

**LOCMAC**

LOCAÇÃO DE MÁQUINAS E EQUIPAMENTOS

**PREGÃO PRESENCIAL – SRP Nº 007/2022 – AADESAM  
PROPOSTA COMERCIAL Nº 001/2022/LIC/ILOMAC****Nome de Fantasia: LOCMAC COMERCIO E SERVICOS DA AMAZONIA****Razão Social: M.A.N TEIXEIRA E CIA LTDA****CNPJ: 20.765.220-13****Endereço: AV RIO JUTAI 670****Bairro: NSA DAS GRACAS****CEP: 69053-020****Telefone: (92)98252-7140****Banco: BRADESCO****Nome e nº da agência bancária: AMAZ SHOPPING 1374****Optante pelo SIMPLES? NÃO****Cidade: MANAUS AM****E-mail: COMERCIAL.LOCMAC@OUTLOOK.COM****Fax:****Conta Bancária: 23324-2**

ITEM	DESCRIÇÃO	UF	QTD	V. UNIT	V. TOTAL
1	(ID-5858) Locação de monitor LCD, contemplando serviços de manutenção preventiva e corretiva sob demanda, com fornecimento de peças.	SV	14400	R\$ 63,00	R\$ 907.200,00
<b>Marca / modelo: MONITOR POSITIVO 22BN550Y</b>					
<b>Valor unitário: sessenta e três reais</b>					
<b>Valor total: novecentos e sete mil e duzentos reais</b>					
2	(ID-5868) Locação de microcomputador tipo ALL IN ONE com processador de 7ª geração e tela embutida de 21,5"	SV	14400	R\$ 405,50	R\$ 5.839.200,00

**LUCMAG**

LOCAÇÃO DE MÁQUINAS E EQUIPAMENTOS

<b>Marca / modelo: POSITIVO ALLINONE PRESARIO CQ-A1</b>			
<b>Valor unitário:</b> quatrocentos e cinco reais e cinquenta centavos			
<b>Valor total:</b> cinco milhões, oitocentos e trinta e nove mil e duzentos reais			
<b>3</b>	(ID-5845) Locação de notebooks, contemplando os serviços de manutenção preventiva e corretiva sob demanda, com fornecimento de peças: incluindo componentes originais e periféricos (mouse)	SV	3600
			R\$ 451,90
			R\$ 1.626.840,00
<b>Marca / modelo: POSITIVO VAIO FE14</b>			
<b>Valor unitário:</b> quatrocentos e cinquenta e um reais e noventa centavos			
<b>Valor total:</b> um milhão, seiscentos e vinte e seis mil, oitocentos e quarenta reais			
<b>4</b>	(ID-5869) Locação de nobreak de 600VA bivolt com 6 tomadas	SV	14400
			R\$ 49,90
			R\$ 718.560,00
<b>Marca / modelo: SMS LEGRAND STATION II 600VA4</b>			
<b>Valor unitário:</b> quarenta e nove reais e noventa centavos			
<b>Valor total:</b> setecentos e dezoito mil quinhentos e sessenta reais			
<b>5</b>	(ID-5846) Locação de impressoras tipo MPF Laser Monocromática, contemplando os serviços de manutenção	SV	2400
			R\$ 415,00
			R\$ 996.000,00



	preventiva e corretiva sob demanda, com fornecimento de peças.				
<b>Marca / modelo: BROTHER DCP-8157DN</b>					
<b>Valor unitário: quatrocentos e quinze reais</b>					
<b>Valor total: novecentos e noventa e seis mil reais</b>					
6	(ID-5847) Locação de impressoras tipo MPF Laser Colorida, contemplando os serviços de manutenção preventiva e corretiva sob demanda, com fornecimento de peças.	SV	1200	R\$ 439,00	R\$ 526.800,00
<b>Marca / modelo: BROTHER DCP-L3551CDW</b>					
<b>Valor unitário: quatrocentos e trinta e nove reais</b>					
<b>Valor total: quinhentos e vinte e seis mil e oitocentos reais</b>					
7	(ID-5870) Serviço de impressão de folha de papel formato A4 em impressora laser monocromática, contemplando os serviços de manutenção preventiva e corretiva sob demanda, com fornecimento de peças.	SV	48000000	R\$ 0,15	R\$ 7.200.000,00
<b>Marca / modelo: BROTHER DCP-8157dn</b>					
<b>Valor unitário: quinze centavos</b>					
<b>Valor total: sete milhões e duzentos mil reais</b>					

*Handwritten signature and initials*

Av. Rio Jataí, 670 - Neves Binheira das Graças, Manaus, AM, CEP 69.053-020  
 CNPJ: 20.755.200/0001-13 | e-mail: comercial@locmac.com



**LOCMAC**

LICITAÇÃO DE MÁQUINAS E EQUIPAMENTOS

<b>8</b>	(ID-102176) Serviço de impressão de folha de papel formato A4 em impressora laser colorida, com o fornecimento dos suprimentos necessários, como toner e papel e operacionalização da solução.	SV	24000000	R\$ 0,93	R\$ 22.320.000,00
<b>Marca / modelo: BROTHER DCP-L3551CDW</b>					
<b>Valor unitário:</b> noventa e três centavos					
<b>Valor total:</b> vinte e dois milhões, trezentos e vinte mil reais					

**Valor global da proposta: R\$ 40.134.600,00 (quarenta milhões, cento e trinta e quatro mil e seiscentos reais).**

Declaramos que:

- Nos valores das propostas de preços estão incluídas todas as despesas com tributos e fornecimento de certidões e documentos, bem como encargos fiscais, sociais, trabalhistas, previdenciários, comerciais e outros de qualquer natureza e, ainda, gastos com transportes e acondicionamento em embalagens adequadas, conforme caso;
- Atendemos a todas as especificações, exigências técnicas mínimas, prazos de entrega ou de prestação, cronograma de execução e as respectivas quantidades, conforme caso;
- Examinamos cuidadosamente o Edital, inteiramo-nos de todas as cláusulas e seus detalhes e com eles concordamos, aceitamos todos os seus termos e condições e a eles desde já nos submetemos. Caso sejamos vencedora no certame, submeteremo-nos a todas as condições estabelecidas neste Edital e na minuta do contrato que o integra, sob pena de rescisão unilateral do contrato;
- Nos termos do item 10.2 do Edital, o treinamento será realizado em até 5 (cinco) dias após a entrega dos equipamentos;
- Nos termos do item 17.2.16 do Edital, manteremos um SLA igual ou superior a 95% (noventa e cinco por cento);
- Nos termos do item 17.2.18 do Edital, manteremos um centro de assistência nas dependências da AADESAM, em até 30 (trinta) dias após a assinatura do contrato;
- Nos termos do item 5.2.10.1 do Termo de Referência, entregaremos em até 10 dias após a assinatura do contrato um modelo de cada equipamento para que sejam efetivadas as configurações padrão da AADESAM.

Handwritten signature and date: 4/2/23

# MONITOR POSITIVO 22BN550Y

Qualidade Superior de Imagem com Economia de Energia



- Imagens excepcionais com a tecnologia LED IPS e resolução Full HD
- Super economia de energia
- Tela com tratamento antirreflexivo
- Ajustes de altura, inclinação e rotação



## TELA IPS LED de 21,5" / Resolução Full HD / Alto Contraste / Plug & Play / DP + HDMI + VGA

Tamanho da Tela	21,5" Widescreen, Formato 16:9
Tipo de Tela	LCD TFT IPS-LED
Investimento da Tela	Anti-Glare, Hard Coating 3H
Suporte de Cores	16,7 milhões de cores
Pixel Pitch	0,248 mm
Bribe	250 cd/m <sup>2</sup>
Contraste	5.000.000:(DIN) / 1000:1(Estática)
Angulo de Visão	178° / 178° (D/V)
Tempo de Resposta	5 ms (GTG)
Resolução Máxima	1920x1080 @ 60Hz
Frequência de Varredura	Analgica: 30 - 83 kHz (H) / 56 - 75 Hz (V)   Digital: 30 - 83 kHz (H) / 56 - 75 Hz (V)
Video	Analgico, Digital
Nito Falamas	Não
Conector	1x DisplayPort, 1x HDMI, 1x D-Sub 15 pines (VGA) e 1x Headphone Out
Recursos	Plug and Play (DDC2B/DDC2A); Abertura para chave tipo Kensington-EDID/VESA - N° Serie/Fabricante/Modelo; Ajuste de idioma através do menu
Fixação para suporte (VESA)	Sim (100 mm x 100 mm)
Tensão	AC 100-240V (+/- 10%)   50/60 Hz (automática), fonte interna
Consumo de Energia (Ligado/Standby)	15W (Típico) / x 0,3 W
Controles Manuais Digitais- Painel Frontal	Teclas: Menu/Reader/Energy Saving/Auto ajuste/Input/Exit/Liga-Design OSD (On Screen Display) Ajustes: brilho, contraste, brilho, posição horizontal/vertical, trava de ajuste; português/inglês LED de equipamento ligado/standby
Ajustes	Apoio da base removível, Inclinação (-5° - +20°), Ajuste de Altura (11 cm), Pivote (90°) e Giro Eixo Horizontal (opcional)
Dimensões (L x A x P)	Produto: 508 x 224 x 356 mm (com base)   Embalagem: 672 x 375 x 193 mm
Cor	Prata
Peso Líquido / Bruto	4,3 Kg / 5,7 Kg
Accessórios	Cabo de força 18m NBR 14136, Cabo D-Sub 18m, Cabo de Alimentação tipo Y (opcional), Alimentador NBR14136-2P+T (opcional), Cabo DisplayPort 1.8m, Cabo HDMI 1.8m, Base do Suporte e Manual do Usuário em português
Certificações	FCC CLASS B, EPEAT, INMETRO, TCO 8.0, ENEC, TÜV, CB, CE, RoHS
Compatibilidade	Windows e Linux



Windows 10  
Compaq recomenda o Windows

# ALL IN ONE PRESARIO CQ-A1

## 4GB/500GB/WINDOWS 10



intel OPTANE  
MEMORY



Microsoft product image: https://www.microsoft.com/pt-br/

*O All in One PRESARIO CQ-A1 possui um design impactante e muita tecnologia para transformar a forma como você pensa um computador. Oferece muita segurança sem complicar a forma de como você usa.*

### Visão geral do Produto

- Memória: 4 GB DDR4
- Disco Rígido: 500GB SATA
- 2 portas USB 3.0
- 3 portas USB 2.0

### Principais Especificações

- Sistema operacional: Windows 10
- Processador: Intel® Core i3™ + Memória Intel Optane 16GB
- Monitor: 23.8" FULL HD, 16:9 LCD – Resolução: 1920x1080
- Memória: 4GB DDR4L 5000MHz
- Armazenamento: 500GB SATA
- Placa de vídeo: Intel® UHD Graphics 620
- Peso do produto: Aproximadamente 3kg
- Teclado: ABNT2

### Recursos do Produto

#### Novidades do Windows 10

1. Uma plataforma unificada - O mesmo Windows 10 utilizado por computadores, será também de tablets, smartfones e outros aparelhos. Softwares produzidos para um tipo de aparelho poderão ser facilmente importado para outros.
2. A volta do Menu Iniciar - A volta de dois dos recursos aos os consumidores após sentiram falta: o Menu Iniciar como era visto até o Windows 7 e também a Área de trabalho como interface principal.
3. As janelas voltam ao topo - Os apps disponíveis na loja Windows Store não serão mais rodados conjuntamente em tela cheia, isso significa que teremos de volta o design "janelas".
4. Múltiplas áreas de trabalho - O Windows 10 traz o recurso de múltiplas áreas de trabalho e o "Snap View", que permite aos consumidores arrastarem aplicativos entre os desktops para que tudo seja organizado da melhor maneira possível para cada momento.
5. Botão "Task View" - A partir de agora a barra de tarefas conta com um novo botão que é responsável pela troca rápida entre arquivos e softwares abertos, permitindo também o acesso às múltiplas áreas de trabalho.

5. Explorador de arquivos - O explorador de arquivos integrado ao Windows garante que os consumidores possam ter mais facilidade na hora de localizar um app ou um documento. O sistema agora mostra arquivos recentes e pastas visitadas com mais frequência de um modo bem inteligente.

7. Abre-se entre Touchscreen e Mouse/Teclado - O Windows 10 conta com uma adaptação total dos aplicativos, que podem ser utilizados com mouse e teclado sempre que isso for a vantagem do usuário.

### Garantia e suporte

- A Garantia de seu produto Compaq inclui:
- Hardware: Garantia Limitada de 1 ano\* (garantia legal incluída)
- Software: Suporte Técnico por 30 dias\*

#### Serviços Adicionais Compaq

Suporte: A Compaq oferece suporte técnico para o Software Compaq, software de terceiros pré-instalado pela Compaq e software de terceiros adquirido da Compaq durante 30 dias após a aquisição.

[www.compaq.com.br](http://www.compaq.com.br)

Telefone: (11) 4020-0142



[compaq.com.br](http://compaq.com.br)

**COMPAQ**  
Compatível com você.

# All in One PRESARIO CQ-A1



Compaq recomenda o Windows

## Especificações

<b>Sistema Operacional</b>	• Windows 10
<b>Processador</b>	• Intel® Core i3™ + Memória Intel Optane 16GB
<b>Controlador de vídeo</b>	• Intel® UHD Graphics 620
<b>Memória RAM</b>	• 4GB DDR4
<b>Display</b>	• LCD 23.8" Full HD (1920 x 1080)
<b>Armazenamento</b>	• 500GB SATA
<b>Microfone</b>	• Embutido
<b>Portas I/O</b>	• 1x DC-In • 1x RJ-45 • 1x HDMI • 3x USB 2.0 • 2x USB 3.0 • 1x Entrada para fone de ouvido • 1x Entrada para microfone
<b>Wifi</b>	• 802.11 b/g/n com Bluetooth embutido
<b>Teclado</b>	• ABNT2
<b>Alimentação</b>	• Entrada: 19Vcc 3,42A - 65W
<b>Dimensões (A x L x C)</b>	• 54,5cm x 38,45cm x 12cm
<b>Peso</b>	• Aprox. 3kg
<b>Cod EAN</b>	
<b>NCM</b>	

## Software

<b>Apps</b>	O Windows 10 vem com vários aplicativos nativos, como Mapas, Fotos, Email e Calendário, Música, Filmes e TV.
-------------	--

## Windows 10

**Você já é um especialista** - O Windows 10 é tão familiar e fácil de usar que você vai se sentir como um verdadeiro especialista. O menu Iniciar está de volta de forma expansível, e também vamos incluir seus aplicativos fixados e favoritos para que eles estejam prontos e à sua espera.

