

CENTRO PAULA SOUZA
ETEC EURO ALBINO DE SOUZA
Ensino Técnico Integrado ao Médio de Desenvolvimento de
Sistemas

Caio Fábio Ferreira da Silva
Luís Gabriel Dias
Maria Fernanda Openheimer
Matheus Barbosa Chambo
Valério Candido da Silva Júnior
Venicius Ferraz de Araujo Neto
Vinícius Parra Alvarenga

S.C.O.R.E: Sistema de Comunicação Offline Rápida Escolar

Mogi Guaçu-SP
2025

Caio Fábio Ferreira da Silva
Luís Gabriel Dias
Maria Fernanda Openheimer
Matheus Barbosa Chambo
Valério Candido da Silva Júnior
Venicius Ferraz de Araujo Neto
Vinícius Parra Alvarenga

S.C.O.R.E: Sistema de Comunicação Offline Rápida Escolar

Trabalho de Conclusão de Curso apresentado ao Curso Técnico Integrado ao Médio de Desenvolvimento de Sistemas da ETEC Euro Albino de Souza, orientado pelo Prof. Maercio Girardi Bisco e Pedro Ramires da Silva Amalfi Costa, como requisito parcial para obtenção do título de técnico em Desenvolvimento de Sistemas.

Mogi Guaçu-SP

2025

S586s

Silva, Caio Fábio Ferreira da; Dias, Luís Gabriel; Openheimer, Maria
Fernanda; Chambo, Matheus Barbosa; Silva Júnior, Valério Candido
da; Alvarenga, Vinícius Parra.

S.C.O.R.E - Sistema de Comunicação Offline Rápida Escolar. – Mogi Guaçu –
SP. 2025.

177 p.

Trabalho de Conclusão do Curso de Ensino Médio com Habilitação
Profissional de Técnico em Desenvolvimento de Sistemas da ETEC
Euro Albino de Souza de Mogi Guaçu.

Orientadores: Professor Especialista Maércio Girardi Bisco e
Professor Mestre Pedro Ramires da Silva Amalfi Costa

1. Aplicativo. 2. Mensagens. 3. Conectividade. I. Bisco, Maércio
Girardi; Costa, Pedro Ramires da Silva Amalfi (orientadores) II. Título.

CDD 371.102

Bibliotecário Guerreiro Ímpar Nunes Guerreiro – CRB-8/8137

DEDICATORIA

Dedicamos o nosso trabalho de conclusão de curso a ETEC e ao professor Pedro Ramires da Silva Amalfi que esteve conosco desde nossos primeiros passos na programação até nosso TCC.

AGRADECIMENTO

Agradecemos primeiramente a Deus e aos nossos familiares por todo apoio e incentivo desde o nosso início na ETEC até o nosso último ano, agradecemos também aos professores que nos ensinaram e nos apoiaram durante toda caminhada na ETEC.

EPÍGRAFE

“A tecnologia move o mundo”

Steve Jobs

RESUMO

S.C.O.R.E ou Sistema de Comunicação *Offline* Rápida Escolar é um TCC (Trabalho de Conclusão de Curso) que consiste na criação de um aplicativo de comunicação *offline*. A proposta é uma ferramenta de comunicação ágil, eficaz e funcional, especialmente no ambiente escolar onde a conectividade com *internet* é limitada. O aplicativo permite que os usuários troquem mensagens sem precisar de uma conexão com a *Internet*, isso é permitido pois os computadores estão conectados a um servidor local para a distribuição dessas mensagens desde que estejam com os cabos conectados aos computadores.

Palavras-Chave: Comunicação, *Offline*, Mensagens.

ABSTRACT

S.C.O.R.E or School Rapid Offline Communication System is a TCC (Course Completion Project) that consists of creating an offline communication application. The proposal is an agile, effective and functional communication tool, especially in the school environment where internet connectivity is limited. The application allows users to exchange messages without needing an internet connection, this is possible because the computers are connected to a local server for the distribution of these messages as long as the cables are connected to the computers.

Keywords: *Communication, Offline, Messages.*

LISTA DE ILUSTRAÇÃO

Figura 1– Tela de Login.....	61
Figura 2 – Tela de Cadastro.....	62
Figura 3 - Tela de Recuperação de Senha.....	63
Figura 4 – Tela verificação de código.....	64
Figura 5 – Tela de Criação de Nova Senha	65
Figura 6 – Confirmação de Alteração de Senha.....	66
Figura 7 – Tela Principal.....	67
Figura 8 – Tela de envio de mensagem	68
Figura 9 – Tela de Seleção e Encaminhamento de Mensagem.....	69
Figura 10 – Tela de selecionar qual local de envio	70
Figura 11 – Tela de selecionar o computador de envio.....	71
Figura 12 – Tela de mensagem encaminhada	72
Figura 13 – Tela de informações do usuário (Administrador).....	73
Figura 14 – Tela de editar informações do usuário	74
Figura 15 – Tela de editar conversas	75
Figura 16 – Tela de Edição de Notificações.....	76
Figura 17 – Tela de Configurações Visualizada por Outros Usuários.....	77
Figura 18 - Diagrama Envio de Mensagem	114
Figura 19 - Diagrama Login.....	115
Figura 20 - Diagrama Gerenciamento de Dados.....	117
Figura 21 - Tela de login.....	120
Figura 22 - Código Tela de Login	122
Figura 23 - Tela inicial do sistema S.C.O.R.E após login da secretaria	123
Figura 24 - Código Tela Inicial 1.....	124
Figura 25 - Código Tela da Tela inicial 2	125
Figura 26 – Código da Tela Inicial 3.....	125
Figura 27 - Tela Lista de Usuários	126
Figura 28 Tela de edição de usuário	126
Figura 29 – Tela de Confirmação de exclusão de usuário	127
Figura 30 - Código Tela Listas de Usuários 1	130
Figura 31 - Código Tela Lista de Usuários 2	130
Figura 32 - Código Tela Lista de Usuários 3	131
Figura 33 - Código Tela Lista de Usuários 4	131
Figura 34 - Tela Cadastrar PC	132
Figura 35 - Código da Tela Cadastrar PC	134
Figura 36 - Tela cadastro SCORE.....	135
Figura 37 - Código Tela Cadastro SCORE.....	137
Figura 38 - Código tela de Cadastro SCORE 2.....	137
Figura 39 - Lista de computadores cadastrados	138
Figura 40 - Edição de informações de um computador	138
Figura 41 - Tela Confirmação de Exclusão	139
Figura 42 - Liberação de acesso ao MySQL pelo IP	139
Figura 43 - Código Lista de computadores 1.....	141
Figura 44 - Código Lista de computadores 2.....	142

Figura 45 - Código Lista de computadores 3.....	142
Figura 46 - Código Lista de computadores 4.....	143
Figura 47 - Código Lista de computadores 5.....	143
Figura 48 - Tela Esqueci Senha	144
Figura 49 - Código da Tela Esqueci senha 1	146
Figura 50 - Código da Tela Esqueci senha 2	146
Figura 51 - Código da Tela Esqueci senha 3	147
Figura 52 - Código da Tela Esqueci senha 4	147
Figura 53 - Tela Chat Secretaria	148
Figura 54 - Tela Seleção de mensagens.....	148
Figura 55 - Tela Mensagens Seleccionadas para exclusão	149
Figura 56 - Tela Confirmação de exclusão.....	150
Figura 57 - Código Chat 1	151
Figura 58 - Código Chat 2	152
Figura 59 - Tela de Configuração.....	152
Figura 60 - Código tela Configuração.....	154
Figura 61 - Redefinir Senha	155
Figura 62 - Código da Tela Redefinir Senha 1	157
Figura 63 - Código da Tela Redefinir Senha 2	157
Figura 64 - Código Redefinir Senha 3.....	158
Figura 65 - Tela de confirmação de saída do sistema.....	158
Figura 66 - Código tela de confirmação de saída do sistema.....	160
Figura 67 - Tela de Chat do Sistema S.C.O.R.E Usuário.....	160
Figura 68 - Tela de Configurações do Chat.....	161
Figura 69 - Tela edição de Perfil do Usuário	162
Figura 70 - Tela de Chat com Mensagens Seleccionadas.....	162
Figura 71 - Tela seleção de mensagens para exclusão	163
Figura 72 - Tela Confirmação de Exclusão de Mensagens.....	163
Figura 73 - Código tela Chat Usuário 1	166
Figura 74 - Código tela Chat Usuário 2	166
Figura 75 - Código tela Chat Usuário 3.....	167
Figura 76 - Código tela Chat Usuário 4.....	167
Figura 77 - Código tela Chat Usuário 5.....	168

LISTA DE TABELAS

Tabela 1- Tabela de Cronograma	47
Tabela 2 - Tabela de Cálculo	49

LISTA DE ABREVIATURAS E SIGLAS

ANATEL – Agência Nacional de Telecomunicações

ANSI – *American National Standards Institute* (Instituto Nacional Americano de Padrões)

API – *Application Programming Interface* (Interface de Programação de Aplicações)

ARPANET – *Advanced Research Projects Agency Network* (Agência para Projetos de Pesquisa Avançada)

AWS – *Amazon Web Services* (Serviços Web da Amazon)

BD – Banco de Dados

BDFL – *Benevolent Dictator for Life* (Ditador Benevolente Vitalício)

BPMN – *Business Process Model and Notation* (Modelo e Notação de Processos de Negócio)

CERN – *Organisation Européenne Pour La Recherche Nucléaire* (Conselho Europeu Para Pesquisa Nuclear)

CODASYL – *Conference on Data Systems Languages* (Comitê de Linguagens de Sistemas de Dados)

CORBA – *Common Object Request Broker Architecture* (Arquitetura de Corretor de Requisições de Objetos Comuns)

CSS – *Cascading Style Sheets* (Folhas de Estilo em Cascata)

CTE – *Common Table Expressions* (Expressões de Tabela Comuns)

DIO – *Digital Innovation One* (Inovação Digital Um)

DOU – Diário Oficial da União

EPC – *Event-driven Process Chain* (Cadeia de Processos Orientada por Eventos)

GB – *Gigabyte*

Gmail – *Google Mail* (Correio Eletrônico do Google)

Google Meet – *Google Meeting Platform* (Plataforma de Reuniões do Google)

GUI – *Graphical User Interface* (Interface Gráfica do Usuário)

HP – *Hewlett-Packard*

HTML – *Hypertext Markup Language* (Linguagem de Marcação de Hipertexto)

HTTP – *Hypertext Transfer Protocol* (Protocolo de Transferência de Hipertexto)

IBM – *International Business Machines Corporation* (Corporação Internacional de Máquinas de Negócios)

IDE – *Integrated Development Environment* (Ambiente Integrado de Desenvolvimento)

IP – *Internet Protocol* (Protocolo de Internet)

ISO – *International Organization for Standardization* (Organização Internacional de Padronização)

JIT – *Just-In-Time* (Justo a Tempo)

JSON – *JavaScript Object Notation* (Notação de Objeto JavaScript)

LAN – *Local Area Network* (Rede Local)

LGPD – Lei Geral de Proteção de Dados Pessoais

LGT – Lei Geral de Telecomunicações

LTS – *Long Term Support* (Suporte de Longo Prazo)

MDarray – *Multidimensional Array* (Vetor Multidimensional)

Microsoft 365 – *Microsoft Productivity Suite* (Pacote de Produtividade da Microsoft)

ML – *Machine Learning* (Aprendizado de Máquina)

NoSQL – *Not Only SQL* (Não Somente SQL)

OMA – *Object Management Architecture* (Arquitetura de Gerenciamento de Objetos)

OMG – *Object Management Group* (Grupo de Gerenciamento de Objetos)

PA – *Public Address* (Sistema de Endereçamento Público)

PC – *Personal Computer* (Computador Pessoal)

PEP – *Python Enhancement Proposal* (Proposta de Melhoria do Python)

PK – *Primary Key* (Chave Primária)

Py3k – Apelido para a linha de lançamento da versão do Python 3.x, originado de “Python 3000”

Qt – *Framework* gráfico (Quadro gráfico)

RAM – *Random Access Memory* (Memória de Acesso Aleatório)

S.C.O.R.E – Sistema de Comunicação *Offline* em Rede Escolar

SEQUEL – *Structured English Query Language* (Linguagem Estruturada de Consulta em Inglês)

SGBD – Sistema de Gerenciamento de Banco de Dados

Slack – *Searchable Log of All Conversation and Knowledge* (Registro Pesquisável de Todas as Conversas e Conhecimento)

SMS – *Short Message Service* (Serviço de Mensagens Curtas)

SP – Estado de São Paulo

SQL – *Structured Query Language* (Linguagem de Consulta Estruturada)

TB – *Terabyte*

TCC – Trabalho de Conclusão de Curso

TCP/IP – *Transmission Control Protocol / Internet Protocol* (Protocolo de Controle de Transmissão / Protocolo de Internet)

TI – Tecnologia da Informação

UML – *Unified Modeling Language* (Linguagem de Modelagem Unificada)

URL – *Uniform Resource Locator* (Localizador Padrão de Recursos)

WhatsApp – *WhatsApp Messenger* (Mensageiro *WhatsApp*)

WWW – *World Wide Web* (Rede Mundial de Computadores)

XAMPP – Conjunto de ferramentas: X (multiplataforma), *Apache*, *MariaDB/MySQL*, *PHP* e *Perl*

SEQUEL - *Structured English Query Language* (Linguagem Estruturada de Consulta em inglês)

LISTA DE SÍMBOLOS

SUMARIO

INTRODUÇÃO	21
1. TECNOLOGIAS	23
1.1 Lógica de programação.....	23
1.2 Linguagem de programação	23
1.2.1 <i>Desktop</i>	24
1.3 PROGRAMAÇÃO WEB.....	25
1.3.1 HTML.....	25
1.3.2 JavaScript.....	28
1.3.3 CSS	29
1.4 Servidores.....	30
1.4.1 Servidor local	31
1.4.2 Servidor <i>Web</i>	31
1.5 Banco de dados	32
1.5.1 Modelagem	33
1.5.1.1 Modelo conceitual	33
1.5.1.1.1 Entidade	33
1.5.1.1.2 Relacionamento	33
1.5.1.1.3 Atributo	34
1.5.1.2 Modelo Logico	34
1.5.1.3 Modelo Físico.....	34
1.6 Br Modelo.....	34
1.6.1 BrModelo <i>Web</i>	35
1.7 SQL.....	35
1.8 XAMPP	37
1.9 MariaDB.....	38
1.10 <i>Design</i>	40
1.10.1 Inkscape	41
2.1 VIABILIDADE	42
2.1.1 VIABILIDADE OPERACIONAL.....	42
2.1.1.1 - Algoritmo de Comunicação em Rede Local.....	42
2.1.1.2 Tecnologias Utilizadas	43
2.1.2 Viabilidade Técnica.....	44

2.1.2.1 Back-End para Lógica e Envio de Mensagens:.....	44
2.1.2.2 Tecnologias:	45
2.1.2.3 Banco de Dados:	45
2.1.2.4 Ferramentas de Protocolo:.....	45
2.1.2.5 Autenticação e Segurança:	46
2.1.2.6 Recursos do Software:.....	46
2.1.2.7 Recursos Futuramente Adicionais:	46
2.1.2.8 Conclusão da Viabilidade Técnica:	47
2.1.3 - Viabilidade de Cronograma	47
2.1.4- Viabilidade Econômica.....	48
2.1.4.1 Custo de uma Aplicação:	49
2.1.4.2 Manutenção e custos:.....	49
2.1.4.3 Cálculo de viabilidade:	49
2.2 TÉCNICA DE LEVANTAMENTO DE REQUISITOS	50
2.2.1-Entrevista	51
2.2.1.1-Entrevista aos Professores e Funcionários.....	51
2.2.2 Análise de texto	55
2.3 prototipagem	60
2.3.1 Aplicativo <i>Desktop</i>.....	60
2.3.1.1 tela de login	61
2.3.1.2 Tela de cadastro	62
2.3.1.3 Tela de Recuperação de Senha	63
2.3.1.4 Tela verificação de código	64
2.3.1.5 Tela de Criação de Nova Senha.....	65
2.3.1.6 tela de Confirmação de Alteração de Senha	66
2.3.1.7 tela principal	67
2.3.1.8 tela de envio de mensagem	68
2.3.1.9 Tela encaminhar ou selecionar	69
2.3.1.10 selecionar qual local do envio	70
2.3.1.11 Tela de selecionar o computador de envio	71
2.3.1.12 tela de mensagem encaminhada.....	72
2.3.1.13 tela informações do usuário (Administrador)	73

2.3.1.14 tela de editar informações do usuário	74
2.3.1.15 tela editar conversas	75
2.3.1.16 tela de editar notificações	76
2.3.1.17 Tela dos outros usuários.....	77
2.4-CLIENTE.....	78
2.4.1-Cliente Interno.....	78
2.4.2 Cliente Externo.....	79
2.4.2.1- Cliente Ativo	80
2.4.2.2- Cliente Inativo.....	81
2.4.3 Cliente Satisfeito.....	83
2.4.4-Cliente Insatisfeito	85
2.5 Área de Atuação	85
2.5.1 Microrregião	86
2.5.2-Macroregião	87
2.5.2.1 Perfil dos Influenciadores.....	88
2.5.2.2 Estimativa de Alcance para Empresas.....	89
2.6 Concorrentes	90
2.6.1 Concorrentes Diretos	91
2.6.2 Concorrentes Indiretos	93
2.7 LEIS E REGRAS CONTRATUAIS.....	94
2.7.1 Relatório de Termos e Condição.....	95
2.7.1.1 Legislação Municipal – Mogi Guaçu (SP).....	96
2.7.1.2 Legislação Estadual – São Paulo	96
2.7.1.3 Legislação Federal – Brasil.....	96
2.7.2 Medidas e Ações.....	98
2.7.2.1 No âmbito municipal – Mogi Guaçu (SP)	98
2.7.2.2 No âmbito estadual – Estado de São Paulo.....	99
2.7.2.3 No âmbito federal – Legislação nacional	99
2.7.3 – CONTRATO DE PRESTAÇÃO DE SERVIÇOS.....	101
3 – DOCUMENTAÇÃO	106
3.1 – OMG.....	106
3.1.1 – Principais padrões do OMG (sintético).....	107
3.1.2 – Utilização no Sistema	108

3.2 BPMN	109
3.2.1 Implementação no Sistema.....	112
3.2.2 Exemplo de aplicação	113
3.3 DIAGRAMA	113
3.3.1 Diagrama de envio de mensagem	114
3.3.2 – Diagrama <i>Login</i>.....	115
3.3.3 – Diagrama Gerenciamento de Dados	117
3.4- TELAS	120
3.4.1 Tela de <i>Login</i>.....	120
3.4.2 Tela inicial do sistema S.C.O.R.E após <i>login</i> da secretaria	122
3.4.3 Tela de Lista de Usuários.....	125
3.4.4 Tela de Cadastro de PC.....	132
3.4.5 Tela Cadastro de Usuário.....	135
3.4.6 Tela Lista de computadores cadastrados	138
3.4.7 Tela Esqueci a Senha	144
3.4.8 Tela <i>Chat</i> Secretaria	148
3.4.9 Tela Configuração.....	152
3.4.10 Tela Redefinir Senha	155
3.4.11 Tela de confirmação de saída do sistema S.C.O.R.E.....	158
3.4.12 Tela de <i>Chat</i> do Sistema S.C.O.R.E Usuário.....	160
Referências bibliograficas.....	171

INTRODUÇÃO

A comunicação é o pilar principal para um funcionamento eficiente de qualquer instituição, principalmente em ambientes escolares, onde a comunicação entre professores, coordenadores, direção, secretaria e demais funcionários é essencial para garantir um ambiente organizado, com um bom andamento nas atividades pedagógicas e atividades administrativas. Todavia, escolas enfrentam dificuldades relacionadas à estrutura e à limitação de acesso à *internet*, deixando a comunicação interna mais lenta e cansativa.

Com o avanço da tecnologia, diversas ferramentas para melhorar a comunicação surgiram, suprimindo essa necessidade. Entretanto, a maioria dessas soluções necessita de uma conexão com a *internet*, o que as torna inviáveis em alguns contextos, como escolas em áreas rurais ou em regiões com sinal instável, até mesmo em escolas onde os recursos financeiros para manter a rede são restritos. Esses cenários evidenciam a necessidade de um desenvolvimento tecnológico que funcione sem precisar de conexão com a *internet*, usando recursos estruturais como cabeamento estruturado.

Diante desses problemas, este TCC (Trabalho de Conclusão de Curso) propõe o desenvolvimento do S.C.O.R.E – Sistema de Comunicação *Offline* Rápida Escolar, ou seja, um aplicativo voltado para melhorar a comunicação, com a principal característica de não precisar de uma conexão com a *internet*, permitindo que os usuários troquem mensagens por meio de uma rede local, utilizando cabos conectados a um servidor responsável por gerenciar as mensagens.

O S.C.O.R.E foi idealizado para oferecer uma solução prática, eficiente e de fácil implementação, suprimindo as dificuldades da comunicação enfrentadas nas instituições, principalmente em escolas com pouca ou nenhuma conectividade externa. O aplicativo possibilita uma troca de mensagens com foco na simplicidade do uso e em sua eficiência.

O trabalho apresenta a concepção, o desenvolvimento e os testes do aplicativo, discutindo aspectos técnicos como arquitetura de rede, protocolos de comunicação, estrutura de banco de dados, interface gráfica e segurança das mensagens.

Ao final, o projeto S.C.O.R.E busca apresentar uma inovação na tecnologia aplicável aos ambientes com conexão instável, bem como destacar a importância de soluções criativas, funcionais e acessíveis, contribuindo assim para uma gestão conectada e eficiente, mesmo com baixa conexão com a *internet*.

1. TECNOLOGIAS

Para o desenvolvimento deste projeto foram utilizadas tecnologias, baseadas em análises e pesquisas.

Atualmente a tecnologia está presente no cotidiano, desempenhando papel fundamental e trazendo muitos benefícios, onde um deles é o tempo poupado, valorizando o tempo pessoal. Em uma sociedade conectada, com os processos tecnológicos acelerados, a forma de lidar com prazos e tarefas foi transformada.

No mundo contemporâneo se faz como um dos pilares, desempenhando um papel além de criações de ferramentas simples. Impacta desde a inovação, a disseminação de informações à construção de soluções inteligentes para os desafios da modernidade.

1.1 LÓGICA DE PROGRAMAÇÃO

A lógica de programação é o conjunto de regras e técnicas que os programadores utilizam para projetar e desenvolver programas de computador. É a habilidade de pensar de forma lógica e estruturada. É a base fundamental para o desenvolvimento de todo tipo de *software*.

Todas as linguagens de programação contêm lógica, desde as linguagens mais simples e básicas até as mais complexas e estruturadas.

No processo para aprender programação, o primeiro passo é estudar a lógica, para que o estudante entenda e aprenda a pensar da mesma forma que um programador, analisar o problema encontrado e buscar soluções de formas lógicas e eficazes.

1.2 LINGUAGEM DE PROGRAMAÇÃO

Linguagem de Programação é uma linguagem escrita e formal que especifica um conjunto de instruções e regras usadas para gerar programas (*software*). Um *software* pode ser desenvolvido para rodar em um computador, dispositivo móvel ou

em qualquer equipamento que permita sua execução. Essa linguagem permite a comunicação entre máquina e ser humano de forma estruturada, possibilitando transmitir comandos que a máquina pode interpretar e executar.

Existem várias linguagens de programação, com características e sintaxes específicas. Duas das linguagens mais conhecidas são Python e C++, além dessas, existem muitas linguagens atualmente. A escolha da linguagem a ser usada depende de vários fatores, como, analisar o tipo de projeto, o desempenho esperado, as funcionalidades desejadas e com quais linguagens os desenvolvedores estão mais familiarizados.

Assim ter autoridade sobre uma linguagem de programação é fundamental para todos que desejam atuar na área de tecnologia da informação, pois é pelo conhecimento das linguagens que se torna possível automatizar tarefas, desenvolver sistemas e aplicativos que contribuem para o cotidiano.

1.2.1 Desktop

Desktop é um tipo de computador de mesa projetado para uso em local fixo, usualmente sobre uma estação de trabalho. Ao contrário de *notebooks* e *tablets* que são aparelhos moveis projetados para portabilidade, esse tipo de computador é composto por componentes separados individualmente, como teclado, monitor, *mouse* e unidade central de processamento o que oferece melhor desempenho.

Atualmente existem uma grande variedade de *desktops* no mercado, cada modelo pensado em uma necessidade específica, são utilizados para uso particular, residencial, em escolas, escritórios para atividades gerais. Também existem computadores mais robustos utilizados para atividades específicas e que exigem alto desempenho e rendimento.

Com o avanço da tecnologia os *desktops* também evoluíram, se adaptando as mudanças e demandas tecnológicas. Continuam sendo uma ferramenta fundamental para a computação pessoal e profissional, oferecendo qualidade e custo-benefício.

1.3 PROGRAMAÇÃO WEB

Programação Web: é definida como o conjunto de técnicas e linguagens usadas para criar aplicações e sites acessíveis via navegadores de internet. Ela permite desenvolver páginas estáticas e dinâmicas, interatividade, comunicação com servidores e integração com bancos de dados.

Em resumo, a programação web é a base para a construção da web moderna, permitindo que conteúdos e serviços sejam disponibilizados de forma acessível e interativa a usuários em todo o mundo. Ela envolve principalmente HTML, CSS e *JavaScript* no *front-end*, e linguagens como *PHP*, *Python*, *Ruby* ou *Node.js* no *back-end*.

É importante destacar que a programação web não surgiu de uma única linguagem, mas de uma necessidade de conectar informações e torná-las interativas na internet. Diferente de *softwares* tradicionais, ela depende de “protocolos de rede, navegadores e servidores para funcionar corretamente.

Outro ponto essencial é que a programação web evoluiu em etapas, acompanhando o crescimento da internet e a complexidade das aplicações, passando de páginas estáticas simples para aplicações web completas, interativas e responsivas.

1.3.1 HTML

Hypertext (Hipertexto): é definido como “o sistema que possibilita a criação de links entre informações”. Na web, esses links são “os que permitem clicar em um elemento e ser direcionado para outra página”. É justamente essa conexão que forma “a teia do *world wide web*”. (REFATORANDO, 2025)

Markup Language (Linguagem de Marcação): corresponde a “um tipo de linguagem usada para organizar e anotar documentos de forma que o conteúdo principal se diferencie da marcação”. No caso do HTML, essas marcações são feitas por meio de “*tags*, que estruturam o conteúdo e indicam como ele deve ser interpretado pelo navegador”. (_____, 2025)

Em resumo, o *HTML* é “a linguagem utilizada para estruturar páginas e criar conexões entre informações na *internet*”. Ele é considerado “um dos três pilares do desenvolvimento web, ao lado do CSS (*Cascading Style Sheets*) e do *JavaScript*”. (_____, 2025)

É importante destacar que o *HTML* “não é uma linguagem de programação”. Diferente de outras linguagens que “executam cálculos ou manipulam dados”, o *HTML* serve apenas para “organizar e exibir informações”. Por isso, ele “não possui recursos para realizar operações complexas ou algoritmos”. (_____, 2025)

Outro ponto essencial é que o *HTML* dificilmente aparece sozinho. Na maioria dos casos, ele é usado em conjunto com o CSS, responsável por “definir a aparência e o estilo das páginas”, e com o *JavaScript*, que permite “adicionar interatividade”. Assim, “*HTML*, *CSS* e *JavaScript* formam a base da maioria dos sites modernos”. (_____, 2025)

O *HTML* foi criado em 1990 por Tim Berners-Lee, físico que trabalhava no CERN (Organização Europeia para a Pesquisa Nuclear), na Suíça. Seu objetivo era desenvolver uma forma simples e eficiente de compartilhar documentos com outros pesquisadores (W3C, 2014).

A primeira versão oficial surgiu em 1995, com o *HTML 2.0*. Essa versão já trazia elementos básicos que continuam em uso até hoje, como títulos (*headings*), *links*, listas e formulários (MDN WEB DOCS, 2025).

Dois anos depois, em 1997, foi lançado o *HTML 3.2*, que introduziu recursos como tabelas e *applets*, permitindo maior interatividade e a inserção de conteúdos multimídia (_____, 2025).

Em 1999, chegou o *HTML 4.01*, uma versão mais robusta que trouxe suporte ao CSS e a scripts, além de melhorias em acessibilidade e internacionalização (W3C, 1999).

A mudança mais significativa ocorreu em 2014, com o lançamento do *HTML5*. Essa versão revolucionou o desenvolvimento web ao introduzir novos elementos semânticos, como `<header>`, `<footer>`, `<article>`, `<section>`, `<nav>`, `<figure>` e `<figcaption>`, além de recursos para mídias, como `<video>` e `<audio>` (W3C, 2014).

Além das *tags*, o HTML5 trouxe também diversas *APIs JavaScript*, como a Geolocalização, o Armazenamento *Web*, o Histórico de Navegação e o Arrastar e Soltar (*Drag and Drop*). Essas funcionalidades abriram caminho para que aplicações web se tornassem mais ricas, interativas e próximas das aplicações de desktop em termos de desempenho (_____, 2025).

Um ponto importante é que o HTML foi evoluindo de forma cumulativa: cada nova versão acrescentou recursos sem eliminar os já existentes. Isso garante compatibilidade retroativa, permitindo que páginas desenvolvidas em versões anteriores continuem funcionando e que o conhecimento adquirido sobre HTML permaneça útil com o tempo (_____, 2025).

Versões:

HTML+ (1993): proposta de evolução feita por Dave Raggett, mas nunca implementada oficialmente (HOMEHOST, 2024).

HTML 2.0 (1995): primeira especificação oficial, apresentada no *World Wide Web Conference*. Tinha como objetivo formalizar os recursos já utilizados de forma prática e não padronizada (_____, 2024).

HTML 3.2 (1997): lançado pelo W3C, corrigiu problemas de compatibilidade da versão anterior e adicionou tabelas, *applets* e suporte a texto ao redor de imagens (_____, 2024).

HTML 4.01 (1999): trouxe três variantes de implementação (*Strict*, *Transitional* e *Frameset*), além de suporte a multimídia e folhas de estilo, sendo amplamente utilizado por muitos anos (_____, 2024).

XHTML 1.0 (2000): reescrita do HTML em XML, com regras mais rigorosas e código mais claro para máquinas, mas com pouca adoção prática (_____, 2024).

Fundação do WHATWG (2004): criado por *Mozilla*, *Apple* e *Opera*, iniciou os trabalhos para a nova geração do HTML (_____, 2024).

HTML5 (2014): trouxe suporte nativo a áudio e vídeo, além de novas *tags* semânticas como `<header>`, `<footer>`, `<section>`, `<article>` e `<nav>`. Como explica a *HomeHost* (2023, não paginado): “O HTML5 fornece o suporte de áudio e vídeo em

alto nível que, até então, não existia nas versões anteriores sem o uso de plug-ins. Essa nova versão também nasceu com uma proposta marcante: ela não é uma versão final, ou seja, seguirá recebendo mudanças ao longo do tempo.” (_____, 2024).

1.3.2 JavaScript

O *JavaScript* foi criado em 1995 por Brendan Eich, enquanto trabalhava na *Netscape Communications*. Originalmente chamado de Mocha, depois renomeado para *LiveScript*, o nome final foi escolhido para se associar ao sucesso do Java (FLANAGAN, 2020).

Com o crescimento da *web*, o *JavaScript* tornou-se essencial para páginas dinâmicas. Em 1997, a linguagem foi padronizada pela *ECMA International*, resultando no padrão *ECMAScript*. Desde então, passou por várias melhorias para atender às demandas modernas da programação (ZAKAS, 2012).

Versões:

ECMAScript 1 (ES1): Lançado em 1997, o ES1 foi a primeira versão padronizada do *JavaScript* pela *Ecma International*. Ela estabeleceu os fundamentos da linguagem, incluindo a sintaxe básica, tipos de dados, operadores e estruturas de controle. (LOCAL WEB, 2024)

ES2: Surgido em 1998, trouxe algumas melhorias à linguagem, como tratamento de exceções, definição de tipos de dados e outras alterações de menor importância. (_____, 2024).

ES3: Introduziu muitos recursos importantes em 1999, como suporte para expressões regulares, novos métodos de *Array* e objetos nativos. (_____, 2024).

ES5: Lançado em 2009, o ES5 trouxe melhorias importantes ao *JavaScript*, como novos métodos para *arrays* e métodos para objetos. Padronizou o *sandboxing* e reforçou a segurança com propriedades não configuráveis, não enumeráveis. (_____, 2024).

ES6: foi lançado em 2015 e marcou o *JavaScript* com classes, *arrow functions* e *let/const*, simplificando a escrita e organização do código. Também trouxe CORS, permitindo comunicação segura entre diferentes origens em aplicações *web*. (_____, 2024).

1.3.3 CSS

CSS (Folhas de Estilo em Cascata – *Cascading Style Sheets*) é uma linguagem de estilos usada para definir a aparência de documentos *web*, como os criados com HTML. Ela permite controlar cores, fontes, layouts e espaçamentos, separando a apresentação visual do conteúdo, garantindo que as páginas sejam exibidas de forma consistente e atraente em diferentes dispositivos e navegadores.

O CSS foi criado por Håkon Wium Lie em 1994, durante seu trabalho no *CERN*, a mesma instituição onde Tim Berners-Lee desenvolveu a *web*. (WC5, 2025).

Lie propôs o CSS como uma maneira de separar o conteúdo da apresentação, permitindo que cores, fontes, layouts e outros estilos fossem gerenciados de forma independente do HTML, tornando a manutenção mais fácil e garantindo consistência visual nas páginas.

Versões:

CSS1: Lançado em 1996, o CSS1 trouxe o suporte básico a cores, fontes, alinhamento de texto, espaçamento e bordas, permitindo a estilização inicial de páginas *web*. (WC5, 2025).

CSS2: Lançado em 1998, o CSS2 introduziu posicionamento absoluto e relativo, camadas (*z-index*) e *media types*, além de oferecer suporte a impressão e maior controle de *layout*. (_____, 2025).

CSS2.1: Lançado em 2011, o CSS2.1 revisou o CSS2, corrigindo erros e inconsistências, tornando-se a versão de referência para navegadores antes do CSS3. (_____, 2025).

CSS3: Desenvolvido a partir de 2001, o CSS3 é dividido em módulos, introduzindo bordas arredondadas (*border-radius*), sombras, gradientes, transições,

animações, *flexbox*, *grid layout* e recursos para layouts responsivos e efeitos avançados sem *JavaScript*. (_____, 2025).

1.4 SERVIDORES

Servidor desempenha um papel fundamental na computação, atuando como uma coluna da *internet* e tecnologia da informação. Um componente essencial que atua em todas as redes *online*, permitindo uma comunicação global. “Sem servidores não haveria *e-mails*, *websites*, armazenamento em nuvem ou mesmo a capacidade de fazer uma simples busca *online*. (LUIS MIRANDA, 2024)

Um servidor pode ser explicado como um *hardware* ou *software* feito para fornecer serviços, dados, recursos ou programas feitos para outros computadores, como clientes em uma única rede. Essas afirmações definem o servidor: um grande facilitador de comunicação e um acesso em uma rede de computadores. (_____, 2024)

Um dos primeiros registros que se tem é da “*Advanced Research Projects Agency Network*”, a ARPANET, ou em português “Agência para Projetos de Pesquisa Avançadas”, uma rede de computadores construída em 1969, um servidor que antecedeu e ajudou a dar início à *internet*, que era utilizado para a transmissão de dados militares sigilosos e interligação de departamentos de pesquisas nos Estados Unidos da América. (A2WORKS, 2025).

Modelos:

Modelo 2010 a 2012: linha M1 da IBM, linha 900 / 950 da DELL e G5 da marca HP, suportavam dois processadores de 2 a 4 núcleos de processamento em média 32 a 64 GB de memória RAM (MSSERVER, 2022).

Modelo 2012 a 2015: linhas M2 e M3 da IBM, linha 10 da DELL e G6 e G7 para os da HP, suportam 2 processadores de 4 a 6 núcleos de processamento e em média 384GB de memória RAM (_____, 2022).

Modelo 2015 a 2018: linhas M4 da IBM, linha 20 DELL e G8 da HP, suportam 2 processadores de até 12 núcleos físicos, 768GB de memória RAM. (_____, 2022).

Modelo 2018 ao mais atuais: linha M5 da IBM, linha 30 e 40 da DELL, G9 e G10 da HP, esses suportam até 2 processadores de até 56 núcleos físicos, 3TB de memória RAM. (_____, 2022).

1.4.1 Servidor local

O servidor local é um equipamento físico instalado dentro de uma empresa, que se responsabiliza por armazenar e gerenciar dados e manter aplicativos funcionando. Uma das vantagens é que a conexão com a *internet* não é necessária para seu funcionamento, além de seu controle absoluto dos dados, já que permanece fisicamente dentro d *HTTP* a organização. Porém sua aquisição e manutenção tem custos altos, além do fator de precisar ser feito por uma equipe especializada.

1.4.2 Servidor *Web*

Um servidor *web* é um sistema computacional composto por *hardware* e *software* que armazena, processa e entrega conteúdo da *web* aos usuários por meio de navegadores. Ele responde a solicitações feitas via protocolo *Hypertext Transfer Protocol* (Protocolo de Transferência de Hipertexto-*HTTP*), fornecendo arquivos como páginas *Hypertext Markup Language* (Linguagem de marcação de hipertexto-HTML), imagens e scripts. Esse processo é fundamental para o funcionamento da internet, permitindo que os usuários acessem *sites* e aplicações *online*.

Além disso, os servidores web podem processar diferentes linguagens de programação, como PHP, Python e JavaScript, possibilitando a criação e entrega de conteúdo dinâmico aos usuários. Esses servidores também incorporam mecanismos de segurança, como certificados SSL, que asseguram a proteção dos dados e a integridade das conexões. Grandes organizações utilizam redes de servidores distribuídos, conhecidas como **CDNs (Content Delivery Networks)**, para otimizar o desempenho e garantir alta disponibilidade dos serviços. Dessa forma, o servidor web representa um componente essencial da infraestrutura tecnológica que sustenta a comunicação e o acesso à informação na era digital.

1.5 BANCO DE DADOS

Um banco de dados é um sistema de armazenamento de informações que permite a coleta, o armazenamento, a recuperação e a manipulação de dados de maneira estruturada e eficiente trata-se de um sistema que armazena dados de forma organizada em um sistema do computador. Com isso permite que o sistema funcione de maneira eficiente sem falhas ou perda de informações importantes. (ALURA, 2024).

O banco de dados foi evoluindo ao longo dos anos iniciando em 1960, em que o Departamento de Defesa dos Estados Unidos, tinha como objetivo fazer uma maneira de armazenar dados criando o evento CODASYL (*Conference on Data Systems Languages* em português Comitê de Linguagens de Sistemas de Dados). O CODASYL foi um grande encontro, que reuniu militares, empresas e universidades. (MEDIUM, 2022).

A IBM teve a criação de Dados Hierárquicos, os dados eram ligados a partir de uma hierarquia, ou seja, possuía um único ligamento, após isso a IBM teve um modelo em rede, onde os dados poderiam se relacionar mais de uma vez. (_____, 2022).

Em 1970 a IBM cria o Modelo Relacional onde começa organizar os dados em forma de tabelas, facilitando a manipulação e a consulta das informações. (_____, 2022).

Em 1980 com base no modelo anterior chega os sistemas comerciais, colocando como principal o SQL (*Structured Query Language* em português Linguagem de Consulta Estruturada). (_____, 2022).

Em 1990 chega o Banco de dados Orientado a Objetos onde manteve a base, mas com os avanços da programação permitiu armazenar objetos mais complexos como imagens e vídeos. (_____, 2022).

Em 2000 o banco de dados evolui exponencialmente com a necessidade de muitos dados serem guardados ao mesmo tempo com isso surge o NoSQL e o banco de dados em Nuvem que possibilita ter um banco de dados online com a *internet*. (_____, 2022).

1.5.1 Modelagem

A Modelagem de dados é o processo de criar uma representação visual, ou esquema, que define os sistemas de coleta e gerenciamento de informações de qualquer organização esse processo facilita para a criação do banco de dados, permitindo análise necessária para criar os modelos conceitual, logico e físico organizando para os dados serem armazenados, manipulados e utilizados de maneira eficiente. (AWS, 2024).

1.5.1.1 Modelo conceitual

A modelagem conceitual é a descrição da informação que o sistema irá gerenciar, sendo um artefato do domínio do problema e não do domínio da solução ou seja, trata-se da descrição, identificando os pontos principais para o banco como as entidades e os relacionamentos. (DEVMEDIA, 2008).

1.5.1.1.1 Entidade

Uma entidade é um objeto único no mundo real. Na verdade, ela pode ser tanto um objeto, uma coisa, quanto uma pessoa, exemplo um cliente para se cadastrar, esses dados ficaram salvo na entidade do cliente. (MEDIUM, 2023).

1.5.1.1.2 Relacionamento

Os Relacionamentos são as ligações entre as entidades indicando como vão interagir por exemplo “o cliente envia uma mensagem para outra pessoa” já existindo os dois clientes para interagir precisam de um relacionamento para enviar a mensagem.

1.5.1.1.3 Atributo

De acordo com os atributos são as características das entidades. Os atributos podem ser multivalorados (como no caso um cliente com dois telefones), ser simples para guardar um único valor ou chave primaria PK (*Primary Key*). (MEDIUM, 2023).

1.5.1.2 Modelo Logico

No Modelo Logico a modelagem é mais detalhada, as tabelas são criadas e as colunas, tipos de dados, chaves primárias e estrangeiras e restrições são definidas ou seja, esse modelo complementa as tabelas, um exemplo colocar se os atributos vão ser apenas números (*Integer*), se vão ser textos (*Varchar* (quantidade de caracteres)), datas (*Date*) números quebrados (*Float*). (DIO, 2023).

1.5.1.3 Modelo Físico

O modelo físico vai converter a modelagem logica na forma do Sistema de Gerenciamento de Banco de Dados (SGBD) deixando de ser um modelo visual para os comandos geralmente na linguagem SQL, permitindo, dessa forma, poder trabalhar com os dados.

1.6 BR MODELO

O BrModelo é uma ferramenta de código aberto e gratuita, lançada em 2005, com o objetivo de auxiliar na modelagem de bancos de dados relacionais. Foi criado como parte de um Trabalho de Conclusão de Curso (TCC) de especialização na área de bancos de dados. A proposta da ferramenta é permitir a criação dos modelos conceitual, lógico e físico de forma visual e intuitiva. (CÂNDIDO,2020).

Versões:

A segunda versão do BrModelo foi desenvolvida em 2006, trazendo melhorias e suporte à modelagem lógica. Com o aumento da popularidade da ferramenta, seu criador, Carlos Henrique Cândido, passou a receber diversos *e-mails* de usuários

relatando experiências, sugerindo melhorias e agradecendo a utilidade da aplicação. Isso o motivou a liberar o código-fonte da ferramenta. (_____, 2020).

