

Capítulo I – Gestor de Sistema: Gecurity

Isabella Aparecida Fraga
Michell Sales Leonne
Pedro Henrique Moraes
Rogério Dagnon Candeira
Fábio Takeshi Matsunaga¹
Lucas Busatta Galhardi²
Anderson Paulo Avila Santos³

RESUMO

O principal objetivo deste trabalho é o desenvolvimento de um sistema que integra as equipes comerciais e técnicas com o gerente da empresa, facilitando assim a análise de um serviço que está para iniciar ou que está em andamento. Na prática o sistema proposto tem total controle das equipes, serviços, produtos, agendamentos, metas, análises, relatórios de serviços e histórico. O desenvolvimento da solução passou por 2 etapas, onde na segunda foi-se melhorado e realizado um re-design no sistema todo que foi além do visual, como a logo, os padrões de cores, tipografia e ícones (identidade visual). Além disso, houve também mudanças direcionadas para a adaptação e a inclusão da acessibilidade, dando às pessoas com deficiência mais oportunidades dentro das empresas. O sistema até o momento está remodelado para pessoas que possuem deficiência visual – baixa visão e daltonismo (ícones/cores e fontes ampliadas) e auditiva (libras). Agora o sistema foi integrado com um aplicativo mobile para agilizar ainda mais os serviços, para que o técnico na rua não perca tempo escrevendo relatórios ou aguardando recebimento de ordens para conclusão do serviço.

Palavras-chave: Gestão, SI, Segurança, Patrimonial, Acessibilidade, Libras, Mobile.

SYSTEM MANAGER: SECURITY

ABSTRACT

The main objective of this work is the development of a system that integrates the commercial and technical teams with the company manager, thus facilitating the

¹ Mestre em Ciência da Computação. E-mail: anderson.avila@sistemafiep.org.br

² Mestre em Ciência da Computação. E-mail: lucas.galhardi@sistemafiep.org.br

³ Mestre em Ciência da Computação. E-mail: fabio.matsunaga@sistemafiep.org.br

analysis of a service that is about to start or that is in progress. In practice, the proposed system has full control of teams, services, products, schedules, goals, analyses, service reports and history. Development of the solution went through 2 stages, in which the second stage was improved, and a re-design was carried out on the entire system that went beyond the visual, such as the logo, color patterns, typography and icons (visual identity). In addition, there have also been changes aimed at adapting and including accessibility, giving people with disabilities more opportunities within companies. The system so far is remodeled for people who have visual impairments – low vision and color blindness (icons/colors and enlarged fonts) and hearing impairment (libras). Now the system has been integrated with a mobile application to further streamline the services, so that the technician on the street does not waste time writing reports or waiting to receive orders to complete the service.

Keywords: Management, IS, Security, Property, Accessibility, Libras, Mobile.

1. INTRODUÇÃO

A crescente violência urbana vem aumentando a busca por sistemas de vigilância mais desenvolvidos, que consigam realizar um monitoramento de forma autônoma, garantindo um melhor controle e segurança.

Diversas empresas de segurança patrimonial contam com grandes equipes técnicas e comerciais, mas não é o suficiente para ter qualidade nos atendimentos e serviços prestados, a falta de estrutura básica de um software para facilitar o dia a dia e a comunicação entre os colaboradores e clientes, pode gerar problemas como atrasos, equívocos e erros gerando assim transtornos. Controle é fundamental para melhorar a agilidade e eficiência no atendimento ao cliente.

Independentemente do tamanho da empresa e da quantidade de sistemas envolvidos na rotina do negócio, controlar a abertura de chamados e finalização de serviços é sinal de organização e eficácia. Pensando nisso, desenvolvemos o gestor de sistema Security, para ajudar as grandes, médias e pequenas empresas de segurança a melhorarem os seus serviços.

O sistema facilita a comunicação entre usuários e responsáveis técnicos, agiliza os processos de solicitação de atendimento e centraliza as informações em um único local para conferência da empresa e de clientes. Assim, nenhum detalhe é perdido, sem conversas cruzadas e falta de interpretação das duas partes.

O sistema registra a data e horário da abertura de chamados e orçamentos, o responsável e demais informações para contato. Além de arquivar orçamentos já realizados e serviços que já foram concluídos para possíveis consultas futuras.

O sistema foi baseado sobre estudos e análises de empresas do ramo de segurança patrimonial da cidade de Londrina, com uma estimativa de 20 empresas na área.

2. FUNDAMENTAÇÃO TEÓRICA

Inovação é um conceito amplo, mas na essência é a exploração com sucesso de novas ideias, conceitos ou práticas, quando disruptiva, transforma um mercado ou até um setor inteiro e geralmente faz isso através da introdução de simplicidade, conveniência e acessibilidade em grande escala.

No mundo dos negócios essas inovações, que podem ou não estarem ligadas a tecnologia, precisam gerar valor ao negócio, melhorando a performance, a qualidade ou a satisfação do cliente. Nesse ponto a organização é fundamental para que todas novas ações melhorem a experiência do cliente com os serviços prestados.

Segurança Patrimonial: Soluções Completas, Customizadas e Eficientes. Para cada cliente. Além de controles dos mais diversos e uma gestão eficiente, todas essas características garantem soluções completas em segurança e outros serviços que fortalecem a parceria de clientes e empresas. O resultado é a melhoria constante das operações que agregam valor ao negócio.