Ele é instalado e retomado com rapidez, tem mais segurança interna para garantir a segurança e foi projetado para funcionar com o software e o hardware que você já possui. **Milhões de pessoas já estão usando o Windows 10** - O Windows Insider Program é uma comunidade global de fãs que adoram o Windows e querem ajudar a melhorá-lo. Os usuários do Windows Insider veem o sistema operacional em seus estágios iniciais e desempenham um papel em sua formação. Os seus comentários contribuem para a criação do melhor Windows de todos os tempos.

**A Web que funciona do seu jeito** - O Windows 10 vem com um navegador totalmente novo, o Microsoft Edge, criado para lhe oferecer uma experiência melhor na Web. Escreva ou digite anotações diretamente em páginas da Web e compartilhe-as com outras pessoas, veja artigos online sem distrações e grave suas leituras favoritas para acesso posterior. Com a Cortana habilitada, você tem acesso imediato às principais ações, como fazer reservas ou ter análises, sem sair da página em que está.

**Multitarefa o ajudará a terminar mais rápido** - Ajuste facilmente até quatro aplicativos e vá de tarefa a tarefa abertas em um único modo de exibição. Você pode até criar áreas de trabalho virtuais quando precisar de mais espaço ou quiser agrupar itens por projeto.

**A Windows Store é sua loja única** - Apresentando a nova Windows Store, uma experiência unificada de compras em todos os dispositivos Windows 10. Navegue pela loja em seu computador, tablet ou celular e compre o melhor conteúdo digital com facilidade, incluindo aplicativos, jogos, músicas, filmes e programas de TV.

**É o Windows mais pessoal de todos os tempos** - Personalize seu menu Iniciar fixando qualquer item (inclusive aplicativos, pessoas e listas de reprodução) para ter fácil acesso ao que você mais usa. Com os blocos de demonstração, você pode ir direto ao conteúdo, sem ter que abrir primeiro o aplicativo.

**Interaja com seu dispositivo da maneira que quiser** - Toque, digite, escreva a esmo ou use sua Assinatura para controlar o que funciona melhor para o que você está fazendo.

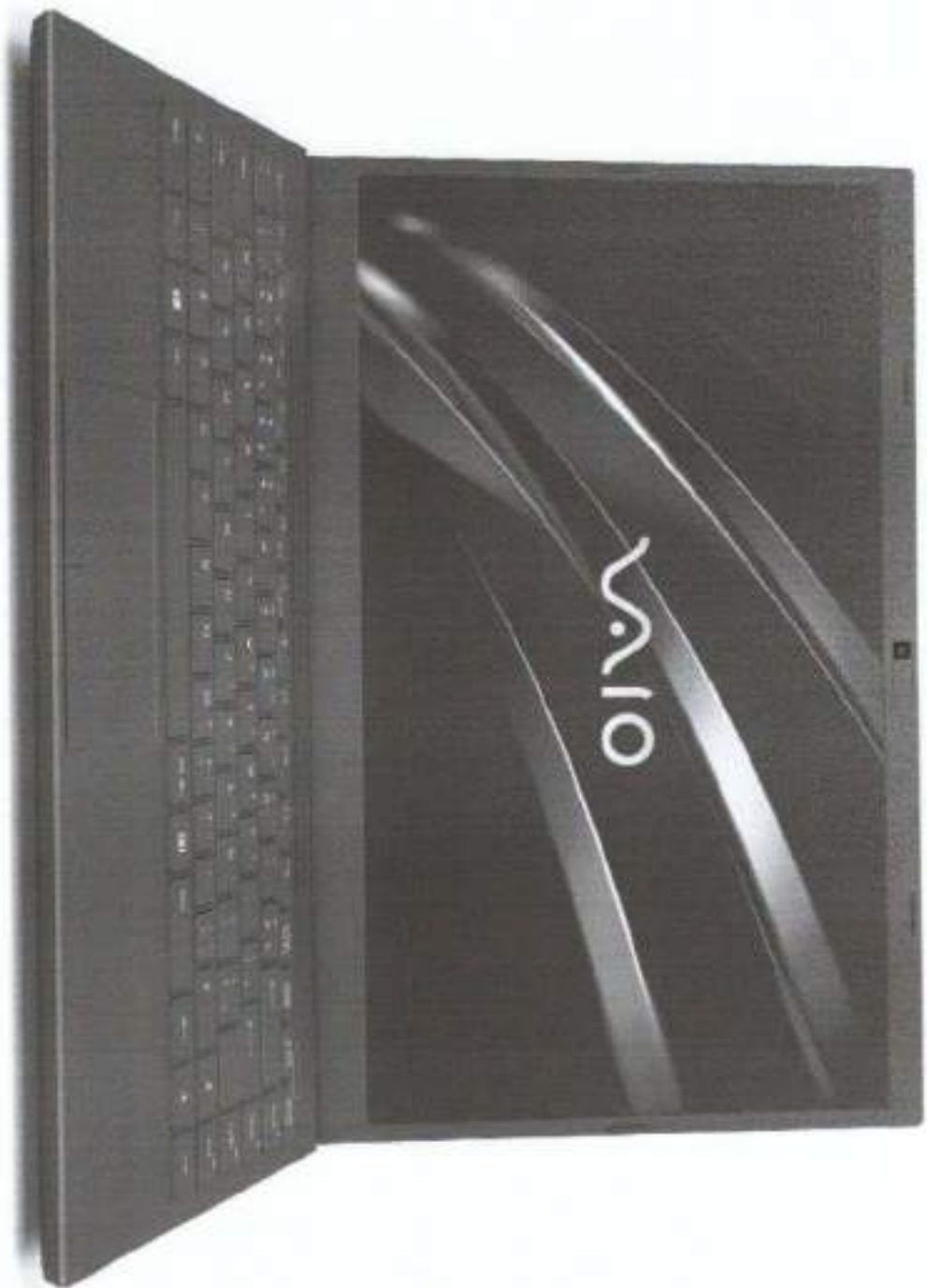
A melhor tela é sempre aquela que você está usando - O Windows 10 adapta seus aplicativos a terem ótima aparência e funcionamento em todos os modos e dispositivos. Em dispositivos conversíveis, a tela pode ser otimizada para funcionar com toque ou com teclado e mouse.

**O Xbox está vindo para o Windows** - As melhores franquias do Xbox e o melhor do Xbox Live estão no Windows 10. Comece a gravar jogos em segundos, completei a carreira jogáveis de console e transmita jogos de seu console Xbox One para seu dispositivo Windows 10 de qualquer lugar em sua casa.



compaq.com.br





# VAIO FE14

The Art of Technology

Handwritten blue ink marks, possibly initials or a signature, located in the bottom right corner of the page.

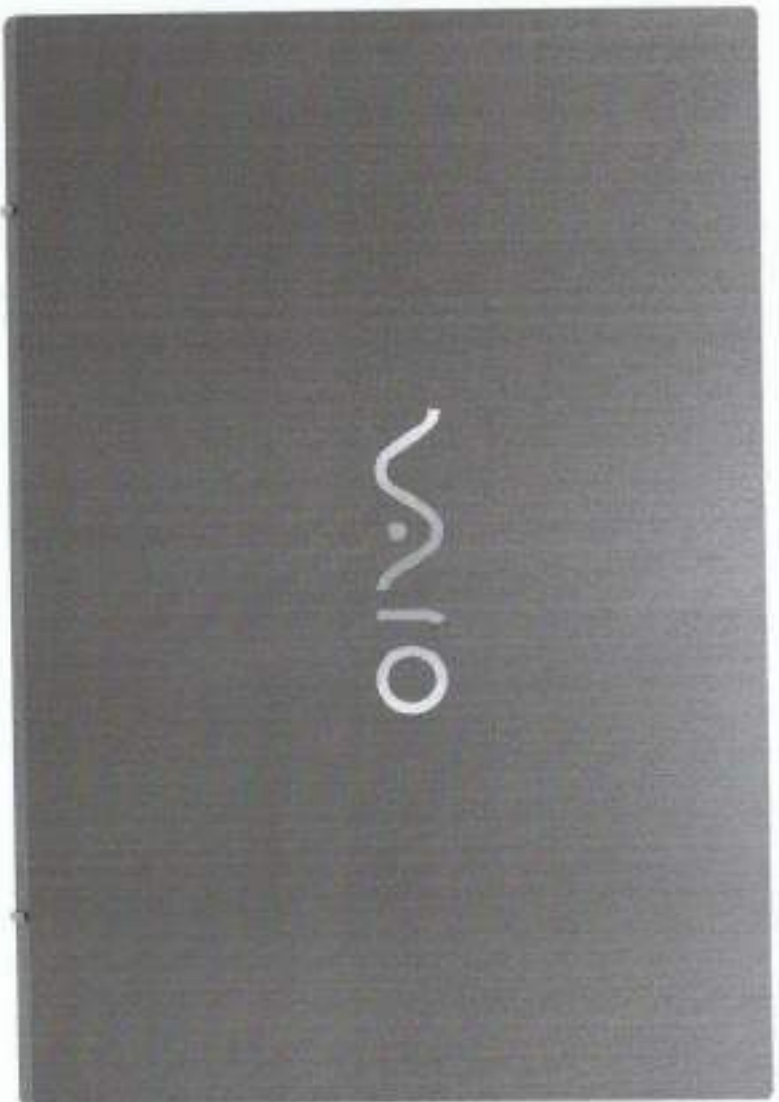


<b>Modelo</b>	<b>VAIO FE14</b>
<b>Processador</b>	Decima Geração Intel® Celeron®, Pentium® e Core™ i3, i5 e i7 Serie U
<b>Placa-mãe</b>	VAIO
<b>Sistema Operacional (opções)</b>	Windows 10 Pro / Windows 10 Home / Windows 11 Pro / Windows 11 Home / Distribuições Linux
<b>BIOS/UEFI</b>	Desenvolvida pela Positivo Tecnologia
<b>Memória RAM</b>	4GB / 8GB / 16GB / 32GB
<b>Slots de Memória</b>	1x slot DDR4 com suporte até 32 GB
<b>Armazenamento</b>	SSD PCIe NVMe até 512GB / SATA
<b>Abertura do LCD</b>	HDD 2.5" até 1TB 5400RPM / 7200RPM
<b>Leitor de Cartões</b>	135°
<b>Webcam</b>	SD Card
<b>Tela</b>	Câmera Frontal 1MP
<b>Vídeo</b>	LCD 14", Widescreen, Antirreflexiva com resolução 1920 x 1080 Full HD, com tecnologia LED
<b>Áudio</b>	Processamento de vídeo integrado Intel® UHD Graphics, Suporte Microsoft® DirectX® 12 e OpenGL 4.5
<b>Conectividade (opções)</b>	Microfone e alto-falantes estéreo embutidos
<b>Portas de Conexão</b>	LAN: 10/100/1000 Mbps, padrão Gigabit Ethernet Rede sem fio Intel® Dual Band Wireless-AC, Bluetooth 5, M.2 2230 Rede sem fio Intel® Dual Band Wi-Fi 6, Bluetooth 5.2, M.2 2230 2x USB 3.2 (Geração 1 Tipo A, com suporte a carregamento), 1x USB 3.2 (Geração 1 Tipo C), 1x USB 2.0, 1x HDMI, 1x (Áudio para microfone + Áudio para fone de ouvido), 1x RJ45 LAN Gigabit, 1x DCIN (alimentação do produto)
<b>Teclado</b>	Português-Brasil, ABNT2 com suporte a derramamento de água
<b>Mouse</b>	Tipo Touchpad com controle preciso do cursor, com toque múltiplo, 2 botões integrados
<b>Carregador</b>	19V / 2.1A / 100W-240V Automático, 40W
<b>Bateria</b>	Li-Ion Polímero / 3 células / 37Wh / interna
<b>Cor (opções)</b>	Chumbo escuro ou Branco
<b>Dimensões</b>	324 x 221 x 19,8 mm (Largura x Profundidade x Altura)
<b>Peso Líquido</b>	1,55 Kg
<b>Conteúdo da embalagem</b>	Notebook, Adaptador CA com cabo padrão 1m metro (carregador), Guia Rápido e Certificado de Garantia
<b>Segurança</b>	Entrada para Trava Kensington® Lock Solução TPM 2.0 integrado no SoC   Chip de criptografia TPM 2.0 discreto (opcional) Leitor de impressões digitais junto ao touchpad (opcional) Limpeza segura dos dados do SSD via UEFI em conformidade com as diretrizes do NIST SP800-88 UEFI compatível com a diretiva de segurança NIST SP 800-147



10  
K  
D

Qualquer uso e/ou reprodução não autorizada sem a permissão da Prefeitura Municipal de Boavista.



# VAIO FE14

The Art of Technology

Handwritten notes in blue ink, including a checkmark and some illegible scribbles.



