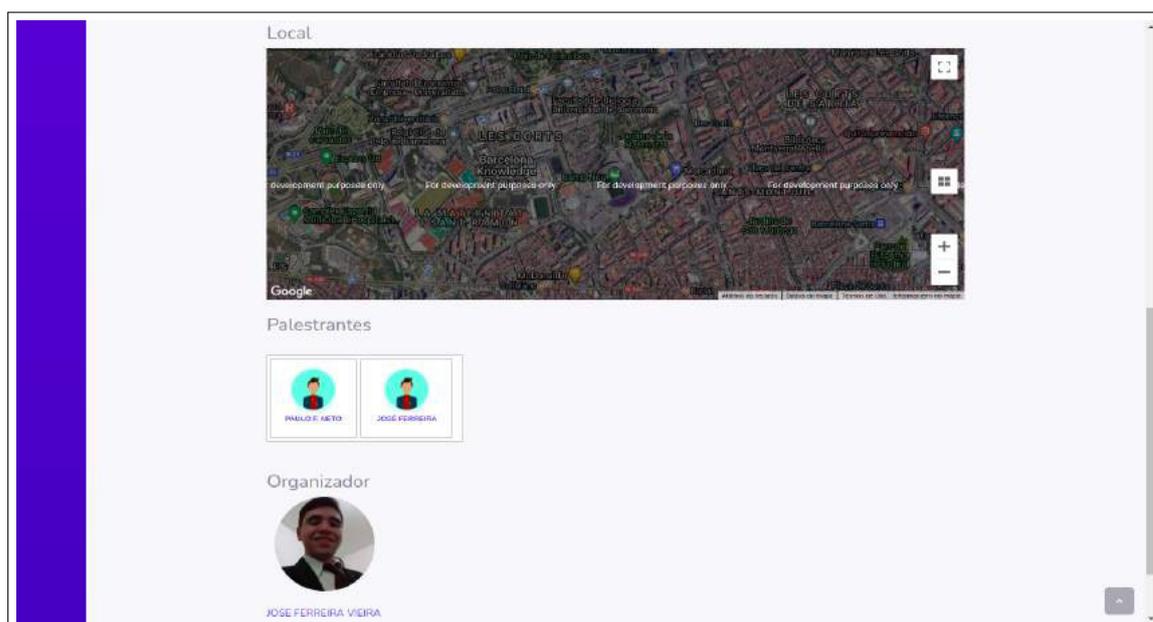
Figura 11 – Tela de apresentação do evento visão web



Fonte: Elaborada pelo autor

A Figura12 mostra mais informações da tela de apresentação do evento como: organizador, palestrantes e a localização

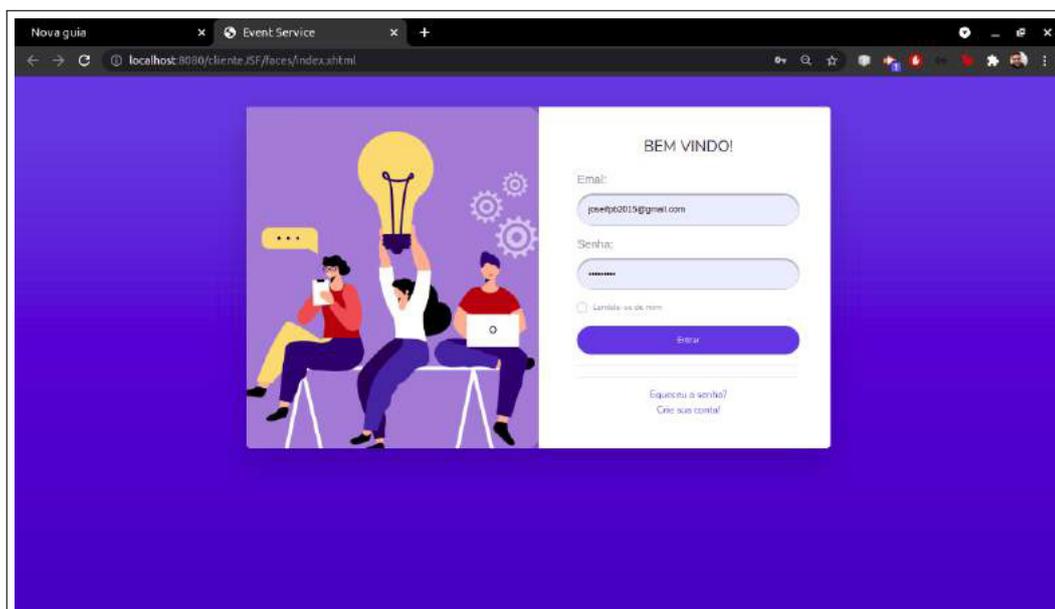
Figura 12 – Tela de apresentação do evento visão web



Fonte: Elaborada pelo autor

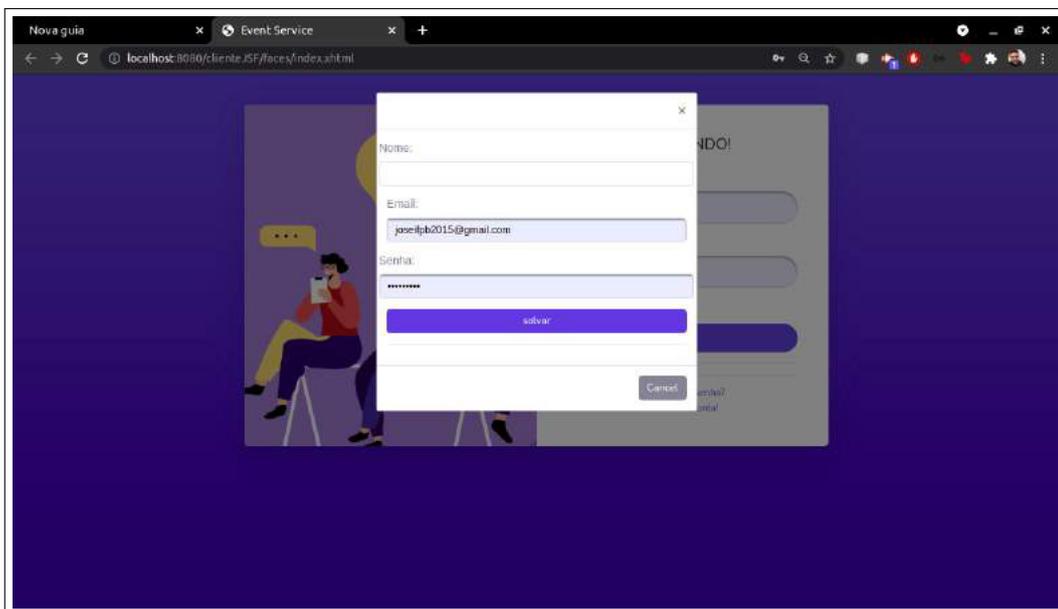
Ao clicar no botão ENTRAR no alto da tela inicial Figura 10 o usuário é direcionado para a página de login Figuras 13 onde ele pode acessar o sistema fazendo login com suas credenciais(email e senha).