A terceira versão foi lançada em 2017, com melhorias na interface e foco em facilitar o aprendizado da modelagem de dados, sendo útil tanto para iniciantes quanto para usuários mais experientes. (_____, 2020).

1.6.1 BrModelo Web

Além das versões para *desktop*, também foi lançada a versão *BrModelo Web*, que funciona diretamente no navegador, sem necessidade de instalação. Ela mantém as funcionalidades principais da versão tradicional e é uma alternativa prática para quem deseja modelar sem depender de sistema operacional específico.

1.7 SQL

SQL foi criado no começo dos anos 70 por pesquisadores da IBM, Raymond Boyce e Donald Chamberlin. (LEARN SQL, 2020). Era chamado no começo de SEQUEL (*Structured English Query Language* traduzindo do inglês Linguagem de consulta estruturada em inglês) e foi criado para manipular e recuperar dados do sistema de gerenciamento de banco de dados da IBM, *System R*. (LEARN SQL, 2020). No final dos anos 70, o *Relational Software, Inc.* (agora se chama Oracle) foi inspirado a criar sua própria versão e introduziu a primeira implementação comercialmente disponível do SQL. (LEARN SQL, 2020). Em 1986, o *American National Standards Institute* (Instituto Nacional Americano de Padrões-ANSI) e o *International Organization for Standardization* (Organização Internacional de Padronização-ISO) adotaram uma definição padrão de SQL. Agora que já tem uma ideia sobre o que é SQL. (LEARN SQL, 2020).

Structured Query Language (Linguagem de Consulta Estruturada-SQL) é a linguagem padrão para gerenciamento de bancos de dados relacionais. Desde sua criação nos anos 1970, o SQL passou por diversas revisões e padronizações. A seguir, apresento uma visão cronológica das principais versões e padrões do SQL. (_____, 2020).

1970: Edgar F. Codd, pesquisador da IBM, propôs o modelo relacional para bancos de dados, estabelecendo as bases teóricas para o SQL. (_____, 2020).

1974: Donald D. Chamberlin e Raymond F. Boyce, também da IBM, desenvolveram a primeira versão do SQL, inicialmente chamada de SEQUEL (*Structured English Query Language* — Linguagem Estruturada de Consulta) e para interagir com o sistema de banco de dados relacional System R da IBM. (_____, 2020).

1986: O *American National Standards Institute* — Instituto Nacional Americano de Padrões. (_____, 2020).

1989: Foi lançada a primeira revisão do padrão, chamada SQL-89, que introduziu pequenas melhorias e correções. (_____, 2020).

1992: O padrão SQL-92, também conhecido como SQL2, trouxe melhorias significativas, incluindo integridade referencial, junções externas e subconsultas mais complexas. (_____, 2020).

1999: O SQL:1999 (SQL3) introduziu recursos avançados, como consultas recursivas, triggers, suporte a tipos de dados estruturados e algumas características orientadas a objetos. (_____, 2020).

2003: O SQL:2003 adicionou funcionalidades relacionadas ao XML (SQL/XML), funções de janela e colunas com valores autogerados. (_____, 2020).

2006: O SQL:2006 expandiu o suporte ao XML, permitindo a importação, armazenamento e manipulação de dados XML dentro do banco de dados. (_____, 2020).

2008: O SQL:2008 introduziu melhorias como *triggers* INSTEAD OF, a instrução TRUNCATE e a cláusula FETCH. (_____, 2020).

2010: Aperfeiçoamentos e Novos Tipos de Dados (_____, 2020).

2011: O SQL:2011 adicionou suporte a dados temporais, permitindo a definição de períodos de validade para os dados. (_____, 2020).

2016: O SQL:2016 incorporou correspondência de padrões em linhas (*row pattern matching*), funções de tabela polimórficas e operações em dados JSON armazenados como *strings* de caracteres. (_____, 2020).

2019: O SQL:2019 introduziu suporte a *arrays* multidimensionais (MDarray) e operadores associados. (_____, 2020).

2023: O SQL:2023 adicionou o tipo de dado JSON nativo e suporte a consultas de grafos de propriedades (*Property Graph Queries* — Consultas de Grafos de Propriedades). (_____, 2020).

1.8 XAMPP

O XAMPP é um ambiente de desenvolvimento gratuito criado em 2002 pela equipe da Apache Friends. Ele permite simular um servidor local no computador, unindo ferramentas como Apache (servidor *web*), MySQL/MariaDB (banco de dados), PHP (linguagem de programação) e Perl (linguagem de *script*). É um *software* de código aberto, distribuído sob os termos da licença GNU GPL, podendo ser copiado e modificado livremente. (APACHEFRIENDS, 2025).

Na prática, o XAMPP é muito utilizado para testar sites e sistemas localmente, sem a necessidade de publicar em um servidor real. É bastante usado por programadores durante o desenvolvimento e também no meio acadêmico, facilitando o aprendizado sobre servidores e banco de dados. (APACHEFRIENDS, 2025).

Versões:

1.0 (2002): A primeira versão foi lançada por *Kai Oswald Seidler* e *Kay Vogelgesang*. Seu objetivo era oferecer uma forma simples de configurar um ambiente de servidor local, funcional em sistemas como *Windows*, *Linux* e *macOS*. (APACHEFRIENDS, 2025).

1.7.7 (2011): Trouxe atualizações como *Apache 2.2.11*, *MySQL 5.1.33* e *PHP 5.2.9*, além de ferramentas de administração como *phpMyAdmin* e o servidor *FTP FileZilla*. (_____, 2025).

1.8.3 (2013): Marcou o início da colaboração com a *BitNami*, e passou a não oferecer mais suporte ao *Windows XP*, refletindo uma modernização da plataforma. (____2025).

5.6.40 (2014): Contou com *PHP 5.6.3*, *MySQL 5.6.21* e melhorias de segurança, como a atualização do *OpenSSL*. (____2025).

7.1.15 (2018): Incluiu suporte ao *PHP 7.1.15*, *MariaDB 10.1.31* e *phpMyAdmin 4.7.9*, atualizando os principais recursos usados por desenvolvedores. (____2025).

8.0.28 (2023): Traz *PHP 8.0.28* e melhorias de desempenho e compatibilidade com tecnologias mais recentes. (____2025).

8.2.12 (2023): Versão mais recente, com *PHP 8.2.12*, *Apache 2.4.58* e *MariaDB 10.4.32*, mantendo o *XAMPP* atualizado com as versões mais recentes dos componentes. (____2025).

1.9 MARIADB

O banco de dados *MariaDB* foi lançado em outubro de 2009, *MariaDB* é um tipo de banco de dados criado pelos mesmos desenvolvedores do *MySQL*. Sendo baseado no *MySQL*, ela pode substituí-lo sem maiores complicações nem grandes esforços de configuração, a ferramenta é de código aberto, tendo como principais atributos a rapidez, alta escalabilidade, robustez de suas ferramentas, variedade de *plugins* e muita capacidade de armazenamento, desde sua criação, o *MariaDB* passou por diversas versões, cada uma trazendo novos recursos, melhorias e correções de *bugs*. (MARIADB, 2025).

Na maioria dos aspectos o *MariaDB* vai funcionar exatamente como o *MySQL*: todos os comandos, interfaces, bibliotecas e *APIs* que existem no *MySQL* também existem no *MariaDB*. Não há nenhuma necessidade de converter um banco de dados para migrar para o *MariaDB*. *MariaDB* consegue substituir o *MySQL*. Além disso, o *MariaDB* novas funcionalidades agradáveis que podem ser aproveitadas. (MARIADB, 2025).

Modelos:**Série 5.x:**

MariaDB 5.1: Lançado em 2009, foi a primeira versão estável do *MariaDB*, baseada no *MySQL 5.1*, com melhorias em desempenho e novos recursos. (MARIADB, 2025).

MariaDB 5.2: Introduzido em 2010, trouxe funcionalidades adicionais, como o mecanismo de armazenamento *Aria* e melhorias no suporte a subconsultas. (_____, 2025).

MariaDB 5.3: Lançado em 2011, focou em aprimoramentos no otimizador de consultas e adicionou suporte a subconsultas mais complexas e índices *Full-Text* para o mecanismo *InnoDB*. (_____,2025).

MariaDB 5.5: Disponível a partir de 2012, alinou-se com o *MySQL 5.5*, incorporando melhorias de desempenho e suporte a *threads*. (_____, 2025).

Série 10.x:

MariaDB 10.0: Lançado em 2014, introduziu replicação *multi-fonte* e tabelas temporárias globais. (_____, 2025).

MariaDB 10.1: Disponível desde 2015, trouxe criptografia de tabelas e suporte a *Galera Cluster* integrado. (_____, 2025).

MariaDB 10.2: Lançado em 2017, adicionou suporte a *CTEs (Common Table Expressions)* e *JSON*. (_____, 2025).

MariaDB 10.3: Introduzido em 2018, trouxe suporte a procedimentos armazenados e funções do *Oracle*, além de tabelas temporárias definidas pelo usuário. (_____, 2025).

MariaDB 10.4: Disponível desde 2019, focou em melhorias de segurança, como autenticação avançada e suporte a *roles*. (_____, 2025).

MariaDB 10.5: Lançado em 2020, introduziu o mecanismo de armazenamento *ColumnStore* e suporte a consultas analíticas. (_____, 2025).

MariaDB 10.6: Disponível desde 2021, trouxe melhorias no otimizador de consultas e suporte aprimorado a *JSON*. (_____, 2025).

MariaDB 10.7 a 10.11: Lançadas entre 2021 e 2023, essas versões introduziram recursos como índices invisíveis, suporte a temporalidade e aprimoramentos no desempenho. (_____, 2025).

Série 11.x:

MariaDB 11.0 a 11.4: Lançadas entre 2023 e 2024, essas versões focaram em otimizações de desempenho, novos recursos de segurança e suporte a funcionalidades avançadas de SQL. (_____, 2025).

MariaDB 11.7: Lançado em fevereiro de 2025, é a versão estável atual, trazendo aprimoramentos no otimizador de consultas e suporte a novos tipos de dados. (_____, 2025).

MariaDB 11.8: Previsto para ser uma versão de suporte de longo prazo (*LTS*), com lançamento planejado para 2025. (_____, 2025).

MariaDB 12.0: Versão futura em desenvolvimento, com lançamento ainda a ser anunciado. (_____, 2025).

1.10 DESIGN

Design é um processo criativo que busca soluções de problemas e melhorar a experiência do usuário por meio de criação e soluções visuais, participativas e estruturais. É um conceito aplicado em praticamente tudo do cotidiano desde roupas, móveis, embalagens, tecnologia, criação de identidade visual, criação de produtos, criação de interfaces para *web*. Original da língua inglesa, *design* significa a concepção de um produto, o produto desta concepção.

O *design* não abrange somente uma disciplina, mas sim várias como *design* gráfico, *design* de interiores, *design* de experiência de usuários, *design* de embalagens, *design* de joias, *design* de *marketing* entre tantos outros. Todas essas áreas têm por objetivo final equilibrar funcionalidade e estética, o produto não pode ser apenas bonito, nem só funcional, tem que atender às necessidades do público-alvo de maneira equilibrada.

Portanto, não é apenas estético, não se limita somente à beleza, mas é um campo que envolve vários processos, como pesquisas de entendimento do problema,

pois para resolver o problema é preciso entendê-lo; pesquisas de mercado e análise, investigar o público-alvo, as tendências, conhecer gostos e necessidades e criar soluções mais direcionadas; criação de conceitos, protótipos, exploração visual e *feedback* inicial, isso permite testar a interação antes da versão final do produto e obter *feedbacks*, garantindo que está no caminho correto; desenvolvimento e refinamento, com os *feedbacks* é possível revisar o *design*, podendo ser feitos os ajustes necessários; implementação e produção; após a aprovação do *design* e ajustes finais, a finalização e entrega, até chegar no produto final.

1.10.1 Inkscape

Inkscape é um *software* leve, criado em dezembro de 2003, livre para editoração eletrônica de imagens e documentos vetoriais que tem alta capacidade de funcionamento até em computadores de baixa capacidade de armazenamento, com um bom tempo de inicialização no geral. (INKSCAPE, 2025).

Versões:

Versão 0.40 essa versão conta com a conversão de imagens *bitmap* em vetor. Nessa versão existe o recurso de colocar texto "*text on path*", comumente utilizado no *CorelDraw*, *Illustrator* e *Freehand*. (INKSCAPE, 2025).

Versão 0.42 disponibilizada em 2005, conta com inovações interessantes, entre elas recurso aprimorado de edição de textos, uma ferramenta prática que pode trabalhar construção de paletas de cores. Também existe a possibilidade expandida de trabalhar com clonagem de objetos e conseguir efeitos semelhantes a caleidoscópios e o suporte total à implementação de scripts em Python e Perl. (_____,2025).

Versão 0.43 essa versão foi disponibilizada em 2005, incluindo o menu Effects, com o total de 22 efeitos gráficos, uma melhoria e refinamento nas ferramentas de edição de nós, com sensibilidade à pressão para usuários de tablets (ou mesas digitalizadoras) na ferramenta de desenho calígrafo e a inclusão do recurso de conexão de elementos, comumente utilizado em fluxogramas. (_____,2025).

Versão 0.44 essa versão disponibilizado em 2006, teve adicionadas Layers dialog, um tipo de suporte para clipping e masking, com melhora da exportação, e melhora geral na performance do software. Essa versão contém atalhos de tecla personalizáveis, o que possibilita usar atalhos de outros programas diminuindo a curvatura de aprendizado. Ela também disponibiliza histórico de ações, o que permite voltar a um momento específico da edição. (_____,2025).

Versão 0.45.1 essa versão foi otimizada adicionando o "Balde de Tinta" para preencher áreas reconhecidas pela coloração, nível do canal alfa, brilho ou outras variantes, facilitando a coloração de áreas grandes tendo o trabalho reduzido e eficiente. (_____,2025).

CAPITULO 2 – ESTUDO DE CASO

A existência de aplicativos de comunicação no meio escolar é fundamental para a transformação inovadora nesse meio, já que ajudaria no processo de transmitir informação entre professores, secretaria, direção e demais funcionários. O maior problema dessas aplicações é o fato de que a maioria necessita de internet para funcionar corretamente, e a rede é um problema muito comum, principalmente as relações às escolas públicas que enfrentam dificuldades quando se trata de conexão *Wi-Fi*. Com um sistema desse tipo, mas *offline*, contribuiria para que a comunicação fosse rápida, eficaz, prática e de fácil entendimento, o que garantiria que a escola como um todo pudesse ficar “conectada” sem enfrentar problemas com a falta de *internet*. Com isso, o sistema seria muito mais acessível, proporcionando uma melhora no meio comunicativo e sendo inclusivo para os membros da comunidade escolar.

2.1 VIABILIDADE

Viabilidade é a qualidade de algo que pode ser realizado, ou seja, que apresenta fortes probabilidades de ser executado com sucesso por reunir todas as condições necessárias. Em resumo, viabilidade significa analisar como determinado projeto pode alcançar resultados positivos. (CONCEITO, 2011).

2.1.1 VIABILIDADE OPERACIONAL

A viabilidade operacional refere-se à capacidade de um empreendimento funcionar de maneira eficiente e sustentável, garantindo que suas atividades cotidianas possam ser realizadas sem comprometer sua estabilidade financeira. Para microempreendedores individuais (MEIs), essa análise é essencial, pois permite verificar se o modelo de negócio adotado é capaz de gerar lucros e atender às exigências do mercado. (CRECERTO, 2024).

2.1.1.1 - Algoritmo de Comunicação em Rede Local

A aplicação fará uso de cabo de rede como meio físico para possibilitar a comunicação entre dispositivos em uma rede local (LAN). A troca de dados ocorrerá com baixa latência, aproveitando a proximidade dos dispositivos. Essa comunicação local permite alta velocidade e confiabilidade na troca de informações.

2.1.1.2 Tecnologias Utilizadas

HTML, CSS e *JavaScript*: As tecnologias de marcação e estilo (HTML e CSS) em conjunto com a linguagem de programação *JavaScript* foram selecionadas para o desenvolvimento da interface do sistema. Essa combinação possibilita a construção de páginas responsivas, interativas e compatíveis com diferentes navegadores, assegurando uma experiência de usuário intuitiva e acessível. O *JavaScript* também será empregado para validações e parte da lógica de interação no lado do cliente.

PHP: Será utilizado como linguagem de programação no lado do servidor, responsável pelo processamento das requisições, manipulação de dados e integração com o banco de dados. Sua ampla adoção, simplicidade e compatibilidade com servidores *web* tornam o PHP uma escolha viável e consolidada para sistemas de comunicação em rede de pequeno e médio porte.

Cabo de Rede: O cabo de rede constitui o meio físico de interconexão local entre os dispositivos. Essa escolha garante baixa latência, elevada velocidade de transmissão e maior segurança em comparação a meios sem fio, sendo adequado para a proposta de comunicação interna em ambientes escolares.

Banco de Dados SQL: Os dados trocados entre os dispositivos serão armazenados em um banco de dados relacional, utilizando MySQL ou SQLite, a depender da escala da aplicação. Ambos os sistemas oferecem confiabilidade, integridade transacional e suporte consolidado, viabilizando a persistência e o gerenciamento eficiente das informações.

PROGRAMAS UTILIZADOS:

Visual Studio Code: IDE recomendada para o desenvolvimento do sistema com HTML, CSS, *JavaScript* e PHP. Oferece recursos avançados de edição, extensões para integração com bancos de dados, além de ferramentas para depuração, testes e otimização do código.

MySQL/SQLite: Sistemas gerenciadores de banco de dados (SGBDs) empregados para o armazenamento das mensagens e demais dados da aplicação. O MySQL será utilizado em cenários de maior escala, devido à sua robustez e

capacidade de suportar múltiplos usuários simultaneamente, enquanto o SQLite será adotado em versões locais mais leves, pela sua simplicidade de implementação e baixo consumo de recursos.

2.1.2 Viabilidade Técnica

A análise de viabilidade técnica visa verificar se a infraestrutura de rede atual comporta a implantação do sistema proposto, avaliando os recursos tecnológicos disponíveis e a compatibilidade com os requisitos do projeto. Essa etapa é essencial para identificar se o ambiente é capaz de suportar o funcionamento do *software* de forma eficiente e contínua. (NEOENERGIA, 2025).

Aplicação Web Local: A escolha por uma aplicação em formato *web*, executada localmente, fundamenta-se na necessidade de compatibilidade com os computadores e notebooks utilizados pelos profissionais da escola. Essa abordagem possibilita o acesso ao sistema diretamente por meio de navegadores, sem depender de serviços externos, garantindo maior flexibilidade de uso e facilidade de manutenção. As tecnologias HTML, CSS e *JavaScript* serão utilizadas na construção da interface interativa e responsiva, enquanto o PHP atuará no processamento das requisições e integração com o banco de dados.

Canal de Mensagens: Considerando o objetivo de assegurar a comunicação sem dependência da internet, optou-se pelo uso da rede local (LAN) como canal principal de transmissão. Essa escolha proporciona maior estabilidade na troca de mensagens e reduz o risco de falhas ocasionadas por instabilidades da conexão externa. Ademais, a utilização da LAN contribui para a segurança da aplicação, evitando a exposição a ameaças virtuais externas e reforçando a confiabilidade do sistema em ambientes escolares.

2.1.2.1 Back-End para Lógica e Envio de Mensagens:

Servidor Local: Para que a lógica de envio e recebimento de mensagens seja gerenciada corretamente, será necessário um servidor dedicado operando na própria rede da escola. Este servidor atuará como intermediador entre os dispositivos

conectados, garantindo o roteamento correto das mensagens, o controle de autenticação e a manutenção do histórico de conversas.

2.1.2.2 Tecnologias:

HTML, CSS e JavaScript: Tecnologias essenciais para o desenvolvimento da interface, responsáveis respectivamente pela estrutura, estilo e interatividade das páginas, garantindo responsividade e usabilidade em diferentes navegadores.

PHP: Linguagem de programação no lado do servidor (back-end), utilizada para processar requisições, manipular dados e integrar o sistema ao banco de dados, destacando-se pela simplicidade, compatibilidade e confiabilidade em aplicações *web*.

2.1.2.3 Banco de Dados:

SQL (MySQL/PostgreSQL): Sistemas robustos para gerenciamento de grandes volumes de dados e com suporte para escalabilidade e segurança. Recomendados em ambientes onde há grande número de mensagens trocadas ou muitos dispositivos conectados simultaneamente.

SQLite: Ideal para aplicações menores ou versões locais que não exigem um servidor de banco de dados separado. Ele permite armazenar informações de maneira leve e eficiente, mantendo a simplicidade de implementação.

2.1.2.4 Ferramentas de Protocolo:

Sockets TCP/IP: Um dos métodos mais tradicionais e confiáveis de comunicação em redes locais. Permite troca de mensagens em tempo real entre dispositivos conectados, com controle preciso sobre o envio e recebimento dos dados. Fundamental para garantir a agilidade do sistema.

2.1.2.5 Autenticação e Segurança:

O sistema contará com um método de autenticação simplificado, onde cada computador será associado a um setor da escola (como “secretaria - pc03” ou “sala dos professores - pc02”). Isso garante um controle mais direto sobre as mensagens enviadas e recebidas, além de permitir rastreabilidade e organização interna.

2.1.2.6 Recursos do Software:

Mensagens de baixa latência: Envio quase instantâneo devido à comunicação local via cabo de rede.

Notificação sonora: Emissão de um sinal sonoro ao receber mensagens novas, garantindo que o usuário perceba a chegada de comunicações importantes.

Histórico de conversas: Registro completo das mensagens enviadas e recebidas, permitindo consulta posterior e auditoria interna.

Bate-papos individuais: Comunicação direta entre setores específicos.

Mensagens prioritárias: Possibilidade de marcar mensagens urgentes com alerta sonoro especial ou notificações destacadas.

2.1.2.7 Recursos Futuramente Adicionais:

Grupos de mensagens: Criação de canais de comunicação entre múltiplos setores simultaneamente (exemplo: “Todos os professores” ou “Coordenação e Secretaria”).

Relatórios de comunicação: Ferramentas para gerar estatísticas de uso do sistema, identificar os setores mais ativos, horários de pico de uso, entre outros indicadores úteis para a gestão.

2.1.2.8 Conclusão da Viabilidade Técnica:

O sistema proposto é tecnicamente viável, pois utiliza tecnologias já consolidadas no mercado, com alta disponibilidade e boa documentação. A implementação em uma rede local reduz os custos e os riscos associados à conectividade externa. Com o devido planejamento e preparação da infraestrutura de rede, a implantação ocorrerá de forma tranquila e eficaz.

2.1.3 - Viabilidade de Cronograma

A viabilidade de cronograma avalia a probabilidade de um projeto ser concluído dentro do prazo proposto. Esse é um estudo de viabilidade crucial, pois os resultados determinarão se o projeto terá sucesso. Ele identifica os principais entraves da iniciativa que podem afetar o cronograma, incluindo restrições internas e externas, como normas, políticas, orçamentos e tecnologia. (LUCID, 2025).

Tabela 1- Tabela de Cronograma

S.C.O.R.E		PLANILHA DE CRONOGRAMA DO TCC			
TAREFAS	PESSOAS	PORCENTAGEM	INICIO	TERMINO	
GRUPO					
PLANEJAMENTO	VALERIO	75%	07/fev	21/fev	
APRESENTAÇÃO TEMA	TODOS	100%	07/fev	14/fev	
MÉTODOS DE BUSCA	TODOS	100%	14/fev	28/fev	
PROJETO DE BAIXA FIDELIDADE	NATHALI E CAIO	70%	28/fev	07/mar	
PROJETO DE MÍDIA FIDELIDADE	NATHALI E CAIO	0%	07/mar	14/mar	
PROJETO DE ALTA FIDELIDADE	NATHALI E CAIO	0%	14/mar	21/mar	
VIABILIDADE	TODOS	80%	21/fev	07/mar	
ENTREGA CAPÍTULO 1	TODOS	0%	07/fev	11/abr	
ENTREGA CAPÍTULO 2	TODOS	0%	11/abr	27/jun	
ENTREGA CAPÍTULO 3	TODOS	0%	27/jun	26/set	
APRESENTAÇÃO	TODOS	0%	26/jul	21/nov	
PESQUISAS					
ÁREA DE ATUAÇÃO	MARIA	100%	14/fev	28/fev	
5 ANOS	MATHEUS E LUIS	100%	14/fev	28/fev	
PROBLEMAS ENCONTRADOS	PARRA E MARIA	100%	14/fev	28/fev	
SOLUÇÕES PARA OS PROBLEMAS	PARRA E MARIA	100%	14/fev	28/fev	
TECNOLOGIAS E RECURSOS	VALERIO E NATHALI	100%	14/fev	28/fev	
ENTREVISTAS	VALERIO, CAIO E MARIA	0%	28/fev	14/mar	
QUESTIONÁRIOS	LUIS	0%	28/fev	14/mar	
ANÁLISE DE TEXTO	NATHALI	0%	28/fev	14/mar	
VIABILIDADE OPERACIONAL	VALERIO	100%	21/fev	07/mar	
VIABILIDADE TÉCNICA	MATHEUS E CAIO	50%	21/fev	07/mar	
VIABILIDADE DE CRONOGRAMA	NATHALI E MARIA	100%	21/fev	07/mar	
VIABILIDADE ECONÔMICA	PARRA E LUIS	90%	21/fev	07/mar	
BANCO DE DADOS					
PESQUISA	LUIS E MATHEUS	0%	04/jul	18/jul	
MODELO CONCEITUAL	LUIS E MATHEUS	0%	04/jul	18/jul	
EXPLICAÇÃO	LUIS E MATHEUS	0%	04/jul	18/jul	
MODELO LÓGICO	LUIS E MATHEUS	0%	04/jul	18/jul	
EXPLICAÇÃO	LUIS E MATHEUS	0%	04/jul	18/jul	
MODELO FÍSICO	LUIS E MATHEUS	0%	04/jul	18/jul	
PROGRAMAÇÃO					
TELA PRINCIPAL	VALERIO	0%	25/jul	08/ago	
TELA DE LOGIN	VALERIO	0%	25/jul	08/ago	
TELA DE CRIAR CONTA	VALERIO	0%	08/ago	22/ago	
CRIOGRAFIA	VALERIO	0%	08/ago	22/ago	
HISTÓRIAS DE USUÁRIO	VALERIO	0%	22/ago	29/ago	
REVISÃO E AJUSTES	VALERIO	0%	29/ago	05/set	
DESIGN					
SELEÇÃO DAS CORES	NATHALI E CAIO	0%	05/mar	14/mar	
LÓGO	NATHALI E CAIO	0%	14/mar	21/mar	
TELA PRINCIPAL	NATHALI E CAIO	0%	05/set	12/set	
TELA DE LOGIN	NATHALI E CAIO	0%	05/set	12/set	
TELA DE CRIAR CONTA	NATHALI E CAIO	0%	19/set	03/out	

FONTE: AUTORIA PRÓPRIA (2025).

A tabela1 apresenta detalhadamente o cronograma das atividades planejadas para o Trabalho de Conclusão de Curso. Todas as ações a serem realizadas estão distribuídas mostrando o trabalho necessário, a quem está atribuído, porcentagem de conclusão, data de início e término.

Grupo: Apresenta no geral os trabalhos que todos do grupo devem realizar, como o planejamento do TCC, apresentação do tema, métodos de busca, entregas dos projetos.

Pesquisas: Essa área mostra a divisão e as etapas das pesquisas a serem realizadas e as que já foram concluídas, como pesquisas de área de atuação, evolução em 5 anos, problemas e soluções encontrados, tecnologias e recursos. Ainda em andamento estão entrevistas, questionários, análise de texto, viabilidade operacional, técnica, de cronograma e econômica.

Banco de dados: Aponta ações para a construção do banco de dados e sua distribuição aos integrantes: pesquisa, modelagem conceitual, lógica e física.

Programação: Mostra ações futuras para programação: tela principal, *login* (*identificação do usuário*), criação de conta, *criptografia*, histórias de usuário, revisão e ajustes.

Design: Trata do *design*: seleção de cores, criação do logo, telas principais e de *login*. Tarefas serão feitas conforme necessidade do projeto.

2.1.4- Viabilidade Econômica

O Estudo de Viabilidade Econômica é uma análise detalhada que visa avaliar a viabilidade financeira de um projeto, empreendimento ou investimento. Ele é fundamental para as empresas, pois fornece informações essenciais para a tomada de decisões estratégicas relacionadas a novos negócios ou expansão. Segundo (VEXPENSES, 2025), esse tipo de estudo é indispensável para prever os custos, calcular os riscos e estabelecer uma base sólida para a viabilidade do projeto.

Esse estudo analisa aspectos financeiros (custos, receitas, fluxo de caixa e retorno do investimento) e fatores externos (mercado, economia e regulamentação) para avaliar a viabilidade de um projeto.

O Estudo de Viabilidade Econômica ajuda gestores a avaliar se um investimento é financeiramente viável e capaz de gerar bons retornos, sendo essencial para planejamento, gestão, redução de riscos e aumento das chances de sucesso.

2.1.4.1 Custo de uma Aplicação:

Um projeto pequeno e com baixa complexidade pode levar de 200 até 1.000 horas de trabalho do seu programador, sendo que esse preço pode variar de R\$ 20 mil até uma média de R\$ 150 mil reais.

2.1.4.2 Manutenção e custos:

As despesas podem variar com a complexidade da aplicação que está sendo feita, este custo fica entre 15% a 20% do valor investido inicialmente, já que o valor mensal pode variar, podendo ser entorno de R\$ 500,00, podendo chegar de perto de R\$ 1000,00

2.1.4.3 Cálculo de viabilidade:

Tabela 2 - Tabela de Cálculo

Custos Fixos	Valores
Energia elétrica	R\$ 1.050/mês
Backup de dados	R\$ 550/mês
Equipe de TI	R\$ 9.000/mês por profissional
Contrato de suporte terceirizado	R\$ 11.000/mês
Licenciamento de software	R\$ 25.500/ano (R\$ 2.125/mês)
Ferramentas de desenvolvimento e testes	R\$ 5.050/ano (R\$ 420/mês)
Atualizações de segurança	R\$ 11.000/ano (R\$ 917/mês)
Custos Variáveis	Valores
Correção de bugs e atualizações	R\$ 27.500/ano (R\$ 2.292/mês)
Compatibilidade com novos sistemas operacionais	R\$ 55.000 (custo pontual a cada atualização)
Otimização de desempenho	R\$ 17.500 (custo pontual por projeto)
Proteção contra acessos não autorizados	R\$ 27.500 (custo pontual)
Treinamento de usuários	R\$ 2.750 por sessão
Descrição	Valores
Servidores locais	R\$ 26.500
Equipamentos de rede (cabos, switches, roteadores)	R\$ 5.250
Software específico para desenvolvimento e implementação	R\$ 27.500

FONTE - AUTONOMIA PROPRIA (2025)

A Tabela 2 apresenta a estimativa de custos associados à manutenção e à implementação do sistema, distribuídos em três categorias principais: custos fixos, custos variáveis e investimentos em infraestrutura.

Na seção de custos fixos, destacam-se despesas recorrentes como consumo de energia elétrica (R\$ 1.050/mês), *backup* de dados (R\$ 550/mês), equipe de TI (R\$ 9.000/mês por profissional), contrato de suporte terceirizado (R\$ 11.000/mês) e licenciamento de *software* (R\$ 25.500/ano ou R\$ 2.125/mês). Também estão incluídos os custos com ferramentas de desenvolvimento e testes (R\$ 5.050/ano) e atualizações de segurança (R\$ 11.000/ano).

Os custos variáveis envolvem atividades técnicas específicas que podem variar conforme a demanda, como correção de *bugs* (R\$ 27.500/ano), compatibilidade com novos sistemas operacionais (R\$ 55.000 por atualização), otimização de desempenho (R\$ 17.500 por projeto), proteção contra acessos não autorizados (R\$ 27.500) e treinamento de usuários (R\$ 2.750 por sessão).

Por fim, a seção de investimentos em infraestrutura inclui os valores relacionados à aquisição de servidores locais (R\$ 26.500), equipamentos de rede como cabos, *switches* e roteadores (R\$ 5.250), além de *software* específico para desenvolvimento e implementação (R\$ 27.500).

Essa análise orçamentária permite compreender a viabilidade financeira do projeto, bem como planejar os recursos necessários para sua sustentação a curto, médio e longo prazo.

2.2 TÉCNICA DE LEVANTAMENTO DE REQUISITOS

As técnicas de levantamento de requisitos são métodos usados para identificar e documentar o que o sistema deve fazer. Elas ajudam a entender as necessidades dos usuários antes do início do desenvolvimento. (SOMMERVILLE, 2011).

Segundo SOMMERVILLE (2011), “o levantamento de requisitos é o processo de descobrir, analisar, documentar e verificar os serviços que o sistema deve

fornecer”. Entre as técnicas mais comuns estão entrevistas, questionários, observação e prototipagem. (_____, 2011).

2.2.1-Entrevista

A entrevista foi realizada com professores e funcionários da ETEC Euro Albino de Souza, aos quais foram feitas perguntas respectivamente. Uma análise foi feita com base nas respostas.

Abaixo de cada pergunta, seguem suas respectivas respostas.

2.2.1.1-Entrevista aos Professores e Funcionários

Resposta do Professor e funcionários da ETEC Euro Albino de Souza

qual sua opinião sobre o método utilizado para mandar mensagens na escola?

Respostas da pergunta 1:

- "Péssimo, perde tempo trabalho perde tempo no deslocamento".
 - "muito ruim, atrapalha a aula ou apresentação, cansativo"
 - "Muito demorado"
 - "O método atual de mandar mensagens na escola às vezes atrapalha as aulas, especialmente quando é algo importante."
 - "Interessante, vai agilizar o processo de comunicação"
 - "Toda parte da comunicação é valido"
 - "Comunicação é delicado, portanto qualquer método que ajude na otimização é benéfico para escola"
 - "Se funcionar, vai ajudar muito a secretária"
- "acha o método utilizado ultrapassado?"

Respostas da pergunta 2:

- "Muito, estamos em uma escola técnica e não usamos direito a tecnologia"
- "Arcaico"
- "Não ultrapassado porque e como e utilizado hoje em dia nas escolas"
- "acredito que o método atual de comunicação na escola é, de fato, ultrapassado."
- "seria a favor de uma melhoria nessa comunicação usando os computadores"
- "Ultrapassado mesmo"
- "É ruim , pois acaba ocupando o tempo da secretária"
- "Cansativo, perda de tempo"
- "Sim"

Respostas da pergunta 3:

- "Completamente a favor"
- "Favor"
- "favor"
- "Eu apoiaria totalmente uma melhoria na comunicação utilizando computadores."

"Sim, sem dúvida"

"Claro"

"Acho importante, seria interessante *mobile* e *desktop*"

"Com certeza"

como o SCORE facilitaria no seu dia a dia?

Respostas da pergunta 4:

- "Não precisaria ir nas salas de aula chamar algum aluno caso precise conversar, pouparia muito tempo não precisando ir até as salas"
- "Não precisa atrapalhar ou interromper a aula, otimiza o tempo, facilita até caso se algum aluno estiver passando mal para poupar tempo"

- "Seria muito mais Rápido e pouparia tempo"
- "seria uma solução boa para o dia a dia, sem atrapalhar as aulas."
- "Agilizando a comunicação mesmo"
- "Reduzindo o tempo"
- "Otimizando a comunicação e melhorando pontos fracos"
- "Vai facilitar muito muito muito, asem ter que atrapalhar o trabalho da secretária tendo que buscar aluno e laboratórios ou sala"

Resposta de pessoas com conhecimento na área de T.I

qual sua opinião sobre o método utilizado para mandar mensagens na escola?

Respostas da pergunta 1:

- "Acredito que o método atual pode ser um pouco eficiente já que pode ocorrer falhas de comunica";
- "na minha visão o método utilizado é eficiente só que as vezes pode ocorrer falhas ou até mesmo esquecer de algum detalhe na hora de passar a mensagem"

"acha o método utilizado ultrapassado?"

Respostas da pergunta 2:

- "um pouco";
- "Olha, acho que o jeito que a escola manda mensagens hoje é sim ultrapassado"

seria a favor de uma melhoria nessa comunicação usando os computadores?

Respostas da pergunta 3:

- "eu seria a favor da melhoraria dessa comunicação usando computadores, isso tornaria o processo mais rápido e também confiável.";

- “Eu super apoiaria a ideia de usar computadores para melhorar isso”

como o SCORE facilitaria no seu dia a dia?

Respostas da pergunta 4:

- “Poderia ser útil principalmente por quererem abordar um sistema *offline* para mandar as mensagens”;

- “seria ótimo porque é *offline* e rápido. Assim, não interfere nas aulas e fica tudo organizado”

Resposta de Alunos

qual sua opinião sobre o método utilizado para mandar mensagens na escola?

Respostas da pergunta 1:

- “Eu acho que o jeito atual de mandar mensagens funciona, mas podia ser melhor”

- “Não acho ruim o método utilizado, só quando estou no meio de uma apresentação e tenho que parar no meio.”

- “Então, o método de agora é bom, mas às vezes demora um pouco ou acabam passando só por cima”

“acha o método utilizado ultrapassado?”

Respostas da pergunta 2:

- “Talvez poderia ser mais prático”

- “Daria para melhorar”

- “Acho que poderia ser mais eficiente”

Respostas da pergunta 3:

- “Sim, acho que combinaria até porque nossa escola é envolvida com tecnologia”

- “totalmente, seria bem interessante”

“Com certeza, seria algo muito bom diferente que não possui em todas as escolas”

-“Seria muito a favor”

como o SCORE facilitaria no seu dia a dia?

Respostas da pergunta 4:

- “Ele ajudaria, porque entregaria as mensagens sem precisar interromper as aulas”

- “Com certeza, seria algo muito bom diferente que não possui em todas as escolas”

-“Deixaria mais organizado e deixaria o método bem mais moderno”

2.2.2 Análise de texto

Professor e funcionários da Etec Euro Albino de Souza

Pergunta 1: Como você avalia o método atual de comunicação utilizado na escola?

Resposta 1: Com base nas respostas fornecidas, é evidente que há um consenso. Em resumo, os entrevistados mencionaram que o método atual de comunicação na escola tem sua funcionalidade, mas ainda enfrenta desafios, como atrasos e interrupções durante as aulas ou atividades. Apesar disso, a maioria acredita que investir em melhorias na comunicação pode trazer grandes benefícios, tornando o processo mais ágil e organizado.

Além disso, muitos destacaram que um sistema como o SCORE seria uma solução prática e eficiente. Por ser *offline* e rápido, ele poderia otimizar o envio de

mensagens sem atrapalhar o andamento das aulas, facilitando a rotina tanto para professores e funcionários quanto para a secretaria. Essa mudança seria bem-vinda para tornar a comunicação mais funcional e harmoniosa no ambiente escolar.

Pergunta 2: Qual sua opinião sobre o estado atual do sistema de comunicação da escola e possíveis melhorias?

Resposta 2: A maioria das pessoas concorda que o método atual de comunicação na escola já está ultrapassado e precisa de melhorias. Alguns destacaram que, sendo uma escola técnica, seria importante aproveitar melhor as tecnologias disponíveis. Outros apontaram que o sistema atual pode ser cansativo, ocupar tempo desnecessário e acabar atrapalhando a rotina da secretaria. Por outro lado, houve quem mencionasse que o método atual ainda é uma prática comum nas escolas hoje em dia. De maneira geral, há um consenso de que melhorias seriam bem-vindas, especialmente com o uso de computadores, que poderiam facilitar e otimizar os processos, trazendo mais eficiência para o dia a dia da escola.

Pergunta 3: Você acredita que a comunicação escolar pode ser otimizada com o uso de computadores? Como?

Resposta 3: Em relação à necessidade de melhorar a comunicação utilizando computadores, a maioria dos entrevistados demonstrou apoio total à ideia, destacando que seria uma iniciativa positiva e interessante, tanto para uso em dispositivos móveis quanto em *desktop*. Essa melhoria é vista como uma forma de otimizar o processo e trazer mais praticidade para o dia a dia.

Pergunta 4: O que você pensa sobre a implementação do sistema SCORE para melhorar a comunicação na escola?

Resposta 4: É possível concluir que o SCORE traria grandes benefícios para o dia a dia na escola, principalmente ao otimizar a comunicação e reduzir o tempo gasto em tarefas como buscar alunos nas salas ou interromper aulas. Além de ser mais rápido e eficiente, o sistema evitaria distrações no ambiente escolar e pouparia o trabalho da secretaria, garantindo uma rotina mais organizada e funcional para todos.

Conclusão: Baseando-se nas respostas, fica claro que o método atual de comunicação na escola possui desafios, como demora, interrupções nas aulas e perda de tempo em tarefas administrativas. Embora ele ainda seja utilizado em muitas escolas, a maioria das pessoas acredita que melhorias seriam bem-vindas. A proposta de implementar um sistema como o SCORE foi muito bem recebida. Os entrevistados destacaram que ele otimizaria o tempo, agilizaria a comunicação e eliminaria a necessidade de interrupções nas aulas ou deslocamentos desnecessários. Apesar de ser mais útil para escolas que lidam com alta demanda de comunicação, é evidente que o SCORE tem grande potencial para transformar a rotina escolar, trazendo mais eficiência e organização ao dia a dia.

Pessoas com conhecimento na área de T.I.

Pergunta 1: Como você avalia o método atual de comunicação da escola?

Resposta 1: Em resumo, os entrevistados mencionaram que o método utilizado para mandar mensagens na escola é considerado eficiente na maioria das vezes. No entanto, algumas falhas podem ocorrer, como problemas de comunicação ou o esquecimento de detalhes importantes. Apesar disso, o sistema atual ainda atende às necessidades básicas da escola, mesmo que exista espaço para possíveis melhorias no futuro.

Pergunta 2: Qual a sua opinião sobre a modernização do sistema de comunicação na escola?

Resposta 2: É notável que as opiniões sobre o método de comunicação utilizado na escola variam. Enquanto alguns acreditam que ele é apenas "um pouco" ultrapassado, outros reconhecem que o sistema poderia ser modernizado. Há um consenso de que a introdução de computadores seria uma melhoria bem-vinda, trazendo mais eficiência e organização para a comunicação no ambiente escolar.

Pergunta 3: Você acredita que a utilização de computadores poderia melhorar a comunicação escolar? Por quê?

Resposta 3: Com base nas respostas fornecidas, fica claro que há um apoio significativo à ideia de melhorar a comunicação utilizando computadores. Os entrevistados destacaram que essa mudança tornaria o processo mais rápido e confiável, o que seria benéfico para o dia a dia escolar. Essa melhoria seria especialmente útil para otimizar o fluxo de informações e garantir uma comunicação mais eficiente e moderna no ambiente escolar.