**SMS**Uma marca  
do Grupo | **legrand**

# STATION II

## UPS LINE INTERACTIVE ● 600, 800 e 1200 VA



### A PROTEÇÃO IDEAL PARA SUAS ATIVIDADES COM DESIGN DIFERENCIADO

- › Modelo monovolt e bivolt automático
- › Estabilizador interno com 4 estágios de regulação
- › Filtro de linha interno
- › Forma de onda senoidal por aproximação
- › Botão liga/desliga temporizado com função Mute
- › Battery Saver: preserva a vida útil da bateria
- › Porta fusível externo com unidade reserva
- › LED bicolor no painel frontal



COMPUTADOR



PUNTO DE VENDA

IMPRESSORA  
MÚLTIPLA-FUNÇÕES  
E NO DE TINTA

SERVIDOR

CÂMERA  
CENTRAL CITY

JOGOS

EQUIPAMENTOS  
MUSICAIS

Handwritten notes in blue ink: a large '9' at the top right, and '12' with a checkmark and an 'X' below it.



# STATION II

Proteção e autonomia para os equipamentos de informática, áudio e vídeo

## 6 Níveis de Proteção

O nobreak Station II alia segurança e praticidade em um produto com design exclusivo. Fabricado em plástico ABS, possui a função DC Start, que possibilita ligar o nobreak mesmo na ausência de rede elétrica. Além disso, possui um recarregador inteligente que preserva a vida útil da bateria.

- Sobreaquecimento no transformador.
- Potência excedida.
- Descarga total da bateria.
- Curto-circuito no inversor.
- Surtos de tensão entre fase e neutro.
- Sub/sobretensão da rede elétrica. Na ocorrência destas, o nobreak passa a operar em modo bateria.

EQUIPAMENTOS QUE TAMBÉM PODEM SER LIGADOS:



## PROTEÇÕES PARA CARGA CONTRA PROBLEMAS DA REDE ELÉTRICA

### Queda de rede (Blackout)

Mantém o fornecimento de energia nas saídas durante a ausência total da rede elétrica.

### Ruído de rede elétrica

Possui filtro de linha interno que atenua ruídos provenientes da rede elétrica, fornecendo energia limpa para os equipamentos.

### Sobretensão de rede elétrica

Durante a ocorrência deste evento, o nobreak utiliza energia das baterias, mantendo a saída em um valor adequado.

### Subtensão de rede elétrica

Durante a ocorrência deste evento, o nobreak utiliza energia das baterias, mantendo a saída em um valor adequado.

### Surtos de tensão na rede

A rede elétrica pode apresentar picos de tensão provenientes, principalmente por descargas elétricas. A proteção é de modo comum e diferencial (entre fase-neutro ou fase-fase).

### Correção de variação da rede elétrica por degrau

A correção da tensão de saída é realizada através do estabilizador interno e é feita por taps de regulação.



Handwritten notes: 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100.

# CARACTERÍSTICAS GERAIS

- **Microprocessado:** Microprocessador RISC de alta velocidade com memória Flash, integrando diversas funções periféricas, aumentando a confiabilidade e o desempenho do circuito eletrônico.
- **Função TRUE RMS:** Analisa corretamente os distúrbios da rede elétrica permitindo a atuação precisa do equipamento.
- **Autoteste:** Ao ser ligado, o nobreak testa todos os circuitos internos, inclusive as baterias.
- **Autodiagnóstico de bateria:** Informa quando a bateria precisa ser substituída.
- **Recarregador Strong Charger:** Permite a recarga das baterias mesmo com níveis muito baixos de carga.
- **Recarga automática de bateria em 4 estágios com compensação de temperatura mesmo com o nobreak desligado, mantendo-a sempre em condições ideais de operação, contribuindo para melhor preservação de sua vida útil.**
- **Permite ser ligado na ausência de rede elétrica (DC Start).**
- **Circuito desmagnetizador:** Garante o valor de tensão adequado na saída do nobreak para equipamentos de informática e similares (cargas não lineares).
- **Gabinete plástico anti-chama.**
- **Alarme audiovisual para queda de rede, subtensão, fim do tempo de autonomia, final de vida útil da bateria, sobretensão, potência excedida e sobret temperatura.**
- **Garantia de 1 ano + 6 meses mediante cadastro no site: [www.sms.com.br/garantia](http://www.sms.com.br/garantia).**



## TEMPO DE AUTONOMIA<sup>(1)</sup>

Configuração Típica	μST 600 (1 bateria 12Vcc x 7Ah)	μST 800 (1 bateria 12Vcc x 7Ah)	μST 1200 (2 baterias 12Vcc x 5Ah)
PC (on-board) + monitor LCD 17"	25 min	25 min	40 min

<sup>(1)</sup> O tempo de autonomia varia significativamente de acordo com as condições de uso do sistema, do número de ciclos de carga e descarga, da temperatura ambiente, tipo e estado de conservação dos equipamentos ligados ao sistema.

134

Uma marca do Grupo | legrand



## STATION II

### UPS line interactive

Características Técnicas								
Código		27395	27390	27392	27396	27391	27393	
Características de Entrada		600 B1	800 B1	1200 B1	600 S	800 S 115	800 S 220	1200 S
Tensão nominal [V-]		Bivolt automático 115 / 127 / 220			115 / 127	115 / 127	220	115 / 127
Variação máxima [V-]		88 a 141 (rede 115V-) 170 a 262 (rede 220V-)	89 a 140 (rede 115V-) 176 a 264 (rede 220V-)	88 a 141 (rede 115V-) 170 a 262 (rede 220V-)	88 a 141	89 a 140	175 a 265	88 a 141
Frequência de rede <sup>(1)</sup> [Hz]		60 ± 4						
Plugue do cabo de força		Padrão NBR T4136						
Características de Saída								
Potência máxima [VA]		600	800	1200	600	800	1200	
Fator de potência		0,5						
Tensão nominal [V-]		115			220		115	
Regulação		± 5% (para operação bateria) <sup>(2)</sup> ± 4% - 10% (para operação rede)						
Frequência		60Hz ± 1% (para operação bateria)						
Forma de onda do inversor		Senoidal por aproximação (retangular PWM - controle de largura e amplitude)						
Número de tomadas (Padrão NBR 14136)		4 tomadas	6 tomadas	4 tomadas	6 tomadas			
Características Gerais								
Rendimento (Modo Rede)		93%	95%	93%	95%			
Acionamento do inversor		± 0,8 ms						
Bateria interna		1 bateria 12Vdc / 7Ah	2 baterias 12Vdc / 5Ah	1 bateria 12Vdc / 7Ah		2 baterias 12Vdc / 5Ah		
Peso líquido		5,3	5,5	9,8	5,2	5,4	8,0	
Peso bruto		5,6	5,8	10,4	5,5	5,7	8,6	
Dimensões A x L x P [mm]		175 x 99 x 298	199 x 99 x 275	250 x 122 x 330	175 x 99 x 298	199 x 99 x 275	290 x 122 x 330	
Comprimento do cabo de força do nobreak		1200 ± 50	1500 ± 50	1200 ± 50	1500 ± 50			
Protetor entre fase e neutro:							426	276
- Máxima energia de surto <sup>(3)</sup> [J]		276					4500	4500
- Máxima corrente de pico 8/20µs [A]		4500					275	175
Tensão de operação [V-]		175						

<sup>(1)</sup> Modelo Station sempre em rede. <sup>(2)</sup> 50% de rendimento TLI PFC por fase e tensão de saída. <sup>(3)</sup> Se usado com MD (C) (C categoria 4).



Av. Presidente, 661  
09190-030 - Diadema - SP  
☎: (11) 4075 7066  
(11) 3575 3500

**SMS**  
Uma marca do Grupo **Legrand**

WWW.SMS.COM.BR

LEI Nº 14.063, DE 23 DE SETEMBRO DE 2020

VERIFIQUE A AUTENTICIDADE DESTA DOCUMENTO EM <https://portalcidadao.prefeitura.boavista.br/verificacao.aspx> INFORMANDO O CODIGO: 876375075

## Folha de Especificações

**brother**  
 at your side

**DCP-8157DN**

Copie • Imprima • Digitalize

**Multifuncional Laser de Alta Velocidade  
 com Rede e Duplex Completo**

 Duplex Automático  
 para Impressão / Cópia /  
 Digitalização

Single-Pass Scanner

 Impressão a partir  
 de Dispositivo Móvel<sup>1)</sup>

 Impressão e Cópia  
 de até 40 ppm

 7 de Setembro de 2011  
 impressora Brother  
 © 2011 BROTHER INC.

**Principais Características:**
**Impressão e Cópia Rápidas**

- Aumente sua produtividade com impressão e cópia de até 40 ppm (preto).

**Duplex Completo**

- Imprima documentos frente e verso e economize papel. Além disso, digitalize ambos os lados da folha simultaneamente em uma única passagem com cópia/digitalização duplex (Single-Pass Scanner).

**Rede Integrada**

- Compartilhe a impressora em rede através da interface Ethernet integrada.

**Manuseio Flexível de Papel**

- Bandeja ajustável com capacidade para 250 folhas, mais uma bandeja multiuso com capacidade para 50 folhas. Alimentador automático de documentos (ADF) com capacidade para 50 folhas.

**Impressão Sem Fio a Partir do Dispositivo Móvel (Tablet e/ou Celular)<sup>2)</sup>**

- Compatível com AirPrint™, Google Cloud Print™, Brother™ iPrint&Scan, Cortado Workplace.

**Vidro do Scanner Tamanho Ofício**

- Copie, ou digitalize em cores, materiais encadernados ou documentos até tamanho Ofício.

**Cartucho de Toner de Ultra Rendimento**

- Reduza seus custos operacionais com o cartucho de reposição de ultra rendimento\* (até 12.000 páginas\*\*).

**Impressão/Digitalização Direta**

- Possui Interface USB Frontal que permite imprimir e digitalizar para um pen drive USB\*.

**Cópia Independente**

- Não é preciso estar conectado a um computador para fazer cópias. Reduza/aumente documentos de 25 a 400%, em incrementos de 1%.

**Suporte ao Consumidor**

- 1 ano de garantia limitada com suporte técnico por telefone durante a vida útil da sua máquina.


<sup>1)</sup> Disponível em alguns modelos e configurações.

<sup>2)</sup> Requer conexão Wi-Fi e acesso à Internet. Para mais informações, consulte o manual de instruções e o site de suporte ao cliente da Brother.

<sup>3)</sup> Disponível em alguns modelos e configurações.

<sup>4)</sup> Taxa máxima de impressão de texto em modo de impressão de alta velocidade (A4, modo de impressão de alta velocidade).

Todas as marcas comerciais e logotipos mencionados aqui são de propriedade de suas respectivas empresas. Windows, Windows Server, Windows Vista e o logotipo de Windows são marcas comerciais ou marcas registradas da Microsoft. Google, Google Cloud Print, Google Cloud Print logo são marcas comerciais da Google Inc. AirPrint™ e o logotipo de AirPrint são marcas comerciais da Apple Inc. Cortado Workplace é uma marca registrada da Cortado AG. Todos os outros nomes de produtos e serviços são marcas comerciais de seus respectivos proprietários.

\*Cópia de segurança não é uma cópia.


 16 X  
 X  
 9

# DCP-8157DN

## Folha de Especificações



### Especificações:

Método de Impressão	Laser Eletrofotográfico
Display LCD (tipo/tamanho)	Display LCD monocromático de 5 linhas/23 caracteres
Tamanho do Papel (máximo)	A4 21,6 cm x 35,6 cm (tamanho ofício)
Velocidade de Impressão (máxima)	Até 40 ppm
Resolução de Impressão (máxima)	Até 1200 x 1200 dpi
Emuladores	PCLE, BB-Script <sup>2</sup> , IBM Proprieter, Epson FX
Memória (padrão/máxima*)	128 MB/384 MB
Capacidade de Impressão Duplex	Sim
Capacidade de Entrada de Papel (máxima)**	Bandeja com capacidade para 350 folhas Bandeja multiuso com capacidade para 50 folhas
Capacidade de Entrada Opcional (máxima)**	Bandeja com capacidade para 500 folhas
Capacidade de Saída (máxima)**	150 folhas (face para baixo), 1 folha (face para cima)
Alimentador Automático de Documentos (ADF)	Até 50 páginas
Interfaces Padrão	Ethernet e USB
Interface USB Direta A / PictBridge	Sim/780
Velocidade de Cópia (máxima)	Até 40 cpm
Capacidade de Cópia Duplex	Sim
Resolução de Cópia (máxima)	1200 x 600 dpi
Redução/Ampliação de Cópia	De 25 a 400%, em incrementos de 1%
Opções de Cópia	Cópias Ordenadas, N em 1, Múltiplas (até 99) Cópia de Documentos de Identidade (ID)
Tipo de Scanner	Mesa Plana Colorida (Vidro com Alimentador Automático de Documentos (ADF))
Tamanho do Vidro do Scanner	21,6 cm x 35,6 cm (tamanho ofício)
Capacidade de Digitalização Duplex	Sim
Resolução do Scanner (máxima)	Óptica: Até 1200 x 1200 dpi Interpolada: Até 19200 x 33200 dpi
Função "Digitalizar para"	Arquivo, Imagem, E-mail, OCR, FTP, USB, Pasta de Rede (SMB)
Visualizador de Documentos/Software de OCR	Nuance Paperport <sup>®</sup> 12SE com OCR para Windows <sup>®</sup> Primo <sup>®</sup> PageMaker <sup>®</sup> 18 com OCR para Mac <sup>®</sup>
Compatibilidade do Sistema Operacional	Windows <sup>®</sup> : Windows 7 / Windows Vista <sup>®</sup> / XP / XP Professional x64 / Server 2003 / 2008 <sup>®</sup> Mac <sup>®</sup> : Mac OS X v10.5.8, v10.6.x, v10.7.x Linux
Compatibilidade do Dispositivo Móvel	Android <sup>™</sup> , Google Cloud Print <sup>™</sup> , Brother <sup>™</sup> iPrint&Scan, Conectado WorkPlace
Ciclo de Trabalho Mensal Máximo <sup>2</sup>	Até 50.000 páginas/mês
Volume Mensal Recomendado <sup>2</sup>	De 100 a 3.500 páginas/mês
Garantia	1 ano de garantia limitada