Figura 13 – Tela de login versão WEB



Fonte: Elaborada pelo autor

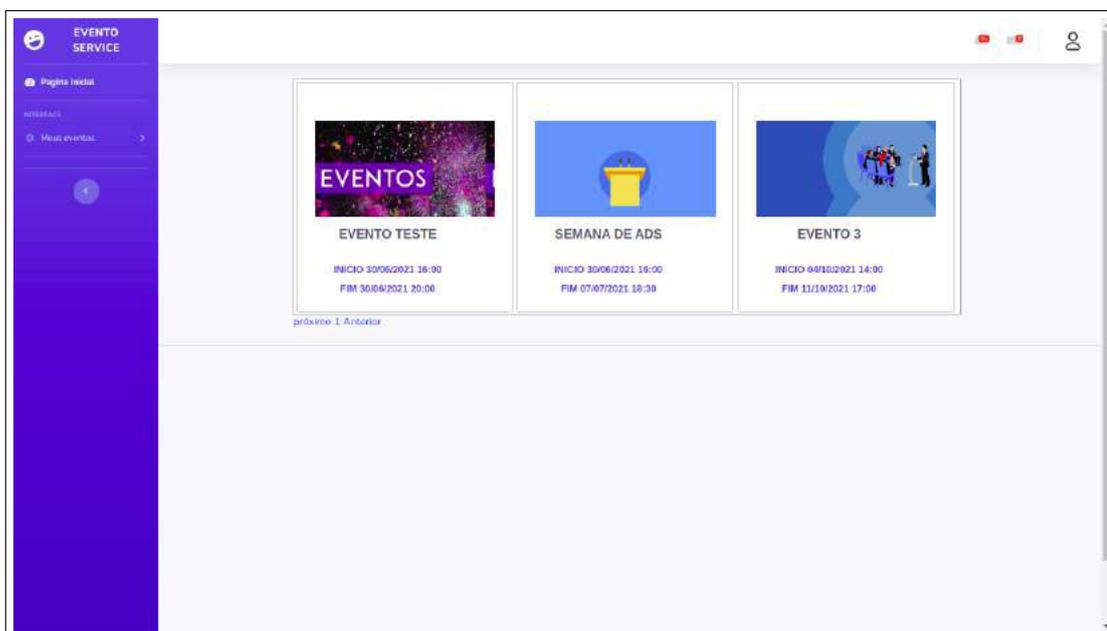
A Figura 14 mostra a tela de cadastro de usuário onde ele caso não tenha cadastro ainda no sistema pode fazê-lo de forma simples informando nome, email e senha (são dados obrigatórios).

Figura 14 – Modal Cadastro de usuário versão WEB

Fonte: Elaborada pelo autor

Após autenticar-se o usuário é direcionado para a página home Figura 15 onde também pode visualizar os eventos disponíveis para inscrição, e ainda clicando no botão “meus eventos” pode criar seu próprio evento Figura 16.

Figura 15 – Tela de home visão WEB



Fonte: Elaborada pelo autor

A Figura 16 apresenta a tela para criar um evento por meio da aplicação. Nesta interface com o usuário é solicitado o preenchimento de algumas informações básicas para o evento (descrição, título, endereço da realização e datas importantes de início e término, assim como, a inserção de uma imagem para marketing e divulgação do evento). Por meio de exemplos apresentados facilita-se o preenchimento das informações. Com todas as informações solicitadas (campos obrigatórios estão marcados com asterisco) preenchidas, após clicar em salvar, o usuário será imediatamente redirecionado para a tela de gerenciamento do evento Figura 17. Neste ambiente o usuário pode adicionar e/ou editar informações básicas, sobre as atividades, convites, colaboradores, participantes, pagamentos, inscrições e confirmação de dados. Com tudo configurado de acordo com que o usuário espera ele pode clicar no botão publicar que disponibilizará o evento para visualização dos usuários interessados.

Figura 16 – Tela para criar evento na aplicação

EventoService x +

localhost:8080/clienteJSF/faces/criar_evento.xhtml

Campos com * são obrigatórios

Descrição*

Titulo*

Rua Bairro Número

Cidade Estado CEP

Localização

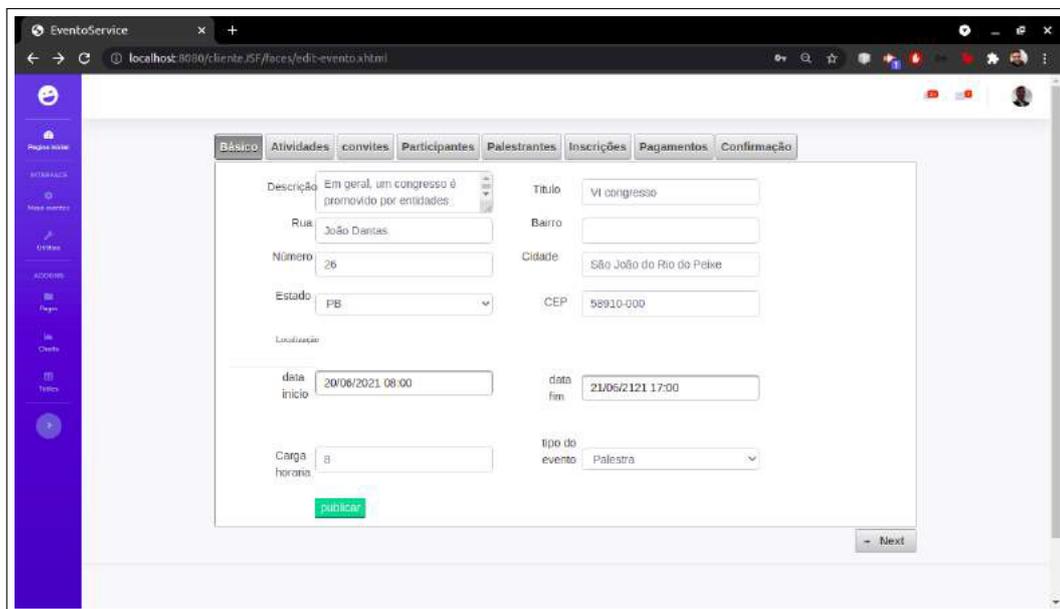
Data inicio* Data Encerramento* Carga horaria