O poder transformador que os avanços tecnológicos vêm provocando em nossas áreas de atuação, principalmente no segmento de segurança nos diz onde deve estar o foco tanto de clientes como para empresas. Contudo, a tecnologia também possui formas criativas e inovadoras de realizar aplicações da tecnologia, e tem proporcionado melhorias em diversas soluções que vão além de mão de obra, com performance superior e redução de custos.

3. METODOLOGIA

3.1 Linguagens Utilizadas

O desenvolvimento web foi feito em Django, um framework para desenvolvimento rápido para web, escrito em Python, onde ele utiliza o padrão model-template-view. Foi-se utilizado o banco de dados padrão do Django: o SQLite.

O concorrente mais direto do Python é o Perl, vamos enfatizar os motivos por ter escolhido o Python como a Linguagem padrão do projeto. Python e Perl vêm de uma base semelhante e possuem também características parecidas, mas no fim têm uma filosofia diferente. Perl enfatiza o suporte para tarefas comuns orientadas a aplicativos. Agora o Python enfatiza o suporte para metodologias de programação comuns, como

desenho de estrutura de dados e programação orientada a objetos, e encoraja os programadores a escrever um código legível e sustentável. Uma semelhança delas que as fazem concorrentes tão diretas são porque suportam a programação procedural e orientada a objetos, e são linguagens multiparadigmáticas, interpretadas e de alto nível. Ambas podem se integrar com banco de dados que utilizam a linguagem SQL como interface. Podem ser usadas para desenvolver interfaces gráficas de usuário e distinguir entre letras maiúsculas e minúsculas em sua programação.

Para estilização utilizamos o CSS, que é uma linguagem para adicionar estilo ao documento web, também utilizamos o Bootstrap, que é um framework web utilizado para desenvolver componentes de interface e front-end para sistemas e aplicações web utilizando o HTML.

Normalmente o Javascript é utilizado junto com linguagens de marcação e regras, como o HTML e o CSS. Assim, é possível ter muito mais do que textos e tabelas coloridas, mas sim um conteúdo que realmente vai prender a atenção dos usuários da web.

3.2 Redesign

O redesign feito no sistema foi além do visual, como a logo, os padrões de cores, tipografia e ícones. Ele foi totalmente baseado nas necessidades reais do consumidor. Por tanto o foco e o objetivo desse semestre foram direcionados para adaptação e inclusão de acessibilidade, dando as pessoas com deficiência mais oportunidades dentro das empresas. Em cima do redesign também foi realizado a solução da avaliação heurística e teste de usabilidade com base nos resultados obtidos.

O sistema até o momento está remodelado para pessoas que possuem deficiência visual como baixa visão e daltonismo, para elas foram alterados os botões para ícones e adicionados pop-up com a descrição da ação. As fontes foram ampliadas para melhor visualização. E para a auditiva foi implementado libras.

O recurso utilizado para a tradução em libras no sistema é o VLibras que é o resultado de uma parceria entre o Ministério do Planejamento, Desenvolvimento e Gestão (MP), por meio da Secretaria de Tecnologia da Informação (STI) e a Universidade Federal da Paraíba (UFPB), a Suíte VLibras consiste em um conjunto de ferramentas computacionais de código aberto, responsável por traduzir conteúdos digitais (texto, áudio e vídeo) para a Língua Brasileira de Sinais - LIBRAS, tornando computadores, dispositivos móveis e plataformas Web acessíveis para pessoas com deficiência auditiva.

3.3 Avaliação Heurística

A capacidade heurística é uma característica humana para descobrir ou resolver problemas a partir da experiência prática, da observação e da criatividade. Por isso, para realizar uma boa avaliação heurística, é necessário um pequeno grupo de avaliadores com experiências sólidas em usabilidade, para examinar a interface e avaliar a sua conformidade com a lista de princípios de usabilidade escolhida previamente. Existem diversas listas de heurísticas definidas por diferentes especialistas, e em geral elas descrevem regras de propriedades comuns em interfaces com boa usabilidade.

Heurísticas: diretrizes de usabilidade que descrevem características desejáveis da interação e da interface. Avaliadores inspecionam sistematicamente a interface em busca de problemas que prejudiquem a usabilidade. Alternativa rápida e de baixo custo comparada a métodos com usuários. Método criado para encontrar problemas de usabilidade durante um processo de design iterativo.

O projeto contou com o planejamento e a realização da Avaliação Heurística e o Teste de Usabilidade.