- \* Disponível em PostScript<sup>™</sup> PS
- \*\* Disponível em opção de compra separada
- 1 Capacidade máxima baseada em papel de 78 g/m<sup>2</sup>
- 2 Impressão em modo normal a 12, 24, 48, 96 e 192 páginas por minuto (ppm)
- 3 Disponível em opção de compra separada
- 4 Disponível em opção de compra separada
- 5 Disponível em opção de compra separada
- 6 Disponível em opção de compra separada
- 7 Disponível em opção de compra separada
- 8 Disponível em opção de compra separada
- 9 Disponível em opção de compra separada
- 10 Disponível em opção de compra separada
- 11 Disponível em opção de compra separada
- 12 Disponível em opção de compra separada
- 13 Disponível em opção de compra separada
- 14 Disponível em opção de compra separada
- 15 Disponível em opção de compra separada
- 16 Disponível em opção de compra separada
- 17 Disponível em opção de compra separada
- 18 Disponível em opção de compra separada
- 19 Disponível em opção de compra separada
- 20 Disponível em opção de compra separada
- 21 Disponível em opção de compra separada
- 22 Disponível em opção de compra separada
- 23 Disponível em opção de compra separada
- 24 Disponível em opção de compra separada
- 25 Disponível em opção de compra separada
- 26 Disponível em opção de compra separada
- 27 Disponível em opção de compra separada
- 28 Disponível em opção de compra separada
- 29 Disponível em opção de compra separada
- 30 Disponível em opção de compra separada
- 31 Disponível em opção de compra separada
- 32 Disponível em opção de compra separada
- 33 Disponível em opção de compra separada
- 34 Disponível em opção de compra separada
- 35 Disponível em opção de compra separada
- 36 Disponível em opção de compra separada
- 37 Disponível em opção de compra separada
- 38 Disponível em opção de compra separada
- 39 Disponível em opção de compra separada
- 40 Disponível em opção de compra separada
- 41 Disponível em opção de compra separada
- 42 Disponível em opção de compra separada
- 43 Disponível em opção de compra separada
- 44 Disponível em opção de compra separada
- 45 Disponível em opção de compra separada
- 46 Disponível em opção de compra separada
- 47 Disponível em opção de compra separada
- 48 Disponível em opção de compra separada
- 49 Disponível em opção de compra separada
- 50 Disponível em opção de compra separada
- 51 Disponível em opção de compra separada
- 52 Disponível em opção de compra separada
- 53 Disponível em opção de compra separada
- 54 Disponível em opção de compra separada
- 55 Disponível em opção de compra separada
- 56 Disponível em opção de compra separada
- 57 Disponível em opção de compra separada
- 58 Disponível em opção de compra separada
- 59 Disponível em opção de compra separada
- 60 Disponível em opção de compra separada
- 61 Disponível em opção de compra separada
- 62 Disponível em opção de compra separada
- 63 Disponível em opção de compra separada
- 64 Disponível em opção de compra separada
- 65 Disponível em opção de compra separada
- 66 Disponível em opção de compra separada
- 67 Disponível em opção de compra separada
- 68 Disponível em opção de compra separada
- 69 Disponível em opção de compra separada
- 70 Disponível em opção de compra separada
- 71 Disponível em opção de compra separada
- 72 Disponível em opção de compra separada
- 73 Disponível em opção de compra separada
- 74 Disponível em opção de compra separada
- 75 Disponível em opção de compra separada
- 76 Disponível em opção de compra separada
- 77 Disponível em opção de compra separada
- 78 Disponível em opção de compra separada
- 79 Disponível em opção de compra separada
- 80 Disponível em opção de compra separada
- 81 Disponível em opção de compra separada
- 82 Disponível em opção de compra separada
- 83 Disponível em opção de compra separada
- 84 Disponível em opção de compra separada
- 85 Disponível em opção de compra separada
- 86 Disponível em opção de compra separada
- 87 Disponível em opção de compra separada
- 88 Disponível em opção de compra separada
- 89 Disponível em opção de compra separada
- 90 Disponível em opção de compra separada
- 91 Disponível em opção de compra separada
- 92 Disponível em opção de compra separada
- 93 Disponível em opção de compra separada
- 94 Disponível em opção de compra separada
- 95 Disponível em opção de compra separada
- 96 Disponível em opção de compra separada
- 97 Disponível em opção de compra separada
- 98 Disponível em opção de compra separada
- 99 Disponível em opção de compra separada
- 100 Disponível em opção de compra separada

### Suprimentos e Acessórios:

TN-3342	Toner com Rendimento Padrão (aprox. 3.000 páginas)*
TN-3382	Toner com Alto Rendimento (aprox. 8.000 páginas)*
TN-3392	Toner com Ultra Rendimento (aprox. 12.000 páginas)*
DR-3302	Unidade do Cilindro (aprox. 60.000 páginas)*
LT-5400	Bandeja Opcional para 500 Folhas

### Informações sobre Configuração:

Código UPC	012502630869
Dimensões da Unidade	49,0 x 41,7 x 41,4 cm (LxAxP)
Peso da Unidade	16,5 kg
Dimensões da Caixa	93,4 x 60,5 x 51,5 cm (LxAxP)
Peso da Caixa	30,2 kg
País de Origem	Vietnã
Tarifa/Código Harmonizado	8443.31.6000

### Conteúdo da Caixa:

- Brother<sup>™</sup> DCP-8157DN
- Unidade do Cilindro DR5302 (60.000 páginas)\*
- Guia de Configuração Rápida e Guia Básico do Usuário
- CD-ROM de Instalação
- Cabo de Força



Trabalhando com você para um meio ambiente melhor. Na Brother, nossa iniciativa verde é simples. Não nos esforçamos para assumir a responsabilidade, agir com respeito e tratar toda uma diferença positiva para ajudar a construir uma sociedade onde o desenvolvimento sustentável pode ser alcançado. Não é só o serviço, esta abordagem da Brother Earth. [www.brotherearth.com](http://www.brotherearth.com)

### Código de Barras Digitalizável:



Handwritten signatures and initials in blue ink.

DCP-L3551CDW

brother  
at your side

# EFFORTLESS BRILLIANCE

Discover the DEFINITION of simplicity



DCP-L3551CDW



Productivity



Connectivity



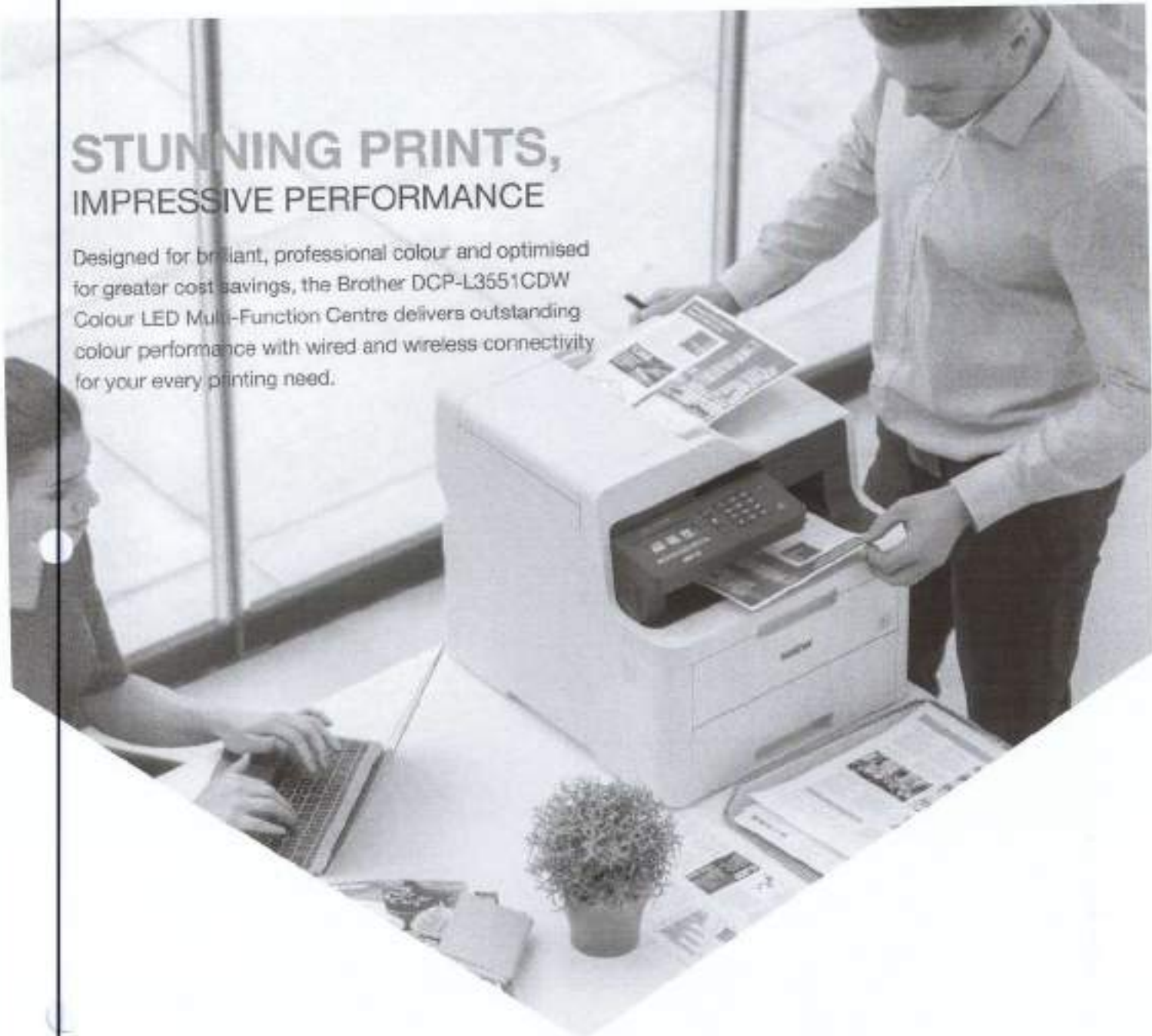
Cost-saving

Handwritten notes: 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100



## STUNNING PRINTS, IMPRESSIVE PERFORMANCE

Designed for brilliant, professional colour and optimised for greater cost savings, the Brother DCP-L3551CDW Colour LED Multi-Function Centre delivers outstanding colour performance with wired and wireless connectivity for your every printing need.



### Remarkable Productivity



#### Fast Print and Scan Speeds

Boost your productivity with fast colour print speeds of up to 180ppm and scan speeds\* of up to 27ppm (A4).



#### Seamless Scanning

The 50 sheet Auto Document Feeder scans multi-page documents in a single pass, cutting down on scan times.



#### Comprehensive Paper Handling

With the 250 sheet default paper tray, the need for frequent paper reloading is reduced. In addition, the single-sheet manual feed slot can handle a greater variety of papers up to 163gsm for your printing convenience.

Handwritten signatures and initials in blue ink, including a large 'G' and some illegible marks.



## Optimise Cost-Savings



### Cut Your Paper Waste by 50%

Reduce your paper wastage and expenses with automatic 2-sided printing.



### Lower Your Toner Costs

Affordable high-yield toner cartridges offer lower cost-per-page, and Toner Save Mode uses less toner while producing crisp and vibrant printouts.



### Wireless and Wired Sharing

Effortlessly set up and start sharing your Brother DCP-L3551CDW with multiple users on your network via its built-in Wi-Fi and Ethernet.



### Print from Your Mobile

Connect your iPhone, iPad, Android™ or Windows® mobile devices and start printing and scanning via Brother iPrint&Scan and other compatible mobile applications.



## A Proud Legacy of Uncompromising Reliability

### Over A Hundred Years of Customer Satisfaction

Even with over a century of customer satisfaction under our belt, we aren't taking anything for granted. All Brother machines have undergone rigorous tests and trials to ensure that they will last for years to come. By bringing together traditional Japanese values of quality, service, and integrity with the latest in technological innovation, you can always count on Brother to take your office productivity further.

### An Award-Winning Track Record

For 5 years in a row, Brother has won PCmag's prestigious Business Choice Award. We have also won PCmag's Readers' Choice Award for an unprecedented 9 years in a row, scoring the highest in overall satisfaction and reliability.

### Peace of Mind. Guaranteed.

When you purchase a Brother printer or multi-function centre and our genuine inks or toners, we'll support you every step of the way with our FREE 3-year warranty commitment. YOU DO NOT NEED TO REGISTER YOUR PRODUCT, our warranty is automatic when you purchase your Brother device. To keep your Brother machine running in optimal condition we recommend using only Brother Genuine Consumables, this will ensure you automatically enjoy our exclusive 3-year warranty.