Inicio inscrições Encerramento das inscrições Tipo do evento

banner* Escolher arquivo Nenhum arquivo selecionado salvar

Fonte: Elaborada pelo autor

Figura 17 – Tela para o gerenciamento do evento



The screenshot displays a web browser window titled "EventoService" with the URL "localhost:8080/cliente.jsf/aces/edit-evento.html". The interface features a vertical navigation menu on the left with icons for "Página inicial", "Atividade", "Novo evento", "Editar", "Acessos", "Login", "Senha", and "Tela". The main content area is a form for editing an event, with tabs for "Básico", "Atividades", "convites", "Participantes", "Palestrantes", "Inscrições", "Pagamentos", and "Confirmação". The "Básico" tab is active, showing the following fields:

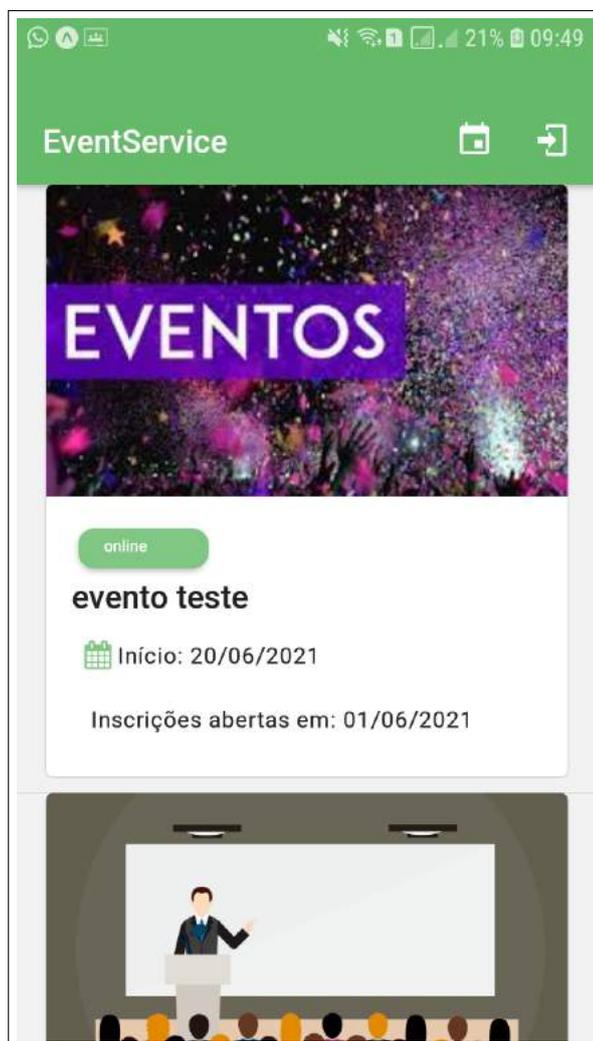
Field	Value
Descrição	Em geral, um congresso é promovido por entidades.
Título	VI congresso
Rua	João Dantas
Bairro	
Número	26
Cidade	São João do Rio do Peixe
Estado	PB
CEP	58910-000
Localização	
data início	20/08/2021 08:00
data fim	21/08/2121 17:00
Carga horaria	8
tipo do evento	Palestra

At the bottom of the form, there is a green "publicar" button and a "Next" button.

Fonte: Elaborada pelo autor

Com o evento publicado, os usuários podem ver no aplicativo a lista dos eventos disponíveis para inscrição Figura 18, onde ele ver informações como: o banner do evento, título, formato (presencial, online ou híbrido), datas de início de inscrições e do início do evento.

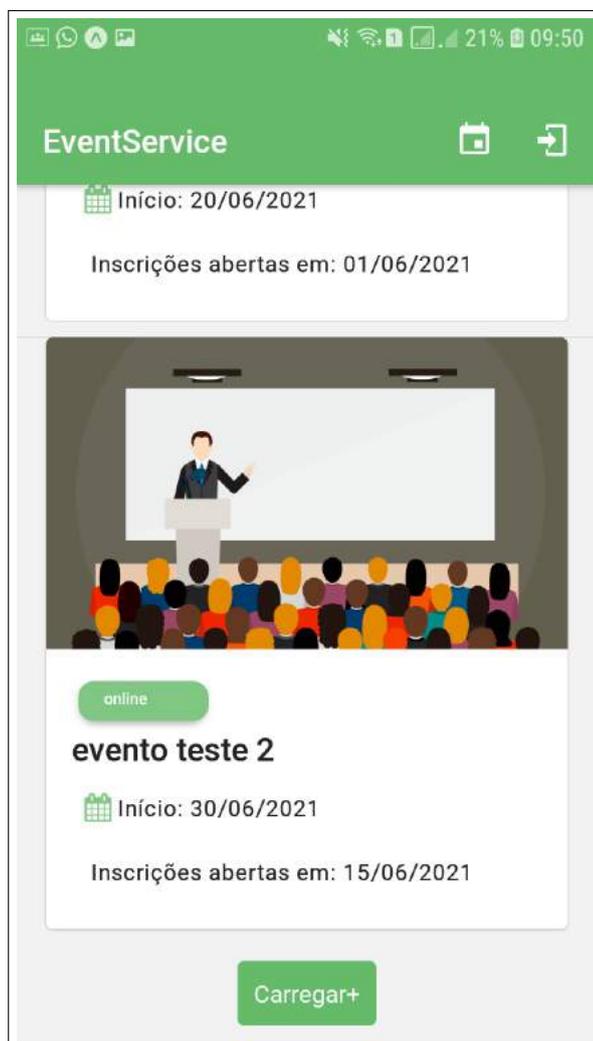
Figura 18 – Tela de principal



Fonte: Elaborada pelo autor

Na Figura 19 podemos ver mais eventos, de início pode ser visto uma lista de 10, se o usuário quiser ver mais é só clicar em carregar+. Ao clicar em um desses eventos o usuário pode ver as informações detalhadas do evento Figura 22.