4. Engenharia de Software do Sistema

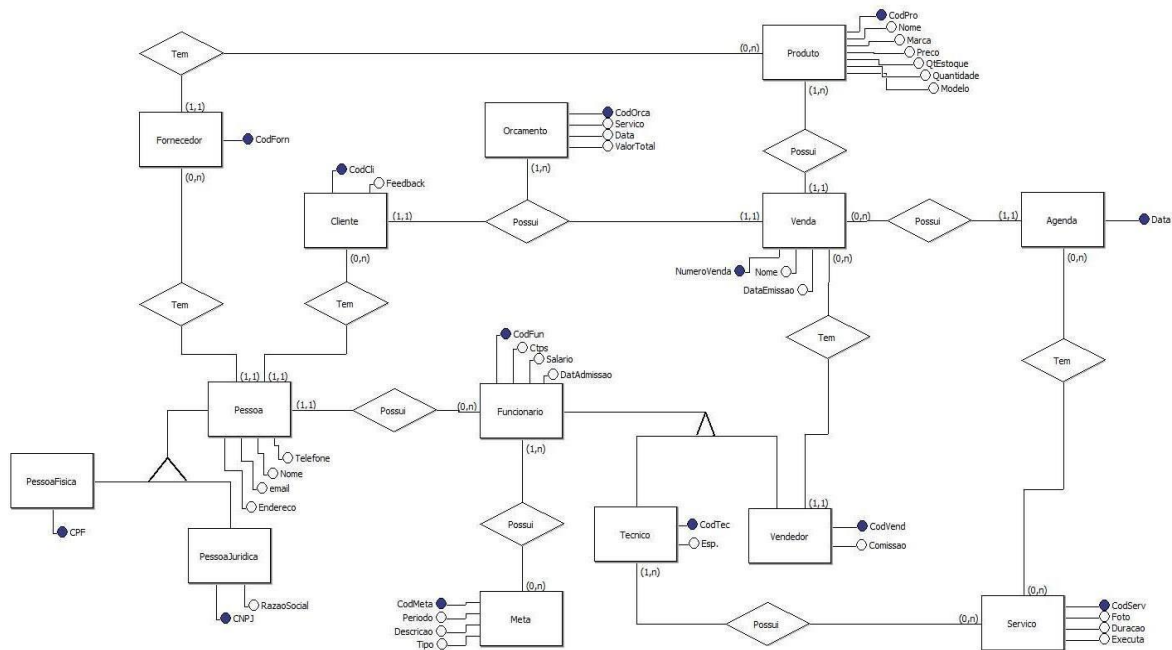
4.1 Requisitos

Após a aplicação da metodologia proposta, foram desenvolvidos os primeiros artefatos do projeto, os quais foram os requisitos funcionais e não-funcionais, agora fizemos um novo levantamento com requisitos funcionais adicionais e não funcionais (apresentados como duas tabelas), além das dependências entre eles (apresentado como um diagrama).

4.2 Diagramas UML

Para demonstrar as classes dentro do banco de dados utilizados no sistema, foram desenvolvidos o diagrama de classes e o diagrama conceitual, onde este segundo pode ser observado no Diagrama 1.

Diagrama 1: Conceitual



Fonte: o autor.

5. Desenvolvimento com Django

O Django possui uma interface de administração embutida onde é possível as informações armazenadas pelo modelo, para sua utilização é necessário adicionar as pastas para sua administração – “security/app1/admin.py”.

O modelo é definido pelo comportamento e campos chamada classes, ele é necessário para armazenagem no banco de dados onde são mapeadas como uma tabela – “security/app1/models.py”.

O Settings é um arquivo que contém toda a configuração do projeto, timezones, registro de apps, configurações da base de dados e etc. –

“security/security/settings.py”

Serializers permitem que dados complexos, como querysets e model instances, sejam convertidos em tipos de dados Python nativo que são renderizados em JSON,

XML etc.– “security/security/serializers.py”

Uma função View, é uma função Python que recebe uma solicitação da Web e retorna uma resposta da Web. Essa resposta pode ser o conteúdo HTML de uma página da Web, um redirecionamento, um erro 404, um documento XML ou uma imagem. – “security/app1/views.py”

As URLs consistem em um mapeamento que cuida de direcionar cada requisição recebida pelo servidor HTTP para uma determinada visão dentro das aplicações do projeto. Elas podem ser configuradas dentro do arquivo “./app1/urls.py”

Os arquivos que não precisam ser interpretados pelo Django, ou seja, imagens, arquivos de estilo CSS, programas em JavaScript etc., ficam por padrão no diretório “static” - “./app1/static//static/app2”.

O Django tem um componente próprio para a criação de formulários que se encarrega da tarefa de montá-los, tratá-los e validá-los – “./app1/forms.py”.

Sendo um framework para aplicações web o Django necessita de um jeito fácil para criar dinamicamente arquivos HTML. E a forma mais prática de fazê-lo é através do uso de templates contendo código HTML.

Foi utilizado bibliotecas adicionais como:

“libgravar”: é específica para acessar e recuperar informações da API do Gravatar.

“django-widget-tweaks”: é uma extensão que permite alteração de atributos dos campos HTML, que são gerados automaticamente no HTML.

Foi utilizado framework adicionais como:

“djangorestframework”: O Django Rest Framework é uma toolkit poderosa e flexível para criar Web APIs ou Rest APIs. Com ele conseguimos implementar Rest APIs de forma muito rápida.

6. Apresentação e discussão dos resultados

6.1 TELA DE LOGIN

Ao iniciar o sistema é necessário realizar o login para que ocorra a autenticação sobre as funções que o usuário pode realizar, é necessário usuário e senha.

Figura 02: Tela de Login



Fonte: o autor.

Tabela 06: Usuários.

TIPO DE USUÁRIO	PERMIÇÃO	SEM ACESSO
Gerente	Total	---
Comercial	Parcial	Cadastrar, editar e remover os produtos, funcionários e fornecedores
Técnico	Parcial	Cadastrar, editar e remover os produtos, funcionários e fornecedores

Fonte: o autor.

6.2 Tela Cliente

Figura 03: Tela Cliente.

Na tela do cliente, podemos realizar o cadastro de novos clientes e verificar os clientes já cadastrados conforme a Figura 03. Foi retirada está tela no atual Security.

Fonte: o autor.