Pergunta 4: Como você avalia a possibilidade de implementar o sistema SCORE para comunicação escolar?

Resposta 4: É possível concluir que o SCORE seria uma ferramenta útil para a comunicação escolar, especialmente por ser um sistema *offline* e rápido. Ele não interferiria nas aulas e ajudaria a manter tudo mais organizado, o que seria um grande benefício para o dia a dia. Enquanto alguns podem não ver a necessidade imediata, sua implementação traria mais agilidade e eficiência para o ambiente escolar.

Conclusão geral: Com base nas respostas fornecidas, é perceptível que o método atual de comunicação na escola ainda cumpre seu papel básico, mas apresenta falhas ocasionais, como problemas de transmissão de informações e lapsos em detalhes importantes. Apesar disso, a ideia de modernizar a comunicação, utilizando computadores ou o sistema SCORE, foi amplamente apoiada. A maioria reconhece que a implementação de um sistema mais eficiente traria benefícios significativos, como maior rapidez, confiabilidade e organização, sem interferir nas aulas. Assim como em outros contextos, a necessidade de adoção pode variar conforme a demanda de cada escola, sendo mais útil em locais onde a comunicação desempenha um papel essencial no dia a dia.

Alunos

Pergunta 1: Como você avalia o método atual de comunicação na escola?

Resposta 1: Em resumo, os entrevistados mencionaram que o método atual de comunicação na escola é funcional e atende às necessidades em boa parte dos

casos. No entanto, algumas limitações foram apontadas, como atrasos, interrupções em momentos inadequados e falhas na transmissão completa das mensagens. Isso sugere que, embora o sistema funcione, há espaço para melhorias que tornem a comunicação mais eficiente e menos invasiva.

Pergunta 2:Quais melhorias você acredita que poderiam ser feitas no sistema atual?

Resposta 2:É notável que as opiniões sobre o método utilizado na escola indicam que ele não é necessariamente ineficaz, mas há um consenso de que há espaço para melhorias. Alguns sugerem que ele poderia ser mais prático e eficiente, destacando a possibilidade de torná-lo mais funcional para atender às necessidades do dia a dia, assim como métodos mais organizados e otimizados em outros contextos, como os utilizados por *motoboys* para planejar rotas de entrega.

Pergunta 3:Você acha que o uso de computadores para comunicação escolar seria uma melhoria? Por quê?

Resposta 3:Com base nas respostas fornecidas, é evidente que há um apoio significativo à ideia de melhorar a comunicação na escola utilizando computadores. Os entrevistados acreditam que essa abordagem seria altamente benéfica, especialmente em um ambiente já envolvido com tecnologia, como uma escola técnica. Além de tornar o processo mais prático e eficiente, seria algo inovador e diferenciado, capaz de trazer melhorias importantes que não estão disponíveis em todas as escolas.

Pergunta 4:Qual é a sua opinião sobre a utilização do sistema SCORE para melhorar a comunicação escolar?

Resposta 4:É possível concluir que o SCORE seria uma solução prática e eficiente para o dia a dia escolar. Ele entregaria as mensagens de forma organizada, sem interromper as aulas, e modernizaria o método atual de comunicação. Além disso, sua implementação seria um diferencial, trazendo benefícios que ainda não

estão presentes em todas as escolas, contribuindo para uma rotina mais fluida e eficiente.

Conclusão geral: Com base nas respostas fornecidas, é evidente que, embora o método atual de comunicação na escola seja funcional em vários aspectos, ele apresenta limitações que impactam sua eficiência, como atrasos e interrupções em momentos inadequados. Há um consenso de que melhorias são necessárias, principalmente para torná-lo mais prático e adaptado às demandas do dia a dia. A proposta de modernizar a comunicação com o uso de computadores ou implementar o SCORE foi amplamente bem recebida. Tais soluções são vistas como práticas, eficientes e inovadoras, especialmente em uma escola técnica, onde a tecnologia desempenha um papel importante. Além disso, o SCORE se destaca como uma opção que, por ser *offline* e organizado, evitaria interrupções nas aulas, trazendo benefícios significativos para a rotina escolar. Apesar de sua relevância ser maior em escolas com alta demanda de comunicação, essas melhorias mostram grande potencial para otimizar o ambiente escolar de maneira geral.

2.3 PROTOTIPAGEM

A prototipagem é uma etapa essencial no processo de desenvolvimento de sistemas e aplicativos, onde se cria uma representação visual, funcional ou estrutural de um produto ainda em construção. Ela permite simular o comportamento do sistema e verificar sua usabilidade antes da implementação final.

“o protótipo serve como um mecanismo de comunicação entre o desenvolvedor e o usuário, facilitando a identificação de requisitos e expectativas”. Ou seja, é uma versão inicial que ajuda a evitar erros futuros, testando ideias de forma rápida e econômica. (PRESMAN,2016).

2.3.1 Aplicativo *Desktop*

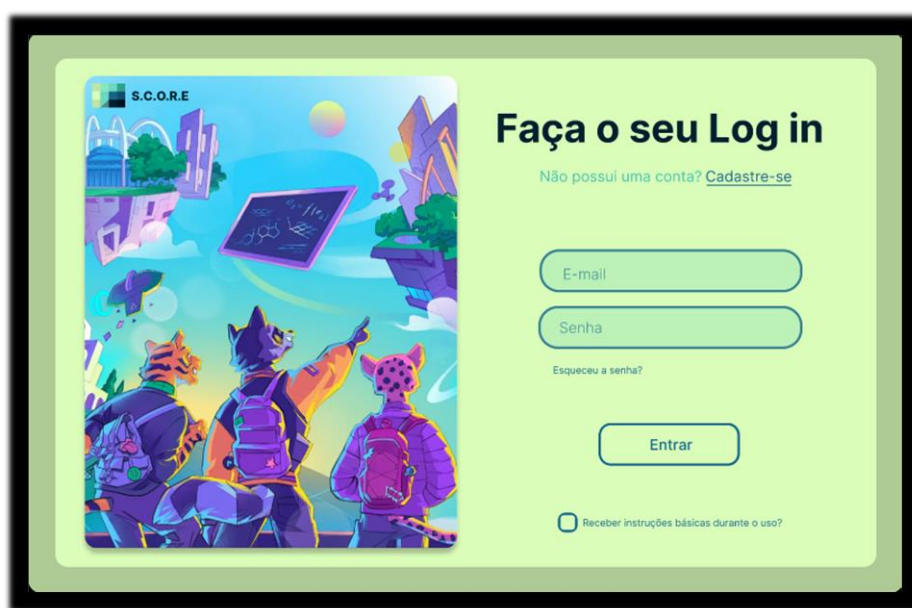
A prototipagem de um aplicativo para *desktop* serve principalmente para validar ideias, fluxos de navegação e interface com o usuário. Ela ajuda a visualizar a

estrutura e o *design* do sistema, antecipando falhas de usabilidade e refinando funcionalidades. Geralmente, esse processo começa logo após o levantamento de requisitos, durante a fase de análise, para transformar as necessidades do usuário em modelos práticos e compreensíveis.

“Prototipagem é uma técnica eficaz para a validação dos requisitos e para a exploração de alternativas de projeto”. Em outras palavras, ela entra em cena quando já se tem uma noção clara do que o sistema deve fazer, mas antes de começar a codar de verdade. (SOMMERVILLE, 2011).

2.3.1.1 tela de login

Figura 1– Tela de *Login*



FONTE – AUTORIA PRÓPRIA (2025)

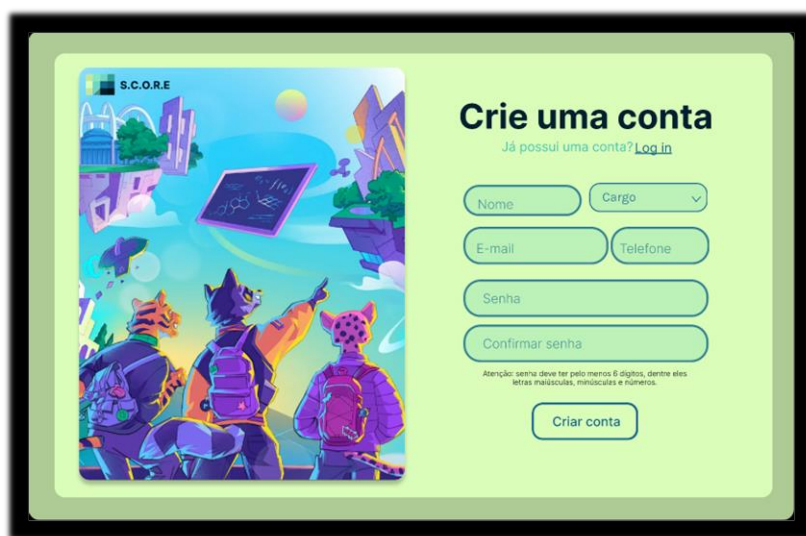
Na Figura 1, mostra-se a tela de *login* de um aplicativo, com um *design* dividido em duas partes: à esquerda, uma ilustração estilizada com estudantes em um cenário futurista e o logotipo do sistema S.C.O.R.E; à direita, o formulário de acesso com fundo verde-claro. O formulário solicita “E-mail” e “Senha” nos campos a serem preenchidos, além de conter o botão “Entrar”, a opção “Cadastrar-se” para novos usuários e um *checkbox* que permite ativar dicas de uso durante a navegação pelo sistema. O *design* combina elementos modernos e lúdicos, com foco na acessibilidade

e atratividade para o público-alvo, que são professores, funcionários e direção de escolas técnicas.

Ao acessar esta tela, o usuário deve inserir suas credenciais para autenticação no sistema. Caso não possua uma conta, pode clicar em “Cadastrar-se” para criar uma. Em caso de esquecimento da senha, o sistema disponibiliza um botão de recuperação. Após o *login* bem-sucedido, o usuário será direcionado para a tela principal do sistema, onde terá acesso às funcionalidades específicas conforme seu perfil (ex: professor, coordenador, secretaria). Essa tela inicial é essencial para garantir a segurança de acesso e separar as permissões de cada usuário dentro da plataforma.

2.3.1.2 Tela de cadastro

Figura 2 – Tela de Cadastro



FONTE – AUTORIA PRÓPRIA (2025)

Na Figura 2, mostra-se a tela de cadastro de usuário do sistema S.C.O.R.E, acessada por meio do botão “Cadastrar-se” na tela de *login*. A interface mantém a identidade visual da tela anterior, com uma ilustração futurista à esquerda e o formulário à direita, mantendo a coerência no *design* e reforçando a proposta gráfica do sistema.

O formulário de cadastro é composto por campos obrigatórios para preenchimento: Nome, Cargo (com seleção via menu suspenso), *E-mail*, Telefone, Senha e Confirmar senha. Abaixo dos campos, há uma mensagem de orientação sobre a criação de senha segura, exigindo pelo menos 6 dígitos, com letras maiúsculas, minúsculas e números. O botão “Criar conta” finaliza o processo.

O objetivo dessa tela é permitir que novos usuários realizem seu cadastro de forma rápida, prática e segura. A escolha dos campos garante a coleta de informações básicas e importantes para a gestão de acessos dentro do sistema. Após preencher e confirmar os dados, o usuário será redirecionado para a tela principal, com acesso liberado conforme o cargo selecionado.

2.3.1.3 Tela de Recuperação de Senha

Figura 3 - Tela de Recuperação de Senha



FONTE – AUTORIA PRÓPRIA (2025)

Na Figura 3, é apresentada a tela de recuperação de senha do sistema S.C.O.R.E. Essa interface é acessada a partir da opção “Esqueceu a senha?” localizada na tela de *login*. O *design* da tela segue o padrão visual do sistema, com fundo verde-claro e *layout* minimalista, visando simplicidade e foco na ação principal.

O único campo visível nesta tela solicita que o usuário digite seu *e-mail* cadastrado, e em seguida clique no botão “Enviar”. Após isso, o sistema envia um código de verificação por *e-mail*, que deve ser usado para redefinir a senha de acesso. A interface também possui um botão de “Voltar”, que retorna o usuário à tela de *login*.

Essa funcionalidade é essencial para garantir a autonomia do usuário em caso de esquecimento de senha, sem necessidade de suporte técnico. O processo é rápido, seguro e automatizado, reforçando a confiabilidade do sistema.

2.3.1.4 Tela verificação de código

Figura 4 – Tela verificação de código

A imagem mostra a tela de verificação de código do sistema S.C.O.R.E. O fundo é verde claro. No canto superior esquerdo, há o logo "S.C.O.R.E" e no canto superior direito, o botão "Voltar". No centro, o texto "Entre com o seu código" é exibido em negrito, seguido por "Enviamos um código para o seu e-mail". Abaixo disso, há quatro caixas de entrada vazias para os dígitos do código. Logo abaixo, há o texto "Não recebeu o e-mail? [Clique aqui para reenviar](#)". No rodapé da seção central, há um botão "Continuar".

FONTE – AUTORIA PRÓPRIA (2025)

A Figura 4 mostra a tela de verificação de código enviada por *e-mail*, parte do processo de recuperação de senha do sistema S.C.O.R.E. Essa tela aparece logo após o usuário inserir o *e-mail* na etapa anterior (Figura 3). O objetivo é garantir que apenas o dono do *e-mail* consiga prosseguir com a redefinição da senha.


O *layout* segue o padrão visual do sistema, com fundo verde-claro, logo no canto superior esquerdo e botão “Voltar” no canto direito. No centro, o usuário deve digitar o código de quatro dígitos recebido por *e-mail*. Há também a opção de reenviar

o código caso não tenha sido recebido. O botão “Continuar” leva à próxima etapa, onde será possível criar uma nova senha.

Essa tela reforça a segurança do sistema, impedindo que pessoas não autorizadas tenham acesso a contas alheias apenas informando o *e-mail*.

2.3.1.5 Tela de Criação de Nova Senha

Figura 5 – Tela de Criação de Nova Senha

A imagem mostra a tela de criação de nova senha do sistema S.C.O.R.E. O fundo é verde claro. No canto superior esquerdo, há o logo S.C.O.R.E. e no canto superior direito, o botão "Voltar". O título central é "Nova senha" em negrito, seguido pelo subtítulo "Digite sua nova senha". Abaixo disso, há dois campos de entrada de texto arredondados: "Senha" e "Confirmar senha". Abaixo dos campos, há uma linha de texto em menor fonte: "Atenção: senha deve ter pelo menos 6 dígitos, dentre eles letras maiúsculas, minúsculas e números." No final, há um botão "Continuar".

FONTE – AUTORIA PRÓPRIA (2025)

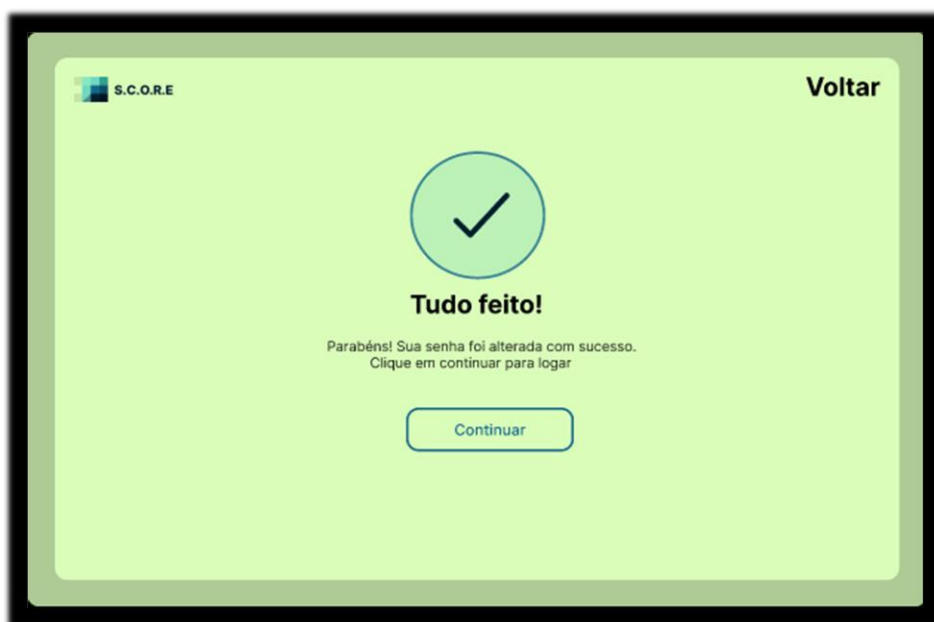
Na Figura 5, a tela de criação de nova senha, última etapa do processo de recuperação de conta no sistema S.C.O.R.E. Essa interface é acessada após a verificação do código enviado por *e-mail* (Figura 4).

O *layout* mantém a identidade visual do sistema, com fundo verde-claro e *design* simples. O usuário deve preencher dois campos: Senha e Confirmar senha, seguindo os critérios de segurança indicados abaixo (mínimo de 6 caracteres com letras maiúsculas, minúsculas e números). Após o preenchimento correto, basta clicar no botão “Continuar” para concluir a redefinição da senha. O botão “Voltar” também está disponível no canto superior direito, permitindo retornar à etapa anterior.

Essa etapa garante que o usuário recupere o acesso de forma segura, sem precisar de ajuda externa, tornando o sistema mais funcional e independente.

2.3.1.6 tela de Confirmação de Alteração de Senha

Figura 6 – Confirmação de Alteração de Senha



FONTE – AUTORIA PRÓPRIA (2025)

A Figura 6 apresenta a tela final do processo de recuperação de senha do sistema S.C.O.R.E. Após o usuário redefinir sua senha com sucesso (Figura 5), ele é direcionado para essa tela de confirmação.

O *design* segue o padrão visual do sistema, com fundo verde-claro e elementos minimalistas. Ao centro, há um ícone de *check* (✓) que indica sucesso na operação, acompanhado da mensagem: “Tudo feito! Parabéns! Sua senha foi alterada com sucesso. *Clique* em continuar para logar.” Abaixo, o botão “Continuar” permite retornar à tela de *login*, encerrando o processo. O botão “Voltar” também está disponível no canto superior direito, mantendo a consistência da navegação.

Essa tela é importante pois fornece um *feedback* claro e positivo ao usuário, garantindo que a operação foi bem-sucedida e orientando os próximos passos de forma simples e objetiva.

2.3.1.7 tela principal

Figura 7 – Tela Principal



FONTE – AUTORIA PRÓPRIA (2025)

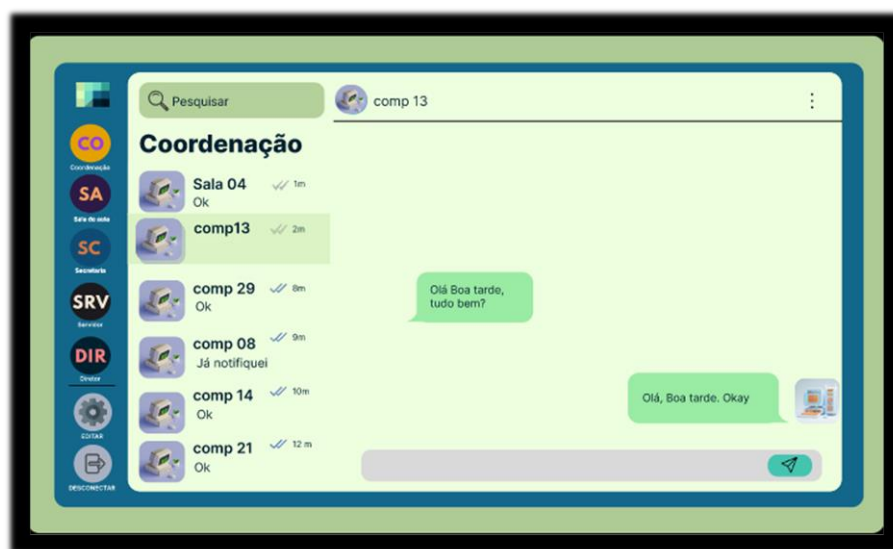
Na Figura 7, apresenta-se a tela principal do sistema S.C.O.R.E, acessada por um computador comum identificado como Comp. 04. Essa tela representa a interface de comunicação entre setores da escola, permitindo que qualquer terminal envie mensagens diretamente para setores específicos, como Coordenação, Sala de Aula, Secretaria, Servidor e Direção, por meio dos botões exibidos na lateral esquerda.

O painel lateral também inclui botões para editar dados e desconectar. No centro da tela, é exibida a lista de mensagens trocadas com os setores, incluindo identificação dos remetentes, pré-visualização da última mensagem e tempo de envio. Ao lado direito, está a área do *chat* ativo com o setor da Coordenação, onde é possível visualizar e enviar mensagens em tempo real.

O campo inferior permite digitar mensagens e enviá-las usando o botão com ícone de avião de papel. Essa interface simples e funcional facilita a comunicação interna escolar de forma prática, mesmo em contextos *offline*, respeitando o fluxo de trabalho e os papéis de cada setor.

2.3.1.8 tela de envio de mensagem

Figura 8 – Tela de envio de mensagem



FONTE – AUTORIA PRÓPRIA (2025)

Na Figura 8, a tela de envio de mensagem do sistema S.C.O.R.E, exibindo uma conversa entre o setor da Coordenação e um terminal identificado como comp13. Essa tela representa a funcionalidade principal do sistema: a comunicação rápida e direta entre setores e computadores da escola.

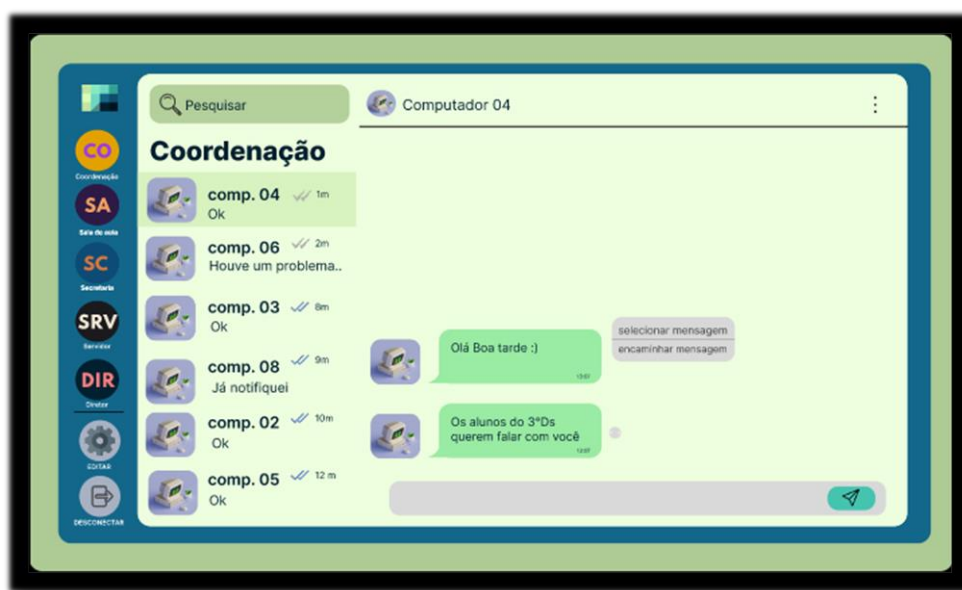
No painel esquerdo, o setor da Coordenação visualiza uma lista de mensagens recebidas de diferentes computadores, como Sala 04, comp29, comp08, entre outros. As mensagens são exibidas com *status* de envio e tempo decorrido desde o recebimento.

No lado direito da tela está a área de conversa ativa, com mensagens trocadas entre a Coordenação e o computador comp13. O *layout* é intuitivo, semelhante a aplicativos de mensagem instantânea, permitindo ao usuário visualizar a conversa em tempo real. O campo de texto na parte inferior permite digitar novas mensagens e enviá-las pelo botão com ícone de avião de papel.

Essa interface mostra claramente como a ferramenta facilita a comunicação descentralizada, sem necessidade de deslocamento físico, e cumpre seu papel de agilizar o fluxo de informação entre setores escolares.

2.3.1.9 Tela encaminhar ou selecionar

Figura 9 – Tela de Seleção e Encaminhamento de Mensagem



FONTE – AUTORIA PRÓPRIA (2025)

A Figura 9 apresenta a funcionalidade de seleção e encaminhamento de mensagens no sistema S.C.O.R.E., durante uma conversa entre o Computador 04 e o setor da Coordenação.

A interface mantém o padrão visual das telas anteriores, com menu lateral, lista de conversas à esquerda e *chat* aberto à direita. Ao interagir com uma mensagem, surgem duas opções flutuantes: “Selecionar mensagem” e “Encaminhar mensagem”.

A primeira permite ao usuário destacar uma ou mais mensagens para ações como copiar, apagar ou visualizar detalhes. Já a segunda possibilita reenviar mensagens a outros setores, facilitando a circulação de informações relevantes.

Essa funcionalidade amplia a flexibilidade do sistema e se alinha a recursos comuns em mensageiros modernos, ao oferecer ações contextuais diretamente sobre o conteúdo.

2.3.1.10 selecionar qual local do envio

Figura 10 – Tela de selecionar qual local de envio



FONTE – AUTORIA PRÓPRIA (2025)

Na Figura 10, é exibida a interface de encaminhamento de mensagens no sistema S.C.O.R.E, após a seleção da opção “Encaminhar mensagem” em uma conversa ativa.

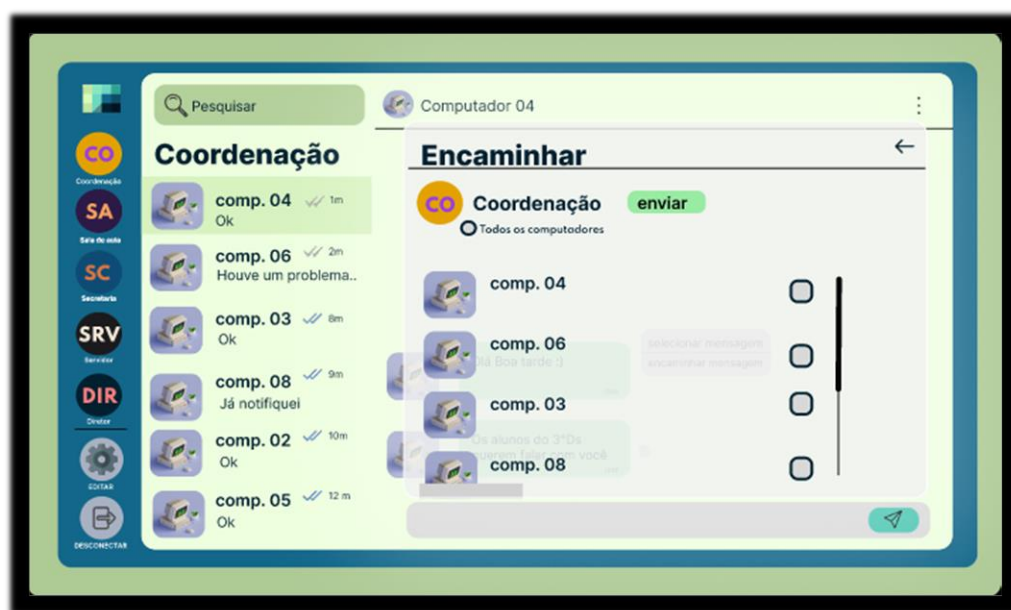
A tela mantém a organização visual tradicional do sistema, com o menu lateral à esquerda, a lista de conversas ao centro e o *chat* aberto à direita. Quando o usuário escolhe encaminhar uma mensagem, é exibida uma sobreposição com o título “Encaminhar”, listando os setores disponíveis para redirecionamento.

Cada setor é apresentado com ícone, nome e a quantidade de computadores conectados naquele grupo — como, por exemplo, “Coordenação” com 6 computadores, “Salas de aula” com 18, e “Diretor” com 1 computador. Essa visualização permite ao usuário decidir rapidamente para qual local deseja reenviar a mensagem, facilitando a comunicação direcionada e eficiente.

Esse recurso amplia o alcance da informação dentro do ambiente institucional e fortalece a integração entre setores, ao possibilitar o compartilhamento ágil de conteúdos relevantes com apenas alguns cliques.

2.3.1.11 Tela de selecionar o computador de envio

Figura 11 – Tela de selecionar o computador de envio



FONTE – AUTORIA PRÓPRIA (2025)

Na Figura 11, observa-se a tela de seleção de computadores específicos para o encaminhamento de mensagens dentro de um setor, no sistema S.C.O.R.E. Essa interface aparece após a escolha do setor de destino (neste caso, “Coordenação”).

A janela de encaminhamento exibe o nome do setor no topo, junto da opção “enviar” e de um botão de seleção global intitulado “Todos os computadores”, que permite encaminhar a mensagem simultaneamente a todos os dispositivos daquele grupo.

Abaixo, é apresentada a lista dos computadores disponíveis no setor, como “comp. 04”, “comp. 06”, “comp. 03” e “comp. 08”, cada um com seu respectivo ícone e caixa de seleção individual. O usuário pode escolher enviar a mensagem para todos os computadores ou selecionar apenas os que forem relevantes para o conteúdo em questão.

Essa funcionalidade oferece maior controle e precisão no direcionamento das informações, evitando comunicações desnecessárias e garantindo que apenas os destinatários apropriados recebam determinada mensagem

2.3.1.12 tela de mensagem encaminhada

Figura 12 – Tela de mensagem encaminhada



FONTE – AUTORIA PRÓPRIA (2025)

Na Figura 12, é apresentada a confirmação visual do envio de uma mensagem no sistema S.C.O.R.E. Após o usuário concluir o processo de encaminhamento, uma janela flutuante central exibe a mensagem “Mensagem enviada com sucesso!” acompanhada do botão “Continuar”.

A tela mantém a estrutura habitual do sistema, com o menu lateral à esquerda, lista de conversas no centro e o *chat* aberto à direita. No histórico da conversa, é possível visualizar a mensagem encaminhada, destacando a fluidez da comunicação.

Esse *feedback* visual imediato garante ao usuário a certeza de que a mensagem foi devidamente reenviada, promovendo confiança e clareza no uso da ferramenta. Além disso, reforça a usabilidade da aplicação ao manter o usuário informado sobre o status de suas ações.

2.3.1.13 tela informações do usuário (Administrador)

Figura 13 – Tela de informações do usuário (Administrador)



FONTE – AUTORIA PRÓPRIA (2025)

Na Figura 13, é apresentada a tela de informações e configurações acessível ao perfil de Administrador no sistema S.C.O.R.E. Ao lado esquerdo, um menu específico surge após a seleção do setor “Diretor”, revelando opções de gerenciamento exclusivas.

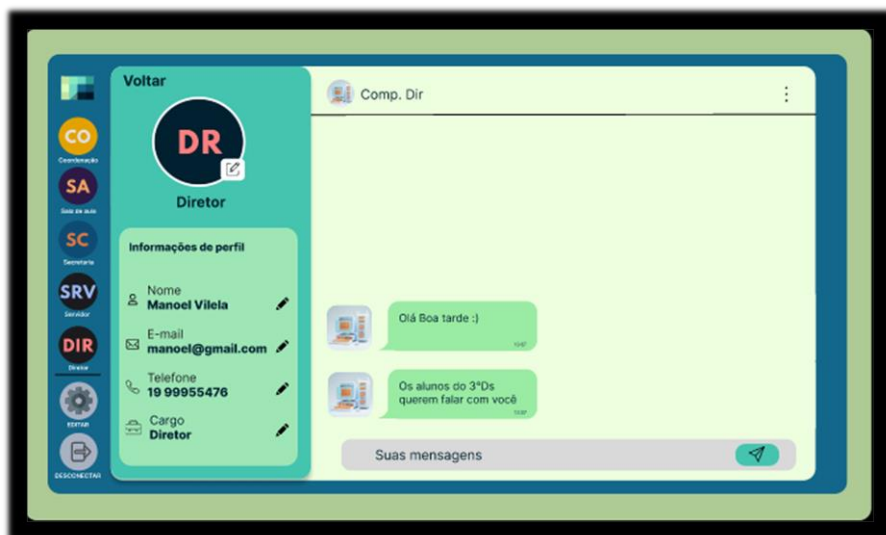
Entre as funcionalidades disponíveis estão: “Perfil”, “Conversas”, “Cadastrar usuário”, “Excluir usuário” e “Notificações”. Esses botões são organizados em uma área distinta com fundo destacado, sinalizando a área administrativa do sistema.

À direita, o campo de mensagens permanece visível, evidenciando que as ações administrativas podem ser realizadas sem perder o contexto da conversa em andamento. A presença das mensagens no *chat* com o “Comp. Dir” reforça a continuidade da comunicação durante o uso das funções administrativas.

Essa tela demonstra o papel central do Administrador, que pode gerenciar usuários e configurações sem sair da interface de comunicação, contribuindo para maior agilidade e integração entre os aspectos técnicos e operacionais do sistema.

2.3.1.14 tela de editar informações do usuário

Figura 14 – Tela de editar informações do usuário



FONTE – AUTORIA PRÓPRIA (2025)

Na Figura 14, é exibida a tela de edição de informações do usuário no sistema S.C.O.R.E, acessível ao perfil do Administrador. A interface apresenta os dados pessoais do usuário selecionado, como nome, e-mail, telefone e cargo, organizados no painel lateral esquerdo.

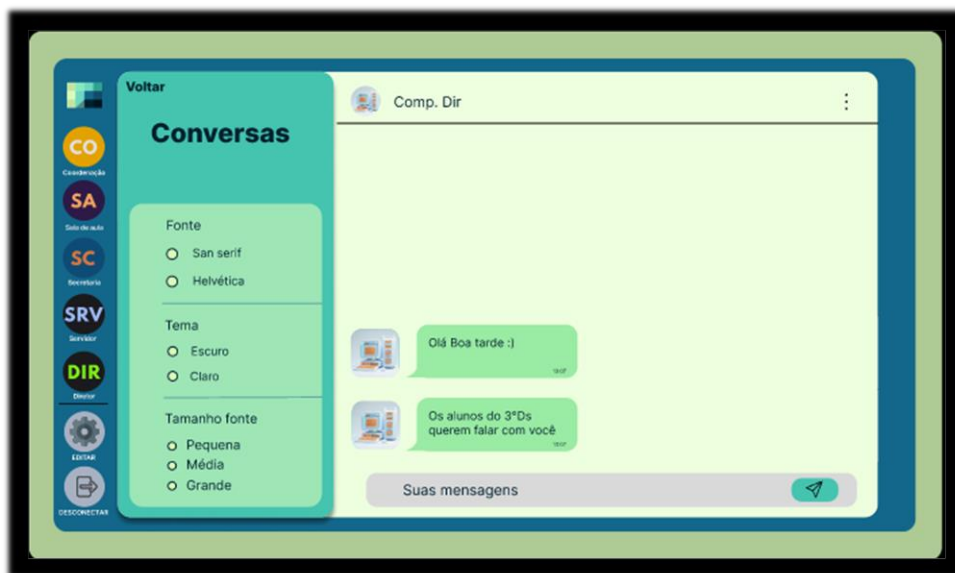
Cada campo informativo possui um ícone de lápis ao lado, indicando que esses dados podem ser editados diretamente. No topo da área de perfil, é exibido o avatar do usuário com a sigla “DR” (de Diretor) e um botão para alterar a imagem de perfil.

Mesmo durante a visualização ou edição das informações cadastrais, a área de conversas permanece ativa à direita, permitindo ao usuário manter o fluxo de comunicação com outros setores, como demonstrado pela conversa com o “Comp. Dir”.

Essa funcionalidade garante praticidade e controle, permitindo atualizações rápidas nos dados dos usuários sem a necessidade de sair do ambiente principal de trabalho, otimizando o gerenciamento de perfis no sistema.

2.3.1.15 tela editar conversas

Figura 15 – Tela de editar conversas



FONTE – AUTORIA PRÓPRIA (2025)

Na Figura 15, é apresentada a interface de personalização da exibição de conversas no sistema S.C.O.R.E. Essa funcionalidade permite que o usuário ajuste preferências visuais para melhorar sua experiência de uso.

No painel lateral esquerdo, são exibidas três categorias de configuração: Fonte, Tema e Tamanho da fonte. O usuário pode escolher entre os estilos de fonte “*San serif*” ou “*Helvética*”, optar entre os temas “*Escuro*” ou “*Claro*”, e selecionar o tamanho da fonte entre “*Pequena*”, “*Média*” ou “*Grande*”.

A área de conversa à direita permanece ativa, com o histórico visível da troca de mensagens entre o usuário e o “*Comp. Dir*”. Isso demonstra que as alterações visuais podem ser feitas em tempo real, sem interromper a comunicação em andamento.

Esse recurso reforça o caráter acessível e adaptável do sistema, ao permitir que diferentes perfis de usuários ajustem a interface de acordo com suas necessidades visuais e preferências de leitura.

2.3.1.16 tela de editar notificações

Figura 16 – Tela de Edição de Notificações



FONTE – AUTORIA PRÓPRIA (2025)

A Figura 16 apresenta a interface destinada à configuração de notificações no sistema S.C.O.R.E., acessada por meio do menu lateral. Essa funcionalidade permite ao usuário personalizar o comportamento das notificações de mensagens recebidas, adequando o sistema às suas preferências e necessidades de uso.

Na lateral esquerda da tela, encontram-se as opções de configuração, onde é possível selecionar o tipo de notificação a ser ajustada, neste caso, “Notificação de mensagem”. A seguir, há a seção de escolha do som da notificação, oferecendo quatro alternativas: Silenciado, Din din, Plano e *Attentive*. Essa variedade de sons proporciona maior flexibilidade para o usuário identificar diferentes tipos de mensagens.

Abaixo, há também a possibilidade de definir a duração da notificação, com três opções: Curta, Média e Longa, o que permite adaptar a persistência visual da notificação conforme a importância ou urgência das mensagens.

À direita, a área de conversação permanece visível, exibindo um exemplo de conversa entre o setor identificado como "Comp. Dir", o que evidencia que as

configurações podem ser feitas sem a necessidade de sair da tela de mensagens, promovendo maior usabilidade e fluidez na experiência do usuário.

Essa funcionalidade demonstra a preocupação do sistema em oferecer uma experiência personalizada, alinhando-se às boas práticas de *design* centrado no usuário.

2.3.1.17 Tela dos outros usuários

Figura 17 – Tela de Configurações Visualizada por Outros Usuários



FONTE – AUTORIA PRÓPRIA (2025)

A Figura 17 exibe a interface de configurações do sistema S.C.O.R.E., conforme visualizada por um usuário vinculado ao setor da Coordenação. Essa tela exemplifica o acesso às principais funcionalidades administrativas disponíveis para diferentes perfis dentro do sistema.

No menu lateral esquerdo, observa-se a identificação do setor ativo (Coordenação), com opções organizadas em uma coluna vertical, entre elas: Perfil, Conversas, Cadastrar usuário e Notificações. Tais opções permitem que o usuário configure sua conta, gerencie interações, adicione novos usuários ao sistema e personalize alertas de mensagens.

À direita da tela, encontra-se a janela de conversação com o “Computador 04”, mantendo o padrão visual adotado nas demais telas do sistema. A disposição das mensagens e o campo de digitação permanecem consistentes com a proposta de interface limpa e funcional, garantindo continuidade na experiência do usuário durante a navegação entre diferentes seções do sistema.

Essa tela evidencia a flexibilidade do S.C.O.R.E. ao adaptar as opções disponíveis de acordo com o setor ou perfil de acesso, além de facilitar a realização de tarefas administrativas sem comprometer a visualização da comunicação ativa. Perfeito! Aqui está a linha isolada:

Além disso, o layout intuitivo contribui para uma experiência mais ágil e organizada para todos os usuários.

2.4-CLIENTE

O cliente é o elemento central de qualquer estratégia empresarial, sendo o principal responsável por impulsionar as atividades comerciais de uma organização. Ele pode ser definido como o indivíduo ou empresa que adquire bens ou serviços, direta ou indiretamente, influenciando o desempenho e a sustentabilidade do negócio. Mais do que apenas comprador, o cliente atual busca valor agregado, bom atendimento, experiência personalizada e soluções que atendam às suas necessidades específicas. Com o avanço da tecnologia e o aumento da concorrência, compreender o perfil, as preferências e o comportamento dos clientes tornaram-se uma exigência estratégica. Empresas que conseguem construir um relacionamento sólido com seu público-alvo, investindo em escuta ativa, qualidade e inovação, tendem a se destacar no mercado e alcançar melhores resultados a longo prazo.

2.4.1-Cliente Interno

O projeto S.C.O.R.E. conta com uma equipe técnica composta por colaboradores internos da empresa, responsáveis por diversas etapas de planejamento, desenvolvimento, teste e manutenção da solução. Esses profissionais

cada qual em sua devida função, colaboram entre si para garantir a entrega de um sistema eficiente, funcional e alinhado ao esperado.

Para fidelizar os colaboradores é necessário criar um ambiente onde eles queiram permanecer e contribuir com motivação. Isso é essencial para manter a qualidade do serviço evitar rotatividade e garantir um bom projeto.

Benefícios dos colaboradores

Salário remunerado – Quando se refere a “Salário Remunerado”, se trata ao pagamento justo e legal, não apenas o salário base (valor fixo acordado para o trabalho), mas incluindo adicionais, como hora extra, adicional noturno, comissões e gratificações.

Vale-alimentação – O objetivo desse benefício é ajudar os colaboradores nas despesas como alimentação, pode ser utilizado para as compras de gêneros alimentícios em supermercados, mercearias, hortifrúteis e similares.

Plano de Saúde- Consiste em oferecer aos colaboradores serviços médicos, hospitalares e laboratoriais por meio de uma rede credenciada com custos reduzidos pela empresa, isso traz mais segurança e tranquilidade ao colaborador e sua família.

Cobertura dos Custos com a Internet para Colaboradores Remotos- Esse benefício consiste em reembolsar as despesas com a conexão de *internet*, garantindo que o funcionário tenha acesso a uma rede rápida para executar suas atividades.

2.4.2 Cliente Externo

O cliente externo representa uma conexão direta entre a equipe desenvolvedora e o ambiente educacional de aplicação, funcionando como um elo essencial para validar o sistema em situações reais de uso. No caso do sistema S.C.O.R.E., o cliente externo assume um papel fundamental ao aplicar a solução tecnológica no cotidiano escolar, oferecendo uma perspectiva prática e concreta sobre sua eficácia.

Sua atuação permite identificar de forma precisa quais funcionalidades são mais relevantes na rotina dos educadores, gestores e estudantes. A experiência direta com o sistema possibilita a emissão de *feedbacks* qualitativos, contribuindo com

sugestões e críticas que orientam ajustes e inovações, tornando o produto final mais funcional, adaptado e significativo para as instituições de ensino.