Figura 19 – Continuação tela de principal



Fonte: Elaborada pelo autor

Na Figura 20 o usuário pode fazer o login no sistema preenchendo os campos email e senha e clicando no botão Entrar, caso tenha esquecido a senha ele pode clicar no botão esqueceu a senha e seguir os passos para criar uma nova, se a pessoa ainda não for usuária do sistema pode criar uma conta clicando no botão criar conta.

Figura 20 – Tela de login



The image shows a mobile application login screen. At the top, there is a green header bar with a white back arrow on the left and the word "Login" in white text. Below the header, the text "EventService" is displayed in a blue font. There are two input fields: "Email" and "Senha", both with light gray backgrounds and thin borders. Below the input fields, there are two buttons: "FECHAR" in an orange box and "ENTRAR" in a green box. At the bottom of the screen, there are two blue links: "Esqueceu a Senha?" and "Criar Conta". The background of the screen is a light gray color. The status bar at the top shows various icons and the time "09:51".

Fonte: Elaborada pelo autor

A tela representada na Figura 21 possibilita que o usuário cadastre-se no aplicativo: informando os dados (nome, email e senha) e clicando no botão Salvar, Versão WEB Figura: 14. Ao finalizar o cadastro com sucesso o usuário é redirecionado para a tela principal, figura 18, onde poderá ver a lista dos eventos disponíveis

Figura 21 – Tela de cadastro



The image shows a mobile application registration screen. At the top, there is a green header bar with a white back arrow on the left and the word 'Cadastro' in white text. Below the header, the main content area has a light gray background. In the center, the text 'Cadastre-se' is displayed in a blue font. Below this, there are three vertically stacked, light gray rectangular input fields with rounded corners. The first field is labeled 'Nome', the second 'Email', and the third 'Senha'. At the bottom of the screen, there are two buttons: an orange button on the left labeled 'CANCELAR' and a green button on the right labeled 'SALVAR'. The top of the screen shows a status bar with various icons and the time '09:51'.

Fonte: Elaborada pelo autor

As demais telas do sistema podem ser vistas no apêndice C.

4 CONCLUSÃO

Content Management Systems (CMS) são sistemas de informação utilizados para facilitar a manutenção do conteúdo na Internet (MENEZES et al., 2016). Diante do problema da criação de uma nova identidade visual cada evento, credenciamento manual e pagamento de taxa sobre o valor de inscrições quando o evento é pago. O EventService, por se tratar de CMS, automatiza várias tarefas na organização de eventos. Com isso, usá-lo na divulgação de eventos acadêmicos é interessante porque o sistema torna a tarefa de criar, manter e atualizar o conteúdo de um evento mais fácil, mesmo para usuários com algum conhecimento técnico. Pessoas com conhecimento considerado mínimo, poderão facilmente criar e gerenciar conteúdo do seu evento.

Tudo isso feito a partir de um editor *front-end*, por onde o usuário conseguirá definir as informações de seu evento; um módulo *back-end* para o processamento e armazenamento dos dados; e funções que obtém o conteúdo e o disponibiliza, com base num modelo predefinido de site.

Inicialmente, não estava planejada a criação de um aplicativo para celular, mas, ao longo do trabalho, notou-se a importância de um *app* para facilitar o uso do sistema por parte do público dos eventos.

Por fim, podemos concluir que conseguimos a partir da Identificação do problema, aplicando análise de software foi possível criar um sistema que possibilita ao usuário mesmo sem conhecimento técnico na área de informática, criar e publicar o seu evento, possibilitando também que o conteúdo acadêmico seja disponibilizado de forma mais acessível para o público tudo isso foi feito utilizando uma estrutura que permite a evolução do sistema de forma simplificada.

4.1 TRABALHOS FUTUROS

O trabalho conseguiu contemplar as funcionalidades planejadas já que o foco era em resolver questões do antes e durante do evento, contudo novas funções podem ser adicionadas para ajudar no pós evento:

- O sistema ainda não emite certificado de participação. O certificado torna mais fácil o acesso a várias atividades de pesquisa relacionadas a uma área de conhecimento..

- O sistema poderia disponibilizar diferentes tipos de relatórios. Ter relatórios detalhados do eventos faz diferença no trabalho. O documento é importante para avaliar os resultados. Além disso, essas informações são de grande valia para uma autoavaliação dos organizadores, que podem identificar os pontos que podem ser melhorados em edições futuras.
- Atualmente o pagamentos de inscrições usa apenas a estratégia do mercadopago, poderia haver outras. Tem várias opções na hora de fazer o pagamento influencia positivamente a pessoa seja para comprar um produto ou adquirir um serviço.

REFERÊNCIAS

ALMEIDA, V. C. de. Uso da linguagem ocl no contexto de diagramas de classe da uml e programas em java. Universidade Federal de Minas Gerais, 2006.

BÁRCIA, L. M. R. C. **A utilização da plataforma Joomla na escola**. Tese (Doutorado) — Universidade Católica Portuguesa, 2008.

BARJIS, J. The importance of business process modeling in software systems design. **Science of Computer Programming**, Elsevier, v. 71, n. 1, p. 73–87, 2008.

BOIKO, B. Understanding content management. **Bulletin of the American Society for Information Science and Technology**, Wiley Online Library, v. 28, n. 1, p. 8–13, 2001.

BONFIM, F. L.; LIANG, M. Aplicações escaláveis com mean stack. **Monografia (Graduação)**, 2014.

CARMO, J. dos S.; PRADO, P. S. T. D. Apresentação de trabalho em eventos científicos: comunicação oral e painéis. **Interação em Psicologia**, v. 9, n. 1, p. 131–142, 2005.