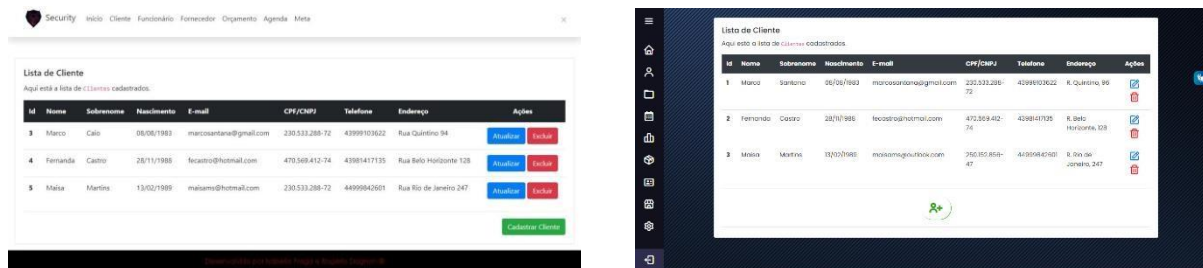
Ao realizar um novo cadastro de cliente temos os campos obrigatórios como na Figura 04: nome, sobrenome, nascimento, e-mail, CPF/CNPJ, telefone e endereço. Caso esteja algum campo incorreto ou faltando ocorre uma validação, se tiver algo errado irá aparecer uma notificação para que se possa ser corrigido antes de finalizar o cadastro.

Figura 04: Cadastro de cliente.

Fonte: o autor.

Na lista de clientes, podemos conferir todos os clientes e seus dados, podendo atualizar ou excluir qualquer cliente, conforme a Figura 05.

Figura 05: Lista de clientes.



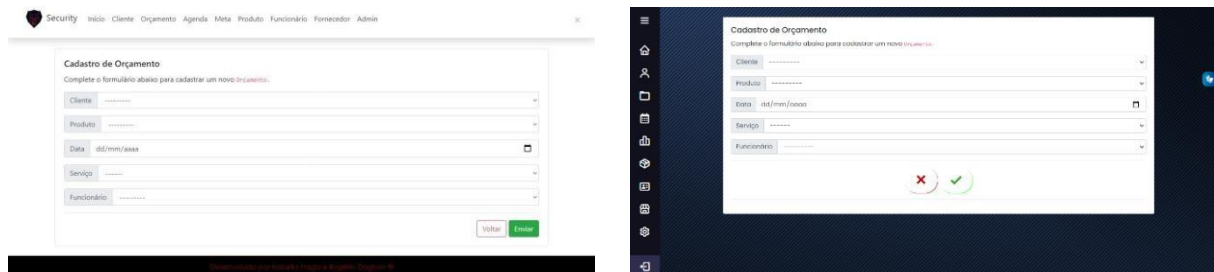
Fonte: o autor.

6.3 Tela Orçamento

Na tela do orçamento, podemos realizar o cadastro de novos orçamentos e verificar orçamentos cadastrados.

Ao realizar um novo cadastro de orçamento temos os campos obrigatórios para cadastro como, cliente, produto, data, serviço e funcionário. Caso esteja algum campo incorreto ou faltando ocorre uma validação, se tiver algo errado irá aparecer uma notificação para que possa ser corrigido antes de finalizar o cadastro. Conforme a Figura 07.

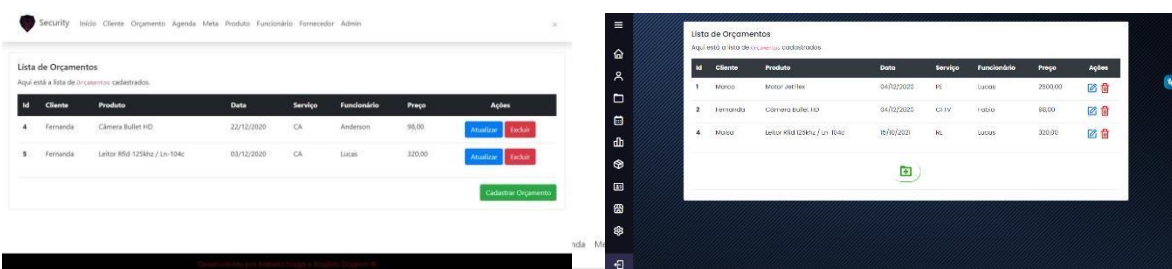
Figura 07: Cadastro de Orçamento.



Fonte: o autor.

Na lista de orçamento, também podemos conferir todos os orçamentos cadastrados e seus dados, podendo atualizar e excluir.

Figura 08: Lista de Orçamento.



Cadastrar Cliente

Cadastre aqui um novo cliente.

[Novo Cliente](#)

Lista de Cliente

Veja aqui a lista de clientes cadastrados.

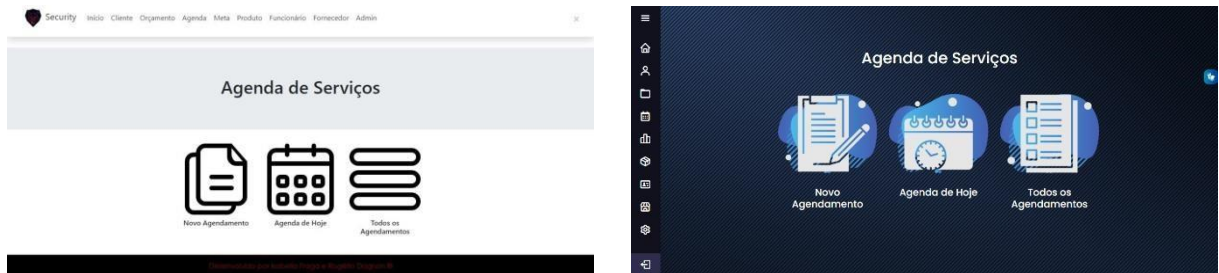
[Vá para Lista](#)

Fonte: o autor.