A participação ativa do cliente externo, como no caso do usuário João, fortalece o processo de melhoria contínua, garantindo que o sistema evolua com base em demandas reais e alinhado às necessidades práticas das escolas parceiras. Esse envolvimento também reforça o compromisso da equipe de desenvolvimento com a escuta ativa e a personalização da solução, elementos essenciais para o sucesso e longevidade de qualquer produto tecnológico educacional.

2.4.2.1- *Cliente Ativo*

Um cliente ativo é aquele que utiliza o sistema com regularidade, demonstrando engajamento contínuo e integração do serviço à sua rotina, "clientes ativos são os que mantêm relacionamento constante com a empresa, sendo usuários recorrentes de seus produtos ou serviços e contribuindo diretamente para a consolidação da marca". No contexto do sistema S.C.O.R.E., o cliente ativo é aquele que realiza acessos frequentes, utiliza as principais funcionalidades da plataforma e demonstra satisfação com o desempenho da solução. (LAS CASAS, 2012).

Para assegurar esse engajamento prolongado, o S.C.O.R.E. foi desenvolvido com foco na usabilidade, compatibilidade e confiabilidade. Entre as principais estratégias adotadas para manter o cliente ativo, destacam-se:

- a. Interface simples e intuitiva: o *design* do sistema prioriza a facilidade de navegação, reduzindo a curva de aprendizado e evitando a necessidade de treinamentos extensivos.
- b. Compatibilidade com sistemas operacionais padrão: o sistema é acessível em qualquer computador com navegadores convencionais, promovendo facilidade de acesso e flexibilidade de uso.
- c. Guia interativo interno: a presença de tutoriais passos a passo dentro da própria plataforma ajuda o usuário a explorar os recursos do sistema sem dependência externa.

- d. Segurança e estabilidade: o sistema conta com mecanismos de proteção das informações internas e funcionamento *offline* estável, o que garante a continuidade do serviço mesmo sem conexão à *internet*.
- e. Armazenamento seguro: os dados são preservados com camadas de proteção e criptografia, assegurando integridade e confidencialidade.
- f. *Feedbacks* e melhorias contínuas: a coleta de opiniões dos usuários orienta o aprimoramento constante do sistema, atendendo às demandas reais da base ativa.

Essas ações contribuem para manter um relacionamento positivo e duradouro com o cliente, transformando o uso recorrente em fidelização efetiva.

2.4.2.2- *Cliente Inativo*

Um cliente inativo é aquele que deixou de consumir ou utilizar um produto ou serviço após determinado período de tempo, mesmo que não tenha formalizado o cancelamento, “cliente inativo é aquele que, por qualquer motivo, deixou de comprar da empresa, mas que já demonstrou interesse anteriormente e pode ser reconquistado”. (LAS CASAS, 2012).

Empresas definem o tempo de inatividade conforme o comportamento médio de seus usuários. No caso do sistema S.C.O.R.E., considera-se inativo o cliente que não acessa ou utiliza o sistema por um período superior a seis meses.

No contexto da gestão do sistema S.C.O.R.E., define-se como cliente inativo aquele que deixa de utilizar a plataforma por um período igual ou superior a seis meses. Essa definição é baseada em práticas de relacionamento com o cliente, segundo as quais a inatividade não indica necessariamente a perda definitiva do usuário, mas sim uma interrupção do vínculo que pode ser revertida por meio de ações estratégicas (LAS CASAS, 2012).

Um caso relevante refere-se à escola que, após adotar o sistema S.C.O.R.E., optou por substituí-lo por uma nova ferramenta de gestão integrada com sistema de mensagens via *Wi-Fi*. Entre os motivos relatados para a descontinuação do uso, destacam-se:

- a. a dificuldade de manutenção dos cabos de rede, exigindo suporte técnico frequente;
- b. a baixa adesão dos demais funcionários, o que limitou o uso a poucos setores;
- c. a falta de funcionalidades modernas, como envio de arquivos, uso de *emojis* e grupos de conversa.

Apesar da suspensão, o cliente demonstrou familiaridade e interesse prévio na plataforma, o que torna viável a reaproximação. Para tanto, propõem-se as seguintes ações estratégicas de recuperação de cliente inativo:

Atualização do sistema: inclusão de funcionalidades como envio de arquivos leves (.pdf, .docx, imagens), *emojis*, mensagens pré-configuradas e grupos de conversa personalizados (ex: "Grupo da Coordenação", "3º Ano").

Ofertas e benefícios: disponibilização de brindes (*kit* personalizado da escola) e promoção de três meses gratuitos de suporte técnico com manutenção garantida na infraestrutura de rede.

Campanha de reaproximação: envio de comunicação formal (*e-mail* ou carta) informando sobre as melhorias implementadas desde a última utilização, com a possibilidade de agendamento de uma demonstração personalizada no ambiente escolar.

Proposta de parceria: sugestão de integração entre o sistema atual da escola e o S.C.O.R.E., utilizando-o como ferramenta de comunicação de *backup* — especialmente em casos de falhas na rede *Wi-Fi* ou em situações críticas e sigilosas que exigem uma rede independente.

Mesmo diante de objeções como a estabilidade do *Wi-Fi* ou a pouca adesão anterior dos colegas, o S.C.O.R.E. pode destacar seus diferenciais: autonomia de conexão, segurança em comunicações internas e melhorias funcionais que o tornam competitivo em relação aos mensageiros modernos.

2.4.3 Cliente Satisfeito

A satisfação do cliente é um dos principais indicadores de sucesso de um sistema, especialmente em ambientes escolares, onde a funcionalidade, a estabilidade e o suporte técnico impactam diretamente a rotina pedagógica e administrativa. Segundo Kotler e Keller (2012), a satisfação do cliente ocorre quando o desempenho do serviço atende ou excede as expectativas criadas. No contexto do sistema S.C.O.R.E., clientes satisfeitos demonstram uso contínuo, *feedbacks* positivos e disposição para recomendar a ferramenta. (KOTLER E KELLER, 2012).

Para manter esse alto nível de satisfação e engajamento, foram adotadas diversas estratégias focadas em melhoria contínua, relacionamento e suporte ativo, conforme descrito a seguir:

1. Atualizações Técnicas com Foco em Qualidade

Atualizações constantes são fundamentais para demonstrar comprometimento com a excelência. Entre as melhorias implementadas estão:

- a. Segurança aprimorada, com proteção contra falhas e perda de dados;
- b. Maior desempenho, com tempo de resposta reduzido e navegação fluida;
- c. Correção ágil de erros (*bugs*) reportados pelos usuários;
- d. Estabilidade operacional, com minimização de quedas e travamentos.

2. Novas Funcionalidades Relevantes

A inserção de novos recursos fortalece a percepção de valor do sistema. Entre as novidades em desenvolvimento estão:

- a. Chamadas de áudio e vídeo internas;
- b. Chat privado entre usuários;
- c. Integração com ferramentas externas (como *e-mail* institucional);
- d. Automatizações, como lembretes e relatórios automáticos;
- e. Suporte multiplataforma (dispositivos móveis e *tablets*);
- f. Personalização da interface conforme as preferências do usuário.

3. Coleta Ativa de *Feedback*

A escuta ativa permite alinhar o sistema às reais necessidades dos usuários. Por meio de formulários via *e-mail*, são feitas perguntas como:

- a. Qual sua experiência recente com a plataforma?
- b. Encontrou alguma dificuldade?
- c. Que melhorias ou recursos você gostaria de ver?

Esse canal contribui para decisões mais assertivas e para o fortalecimento do vínculo com o cliente.

4. Suporte Humanizado e Ágil

O atendimento ao usuário é um dos pilares da satisfação. O S.C.O.R.E. oferece:

- a. Respostas rápidas e personalizadas;
- b. Atendimento empático e cordial;
- c. Múltiplos canais de suporte (*WhatsApp*, *e-mail*, telefone);
- d. Base de conhecimento com FAQ atualizado.

5. Treinamento e Materiais de Apoio

Capacitar o usuário amplia seu domínio sobre o sistema e reduz atritos. As ações incluem:

- a. Vídeos curtos e objetivos;
- b. *Webinars* ao vivo com perguntas e respostas;
- c. Guias práticos e artigos na central de ajuda;
- d. Processo de *onboarding* guiado para novos usuários.

6. Acompanhamento Proativo

A manutenção do relacionamento vai além da resolução de problemas. Por isso, são realizadas ações como:

- a. *Check-ins* periódicos com usuários-chave;
- b. Relatórios mensais com métricas de uso e sugestões;
- c. Contato direto ao identificar queda no uso da plataforma.

Tais estratégias visam não apenas reter clientes atuais, mas transformá-los em defensores da solução, fortalecendo a imagem institucional do S.C.O.R.E.

2.4.4-Cliente Insatisfeito

O cliente insatisfeito constitui um dos principais desafios no relacionamento entre empresas e consumidores, especialmente no setor de tecnologia educacional, onde a percepção de valor está diretamente ligada à funcionalidade, usabilidade e suporte oferecido. De acordo com Kotler e Keller (2012), “clientes insatisfeitos podem não apenas abandonar a marca, mas também compartilhar suas experiências negativas com outros, o que amplia o impacto da insatisfação”. (KOTLER E KELLER, 2012).

As causas da insatisfação podem incluir falhas no atendimento, dificuldade de acesso ao sistema, ausência de funcionalidades esperadas, baixa qualidade técnica ou falta de suporte pós-venda. No caso de sistemas como o S.C.O.R.E., isso pode se manifestar em atrasos em atualizações, travamentos, ausência de recursos modernos ou uma curva de aprendizado difícil.

Além de interromper o uso da plataforma, o cliente insatisfeito pode influenciar negativamente a opinião de outros usuários, sobretudo por meio das redes sociais e plataformas de avaliação pública. Por isso, adotar uma postura proativa é essencial: ouvir os *feedbacks*, agir com agilidade na resolução de problemas e investir em melhoria contínua são estratégias fundamentais para recuperar a confiança do usuário.

Quando a insatisfação é bem gerida, ela pode ser revertida em uma oportunidade de fortalecimento do relacionamento. Demonstrar disposição para ouvir, corrigir e melhorar transmite ao cliente o compromisso da empresa com sua experiência, transformando uma situação negativa em um diferencial competitivo.

2.5 ÁREA DE ATUAÇÃO

A área de atuação diz respeito ao segmento em que as profissões estão inseridas. Defini-la pode ser o primeiro passo para escolher qual carreira seguir profissionalmente. Dentro das possibilidades para atuar, é possível encontrar, além das profissões, os cargos, a média salarial e as ocupações atribuídas. Algumas podem ser amplas, abrangendo uma quantidade maior de funções. Outras podem ser mais limitadas, com nichos específicos de atuação. (EQUIPEGRAN, 2024).

2.5.1 Microrregião

A microrregião é uma divisão territorial utilizada pelo Instituto Brasileiro de Geografia e Estatística (IBGE) para fins de estudo e análise de dados socioeconômicos. Ela representa uma área geográfica delimitada que engloba um conjunto de municípios próximos e com características semelhantes. Essa divisão é importante para compreender a realidade de determinada região, permitindo a identificação de particularidades e potencialidades locais. (AMX, 2025).

Para alcançar a microrregião de Mogi Guaçu de forma eficaz, a proposta de *marketing* da empresa contempla estratégias *offline* que envolvem diretamente a comunidade local. Dentre as ações planejadas, destacam-se as parcerias com comércios regionais, eventos escolares e espaços culturais, além de alianças estratégicas com cursos técnicos, escolas públicas e particulares. Também estão previstas apresentações presenciais voltadas a secretarias de educação e direções escolares, com o objetivo de apresentar os benefícios da plataforma e estabelecer uma relação direta com as instituições de ensino.

Além das parcerias, serão realizadas ações de divulgação regional, como a distribuição de folhetos em escolas, comércios e eventos locais, bem como a instalação de faixas e banners físicos em pontos estratégicos da cidade. Essas medidas visam aumentar a visibilidade da marca de maneira acessível e direcionada ao público-alvo da região.

Para garantir a fidelização e o engajamento contínuo das instituições, será implantado o programa SCORE+, que oferecerá benefícios exclusivos para escolas e empresas que utilizarem a plataforma com frequência. Entre esses benefícios estão pacotes personalizados, descontos progressivos conforme o uso, além da possibilidade de participação nas melhorias da ferramenta e acesso antecipado a novas funcionalidades.

A comunicação com o público será feita de forma direta e personalizada, com o uso de *e-mail marketing* segmentado por região, envio de mensagens automáticas com atualizações e promoções, e suporte contínuo por meio de um canal exclusivo via *WhatsApp* e redes sociais. Essa abordagem visa manter a comunidade informada, próxima e participativa.

O público-alvo na microrregião de Mogi Guaçu inclui escolas municipais e estaduais – sendo que o município conta com mais de 50 unidades públicas de ensino – além de instituições particulares de renome, como Objetivo, Anglo e Dom Bosco. Também fazem parte do perfil de clientes em potencial os cursos técnicos e profissionalizantes, como ETEC, Senai e Sest/Senat, além de empresas do setor educacional, como editoras, fornecedores de materiais escolares e prestadores de serviços de transporte escolar. (EDU,2025).

O objetivo inicial é alcançar entre 20 e 30 instituições durante a fase de implantação do projeto, consolidando a plataforma SCORE como uma referência regional em comunicação educacional. Para isso, estão previstas estratégias de marketing *digital* e *offline*, com estimativas de custo detalhadas. (_____,2025).

No *marketing digital*, a atuação incluirá a criação de redes sociais segmentadas por região, com investimento médio de R\$ 300 por canal, campanhas geolocalizadas em plataformas como *Google Ads* e *Instagram Ads*, com valor estimado de R\$ 1.000 mensais, além da produção de vídeos e depoimentos de usuários por R\$ 800 em investimento único. No *marketing offline*, o valor estimado para impressão de folhetos e *banners* regionais é de R\$ 1.500, a participação em feiras e eventos escolares terá um custo médio de R\$ 1.200, e os brindes promocionais para ações locais estão orçados em aproximadamente R\$ 500. O investimento inicial médio total em *marketing* varia entre R\$ 4.500 e R\$ 5.500. (_____,2025).

2.5.2-Macrorregião

De acordo com o Instituto Brasileiro de Geografia e Estatística (IBGE), as macrorregiões são áreas territoriais amplas que compartilham características geográficas, econômicas e sociais comuns. O Brasil é dividido em cinco grandes regiões: Norte, Nordeste, Centro-Oeste, Sudeste e Sul. Este trabalho foca na Região Sudeste, que é a mais industrializada do país e abriga os maiores centros econômicos do Brasil. (IBGE, 2018).

A escolha da Região Sudeste para o lançamento inicial do sistema proposto se justifica pela forte concentração de empresas, dinamismo econômico e ampla

conectividade, que são características essenciais para o sucesso de soluções tecnológicas voltadas à comunicação empresarial. A região concentra uma das maiores concentrações industriais do país, destacando-se principalmente em São Paulo, o maior centro econômico do Brasil, e em cidades como Rio de Janeiro e Belo Horizonte. Este cenário favorece a implementação do sistema, dado o perfil empresarial dinâmico e a alta demanda por soluções tecnológicas avançadas. (BRASIL ESCOLA, 2025).

A indústria na Região Sudeste é predominantemente voltada para setores como automotivo, químico, tecnologia da informação, alimentação e energia, o que representa um mercado robusto e crescente para soluções inovadoras. São Paulo, por exemplo, é sede de um grande número de empresas de médio e grande porte, muitas das quais estão localizadas na Grande São Paulo, que abriga cerca de 60.000 a 70.000 indústrias (IBGE, 2021).

A estratégia de *marketing* será centrada na segurança e velocidade do sistema, com a produção de conteúdos *digitais* (*posts* e vídeos) que reforçam esses atributos. Para aumentar a visibilidade da empresa, será também desenvolvido um site institucional, proporcionando uma comunicação mais eficaz com os clientes e facilitando a expansão nacional. (_____,2021).

2.5.2.1 Perfil dos Influenciadores

A parceria com influenciadores que possuem grande alcance e um público focado em negócios, empreendedorismo e transformação *digital* é uma das estratégias centrais da campanha de *marketing*. O perfil dos influenciadores selecionados é altamente alinhado com o perfil empresarial da solução, sendo que eles atingem diretamente um público, composto por empresários, *startups*, profissionais de tecnologia e líderes de negócios.

O perfil de cada influenciador é apresentado com base no número de seguidores e no tipo de conteúdo que produzem:

Endeavor Brasil: Com 169 mil seguidores no Instagram e 421 mil inscritos no *YouTube*, o canal é focado em empreendedorismo, inovação e gestão de empresas, oferecendo conteúdo que atrai empresários de pequeno, médio e grande porte que

buscam aprimorar seus negócios. O público é majoritariamente adulto, com interesse em desenvolvimento corporativo, estratégias de negócios e tecnologia. (INFOMONEY25, 2025).

Do Zero ao Topo: Com 40 mil seguidores no *YouTube* e 99,9 mil no *Instagram*, este canal se foca em histórias de sucesso no empreendedorismo e transformação empresarial, atraindo uma audiência jovem e adulta, composta por empreendedores iniciantes, *startups* e profissionais de tecnologia e negócios. (_____,2025).

Hey Investidor: Com 1,4 milhões de seguidores no Instagram e 918 mil inscritos no YouTube, o Hey Investidor é um canal voltado para educação financeira, investimentos e gestão de negócios, atraindo empresários, investidores e profissionais da área financeira, além de indivíduos com interesse em tecnologia financeira. (_____,2025).

2.5.2.2 Estimativa de Alcance para Empresas

Com a escolha estratégica desses influenciadores, o alcance potencial para o público-alvo pode ser substancial, especialmente considerando o alto grau de segmentação e o perfil empresarial dos seguidores. As campanhas com influenciadores digitais podem alcançar de 5% a 10% do total de seguidores, dependendo do engajamento, da relevância do conteúdo e da afinidade com o público-alvo. (ROCK CONTENT, 2024)

Com base nessa referência, as estimativas de impacto para empresas da Região Sudeste são:

Endeavor Brasil: Com aproximadamente 169 mil seguidores no *Instagram*, estima-se que entre 5% e 10% desse público possa ser impactado diretamente por campanhas alinhadas com soluções de tecnologia e inovação. Isso representa um alcance entre 8.000 e 17.000. (_____, 2024).

Do Zero ao Topo: Com cerca de 40 mil inscritos no *YouTube* e quase 100 mil seguidores no *Instagram*, o canal atinge um público engajado composto por empreendedores e pequenos empresários. Aplicando a mesma taxa de 5% a 10%, o

alcance estimado fica entre 3.000 e 6.000 empresas potenciais impactadas diretamente pelo conteúdo patrocinado. (_____, 2024).

Hey Investidor: Com 1,4 milhão de seguidores no *Instagram* e 918 mil inscritos no *YouTube*, é o influenciador com maior alcance da campanha. Sua audiência é formada majoritariamente por investidores, empresários e profissionais da área financeira, altamente compatíveis com a proposta do sistema. Assim, estima-se que ele possa impactar entre 14.000 e 20.000 empresas da Região Sudeste. (_____, 2024).

Somando os três canais, as campanhas podem atingir diretamente entre 25.000 e 43.000 empresas com interesse em soluções de comunicação digital, dentro de um público qualificado e propenso à adesão de novos serviços tecnológicos.

Essa estimativa está em conformidade com as práticas que destaca o *marketing* de influência como uma das estratégias mais eficazes para gerar conexões reais com decisores de negócios, especialmente no ambiente digital. (_____, 2024).

2.6 CONCORRENTES

A concorrência é um fator essencial no mercado, impulsionando inovação e aprimoramento contínuo. "Os concorrentes desafiam empresas a evoluir, aprimorar seus serviços e atender melhor às necessidades dos consumidores". Eles podem ser classificados em duas categorias principais: concorrentes diretos, que oferecem serviços semelhantes, e concorrentes indiretos, que atendem às mesmas necessidades por meio de abordagens distintas. (BENNETT, 2024)

Para superar a concorrência, as empresas devem adotar estratégias eficazes. Uma empresa que deseja se destacar precisa oferecer um diferencial, seja pela inovação, eficiência ou experiência do usuário. Algumas das principais abordagens incluem: (PRICEVA, 2024),

Valor agregado: Oferecer funcionalidades exclusivas que resolvam problemas específicos do público (BENNETT, 2024).

Experiência do usuário: A interface intuitiva e o suporte eficiente são fatores-chave para conquistar e fidelizar clientes. (MEDIUM, 2022).

Segurança e privacidade: Conforme destaca, A proteção de dados tornou-se um diferencial competitivo essencial em um mundo *digital*. (KILBERG, 2025)

Acessibilidade: A viabilidade da comunicação em diferentes condições, especialmente em cenários de baixa conectividade, é um elemento estratégico (WHELLHOUSE, 2025).

Diante dessa realidade, o S.C.O.R.E se posiciona como uma alternativa diferenciada ao oferecer uma comunicação segura, rápida e *offline*, eliminando dependências externas e reforçando o controle institucional.

2.6.1 Concorrentes Diretos

Concorrentes diretos são empresas que oferecem produtos ou serviços semelhantes ou idênticos, direcionados à mesma base de clientes e segmentos de mercado. Esse tipo de concorrência é mais evidente quando as empresas operam na mesma área geográfica ou atendem às mesmas necessidades dos clientes (BENNETT, 2024).

Concorrentes diretos do S.C.O.R.E

Os principais concorrentes do S.C.O.R.E são plataformas consolidadas de comunicação interna, como mensageiros instantâneos, serviços de videoconferência e SMS, que inclui o *WhatsApp*, *Discord*, *Microsoft Teams*, *Slack* e *Google Meet*.

WhatsApp / WhatsApp Business

Aplicativos populares de mensagens que permitem envio de texto, voz, imagens, vídeos e documentos. Vantagens: Comunicação instantânea e multimídia sem necessidade de infraestrutura própria. O *WhatsApp Business* oferece recursos avançados, como catálogo, respostas automáticas e integração via API. Desvantagens: Depende totalmente de *internet*; pode falhar em conexões instáveis. Há coleta de metadados e riscos à privacidade, mesmo com criptografia.

Discord

Plataforma gratuita com canais de texto e voz, utilizada por comunidades e grupos de trabalho. Vantagens: Organização por canais temáticos e suporte a *bots*,

facilitando a moderação e automação de atividades. Desvantagens: Pode gerar distrações por múltiplos servidores e falta de controle institucional sobre quem e como participa. Além disso, a plataforma permite a integração com diversas ferramentas externas, como serviços de streaming e aplicativos de produtividade. Oferece também opções de personalização, permitindo adaptar servidores às necessidades de cada grupo. A comunicação em tempo real favorece a colaboração entre membros, mesmo em equipes distribuídas geograficamente. No entanto, requer uma boa gestão para evitar sobrecarga de informações e garantir um uso eficiente dos recursos disponíveis.

Microsoft Teams

Ferramenta corporativa integrada ao Microsoft 365, com *chat*, videoconferência e gestão de arquivos. Vantagens: Alta segurança e *compliance*, eficiência comprovada com ganhos de produtividade e economia financeira. Desvantagens: Requer licenciamento, treinamento, infraestrutura de *internet* confiável e pode ser complexo para ambientes menores.

Slack

Plataforma profissional de *chat* por canais, com integrações extensivas e *bot-friendly*. Vantagens: Mensagens em tempo real, busca eficiente e integração com milhares de *apps* e serviços. Desvantagens: Limitações no plano gratuito (armazenamento e histórico), risco de excesso de notificações e críticas ao uso institucional. (BULKGATE, 2012).

Google Meet

Serviço de videoconferência parte do *Google Workspace*, com *chat* e integração ao *Gmail* e Agenda (WHEELHOUSE, 2025). Vantagens: Interface simples, integração direta com Agenda e *Gmail*, *criptografia* e legendas em tempo real. Desvantagens: Recursos limitados no plano gratuito (salas, *layouts*), sem *whiteboard* nativo e pode apresentar instabilidade de navegação.

SMS (Short Message Service – Serviço de Mensagens)

Serviço de mensagens via rede celular, amplamente usado para comunicações urgentes. Vantagens: Alta taxa de leitura (~98%), entrega rápida e custo baixo, ideal para alertas pontuais. Desvantagens: Limite de caracteres, mensagens podem ser interpretadas como *spam*, exigem *opt-in* e podem ter custos cumulativos (KIXIE, 2025).

Enquanto os concorrentes diretos oferecem soluções robustas e populares, todos dependem de conectividade estável e expõem dados a servidores externos. O S.C.O.R.E se diferencia por funcionar *offline*, de forma local e segura, eliminando a necessidade de internet e reforçando o controle interno da comunicação institucional.

2.6.2 Concorrentes Indiretos

Concorrentes indiretos, por outro lado, atendem às mesmas necessidades dos clientes, mas com produtos ou serviços diferentes. Eles podem não competir na mesma categoria de produto, mas ainda disputam o mesmo gasto do consumidor (BENNETT, 2024).

Plataformas de Gestão Escolar / Portais Educacionais

Como o Bridge, que centraliza a comunicação entre professores, alunos e responsáveis, com posts, avisos, tarefas, agendamento de reuniões e tutoriais para engajamento durante o ensino híbrido. Vantagens: Integração a sistemas escolares, engajamento acadêmico e interação com responsáveis. Desvantagens: Dependem de *internet*, muitas vezes requerem desenvolvimento ou contratação personalizada e podem não oferecer a robustez de um canal físico dedicado. (MEDIUM, 2022)

Comunicações por *E-mail* Institucional

Ferramenta universal de comunicação escrita, usada para envio de boletins, comunicados e documentos formais. Vantagens: Registro histórico, formalidade, possibilidade de anexos e controle de acesso. Desvantagens: Pouco imediatismo,

pode ser ignorado ou filtrado como *spam*; não atende bem a urgências ou avisos rápidos.

Avisos Físicos / Murais e Comunicados Impressos

Cartazes, recados, quadros de avisos e circulares colados em locais estratégicos da escola. Vantagens: Método simples, sem necessidade de tecnologia; fácil acesso para todos. Desvantagens: Comunicação unidirecional, sem confirmação de leitura; requer impressão constante e pode ficar desatualizado.

Páginas e Redes sociais institucionais

Uso de *Facebook*, *Instagram* e *blog* da escola para publicações, eventos, fotos e vídeos. Vantagens: Alto alcance, engajamento visual, fácil atualização; pode gerar *marketing* para a instituição. Desvantagens: Exposição pública de informações, possibilidade de críticas; depende de internet e aprovação da comunidade escolar.

Rádio ou Sistema de PA (*Public Address*)

Sistema de alto-falante interno usado para chamadas e avisos gerais. Vantagens: Excelente para avisos urgentes, cobertura para áreas amplas. Desvantagens: Apenas comunicação em tempo real, sem registro nem personalização; pode ser perturbador e invasivo se utilizado com frequência.

Os concorrentes indiretos, embora variados e úteis, não oferecem comunicação bidirecional, em tempo real e sem internet. O S.C.O.R.E surge como uma solução híbrida entre acessibilidade, agilidade e confiabilidade, permitindo comunicações rápidas e seguras mesmo em ambientes com baixa conectividade.

2.7 LEIS E REGRAS CONTRATUAIS

As leis e regras contratuais constituem o conjunto de normas jurídicas que regulam a formação, validade, execução e extinção dos contratos. No Brasil, essas

normas estão principalmente previstas no Código Civil (Lei nº 10.406/2002), Art. 421. O contrato deve obedecer aos princípios da função social e da boa-fé objetiva. Mas também são complementadas por princípios constitucionais, jurisprudência e doutrina especializada.

“O contrato é um acordo de vontades entre duas ou mais partes, com o objetivo de criar, modificar ou extinguir obrigações, sendo regido por princípios como a autonomia da vontade, a boa-fé objetiva e a função social”. Esses princípios garantem que os contratos sejam instrumentos legítimos de organização das relações jurídicas e econômicas, respeitando os direitos das partes envolvidas. (GONÇALVES, 2023)

O Direito Contratual, ramo do Direito Civil, estabelece que os contratos devem observar requisitos essenciais, como: agente capaz, objeto lícito, forma prescrita ou não proibida por lei e consentimento livre de vícios (erro, dolo, coação). Além disso, os contratos devem ser executados com lealdade e transparência, conforme o princípio da boa-fé objetiva (SARAIVA EDUCAÇÃO, 2023).

No contexto de projetos tecnológicos como o S.C.O.R.E, as regras contratuais são fundamentais para formalizar parcerias, definir responsabilidades, garantir a proteção de dados e assegurar a prestação de serviços conforme os termos acordados. A elaboração de contratos claros e juridicamente válidos contribui para a segurança das relações institucionais e a sustentabilidade do projeto.

2.7.1 Relatório de Termos e Condição

O desenvolvimento do S.C.O.R.E — um aplicativo de comunicação institucional que opera por meio de cabos de rede e funciona *offline* — exige atenção especial à legislação em três esferas: municipal, estadual e federal.

A seguir, estão as principais normas, seus efeitos no projeto e recomendações para adequação jurídica e institucional.

2.7.1.1 Legislação Municipal – Mogi Guaçu (SP)

A Lei Municipal nº 5.217/2019 trata da transparência da administração pública, regulamentando o acesso à informação pelos cidadãos no município:

“A administração pública municipal direta e indireta deverá garantir, mediante procedimentos objetivos e ágeis, o direito de acesso à informação de interesse público” (MOGI GUAÇU, 2019).

Ou seja, incentiva a criação de canais institucionais de comunicação, como o S.C.O.R.E, que favorecem a transparência e a gestão pública eficiente. Embora Mogi Guaçu não disponha de legislação específica sobre infraestrutura de telecomunicações, a iniciativa pode ser enquadrada como mecanismo de apoio à política de acesso à informação.

2.7.1.2 Legislação Estadual – São Paulo

A Lei Estadual nº 17.334/2021 criou o Cadastro de Bloqueio de *Telemarketing*, incluindo o bloqueio do envio de mensagens por aplicativos ou SMS:

“O consumidor poderá manifestar sua vontade de não receber chamadas de *telemarketing* ou mensagens de qualquer natureza” (SÃO PAULO, 2021, art. 1º).

Embora direcionada a empresas com fins comerciais, essa norma reflete a sensibilidade do Estado à privacidade do usuário. Assim, mesmo em sistemas de comunicação local como o S.C.O.R.E, recomenda-se a inclusão de controles de *opt-in* (aceitação) e *opt-out* (cancelamento) para o recebimento de mensagens em massa, respeitando a vontade dos usuários.

2.7.1.3 Legislação Federal – Brasil

Marco Civil da Internet (Lei nº 12.965/2014)
Define os direitos e deveres no uso da internet no Brasil, incluindo princípios aplicáveis à comunicação *digital*, mesmo em redes privadas:

“A disciplina do uso da internet no Brasil tem como fundamentos o respeito à liberdade de expressão e à proteção da privacidade e dos dados pessoais” (BRASIL, 2014, art. 3º).
“O sigilo do fluxo de comunicações pela *internet* é inviolável, salvo por ordem judicial” (BRASIL, 2014, art. 7º).

Mesmo funcionando *offline*, o S.C.O.R.E deve manter políticas de sigilo, privacidade e segurança, alinhadas aos princípios dessa lei.

Lei Geral de Proteção de Dados – LGPD (Lei nº 13.709/2018)
Aplica-se a qualquer operação de tratamento de dados pessoais, inclusive em ambientes sem conexão com a *internet*.

“O tratamento de dados pessoais somente poderá ser realizado mediante o fornecimento de consentimento pelo titular” (BRASIL, 2018, art. 7º).

Se o aplicativo armazenar nomes, mensagens, registros ou outras informações sensíveis, deverá implementar medidas de segurança, obter consentimento explícito e manter transparência sobre o uso dos dados.

Lei nº 9.296/1996 – Interceptação de Comunicações
Estabelece que comunicações só podem ser acessadas com ordem judicial:

“A interceptação de comunicações telefônicas, de qualquer natureza, (...) dependerá de ordem do juiz competente, da forma que a lei estabelecer para fins de investigação criminal” (BRASIL, 1996, art. 1º).

Essa legislação reforça o sigilo das mensagens trocadas dentro do aplicativo S.C.O.R.E, resguardando a confidencialidade dos usuários.

Lei Geral das Antenas – Lei nº 13.116/2015
Facilita a instalação de infraestrutura de telecomunicações em espaços públicos:

“É garantido o direito de passagem em vias públicas para instalação de infraestrutura sem ônus para a prestadora” (BRASIL, 2015, art. 12).

Isso pode beneficiar o projeto caso a instalação de cabos físicos envolva espaços compartilhados, como escolas públicas ou repartições municipais.

Lei Geral de Telecomunicações – Lei nº 9.472/1997 Define que qualquer forma de comunicação por meio físico ou eletromagnético é considerada serviço de telecomunicações:

“Serviço de telecomunicações é a transmissão, emissão ou recepção de símbolos, caracteres, sinais, escritos, imagens, sons ou informações de qualquer natureza por fio, rádio, eletricidade ou outro meio eletromagnético” (BRASIL, 1997, art. 60).

Se o S.C.O.R.E for expandido futuramente para além da rede local (por exemplo, via *internet* ou interconexão com outras instituições), pode ser necessária autorização da Anatel e observância das normas técnicas específicas da agência.

2.7.2 Medidas e Ações

O S.C.O.R.E, por ser um sistema de comunicação local e institucional baseado em cabos de rede, precisa não apenas estar tecnicamente funcional, mas também juridicamente adequado e socialmente aceitável. Para isso, é essencial que sua implantação seja acompanhada de medidas práticas que o alinhem às legislações nas esferas municipal, estadual e federal. A seguir, são apresentadas ações recomendadas conforme os diferentes níveis de legislação analisados:

2.7.2.1 No âmbito municipal – Mogi Guaçu (SP)

Objetivo: garantir que o S.C.O.R.E seja reconhecido como ferramenta legítima de apoio à transparência pública e à modernização dos canais de comunicação interna.

Apresentar o projeto à administração pública municipal, demonstrando como o sistema contribui diretamente com os princípios da Lei Municipal nº 5.217/2019, ao facilitar o acesso à informação institucional e a fluidez da comunicação entre os setores administrativos das escolas.

Buscar apoio da Secretaria de Educação do município, com a proposta de implementar o S.C.O.R.E inicialmente em escolas públicas como projeto-piloto,

destacando que o sistema não depende de *internet* e utiliza infraestrutura simples (cabos de rede).

Propor à Câmara Municipal o reconhecimento do projeto por meio de moção de apoio ou indicação legislativa, o que pode fortalecer a legitimidade institucional do aplicativo e abrir caminhos para futuras parcerias ou programas educacionais baseados em tecnologia *offline*.

2.7.2.2 No âmbito estadual – Estado de São Paulo

Objetivo: garantir que a aplicação do S.C.O.R.E respeite a privacidade dos usuários e esteja de acordo com normas estaduais sobre envio de mensagens.

Inserir mecanismos de consentimento no aplicativo, como caixas de seleção (*opt-in*) para autorizar o recebimento de mensagens e a opção de cancelamento (*opt-out*), atendendo à Lei Estadual nº 17.334/2021 Altera a Lei nº 13.226/2008 para atualizar as regras do Cadastro para Bloqueio de Recebimento de Ligações de Telemarketing no Estado de São Paulo, permitindo que consumidores evitem chamadas indesejadas de oferta de produtos e serviços., que protege o cidadão contra comunicações indesejadas.

Desenvolver uma Política de Privacidade acessível e objetiva, mesmo *offline*, que explique de forma clara os tipos de mensagens enviadas, o propósito das comunicações e a frequência de envio. Essa política pode estar disponível no menu principal do aplicativo, em formato visual simples e de fácil leitura.

Manter registro do consentimento dos usuários, com data e hora, armazenado localmente no servidor ou computador da escola, criando uma trilha de conformidade que pode servir como evidência legal.

2.7.2.3 No âmbito federal – Legislação nacional

Objetivo: assegurar que o S.C.O.R.E esteja em conformidade com as principais normas nacionais de proteção de dados, sigilo de comunicação e uso de infraestrutura de telecomunicações.

Conformidade com a LGPD (Lei nº 13.709/2018):

Coletar apenas os dados estritamente necessários (princípio da minimização).

Solicitar o consentimento explícito do usuário antes de armazenar qualquer dado pessoal, como nomes, mensagens ou identificadores.

Utilizar práticas de segurança da informação, como autenticação por senha, criptografia local e controle de acesso aos dados armazenados.

Garantir ao usuário o direito de acesso, correção e exclusão de seus dados sempre que solicitado.

Respeito ao sigilo de comunicação (Lei nº 9.296/1996):

Implementar o sistema de forma que apenas usuários autorizados tenham acesso ao conteúdo das mensagens.

Impedir qualquer forma de interceptação interna ou externa sem autorização judicial, garantindo a confidencialidade conforme exigido pela legislação.

Incluir cláusula no termo de uso informando que o conteúdo das mensagens será protegido por sigilo, conforme a lei.

Atenção à infraestrutura de telecomunicações:

Utilizar o direito de passagem previsto na Lei nº 13.116/2015 Art. 1º – Esta Lei estabelece normas gerais para o licenciamento, a instalação e o compartilhamento da infraestrutura de telecomunicações, com o objetivo de promover a expansão da rede e ampliar o acesso aos serviços no território nacional.

para facilitar a instalação de cabos em ambientes públicos como escolas e repartições sem custo adicional de autorização.

Planejar a instalação dos cabos respeitando a estrutura física dos prédios, evitando furos em paredes de áreas protegidas, respeitando normas de segurança elétrica e estrutural.

Se houver expansão para além do ambiente local (como entre prédios ou escolas diferentes), avaliar a necessidade de licenciamento na Anatel conforme previsto na Lei nº 9.472/1997 Art. 2º – O serviço de telecomunicações tem como finalidade o atendimento ao interesse público, devendo ser prestado com base nos

princípios da universalização do serviço e da continuidade. Isso inclui eventual registro do sistema como serviço de comunicação institucional interna, mesmo sem fins comerciais.

2.7.3 – CONTRATO DE PRESTAÇÃO DE SERVIÇOS

Este documento tem como objetivo formalizar o Contrato de Prestação de Serviços entre as partes interessadas na implementação do Sistema de Comunicação *Offline* para Redes Escolares (S.C.O.R.E).

Este contrato garante a legalidade e a conformidade jurídica do acordo, assegurando que todas as cláusulas estejam em conformidade com o Código Civil Brasileiro, bem como com demais normas técnicas e operacionais voltadas à prestação de serviços e fornecimento de soluções tecnológicas.

O S.C.O.R.E é uma solução tecnológica voltada para a melhoria da comunicação interna em instituições educacionais, utilizando redes físicas (cabos de rede) para transmitir mensagens sem depender da *internet*.

O sistema foi desenvolvido por Valério Cândido da Silva Junior, líder do projeto e responsável técnico pelo seu desenvolvimento.

O presente contrato define claramente os direitos e responsabilidades de cada parte, estabelecendo obrigações, prazos e condições para a implementação e manutenção do sistema.

Além disso, o documento aborda aspectos relacionados à confidencialidade das informações, à segurança dos dados e à proteção do patrimônio tecnológico das instituições envolvidas.

Também são especificadas as condições de suporte técnico, treinamento de usuários e eventuais atualizações do *software*, garantindo a continuidade e eficiência do serviço. O contrato prevê ainda mecanismos de resolução de conflitos e penalidades em caso de descumprimento das cláusulas, proporcionando segurança jurídica para ambas as partes.

Segue o contrato abaixo:

Contrato de Prestação de Serviços - S.C.O.R.E.

CONTRATANTE: [Nome da Instituição/Empresa], inscrita no CNPJ sob o nº _____, com sede na _____, e endereço eletrônico _____, neste ato representada por _____, doravante denominada simplesmente "CONTRATANTE".

CONTRATADO: Valério Cândido da Silva Junior, brasileiro, estudante e desenvolvedor, inscrito no CPF sob o nº _____, com endereço à _____, e e-mail _____, doravante denominado simplesmente "CONTRATADO".

CLÁUSULA 1 – DO OBJETO

Este contrato tem por objeto a prestação de serviços técnicos pelo CONTRATADO à CONTRATANTE, compreendendo a entrega, instalação, configuração, manutenção e suporte do sistema S.C.O.R.E. (Sistema de Comunicação Offline para Redes Escolares), tecnologia destinada à comunicação interna por meio de redes físicas, sem uso de internet.

CLÁUSULA 2 – DAS FUNDAMENTAÇÕES LEGAIS

Para garantir a legalidade, a segurança jurídica e a adequação institucional do serviço prestado, as partes reconhecem como base deste contrato as seguintes leis, com suas respectivas justificativas:

2.1 – Código Civil Brasileiro (Lei nº 10.406/2002)

Este contrato fundamenta-se nas disposições do Código Civil Brasileiro, especialmente nos princípios da autonomia da vontade, boa-fé objetiva e função social do contrato, assegurando sua validade, formação, execução e encerramento de acordo com a legislação vigente.

2.2 – Lei nº 13.709/2018 – Lei Geral de Proteção de Dados (LGPD)

O tratamento de dados pessoais no âmbito deste sistema observará rigorosamente as disposições da LGPD, garantindo o consentimento explícito dos

titulares, a minimização da coleta de dados e o pleno respeito aos direitos dos titulares, mesmo em ambiente offline.

2.3 – Lei nº 12.965/2014 – Marco Civil da Internet

O sistema assegurará a proteção da privacidade, a liberdade de expressão e o sigilo das comunicações, conforme estabelecido pelo Marco Civil da Internet, aplicando tais princípios inclusive em redes privadas.

2.4 – Lei nº 9.296/1996 – Interceptação de Comunicações

As comunicações realizadas por meio do sistema serão mantidas em sigilo, sendo vedado qualquer acesso não autorizado, exceto mediante ordem judicial, em conformidade com a legislação aplicável.

2.5 – Lei nº 13.116/2015 – Lei Geral das Antenas

A instalação da infraestrutura necessária para o funcionamento do sistema observará as disposições da Lei Geral das Antenas, garantindo a facilitação da instalação em locais públicos, sem custos adicionais.

2.6 – Lei nº 9.472/1997 – Lei Geral de Telecomunicações

Eventual expansão do sistema envolvendo serviços de telecomunicações será realizada em estrita observância às normas da Agência Nacional de Telecomunicações (ANATEL), conforme disposto na Lei Geral de Telecomunicações.

2.7 – Lei Estadual nº 17.334/2021 (São Paulo)

Será assegurado ao usuário o direito de não receber mensagens não solicitadas, sendo disponibilizadas opções claras de consentimento prévio (*opt-in*) e de exclusão (*opt-out*) para comunicações realizadas pelo sistema.

2.8 – Lei Municipal nº 5.217/2019 (Mogi Guaçu – SP)

O sistema S.C.O.R.E. contribuirá para o acesso à informação e à transparência na administração pública, em conformidade com os princípios estabelecidos pela referida legislação municipal.