CARNIEL, A. C.; SÁ, A. de A.; BRISIGHELLO, V. H. P.; RIBEIRO, M. X.; BUENO, R.; CIFERRI, R. R.; CIFERRI, C. D. de A. Query processing over data warehouse using relational databases and nosql. In: IEEE. **2012 XXXVIII Conferencia Latinoamericana En Informatica (CLEI)**. [S.l.], 2012. p. 1–9.

CATTELL, R. Scalable sql and nosql data stores. **Acm Sigmod Record**, ACM New York, NY, USA, v. 39, n. 4, p. 12–27, 2011.

CORBELLINI, A.; MATEOS, C.; ZUNINO, A.; GODOY, D.; SCHIAFFINO, S. Persisting big-data: The nosql landscape. **Information Systems**, Elsevier, v. 63, p. 1–23, 2017.

COSTA, P. W. S. d. **Uma abordagem para escolha entre os paradigmas de banco de dados Relacional e NoSQL em projetos de software**. Dissertação (B.S. thesis) — Universidade Federal do Rio Grande do Norte, 2019.

DB4BEGINNERS. **Bancos de Dados Chave-Valor**. 2021. Disponível em: <<http://db4beginners.com/blog/escolha-o-tipo-bd/>>.

_____. **Bancos de Dados orientados a Grafos**. 2021. Disponível em: <<http://db4beginners.com/blog/escolha-o-tipo-bd/>>.

DB4BEGINNERS, S. **Bancos de Dados orientados a Documentos**. 2021. Disponível em: <<http://db4beginners.com/blog/escolha-o-tipo-bd/>>.

DUBINSKY, Y.; HAZZAN, O. extreme programming as a framework for student-project coaching in computer science capstone courses. In: IEEE. **Proceedings 2003 Symposium on Security and Privacy**. [S.l.], 2003. p. 53–59.

ELMASRI, R.; NAVATHE, S. B.; PINHEIRO, M. G. et al. Sistemas de banco de dados. Pearson Addison Wesley São Paulo, 2005.

FIELDING, R. T.; TAYLOR, R. N. **Architectural styles and the design of network-based software architectures**. [S.l.]: University of California, Irvine Irvine, 2000. v. 7. 16 p.

FILHO, H. P. B. **Predição do Comportamento do Mercado Financeiro Utilizando Notícias em Português**. Tese (Doutorado) — PUC–Rio, 2014.

GARCIA, F. P.; LIMA, A. H. G.; FERREIRA, D. d. S.; JUNIOR, F. L.; ROCHA, G. R. C. d.; MENDES, G. W. D.; PONTES, R. F. d.; ROCHA, V. K. d. H.; DANTAS, V. F. easyprocess: Um processo de desenvolvimento para uso no ambiente acadêmico. In: **XII WEI-Workshop de Educação em Computação, XXIV Congresso da Sociedade Brasileira de Computação**. [S.l.: s.n.], 2004.

GARLAND, S. M. Documenting software architectures: Views and beyond. 2nd. **Gynecologic oncology**, Elsevier, n. 2, p. 492, 2010.

HIGA, G. T. A. Bancos de dados orientados a grafos. 2016.

HUMPHREY, W. S. **Managing the software process**. [S.l.]: Addison-Wesley Longman Publishing Co., Inc., 1989.

KYLE, B. MongoDB in action. **Shelter Island: Manning Publications Co**, 2011.

LEMON, S.; ROSSI, K. on object oriented device driver model. In: IEEE. **Digest of Papers. COMPCON'95. Technologies for the Information Superhighway**. [S.l.], 1995. p. 360–366.

LÓSCIO, B. F.; OLIVEIRA, H. d.; PONTES, J. d. S. Nosql no desenvolvimento de aplicações web colaborativas. **VIII Simpósio Brasileiro de Sistemas Colaborativos**, sn, v. 10, n. 1, p. 11, 2011.

MARCHIORI, P. Z.; ADAMI, A.; FERREIRA, S.; CRISTOFOLI, F. Fatores motivacionais da comunidade científica para publicação e divulgação de sua produção em revistas científicas. **Seminário Nacional de Bibliotecas Universitárias**, v. 14, 2006.

MARTIN, V. **Manual prático de eventos**. [S.l.]: Elsevier Brasil, 2017. v. 1.

MEADOWS, A. J.; LEMOS, A. A. B. de L. **A comunicação científica**. [S.l.]: Briquet de Lemos/livros, 1999. 39 p.

MENEZES, J. S. S. d. et al. Processo de avaliação de software aplicado à seleção de sistemas gerenciadores de conteúdo. Universidade Federal de Sergipe, 2016.

MINDTEK.COM.BR, S. **arquitetura de um CMS**. 2021.

Disponível em: <<https://www.mindtek.com.br/2020/03/voce-sabe-o-que-e-sistema-de-gerenciamento-de-conteudo-cms/>>.

NAYAK, A.; PORIYA, A.; POOJARY, D. Tipo de bancos de dados nosql e sua comparação com bancos de dados relacionais. **Jornal Internacional de Sistemas de Informação Aplicados**, v. 5, n. 4, p. 16–19, 2013.

OLIVEIRA, M. de S.; MELO, N. de F.; SANTOS, L. C. dos; OLIVEIRA, W. P. de. Banco de dados no-sql x banco de dados sql. **South American Development Society Journal**, v. 4, n. 11, p. 298, 2018.

PAUTASSO, C.; ZIMMERMANN, O.; LEYMANN, F. Restful web services vs."big"web services: making the right architectural decision. In: **Proceedings of the 17th international conference on World Wide Web**. [S.l.: s.n.], 2008. p. 807.

PEREIRA, F. S.; BORGES, H. P.; SOARES, H. R.; SANTANA, S. A. Utilização de banco de dados nosql em ambientes corporativos. **e-RAC**, v. 3, n. 1, 2013.

POKORNY, J. Nosql databases: a step to database scalability in web environment. **International Journal of Web Information Systems**, Emerald Group Publishing Limited, 2013.

PRESSMAN, R.; MAXIM, B. **Engenharia de Software-8ª Edição**. [S.l.]: McGraw Hill Brasil, 2016.

RICHARDSON, L.; RUBY, S. **RESTful web services**. [S.l.]: O'Reilly Media, Inc., 2008. 16 p.

RYS, M. scalable sql. **Queue**, ACM New York, NY, USA, v. 9, n. 4, p. 30–37, 2011.