6.4 Tela Agenda

Na tela da agenda, podemos realizar o cadastro de novos agendamentos, verificar agendamento do dia e verificar todos os agendamentos, conforme a Figura 09.

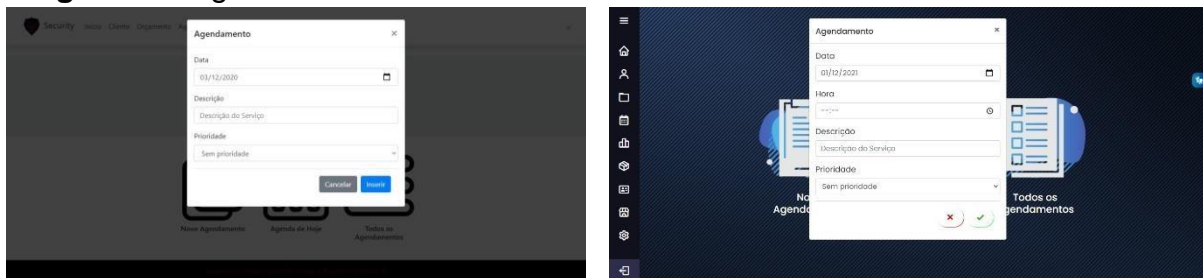
Figura 09: Tela Agenda.



Fonte: o autor.

Para realizar um novo agendamento, será necessário inserir os dados, data, descrição e prioridade, conforme a Figura 10.

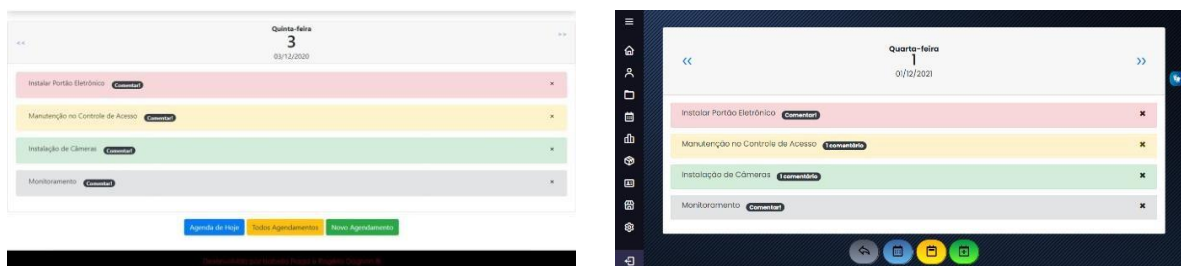
Figura 10: Agendamento.



Fonte: o autor.

Na agenda de hoje pode ser verificado os agendamentos para o dia específico, conforme a Figura 11.

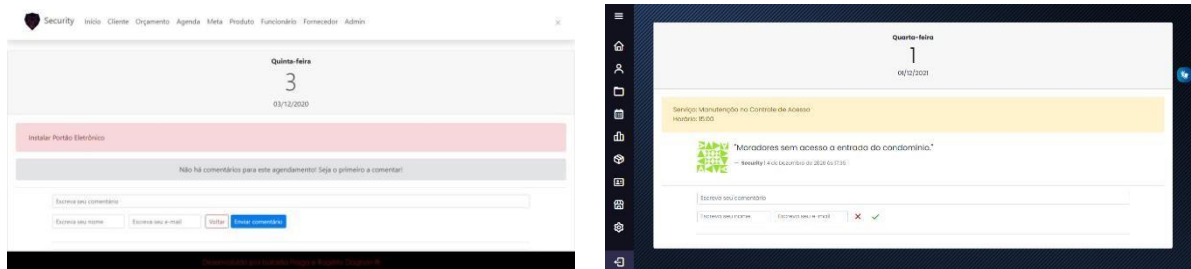
Figura 11: Agenda de Hoje.



Fonte: o autor.

Na área de comentário, podemos fixar um comentário em cada item agendado caso necessário, conforme a Figura 12.

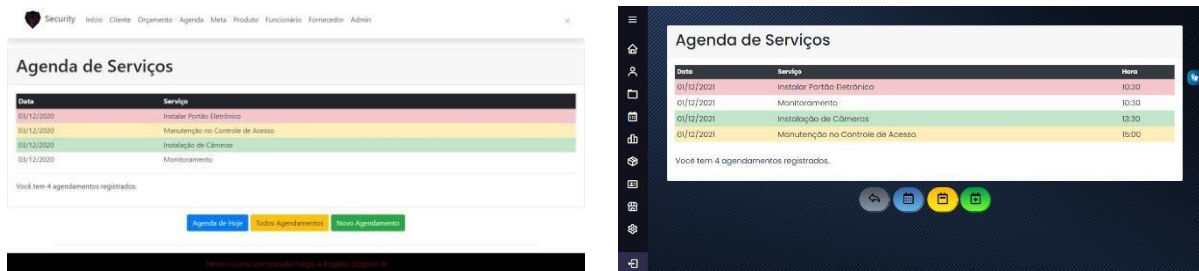
Figura 12: Comentário no Agendamento.



Fonte: o autor.

Na área de todos os agendamentos, podemos verificar todos os agendamentos realizados e quantos agendamentos temos registrado, conforme a Figura 13.