CLÁUSULA 3 – OBRIGAÇÕES DO CONTRATADO

1. Fornecer o sistema S.C.O.R.E. pronto para uso, com instalação e configuração inicial;
2. Garantir estabilidade, segurança e atualizações durante a vigência;
3. Assegurar sigilo total das comunicações (conforme Lei nº 9.296/1996);
4. Fornecer infraestrutura necessária (como cabos);
5. Manter contato com a CONTRATANTE semestralmente para avaliação técnica.

CLÁUSULA 4 – OBRIGAÇÕES DA CONTRATANTE

1. Disponibilizar estrutura física e elétrica para o sistema;
2. Permitir o acesso e a instalação da rede local;
3. Colaborar com feedbacks e avaliações;
4. Efetuar os pagamentos em dia, com multa de 2% e juros de 1% ao mês em caso de atraso superior a 7 dias.

CLÁUSULA 5 – VIGÊNCIA E RENOVAÇÃO

Este contrato terá validade inicial de 1 (um) mês, podendo ser prorrogado por acordo mútuo entre as partes, por meio de termo aditivo.

CLÁUSULA 6 – TESTEMUNHAS (OPCIONAL)

Nos termos do art. 104 do Código Civil, não é obrigatória a presença de testemunhas para a validade deste contrato. Entretanto, se desejado por uma das partes, poderão ser incluídas para fins de segurança jurídica, conforme o art. 784, III, do CPC.

CLÁUSULA 7 – DISPOSIÇÕES FINAIS

1. Este contrato não gera vínculo empregatício entre as partes;
2. Alterações contratuais só terão validade mediante aditivo escrito e assinado;

3. O foro eleito para dirimir eventuais controvérsias é o da Comarca de Mogi Guaçu/SP.

Mogi Guaçu, _____ de _____ de 2024

CONTRATANTE:

[Nome da Instituição/Empresa]

CONTRATADO:

Valério Cândido da Silva Junior

TESTEMUNHAS:

Nome: _____ – CPF: XXX.XXX.XXX-XX

Nome: _____ – CPF: XXX.XXX.XXX-XX

3 – DOCUMENTAÇÃO

Remete à noção de conjunto de documentos, bem como se refere às ações de coleta, processamento técnico e disseminação de informações. Em linhas gerais, o termo documentação pode ser compreendido como prática com e/ou sobre algum documento ou conjunto documental. Importa destacar a característica de evidenciar vários tipos de registros, garantindo a permanência da informação ao longo dos diferentes contextos históricos. Muito além de agregar as múltiplas expressões do conhecimento humano, a documentação assume a função de representar ideias e objetos que nos informam sobre algo. (IPHAN,2025)

3.1 – OMG

O OMG (*Object Management Group* – Grupo de Gerenciamento de Objetos) é um consórcio internacional, aberto à participação de empresas, academia e órgãos governamentais, que desempenha papel central na padronização de tecnologias de informação. Fundado em 1989, o OMG organiza seu trabalho por meio de *Task Forces* responsáveis por propor, manter e evoluir padrões abertos voltados à interoperabilidade entre sistemas, domínios e plataformas (OMG, 2025).

Nesse contexto surgiu a *Object Management Architecture* (OMA), a arquitetura de referência do OMG que estrutura conceitos, interfaces e serviços para sistemas orientados a objetos, fornecendo uma base uniforme para especificações técnicas e facilitando a integração entre aplicações heterogêneas. A OMA define componentes, protocolos e operações essenciais (por exemplo, como objetos fazem requisições e respostas), atuando como fundamento conceitual para outros padrões do OMG (OMG, 1998.).

O propósito institucional do OMG permanece o de promover interoperabilidade e integração via padrões abertos e consensuais, sem fins lucrativos, desde sua criação em 1989 (OMG, 2025).

A necessidade de padrões de interoperabilidade em um mercado de *software* em rápida evolução levou empresas de tecnologia a fundar o consórcio e priorizar *middleware* e integração desde os primeiros anos (OMG, 2025).

Nos anos iniciais, a OMG concentrou-se em padrões como a CORBA (*Common Object Request Broker Architecture*), que viabiliza chamadas entre objetos

distribuídos em diferentes linguagens e plataformas por meio de *Object Request Brokers* (OMG, 1995).

Entre 1996 e 2005, o portfólio do OMG se expandiu com iniciativas como MDA (*Model-Driven Architecture*), UML (*Unified Modeling Language*) e BPMN (*Business Process Model and Notation*). A UML consolidou-se como padrão de mercado para modelagem de *software* e sistemas, com suas versões mantidas pelo OMG (OMG, 2017).

De 2006 a 2015, o OMG seguiu atualizando e criando especificações para novas demandas. Destacam-se DDS (*Data Distribution Service*), um padrão de publicação-assinatura data-cêntrico para comunicação em tempo real e sistemas distribuídos, e a SoaML (*Service-Oriented Architecture Modeling Language*), um perfil UML para modelagem de serviços em arquiteturas orientadas a serviços (OMG, 2015; OMG, 2012).

Desde 2016, o OMG continua a evoluir seus padrões para novas tecnologias. A BPMN tem sua versão 2.0.2 como estado-da-arte mantido pelo OMG e também ratificada como ISO 19510, frequentemente combinada com motores de processo e automação. Em paralelo, o consórcio intensificou frentes em *IoT* (com a família DDS) e em Inteligência Artificial (por meio da *AI Platform Task Force* e grupos conjuntos), reforçando a interoperabilidade entre gêmeos digitais, robótica e outros domínios (OMG, 2024).

3.1.1 – Principais padrões do OMG (sintético)

CORBA – *middleware* para objetos distribuídos; base histórica de interoperabilidade. (OMG, 1995).

UML – linguagem de modelagem unificada e adotada amplamente (versão 2.5.1). (OMG, 2017).

MDA – abordagem dirigida a modelos lançada pelo OMG em 2001. (POOLE, 2001).

BPMN – notação de processos de negócio (versão 2.0.2). (OMG, 2014).

DDS – *data-centric publish-subscribe* para tempo real e IoT/IIoT. (OMG, 2015).

SoaML – perfil UML para arquitetura orientada a serviços. (OMG, 2012).

3.1.2 – Utilização no Sistema

No desenvolvimento do S.C.O.R.E (Sistema de Comunicação Offline Rápida Escolar), a aplicação dos padrões estabelecidos pelo *Object Management Group* (OMG) foi fundamental para garantir estruturação, clareza e interoperabilidade entre os componentes do projeto. Como a proposta envolve um ambiente educacional que precisa funcionar mesmo em condições de conectividade limitada ou inexistente, foi essencial adotar metodologias e notações que possibilitassem a modelagem precisa dos processos e a consistência operacional do sistema.

A utilização da BPMN (*Business Process Model and Notation*) permitiu mapear e documentar os fluxos de comunicação interna da escola, como: envio de mensagens entre professores e alunos, compartilhamento de avisos administrativos e notificações de atividades. A notação gráfica da BPMN garante que tais processos sejam visualmente compreensíveis, facilitando o entendimento tanto para profissionais da tecnologia quanto para os usuários finais (OMG, 2014).

Além disso, a UML (*Unified Modeling Language*) foi aplicada na definição da arquitetura do S.C.O.R.E, especialmente nos diagramas de casos de uso, classes e sequência. Essa modelagem padronizada proporcionou uma visão detalhada de como os usuários interagem com o sistema, descrevendo atores (alunos, professores, coordenação), casos de uso (enviar mensagem, consultar aviso, atualizar mural *offline*) e os relacionamentos entre módulos. Com isso, a UML reforçou a clareza e a documentação técnica, fundamentais para a manutenção e evolução do sistema (OMG, 2017).

Para viabilizar a sincronização de dados entre dispositivos, mesmo em ambientes *offline*, a especificação DDS (*Data Distribution Service*) pode ser explorada como modelo conceitual. Embora seja originalmente utilizada em sistemas de tempo real e IoT, o DDS fornece mecanismos de publicação-assinatura data-cêntrica que

permitem a troca de informações em redes locais sem depender diretamente da internet. Assim, quando houver reconexão à rede, o sistema poderá realizar a sincronização de dados de forma confiável e escalável (OMG, 2015).

A aplicação desses padrões garante que o S.C.O.R.E seja um sistema bem estruturado, adaptável e sustentável, mantendo a clareza nos fluxos de processos (BPMN), a documentação técnica padronizada (UML) e a capacidade de sincronização *offline-para-online* (DDS). Dessa forma, o projeto não apenas atende às necessidades atuais da instituição escolar, mas também se encontra preparado para futuras expansões, como a integração com bibliotecas digitais, sistemas de gestão acadêmica ou módulos de aprendizagem interativa.

3.2 BPMN

BPMN significa *Business Process Model and Notation* (Modelo e Notação de Processos de Negócio) e corresponde a um padrão internacional usado para representar processos empresariais por meio de símbolos gráficos e regras específicas. Ele funciona como uma linguagem visual comum que possibilita a descrição, análise e melhoria de fluxos de trabalho, tornando a comunicação mais clara entre analistas, profissionais técnicos e usuários do negócio, com o objetivo de aperfeiçoar as operações organizacionais.

Conforme dito anteriormente, a criação do BPMN foi impulsionada *pelo Object Management Group* (OMG), uma organização sem fins lucrativos que desenvolve e mantém padrões abertos para a computação e tecnologia. O desenvolvimento do BPMN começou em 2001, quando a OMG percebeu a necessidade de uma notação padronizada que pudesse ser amplamente adotada para modelar processos de negócio. (SANKHYA, 2023)

O BPMN, surgiu da necessidade de unificar a maneira como as organizações representavam seus processos de negócio. Durante muito tempo, cada empresa utilizava métodos diferentes para descrever suas atividades: alguns recorriam a diagramas de atividades da UML, outros a cadeias de processos baseadas em eventos (EPCs) ou ainda a notações próprias. Essa variedade criava um problema evidente: enquanto os profissionais de tecnologia conseguiam compreender

determinados modelos, os gestores e analistas de negócio, muitas vezes, não tinham clareza sobre o que estava sendo representado. Faltava uma linguagem comum que facilitasse o diálogo entre todos os envolvidos.

Foi nesse contexto, no início dos anos 2000, que especialistas e empresas começaram a trabalhar na ideia de criar uma notação gráfica padronizada, que fosse simples o bastante para que qualquer usuário de negócio pudesse entender, mas ao mesmo tempo precisa e robusta para que desenvolvedores pudessem utilizá-la na construção de soluções tecnológicas. Essa busca resultou no BPMN, que rapidamente ganhou espaço por oferecer algo que até então não existia: uma forma de representar processos que funcionava como uma verdadeira “ponte” entre áreas técnicas e áreas de gestão.

Com símbolos intuitivos e regras claras, o BPMN possibilitou a visualização de fluxos de trabalho complexos de maneira compreensível para todos, transformando-se em uma ferramenta poderosa de documentação, análise e otimização dos processos. A sua adoção global consolidou-se ao longo dos anos, acompanhando as transformações do mundo empresarial e tornando-se peça-chave na era da transformação digital. Hoje, o BPMN é visto não apenas como um padrão de modelagem, mas como um facilitador essencial da comunicação e da integração entre pessoas, negócios e tecnologia. (IBM, 2024).

Versões:

Versão 1.0 (2004):

A primeira versão oficial da BPMN foi lançada em maio de 2004 pela BPMI. Esta versão inicial incluía uma notação básica para modelar processos de negócios, focando em fornecer uma linguagem clara e compreensível tanto para analistas de negócios quanto para técnicos. A BPMN 1.0 foi bem recebida pela comunidade de modelagem por sua clareza e utilidade. (TRISOTECH, 2024)

Versão 1.1 (2006):

Em fevereiro de 2006, após a fusão da BPMI com o OMG, foi lançada a versão 1.1. Esta atualização introduziu melhorias e correções na notação, refinando conceitos

e símbolos para melhorar a precisão e a usabilidade da modelagem de processos. (VISUAL PARADIGM, 2024)

Versão2.0(2011):

Desenvolvida pela OMG e lançada oficialmente em janeiro de 2011, a BPMN 2.0 trouxe várias melhorias significativas, como a formalização da semântica de execução para todos os elementos, a introdução de mecanismos de extensibilidade, o refinamento da definição e correlação de eventos, a ampliação das interações humanas dentro dos processos, a inclusão de um modelo de coreografia para representar interações entre diferentes participantes e a adição de uma visão de conversação para diagramas de colaboração. (OMG, 2024)

Versão2.0.2(2013):

Publicada como padrão ISO/IEC 19510:2013 em julho de 2013, a versão 2.0.2 trouxe ajustes menores e refinamentos na notação, garantindo maior compatibilidade com outras normas e práticas de modelagem de processos de negócios. (ISO, 2024)

versão1.0(2004):

A primeira versão oficial da BPMN foi lançada em maio de 2004 pela BPMI. Esta versão inicial incluía uma notação básica para modelar processos de negócios, focando em fornecer uma linguagem clara e compreensível tanto para analistas de negócios quanto para técnicos. A BPMN 1.0 foi bem recebida pela comunidade de modelagem por sua clareza e utilidade. (TRISOTECH, 2024)

Versão1.1(2006):

Em fevereiro de 2006, após a fusão da BPMI com o OMG, foi lançada a versão 1.1. Esta atualização introduziu melhorias e correções na notação, refinando conceitos e símbolos para melhorar a precisão e a usabilidade da modelagem de processos. (VISUAL PARADIGM, 2024)

Versão2.0(2011):

Desenvolvida pela OMG e lançada oficialmente em janeiro de 2011, a BPMN 2.0 trouxe várias melhorias significativas, como a formalização da semântica de execução para todos os elementos, a introdução de mecanismos de extensibilidade, o refinamento da definição e correlação de eventos, a ampliação das interações

humanas dentro dos processos, a inclusão de um modelo de coreografia para representar interações entre diferentes participantes e a adição de uma visão de conversação para diagramas de colaboração. (OMG, 2024)

Versão 2.0.2 (2013):

Publicada como padrão ISO/IEC 19510:2013 em julho de 2013, a versão 2.0.2 trouxe ajustes menores e refinamentos na notação, garantindo maior compatibilidade com outras normas e práticas de modelagem de processos de negócios. (ISO, 2024)

3.2.1 Implementação no Sistema

Em um aplicativo de comunicação *offline* para escolas, a notação BPMN pode ser usada para organizar, automatizar e aprimorar continuamente os processos de envio de mensagens entre alunos, professores e coordenadores, oferecendo uma visão clara de cada etapa, desde a criação da mensagem até sua entrega. Ao representar o fluxo, é possível identificar e melhorar atividades como redigir a mensagem, transmitir pelos dispositivos conectados, receber confirmações de leitura e atualizar o histórico de comunicação, o que ajuda a reduzir erros, evitar perdas de informações e aumentar a eficiência do sistema. Além disso, o acompanhamento dos processos permite detectar pontos críticos e implementar melhorias constantes, tornando a comunicação mais confiável e adaptada às necessidades dos usuários. Dessa forma, o uso da BPMN contribui para tornar o aplicativo mais funcional, rápido e eficiente, proporcionando uma experiência melhor para toda a comunidade escolar.

Modelagem de Processos de Negócio: O BPMN permite que os processos do aplicativo escolar (como envio de mensagens, confirmação de leitura, sincronização entre dispositivos e notificações de atividades) sejam representados de forma clara e compreensível. Cada etapa do fluxo de comunicação pode ser visualizada em um diagrama, facilitando a análise e o aprimoramento dos processos.

Automatização e Implementação: Depois de modelados, os processos BPMN podem servir como base para automação. Por exemplo, um fluxo modelado pode ser integrado ao sistema do aplicativo para automatizar tarefas como envio de notificações, atualização automática do histórico de mensagens e sincronização entre dispositivos *offline*.

Monitoramento e Melhoria Contínua: O BPMN também é útil para acompanhar a eficiência dos processos ao longo do tempo. Com base nos diagramas, a equipe de desenvolvimento e gestão pode identificar gargalos ou falhas na comunicação e implementar melhorias, garantindo que a troca de mensagens seja mais rápida, confiável e eficiente.

3.2.2 Exemplo de aplicação

Envio de Mensagens: Um aluno ou professor envia uma mensagem através do aplicativo. O processo BPMN modela todas as etapas, desde a criação da mensagem, transmissão pelo sistema *offline*, recebimento pelo destinatário, até a confirmação de leitura.

Sincronização de Dispositivos: Um diagrama BPMN pode representar o fluxo de sincronização entre os dispositivos conectados, definindo quando e como as mensagens e dados são atualizados, garantindo que todos os usuários tenham as informações mais recentes, mesmo sem conexão constante à *internet*.

Notificações e Alertas: É possível modelar o fluxo de envio de notificações, desde a geração do aviso (como lembrete de atividade ou aviso da escola), até a entrega e registro da notificação no dispositivo do usuário, garantindo que a comunicação seja eficiente e registrada corretamente.

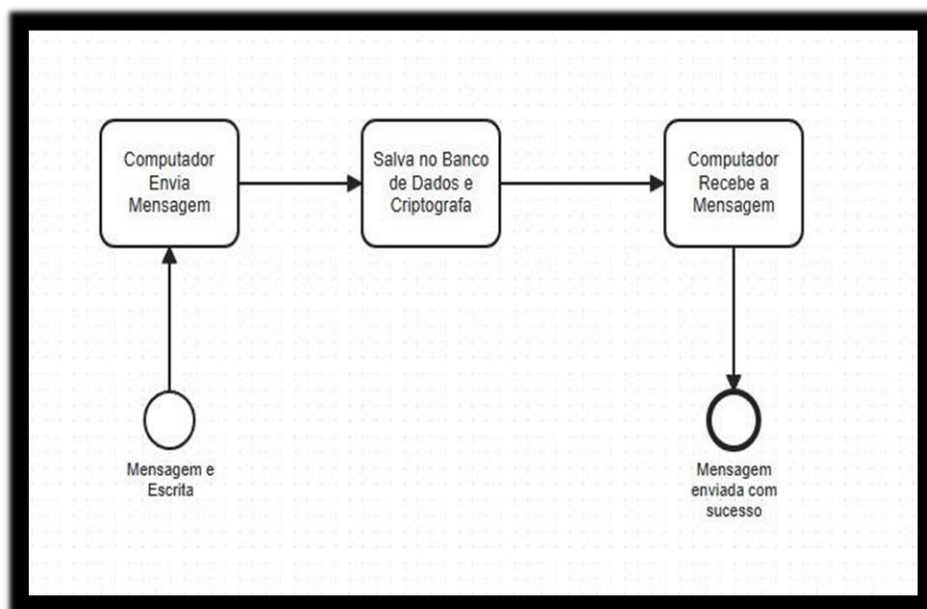
3.3 DIAGRAMA

Diagrama é um desenho simplificado mostrando a aparência, estrutura ou funcionamento de algo, ou seja, uma representação esquemática. Os diagramas são utilizados em diversas áreas de estudos acadêmicos e dentro das empresas para simplificar estruturas ou grandes conjuntos de informações. Diagramas possuem diversas formas como círculos, retângulos, linhas e setas que juntas representam uma sequência que pode ser interpretada de forma rápida. A utilização dessas formas facilita a interpretação de quem está analisando o diagrama, economizando tempo e proporcionando um cenário mais seguro para tomada de decisões. (GO PRATICO, 2025)

3.3.1 Diagrama de envio de mensagem

O diagrama representa o envio de uma mensagem entre dois computadores. O processo se inicia quando um computador escreve e envia uma mensagem. Essa mensagem é salva no banco de dados e criptografada, garantindo a segurança da informação. Em seguida, outro computador recebe a mensagem. Ao final do processo, é exibida uma confirmação de que a mensagem foi enviada com sucesso.

Figura 18 - Diagrama Envio de Mensagem



FONTE – AUTORIA PRÓPRIA (2025)

Início (Círculo)

O processo se inicia no lado esquerdo com o símbolo de um círculo, representando o momento em que uma mensagem é escrita em um computador. Esse é o ponto de entrada do sistema, onde a mensagem é preparada para envio.

Computador Envia Mensagem (Retângulo)

Após a mensagem ser escrita, o computador executa a ação de enviar a mensagem. Esta etapa é representada por um retângulo, simbolizando uma ação/processo.

Salva no Banco de Dados e Criptografa (Retângulo)

A mensagem enviada é então armazenada em um banco de dados e criptografada. Essa etapa garante a segurança e a integridade da mensagem, impedindo acessos não autorizados. A criptografia é uma medida fundamental para proteger os dados durante o trânsito e o armazenamento.

Computador Recebe a Mensagem (Retângulo)

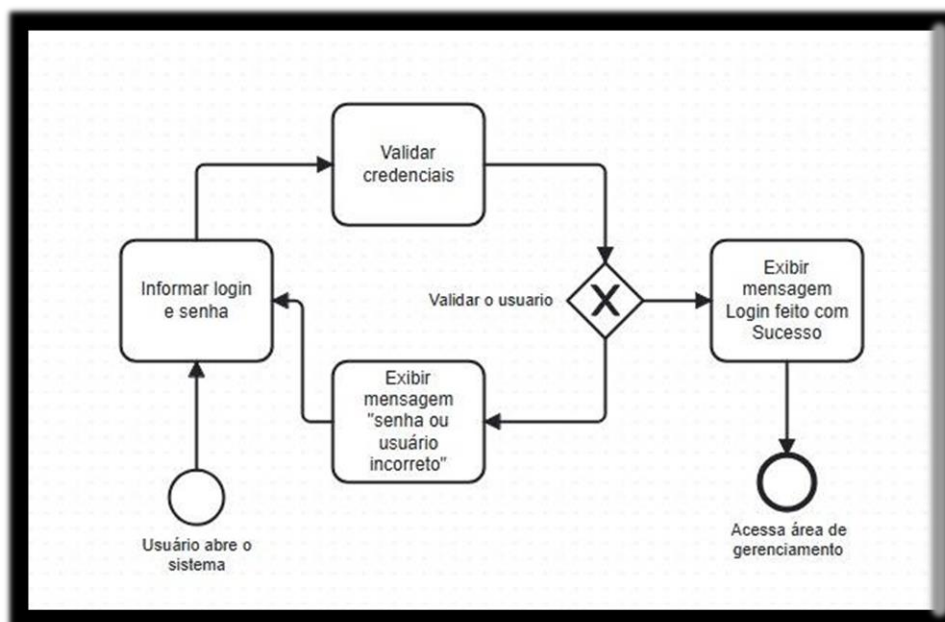
Após ser criptografada e salva, a mensagem é disponibilizada para o computador destinatário, que recebe a mensagem. Isso marca a conclusão do envio.

Mensagem Enviada com Sucesso (Círculo)

Por fim, o processo é encerrado com um círculo, que indica que a mensagem foi enviada com sucesso. Esse ponto confirma que todas as etapas anteriores foram executadas corretamente, desde o envio até o recebimento.

3.3.2 – Diagrama *Login*

Figura 19 - Diagrama *Login*



FONTE – AUTORIA PRÓPIA (2025)

O diagrama mostra o fluxo para autenticação de usuários. O processo começa quando o usuário acessa o sistema e insere suas credenciais (*login* e senha). Em seguida, o sistema valida essas informações. Caso os dados estejam corretos, o usuário recebe uma mensagem de *login* bem-sucedido e é redirecionado à área de gerenciamento. Caso contrário, é exibida uma mensagem informando que o *login* ou a senha estão incorretos, e o usuário é direcionado novamente para a tela de *login*.

Início (Círculo)

O processo se inicia com o usuário abrindo o sistema, representado pelo círculo no canto inferior esquerdo. Este é o ponto de entrada do usuário na aplicação, onde ele se depara com a interface de *login*.

Informar *Login* e Senha (Retângulo)

Assim que o sistema é aberto, o usuário é solicitado a informar seu *login* e senha. Esta etapa é essencial para verificar se o usuário possui permissão de acesso ao sistema. O campo de *login* e senha permite a autenticação baseada em credenciais.

Validar Credenciais (Retângulo)

Após o envio dos dados, o sistema executa a validação das credenciais inseridas. Neste momento, ocorre a verificação do nome de usuário e da senha, normalmente comparando-os com os dados armazenados em um banco de dados seguro.

Decisão: Validar o Usuário (Losango com "X")

Em seguida, o fluxo chega a uma etapa de decisão, onde o sistema avalia se as credenciais estão corretas:

Se corretas, o fluxo segue para o *login* bem-sucedido.

Se incorretas, o usuário é informado sobre o erro.

Exibir Mensagem "Senha ou Usuário Incorreto" (Retângulo)

Caso os dados fornecidos estejam errados, o sistema exibe uma mensagem de erro, alertando que o "usuário ou senha estão incorretos". O usuário, então, é redirecionado novamente para a tela de *login* para tentar novamente.

Exibir Mensagem "Login Feito com Sucesso" (Retângulo)

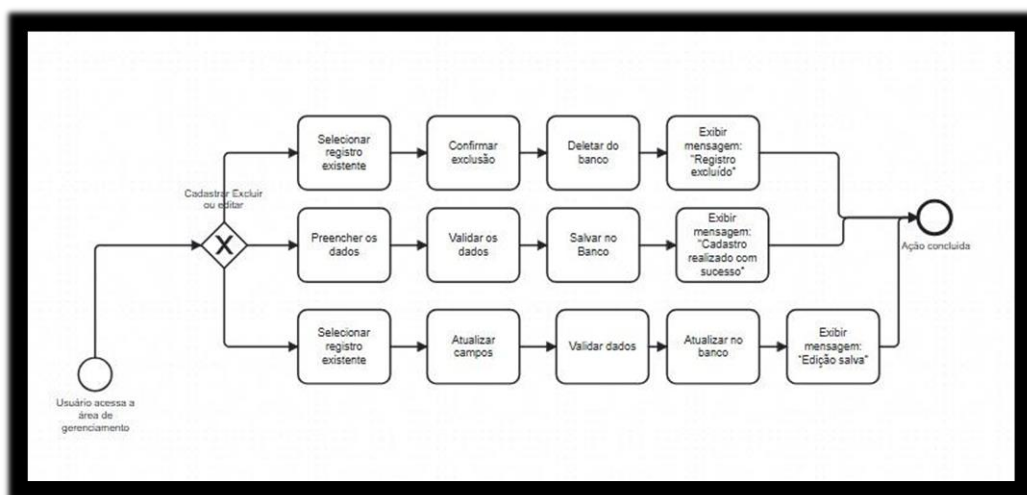
Se as credenciais forem validadas com sucesso, o sistema confirma o sucesso do login, exibindo uma mensagem de boas-vindas ou de confirmação: "Login feito com sucesso".

Acesso à Área de Gerenciamento (Círculo)

Por fim, com o *login* validado, o usuário acessa a área de gerenciamento do sistema, onde poderá executar as funções disponíveis conforme seu perfil de acesso. Este círculo marca o fim do processo de autenticação e o início do uso do sistema.

3.3.3 – Diagrama Gerenciamento de Dados

Figura 20 - Diagrama Gerenciamento de Dados



FONTE – AUTORIA PRÓPRIA (2025)

O diagrama mostra o gerenciamento de dados no sistema, contendo as ações de cadastro, edição e exclusão dos registros. Quando o usuário acessa a área de gerenciamento, pode optar por:

Cadastrar um novo registro: o usuário preenche os dados, o sistema valida as informações, salva no banco e exibe uma mensagem de sucesso.

Editar um registro existente: o usuário seleciona o registro, atualiza os campos, o sistema valida os dados, atualiza no banco e exibe a mensagem de edição salva.

Excluir um registro: o usuário seleciona o registro, confirma a exclusão, o sistema deleta do banco e exibe uma mensagem indicando que o registro foi excluído.

Início (Círculo)

O processo se inicia quando o usuário acessa a área de gerenciamento do sistema, representado pelo círculo à esquerda. Neste ponto, o usuário tem a possibilidade de realizar ações de cadastro, edição ou exclusão de registros.

Decisão Inicial: Cadastrar, Editar ou Excluir (Losango com "X")

Após acessar a área de gerenciamento, o usuário se depara com uma decisão sobre qual ação deseja executar:

Cadastrar novo registro

Editar registro existente

Excluir registro

Caminho 1: Exclusão de Registro

Selecionar registro existente (Retângulo)

O usuário escolhe o item que deseja remover do sistema.

Confirmar exclusão (Retângulo)

O sistema solicita a confirmação da ação de exclusão, evitando remoções acidentais.

Deletar do banco (Retângulo)

Após a confirmação, o registro é removido do banco de dados.

Exibir mensagem: "Registro excluído" (Retângulo)

O sistema informa que a exclusão foi realizada com sucesso.

Ação concluída (Círculo)

O fluxo é finalizado com o círculo à direita, indicando o término da operação.

Caminho 2: Cadastro de Novo Registro

Preencher os dados (Retângulo)

O usuário insere as informações necessárias para o novo cadastro.

Validar os dados (Retângulo)

O sistema verifica se os campos foram preenchidos corretamente (formato, obrigatoriedade, etc.).

Salvar no banco (Retângulo)

Os dados validados são armazenados no banco de dados.

Exibir mensagem: "Cadastro realizado com sucesso" (Retângulo)

O sistema confirma que o novo registro foi salvo com êxito.

Ação concluída (Círculo)

O fluxo termina com sucesso.

Caminho 3: Edição de Registro

Selecionar registro existente (Retângulo)

O usuário escolhe qual item deseja modificar.

Atualizar campos (Retângulo)

Os dados são editados conforme a necessidade do usuário.

Validar dados (Retângulo)

O sistema realiza nova validação dos campos atualizados.

Atualizar no banco (Retângulo)

As alterações são salvas no banco de dados, substituindo os dados anteriores.

Exibir mensagem: "Edição salva" (Retângulo)

O sistema confirma que a atualização foi realizada com sucesso.

Ação concluída (Círculo)

O processo de edição é encerrado com sucesso.

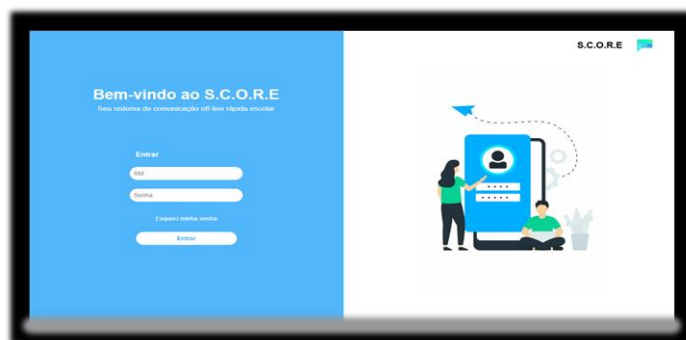
3.4- TELAS

As páginas do site serão exibidas e descritas de acordo com suas funcionalidades. Cada uma desempenha um papel específico dentro do sistema, que será explicado a partir dos códigos desenvolvidos para atender às demandas do projeto. As ilustrações das telas permitem que os usuários compreendam melhor sua construção e modo de operação. Além disso, essa avaliação evidencia o processo de desenvolvimento e a lógica aplicada, tornando mais simples entender como cada recurso foi planejado e implementado para funcionar adequadamente. Dessa forma, os usuários passam a ter uma percepção mais clara sobre o que foi elaborado.

3.4.1 Tela de *Login*

A imagem 21 mostra a tela inicial, que é a mesma para todos os usuários. Essa tela possui uma mensagem de “Bem-vindo ao S.C.O.R.E”, e mostra os campos de “login”, “senha” e um botão de entrar

Figura 21 - Tela de *login*



FONTE – AUTORIA PRÓPRIA (2025)

A Figura X apresenta a interface inicial do sistema S.C.O.R.E., composta por uma estrutura equilibrada entre elementos funcionais e visuais. O código *HTML* exibido define a organização dessa página de login, que representa a porta de entrada do sistema e permite a autenticação dos usuários cadastrados.

Início da Estrutura O documento inicia com a declaração `<!DOCTYPE html>` e a definição do idioma em `<html lang="pt-br">`, assegurando compatibilidade e acessibilidade. No cabeçalho, as *tags* `<meta>` configuram a codificação de caracteres (*UTF-8*) e a adaptação da página a diferentes tamanhos de tela por meio da *viewport*. Em seguida, o arquivo *style.css* é vinculado para o controle da aparência visual, enquanto o script *index.js* é carregado de forma assíncrona (*defer*), responsável pela interação dinâmica com o usuário.

Estrutura Principal: No corpo da página (`<body>`), encontra-se a *div* principal com a classe *container*, que agrupa toda a interface dividida em três seções principais: *left-side*, *header* e *right-side*. Essa separação garante uma disposição clara entre os elementos de interação e os gráficos informativos.

Seção Esquerda – Formulário de Login: O lado esquerdo (*left-side*) contém a área de boas-vindas e o formulário de acesso. Dentro da *div* com a classe *welcome-text*, o título “Bem-vindo ao S.C.O.R.E” é apresentado em destaque por meio da classe *fade-in*, que aplica um efeito de transição visual. Logo abaixo, o subtítulo “Seu sistema de comunicação off-line rápida escolar” reforça o propósito da aplicação.

O formulário de autenticação é definido pela tag `<form>` com a classe *login-form* e o identificador *formLogin*. Ele agrupa os campos de entrada, mensagens de erro e o botão de ação. O campo de “RM” é implementado com `<input type="number">` e possui *placeholder* “RM”, sendo obrigatório (*required*). O campo de senha utiliza `<input type="password">` e também requer preenchimento. Ambos os campos contam com ícones de validação — *success-icon* e *error-icon* — e mensagens de erro específicas, garantindo feedback visual imediato ao usuário.

Logo abaixo, encontra-se o link `<a>` “Esqueci minha senha”, direcionando à página *esqueci_senha.html*, que oferece suporte à recuperação de credenciais. O botão `<button type="button" id="btnLogin">Entrar</button>` é o responsável por acionar o processo de validação dos dados via *JavaScript*, sem recarregar a página.

Cabeçalho e Identidade Visual: A seção *header* apresenta o nome do sistema (“S.C.O.R.E”) e o logotipo institucional (*logo.jpg*), reforçando a identidade visual e a credibilidade da aplicação. Essa área é fixada na parte superior direita, mantendo-se visível e reconhecível em toda a interface.

Seção Direita – Elemento Ilustrativo: Por fim, o bloco *right-side* contém uma imagem vetorial (*login.png*), que representa dois personagens interagindo com um dispositivo móvel. Esse elemento cumpre função estética e comunicativa, ilustrando o conceito de conexão e autenticação digital de forma intuitiva e moderna. O código cria uma interface de login funcional e esteticamente agradável, unindo HTML, CSS e JavaScript. Essa combinação proporciona uma experiência de uso fluida, interativa e acessível ao usuário.

Figura 22 - Código Tela de *Login*

```

<!DOCTYPE html>
<html lang="pt-br">
  <head>
    <meta charset="UTF-8" />
    <title>Home - S.C.O.R.E</title>
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <link rel="stylesheet" href="style.css">
    <script src="index.js" defer //script>
  </head>
  <body>
    <div class="container">
      <div class="left-side">
        <div class="welcome-text">
          <h1 class="fade-in">Bem-vindo ao S.C.O.R.E</h1>
          <p class="subtitle">Seu sistema de comunicação off-line rápida escolar</p>
        </div>
        <div class="login-form" id="formLogin">
          <p class="login-title">Entrar</p>
          <div class="input-group" id="userGroup">
            <input type="text" id="email" placeholder="Email" required />
            <span class="icon success-icon">✓</span>
            <span class="icon error-icon">X</span>
            <div class="error-message">Por favor, preencha o email.</div>
          </div>
          <div class="input-group" id="senhaGroup">
            <input type="password" id="senha" placeholder="Senha" required />
            <span class="icon success-icon">✓</span>
            <span class="icon error-icon">X</span>
            <div class="error-message">Por favor, preencha a senha.</div>
          </div>
          <span id="mensagem"></span>
          <a href="esqueci-senha.html" class="forgot-password">Esqueci minha senha</a> <!-- novo link -->
          <button type="button" id="btnLogin">Entrar</button>
        </div>
      </div>
      <div class="header">
        <span>S.C.O.R.E</span>
        
      </div>
      <div class="right-side">
        
      </div>
    </div>
  </body>
</html>

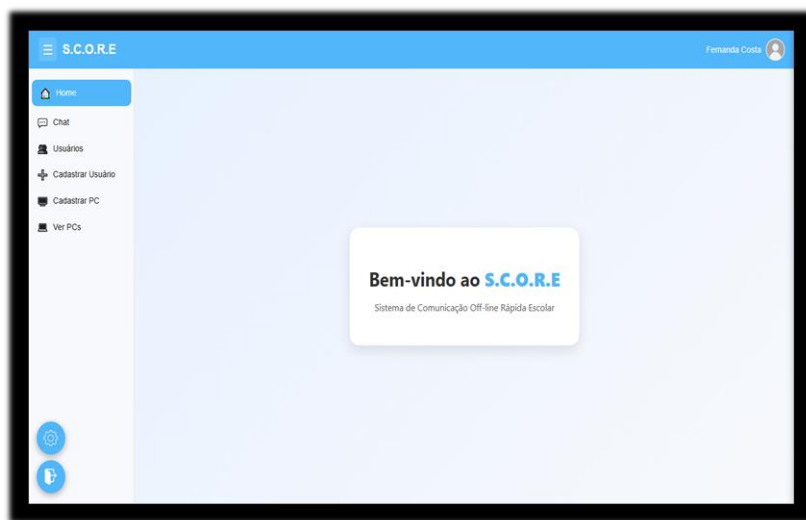
```

FONTE – AUTORIA PRÓPRIA (2025)

3.4.2 Tela inicial do sistema S.C.O.R.E após *login* da secretaria

A imagem 23 mostra a tela inicial do sistema com o login da secretaria. Essa tela possui campos onde é possível verificar os usuários, os computadores e cadastrar facilitando também o acesso ao *chat*

Figura 23 - Tela inicial do sistema S.C.O.R.E após login da secretaria



FONTE – AUTORIA PRÓPRIA (2025)

A Figura 23 apresenta a tela inicial do sistema S.C.O.R.E., exibida logo após o acesso realizado com sucesso por um usuário da secretaria. O código *HTML* é responsável por construir uma interface simples, moderna e acolhedora, que dá as boas-vindas ao usuário, reforçando a identidade visual e institucional do sistema.

Estrutura e Layout: O documento inicia com a declaração `<!DOCTYPE html>` e a definição do idioma da página em `<html lang="pt-br">`, garantindo compatibilidade e acessibilidade. O corpo da página (`<body>`) é estilizado com CSS para exibir um fundo em *degradê* (*linear-gradient*) nos tons azul e branco, criando um visual leve e harmônico. O conteúdo central é posicionado de forma responsiva e centralizada vertical e horizontalmente na tela, utilizando as propriedades *flexbox* (`display: flex; align-items: center; justify-content: center;`).

Conteúdo Principal: O elemento principal da página é uma *div* com a classe *container*, que contém o título e a descrição do sistema. Esse contêiner é apresentado sobre um painel branco com bordas arredondadas e *sombra projetada* (*box-shadow*), conferindo destaque e profundidade à área de boas-vindas.

Título e Identidade Visual: Dentro do contêiner, a tag `<h2>` exibe a mensagem “Bem-vindo ao S.C.O.R.E”, com o nome do sistema em destaque por meio da tag

, colorida em azul (#51b6fa). Essa combinação reforça a identidade visual e mantém a consistência com o cabeçalho do sistema principal.

Mensagem Descritiva: Logo abaixo, a tag <p> apresenta o texto “Sistema de Comunicação *Off-line* Rápida Escolar”, que define o propósito da aplicação. Esse elemento utiliza um efeito de *animação tipográfica (typing animation)* desenvolvido com *CSS keyframes*, simulando o movimento de digitação e piscagem de cursor, o que confere dinamismo e modernidade à interface.

Animações e Efeitos Visuais: O código utiliza múltiplas animações em CSS — *fadeIn*, *slideIn*, *typing* e *blink* — aplicadas ao título e à mensagem principal. Esses efeitos criam uma transição suave na entrada dos elementos e reforçam a sensação de movimento e interatividade. Além disso, a aplicação de *box-shadow* e *border-radius* contribui para um design limpo e agradável.

Em resumo, a tela inicial do S.C.O.R.E. funciona como uma página de boas-vindas elegante e responsiva, construída integralmente em *HTML* e *CSS*. Ela combina cores suaves, animações sutis e uma disposição equilibrada dos elementos para proporcionar uma experiência visual acolhedora e intuitiva ao usuário recém-autenticado.

Figura 24 - Código Tela Inicial 1

```

10 <div class="input-group">
11 <select name="tipo" id="tipo" required>
12 <option value="">Selecione o tipo</option>
13 <option value="Secretaria">Secretaria</option>
14 <option value="Professor">Professor</option>
15 </select>
16 </div>
17
18 <div class="input-group">
19 <input type="email" name="email" id="email" placeholder="E-mail" required>
20 </div>
21
22 <div class="input-group">
23 <input type="email" name="confirmar_email" id="confirmar_email" placeholder="Confirmar E-mail" required>
24 <span class="error-message">E-mails não coincidem.</span>
25 </div>
26
27 <div class="input-group">
28 <input type="password" name="senha" id="senha" placeholder="Senha" required>
29 </div>
30
31 <div class="input-group">
32 <input type="password" name="confirmar_senha" id="confirmar_senha" placeholder="Confirmar Senha" required>
33 <span class="error-message">Senhas não coincidem.</span>
34 </div>
35
36 <button type="submit" class="btn btn-primary">Cadastrar</button>
37 </form>
38
39 <!-- Botão voltar para página anterior -->
40 <a href="#" class="btn btn-link" title="Voltar">
41 <img alt="Ícone de voltar" data-bbox="252 639 848 874" style="vertical-align: middle; height: 1em; margin-right: 0.5em;"/>
42 <span>Voltar</span>
43 </a>
44
45 <a href="#" class="btn btn-link" title="Ajuda / Tutorial">
46 <img alt="Ícone de ajuda" data-bbox="252 639 848 874" style="vertical-align: middle; height: 1em; margin-right: 0.5em;"/>
47 <span>Ajuda / Tutorial</span>
48 </a>
49
50 <div class="footer">
51 <small>© 2025 S.C.O.R.E. Todos os direitos reservados.
52 </small>
53 </div>

```

FONTE – AUTORIA PRÓPRIA (2025)

Figura 27 - Tela Lista de Usuários



RM	NOME	CPF	TIPO	EMAIL	SENHA	AÇÕES
101	Ana Maria	12345678901	Professor	ana.maria@example.com	senha123	Editar Excluir
102	Carlos Silva	98765432100	Professor	carlos.silva@example.com	senha456	Editar Excluir
103	Fernanda Costa	32165498700	Secretaria	fernanda.costa@example.com	senha789	Editar Excluir
108	pedro	49750033841	Professor	valeriojunior345@gmail.com	123123	Editar Excluir

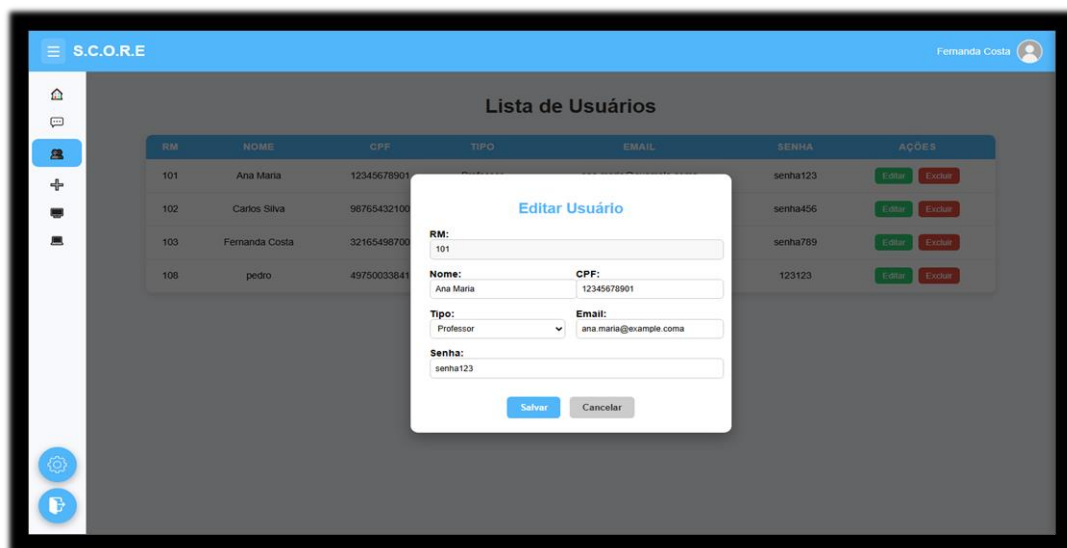
FONTE – AUTORIA PRÓPRIA (2025)

A Figura 27 apresenta a tela de gerenciamento de usuários do sistema S.C.O.R.E, identificada pelo título “Lista de Usuários”. Essa seção tem como principal objetivo possibilitar o controle e a administração dos perfis cadastrados na plataforma. No centro da tela, encontra-se uma tabela organizada e de fácil leitura, composta pelas colunas “RM”, “Nome”, “CPF”, “Tipo”, “E-mail”, “Senha” e “Ações”.