SACCOL, D. d. B. Materialização de visões xml. 2001.

SADALAGE, P. J.; FOWLER, M. **NoSQL distilled: a brief guide to the emerging world of polyglot persistence**. [S.l.]: Pearson Education, 2013.

SANTOS, E. U.; SILVA, M. Abordagem ao banco de dados orientado a grafos neo4j em um nível empresarial. 2013.

SANTOS, L. A methodology to apply formal verification to uml-based software. **São José dos Campos**, 2015.

SCHMIDT, L.; OHIRA, M. L. B. Bibliotecas virtuais e digitais: análise das comunicações em eventos científicos (1995/2000) virtual and digital libraries: analysis communications in scientific events (1995-2000) p. 73-97. **Revista ACB**, v. 7, n. 1, p. 73–97, 2002.

SOMMERVILLE. **Engenharia de Software. 9o Edição**. [S.l.]: Pearson Education-BR, 2011.

SOMMERVILLE, I. Engenharia de software, 8 edição. **Pearson, Addison Wesley**, v. 8, n. 9, p. 10, 2007.

SOUZA, C. C. N. **Um metamodelo e uma ferramenta CASE para projeto conceitual de banco de dados segundo o modelo ER**. Dissertação (Mestrado) — Universidade Federal de Pernambuco, 2011.

SOUZA, J. A. Nosql2: administrando banco de dados nosql com a linguagem sql. 2016.

TARGINO, M. das G. Comunicação científica: uma revisão de seus elementos básicos. **Informação & Sociedade**, Universidade Federal da Paraíba-Programa de Pós-Graduação em Ciência da ... , v. 10, n. 2, p. 19, 2000.

TIWARI, S. **Professional NoSQL**. [S.l.]: John Wiley & Sons, 2011.

APÊNDICE A – DOCUMENTO DE VISÃO

Este documento apresenta uma solução de software para ser avaliada como TCC, solicitado pelos professores. Como cliente, o professor Francisco Paulo de Freitas Neto (Orientador).

Demonstra também os problemas que devem ser solucionados, as necessidades dos usuários envolvidos, a abrangência do projeto e as funcionalidades que o sistema deve oferecer.

A.1 OBJETIVO

O objetivo principal desse projeto é facilitar a criação, publicação e gerenciamento de eventos acadêmicos fornecendo uma ferramenta capaz de proporcionar funcionalidades que simplifiquem esse processo.

A.2 ABRANGÊNCIA DO PROJETO

Esse projeto é voltado ao âmbito educacional, contemplando todos aqueles que desejam criar um evento e divulgá-lo. O produto final é uma plataforma que proporciona aos organizadores gerenciar e disponibilizar seu evento e os interessados nesse tipo de conteúdo terão várias opções em um único lugar.

A.3 DEFINIÇÕES, ACRÔNIMOS E ABREVIACÕES

CMS - Content Management System (Sistema de Gerenciamento de Conteúdo)

NOSQL - Not Only SQL

REST - representational State Transfer

A.4 ORGANIZAÇÃO DO DOCUMENTO

Nas seções seguinte trataremos sobre a descrição do problema, os stakeholders envolvidas na utilização do sistema, a solução apontada para os problemas elencados, na sequência, as funcionalidades disponibilizadas por esta solução, depois será elencadas: restrições do sistema e implantação.

A.5 DESCRIÇÃO DO PROBLEMA

Segundo o que foi levantado, no IFPB-Cajazeiras quando se pensa em divulgar um evento geralmente recorrem a empresas de desenvolvimento de software para criar um site, e, qualquer tipo de modificação a ser feita, necessita da intervenção direta da equipe de desenvolvimento.

O impacto deste problema para a organização do evento é a demora, tanto no desenvolvimento quanto na modificação de alguma parte do site. Por exemplo suponha que, por algum motivo queira modificar o formulário de inscrição online. A equipe então, levará um certo tempo (dependendo da complexidade de modificação) até entender completamente o que o cliente deseja para então realizar a modificação e deixar do jeito que o cliente pediu, e é claro, na maioria dos casos, com um custo considerável. Também por diversas circunstâncias é preciso vender entradas. Quando isso acontece, geralmente, é preciso recorrer a uma plataforma de gerência de eventos. O problema das plataformas atuais de gerenciamento de eventos é que elas cobrem cerca de 10% do valor das entradas como taxa de serviço. Então, temos dois problemas principais: demora e custo.

Uma solução ideal permitiria seria um sistema que possibilitasse a criação, personalização e gerenciamento de um evento online sem intervenção de pessoal especializado, e sem taxas. O levantamento de requisitos para a resolução desta problemática foi feita através de entrevistas com a comissão de divulgação de eventos do IFPB Campus Cajazeiras, alunos e professores.

A.6 RESUMO DOS ENVOLVIDOS

Os prováveis interessados neste trabalho são: instituições de ensino, organizadores de eventos acadêmicos e estudantes.

A.7 RESUMO DOS USUÁRIOS

Uma lista resumida de todos os usuários identificados e suas responsabilidades pode ser vista no Quadro 7

Quadro 7 – Resumo dos usuários

Nome	Descrição	Responsabilidades	Envolvido
Organizador de evento	Profissional responsável por planejar e produzir de forma estratégica: conferências, palestras, feiras dentre outros tipos de evento	Criação e conceito do evento, providenciar toda a documentação necessária, pesquisa de orçamentos, contato com fornecedores e equipe de trabalho.	Organizador e equipe
Público	professores, estudantes e todos aqueles que se interessam por produções científicas	Efetuar inscrição para participar do evento	Qualquer pessoa que tenha interesse em uma determinada área de conhecimento científico

Fonte: Elaborado pelo autor ¹

A.8 DESCRIÇÃO DA SOLUÇÃO PROPOSTA

Percebendo que algumas das necessidades inerentes, principalmente ao contexto deste estudo o IFPB, Campus Cajazeiras, não estavam sendo supridas por aplicações/aplicativos disponíveis na internet. Por meio deste estudo, é proposta a construção da ferramenta/aplicação nomeada de “EventoService”. Que terá como característica fazer uso das possibilidades para promoção da geração de conteúdos acessíveis na através Rede Mundial de Computadores para auxiliar no gerenciamento de eventos.