Figura 13: Todos os Agendamentos.

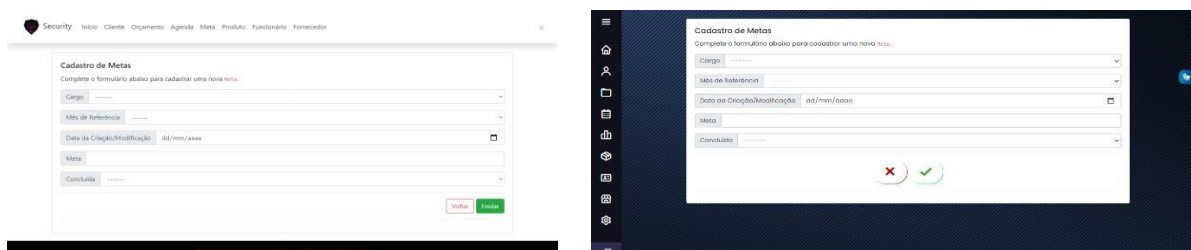


Fonte: o autor.

6.5 Tela Meta

Na tela das metas, podemos realizar o cadastro de novas metas, verificar metas cadastradas. Ao realizar um novo cadastro de meta temos os campos obrigatórios para cadastro como, cargo, mês de referência, data da criação/modificação, meta e concluída. Caso esteja algum campo incorreto ou faltando ocorre uma notificação para que possa ser corrigido antes de finalizar o cadastro.

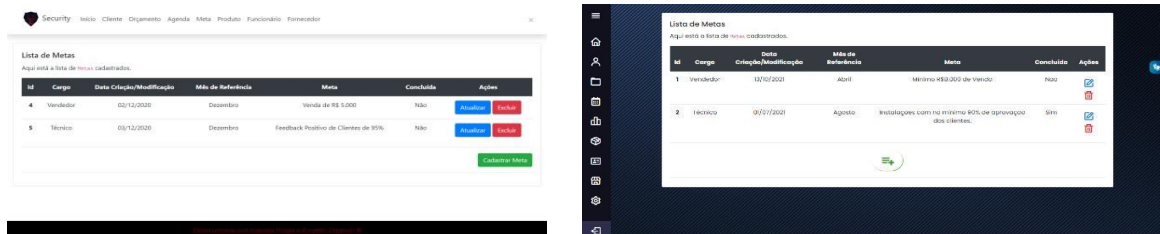
Figura 14: Cadastro Meta.



Fonte: o autor.

Na lista de metas, podemos conferir todas as metas e seus dados, podendo atualizar e excluir.

Figura 15: Lista Meta.



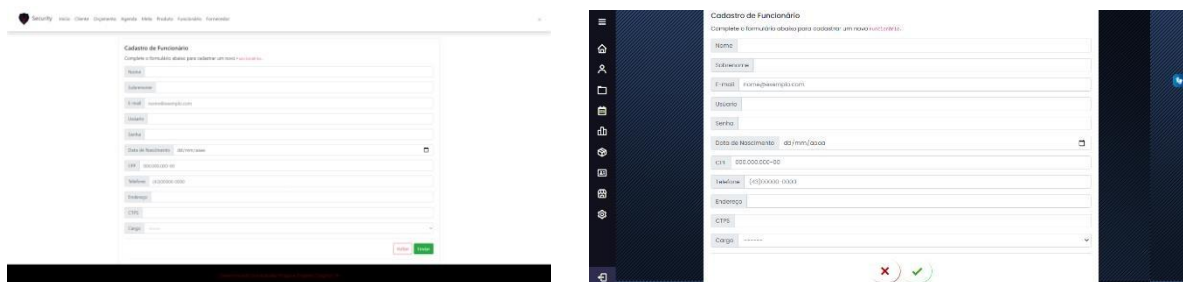
Fonte: o autor.

6.6 Tela Funcionário

Na tela do funcionário, podemos realizar o cadastro de novos clientes e verificar os clientes já cadastrados.

Ao realizar um novo cadastro de cliente temos os campos obrigatórios para cadastro, conforme a Figura 16, nome, sobrenome, e-mail, usuário, senha, data de nascimento, CPF, telefone, endereço, CTPS e cargo. Caso esteja algum campo incorreto ou faltando ocorre uma notificação para que possa ser corrigido antes de finalizar o cadastro.

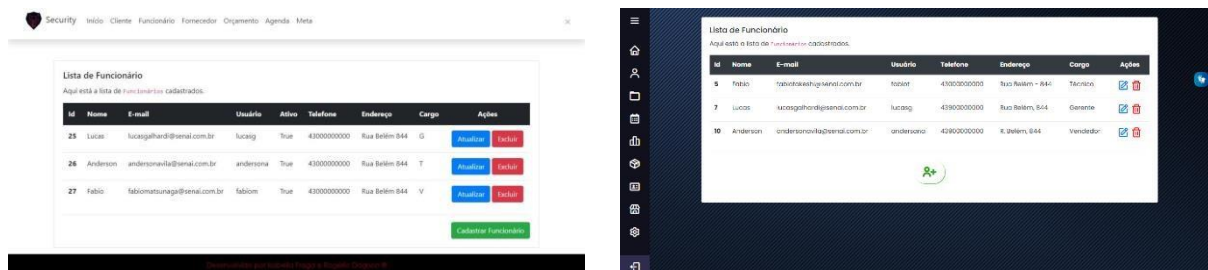
Figura 16: Cadastro Funcionário.



Fonte: o autor.