Cada linha da tabela corresponde a um usuário previamente cadastrado no sistema, exibindo de forma clara suas informações básicas. Na coluna “Ações”, estão disponíveis dois botões interativos: o botão “Editar”, destacado na cor verde, e o botão “Excluir”, apresentado na cor vermelha. O primeiro permite realizar alterações nos dados do usuário, enquanto o segundo oferece a opção de remover permanentemente o registro.

Além disso, a interface mantém o padrão visual característico do sistema, com uma barra superior azul que reforça a identidade visual da aplicação e um menu lateral à esquerda, onde estão localizadas as demais opções de navegação. Essa disposição contribui para uma experiência de uso intuitiva e organizada, facilitando o acesso às funcionalidades e garantindo consistência visual em todas as telas do sistema.

Figura 28 Tela de edição de usuário



FONTE – AUTORIA PRÓPRIA (2025)

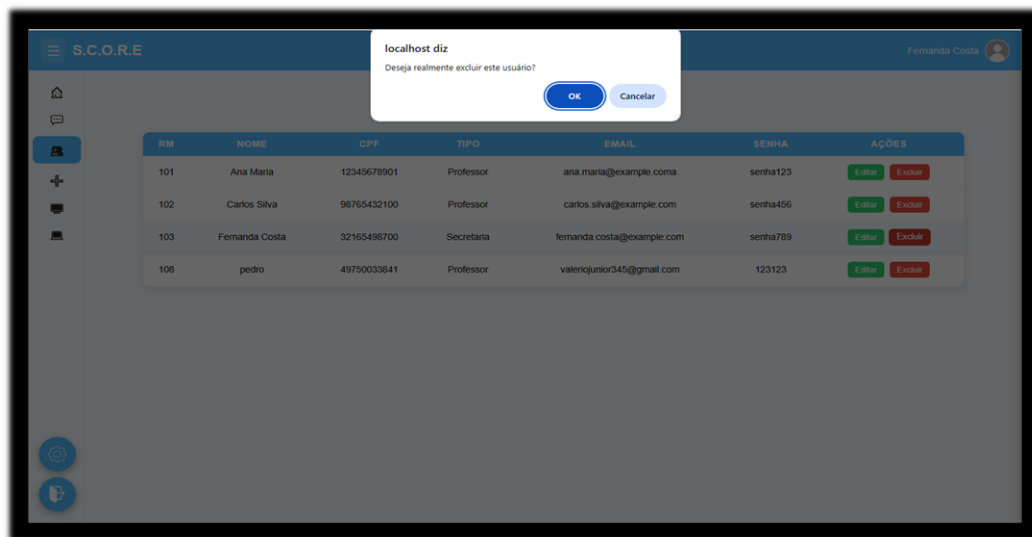
A Figura 28 ilustra a funcionalidade de edição de dados de um usuário no sistema S.C.O.R.E. Nessa etapa, é possível realizar modificações nas informações previamente cadastradas, garantindo a atualização correta dos registros. Ao acionar a opção de edição, surge um *pop-up* centralizado na tela, identificado pelo título “Editar Usuário”, que se destaca visualmente em relação ao restante da interface.

Esse *pop-up* exibe campos de formulário preenchidos automaticamente com os dados atuais do usuário selecionado, como “RM”, “Nome”, “CPF”, “Tipo”, “E-mail” e “Senha”, permitindo que o operador visualize e altere apenas as informações desejadas sem a necessidade de redigitar todos os dados. Essa abordagem torna o processo de edição mais rápido, eficiente e intuitivo, além de minimizar erros durante a atualização dos registros.

Na parte inferior da janela, encontram-se dois botões de ação: “Salvar”, responsável por confirmar e aplicar as alterações realizadas, e “Cancelar”, que possibilita desistir da modificação e retornar à tela anterior sem efetuar mudanças. Enquanto o *pop-up* está ativo, o restante da interface permanece escurecido, criando um efeito de destaque visual que direciona a atenção do usuário exclusivamente para o formulário de edição. Essa característica contribui para uma experiência de uso mais

focada, organizada e agradável, reforçando a identidade visual e a usabilidade do sistema.

Figura 29 – Tela de Confirmação de exclusão de usuário



FONTE – AUTORIA PRÓPRIA (2025)

A Figura 29 exibe a caixa de diálogo gerada ao clicar em “Excluir”. O *pop-up* apresenta a mensagem “Deseja realmente excluir este usuário?” e oferece duas opções: “OK” (para confirmar a exclusão) e “Cancelar” (para interromper o processo). Esse recurso de confirmação evita exclusões acidentais e garante maior segurança na gestão de cadastros dentro do sistema.

Na parte inicial, o *script PHP* realiza a abertura da sessão com *session_start()* e verifica se a variável `$_SESSION['rm']` está definida, garantindo o acesso apenas a usuários autenticados. Em seguida, é feita a conexão com o banco de dados por meio do arquivo *conexao.php* e executada uma consulta SQL (*SELECT RM, Nome, CPF, Tipo, Email, Senha FROM usuario*), que retorna todos os cadastros armazenados na tabela *usuario*.

A camada de apresentação, escrita em *HTML*, estrutura a página com o título “Lista de Usuários” e uma tabela centralizada que contém as colunas “RM”, “Nome”, “CPF”, “Tipo”, “Email”, “Senha” e “Ações”. Cada linha (*<tr>*) é gerada dinamicamente

pelo *PHP* e exibe os dados de um usuário, além de dois botões de controle: *Editar* (em verde) e *excluir* (em vermelho). O design da interface é definido por regras de estilo internas (`<style>`), que aplicam tipografia *sans-serif*, bordas arredondadas, sombreamento leve e destaque em azul para o cabeçalho da tabela, garantindo um *layout* limpo e responsivo.

O comportamento interativo da página é controlado por *JavaScript*. O trecho dedicado à exclusão de registros utiliza o método *confirm()* para exibir uma caixa de diálogo com a mensagem “Deseja realmente excluir este usuário?”. Caso o usuário confirme, uma requisição *fetch()* do tipo *POST* é enviada para o arquivo *excluir.php*, transmitindo o *RM* correspondente. Após o retorno da resposta, a linha é removida dinamicamente da tabela, sem recarregar a página.

A funcionalidade de edição é implementada de forma dinâmica, por meio da criação de um *modal* em tempo real. Ao clicar no botão *Editar*, o *JavaScript* coleta os dados da linha selecionada (como *Nome*, *CPF*, *Tipo*, *Email* e *Senha*) e gera um formulário dentro de uma *div* sobreposta à interface principal. Esse *pop-up*, estilizado com cores claras e efeito de escurecimento do fundo, contém campos pré-preenchidos e os botões *Salvar* e *Cancelar*.

Quando o formulário é submetido, os dados são enviados via *fetch()* para o *script editar.php*, que atualiza as informações no banco de dados. Em seguida, os valores na tabela são atualizados em tempo real, refletindo a modificação sem necessidade de recarregar a página. O fechamento do *modal* pode ser feito clicando no botão *Cancelar* (*id="fecharModal"*) ou após a conclusão da operação de salvamento.

Por fim, o uso de animações CSS (`@keyframes fadeIn` e `popIn`) adiciona uma transição suave à abertura da janela de edição, aprimorando a experiência visual. Assim, o código integra de forma eficiente a manipulação de dados com o banco de dados e a atualização dinâmica da interface, tornando o gerenciamento de usuários do sistema S.C.O.R.E. mais intuitivo e seguro.

Além disso, a aplicação demonstra boas práticas de design responsivo, garantindo que a interface se adapte a diferentes tamanhos de tela e dispositivos. O uso de feedback visual, como alterações sutis em cores e efeitos de foco, contribui

para uma navegação mais clara e agradável. A estrutura modular do código facilita manutenções e expansões, permitindo adicionar novas funcionalidades com menor esforço e integrando estética, usabilidade e desempenho.

Figura 30 - Código Tela Listas de Usuários 1

```

<?php
session_start();
require_once "conexao.php";

if (!isset($_SESSION['rm'])) {
    header("Location: ../index.html");
    exit;
}

$sql = "SELECT RM, Nome, CPF, Tipo, Email, Senha FROM usuario";
$stmt = $pdo->query($sql);
$usuarios = $stmt->fetchAll(PDO::FETCH_ASSOC);
?>
<!DOCTYPE html>
<html lang="pt-br">
<head>
<meta charset="UTF-8">
<title>Lista de Usuários - S.C.O.R.E./<title>
<meta name="viewport" content="width=device-width, initial-scale=1">
</head>
<body>
<h1>Lista de Usuários</h1>
<table>
<thead>
<tr>
<th>RM</th>
<th>Nome</th>
<th>CPF</th>
<th>Tipo</th>
<th>Email</th>
<th>Senha</th>
<th>Ações</th>
</tr>
</thead>
<tbody>
<?php foreach ($usuarios as $row) ?>
<tr>
<td data-rm="<?php htmlspecialchars($row['RM']) ?>">
<td><?php htmlspecialchars($row['Nome']) ?></td>
<td class="nome"><?php htmlspecialchars($row['Nome']) ?></td>
<td class="cpf"><?php htmlspecialchars($row['CPF']) ?></td>
<td class="tipo"><?php htmlspecialchars($row['Tipo']) ?></td>
<td class="email"><?php htmlspecialchars($row['Email']) ?></td>
<td class="senha"><?php htmlspecialchars($row['Senha']) ?></td>
<td>
<button class="btn-acao btn-editar">Editar</button>
<button class="btn-acao btn-excluir">Excluir</button>
</td>
</tr>
<?php endforeach ?>
</tbody>
</table>

```

FONTE – AUTORIA PRÓPRIA (2025)

Figura 31 - Código Tela Lista de Usuários 2

```

<?php
// == Excluir Usuário ==
document.querySelectorAll(" .btn-excluir").forEach(btn => {
    btn.addEventListener("click", () => {
        const tr = btn.closest("tr");
        const rm = tr.dataset.rm;

        if (confirm("Deseja realmente excluir este usuário?")) {
            fetch("excluir.php", {
                method: "POST",
                headers: { "Content-Type": "application/x-www-form-urlencoded" },
                body: "rm=" + encodeURIComponent(rm)
            })
                .then(res => res.text())
                .then(msg => {
                    alert(msg);
                    tr.remove();
                })
        }
    });
});

// == Editar Usuário (Modal) ==
document.querySelectorAll(" .btn-editar").forEach(btn => {
    btn.addEventListener("click", () => {
        const tr = tr.closest("tr");

        const rm = tr.dataset.rm;
        const nome = tr.querySelector(" .nome").textContent;
        const cpf = tr.querySelector(" .cpf").textContent;
        const tipo = tr.querySelector(" .tipo").textContent;
        const email = tr.querySelector(" .email").textContent;
        const senha = tr.querySelector(" .senha").textContent;

        const modal = document.createElement("div");
        modal.innerHTML = `
<div style="
position: fixed; inset: 0; background: rgba(0,0,0,0.6);
display: flex; justify-content: center; align-items: center; z-index: 1000;
animation: fadeIn 0.1s ease;
">
<div style="
background: white; border-radius: 12px; padding: 20px 30px;
width: 400px; max-width: 80%;
box-shadow: 0 8px 24px rgba(0,0,0,0.3);
font-family: Arial, sans-serif; animation: popIn 0.2s ease;
">
<h2 style="color: #1b6a7a; text-align: center; margin-bottom: 20px;">Editar Usuário</h2>
<form id="formEditarUsuario" style="display: flex; flex-direction: column; gap: 10px;">
<input type="text" value="RM" value="<?php htmlspecialchars($row['RM']) ?>" style="width: 100%; padding: 5px; border: 1px solid #ccc; border-radius: 6px; background: #f9f9f9;">

```

FONTE – AUTORIA PRÓPRIA (2025)

Figura 32 - Código Tela Lista de Usuários 3

```

<!-- Nome e CPF -->
<div style="display:flex; gap:16px;">
  <div style="flex:1;">
    <label style="font-weight:bold;">Nome</label>
    <input type="text" name="nome" value="{nome}" required
      style="width:100%; padding:8px; border:1px solid #ccc; border-radius:6px;">
    </div>
    <div style="flex:1;">
      <label style="font-weight:bold;">CPF</label>
      <input type="text" name="CPF" value="{cpf}" required
        style="width:100%; padding:8px; border:1px solid #ccc; border-radius:6px;">
    </div>
  </div>
  <!-- Tipo e Email -->
  <div style="display:flex; gap:16px;">
    <div style="flex:1;">
      <label style="font-weight:bold;">Tipo</label>
      <select name="tipo" required
        style="width:100%; padding:8px; border:1px solid #ccc; border-radius:6px;">
        <option value="Professor" {tipo === "Professor" ? "selected" : ""}>Professor</option>
        <option value="Secretaria" {tipo === "Secretaria" ? "selected" : ""}>Secretaria</option>
      </select>
    </div>
    <div style="flex:1;">
      <label style="font-weight:bold;">Email</label>
      <input type="email" name="Email" value="{email}" required
        style="width:100%; padding:8px; border:1px solid #ccc; border-radius:6px;">
    </div>
  </div>
  <!-- Senha -->
  <div>
    <label style="font-weight:bold;">Senha</label>
    <input type="text" name="Senha" value="{senha}" required
      style="width:100%; padding:8px; border:1px solid #ccc; border-radius:6px;">
  </div>
  <!-- Botões -->
  <div style="margin-top:16px; text-align:center;">
    <button type="submit" style="
      background:#51b6fa; color:white; border:none;
      padding:10px 20px; border-radius:6px; cursor:pointer;
      font-size:0.9rem; font-weight:bold; margin-right:10px;">
      Salvar</button>
    <button type="button" id="fecharModal" style="
      background:#ccc; color:#333; border:none;
      padding:10px 20px; border-radius:6px; cursor:pointer;
      font-size:0.9rem; font-weight:bold;">
      Cancelar</button>
  </div>
</form>
</div>

```

FONTE – AUTORIA PRÓPRIA (2025)

Figura 33 - Código Tela Lista de Usuários 4

```

<style>
  @keyframes fadeIn { from {opacity:0;} to {opacity:1;} }
  @keyframes popIn { from {transform:scale(0.9); opacity:0;} to {transform:scale(1); opacity:1;} }
</style>
</div>
</div>
);
document.body.appendChild(modal);
// Fechar modal
modal.querySelector("#fecharModal").addEventListener("click", () => modal.remove());
// Salvar edicão
modal.querySelector("#formEditarUsuario").addEventListener("submit", e => {
  e.preventDefault();
  const dados = new URLSearchParams(new FormData(e.target));

  fetch("editar.php", {
    method: "POST",
    body: dados
  })
  .then(res => res.text())
  .then(msg => {
    alert(msg);
    tr.querySelector(".nome").textContent = e.target.Nome.value;
    tr.querySelector(".cpf").textContent = e.target.CPF.value;
    tr.querySelector(".tipo").textContent = e.target.Tipo.value;
    tr.querySelector(".email").textContent = e.target.Email.value;
    tr.querySelector(".senha").textContent = e.target.Senha.value;
    modal.remove();
  });
});
});
</script>
</body>
</html>

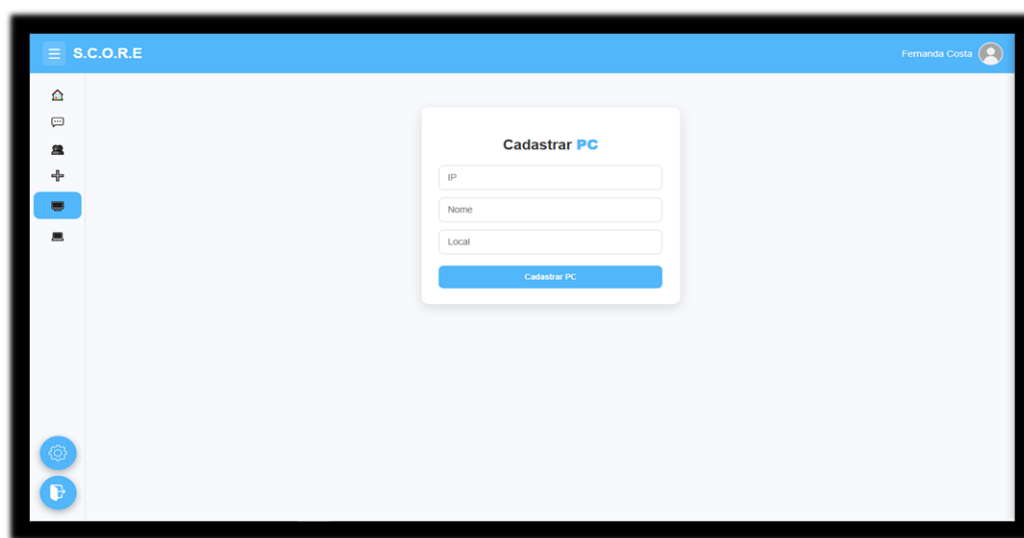
```

FONTE – AUTORIA PRÓPRIA (2025)

3.4.4 Tela de Cadastro de PC

A imagem 34 mostra a tela de cadastro de PC, que é a mesma para todos os usuários. Essa tela possui campos de “ip”, “nome”, “local”, “senha”, e um botão de cadastrar.

Figura 34 - Tela Cadastrar PC



FONTE – AUTORIA PRÓPRIA (2025)

A Figura 34 mostra a interface utilizada para o registro de novos computadores dentro do sistema S.C.O.R.E. No centro da tela, há um formulário identificado pelo título “Cadastrar PC”.

Os campos disponíveis para preenchimento são: “IP”, “Nome” e “Local”, seguidos pelo botão “Cadastrar PC”, em destaque azul, que confirma a operação de cadastro.

A interface mantém o padrão visual do sistema, com um *design* limpo e organizado: fundo branco, barra superior azul contendo o nome do usuário *logado* (“Fernanda Costa”) e menu lateral com ícones de navegação, entre eles o ícone selecionado correspondente à função de cadastro de computadores.

A estrutura do documento inicia-se com a declaração `<!DOCTYPE html>` e a tag `<html lang="pt-BR">`, que especifica o idioma da página. No cabeçalho (`<head>`), encontram-se as definições de metadados, o título “Cadastrar PC - S.C.O.R.E.” e o bloco de estilos internos (`<style>`), responsável pelo design da interface.

O corpo do documento (`<body>`) contém a *div* principal, denominada *cadastro-module*, que centraliza o conteúdo e aplica uma animação de entrada suave, definida pela regra `@keyframes fadeIn`. Dentro dela, a *div* *cadastro-box* agrupa o título e o formulário, compondo a área de interação do usuário. O título, “Cadastrar PC”, é centralizado e apresenta destaque visual na palavra “PC”, exibida em azul para manter a identidade cromática do sistema.

O formulário é implementado com a *tag* `<form>` (identificado pelo `id="formCadastroPC"`) e utiliza o método *POST*, com o atributo `action="cadastrar_pc.php"`, que define o *script* responsável pelo processamento do cadastro. Os campos disponíveis para entrada de dados são criados por meio de elementos `<input>`, permitindo o preenchimento dos campos “IP”, “Nome” e “Local”. Todos são marcados como obrigatórios (*required*), garantindo a integridade das informações antes do envio.

O botão de submissão, `<button type="submit">Cadastrar PC</button>`, ocupa toda a largura da área do formulário e possui destaque na cor azul (`background:#51b6fa`), alterando para um tom mais escuro ao passar o cursor (*hover*). O efeito de transição é complementado por uma leve animação de ampliação, tornando a interação mais fluida.

A seção de estilos (CSS) define um *layout* limpo e moderno, com fundo branco, bordas arredondadas (*border-radius*), sombras sutis (*box-shadow*) e espaçamento equilibrado entre os elementos. O *design* é responsivo, adaptando automaticamente a largura da caixa de cadastro em telas menores (abaixo de 520 pixels), conforme a regra `@media (max-width:520px)`.

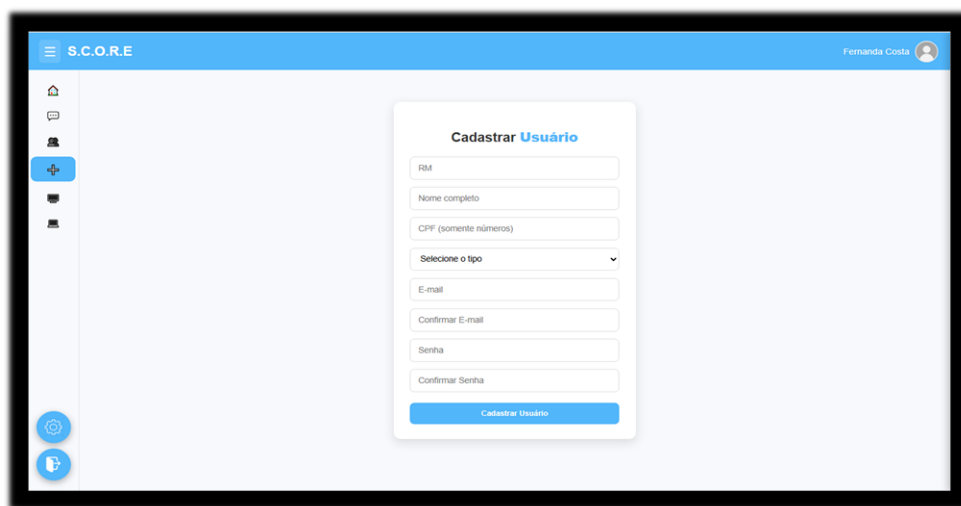
A funcionalidade de validação e envio assíncrono é implementada em *JavaScript* através de uma função autoexecutável (*IIFE*). O código intercepta o evento *submit* do formulário, impede o envio tradicional (`preventDefault()`) e utiliza a API `fetch()` para enviar os dados ao servidor de forma assíncrona, sem recarregar a página.

O retorno do servidor é tratado no formato *JSON* e interpretado para fornecer *feedback* imediato ao usuário: quando o cadastro é bem-sucedido, uma mensagem verde é exibida com o texto “PC cadastrado com sucesso”; em caso de falha, a

3.4.5 Tela Cadastro de Usuário

A imagem 36 mostra a tela de cadastro de usuário feito pela secretaria.

Figura 36 - Tela cadastro SCORE



FONTE – AUTORIA PRÓPRIA (2025)

A Figura 36 apresenta a interface destinada ao cadastro de novos usuários no sistema S.C.O.R.E. No centro da tela, há um formulário identificado pelo título “Cadastrar Usuário”. Os campos disponíveis incluem: “RM”, “Nome completo”, “CPF (somente números)”, um menu suspenso para “Selecionar o tipo”, “E-mail”, “Confirmar E-mail”, “Senha” e “Confirmar Senha”.

Abaixo dos campos, encontra-se o botão “Cadastrar Usuário”, destacado em azul, responsável por concluir o processo de registro.

O *layout* mantém a identidade visual do sistema, com fundo branco, barra superior azul contendo o nome do usuário *logado* (“Fernanda Costa”) e menu lateral para navegação entre as funcionalidades.

A estrutura da página é iniciada com a declaração `<!DOCTYPE html>` e a tag `<html lang="pt-BR">`, indicando que o conteúdo está em português. No `<head>`, são

definidos o título “Cadastrar Usuário”, as configurações de codificação e um bloco de estilos internos (`<style>`) que determina o *layout* da tela.

O corpo do documento (`<body>`) contém a *div* principal com a classe *cadastro-module*, que centraliza o conteúdo na tela, e outra *div* interna, *cadastro-box*, responsável por agrupar o título e o formulário. O cabeçalho da caixa exibe o texto “Cadastrar Usuário”, destacando a palavra “Usuário” em azul, em conformidade com a identidade visual do sistema.

O formulário, criado com a *tag* `<form>` e identificado pelo *id* “*formCadastroUsuario*”, utiliza o método *POST* e o atributo *action* “*cadastar.php*”, que define o arquivo responsável por processar o envio dos dados. Os campos de entrada são implementados por meio de elementos `<input>` e `<select>`, permitindo que o usuário insira informações como “RM”, “Nome completo”, “CPF (somente números)”, “Tipo” (com opções *Secretaria* ou *Professor*), “E-mail”, “Confirmar E-mail”, “Senha” e “Confirmar Senha”.

O botão principal, `<button type="submit">Cadastrar Usuário</button>`, ocupa toda a largura da caixa e recebe destaque em azul, alterando ligeiramente de tom ao passar o cursor sobre ele, por meio do seletor `:hover`. Esse comportamento visual é complementado por animações definidas na regra `@keyframes fadeIn`, que cria uma transição suave de entrada da caixa de cadastro.

A estilização utiliza uma paleta de cores leves, bordas arredondadas e sombras sutis (*box-shadow*), mantendo coerência com o padrão visual do sistema S.C.O.R.E. O *layout* também é responsivo, adaptando-se automaticamente a dispositivos com largura inferior a 520 pixels.

A validação e o envio dos dados são feitos por uma função autoexecutável em JavaScript, que impede o envio automático do formulário e verifica se e-mail e senha coincidem, exibindo mensagens de erro em vermelho quando necessário.

Quando todas as validações são concluídas com sucesso, o código envia os dados ao servidor utilizando o método `fetch()` com corpo do tipo `FormData`. O retorno, esperado em formato JSON, é interpretado para informar ao usuário se o cadastro foi realizado corretamente (`data.sucesso == true`). Em caso positivo, a mensagem é

exibida em verde; caso contrário, em vermelho, com *feedback* claro sobre o problema. O código combina design responsivo, validação interativa e comunicação assíncrona, tornando o registro de usuários no S.C.O.R.E. mais intuitivo e seguro, garantindo uma experiência eficiente em toda a aplicação.

Figura 37 - Código Tela Cadastro SCORE

```

<!DOCTYPE html>
<html lang="pt-BR">
<head>
<meta charset="UTF-8">
<title>Cadastrar Usuário</title>
<style>--
</style>
</head>
<body>
<div class="cadastro-module">
<div class="cadastro-box" role="region" aria-label="Formulário de cadastro de usuário">
<h2>Cadastrar <strong>Usuário</strong></h2>

<form id="formCadastroUsuario" action="cadastrar.php" method="post" autocomplete="off">
<input type="number" name="ra" placeholder="RA" required>
<input type="text" name="nome" placeholder="Nome completo" required>
<input type="text" name="cpf" placeholder="CPF (somente números)" maxlength="11" required>

<select name="tipo" required>
<option value="">Selecione o tipo</option>
<option value="Secretaria">Secretaria</option>
<option value="Professor">Professor</option>
</select>

<input type="email" name="email" placeholder="E-mail" required>
<input type="email" name="confirmar_email" placeholder="Confirmar E-mail" required>

<input type="password" name="senha" placeholder="Senha" required>
<input type="password" name="confirmar_senha" placeholder="Confirmar Senha" required>

<button type="submit">Cadastrar Usuário</button>
</form>

<div id="cadastroMsg" class="cadastro-msg" role="status" aria-live="polite"></div>
</div>
</div>

```

FONTE – AUTORIA PRÓPRIA (2025)

Figura 38 - Código tela de Cadastro SCORE 2

```

<script>
(function() {
const form = document.getElementById('formCadastroUsuario');
const msg = document.getElementById('cadastroMsg');

form.addEventListener('submit', async function(e) {
e.preventDefault();
msg.style.display = 'none';
const fd = new FormData(form);

if (fd.get('email') !== fd.get('confirmar_email')) {
msg.textContent = 'E-mails não coincidem';
msg.style.display = 'block';
msg.style.background = '#f8d7da';
msg.style.color = '#721c24';
return;
}

if (fd.get('senha') !== fd.get('confirmar_senha')) {
msg.textContent = 'Senhas não coincidem';
msg.style.display = 'block';
msg.style.background = '#f8d7da';
msg.style.color = '#721c24';
return;
}

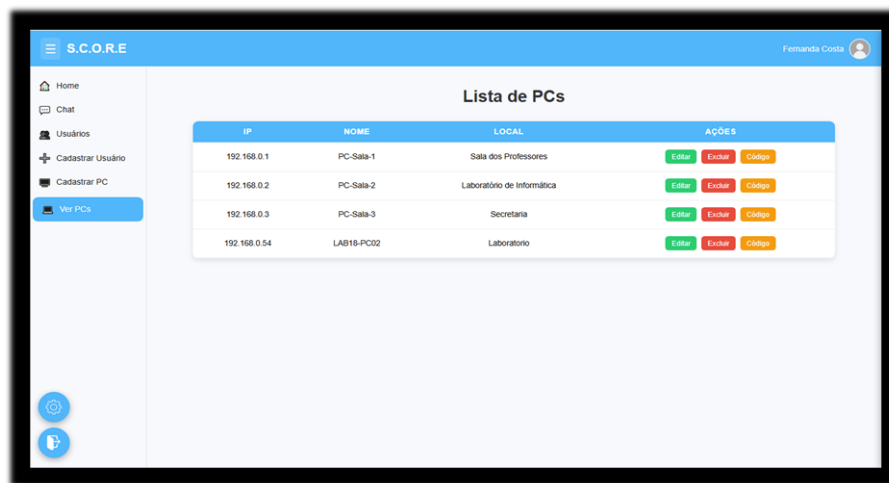
try {
const res = await fetch(form.action, { method: 'POST', body: fd });
const data = await res.json();
if (data.sucesso) {
msg.textContent = data.mensagem || 'Cadastrado com sucesso';
msg.style.display = 'block';
msg.style.background = '#e2e3e5';
msg.style.color = '#353738';
} else {
msg.textContent = data.mensagem || 'Erro no cadastro';
msg.style.display = 'block';
msg.style.background = '#f8d7da';
msg.style.color = '#721c24';
}
} catch (err) {
msg.textContent = 'Erro de rede ou resposta inválida.';
msg.style.display = 'block';
msg.style.background = '#f8d7da';
msg.style.color = '#721c24';
console.error(err);
}
})();
</script>
</body>
</html>

```

FONTE – AUTORIA PRÓPRIA (2025)

3.4.6 Tela Lista de computadores cadastrados

Figura 39 - Lista de computadores cadastrados



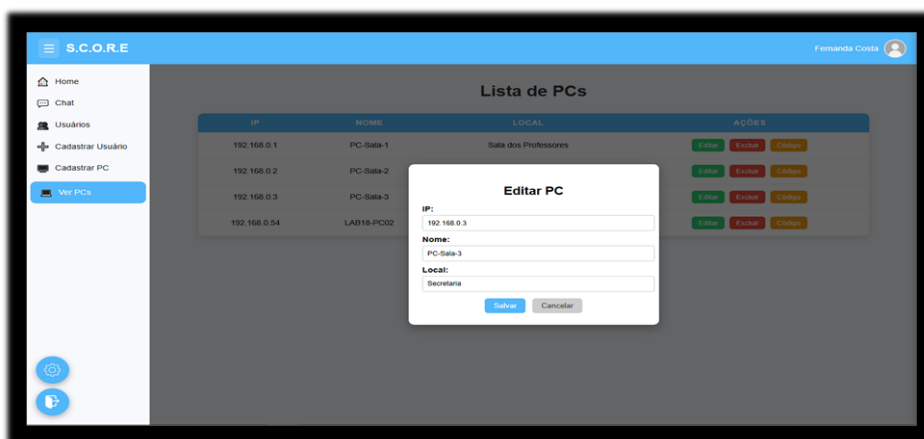
IP	NOME	LOCAL	AÇÕES
192.168.0.1	PC-Sala-1	Sala dos Professores	Editar Excluir Código
192.168.0.2	PC-Sala-2	Laboratório de Informática	Editar Excluir Código
192.168.0.3	PC-Sala-3	Secretaria	Editar Excluir Código
192.168.0.54	LAB18-PC02	Laboratório	Editar Excluir Código

FONTE – AUTORIA PRÓPRIA (2025)

A Figura 39 apresenta a tela “Lista de PCs”, onde são exibidos os computadores previamente cadastrados no sistema. A tabela contém as colunas “IP”, “Nome”, “Local” e “Ações”.

Na coluna de ações, três botões — “Editar” (verde), “Excluir” (vermelho) e “Código” (laranja) — permitem gerenciar cada computador, mantendo o layout padrão do sistema com barra azul superior e menu lateral.

Figura 40 - Edição de informações de um computador



IP	NOME	LOCAL	AÇÕES
192.168.0.1	PC-Sala-1	Sala dos Professores	Editar Excluir Código
192.168.0.2	PC-Sala-2	Laboratório de Informática	Editar Excluir Código
192.168.0.3	PC-Sala-3	Secretaria	Editar Excluir Código
192.168.0.54	LAB18-PC02	Laboratório	Editar Excluir Código

Editar PC

IP:

Nome:

Local:

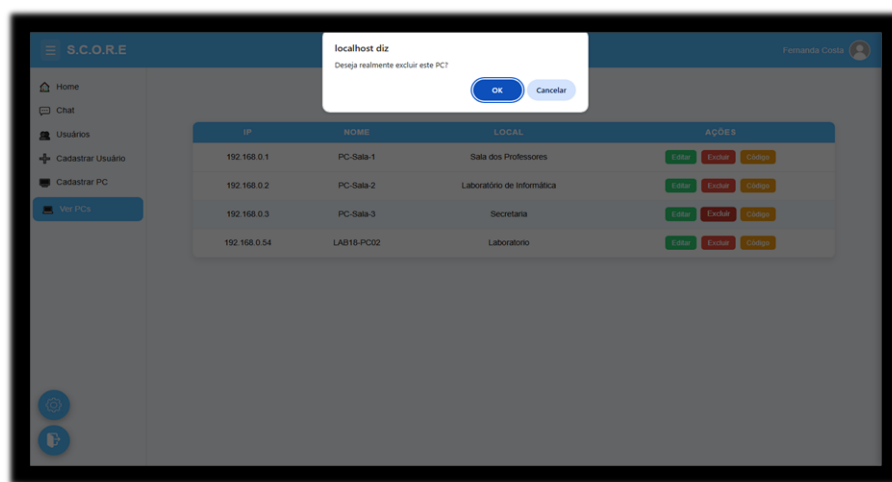
[Salvar](#) [Cancelar](#)

FONTE – AUTORIA PRÓPRIA (2025)

A Figura 40 é mostra o *pop-up* “Editar PC”, que permite modificar os dados de um computador cadastrado. O formulário contém os campos “IP”, “Nome” e “Local”, preenchidos automaticamente com as informações atuais.

Na parte inferior, encontram-se os botões “Salvar” e “Cancelar”, utilizados para confirmar ou abortar as alterações. O restante da tela é levemente escurecido, destacando o formulário de edição

Figura 41 - Tela Confirmação de Exclusão



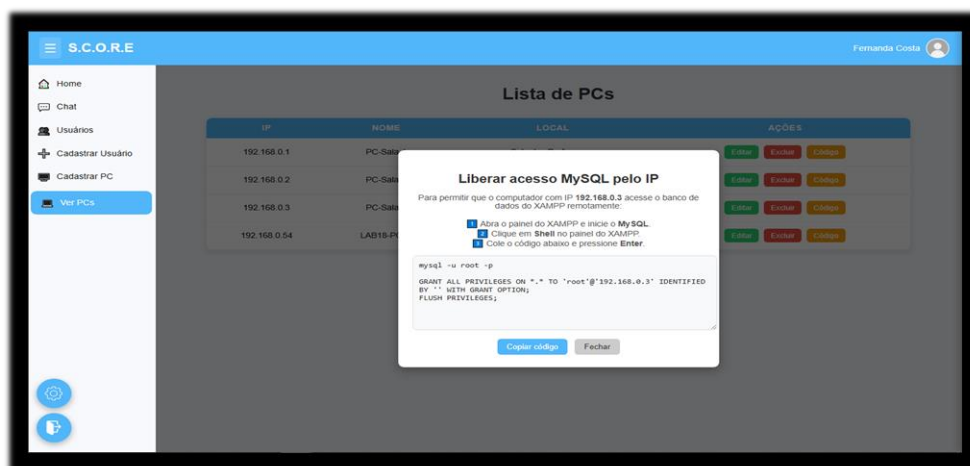
FONTE – AUTORIA PRÓPRIA (2025)

A Figura 41 mostra a caixa de diálogo exibida ao clicar no botão “Excluir”. Ao acionar essa opção, o sistema apresenta um *pop-up* centralizado com a mensagem “Deseja realmente excluir este PC?”, acompanhada das opções “OK” e “Cancelar”.

Essa janela de confirmação tem a função de garantir que o usuário realmente deseja prosseguir com a exclusão, evitando que registros sejam removidos de forma equivocada. O botão “OK” confirma a ação e efetua a exclusão definitiva do dispositivo, enquanto o botão “Cancelar” permite retornar à tela anterior, interrompendo o processo antes que qualquer alteração seja feita.

Esse recurso de confirmação prévia é fundamental para reforçar a segurança e a confiabilidade do sistema, assegurando que apenas ações intencionais sejam realizadas. Além disso, contribui para uma experiência de uso mais segura e controlada, demonstrando a preocupação do sistema em preservar a integridade dos dados e o bom gerenciamento dos dispositivos cadastrados.

Figura 42 - Liberação de acesso ao MySQL pelo IP



FONTE – AUTORIA PRÓPRIA (2025)

A Figura 42 exibe a janela “*Liberar acesso MySQL pelo IP*”, que orienta o usuário sobre como permitir que um computador acesse o banco de dados por meio de um endereço *IP* específico. O *pop-up* contém instruções passo a passo e um campo com comandos prontos para execução, incluindo o botão “*Copiar código*” para facilitar o processo. Essa funcionalidade é essencial para a configuração e integração segura entre o sistema S.C.O.R.E e o banco de dados.

A estrutura do código inicia-se com um bloco *PHP* responsável pela validação de sessão do usuário, impedindo o acesso não autorizado à página. Em seguida, é realizada a conexão com o banco de dados por meio do arquivo *conexao.php*. Uma consulta *SQL (SELECT)* é executada para recuperar os registros da tabela “*computador*”, retornando os campos “*IP*”, “*Nome*” e “*Local*”, que são exibidos dinamicamente dentro da tabela *HTML*.

O documento é estruturado conforme o padrão *HTML5*, começando com a declaração `<!DOCTYPE html>` e a tag `<html lang="pt-BR">`, que define o idioma da interface. No cabeçalho (`<head>`), encontram-se as configurações de metadados, o título da página (“*Ver PCs - S.C.O.R.E*”) e o bloco de estilos internos (`<style>`), responsável pelo *design* visual. O CSS aplicado segue o mesmo padrão das demais telas do sistema, utilizando tons de azul (`#51b6fa`) como cor principal, tipografia legível (*Arial, sans-serif*) e espaçamento equilibrado entre os elementos. As bordas arredondadas (*border-radius*), as sombras sutis (*box-shadow*) e os efeitos de transição (*transition*) conferem uma aparência moderna e profissional à interface.

No corpo do documento (), o título “Lista de PCs” é centralizado, seguido por uma tabela HTML que organiza os dados dos computadores, gerados dinamicamente via PHP foreach. Na coluna “Ações”, três botões — “Editar” (verde), “Excluir” (vermelho) e “Liberar” (laranja) — permitem gerenciar os registros de forma intuitiva.

A interatividade é implementada em JavaScript, permitindo excluir PCs com confirmação e requisições assíncronas (fetch) ao script “excluir_pc.php”. Para edição, um pop-up modal pré-preenche os campos “IP”, “Nome” e “Local”, enviando as alterações via AJAX para “editar_pc.php”, enquanto o botão “Cancelar” fecha o modal sem alterações, mantendo o foco na área de edição.

A funcionalidade “Código” exibe um pop-up detalhado com instruções e o comando SQL necessário para liberar o acesso remoto ao MySQL pelo IP do computador selecionado, facilitando a configuração do banco de dados. O pop-up inclui um botão “Copiar código”, que permite copiar automaticamente o comando para a área de transferência, tornando o processo rápido e seguro.

Integrando PHP, HTML, CSS e JavaScript a tela “Lista de PCs” permite gerenciar dispositivos cadastrados, com consulta, edição, exclusão e liberação de acesso remoto de forma prática. A interface mantém o padrão visual do S.C.O.R.E., garantindo usabilidade, segurança e responsividade.