**6 YEARS  
IN A ROW**  
2018



**10 YEARS  
IN A ROW**  
2010



20x  
d e 9



# DCP-L3551CDW

Key Functions		Print, Copy, Scan Automatic 2-sided Colour Print with Wireless Connectivity
<b>PRINT FUNCTIONS</b>		
Processor	Main: Cortex-A9 800MHz; Sub: ARM946 133MHz	
Print Speed (A4/Letter)	Up to 18/18 ppm	
Print Features	Automatic 2-sided Printing Poster Printing, Header-Footer Printing, Watermark Printing (Windows Post-Based/PCLXL) N-up Printing, SecurePrint (Windows Post-Based/PCLXL) / Mac	
Print Resolution	600 x 600 dpi, 2,400 dpi (600 x 2,400) quality	
Warm Up Time (From Sleep)	Less than 28 sec	
First Print Out Time (Ready)	Less than 15.5 sec (Mono/Colour)	
<b>COPY FUNCTIONS</b>		
Copy Speed	Up to 18/18 cpm	
Reduction/Enlargement (%)	25% - 400% in 1% increments	
Copy Resolution	600 x 600 dpi	
Copy Features	N in 1	
<b>SCAN FUNCTIONS</b>		
Scan Speed (Mono/Colour)*	27/21 ipm (A4); 29/22 ipm (LTR)	
Resolution (Optical)	1,200 x 2,400 dpi	
Resolution (Interpolated)	19,200 x 19,200 dpi	
Scan Features	Auto Detect, SharePoint (Firm & CC4), Email, Image, OCR, File, FTP, Network (Windows only)	
<b>PAPER HANDLING</b>		
Paper Input Tray	250 sheets	
Manual Feed Slot	1 sheet	
Auto Document Feeder	50 sheets	
Paper Output	100 sheets Face Down	
Media Weights / Manual Feed Slot	60 to 163g/m <sup>2</sup>	
Media Sizes / Standard Tray	A4, Letter, A5, A5 Long Edge, A6, Executive, Legal, Folio	
<b>CONNECTIVITY</b>		
Interface	Hi-Speed 2.0, 10Base-T/100Base-TX, IEEE 802.11b/g/n (Wireless), IEEE 802.11g/n (Wi-Fi Direct)	
Mobile Solutions	Brother Print&Scan, Google Cloud Print, AirPrint, Mopria®, Windows Mobile Printing	
<b>OTHERS</b>		
Memory	512MB	
Emulation	PCL6, BR-Script3, PDF Version 1.7, XPS Version 1.0	
Display	16 characters x 2 lines	
OS Capability	Windows: Server 2008, Server 2008 R2, 7 SP1, 8, 8.1, Server 2012, Server 2012 R2, 10, Server 2016, Mac: macOS v10.11.6, 10.12.x, 10.13.x (Download only), Linux: GLPS, LRD/LPRng (686/x64 environment)	
Utilities & Solutions	BRAdmin Light®, BRAdmin Pro®, Web Based Management	
Security Features	Setting Lock, Secure Function Lock Version 3.0*	
Energy Star	Yes	
Power Consumption (Printing / Deep Sleep)	Approx. 400W / 1.2W	
Power Source	220-240 V, AC 50Hz	
Noise Level	LpA(m) = 46 dB(A)	
<b>SUPPLIES</b>		
Box Toner Yield	BK: Approx. 1,000 pages; CMY: Approx. 1,000 pages each	
High Yield Toner	TN-277BK: Approx. 3,000 pages; TN-277CMY: Approx. 2,300 pages each	
Drum Unit (DR-273CL C/M/Y/BK)	Approx. 58,000 pages	
<b>DIMENSIONS &amp; WEIGHT</b>		
Dimensions (W x D x H)	410 x 475 x 414 mm	
Machine Weight	24.6 kg	

In accordance with ISO/IEC 17911  
For Windows by download only  
Up to 100 registrable users  
In accordance with ISO/IEC 18750

**brother**  
at your side

Brother International South Africa  
Overseign Drive, Route 21 Corporate Park, Irene  
1709 JNB 017 SAC 2000 City 1709 JNB 017 SAC 2000



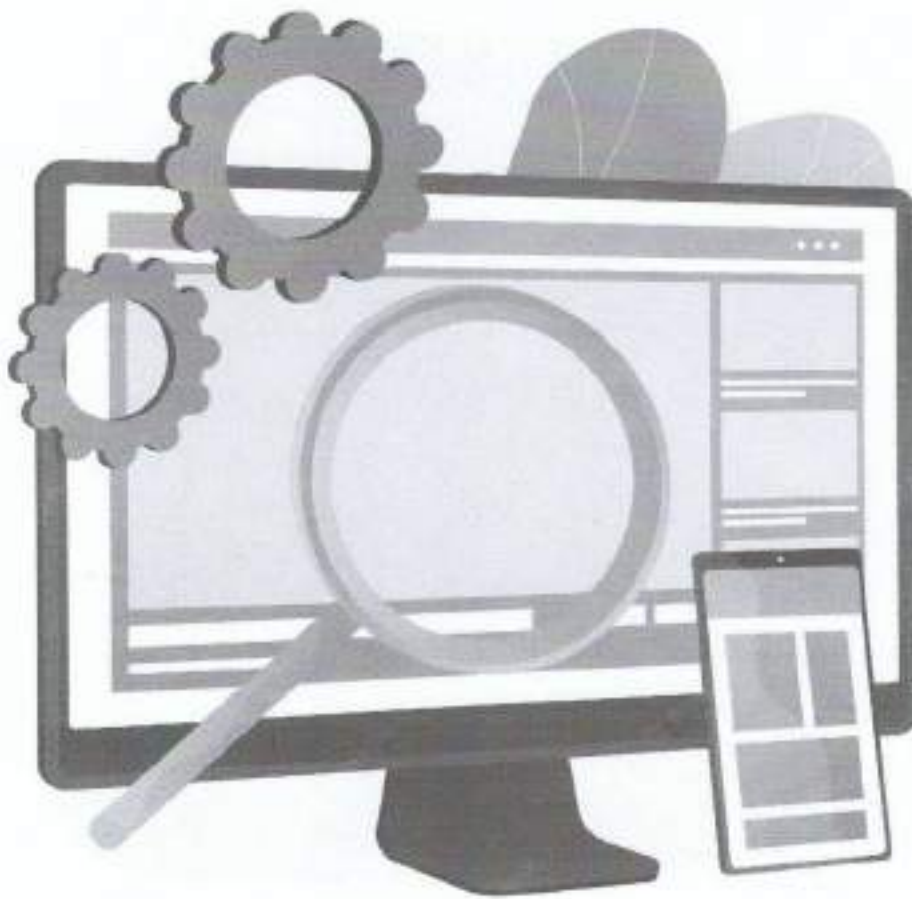
Brother  
Earth



Japanese  
Excellence

TAV Web

# Manual do Usuário



## Help Desk System

*Handwritten signature and date: 02/09/2023*



## SUMÁRIO

1. INTRODUÇÃO.....	3
2. FUNCIONALIDADES.....	3
3. BENEFÍCIOS ESPERADOS.....	3
4. COMO ACESSAR O SISTEMA?.....	3
5. SISTEMA APÓS FAZER O LOGIN - USUÁRIO.....	4
5.1. Menu Superior.....	5
5.2. Botão Adicionar Chamado e Aba de Pesquisa.....	5
5.3. Tabela de Chamados.....	5
5.4. Tela de Abertura de Chamado.....	6
5.5. Tela de Detalhes do Chamado.....	6
6. SISTEMA APÓS FAZER O LOGIN - ADMINISTRADOR.....	7
6.1. Menu Superior.....	8
6.2. Filtro Lateral.....	8
6.3. Tabela de Chamados.....	8
6.4. Tela de Modelos de Equipamentos.....	9
6.4.1. Botão Adicionar Novo Modelo de Equipamento e Aba de Pesquisa.....	9
6.4.2. Tabela de Modelo de Equipamentos.....	10
6.4.3. Tela de Cadastro de Modelo de Equipamento.....	10
6.4.4. Tela de Detalhes do Modelo de Equipamento.....	11
6.5. Tela de Equipamentos.....	11
6.4.1. Botão Adicionar Novo Equipamento e Aba de Pesquisa.....	12
6.4.2. Tabela de Equipamentos.....	12
6.4.3. Tela de Cadastro de Equipamento.....	13
6.4.4. Tela de Detalhes do Equipamento.....	13
6.6. Tela de Usuários.....	11
6.7. Tela de Relatório.....	11



  
 2



## 1. INTRODUÇÃO

Os Sistemas Help Desk são imprescindíveis nas organizações, para se ter maior controle e eficiência aos reparos prestados a equipamentos eletrônicos. O nosso Sistema Help Desk visa atuar sendo um intermediador entre o usuário final que tem em sua posse um equipamento para uso e o profissional que auxilia nas demandas referente ao mesmo.

## 2. FUNCIONALIDADES

- Cadastro de Modelo de Equipamento
- Cadastro de Equipamento
- Abertura de Chamado pelo Usuário
- Relatório Mensal

## 3. BENEFÍCIOS ESPERADOS

- Geração de Relatórios para auxiliar na tomadas de decisões
- Aumentar a eficiência e diminuir o tempo de reparo de equipamentos

## 4. COMO ACESSAR O SISTEMA?

Para ingressar na solução Help Desk System, o usuário deverá primeiramente acessar o endereço eletrônico onde está publicada a ferramenta e inserir os dados de usuário e senha nos campos indicados, conforme demonstrado na figura abaixo.

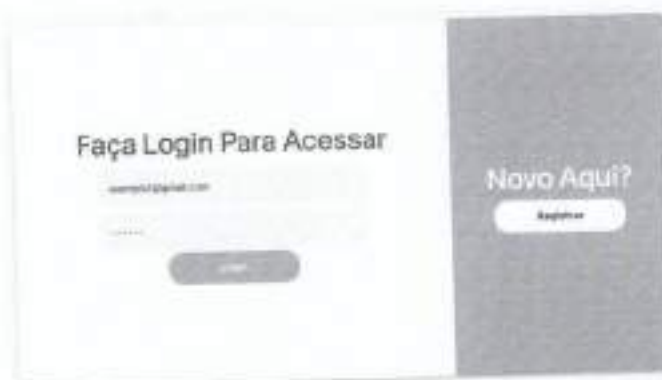


Figura 4-1

Caso o usuário ainda esteja registrado é só clicar no botão registrar na figura mostrada acima que o usuário será redirecionado para a página de registro como mostra a figura abaixo.

*Handwritten notes:*  
24  
3  
g

TAV Web

Figura 4-2

O sistema apenas irá registrar aqueles que preencherem os dados e pressionar para se registrar, após se registrar o usuário será redirecionado para a página de login onde ele conseguirá acessar o sistema após o administrador liberar o seu acesso (Figura 6.6.1).

## 5. SISTEMA APÓS FAZER O LOGIN - USUÁRIO

Ao realizar login, o usuário visualizará um menu na parte superior da tela, um botão, uma caixa de pesquisa e uma tabela com os chamados já cadastrados.

ID Chamado	Assunto	Data	Status	Ações
1	Assunto 1	12/12/2022	Aberto	Visualizar Chamado
2	Assunto 2	12/12/2022	Aberto	Visualizar Chamado
3	Assunto 3	12/12/2022	Aberto	Visualizar Chamado
4	Assunto 4	12/12/2022	Aberto	Visualizar Chamado
5	Assunto 5	12/12/2022	Aberto	Visualizar Chamado
6	Assunto 6	12/12/2022	Aberto	Visualizar Chamado
7	Assunto 7	12/12/2022	Aberto	Visualizar Chamado
8	Assunto 8	12/12/2022	Aberto	Visualizar Chamado

Figura 5-1

Handwritten notes in blue ink: '15', '12', '4', and a signature.

## 5.1 - Menu Superior

No menu superior aparece para o usuário a opção para ir para a tela principal dos chamados que é a mesma mostrada na "Figura 5-1" e a opção de Logout que permite ao usuário se desconectar do sistema.

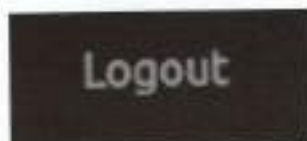


Figura 5.1-1

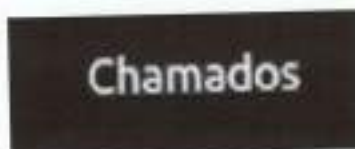


Figura 5.1-2

## 5.2 - Botão Adicionar Chamado e Aba de Pesquisa

Ao clicar no botão abaixo o usuário será redirecionado para a página onde poderá cadastrar um novo chamado como mostra a "Figura 5.4-1".

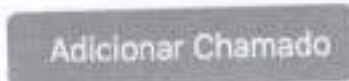


Figura 5.2-1

A aba de pesquisa permite ao usuário filtrar ou encontrar o que ele deseja na tabela presente na "Figura 5.3-1".

Pesquisar:

Pesquisar ...

Figura 5.2-2

## 5.3 - Tabela de Chamados

Abaixo temos a tabela que apresenta os chamados cadastrados pelo usuário, nela podemos ver o assunto, equipamento, data e status do chamado. Para vermos mais detalhes referente ao chamado é só clicar em cima do assunto do chamado desejado que o usuário será redirecionado para a página com os detalhes como mostra a "Figura 5.5-1". Se

Handwritten notes and signatures in the bottom right corner, including the number "5" and some illegible scribbles.

TAV Web

o chamado já foi resolvido, o usuário pode clicar no botão "Encerrar Chamado" para finalizar o chamado.

#	Assunto	Equipamento	Data	Status	Ações
1	Interação de tela de hardware	Impressora multifuncional HP DeskJet Ink Advantage 3774 com Wi-Fi	23/09/2022	Chamado Encerrado	Encerrar Chamado
2	Monitoramento de rede	Monitor 20 UltraWide 99WV0020-29	23/09/2022	Chamado em Aberto	Encerrar Chamado

Figura 5.3-1

#### 5.4 - Tela de Abertura de Chamado

Abaixo temos um formulário onde o usuário necessita preencher todos os campos para que possa abrir uma solicitação de chamado.

Figura 5.4-1

#### 5.5 - Tela de Detalhes do Chamado

Abaixo temos as informações detalhadas da solicitação do chamado registrada pelo usuário em sua abertura.

Handwritten signatures and initials in blue ink, including a large '6' and a '9'.

TAV Web

**Status: Chamado aberto**

**Dados do Chamado:**

Assunto	Problema com Wi-Fi no 1º andar
Localização	Edifício Administrativo - Centro Administrativo Boavista (1º andar)
Ordem de Prioridade	02
Local de Atendimento	010000
Descrição	Apresenta problemas de conexão com a internet

**Dados do Usuário:**

Nome	Felipe
Telefone	47332222
Responsável de Local	Tiago Lima

© Prefeitura de Boavista - 2023

Figura 5.5-1

## 6. SISTEMA APÓS FAZER O LOGIN - ADMINISTRADOR

Ao realizar login, o usuário visualizará um menu na parte superior da tela, um botão, uma caixa de pesquisa e uma tabela com os chamados já cadastrados.