A.9 FUNCIONALIDADES

Na Quadro 8 pode ser visto a descrição das funcionalidades do sistema, o nível de detalhes foi reduzido a fim de facilitar o entendimento.

Quadro 8 – Definição das funcionalidades

Funcionalidades	Descrição
Gerenciar perfil	Inclusão, alteração e remoção de um perfil/-conta no sistema
Gerenciamento de evento	Inclusão, alteração e remoção de um evento no sistema
Convidar colaborador	Enviar convite para um possível colaborador (Pessoa para trabalhar no evento)
Convidar palestrante	Enviar convite para um possível palestrante.
Visualizar evento	Permitir a visualização dos eventos

Fonte: Elaborado pelo autor

A.10 RESTRIÇÕES DO PROJETO

O sistema conta com uma interface WEB e uma mobile, disponível para os usuários: a versão WEB não tem restrições de sistema operacional, já a mobile foi construída com uma tecnologia híbrida (React native) mais somente testado no sistema operacional Android.

APÊNDICE B – ALOCAÇÃO DE TAREFAS E TESTES DE ACEITAÇÃO

Quadro 9 – Alocação de Tarefas - US01 US02

Tarefa	Descrição	Responsável	Estimativa de tempo (Horas)	Tempo gasto (horas)
T1	Configurar base do projeto.	José Ferreira	3	0.3
T2	Implementa script de criação do usuário.	José Ferreira	3	4.5
T3	Implementa script de validação do usuário.	José Ferreira	3	1
T4	Implementa script de enviar email para o usuário.	José Ferreira	1	1
T5	Implementar script para editar o usuário.	José Ferreira	1	1
T6	Implementar script para excluir o perfil.	José Ferreira	1	1
T7	Implementar script para autenticação do usuário.	José Ferreira	6	10
T6	Implementar script para trocar senha.	José Ferreira	2	1

Fonte: Elaborado pelo autor

Quadro 10 – Teste de aceitação - US01 US92

Tarefa	Teste de aceitação	Status
TA1	Quando estou na página de cadastro de novo usuário, ao preencher o formulário com os informações de forma correta, o usuário deve ser inserido. Caso, eu preenche o formulário com os dados obrigatórios de forma incorreta, o usuário não deve ser salvo e uma mensagem de erro deve ser exibida.	Finalizado.
TA2	Ao entrar na página de perfil do usuário, os dados devem ser exibidos de formar que permita edita-los. Ao modificar as informações de forma correta e clicar no botão salvar, os novos dados do usuário devem ser guardados no banco. Se eu preencher os dados obrigatórios de forma incorreta, as alterações não deve ser salvas, e uma mensagem de erro deve ser exibida.	Finalizado
TA3	Dado que estou na página de perfil do usuário ao clicar no botão excluir conta, o sistema deve exibir uma mensagem perguntando se realmente quero excluir a conta, ao clicar em sim a conta deve ser excluída, se clicar em não nada deve acontecer.	Finalizado

Fonte: Elaborado pelo autor

Quadro 11 – Alocação de Tarefas - US05 US06 US07

Tarefa	Descrição	Responsável	Estimativa de tempo (Horas)	Tempo gasto (horas)
T1	Criar script para listar eventos.	José Ferreira	1	1
T2	Implementa script de buscar evento por seu id.	José Ferreira	1	0.5
T3	Implementa script de buscar evento por organizador.	José Ferreira	1	1
T4	Implementar script para editar o usuário.	José Ferreira	1	1
T5	Implementar script para inscrição no evento.	José Ferreira	6	6
T6	Implementar script para cancelar inscrição no evento	José Ferreira	1	1
T7	implementar script para credenciamento evento	José Ferreira	12	10
T8	implementar script para permitir entrada no evento	José Ferreira	4	12
T9	Implementar script para criar layout para exibição das funcionalidades	José Ferreira	15	23

Fonte: Elaborado pelo autor

Quadro 12 – Teste de aceitação - US05 US96 US07

Tarefa	Teste de aceitação	Status
TA1	Quando entro na página inicial do sistema já deve estar disponibilizado os eventos para visualização. Ao clicar no card do evento devo ser redirecionado para uma página onde mostra os detalhes sobre o evento	Finalizado.
TA2	usuário devidamente autenticado no sistema, ao clicar no botão de escreve-se, caso o evento não necessite de informações adicionais a inscrição deve ser confirmada. Deve ser enviado um email de confirmação de inscrição. Se o evento necessita de outros dados do usuário, no momento da inscrição, o usuário é direcionado para uma página onde poderá inserir as informações solicitadas pelos evento. Depois da confirmação dos dados a inscrição é realizado, e o e-mail de confirmação é enviado para o usuário	Finalizado
TA3	No momento da entrada no evento o usuário deve logar no sistema através do seu celular, buscar o evento ao qual deseja credenciar, clicar no botão credenciar-se, será exibido o QR Code de credenciamento na tela do celular: o participante deve mostrar esse código para o colaborador na entrada do evento que verificar as credenciais de acesso.	Finalizado

Fonte: Elaborado pelo autor

APÊNDICE C – TELAS DO SISTEMA

Figura 22, Aqui podemos ver com mais detalhes as informações do evento, nesta tela temos: as datas de início e término do evento, início e término das inscrições, descrição, mapa com a localização do evento, além de um botão para se credenciar e outro para inscrever-se.