Figura 43 - Código Lista de computadores 1

```

<code>
</code>

```

FONTE – AUTORIA PRÓPRIA (2025)

Figura 44 - Código Lista de computadores 2

```

<script>
// === EXCLUIR PC ===
document.querySelectorAll("#btn-excluir").forEach(btn => {
  btn.addEventListener("click", () => {
    const tr = btn.closest("tr");
    const ip = tr.dataset.ip;

    if (confirm("Deseja realmente excluir este PC?")) {
      fetch("excluir_pc.php", {
        method: "POST",
        headers: { "Content-Type": "application/x-www-form-urlencoded" },
        body: "ip=" + encodeURIComponent(ip)
      })
      .then(res => res.text())
      .then(msg => {
        alert(msg);
        if (!msg.toLowerCase().includes("erro")) {
          tr.remove();
        }
      })
      .catch(() => alert("Erro na requisição."));
    }
  });
});

// === EDITAR PC ===
document.querySelectorAll("#btn-editar").forEach(btn => {
  btn.addEventListener("click", () => {
    const tr = btn.closest("tr");
    const ip = tr.querySelector("#ip").textContent.trim();
    const nome = tr.querySelector("#nome").textContent.trim();
    const local = tr.querySelector("#local").textContent.trim();

    mostrarModalEditar(ip, nome, local, tr);
  });
});

function mostrarModalEditar(ip, nome, local, tr) {
  const modal = document.createElement("div");
  modal.innerHTML = `
  <div style="
    position: fixed; inset: 0; display: flex; justify-content: center; align-items: center;
    background: rgba(0,0,0,0.6); z-index: 1000;
  ">
    <div style="
      background: white; border-radius: 10px; padding: 10px; width: 400px;
      min-width: 300px; box-shadow: 0 0 10px rgba(0,0,0,0.3);
      font-family: Arial, sans-serif;
    ">
      <h2 style="text-align: center; margin-bottom: 10px;">Editar PC/</h2>
      <div id="form-editar">
        <label style="font-weight: bold;">IP:</label>
        <input type="text" id="edit-ip" value="${ip}" required readonly
          style="width: 100%; padding: 5px; margin: 5px 0 5px; border: 1px solid #ccc; border-radius: 5px;">
        <label style="font-weight: bold;">Nome:</label>
        <input type="text" id="edit-nome" value="${nome}" required
          style="width: 100%; padding: 5px; margin: 5px 0 5px; border: 1px solid #ccc; border-radius: 5px;">
      </div>
    </div>
  `;
  document.body.appendChild(modal);

  const form = modal.querySelector("#form-editar");
  const fechar = modal.querySelector("#fechar-editar");

  form.addEventListener("submit", e => {
    e.preventDefault();

    const novoIP = modal.querySelector("#edit-ip").value.trim();
    const novoNome = modal.querySelector("#edit-nome").value.trim();
    const novoLocal = modal.querySelector("#edit-local").value.trim();

    fetch("editar_pc.php", {
      method: "POST",
      headers: { "Content-Type": "application/x-www-form-urlencoded" },
      body: new URLSearchParams({
        ip: novoIP,
        nome: novoNome,
        local: novoLocal
      })
    })
    .then(res => res.text())
    .then(msg => {
      alert(msg);
      if (!msg.toLowerCase().includes("erro")) {
        tr.querySelector("#ip").textContent = novoIP;
        tr.querySelector("#nome").textContent = novoNome;
        tr.querySelector("#local").textContent = novoLocal;
        tr.dataset.ip = novoIP;
        modal.remove();
      }
    })
    .catch(() => alert("Erro na requisição."));
  });

  fechar.addEventListener("click", () => modal.remove());
}

```

FONTE – AUTORIA PRÓPRIA (2025)

Figura 45 - Código Lista de computadores 3

```

<input type="text" id="edit-local" value="${local}" required
  style="width: 100%; padding: 5px; margin: 5px 0 5px; border: 1px solid #ccc; border-radius: 5px;">
</div>
<div style="text-align: center;">
  <button type="submit" style="
    background: #10a64a; color: white; border: none; padding: 5px 10px;
    border-radius: 5px; cursor: pointer; font-size: 0.9em; margin-right: 5px;
  ">Salvar/Atualizar
  <button type="button" id="fechar-editar" style="
    background: #ccc; color: #333; border: none; padding: 5px 10px;
    border-radius: 5px; cursor: pointer; font-size: 0.9em;
  ">Cancelar/Fechar
</div>
</form>
</div>
</div>
document.body.appendChild(modal);

const form = modal.querySelector("#form-editar");
const fechar = modal.querySelector("#fechar-editar");

form.addEventListener("submit", e => {
  e.preventDefault();

  const novoIP = modal.querySelector("#edit-ip").value.trim();
  const novoNome = modal.querySelector("#edit-nome").value.trim();
  const novoLocal = modal.querySelector("#edit-local").value.trim();

  fetch("editar_pc.php", {
    method: "POST",
    headers: { "Content-Type": "application/x-www-form-urlencoded" },
    body: new URLSearchParams({
      ip: novoIP,
      nome: novoNome,
      local: novoLocal
    })
  })
  .then(res => res.text())
  .then(msg => {
    alert(msg);
    if (!msg.toLowerCase().includes("erro")) {
      tr.querySelector("#ip").textContent = novoIP;
      tr.querySelector("#nome").textContent = novoNome;
      tr.querySelector("#local").textContent = novoLocal;
      tr.dataset.ip = novoIP;
      modal.remove();
    }
  })
  .catch(() => alert("Erro na requisição."));
});

fechar.addEventListener("click", () => modal.remove());
}

```

FONTE – AUTORIA PRÓPRIA (2025)

Figura 46 - Código Lista de computadores 4

```

// == LIBERAR / BLOQUEAR PC ==
document.querySelectorAll(".btn-liberar").forEach(btn => {
  btn.addEventListener("click", () => {
    const tr = btn.closest("tr");
    const ip = tr.dataset.ip;

    mostrarModalComCodigoMySQL(ip);
  });
});

function mostrarModalComCodigoMySQL(ip) {
  const codigo = `mysql -u root -p

GRANT ALL PRIVILEGES ON *.* TO 'root'@'${ip}' IDENTIFIED BY '' WITH GRANT OPTION;
FLUSH PRIVILEGES;`;

  const modal = document.createElement("div");
  modal.innerHTML = `
<div style="
  position: fixed; inset: 0; display: flex; justify-content: center; align-items: center;
  background: rgba(0,0,0,0.6); z-index: 1000;
">
  <div style="
    background: white; border-radius: 10px; padding: 25px; width: 500px;
    max-width: 90%; box-shadow: 0 6px 20px rgba(0,0,0,0.3);
    font-family: Arial, sans-serif; text-align: center;
  ">
    <h2 style="margin-bottom: 10px; color: #333;">liberar acesso MySQL pelo IP</h2>
    <p style="color: #555; font-size: 0.95em; margin-bottom: 15px;">
      Para permitir que o computador com IP <b>${ip}</b> acesse o banco de dados do XAMPP remotamente:<br><br>
      1. Abra o painel do XAMPP e inicie o <b>MySQL</b>.
      2. Clique em <b>Shell</b> no painel do XAMPP.
      3. Cole o código abaixo e pressione <b>Enter</b>.
      Isso liberará o IP para acessar o MySQL remotamente.
    </p>
    <div style="border: 1px solid #ddd; border-radius: 5px; background: #f8f9fa; color: #222; font-family: monospace; font-size: 0.9em; margin-bottom: 15px; padding: 5px 10px 5px 10px;">
      <code>${codigo}</code>
    </div>
    <button id="copiar-codigo" style="background: #130f4a; color: white; border: none; padding: 5px 15px; border-radius: 5px; cursor: pointer; font-size: 0.9em; margin-right: 5px;">
      Copiar código
    </button>
    <button id="fechar-modal" style="background: #ccc; color: #333; border: none; padding: 5px 15px; border-radius: 5px; cursor: pointer; font-size: 0.9em;">
      Fechar
    </button>
  </div>
</div>

```

FONTE – AUTORIA PRÓPRIA (2025)

Figura 47 - Código Lista de computadores 5

```

document.body.appendChild(modal);

modal.querySelector("#copiar-codigo").addEventListener("click", () => {
  const textarea = modal.querySelector("#codigo-mysql");
  if (navigator.clipboard) {
    navigator.clipboard.writeText(textarea.value).then(() => {
      alert("Código copiado para a área de transferência!");
    }).catch(() => alert("Falha ao copiar o código."));
  } else {
    textarea.select();
    document.execCommand("copy");
    alert("Código copiado para a área de transferência!");
  }
});

modal.querySelector("#fechar-modal").addEventListener("click", () => {
  modal.remove();
});
</script>
</body>
</html>

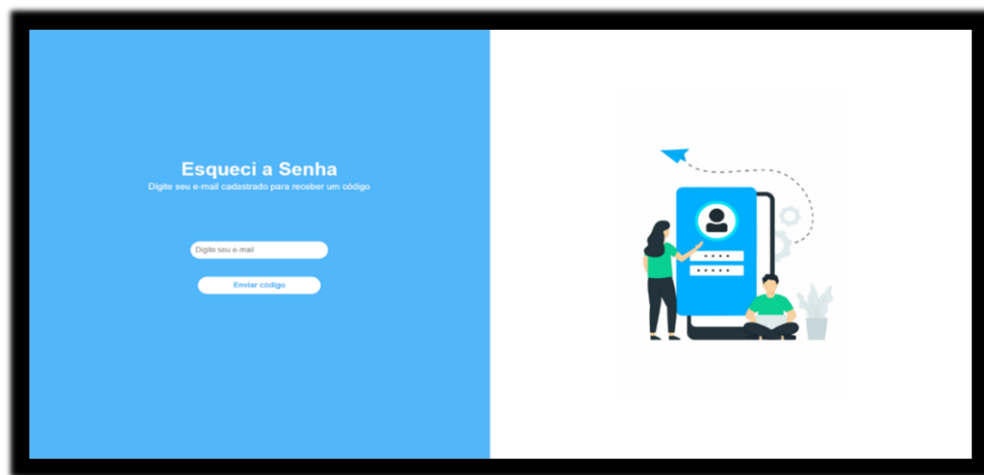
```

FONTE – AUTORIA PRÓPRIA (2025)

3.4.7 Tela Esqueci a Senha

A imagem 34 mostra a tela de esqueci senha para recuperação de acesso, essa tela possui os campos “Digite seu *e-mail*” e o botão “Enviar código”.

Figura 48 - Tela Esqueci Senha



FONTE – AUTORIA PRÓPRIA (2025)

As figuras 35, 36, 37 e 38 mostram a implementação do recurso de recuperação de senha do sistema. Ela é composta por um arquivo HTML, que apresenta a interface para o usuário, e por um *script* em PHP, que processa a requisição e envia o *e-mail* de recuperação.

Estrutura da Página HTML

O documento é definido em HTML5 com o idioma configurado para português do Brasil (pt-br) e *meta tag viewport* para garantir responsividade em diferentes dispositivos. O título exibido na aba do navegador é “Esqueci Minha Senha – S.C.O.R.E.” e o arquivo de estilo *style.css* é importado para a formatação visual.

Interface do Usuário

*Int*O corpo da página é organizado em um *container* com duas áreas:

Lado esquerdo: apresenta o título “Esqueci a Senha” e um subtítulo que orienta o usuário a digitar o *e-mail* cadastrado para receber um código de recuperação.

Formulário: identificado por *id="formEsqueci"*, possui um campo de entrada do tipo *e-mail* e um botão para enviar o pedido de código. Logo abaixo, há uma *div* com *id="mensagem"* destinada a exibir o retorno da operação (sucesso ou erro), com cor que muda dinamicamente.

Lado direito: exibe uma imagem ilustrativa da funcionalidade.

Funcionalidade em *JavaScript*

Um *script* no final da página adiciona um *event listener* ao formulário para capturar o envio. Quando o usuário informa o *e-mail* e clica em “Enviar código”, a função em *JavaScript* impede o envio padrão (*preventDefault*), coleta o valor digitado e envia uma requisição *fetch* assíncrona para o arquivo *esqueci_senha.php*, no formato JSON (*method:POST*).

Quando a resposta chega, o *script* converte para JSON, exibe a mensagem de retorno na *div* “mensagem” e altera a cor para *lightgreen* em caso de sucesso ou *red* em caso de erro. Se o *status* for “ok”, após dois segundos o usuário é redirecionado para a página *resetar_senha.html* com o *e-mail* codificado na URL.

Processamento em PHP

O arquivo *esqueci_senha.php* trata a lógica de envio do código. Inicialmente, configura as opções de exibição e registro de erros, define o *header* para retorno em JSON e cria uma conexão com o banco de dados usando PDO com tratamento de exceções.

Em seguida, decodifica os dados recebidos em JSON e verifica se o *e-mail* foi informado. Caso o *e-mail* não exista na base, retorna um JSON com *status* de erro. Se o *e-mail* existir, é gerado um código numérico de 6 dígitos utilizando *random_int* e *str_pad*, que é salvo no campo de recuperação do usuário no banco de dados *UPDATE*.

É preparado o conteúdo do *e-mail*, com assunto, mensagem e *headers*, e a função *mail()* é utilizada para enviar o código ao usuário. Se o envio for bem-sucedido, retorna um JSON com *status* “ok” e a mensagem de confirmação. Caso contrário, informa que não foi possível enviar o *e-mail*. Qualquer exceção é capturada e devolvida em formato JSON, garantindo que o cliente receba sempre uma resposta estruturada.

Figura 49 - Código da Tela Esqueci senha 1

```

<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <title>Esqueci Minha Senha - S.C.O.R.E</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <div class="container">
    <!-- Lado esquerdo -->
    <div class="left-side">
      <div class="welcome-text">
        <h1 class="fade-in">Esqueci a Senha</h1>
        <p class="subtitle">Digite seu e-mail cadastrado para receber um código</p>
      </div>
      <form id="formEsqueci" class="login-form">
        <div class="input-group">
          <input type="email" id="email" placeholder="Digite seu e-mail" required>
        </div>
        <button type="submit">Enviar código</button>
      </form>
      <div id="mensagem" style="margin-top:10px; color:■white; font-weight:bold;"></div>
    </div>
    <!-- Lado direito -->
    <div class="right-side">
      
    </div>
  </div>
  <script>
    document.getElementById("formEsqueci").addEventListener("submit", async function(e) {
      e.preventDefault();

      let email = document.getElementById("email").value;

      let resposta = await fetch("esqueci_senha.php", {
        method: "POST",
        headers: { "Content-Type": "application/json" },
        body: JSON.stringify({ email: email })
      });
  </script>

```

FONTE – AUTORIA PRÓPRIA (2025)

Figura 50 - Código da Tela Esqueci senha 2

```

let data = await resposta.json();
document.getElementById("mensagem").innerText = data.mensagem;
document.getElementById("mensagem").style.color = (data.status === "ok") ? "lightgreen" : "red";

// ♦ Se deu certo, redireciona para resetar senha
if (data.status === "ok") {
  setTimeout(() => {
    window.location.href = "resetar_senha.html?email=" + encodeURIComponent(email);
  }, 2000);
}
});
</script>
</body>
</html>

```

FONTE – AUTORIA PRÓPRIA (2025)

Figura 51 - Código da Tela Esqueci senha 3

```

<?php
error_reporting(error_level: E_ALL);
ini_set(option: "display_errors", value: 0); // não mostrar erros na tela
ini_set(option: "log_errors", value: 1);
ini_set(option: "error_log", value: __DIR__ . "/php_errors.log");

header(header: "Content-Type: application/json; charset=utf-8");

try {
    $pdo = new PDO(dsn: "mysql:host=localhost;dbname=score;charset=utf8", username: "root", password: "");
    $pdo->setAttribute(attribute: PDO::ATTR_ERRMODE, value: PDO::ERRMODE_EXCEPTION);

    $data = json_decode(json: file_get_contents(filename: "php://input"), associative: true);

    if (!isset($data["email"])) {
        echo json_encode(value: ["status" => "erro", "mensagem" => "Email não enviado"]);
        exit;
    }

    $email = trim(string: $data["email"]);

    // ♦ Verifica se o email existe
    $stmt = $pdo->prepare(query: "SELECT RM FROM usuario WHERE Email = :email LIMIT 1");
    $stmt->execute(params: [":email" => $email]);
    $usuario = $stmt->fetch(mode: PDO::FETCH_ASSOC);

    if (!$usuario) {
        echo json_encode(value: ["status" => "erro", "mensagem" => "Email não encontrado"]);
        exit;
    }

    // ♦ Gera um código de 6 dígitos
    $codigo = str_pad(string: random_int(min: 0, max: 999999), length: 6, pad_string: "0", pad_t: STR_PAD_LEFT);

    // ♦ Atualiza no banco
    $stmt = $pdo->prepare(query: "UPDATE usuario SET codigo_recuperacao = :codigo WHERE Email = :email");
    $stmt->execute(params: [":codigo" => $codigo, ":email" => $email]);

    // ♦ Monta a mensagem do e-mail
    $assunto = "Código de Recuperação - S.C.O.R.E.";
    $mensagem = "Seu código de recuperação é: $codigo\n\nUse este código para redefinir sua senha.";

    // ♦ Cabeçalhos do e-mail
    $headers = "From: S.C.O.R.E <score_empresa01@gmail.com>\r\n";
    $headers .= "Reply-To: score_empresa01@gmail.com\r\n";
    $headers .= "MIME-Version: 1.0\r\n";
    $headers .= "Content-Type: text/plain; charset=UTF-8\r\n";
    $headers .= "X-Mailer: PHP/" . phpversion();
}

```

FONTE – AUTORIA PRÓPRIA (2025)

Figura 52 - Código da Tela Esqueci senha 4

```

// ♦ Envia o email
if (mail(to: $email, subject: $assunto, message: $mensagem, additional_headers: $headers)) {
    echo json_encode(value: [
        "status" => "ok",
        "mensagem" => "Código enviado para o email cadastrado"
    ]);
} else {
    echo json_encode(value: [
        "status" => "erro",
        "mensagem" => "Não foi possível enviar o e-mail. Verifique a configuração do servidor."
    ]);
}

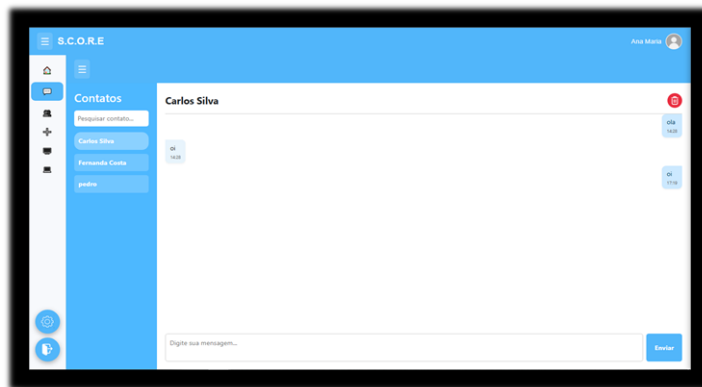
} catch (Exception $e) {
    echo json_encode(value: [
        "status" => "erro",
        "mensagem" => "Erro no servidor: " . $e->getMessage()
    ]);
}

```

FONTE – AUTORIA PRÓPRIA (2025)

3.4.8 Tela *Chat* Secretaria

Figura 53 - Tela *Chat* Secretaria

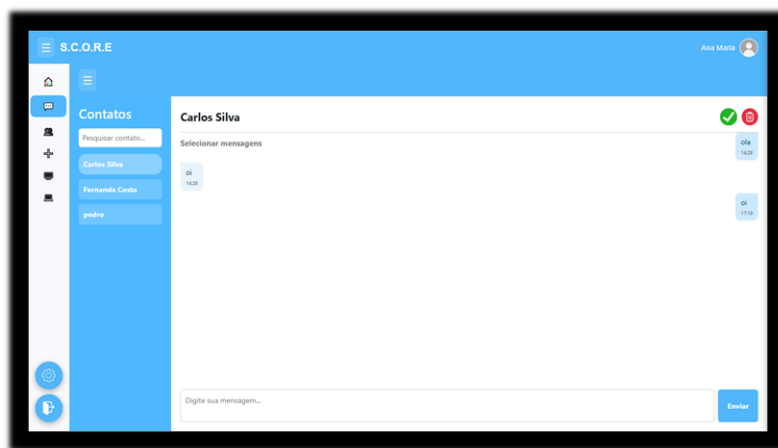


FONTE – AUTORIA PRÓPRIA (2025)

A Figura 53 apresenta a interface de comunicação interna do sistema S.C.O.R.E, destinada à troca de mensagens entre usuários. destinada à troca de mensagens entre usuários. No menu lateral esquerdo, sobre fundo azul, está a seção “Contatos” com barra de pesquisa e lista de usuários disponíveis, como “Carlos Silva”, “Fernanda Costa” e “Pedro”.

No centro, a conversa com “Carlos Silva” mostra mensagens e horários. Abaixo, o campo “Digite sua mensagem...” e o botão “Enviar”. No canto superior direito, “Ana Maria” indica o usuário logado.

logado. *Figura 54 - Tela Seleção de mensagens*

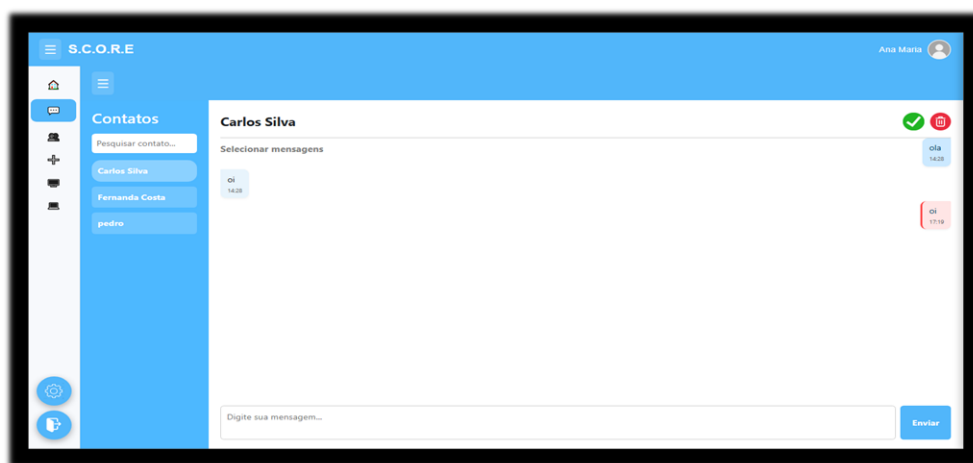


FONTE – AUTORIA PRÓPRIA (2025)

A Figura 54 apresenta a interface de *chat* do sistema S.C.O.R.E durante o processo de exclusão de mensagens. Ao selecionar o ícone de exclusão, representado por uma lixeira vermelha, o sistema entra automaticamente no modo de seleção, permitindo ao usuário escolher quais mensagens deseja remover. Nesse modo, surge na parte superior da tela a opção “Selecionar mensagens”, indicando que o *chat* está temporariamente voltado para a função de exclusão.

Cada mensagem exibida na conversa pode ser marcada individualmente, oferecendo maior controle sobre quais conteúdos serão apagados. Além disso, novos ícones são apresentados no canto superior direito da área de mensagens: um ícone verde, que representa a confirmação da exclusão, e outro vermelho, responsável por cancelar a operação e retornar à visualização normal do *chat*.

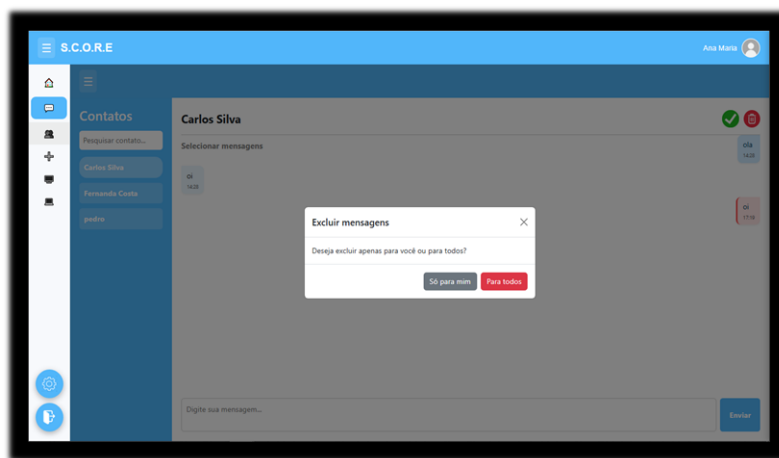
Figura 55 - Tela Mensagens Seleccionadas para exclusão



FONTE – AUTORIA PRÓPRIA (2025)

Nesta etapa A Figura 55, o sistema apresenta mensagens destacadas em cores distintas: as selecionadas para exclusão são realçadas, indicando que estão prontas para serem removidas. Os botões de confirmação e cancelamento permanecem visíveis na parte superior da tela, permitindo que o usuário confirme ou desista da ação.

Figura 56 - Tela Confirmação de exclusão



FONTE – AUTORIA PRÓPRIA (2025)

A Figura 56 exibe a janela de confirmação da exclusão. Um *pop-up* centralizado traz o título “Excluir mensagens” e a pergunta: “Deseja excluir apenas para você ou para todos?”. Abaixo, dois botões oferecem as opções “Só para mim” e “Para todos”. Caso o usuário queira cancelar a operação, é possível clicar novamente no ícone de lixeira ou no botão “X” localizado no canto superior direito da janela.

O arquivo inicia com comandos *PHP* que executam a função *session_start()* e verificam se a variável de sessão `$_SESSION['rm']` está definida. Caso contrário, ocorre o redirecionamento automático para a página de *login*, garantindo que apenas usuários autenticados possam acessar o *chat*. Em seguida, são carregadas informações do usuário *logado*, como nome e *e-mail*, e realizadas consultas ao banco de dados por meio de *PDO* para listar os demais usuários registrados.

Na estrutura *HTML*, a página é dividida em dois blocos principais: a barra lateral (`<aside class="sidebar">`) e o corpo da conversa (`<main class="main-content">`). A seção lateral contém o título “Contatos”, um campo de busca (`<input type="text" id="pesquisaContato">`) e botões gerados dinamicamente com *PHP* para cada usuário retornado do banco. Cada botão recebe o atributo *data-rm*, permitindo identificar o destinatário da conversa.

O painel principal exibe o cabeçalho da conversa ativa, implementado pela *div* `<div id="topoChat">`, que apresenta o nome do contato selecionado e ícones de ação

Figura 58 - Código Chat 2

```

<main class="main-content">
  <div id="topoChat">
    <h2 id="titulo">Selecione um contato:</h2>
    <div id="acoesMensagem">
      
      
    </div>
    <span id="msgSeleciao" style="display:none;">Selecionar mensagens</span>
  </div>

  <div id="chatbody" class="mensagens"></div>

  <form id="formEnvio">
    <textarea id="texto" placeholder="Digite sua mensagem..."></textarea>
    <button type="submit">Enviar</button>
  </form>
</main>

</div>
<!-- Modal para exclusão -->
<div class="modal fade" id="modalExclusao" tabindex="-1">
  <div class="modal-dialog modal-dialog-centered">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title">Excluir mensagens</h5>
        <button type="button" class="btn-close" data-bs-dismiss="modal"></button>
      </div>
      <div class="modal-body">
        Deseja excluir apenas para você ou para todos?
      </div>
      <div class="modal-footer">
        <button id="excluirIndico" class="btn btn-secondary" data-bs-dismiss="modal">Para mim</button>
        <button id="excluirTodos" class="btn btn-danger" data-bs-dismiss="modal">Para todos</button>
      </div>
    </div>
  </div>
</div>

<script>
  window.CHAT = {
    mensagem: {<!-- json_encode($mensagem) -->},
    nome: {<!-- json_encode($SESSIDM['nome']) -->}},
    contatoSelecionado: null
  }
</script>

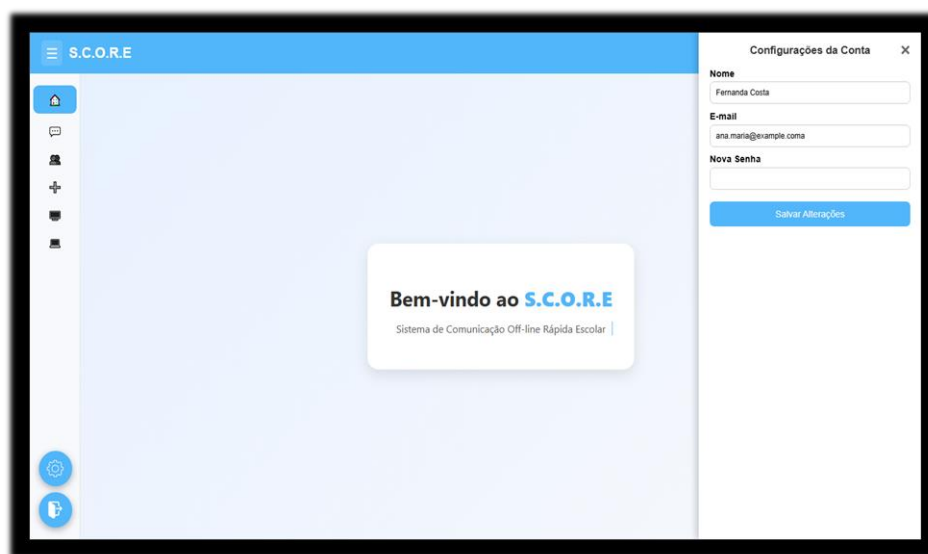
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"></script>
<script src="script2.js" defer"></script>
</body>
</html>

```

FONTE – AUTORIA PRÓPRIA (2025)

3.4.9 Tela Configuração

Figura 59 - Tela de Configuração



FONTE – AUTORIA PRÓPRIA (2025)

A Figura X apresenta a tela exibida ao clicar no ícone de configurações do sistema S.C.O.R.E. À direita, surge um painel lateral identificado como “*Configurações da Conta*”, contendo campos para edição dos dados do usuário *logado*.

Os campos disponíveis são “*Nome*”, “*E-mail*” e “*Nova Senha*”, permitindo que o usuário atualize suas informações pessoais e de acesso. Abaixo, há o botão “*Salvar Alterações*”, destacado em azul, que confirma as modificações realizadas. No restante da tela, mantém-se a mensagem central “*Bem-vindo ao S.C.O.R.E – Sistema de Comunicação Off-line Rápida Escolar*”, preservando o padrão visual e funcional do sistema.

O documento é construído com base no padrão *HTML5*, iniciando pela declaração `<!DOCTYPE html>` e pela tag `<html lang="pt-br">`, que define o idioma da página. No cabeçalho (`<head>`), são configurados os metadados, o título “*Home – S.C.O.R.E*” e o bloco de estilos internos (`<style>`), responsável pela formatação visual dos elementos. O estilo segue a identidade gráfica do sistema, com predominância da cor azul (`#51b6fa`), bordas arredondadas, espaçamento equilibrado e *layout* responsivo.

O corpo da página (`<body>`) é composto por três seções principais: o cabeçalho (`<header>`), a barra lateral de navegação (`<aside class="sidebar">`) e o conteúdo principal (`<main>`). O cabeçalho contém o logotipo, o botão de menu e as informações do usuário autenticado, incluindo uma imagem de perfil. A barra lateral apresenta os ícones de navegação do sistema, como “*Home*”, “*Chat*”, “*Usuários*”, “*Cadastrar Usuário*”, “*Cadastrar PC*” e “*Ver PCs*”, sendo que o item ativo é destacado por meio da classe `.active`, que aplica o fundo azul e o texto branco.

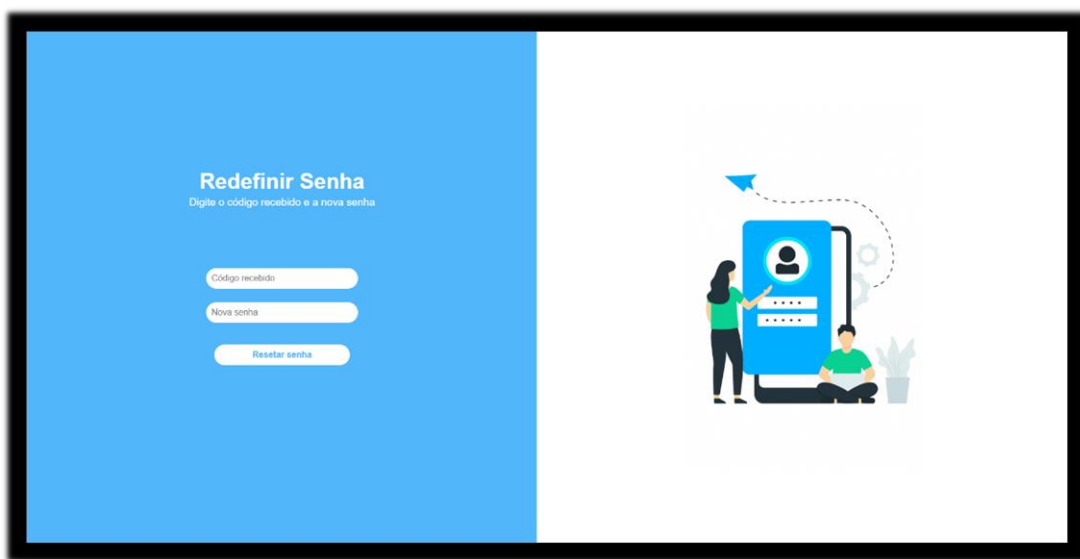
O ícone de configurações é implementado por meio de um botão flutuante (`<button class="btn-fab btn-config">`), posicionado no canto inferior esquerdo da interface. Ele exibe uma imagem representativa (`iconconfig2.png`) e, ao ser clicado, ativa o painel de configurações por meio da adição da classe `.active` ao elemento `<div class="config-panel">`. Esse painel desliza a partir do lado direito da tela e apresenta o título “*Configurações da Conta*”, seguido de um formulário com os campos “*Nome*”, “*E-mail*” e “*Nova Senha*”. O formulário utiliza o método *POST* e o atributo

3.4.10 Tela Redefinir Senha

A imagem 61 mostra a tela de redefinição de senha para a recuperação de acesso.

Essa tela possui os campos “Código recebido” e “Nova senha”, além do botão “Resetar senha”.

Figura 61 - Redefinir Senha



FONTE – AUTORIA PRÓPRIA (2025)

As figuras 62, 63 e 64 mostram o código que permitem que o usuário redefina sua senha usando o código de verificação enviado anteriormente por *e-mail*. Assim como na etapa de solicitação de código, essa funcionalidade é composta por um arquivo HTML que apresenta a interface de interação e por *scripts* em *JavaScript* e *PHP* que executam o processamento no servidor.

Estrutura da página HTML

O documento é definido em HTML5 com o idioma configurado para português do Brasil (pt-br) e a *meta tag* `<meta> viewport` que garante a responsividade em diferentes dispositivos. O título da aba do navegador é “Resetar Senha – S.C.O.R.E.” e o arquivo de *estilo* `style.css` é importado para aplicar a formatação visual.

Interface

O corpo da página é organizado em um `<div>` de classe `container` dividido em duas áreas.

- Lado esquerdo: contém um `<h1>` com o texto “Redefinir Senha” e um `<p>` que orienta o usuário a digitar o código recebido e a nova senha.
- Formulário: identificado pelo `id formResetar`, é um `<form>` com duas entradas. A primeira é um `<input>` de tipo `text` e `id` `codigo`, para o código de recuperação; a segunda é um `<input>` de tipo `password` e `id` `senha`, para a nova senha. Um `<button>` do tipo `submit` envia os dados. Logo abaixo há um `<div>` com `id` `mensagem`, que exibe as respostas do sistema e tem a cor ajustada dinamicamente.

- Lado direito: exibe, dentro de um `<div>` de classe `right-side`, uma imagem representativa da funcionalidade, definida pela `tag `. Funcionalidade `JavaScript` Um `<script>` no final da página adiciona um `event listener` ao `<form>` para capturar o envio. Ao clicar em “Resetar senha”, o `script` impede o comportamento padrão com `preventDefault`, obtém os valores dos `<input>` `codigo` e `senha` e verifica se foram preenchidos. Caso algum campo esteja vazio, escreve a mensagem “Preencha todos os campos.” no `<div>` `mensagem` e interrompe o processo. Se os campos estiverem corretos, é feita uma requisição assíncrona `fetch` para o arquivo `resetar_senha.php`, com `method` `POST`, cabeçalho `Content-Type: application/json` e corpo em `JSON` contendo o código e a nova senha. A resposta é convertida para `JSON` e exibida no `<div>` `mensagem`. Se o `status` retornado for “sucesso” e houver um endereço de redirecionamento, o `script` aguarda dois segundos e então altera a `location.href` para o valor indicado, levando o usuário à página final.

Processamento/PHP: O arquivo `resetar_senha.php` é responsável por atualizar a senha no banco de dados. Inicialmente define o `header` para retorno em `JSON` e cria uma conexão com o banco através de `PDO`, com tratamento de exceções. Em seguida, obtém os dados enviados em `JSON` por `file_get_contents("php://input")` e os converte para um `array` associativo. Caso o código ou a senha estejam vazios, retorna um `JSON` com `status` “erro”. Se os campos foram preenchidos, executa um `SELECT` para localizar o usuário com o `codigo_recuperacao` informado. Se não encontrar, devolve `JSON` com a mensagem “Código inválido ou expirado.”. Quando o usuário é encontrado, executa um `UPDATE` para definir a nova senha e limpar o

Figura 64 - Código Redefinir Senha 3

```

<?php
header(header: "Content-Type: application/json; charset=UTF-8");
//$host = "localhost 192.168.0.31";
try {
    $pdo = new PDO(dsn: "mysql:host=192.168.0.31;dbname=score;charset=utf8", username: "root", password: "");
    $pdo->setAttribute(attribute: PDO::ATTR_ERRMODE, value: PDO::ERRMODE_EXCEPTION);

    $dados = json_decode(json: file_get_contents(filename: "php://input"), associative: true);
    $codigo = $dados["codigo"] ?? null;
    $senha = $dados["senha"] ?? null;

    if (empty($codigo) || empty($senha)) {
        echo json_encode(value: ["status" => "erro", "mensagem" => "Preencha todos os campos."]);
        exit;
    }

    // Encontra o usuário pelo código
    $stmt = $pdo->xprepare(query: "SELECT RM FROM usuario WHERE codigo_recuperacao = :codigo LIMIT 1");
    $stmt->execute(params: [":codigo" => $codigo]);
    $usuario = $stmt->fetch(mode: PDO::FETCH_ASSOC);

    if (!$usuario) {
        echo json_encode(value: ["status" => "erro", "mensagem" => "Código inválido ou expirado."]);
        exit;
    }

    // Atualiza a senha e limpa o código
    $stmt = $pdo->xprepare(query: "UPDATE usuario SET Senha = :senha, codigo_recuperacao = NULL WHERE RM = :rm");
    $stmt->execute(params: [
        ":senha" => $senha, // (texto puro, como você pediu)
        ":rm" => $usuario["RM"]
    ]);

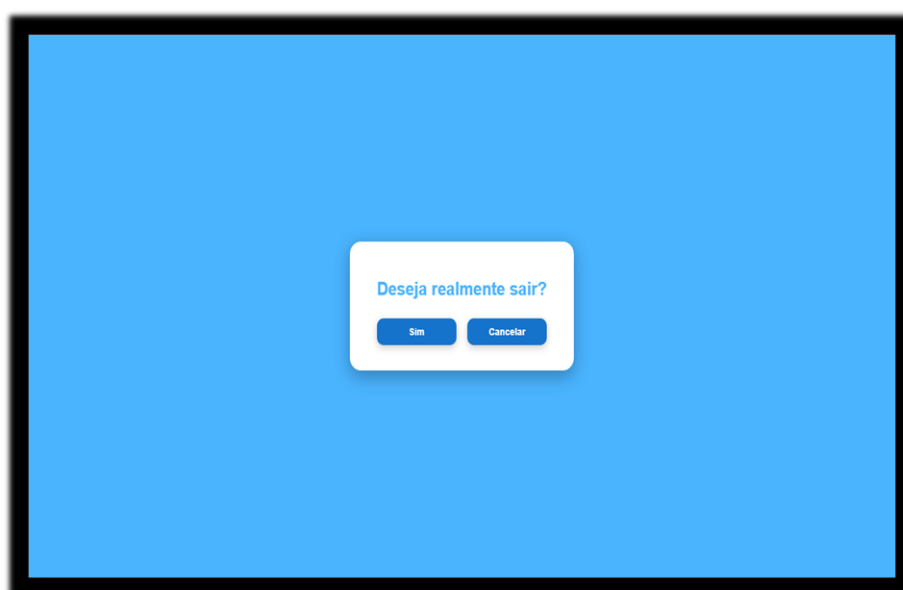
    // ⚠ Use caminho ABSOLUTO da sua app para não errar pasta:
    echo json_encode(value: [
        "status" => "sucesso",
        "mensagem" => "Senha redefinida com sucesso!",
        "redirect" => "/ScoreSeparado/Index.html"
    ]);
} catch (Exception $e) {
    echo json_encode(value: ["status" => "erro", "mensagem" => "Erro no servidor: " . $e->getMessage()]);
}

```

FONTE – AUTORIA PRÓPRIA (2025)

3.4.11 Tela de confirmação de saída do sistema S.C.O.R.E

Figura 65 - Tela de confirmação de saída do sistema



FONTE – AUTORIA PRÓPRIA (2025)

A Figura 65 exibe a janela de confirmação que surge ao tentar encerrar a sessão no sistema S.C.O.R.E. Essa etapa tem como objetivo garantir que o usuário confirme a ação antes de efetuar o *logout*, evitando saídas acidentais. No centro da tela, há uma caixa de diálogo com o texto “Deseja realmente sair?”, apresentado em tom azul destacado, reforçando a identidade visual do sistema. Logo abaixo da mensagem, encontram-se dois botões de ação: “Sim”, que confirma o encerramento da sessão e realiza o *logout*, e “Cancelar”, que permite desistir da operação e retornar ao ambiente principal do sistema.

O fundo azul uniforme de toda a página complementa o padrão visual da aplicação, transmitindo coerência estética e continuidade com as demais interfaces do sistema. Essa padronização cromática reforça a identidade visual do S.C.O.R.E, valorizando o aspecto profissional e moderno da aplicação.

O documento HTML inicia-se com a declaração `<!DOCTYPE html>` e a definição do idioma `lang="pt-br"`, assegurando compatibilidade com navegadores modernos e a padronização da linguagem utilizada na interface. No cabeçalho (`<head>`), são definidos os metadados de codificação e responsividade, além do título da página, “*Logout - S.C.O.R.E*”, e do bloco de estilos internos (`<style>`), responsável por determinar a aparência visual do componente.

O estilo aplicado segue o padrão cromático do sistema, predominando o tom azul `#4ab4ff`, cor que simboliza a identidade visual do S.C.O.R.E. Essa escolha transmite uma sensação de confiança e harmonia, mantendo a consistência visual com as demais telas da aplicação.

No corpo da página (`<body>`), é utilizado o modelo de *layout* Flexbox, que centraliza o conteúdo tanto vertical quanto horizontalmente, garantindo equilíbrio e responsividade. O conteúdo principal está estruturado dentro da `<div class="logout-container">`, uma caixa de diálogo posicionada sobre o fundo azul. Essa caixa apresenta fundo branco, bordas arredondadas, sombra suave e texto em azul, o que cria contraste visual agradável e destaca a mensagem de confirmação.