**Todos os Chamados**

#	Assunto	Localização	Data	Status	Ações
1	Problema com Wi-Fi no 1º andar	Edifício Administrativo - Centro Administrativo Boavista (1º andar)	10/10/2023	Chamado Aberto	[Botão de Ação]
2	Problema com Wi-Fi no 1º andar	Edifício Administrativo - Centro Administrativo Boavista (1º andar)	10/10/2023	Chamado Aberto	[Botão de Ação]
3	Problema com Wi-Fi no 1º andar	Edifício Administrativo - Centro Administrativo Boavista (1º andar)	10/10/2023	Chamado Aberto	[Botão de Ação]

© Prefeitura de Boavista - 2023

Figura 6-1

Handwritten notes and signatures in the bottom right corner of the page, including the number '82' and several illegible signatures.

### 6.1 - Menu Superior

No menu superior aparece para o usuário a opção para ir para a tela principal dos chamados que é a mesma mostrada na "Figura 6-1" e demais telas (Modelo de Equipamentos, Equipamentos, Usuários e Relatório), a opção de Logout que permite ao usuário se desconectar do sistema.

**Chamados Modelos de Equipamentos Equipamentos Usuários Relatório**

Figura 6.1-1

**Logout**

Figura 6.1-2

### 6.2 - Filtro Lateral

Na parte esquerda encontramos os Filtros, nele o administrador consegue filtrar o que deseja ver na tabela ao lado (Figura 6.3-1).

**Filtros**

Assunto

Equipamento

Status

De:

Até:

Figura 6.2-1

### 6.3 - Tabela de Chamados

Abaixo temos a tabela que apresenta os chamados cadastrados, nela podemos ver o assunto, equipamento, data e status do chamado. Para vermos mais detalhes referente ao chamado é só clicar em cima do assunto do chamado desejado que o administrador será redirecionado para a página com os detalhes como mostra a "Figura 5.5-1". Se o chamado

Já foi resolvido, o administrador pode clicar no botão "Encerrar Chamado" para finalizar o chamado.

#	Assunto	Equipamento	Data	Status	Ações
1	Impressora não ou de funcionar	Impressora multifuncional HP DeskJet Ink Advantage 2774 com Wi-Fi	12/12/2022	Chamado Aberto	Encerrar Chamado
2	Monitor não ou de funcionar	Monitor LG Ultrawide 29WLS60-29	12/12/2022	Chamado Aberto	Encerrar Chamado
3	Equipamento está fazendo muito barulho	Optilex 3090 Micro	12/12/2022	Chamado Aberto	Encerrar Chamado

Figura 6.3-1

#### 6.4 - Tela de Modelos de Equipamentos

Ao clicar no item "Modelo de Equipamento" menu superior (Figura 6.1-1) o administrador é redirecionado para a página mostrada na figura abaixo.

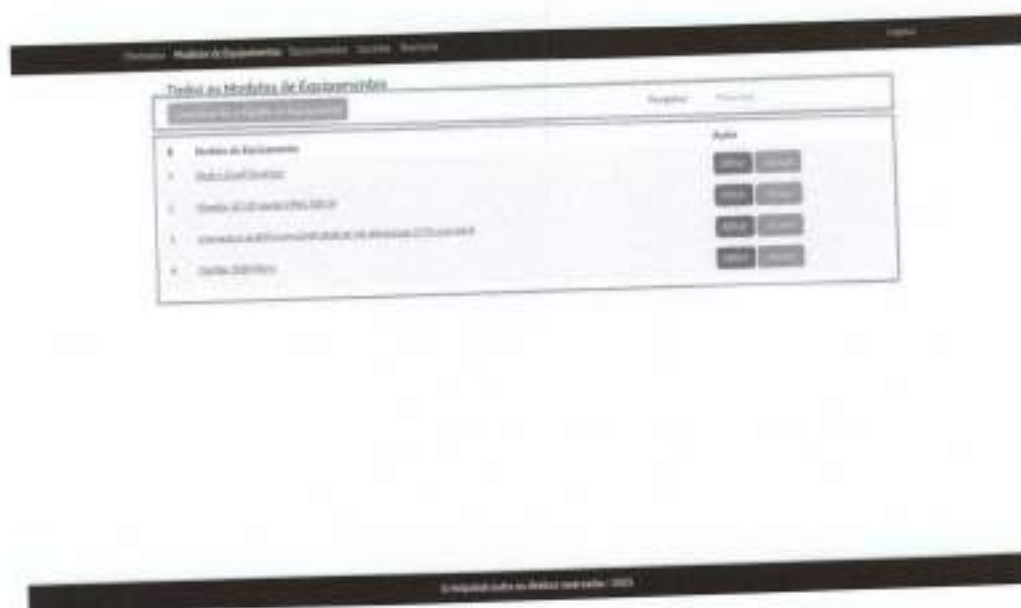


Figura 6.4-1

##### 6.4.1 - Botão Adicionar Novo Modelo de Equipamento e Aba de Pesquisa

Ao clicar no botão abaixo o usuário será redirecionado para a página onde poderá cadastrar um novo modelo de equipamento como mostra a "Figura 6.4.3-1".

Adicionar Novo Modelo de Equipamento

Figura 6.4.1-1

A aba de pesquisa permite ao usuário filtrar ou encontrar o que ele deseja na tabela presente na "Figura 6.4.2-1".

Pesquisar:

Figura 6.4.1-2

### 6.4.2 - Tabela de Modelo de Equipamentos

Abaixo temos a tabela que lista os modelos de equipamentos cadastrados, nela podemos ver o modelo de equipamento e as possíveis ações (Editar, Apagar). Para vermos mais detalhes referente ao modelo de equipamento é só clicar em cima do modelo de equipamento desejado que o administrador será redirecionado para a página com os detalhes como mostra a "Figura 6.4.4-1". O administrador ainda tem opção de editar conteúdo ou apagar como mostra a coluna de ações.

#	Modelo do Equipamento	Ações
1	Arco Small Desktop	<input type="button" value="Editar"/> <input type="button" value="Apagar"/>
2	Monitor LG UltraWide 35WQ500-CB	<input type="button" value="Editar"/> <input type="button" value="Apagar"/>
3	Impressora Multifuncional HP DeskJet 2720 com Wi-Fi	<input type="button" value="Editar"/> <input type="button" value="Apagar"/>
4	Cartão 3000 Micro	<input type="button" value="Editar"/> <input type="button" value="Apagar"/>

Figura 6.4.2-1

### 6.4.3 - Tela de Cadastro de Modelo de Equipamento

Abaixo temos um formulário onde o usuário necessita preencher todos os campos para que possa adicionar um novo modelo de equipamento.

31  
10  
9

© Copyright 2023 by Interim - Interimidade, 2023.

Figura 6.4.3-1

#### 6.4.4 - Tela de Detalhes do Modelo de Equipamento

Abaixo temos as informações detalhadas do modelo de equipamento registrado pelo usuário em sua abertura.

© Copyright 2023 by Interim - Interimidade, 2023.

Figura 6.4.4-1

#### 6.5 - Tela de Equipamentos

Ao clicar no item "Equipamento" menu superior (Figura 6.1-1) o administrador é redirecionado para a página mostrada na figura abaixo.

32

11



Figura 6.5-1

### 6.5.1 - Botão Adicionar Novo Equipamento e Aba de Pesquisa

Ao clicar no botão abaixo o usuário será redirecionado para a página onde poderá cadastrar um novo equipamento como mostra a "Figura 6.5.3-1".

Adicionar Novo Equipamento

Figura 6.5.1-1

A aba de pesquisa permite ao usuário filtrar ou encontrar o que ele deseja na tabela presente na "Figura 6.5.2-1".

Pesquisar:

Figura 6.5.1-2

### 6.5.2 - Tabela de Equipamentos

Abaixo temos a tabela que lista os equipamentos cadastrados, nela podemos ver o modelo de equipamento, código do equipamento, lugar de instalação e as possíveis ações (Editar, Apagar). Para vermos mais detalhes referente ao equipamento é só clicar em cima do modelo do equipamento desejado que o administrador será redirecionado para a página

com os detalhes como mostra a "Figura 6.5.4-1". O administrador ainda tem opção de editar conteúdo ou apagar como mostra a coluna de ações.

#	Modelo do Equipamento	Código do Equipamento	Lugar de Instalação	Ações
1	DESKTOP 2000 ANOS	401	Atendimento	Editar Apagar
2	DESKTOP 2000 MICRO	402	Secretaria	Editar Apagar
3	IMPRIMTA SMALL BUSINESS	501	Secretaria	Editar Apagar
4	MONITOR LG ULTRAVIEW 23WV000-ZB	502	Secretaria	Editar Apagar
5	IMPRESSORA MULTIFUNÇÃOAL (2) DUELJET 150 ADVANTAGE 2774 COM Wi-Fi	601	Escritório	Editar Apagar
6	IMPRESSORA MULTIFUNÇÃOAL (2) DUELJET 150 ADVANTAGE 2774 COM Wi-Fi	104	Secretaria	Editar Apagar
7	MONITOR LG ULTRAVIEW 23WV000-ZB	1034	Desenvolvimento	Editar Apagar
8	IMPRIMTA SMALL BUSINESS	1054	Desenvolvimento	Editar Apagar

Figura 6.5.2-1

### 6.5.3 - Tela de Cadastro de Equipamento

Abaixo temos um formulário onde o usuário necessita preencher todos os campos para que possa adicionar um novo equipamento.

Figura 6.5.3-1

#### 6.5.4 - Tela de Detalhes do Equipamento

Abaixo temos as informações detalhadas do equipamento registrado pelo o usuário em sua abertura.



Figura 6.5.4-1

#### 6.6 - Tela de Usuários

Ao clicar no item "Usuários" menu superior (Figura 6.1-1) o administrador é redirecionado para a página mostrada na figura abaixo, onde ele pode liberar o acesso ou negar o acesso de usuários que fizeram cadastro.



Figura 6.6-1

35

14

## 6.7 - Tela de Relatório

Ao clicar no item "Relatório" menu superior (Figura 6.1-1) o administrador é redirecionado para a página mostrada na figura abaixo.

**Relatório**

Resumo de Reservas

Resumo de Reservas	Total	Total em Aberto
Reservas em Aberto	1	1
Reservas Canceladas	0	0
Reservas Confirmadas	0	0
<b>Total</b>	<b>1</b>	<b>1</b>

Resumo de Pedidos

Resumo de Pedidos	Total	Total em Aberto
Pedidos em Aberto	0	0
Pedidos Cancelados	0	0
Pedidos Confirmados	0	0
<b>Total</b>	<b>0</b>	<b>0</b>

Resumo de Serviços

Resumo de Serviços	Total	Total em Aberto
Serviços em Aberto	0	0
Serviços Cancelados	0	0
Serviços Confirmados	0	0
<b>Total</b>	<b>0</b>	<b>0</b>

Resumo de Contratos

Resumo de Contratos	Total	Total em Aberto
Contratos em Aberto	0	0
Contratos Cancelados	0	0
Contratos Confirmados	0	0
<b>Total</b>	<b>0</b>	<b>0</b>

Figura 6.7-1

TAV Web

# Código Fonte



# Help Desk System

Handwritten marks in blue ink, including a vertical line, a checkmark, and some illegible scribbles.



TAV Web

## Back End

```

> node_modules
> src
  > controllers
    call.controller.ts
    equipment.controller.ts
    modelEquipment.controller.ts
    requestsToChangePassword.controller.ts
    user.controller.ts
  > database
    index.ts
  > entity
    call.ts
    ChangePassword.ts
    Equipment.ts
    EquipmentModel.ts
    User.ts
  > middleware
    authorization.middleware.ts
    call.verify.middleware.ts
    emailAlreadyExists.middleware.ts
    equipment.validate.middleware.ts
    equipmentModel.validate.middleware.ts
    index.ts
    isActive.middleware.ts
    isAdmin.middleware.ts
    requestExists.middleware.ts
    user.validate.middleware.ts
    userAlreadyActive.middleware.ts
    userNotExists.middleware.ts
  > routes
    call.routes.ts
    equipment.routes.ts
    index.ts
    modelEquipment.routes.ts
    requestsToChangePassword.routes.ts
    user.routes.ts
  > services
    call.service.ts
    equipment.service.ts
    equipmentModel.service.ts
    requestToChangePassword.service.ts
    user.service.ts
  > types
    call.types.ts
    equipment.types.ts
    equipmentModel.types.ts
    requestsToChangePassword.types.ts
    user.types.ts
  > utils
    error.ts
    app.ts
    server.ts
  .env
  .gitignore
  package-lock.json
  package.json
  README.md
  tsconfig.json
  yarn.lock

```

6 32

TAV Web

A pasta (node\_modules) e o arquivo (package-lock.json) não foram adicionados nesse documento por serem pastas criadas automaticamente ao criarmos uma aplicação Nodejs, somadas elas apresentam milhares de linhas que fazem o algoritmo funcionar e as mesmas não necessitam estar neste arquivo. O arquivo (yarn.lock) não foi adicionado por um arquivo gerado automaticamente ao instalarmos yarn no projeto e o arquivo (.env) não foi adicionado, nele apenas tem os dados para se conectar ao banco de dados