Na lista de funcionários, podemos conferir todos os funcionários e seus dados, podendo atualizar, excluir e bloquear, conforme a Figura 17.

Figura 17: Lista de Funcionário.

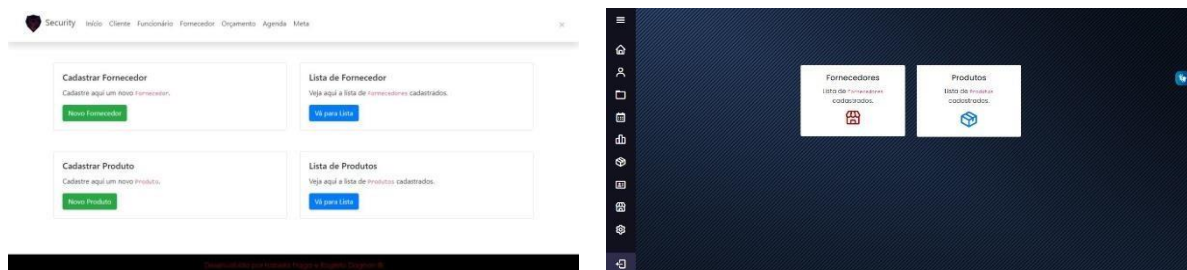


Fonte: o autor.

6.7 Tela Fornecedor

Na tela do fornecedor, podemos realizar o cadastro de novos fornecedores, verificar fornecedores cadastrados, cadastrar produtos e verificar produtos cadastrados, conforme a Figura 17.

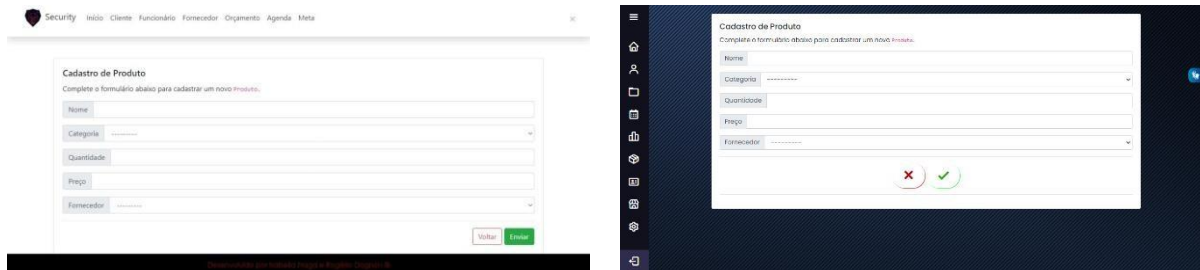
Figura 17: Tela de Fornecedor.



Fonte: o autor.

Ao realizar um novo cadastro de fornecedor, conforme a Figura 18, temos os campos obrigatórios para cadastro como, nome, e-mail, telefone, endereço, CNPJ e categoria. Para um novo cadastro de produto, também temos os campos obrigatórios de cadastro como, nome, categoria, quantidade, preço e fornecedor. Caso esteja algum campo incorreto ou faltando ocorre uma notificação para que possa ser corrigido antes de finalizar o cadastro tanto para os fornecedores quanto para os produtos.

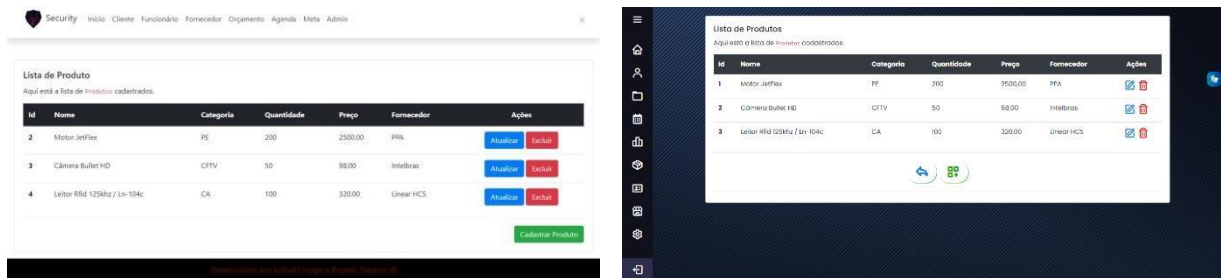
Figura 18: Cadastro de Produto.



Fonte: o autor.

Na lista de produtos, também podemos conferir todos os fornecedores e seus dados, podendo atualizar e excluir.

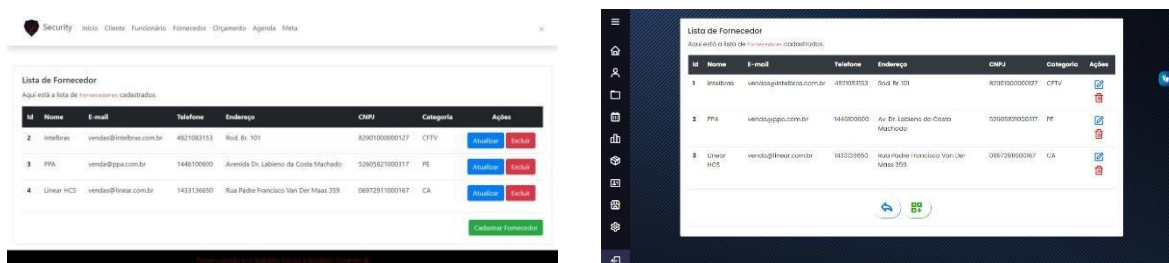
Figura 19: Lista de Produto.