O contêiner exibe o título centralizado “Deseja realmente sair?” e, logo abaixo, dois botões — “Sim” e “Cancelar” — estilizados com bordas arredondadas, cor azul e efeitos visuais de hover, destacando a ação e melhorando a interação do usuário.

O script JavaScript da página controla o logout, acionando a função `logout()` ao clicar em “Sim” para encerrar a sessão e redirecionar à página inicial, enquanto `cancelLogout()` retorna à página anterior ao clicar em “Cancelar”, garantindo que o usuário tenha controle seguro sobre a ação.

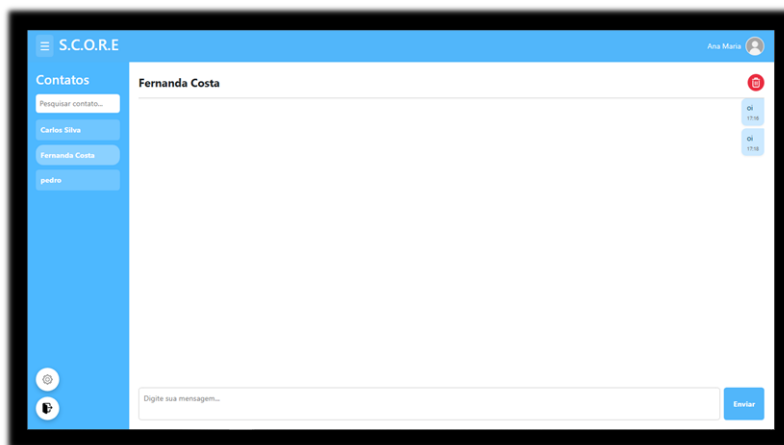
Figura 66 - Código tela de confirmação de saída do sistema

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <title>Logout - S.C.O.R.E</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <style>
  </style>
</head>
<body>
  <div class="logout-container">
    <h1>Deseja realmente sair?</h1>
    <div class="button-wrapper">
      <button class="btn btn-yes" onclick="logout()">Sim</button>
      <button class="btn btn-no" onclick="cancelLogout()">Cancelar</button>
    </div>
  </div>
  <script>
    function logout() {
      window.location.href = "../index.html";
    }
    function cancelLogout() {
      window.history.back();
    }
  </script>
</body>
</html>
```

FONTE – AUTORIA PRÓPRIA (2025)

3.4.12 Tela de *Chat* do Sistema S.C.O.R.E Usuário

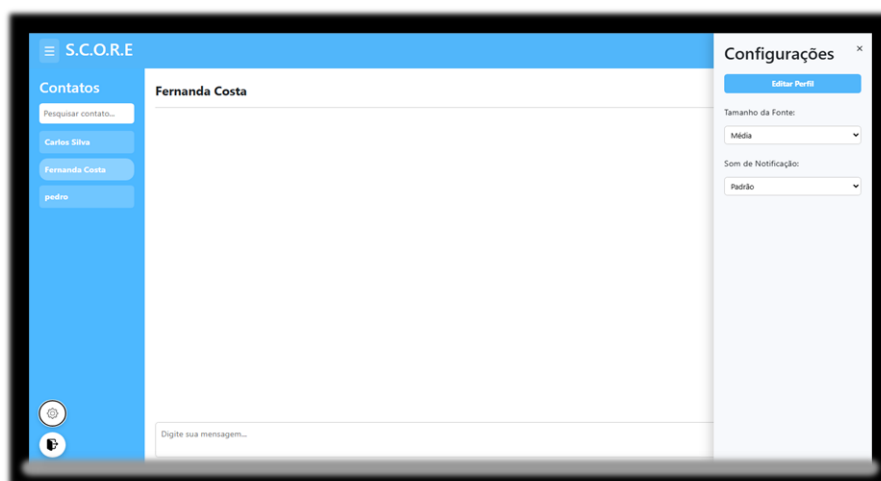
Figura 67 - Tela de *Chat* do Sistema S.C.O.R.E Usuário



FONTE – AUTORIA PRÓPRIA (2025)

A Figura 67 apresenta a interface principal do chat do sistema S.C.O.R.E., exibindo a lista de contatos à esquerda e a área de mensagens ao centro, permitindo que o usuário visualize facilmente suas conversas ativas. No exemplo, o usuário “Ana Maria” está em comunicação com “Fernanda Costa”, com o histórico de mensagens exibido de forma clara e organizada. A interface inclui um campo de texto para digitar novas mensagens e um botão “Enviar” localizado no canto inferior direito, facilitando a interação em tempo real. Além disso, a interface mantém o padrão visual do sistema, com cores, fontes e elementos consistentes, proporcionando uma experiência intuitiva e responsiva em diferentes dispositivos. Funcionalidades adicionais, como rolagem automática e destaque de mensagens recentes, garantem que a comunicação seja contínua e eficiente.

Figura 68 - Tela de Configurações do *Chat*



FONTE – AUTORIA PRÓPRIA (2025)

Na Figura 68, é possível observar o painel lateral de configurações do sistema S.C.O.R.E., que oferece opções para personalizar a experiência do usuário. O painel permite alterar o tamanho da fonte das mensagens, com quatro opções disponíveis: pequena, média, grande e muito grande, garantindo melhor legibilidade conforme a preferência do usuário. Também é possível selecionar o som de notificação entre as opções padrão, som 1 e som 2, proporcionando alertas personalizados para novas mensagens. Além disso, o painel inclui o botão “Editar Perfil”, que abre um formulário para atualização dos dados pessoais do usuário, permitindo modificar informações

como nome, e-mail e senha de forma prática e segura. Todo o painel segue o padrão visual e interativo do sistema, mantendo consistência e facilidade de uso.

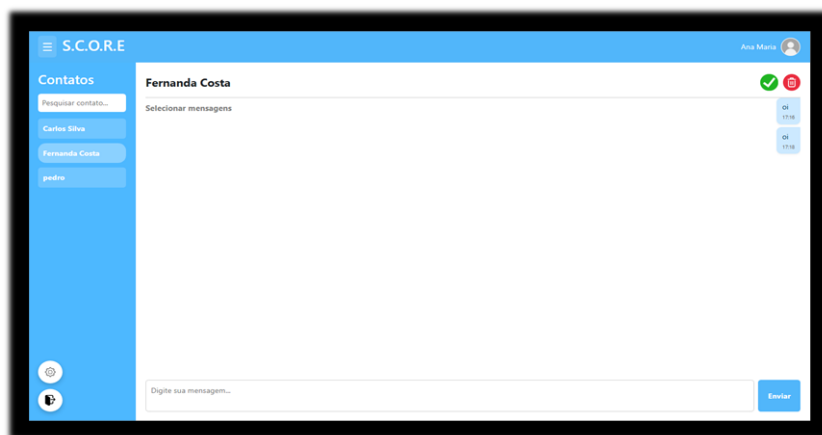
Figura 69 - Tela edição de Perfil do Usuário

The image shows a mobile application interface for editing a user profile. The main window is titled 'Editar Usuário' and contains the following fields: 'RM' with the value '101', 'Nome' with 'Ana Maria', 'CPF' with a masked value '*****', 'Tipo' with 'Professor', 'Email' with 'ana.maria@example.com', 'Confirmar Email' with 'ana.maria@example.com', 'Senha' with a masked value '***** (digite em branco para não alterar)', and 'Confirmar Senha' with a masked value. A blue 'Confirmar' button is located at the bottom of the form. The background shows a chat interface for 'Fernanda Costa' and a 'Configurações' (Settings) panel on the right with options for 'Tamanho da Fonte' (Média) and 'Som de Notificação' (Padrão).

FONTE – AUTORIA PRÓPRIA (2025)

A Figura 69 exibe o formulário “Editar Usuário”, onde é possível modificar informações como “RM”, “nome”, “CPF”, “tipo de usuário”, “e-mail” e “senha”. O *layout* é composto por campos de texto organizados verticalmente e o botão “Confirmar” ao final do formulário.

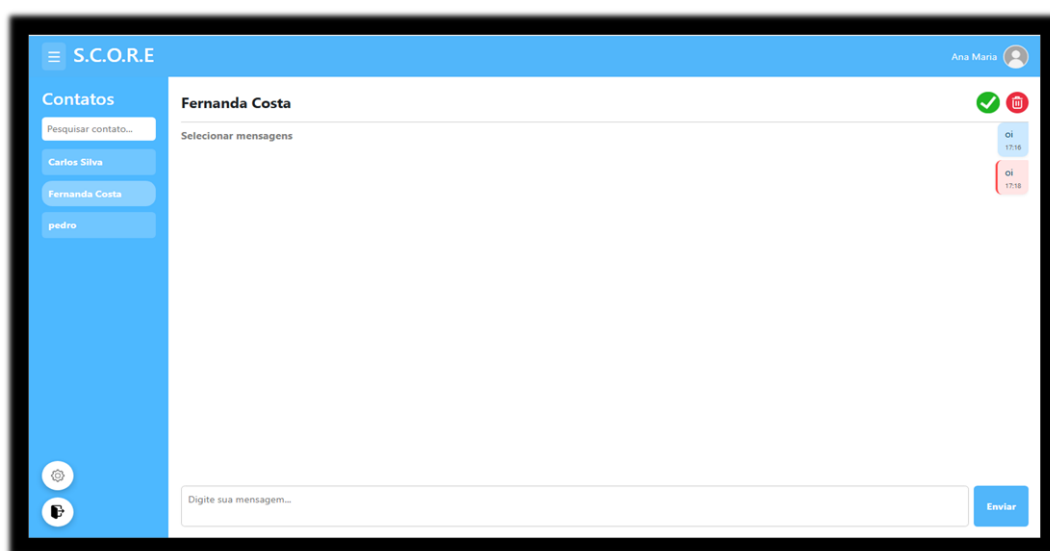
Figura 70 - Tela de Chat com Mensagens Selecionadas



FONTE – AUTORIA PRÓPRIA (2025)

A Figura 70 apresenta a interface do *chat* no sistema S.C.O.R.E., onde o usuário “Ana Maria” interage com “Fernanda Costa”. Nesta tela, é possível observar duas mensagens marcadas, indicadas pelos ícones de seleção (em verde e vermelho) ao lado das mensagens enviadas. Essa funcionalidade permite ao usuário escolher mensagens específicas para realizar ações posteriores, como exclusão ou arquivamento. O *layout* mantém o mesmo padrão visual do sistema, com o painel de contatos à esquerda, a área de mensagens ao centro e o campo de digitação na parte inferior.

Figura 71 - Tela seleção de mensagens para exclusão

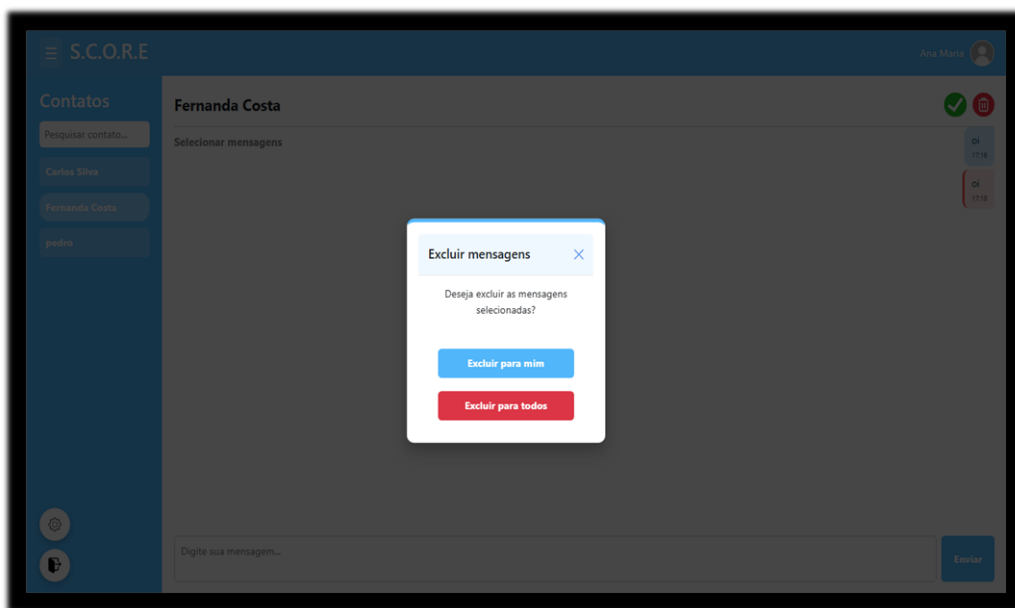


FONTE – AUTORIA PRÓPRIA (2025)

Na Figura 71, o usuário realiza a seleção de mensagens específicas dentro da conversa no sistema S.C.O.R.E. As mensagens escolhidas para exclusão são destacadas visualmente com a cor vermelha, permitindo identificar de forma clara quais conteúdos estão prontos para serem removidos.

Esse destaque colorido fornece *feedback* visual imediato, ajudando o usuário a confirmar sua seleção antes de executar a ação de exclusão. Dessa forma, o sistema previne erros acidentais e garante maior controle e precisão no gerenciamento das mensagens dentro do *chat*, contribuindo para uma experiência de uso mais segura e organizada.

Figura 72 - Tela Confirmação de Exclusão de Mensagens



FONTE – AUTORIA PRÓPRIA (2025)

A Figura 72 apresenta uma janela *modal* com o título “Excluir mensagens”, oferecendo as opções “Excluir para mim” e “Excluir para todos”, além do botão para cancelar a ação. Essa tela garante segurança no processo de remoção de mensagens, evitando exclusões acidentais.

O código inicia-se com um bloco em *PHP* que gerencia a sessão do usuário. Caso o identificador “RM” não esteja definido, ocorre o redirecionamento automático para a página de *login*, garantindo a segurança do acesso. Em seguida, são executadas consultas *SQL* para buscar os contatos disponíveis (demais usuários cadastrados) e os dados completos do usuário *logado* — informações que são armazenadas em variáveis de sessão, incluindo nome, CPF, tipo e *e-mail*.

A estrutura visual é definida a partir da declaração `<!DOCTYPE html>` e da tag `<html lang="pt-br">`, que assegura compatibilidade e acessibilidade. No cabeçalho (`<head>`), são incluídas as referências ao arquivo de estilos (*style.css*), ao *framework Bootstrap 5.3.2*, e as configurações de metadados e título da página (“*Chat Oficial*”).

O corpo do documento (`<body>`) é dividido em três seções principais: cabeçalho, painel lateral (contatos) e área de mensagens.

O cabeçalho (*<header>*) contém o logotipo e o nome do sistema S.C.O.R.E, além do nome e imagem do usuário *logado*. A barra lateral (*<aside class="sidebar">*) exibe a lista de contatos, acompanhada de um campo de busca e de dois botões flutuantes: um para abrir as configurações do *chat* e outro para realizar o *logout* do sistema.

A área central (*<main class="main-content">*) apresenta o conteúdo principal da conversa. Nela, o título do *chat* exibe o nome do contato selecionado, e um conjunto de ícones permite ações de gerenciamento de mensagens, como seleção e exclusão. O corpo do *chat* (*<div id="chatBody">*) exibe as mensagens trocadas entre os usuários, enquanto o formulário de envio (*<form id="formEnvio">*) contém um campo *textarea* para digitação e o botão “Enviar”, que dispara a transmissão da mensagem.

Além do *chat*, o código define o painel de configurações lateral (*<div class="sidebar-config">*), no qual o usuário pode ajustar o tamanho da fonte das mensagens e selecionar o som de notificação desejado. Esse painel também oferece o botão “Editar Perfil”, que abre uma janela *modal* de edição de dados pessoais.

O formulário de edição de perfil (*<div id="modalEditar">*) permite atualizar informações como RM, nome, CPF, tipo, *e-mail* e senha. Campos como RM, CPF e tipo são exibidos como somente leitura, reforçando a integridade dos dados do sistema. O botão “Confirmar” conclui o processo de atualização e exibe mensagens de retorno dinâmicas por meio do elemento *id="msgEditar"*.

Há também a janela *modal* de exclusão de mensagens (*<div id="modalExcluir">*), que apresenta as opções “Excluir para mim” e “Excluir para todos”, garantindo que o usuário confirme a ação antes da remoção definitiva das mensagens selecionadas.

Por fim, o *script JavaScript* define o objeto global *window.CHAT*, que armazena os dados do usuário *logado* e gerencia a interação entre o *frontend* e o *backend*. A integração com o *Bootstrap* e o arquivo externo *script.js* garante o funcionamento dos *modais*, dos botões e das interações assíncronas do *chat*.

Dessa forma, o código consolida uma interface robusta e responsiva, alinhada à identidade visual do sistema S.C.O.R.E., permitindo ao usuário trocar mensagens,

personalizar o ambiente de uso, editar seus dados e gerenciar conversas de forma prática e intuitiva.

Figura 73 - Código tela *Chat Usuário 1*

```

<?php
session_start();
require "conexao.php";

if (!isset($_SESSION['rm'])) {
    header("Location: login.php");
    exit;
}

$meuRM = $_SESSION['rm'];
$meuNome = $_SESSION['nome'] ?? '';

// Buscar contatos (outros usuários)
$sql = "SELECT RM, Nome FROM usuario WHERE RM <> ?";
$stmt = $pdo->prepare($sql);
$stmt->execute([$meuRM]);
$contatos = $stmt->fetchAll(PDO::FETCH_ASSOC);

// Buscar dados completos do usuário logado
$sqlUsuario = "SELECT RM, Nome, CPF, Tipo, Email FROM usuario WHERE RM = ?";
$stmtUsuario = $pdo->prepare($sqlUsuario);
$stmtUsuario->execute([$meuRM]);
$usuario = $stmtUsuario->fetch(PDO::FETCH_ASSOC);

// Atualiza a sessão caso ainda não tenha os dados
$_SESSION['cpf'] = $usuario['CPF'] ?? $_SESSION['cpf'] ?? '';
$_SESSION['tipo'] = $usuario['Tipo'] ?? $_SESSION['tipo'] ?? '';
$_SESSION['email'] = $usuario['Email'] ?? $_SESSION['email'] ?? '';

// Prepara o CPF mascarado
$cpfReal = $_SESSION['cpf'] ?? '';
$cpfMasked = $cpfReal ? str_repeat('*', strlen(preg_replace('/\D/', '', $cpfReal))) : '';
?>

<!DOCTYPE html>
<html lang="pt-br">

<head>
<meta charset="UTF-8">
<title>Chat Oficial</title>
<link rel="stylesheet" href="style.css">
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet">
</head>

```

FONTE – AUTORIA PRÓPRIA (2025)

Figura 74 - Código tela *Chat Usuário 2*

```

<body>

<header>
<div class="logo-container">
<span id="menu-toggle" class="menu-toggle"></span>
<h2>S.C.O.R.E</h2>
</div>
<div class="user-container">
<span><?php echo htmlspecialchars($meuNome);</span>

</div>
</header>
<div class="layout">
<aside class="sidebar" id="sidebar">
<h3>Contatos</h3>
<input type="text" id="pesquisaContato" placeholder="Pesquisar contato...">
<?php foreach ($contatos as $c):>
<button class="btn-computador" data-rm="<?php echo $c['RM'];>">
<?php echo htmlspecialchars($c['Nome']);>
</button>
</?php endforeach;
</div>
<div class="sidebar-buttons">
<button class="btn-flutuante btn-config" id="openConfig" aria-label="Configurações">

</button>
<form action="logout.php">
<button class="btn-flutuante btn-logout" aria-label="Sair">

</button>
</form>
</div>
</aside>

```

FONTE – AUTORIA PRÓPRIA (2025)

Figura 75 - Código tela *Chat* Usuário 3

```

<main class="main-content">
  <div id="topoChat">
    <h2 id="chatTitulo">Selecione um contato:</h2>
    <div id="acoesMensagens">
      
      
    </div>
    <span id="msgSelecao">Selecionar mensagens</span>
  </div>

  <div id="chatBody" class="mensagens"></div>
  <form id="formEnvio">
    <textarea id="texto" placeholder="Digite sua mensagem..."></textarea>
    <button type="submit">Enviar</button>
  </form>
</main>
</div>

<!-- Sidebar de Configurações -->
<div class="sidebar-config" id="sidebarConfig">
  <span class="close-sidebar" id="closeSidebar">&times;</span>
  <h2>Configurações</h2>
  <button class="btn-editar" id="openEditar">Editar Perfil</button>

  <label for="tamanhoFonte">Tamanho da Fonte:</label>
  <select id="tamanhoFonte">
    <option value="14px">Pequena</option>
    <option value="16px" selected>Média</option>
    <option value="18px">Grande</option>
    <option value="30px">Muito Grande</option>
  </select>

  <label for="somNotificacao">Som de Notificação:</label>
  <select id="somNotificacao">
    <option value="padrao">Padrão</option>
    <option value="som1">Som 1</option>
    <option value="som2">Som 2</option>
  </select>
</div>

```

FONTE – AUTORIA PRÓPRIA (2025)

Figura 76 - Código tela *Chat* Usuário 4

```

<!-- Model Editor Usuario -->
<div id="modelEditar" class="modal" style="display:none">
  <div class="modal-content">
    <div style="display:flex; justify-content:space-between; align-items:center; margin-bottom:10px;">
      <h2 style="margin:0; font-size:1.3em;">Editor Usuario</h2>
      <span id="closeEditar" style="cursor:pointer; font-size:1.5em; font-weight:bold;">&times;</span>
    </div>
    <hr style="border:none; border-top:1px solid #ccc; margin-bottom:20px;">
    <form id="formEditar" style="display:flex; flex-direction:column; gap:15px;">
      <div>
        <label for="idUser" class="label"></label>
        <input type="text" name="ru" id="idUser" value=""> <!-- htmlspecialchars($usuario['ru']) --> <!-- $ru -->
        <input type="text" name="nome" id="nomeUser" value=""> <!-- htmlspecialchars($usuario['nome']) --> <!-- $nome -->
        <input type="hidden" name="cpf" id="cpfUser" value=""> <!-- htmlspecialchars($cpfReal) -->
      </div>
      <div>
        <label for="cpfUser" class="label"></label>
        <input type="text" id="cpfMasked" value=""> <!-- htmlspecialchars($cpfMasked) --> <!-- $cpf -->
        <input type="hidden" name="cpf" id="cpfUser" value=""> <!-- htmlspecialchars($cpfReal) -->
      </div>
      <div>
        <label for="tipoUser" class="label"></label>
        <input type="text" name="tipo" id="tipoUser" value=""> <!-- htmlspecialchars($usuario['tipo']) --> <!-- $tipo -->
        <input type="text" name="email" id="emailUser" value=""> <!-- htmlspecialchars($usuario['email']) --> <!-- $email -->
      </div>
      <div>
        <label for="confirmarEmail" class="label"></label>
        <input type="text" name="confirm_email" id="confirmarEmail" placeholder="Confirme seu email" value=""> <!-- htmlspecialchars($usuario['email']) --> <!-- $email -->
      </div>
      <div>
        <label for="senhaUser" class="label"></label>
        <input type="password" name="senha" id="senhaUser" placeholder="***** (Digite em branco para não alterar)" value="">
      </div>
    </form>
  </div>
</div>

```

FONTE – AUTORIA PRÓPRIA (2025)

Figura 77 - Código tela Chat Usuário 5

```

<div>
  <label for="confirmaSenhaUser">Confirma Senha</label>
  <input type="password" name="confirma_senha" id="confirmaSenhaUser"
    placeholder="Confirma a sua senha"
    style="width:100%; padding:8px; border-radius:5px; border:1px solid #ccc;">
</div>
<button type="submit"
  style="padding:10px; border:none; border-radius:5px; background:#007bff; color:#fff; font-weight:bold; cursor:pointer; transition:0.2s;">Confirmar</button>
</form>
</div>
</div>
<!-- Modal de exclusão -->
<div class="modal fade" id="modalExcluir" tabindex="-1">
  <div class="modal-dialog">
    <div class="modal-content text-center">
      <div class="modal-header">
        <h5 class="modal-title">Excluir mensagens</h5>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Fechar"></button>
      </div>
      <div class="modal-body">
        <p>Deseja excluir as mensagens selecionadas?</p>
      </div>
      <div class="modal-footer d-flex justify-content-between">
        <button type="button" id="btnExcluirSoParamis" class="btn btn-secondary"
          data-bs-dismiss="modal">
          Excluir para mim
        </button>
        <button type="button" id="btnExcluirParaTodos" class="btn btn-danger"
          data-bs-dismiss="modal">
          Excluir para todos
        </button>
      </div>
    </div>
  </div>
</div>
</div>
<script>
  window._CHAT_ = {
    rmLogado: <?> json_encode($rmLog) ?>,
    nome: <?> json_encode($SESSION['nome'] ?? '') ?>,
    cpf: <?> json_encode($SESSION['cpf'] ?? '') ?>,
    tipo: <?> json_encode($SESSION['tipo'] ?? '') ?>,
    email: <?> json_encode($SESSION['email'] ?? '') ?>,
    contatoSelecionado: null
  };
</script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" defer</script>
<script src="script.js" defer</script>
</body>
</html>

```

FONTE – AUTORIA PRÓPRIA (2025)

CONSIDERAÇÕES FINAIS

O desenvolvimento do sistema S.C.O.R.E – Sistema de Comunicação *Off-line* Rápida Escolar proporcionou uma experiência prática significativa na aplicação dos conhecimentos adquiridos ao longo do curso, resultando em uma ferramenta funcional voltada para a comunicação interna no ambiente escolar. O projeto atingiu seus principais objetivos, demonstrando eficiência, boa usabilidade e coerência com a proposta inicial de oferecer um meio de comunicação simples, seguro e acessível mesmo em contextos com limitações de conexão.

O sistema foi planejado com foco na praticidade, segurança e facilidade de uso, apresentando uma interface intuitiva e coerente com o padrão visual definido. A aplicação mostrou-se eficiente em seu propósito principal: facilitar a troca de informações entre usuários de diferentes setores da instituição, reduzindo barreiras comunicativas e otimizando processos administrativos.

OBJETIVOS ALCANÇADOS

Entre os objetivos alcançados, destaca-se a implementação do *chat* funcional, que permite a troca de mensagens em tempo real entre os usuários por meio do cabeamento de rede sem precisar de internet. A funcionalidade de notificação sonora no *chat* da secretaria foi integrada com sucesso, alertando sobre novas mensagens recebidas e garantindo uma comunicação mais dinâmica.

Também foi estabelecido um limite de cinco mensagens consecutivas por envio, medida que evita sobrecarga no banco de dados e mantém a estabilidade do sistema. O sistema de *login* foi concluído e validado, assegurando que apenas usuários cadastrados tenham acesso à plataforma.

Além disso, a funcionalidade “Esqueci minha senha”, que envia instruções de recuperação por *e-mail*, foi implementada com êxito, ampliando a segurança e a autonomia dos usuários. Outras melhorias incluem a pesquisa de contatos, facilitando

a localização de usuários na lista, e a configuração do tamanho da fonte, que aumenta a acessibilidade e o conforto visual durante o uso do *chat*.

Essas implementações demonstram o cumprimento dos objetivos técnicos propostos e a consolidação de uma base sólida para futuras expansões do sistema.

OBJETIVOS NÃO ALCANÇADOS E CONTINUAÇÃO DO PROJETO

Apesar dos avanços obtidos, algumas funcionalidades planejadas não foram concluídas nesta versão do sistema, principalmente devido a limitações de tempo e priorização de recursos essenciais. Entre os objetivos não alcançados, destacam-se a implementação do envio de imagens e áudios, a inclusão de foto de perfil personalizada, a notificação sonora no chat do usuário (ainda restrita à secretaria) e a opção de alternar entre temas claro e escuro.

Esses recursos, embora não essenciais ao funcionamento atual do sistema, representam etapas futuras de aprimoramento. Suas implementações estão previstas como atualizações ou evoluções do projeto, buscando tornar o S.C.O.R.E ainda mais moderno, acessível e interativo.

Com essas melhorias, espera-se que o sistema se consolide como uma plataforma completa de comunicação institucional, capaz de atender plenamente às necessidades de interação entre professores, secretaria e demais setores escolares.

REFERÊNCIAS BIBLIOGRÁFICAS

A2WORKS. Evolução dos Servidores: Veja como foi. A2Works. Disponível em: <<https://a2works.com.br/noticias/evolucao-dos-servidores/>>. Acesso em: 9 abr. 2025.

A3A ENGENHARIA. Adequação de infraestrutura de rede: entenda a importância e como fazer. Disponível em: <<https://a3aengenharia.com.br/blog/adequacao-de-infraestrutura-de-rede/>>. Acesso em: 6 mar. 2025.

ALURA. Banco de dados. Alura. Disponível em: <<https://www.alura.com.br/artigos/banco-de-dados>>. Acesso em: 9 abr. 2025.

ALURA. Figma: o que é a ferramenta, design e como usar. Alura. Disponível em: <<https://www.alura.com.br/artigos/figma>>. Acesso em: 10 abr. 2025.

AMAZON WEB SERVICES. O que é Python?. AWS. Disponível em: <<https://aws.amazon.com/pt/what-is/python/>>. Acesso em: 10 abr. 2025.

APACHE FRIENDS. New XAMPP 20180315. Apache Friends Blog, 15 mar. 2018. Disponível em: <https://www.apachefriends.org/blog/new_xampp_20180315.html>. Acesso em: 9 abr. 2025.

APACHE FRIENDS. Página Inicial. Apache Friends. Disponível em: <https://www.apachefriends.org/pt_br/index.html>. Acesso em: 10 abr. 2025.

APACHE FRIENDS. Sobre o XAMPP. Apache Friends. Disponível em: <https://www.apachefriends.org/pt_br/about.html>. Acesso em: 9 abr. 2025.

AWARI. Python: a história do surgimento da linguagem de programação. Awari. Disponível em: <<https://awari.com.br/python-a-historia-do-surgimento-da-linguagem-de-programacao/>>. Acesso em: 9 abr. 2025.

AWS. Modelo de dados lógicos x físicos — Diferença na modelagem de dados. Amazon Web Services. Disponível em: <<https://aws.amazon.com/pt/what-is/logical-vs-physical-data-models/>>. Acesso em: 9 abr. 2025.

AWS. O que é modelagem de dados? – Explicação sobre modelagem de dados. Amazon Web Services. Disponível em: <<https://aws.amazon.com/pt/what-is/data-modeling/>>. Acesso em: 9 abr. 2025.

BRASIL ESCOLA. A indústria contemporânea no Brasil. Disponível em: <<https://brasilecola.uol.com.br/brasil/a-industria-contemporanea-no-brasil.htm#:~:text=A%20maior%20concentra%C3%A7%C3%A3o%20de%20ind%C3%BAstrias,processo%20de%20industrializa%C3%A7%C3%A3o%20teve%20in%C3%ADcio>>. Acesso em: 6 jun. 2025.

BRASIL. Instituto Brasileiro de Geografia e Estatística. Divisões Regionais do Brasil. Disponível em: <<https://www.ibge.gov.br/geociencias/cartas-e-mapas/redes-geograficas/15778-divisoes-regionais-do-brasil.html>>. Acesso em: 6 jun. 2025.

BRASIL. Instituto Brasileiro de Geografia e Estatística. Pesquisa Industrial Anual (PIA). Disponível em: <<https://www.ibge.gov.br/estatisticas/economicas/industria/9042-pesquisa-industrial-anual.html>>. Acesso em: 6 jun. 2025.

BRASIL. Lei nº 12.965, de 23 de abril de 2014. Marco Civil da Internet. Diário Oficial da União: seção 1, Brasília, DF, 24 abr. 2014.

BRASIL. Lei nº 13.116, de 20 de abril de 2015. Lei Geral das Antenas. Diário Oficial da União: seção 1, Brasília, DF, 22 abr. 2015.

BRASIL. Lei nº 13.709, de 14 de agosto de 2018. Lei Geral de Proteção de Dados Pessoais. Diário Oficial da União: seção 1, Brasília, DF, 15 ago. 2018.

BRASIL. Lei nº 9.296, de 24 de julho de 1996. Regula a interceptação de comunicações telefônicas. Diário Oficial da União: seção 1, Brasília, DF, 25 jul. 1996.

BRASIL. Lei nº 9.472, de 16 de julho de 1997. Dispõe sobre a organização dos serviços de telecomunicações. Diário Oficial da União: seção 1, Brasília, DF, 17 jul. 1997.

BRISANET. Servidor local ou em nuvem: como escolher a melhor opção?. Brisanet. Disponível em: <<https://www.brisanet.com.br/blog/servidor-local-ou-em-nuvem-como-escolher-a-melhor-opcao/>>. Acesso em: 10 abr. 2025.

CASTELANO. Aula 01 - Introdução a banco de dados. Castelano.com.br. Disponível em: <<https://castelano.com.br/site/aulas/bd/Aula%2001%20-%20Introdu%C3%A7%C3%A3o.pdf>>. Acesso em: 9 abr. 2025.

CHEN, W. Redes de Computadores. 4. ed. São Paulo: Pearson, 2013.

COGINITI. The Evolution of SQL: From SQL-86 to SQL:2023. Coginiti. Disponível em: <<https://www.coginiti.co/blog/the-evolution-of-sql-from-sql-86-to-sql-2023/>>. Acesso em: 9 abr. 2025.

COODESH. O que é MariaDB?. Coodesh. Disponível em: <<https://coodesh.com/blog/dicionario/o-que-e-mariadb/>>. Acesso em: 9 abr. 2025.

CRECERTO. Viabilidade Operacional. 2024.

DEV MEDIA. Introdução à Modelagem Conceitual. DevMedia. Disponível em: <<https://www.devmedia.com.br/introducao-a-modelagem-conceitual/14339>>. Acesso em: 9 abr. 2025.

DEVMEDIA. MER e DER: Modelagem de Bancos de Dados. DevMedia. Disponível em: <<https://www.devmedia.com.br/mer-e-der-modelagem-de-bancos-de-dados/14338>>. Acesso em: 9 abr. 2025.

DJANGO SOFTWARE FOUNDATION. Django Documentation. Disponível em: <<https://docs.djangoproject.com/>>. Acesso em: 6 mar. 2025.

DO ZERO AO TOPO. Canal no YouTube. Disponível em: <<https://www.youtube.com/c/dozeroaotopo>>. Acesso em: 6 jun. 2025.

DO ZERO AO TOPO. Perfil no Instagram. Disponível em: <https://www.instagram.com/dozeroaotopo_oficial/>. Acesso em: 6 jun. 2025.

ENDEAVOR BRASIL. Canal no YouTube. Disponível em: <<https://www.youtube.com/cloud/endeavorbrasil>>. Acesso em: 6 jun. 2025.

ENDEAVOR BRASIL. Organização. Disponível em: <<https://endeavor.org.br/>>. Acesso em: 6 jun. 2025.

ENDEAVOR BRASIL. Perfil no Instagram. Disponível em: <<https://www.instagram.com/endeavorbrasil/>>. Acesso em: 6 jun. 2025.

FLORENTINO, Pedro Ivo. A origem e evolução dos bancos de dados. Medium, 2021. Disponível em: <<https://medium.com/@pedroivoflorentino/a-origem-e-evolu%C3%A7%C3%A3o-dos-bancos-de-dados-72be6480c94>>. Acesso em: 9 abr. 2025.

GITHUB. brModelo - Official Versions. GitHub. Disponível em: <<https://github.com/gbd-ufsc/brModelo-Official-Versions>>. Acesso em: 9 abr. 2025.

GUIMARÃES FILHO, Edvaldo B. A História dos Servidores Web e o Papel de Tim Berners-Lee. Edvaldo Guimarães, 12 set. 2024. Disponível em: <<https://edvaldoguimaraes.com.br/2024/09/12/a-historia-dos-servidores-web-e-o-papel-de-tim-berners-lee/>>. Acesso em: 9 abr. 2025.

HARVE. O que é SQL?. Harve. Disponível em: <<https://harve.com.br/blog/analise-de-dados/o-que-e-sql/>>. Acesso em: 9 abr. 2025.

HASHTAG TREINAMENTOS. O que é Python?. Hashtag Treinamentos. Disponível em: <<https://www.hashtagtreinamentos.com/o-que-e-python>>. Acesso em: 9 abr. 2025.

HEY INVESTIDOR. Canal no YouTube. Disponível em: <<https://www.youtube.com/@HeyInvestidor>>. Acesso em: 6 jun. 2025.

HEY INVESTIDOR. Perfil no Instagram. Disponível em: <<https://www.instagram.com/heyinvestidor/>>. Acesso em: 6 jun. 2025.

HIPP, D. R. SQLite Documentation. Disponível em: <<https://www.sqlite.org/docs.html>>. Acesso em: 5 mar. 2025.

IBM. Socket addresses in TCP/IP. Disponível em: <<https://www.ibm.com/docs/pt-br/aix/7.3>>. Acesso em: 6 mar. 2025.

IED. Saiba o que é design e sua importância. IED – Istituto Europeo di Design. Disponível em: <<https://ied.edu.br/noticias/saiba-o-que-e-design-e-sua-importancia>>. Acesso em: 10 abr. 2025.

INFORMER. XAMPP Versions. Informer. Disponível em: <<https://xampp.informer.com/versions/>>. Acesso em: 9 abr. 2025.

ISO/IEC. Programming language C++. Disponível em: <<https://isocpp.org/>>. Acesso em: 5 mar. 2025.

JOANPAON. XAMPP 1.8.3 Beta Released. Joanpaon (WordPress), 22 jul. 2013. Disponível em: <<https://joanpaon.wordpress.com/2013/07/22/xampp-1-8-3-beta-released/>>. Acesso em: 10 abr. 2025.

KOTLER, Philip; KELLER, Kevin Lane. Administração de marketing. 14. ed. São Paulo: Pearson Prentice Hall, 2012.

KROENKE, D. M.; AUER, D. P. Database Processing: Fundamentals, Design, and Implementation. 12. ed. Upper Saddle River: Pearson, 2015.

LAS CASAS, Alexandre Luzzi. Marketing de relacionamento: conceitos e aplicações. 2. ed. São Paulo: Atlas, 2012.

LEARN SQL. History of SQL Standards. LearnSQL. Disponível em: <<https://learnsql.com/blog/history-of-sql-standards/>>. Acesso em: 10 abr. 2025.

LEYENDECKER, Davi. Os três níveis da modelagem de dados: conceitual, lógico e físico. DIO. Disponível em: <<https://web.dio.me/articles/os-tres-niveis-da-modelagem-de-dados-conceitual-logico-e-fisico>>. Acesso em: 9 abr. 2025.

LOCAWEB. Servidor web: entenda o que é e como funciona. Locaweb Blog. Disponível em: <<https://www.locaweb.com.br/blog/temas/codigo-aberto/servidor-web/>>. Acesso em: 9 abr. 2025.

LUCID. Viabilidade de cronograma. 2025.

MARIADB. Release Notes. MariaDB Knowledge Base. Disponível em: <<https://mariadb.com/kb/en/release-notes/>>. Acesso em: 10 abr. 2025.

MOGI GUAÇU. Lei nº 5.217, de 22 de outubro de 2019. Dispõe sobre o direito à informação e transparência da administração pública. Mogi Guaçu: Câmara Municipal, 2019.

MOZILLA DEVELOPER NETWORK. O que é um servidor web?. MDN Web Docs. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Learn_web_development/Howto/Web_mechanics/What_is_a_web_server>. Acesso em: 10 abr. 2025.

MSSERVER. Evolução dos Servidores. MS Server. Disponível em: <<https://mserver.com.br/evolucao-dos-servidores/>>. Acesso em: 9 abr. 2025.

MYSQL. MySQL Documentation. Disponível em: <<https://dev.mysql.com/doc/>>. Acesso em: 5 mar. 2025.

NEOENERGIA. Estudo de Viabilidade Técnica. 2025.

ORACLE. What is a database? Oracle Brasil. Disponível em: <<https://www.oracle.com/br/database/what-is-database/>>. Acesso em: 9 abr. 2025.

PROFISSÃO DBA. Histórico dos bancos de dados. Profissão DBA (WordPress). Disponível em: <<https://profissaodba.wordpress.com/historico-dos-bancos-de-dados/>>. Acesso em: 9 abr. 2025.

PYPI. PySide6. Python Package Index. Disponível em: <<https://pypi.org/project/PySide6/>>. Acesso em: 10 abr. 2025.

PYTHON SOFTWARE FOUNDATION. Flask Documentation. Disponível em: <<https://flask.palletsprojects.com/>>. Acesso em: 6 mar. 2025.

PYTHON SOFTWARE FOUNDATION. Python documentation by version. Disponível em: <<https://www.python.org/doc/versions/>>. Acesso em: 9 abr. 2025.

QT DOCUMENTATION. Qt for Python 6. Qt Documentation. Disponível em: <<https://doc.qt.io/qtforpython-6/>>. Acesso em: 9 abr. 2025.

QUERO BOLSA. O que é um servidor? Veja a definição e confira exemplos. Revista Quero. Disponível em: <<https://querobolsa.com.br/revista/o-que-e-um-servidor>>. Acesso em: 10 abr. 2025.

ROCK CONTENT. Marketing de influência: o que é e como fazer. Disponível em: <<https://rockcontent.com/br/blog/marketing-de-influencia/>>. Acesso em: 6 jun. 2025.

SÃO PAULO (Estado). Lei nº 17.334, de 27 de março de 2021. Dispõe sobre o cadastro para bloqueio do recebimento de chamadas de telemarketing. São Paulo: Assembleia Legislativa, 2021.

SIS4. brModelo. SIS4. Disponível em: <<http://www.sis4.com/brModelo/>>. Acesso em: 9 abr. 2025.

SOWEK, Carlos Alberto. Banco de dados: uma retrospectiva. Bate Byte, Governo do Paraná. Disponível em: <<https://www.batebyte.pr.gov.br/Pagina/Banco-de-dados-uma-retrospectiva>>. Acesso em: 9 abr. 2025.

TCC ARQUITETURA. Estudo de viabilidade técnica e econômica para TCC. Disponível em: <<https://tccarquitetura.com.br/>>. Acesso em: 6 mar. 2025.

THE Qt COMPANY. Qt Creator. Disponível em: <<https://www.qt.io/qt-creator>>. Acesso em: 5 mar. 2025.

THE Qt COMPANY. Qt Documentation. Disponível em: <<https://doc.qt.io/>>. Acesso em: 5 mar. 2025.

UPTODOWN. Versões antigas de Inkscape (Windows). Disponível em: <<https://inkscape.br.uptodown.com/windows/versions>>. Acesso em: 7 abr. 2025.

VEXPENSES. Estudo de Viabilidade Econômica. Disponível em: <<https://vexpenses.com.br/blog/estudo-de-viabilidade-economica/>>. Acesso em: 6 mar. 2025.

XAMPP.INFO. Download. XAMPP.info. Disponível em: <<https://xampp.info/download/>>. Acesso em: 10 abr. 2025.