### src/controllers/call.controller.ts

```

import { Request, Response } from 'express';
import { createCallService, createCallService, getCallService, listCallService, updateCallService } from '../services/callService';

export const createCall = async (req: Request, res: Response, next: NextFunction) => {
  try {
    const data = req.body;
    const call = await createCallService(data);
    res.status(201).send(call);
  } catch (err) {
    next(err);
  }
};

export const getCall = async (req: Request, res: Response, next: NextFunction) => {
  try {
    const callId = parseInt(req.params.callId);
    const call = await getCallService(callId);
    res.status(200).send(call);
  } catch (err) {
    next(err);
  }
};

export const deleteCall = async (req: Request, res: Response, next: NextFunction) => {
  try {
    const callId = parseInt(req.params.callId);
    await deleteCallService(callId);
    res.status(204).send();
  } catch (err) {
    next(err);
  }
};

export const listCall = async (req: Request, res: Response, next: NextFunction) => {
  try {
    const call = await listCallService();
    res.status(200).send(call);
  } catch (err) {
    next(err);
  }
};

export const updateCall = async (req: Request, res: Response, next: NextFunction) => {
  try {
    const callId = parseInt(req.params.callId);
    const updateCallService = await updateCallService(callId);
    res.status(204).send();
  } catch (err) {
    next(err);
  }
};

export const createCallPart = async (req: Request, res: Response, next: NextFunction) => {
  try {
    const data = req.body;
    const callPart = await createCallPartService(data);
    res.status(201).send(callPart);
  } catch (err) {
    next(err);
  }
};

```

TAV Web

## src/controllers/equipment.controller.ts

```

1  import { NextFunction, Request, Response } from "express";
2  import { createEquipmentService, deleteEquipmentService, getOneEquipmentService,
3    listEquipmentsService, updateEquipmentService } from "../services/equipment.service";
4
5  export const createEquipment = async (req: Request, res: Response, next: NextFunction) => {
6    try {
7      const data = req.body;
8      const equipment = await createEquipmentService(data);
9      res.status(201).send(equipment);
10   } catch (err) {
11     next(err);
12   }
13 }
14
15 export const listEquipments = async (req: Request, res: Response, next: NextFunction) => {
16   try {
17     const equipments = await listEquipmentsService();
18     res.status(200).send(equipments);
19   } catch (err) {
20     next(err);
21   }
22 }
23
24 export const getOneEquipment = async (req: Request, res: Response, next: NextFunction) => {
25   try {
26     const equipmentId = parseInt(req.params.equipmentId);
27     const equipment = await getOneEquipmentService(equipmentId);
28     res.status(200).send(equipment);
29   } catch (err) {
30     next(err);
31   }
32 }
33
34 export const deleteEquipment = async (req: Request, res: Response, next: NextFunction) => {
35   try {
36     const equipmentId = parseInt(req.params.equipmentId);
37     await deleteEquipmentService(equipmentId);
38     res.status(204).send({});
39   } catch (err) {
40     next(err);
41   }
42 }
43
44
45 export const updateEquipment = async (req: Request, res: Response, next: NextFunction) => {
46   try {
47     const equipmentId = parseInt(req.params.equipmentId);
48     const data = req.body;
49     const equipment = await updateEquipmentService(equipmentId, data);
50     res.status(200).send(equipment);
51   } catch (err) {
52     next(err);
53   }
54 }

```

Handwritten marks and signatures in the bottom right corner of the page.

TAV Web

## src/controllers/modelEquipment.controller.ts

```

1 import { NextFunction, Request, Response } from "express"
2 import { EquipmentModel } from "../entity/EquipmentModel"
3 import { createEquipmentModelService, deleteEquipmentModelService, getOneEquipmentModelService,
4       listEquipmentModelService, updateEquipmentModelService } from "../services/equipmentModel.service"
5 //Downloads/testedeploy-master 2/src/ => ../types/equipmentModel.types
6 services/call service.ts
7 export const createEquipmentModel = async (req: Request, res: Response, next: NextFunction) => {
8   try{
9     const data = req.body
10    const equipmentModel: IEquipmentModelCreate = await createEquipmentModelService(data)
11    res.status(201).json(equipmentModel)
12  }catch(err){
13    next(err)
14  }
15 }
16
17 export const listEquipmentModel = async (req: Request, res: Response, next: NextFunction) => {
18   try{
19     const equipmentModels: IEquipmentModel[] = await listEquipmentModelService()
20     res.status(200).send(equipmentModels)
21   }catch(err){
22     next(err)
23   }
24 }
25
26 export const getOneEquipmentModel = async (req: Request, res: Response, next: NextFunction) => {
27   try{
28     const modelId = parseInt(req.params.modelId)
29     const equipmentModel = await getOneEquipmentModelService(modelId)
30     res.status(200).send(equipmentModel)
31   }catch(err){
32     next(err)
33   }
34 }
35
36 export const updateEquipmentModel = async (req: Request, res: Response, next: NextFunction) => {
37   try{
38     const data = req.body
39     const modelId = parseInt(req.params.modelId)
40     const equipmentModelUpdated = await updateEquipmentModelService(modelId, data)
41     res.status(200).send(equipmentModelUpdated)
42   }catch(err){
43     next(err)
44   }
45 }
46
47 export const deleteEquipmentModel = async (req: Request, res: Response, next: NextFunction) => {
48   try{
49     const modelId = req.params.modelId
50     await deleteEquipmentModelService(modelId)
51     res.send(204).send("")
52   }catch(err){
53     next(err)
54   }
55 }

```

TAV Web

## src/controllers/requestsToChangedPassword.controller.ts

```

1  import { NextFunction, Request, Response } from "express";
2  import { changePasswordService, createRequestService,
3    getOneRequestService, listRequestService } from "../services/requestsToChangePassword.service";
4
5  export const changePassword = async(req: Request, res: Response, next: NextFunction) => {
6    try{
7      const data = req.body
8      const newRequest = await changePasswordService(data);
9      res.status(200).send(newRequest);
10   }catch(err){
11     next(err);
12   }
13 }
14
15 export const requestChange = async(req: Request, res: Response, next: NextFunction) => {
16   try{
17     const email = req.body.email
18
19     await createRequestService(email)
20
21     res.status(200).send({
22       message: "Solicitação enviada"
23     })
24
25   }catch(err){
26     next(err);
27   }
28 }
29
30 export const listRequests = async(req: Request, res: Response, next: NextFunction) => {
31   try{
32     const requestsToChangePassword = await listRequestService();
33     res.send(requestsToChangePassword);
34   }catch(err){
35     next(err);
36   }
37 }
38
39 export const getOneRequest = async(req: Request, res: Response, next: NextFunction) => {
40   try{
41     const requestId = parseInt(req.params.requestId);
42     const requestToChangePassword = await getOneRequestService(requestId);
43     res.send(requestToChangePassword);
44   }catch(err){
45     next(err);
46   }
47 }

```

9 X  
42 ✓

TAV Web

## src/controllers/user.controller.ts

```
1 import { NextFunction, Request, Response } from "express";
2 import { acceptOrRejectUsersService, createUserService,
3       listUsersNotActiveService, loginUserService } from "../services/user-service";
4
5 export const createUser = async (req: Request, res: Response, next: NextFunction) => {
6   try {
7     const data = req.body;
8     const userCreated = await createUserService(data);
9     res.status(201).send(userCreated);
10  } catch (err) {
11    next(err);
12  }
13 }
14
15 export const makeLogin = async (req: Request, res: Response, next: NextFunction) => {
16   try {
17     const data = req.body;
18     const token = await loginUserService(data);
19     res.status(200).send(token);
20   } catch (err) {
21     next(err);
22   }
23 }
24
25 export const listUsersNotActive = async (req: Request, res: Response, next: NextFunction) => {
26   try {
27     const users = await listUsersNotActiveService();
28     res.status(200).send(users);
29   } catch (err) {
30     next(err);
31   }
32 }
33
34 export const acceptOrRejectUsers = async (req: Request, res: Response, next: NextFunction) => {
35   try {
36     const userId = parseInt(req.params.userId);
37     const isActive = req.body.isActive;
38     await acceptOrRejectUsersService(userId, isActive);
39     res.status(200).send({});
40   } catch (err) {
41     next(err);
42   }
43 }
```

9 X  
43

TAV Web

## src/database/index.ts

```
1 import "reflect-metadata"
2 import { DataSource } from "typeorm"
3 import { Call } from "../entity/Call"
4 import { Equipment } from "../entity/Equipment"
5 import { EquipmentModel } from "../entity/EquipmentModel"
6 import { User } from "../entity/User"
7 import { RequestToChangePassword } from "../entity/ChangePassword"
8 import dotenv from 'dotenv';
9 dotenv.config();
10
11 export const AppDataSource = new DataSource({
12   type: "postgres",
13   host: process.env.HOST,
14   port: parseInt(process.env.PORT as string),
15   username: process.env.USERNAME,
16   password: process.env.PASSWORD,
17   database: process.env.DATABASE,
18   synchronize: true,
19   logging: false,
20   ssl: {
21     rejectUnauthorized: false
22   },
23   entities: [User, Call, Equipment, EquipmentModel, RequestToChangePassword],
24 })
25
```

9 X  
8 44  
25

TAV Web

## src/entity/Call.ts

```

1 import { Entity, PrimaryGeneratedColumn, Column, CreateDateColumn, ManyToOne } from "typeorm"
2 import { Equipment } from "../Equipment"
3 import { User } from "../User"
4
5 @Entity("calls")
6 export class Call {
7
8     @PrimaryGeneratedColumn()
9     id: number
10
11
12
13     @Column()
14     subject: string
15
16
17     @Column()
18     phoneUser: string
19
20
21     @CreateDateColumn()
22     date: Date
23
24
25     @Column()
26     description: string
27
28
29     @Column()
30     responsibleLocal: string
31
32
33     @Column()
34     status: string
35
36
37     @ManyToOne(() => User)
38     userRequester: User
39
40
41     @Column()
42     userRequesterName: string
43
44
45     @ManyToOne(() => User, { nullable: true })
46     userSolved: User
47
48
49     @Column({ nullable: true })
50     userSolvedName: string
51
52
53     @ManyToOne(() => Equipment, { onDelete: "CASCADE" })
54     equipment: Equipment
55
56
57     @Column()
58     equipmentCode: string
59
60
61     @Column()
62     equipmentPlace: string
63
64
65     @Column()
66     equipmentName: string
67
68 }

```

Handwritten notes and signatures in blue ink, including a large 'G' and some illegible scribbles.

TAV Web

## src/entity/ChangePassword.ts

```

1 import { Entity, PrimaryGeneratedColumn, Column, CreateDateColumn, ManyToOne } from "typeorm"
2 import { User } from "../User"
3
4 @Entity("requests_to_change_password")
5 export class RequestToChangePassword {
6
7     @PrimaryGeneratedColumn()
8     id: number
9
10    @CreateDateColumn()
11    date: Date
12
13    @ManyToOne(() => User)
14    user: User
15
16    @Column({default: "opening"})
17    status: string
18 }
19

```

## src/entity/Equipment.ts

```

1 import { Entity, PrimaryGeneratedColumn, Column, OneToMany, ManyToOne } from "typeorm"
2 import { EquipmentModel } from "../EquipmentModel"
3
4 @Entity("equipment")
5 export class Equipment {
6
7     @PrimaryGeneratedColumn()
8     id: number
9
10    @Column()
11    name: string
12
13    @Column()
14    code: string
15
16    @Column()
17    place: string
18
19    @ManyToOne(() => EquipmentModel, {onDelete: "CASCADE"})
20    equipmentModel: EquipmentModel
21 }
22

```

9  
46  
A

TAV Web

**src/entity/EquipmentModel.ts**

```

1  import { Entity, PrimaryGeneratedColumn, Column, ManyToOne, OneToMany } from "typeorm"
2  import { Equipment } from "../Equipment"
3
4  @Entity("equipment_models")
5  export class EquipmentModel {
6
7      @PrimaryGeneratedColumn()
8      id: number
9
10     @Column()
11     name: string
12
13     @Column()
14     description: string
15
16     @OneToMany(() => Equipment, equipment => equipment.id)
17     equipments!: Equipment[]
18 }
19

```

**src/entity/User.ts**

```

1  import { Entity, PrimaryGeneratedColumn, Column, OneToMany } from "typeorm"
2  import { Call } from "../Call"
3
4  @Entity("users")
5  export class User {
6
7      @PrimaryGeneratedColumn()
8      id: number
9
10     @Column()
11     firstName: string
12
13     @Column()
14     lastName: string
15
16     @Column()
17     phone: string
18
19     @Column()
20     email: string
21
22     @Column()
23     password: string
24
25     @Column({default: false})
26     isAdmin: boolean
27
28     @Column()
29     terms_of_user: boolean
30
31     @Column({default: false, nullable: true})
32     isActive: boolean
33
34     @OneToMany(() => Call, call => call.userSolved, {inverse: false})
35     solvedCalls: Call[]
36
37     @OneToMany(() => Call, call => call.userRequester, {inverse: false})
38     requestedCalls: Call[]
39
40 }
41

```

TAV Web

## src/middlewares/authorization.middleware.ts

```

1 import { NextFunction, Request, Response } from "express"
2 import { ErrorHandler } from "../utils/error"
3 import jwt from "jsonwebtoken"
4
5 export const authorization = (req: Request, res: Response, next: NextFunction) => {
6     const token = req.headers.authorization?.split(" ")[1]
7
8     const config = {
9         secret: "super",
10        expiresIn: "1h"
11    }
12
13    if (!token) {
14        next(new ErrorHandler(400, "token is missing"))
15    } else {
16        jwt.verify(token, config.secret, { err: any, decoded: any }) => {
17            if (err) {
18                next(new ErrorHandler(400, "The token does not belong to any account registered in the system"))
19            }
20
21            const email = decoded.email
22            req["userEmail"] = email
23            next()
24        }
25    }
26 }