Fonte: o autor.

Na lista de fornecedores, podemos conferir todos os fornecedores e seus dados, podendo atualizar e excluir.

Figura 20: Lista de Fornecedor.



Fonte: o autor.

7. APLICATIVO MOBILE

O aplicativo mobile desenvolvido para este projeto tem o objetivo de estender o sistema web auxiliando na interação do usuário. Ele foi desenvolvido em React Native que é uma biblioteca Javascript para construir interfaces de usuários, que fornece um conjunto de plataformas e componentes nativos agnósticos como View, Text e Image além de interagir com APIs nativas por meio do paradigma de IU declarativo do React e do Javascript.

Para realizar a conexão do aplicativo mobile com o sistema web, utilizamos a API Fetch do próprio React Native para atender todas as necessidades de rede, com ela conseguimos acessar e manipular partes do HTTP, buscando os arquivos em JSON para apresentar os dados direto do banco no aplicativo.

Com a API Fetch também trabalhamos as solicitações e requisições POST, GET, DELETE e PATCH, para atuarem na parte das anotações, armazenando assim no banco de dados do sistema onde não teremos problemas de percas ou falhas como ocasionado se fosse salvo localmente no próprio aplicativo.

8. CONSIDERAÇÕES FINAIS

Através desse trabalho conseguimos dar continuidade ao sistema web, adicionando mais requisitos e interações, passando assim a conhecer um pouco mais sobre a interação do aplicativo mobile com o software e a facilidade gerada ao usuário em sua utilização. Tornando possível o gerenciamento de diversos setores para as empresas de segurança.

REFERÊNCIAS

SAUDATE, Alexandre. **A. Rest – Construa API’S inteligentes de maneira simples**. 1. Ed. São Paulo: Casa do código, 2013. 314 p.

FACEBOOK OPEN SOURCE. Disponível em: <<https://reactnative.dev>>. Acessado em: 20 ago. 2021.

MOZILLA AND INDIVIDUAL CONTRIBUTORS. Disponível em:

<<https://developer.mozilla.org/en-US/>> Acessado em: 22 ago. 2021.

STREPHONSAYS - Diferença entre Perl e Python, Disponível em:

<<https://pt.strephonsays.com/perl-and-vs-python-9014>> Acessado em: 11 nov 2021.

CETAX - Python: Comparando Python com outras linguagens, Disponível em:

<<https://www.cetax.com.br/blog/comparando-python-com-outras-linguagens/>>
Acessado em: 11 nov 2021.

STREPHONSAYS - Diferença entre JavaScript e TypeScript, Disponível em:
<<https://pt.strephonsays.com/javascript-and-vs-typescript-13697>> Acessado em: 11 nov 2021.

Redação Impacta - Typescript x JavaScript: entenda o que é e saiba as diferenças, Disponível em: <<https://www.impacta.com.br/blog/typescript-javascript-entenda-quesaiba-diferencas>> Acessado em: 11 nov 2021.

Ipea – Objetivos do Desenvolvimento Sustentável, Disponível em:

<<https://www.ipea.gov.br/ods/index.html>>, Acessado em: 15 nov. 2021.

Estratégia ODS – O QUE SÃO OS ODS?, Disponível em:

<<https://www.estrategiaods.org.br/conheca-os-ods/>>, Acessado em: 15 nov. 2021.

Polen Portal Crédito de Logística Reversa – Objetivos de Desenvolvimento

Sustentável (ODS) da ONU: tudo o que você precisa saber, Disponível em:

<<https://www.creditodelogisticareversa.com.br/post/t-objetivos-de-desenvolvimentosustentavel-ods-da-onu-tudo-o-que-voce-precisa-saber>>, Acessado em: 15 nov. 2021.

MERGO - Avaliação Heurística na análise de interfaces, Disponível em:
<<https://uxdesign.blog.br/avalia%C3%A7%C3%A3o-heur%C3%ADstica-naan%C3%A1lise-de-interfaces-218c2dd46164>>, Acessado em: 2 dez. 2021.

WayCarbon. Conheça os 17 objetivos do desenvolvimento sustentável da ONU, Disponível em: <<https://blog.waycarbon.com/2015/10/conheca-os-17-objetivos-dodesenvolvimento-sustentavel-da-onu/>>, Acessado em: 15 nov. 2021.

EQUIPE CORE. Bootstrap, Disponível em: <<https://getbootstrap.com.br/>>. Acessado em: 30 set. 2020.

DJANGO SOFTWARE FOUNDATION. DjangoProject, Disponível em:

<<https://www.djangoproject.com/>>. Acessado em: 10 nov. 2020

W3.CSS. W3schools, Disponível em: <<https://www.w3schools.com/>>. Acessado em: 20 out. 2020.

PYTHON ACADEMY. Desenvolvimento Web com Python e Django, Disponível em: <<https://pythonacademy.com.br/blog/desenvolvimento-web-com-python-e-djangotemplate/>>. Acessado em: 30 nov. 2020.

GIOVANNI NUNES. Exemplo em Django Autenticação, Disponível em:

<<https://giovannireisnunes.wordpress.com/2019/09/27/exemplo-em-django-autenticacao/>>. Acessado em: 28 nov. 2020.

ZKoder. Django Rest Framework - MySQL CRUD RESTful APIs, Disponível em <<https://bezkoder.com/django-crud-mysql-rest-framework/>>. Acessado em: 20 nov. 2020.