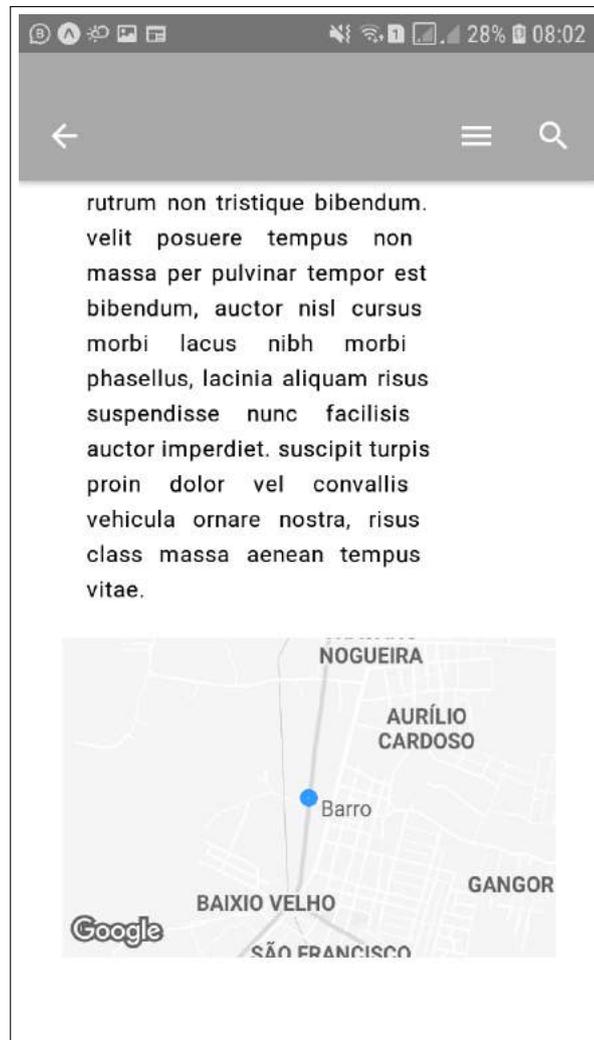
Figura 22 – Tela de detalhe do evento



Fonte: Elaborada pelo autor

Na figura 23, temos a continuação da tela de detalhe do evento: onde é possível ver a descrição do evento e a localização.

Figura 23 – Continua tela de detalhe do evento

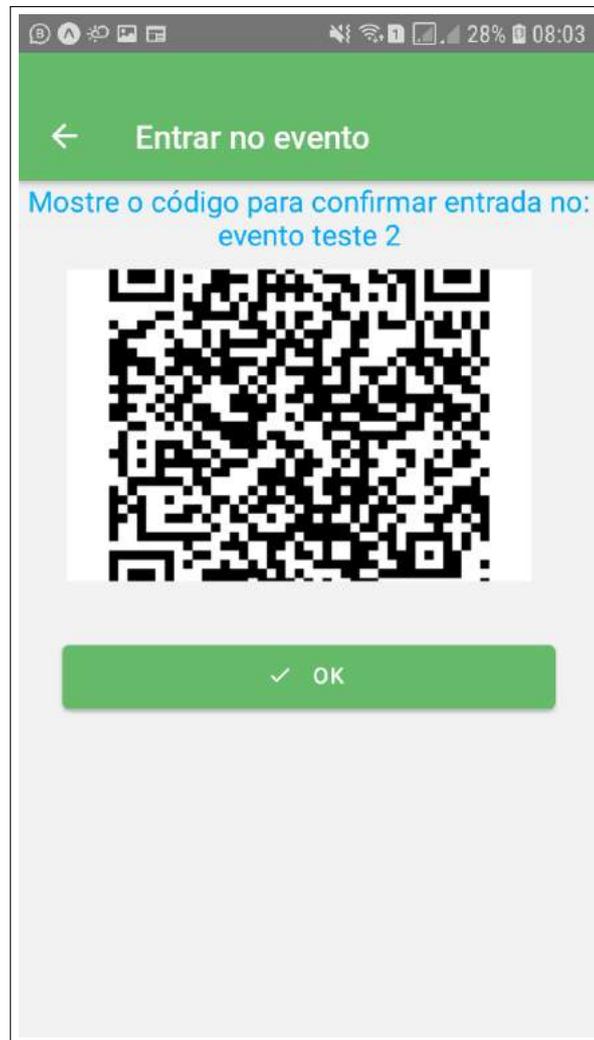


Fonte: Elaborada pelo autor

Ao clicar no botão credenciar-se figura: 22 o sistema exibe o QRcode de autenticação do usuário figura 24, o participante deve mostrar esse QRcode para que possa ser autorizada a entrada no evento.

Essa tela, figura 24 está disponível apenas na versão mobile.

Figura 24 – Tela de credenciamento no evento



Fonte: Elaborada pelo autor

Figura 13 essa é a versão web da tela de login do sistema, arquivo usuário pode fazer login digitando e-mail e senha e clicando no botão entrar. Se o usuário ainda não possui uma conta ele pode clicar no botão criar conta, figura 14, será exibido o modal onde ele poderá digitar nome, e-mail e senha e se cadastrar no sistema. caso tenha esquecido sua senha pode clicar no botão 'esqueceu a senha?' e criar uma nova.

Ao clicar na foto de perfil na tela de home 15 o usuário é redirecionado para tela de perfil Figuras 25 onde pode visualizar as informações do seu perfil e editá-las. Essas informações podem ser solicitadas nas inscrição caso o evento necessite. Esse mesmo fluxo é seguido no aplicativo para o perfil Figura 27.

Figura 25 – Tela de perfil versão WEB

Escolher arquivo Nenhum arquivo selecionado

Dados Pessoais

CPF:

telefone fixo:

celular:

Nacionalidade:

biografia:

tattoos:

linkedin:

Sexo: MASCULINO

Nascimento:

Endereço

rua:

bairro:

numero:

estado: PB

Formação

Formação:

instituição:

area:

Fonte: Elaborada pelo autor

Na Figura 26 a continuação da tela de perfil, caso usuária altere alguma informação ao clicar no botão salvar os dados serão gravados no banco

Figura 26 – Tela de perfil versão WEB

numero: 0

estado: PB

CEP:

Salvar

© Copyright SP-POS Capaxines 2021

Fonte: Elaborada pelo autor

Quando o usuário está logado no aplicativo caso clique na sua foto ele será direcionado para a tela de perfil Figura 27 onde pode visualizar todas as suas informações e alterá-las clicando no botão editar perfil caso deseje.

Figura 27 – Tela de perfil aplicativo



Fonte: Elaborada pelo autor



Documento Digitalizado Ostensivo (Público)

Entrega do TCC Versão final

Assunto: Entrega do TCC Versão final
Assinado por: Jose Ferreira
Tipo do Documento: Tese
Situação: Finalizado
Nível de Acesso: Ostensivo (Público)
Tipo do Conferência: Cópia Simples

Documento assinado eletronicamente por:

- **Jose Ferreira Vieira, ALUNO (201312010100) DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - CAJAZEIRAS**, em 07/07/2021 11:41:43.

Este documento foi armazenado no SUAP em 07/07/2021. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

Código Verificador: 270197

Código de Autenticação: 0f32ee0386