```

## src/middlewares/call.validate.middleware.ts

```

1 import { ErrorHandler } from "../utils/error"
2 import { Response, Request, NextFunction } from "express"
3
4 export const validateCreateCall = async (req: Request, res: Response, next: NextFunction) => {
5
6     const keys = [Object, "description", "descriptionLocate"]
7     const userKeys = Object.keys(req.body)
8
9     const keysMissing = []
10    for (let i = 0; i < keys.length; i++) {
11        if (!keys.includes(keys[i]) || !req.body[keys[i]]) {
12            keysMissing.push(keys[i])
13        }
14    }
15
16    if (keysMissing.length > 0) {
17        next(new ErrorHandler(400, "keys missing: " + keysMissing.join(", ")))
18    }
19
20    const keysMore = []
21    for (let i = 0; i < userKeys.length; i++) {
22        if (keys.includes(userKeys[i]) || !req.body[userKeys[i]]) {
23            keysMore.push(userKeys[i])
24        }
25    }
26
27    if (keysMore.length > 0) {
28        next(new ErrorHandler(400, "keys more: " + keysMore.join(", ")))
29    }
30
31    const data = req.body
32    for (let i = 0; i < userKeys.length; i++) {
33        if (typeof data[userKeys[i]] != "string") {
34            next(new ErrorHandler(400, "The " + userKeys[i] + " value must be a string"))
35        }
36    }
37
38    next()
39 }

```

48



TAV Web

## src/middlewares/emailAlreadyExists.middleware.ts

```

1 import { NextFunction, Request, Response } from "express"
2 import { AppDataSource } from "../database"
3 import { User } from "../entity/user"
4 import { ErrorHandler } from "../utils/error"
5
6 export const emailAlreadyExists = async (req: Request, res: Response, next: NextFunction) => {
7
8     const email = req.body.email
9     const userRepository = AppDataSource.getRepository(User)
10
11     const user = await userRepository.findOne({
12         where: {
13             email: email
14         }
15     })
16
17     if(!user){
18         next(new ErrorHandler(400, "email already registered in the system"))
19     }
20
21     next()
22 }

```

## src/middlewares/equipment.validate.middleware.ts

```

1 import { ErrorHandler } from "../utils/error"
2 import { Response, Request, NextFunction } from "express"
3
4 export const validateCreateEquipment = async (req: Request, res: Response, next: NextFunction) => {
5
6     const keys = ["name", "user", "plate", "keychainid"]
7     const userKeys = Object.keys(req.body)
8
9     const keyMissing = []
10     for(let i = 0; i < keys.length; i++){
11         if(!userKeys.includes(keys[i])){
12             keyMissing.push(keys[i])
13         }
14     }
15
16     if(keyMissing.length > 0){
17         next(new ErrorHandler(400, "keys missing: " + keyMissing.join(", ")))
18     }
19
20     const keychain = []
21     for(let i = 0; i < userKeys.length; i++){
22         if(!keys.includes(userKeys[i])){
23             keychain.push(userKeys[i])
24         }
25     }
26
27     if(keychain.length > 0){
28         next(new ErrorHandler(400, "keys extra: " + keychain.join(", ")))
29     }
30
31     const data = req.body
32     for(let i = 0; i < userKeys.length; i++){
33         if(!userKeys[i] || !userKeys[i].trim()){
34             if(!req.body["equipmentname"]) {
35                 console.log(i)
36                 next(new ErrorHandler(400, "EquipmentName is not a number"))
37             }
38         }
39     }
40     if(!data[userKeys[i]] || !data[userKeys[i]].trim()){
41         next(new ErrorHandler(400, "The " + userKeys[i] + " equipment must be a string"))
42     }
43
44     next()
45 }

```

TAV Web

## src/middlewares/equipmentModel.validate.middleware.ts

```

1 import { ErrorHandler } from "../utils/errors"
2 import { Response, Request, NextFunction } from "express"
3
4 export const validateCreateEquipmentModel = async (req: Request, res: Response, next: NextFunction) => {
5
6     const keys = ["name", "description"]
7     const userKeys = Object.keys(req.body)
8
9     const keysMissing = []
10    for(let i = 0; i < keys.length; i++){
11        if(!userKeys.includes(keys[i]) == false){
12            keysMissing.push(keys[i])
13        }
14    }
15
16    if(keysMissing.length > 0){
17        next(new ErrorHandler(400, "Keys Missing: " + keysMissing.join(", ")))
18    }
19
20    const keysMore = []
21    for(let i = 0; i < userKeys.length; i++){
22        if(!keys.includes(userKeys[i]) == false){
23            keysMore.push(userKeys[i])
24        }
25    }
26
27    if(keysMore.length > 0){
28        next(new ErrorHandler(400, "Keys More: " + keysMore.join(", ")))
29    }
30
31    const data = req.body
32    for(let i = 0; i < userKeys.length; i++){
33        if(typeof data[userKeys[i]] != "string"){
34            next(new ErrorHandler(400, "The " + userKeys[i] + " column must be a string"))
35        }
36    }
37
38    next()
39 }
40

```

## src/middlewares/index.ts

```

1 import { validateCreate } from "../user.validate.middleware"
2 import { emailAlreadyExists } from "../emailAlreadyExists.middleware"
3 import { authorization } from "../authorization.middleware"
4 import { isAdmin } from "../isAdmin.middleware"
5 import { validateCreateEquipmentModel } from "../equipmentModel.validate.middleware"
6 import { isActive } from "../isActive.middleware"
7 import { validateCreateCall } from "../call.validate.middleware"
8 import { validateCreateEquipment } from "../equipment.validate.middleware"
9 import { userNotExists } from "../userNotExists.middleware"
10 import { userAlreadyActive } from "../userAlreadyActive.middleware"
11 import { requestExists } from "../requestExists.middleware"
12
13 export { validateCreate, emailAlreadyExists, authorization, isAdmin, validateCreateEquipmentModel,
14     isActive, validateCreateCall, validateCreateEquipment, userNotExists, userAlreadyActive, requestExists }

```

Handwritten signature and initials in blue ink.

TAV Web

**src/middlewares/isActive.middleware.ts**

```

1 import { NextFunction, Request, Response } from "express";
2 import { AppDataSource } from "../../../database";
3 import { User } from "../../../entity/User";
4 import { ErrorHandler } from "../../../utils/error";
5
6 export const isActive = async (req: Request, res: Response, next: NextFunction) => {
7     const userRepository = AppDataSource.getRepository(User);
8
9     const email = req.body.email;
10
11     const user = await userRepository.findOne({
12         where: {
13             email: email
14         }
15     });
16
17     if(!user){
18         next(new ErrorHandler(400, "O email informado não está associado a nenhuma conta cadastrada no sistema"));
19     }
20
21     if(user.isActive === false){
22         next(new ErrorHandler(400, "O usuário não está ativo"));
23     }
24
25     next();
26 }

```

**src/middlewares/isAdmin.middleware.ts**

```

1 import { NextFunction, Request, Response } from "express";
2 import { AppDataSource } from "../../../database";
3 import { User } from "../../../entity/User";
4 import { ErrorHandler } from "../../../utils/error";
5
6 export const isAdmin = async (req: Request, res: Response, next: NextFunction) => {
7
8     const userRepository = AppDataSource.getRepository(User);
9     const user = await userRepository.findOne({
10         where: {
11             email: req["userEmail"]
12         }
13     });
14
15     if(!user || user.isAdmin === false){
16         next(new ErrorHandler(401, "O usuário não tem permissão para utilizar esse recurso"));
17     }
18
19     next();
20 }

```

8 51 29

TAV Web

## src/middlewares/requestExists.middleware.ts

```

1 import { NextFunction, Request, Response } from "express"
2 import { AppDataSource } from "../database"
3 import { RequestToChangePassword } from "../entity/ChangePassword"
4 import { ErrorHandler } from "../utils/error"
5
6 export const requestExists = async (req: Request, res: Response, next: NextFunction) => {
7     const requestsToChangePasswordRepository = AppDataSource.getRepository(RequestToChangePassword)
8
9     const requestId = parseInt(req.params.requestId)
10
11     const request = await requestsToChangePasswordRepository.findOne({
12         where: {
13             id: requestId
14         }
15     })
16
17     if(!request){
18         next(new ErrorHandler(404, "O requestId informado não pertence a nenhum usuário cadastrado no sistema!"))
19     }
20
21     next()
22 }
23

```

## src/middlewares/user.validate.middleware.ts

```

1 import { ErrorHandler } from "../utils/error"
2 import { Response, Request, NextFunction } from "express"
3
4 export const validateCreate = async (req: Request, res: Response, next: NextFunction) => {
5
6     const keys = ["firstName", "email", "password", "lastName", "phone", "terms_of_use"]
7     const userKeys = Object.keys(req.body)
8
9     const keysMissing = []
10     for(let i = 0; i < keys.length; i++){
11         if(!userKeys.includes(keys[i])) {
12             keysMissing.push(keys[i])
13         }
14     }
15
16     if(keysMissing.length > 0){
17         next(new ErrorHandler(400, "Keys missing: " + keysMissing.join(", ")))
18     }
19
20     const keysMore = []
21     for(let i = 0; i < userKeys.length; i++){
22         if(!keys.includes(userKeys[i])) {
23             keysMore.push(userKeys[i])
24         }
25     }
26
27     if(keysMore.length > 0){
28         next(new ErrorHandler(400, "Keys more: " + keysMore.join(", ")))
29     }
30
31     const data = req.body
32     for(let i = 0; i < userKeys.length; i++){
33         if(userKeys[i] == "terms_of_use"){
34             if(!data[keys[i]] || !data[keys[i]] == "checked"){
35                 console.log(i)
36                 next(new ErrorHandler(400, "Terms of use must be checked!"))
37             }
38         }
39         if(!userKeys[i] || !data[userKeys[i]]) {
40             next(new ErrorHandler(400, "The " + userKeys[i] + " column must be a string!"))
41         }
42     }
43
44     next()
45 }
46

```

Handwritten signatures and initials in blue ink, including a large signature and the number '52'.



### src/middlewares/userAlreadyActive.middleware.ts

```

1 import { NextFunction, Request, Response } from "express";
2 import { AppDataSource } from "../database";
3 import { User } from "../entity/User";
4 import { ErrorHandler } from "../utils/error";
5
6 export const userAlreadyActive = async (req: Request, res: Response, next: NextFunction) => {
7     const userRepository = AppDataSource.getRepository(User);
8
9     const userId = parseInt(req.params.userId);
10
11     const user = await userRepository.findOne({
12         where: {
13             id: userId,
14             isActive: false
15         }
16     });
17
18     if(!user){
19         next(new ErrorHandler(400, "O usuário já está ativo no sistema"));
20     }
21
22     next();
23 }

```

### src/middlewares/userNotExists.middleware.ts

```

1 import { NextFunction, Request, Response } from "express";
2 import { AppDataSource } from "../database";
3 import { User } from "../entity/User";
4 import { ErrorHandler } from "../utils/error";
5
6 export const userNotExists = async (req: Request, res: Response, next: NextFunction) => {
7     const userRepository = AppDataSource.getRepository(User);
8
9     const userId = parseInt(req.params.userId);
10
11     const user = await userRepository.findOne({
12         where: {
13             id: userId
14         }
15     });
16
17     if(!user){
18         next(new ErrorHandler(400, "O id informado não pertence a nenhum usuário cadastrado no sistema"));
19     }
20
21     next();
22 }

```

9  
53  
8

TAV Web

## src/routes/call.route.ts

```

1 import { Router, Express } from "express";
2 import { createCall, createReport, deleteCall, getOneCall, listCall, updateCall } from "../controllers/call.controller";
3 import { authorization, isAdmin } from "../middlewares";
4
5 const route = Router();
6
7 export const callRoute = (app: Express) => {
8   route.post("/call", authorization, createCall);
9   route.get("/call", authorization, listCall);
10  route.delete("/call/:callId", authorization, deleteCall);
11  route.get("/call/:callId", authorization, getOneCall);
12  route.patch("/call/:callId", authorization, updateCall);
13  route.post("/report", authorization, isAdmin, createReport);
14  app.use("", route);
15 }

```

## src/routes/equipment.route.ts

```

1 import { Router, Express } from "express";
2 import {
3   createEquipment,
4   listEquipments,
5   getOneEquipment,
6   deleteEquipment,
7   updateEquipment,
8 } from "../controllers/equipment.controller";
9 import { authorization, isAdmin } from "../middlewares";
10
11 const route = Router();
12
13 export const equipmentRoute = (app: Express) => {
14   app.post("/equipments", authorization, isAdmin, createEquipment);
15   app.get("/equipments", authorization, listEquipments);
16   app.get("/equipments/:equipmentId", authorization, getOneEquipment);
17   app.delete("/equipments/:equipmentId", authorization, isAdmin, deleteEquipment);
18   app.patch("/equipments/:equipmentId", authorization, isAdmin, updateEquipment);
19   app.use("", route);
20 };
21

```

## src/routes/index.ts

```

1 import express, { Express } from "express";
2 import { userRoute } from "./user.route";
3 import { callRoute } from "./call.route";
4 import { modelEquipmentRoute } from "./modelEquipment.route";
5 import { equipmentRoute } from "./equipment.route";
6 import { passwordRoute } from "./requestsToChangePassword.route";
7
8 export const routesApp = (app: Express) => {
9   app.use(express.json());
10  userRoute(app);
11  callRoute(app);
12  modelEquipmentRoute(app);
13  equipmentRoute(app);
14  passwordRoute(app);
15 }

```

9  
 ✗  
 34  
 ✗



TAV Web

## src/routes/modelEquipment.route.ts

```

1 import { Router, Express } from "express"
2 import { createEquipmentModel, deleteEquipmentModel, getOneEquipmentModel,
3   listEquipmentModel, updateEquipmentModel } from "../controllers/modelEquipment.controller"
4 import { authorization, isAdmin } from "../middlewares"
5 import { validateCreateEquipmentModel } from "../middlewares/equipmentModel.validate.middleware"
6
7 const route = Router()
8
9 export const modelEquipmentRoute = (app: Express) => {
10   route.get("/models", authorization, listEquipmentModel)
11   route.get("/models/:modelId", authorization, getOneEquipmentModel)
12   route.post("/models", authorization, isAdmin, validateCreateEquipmentModel, createEquipmentModel)
13   route.patch("/models/:modelId", authorization, isAdmin, updateEquipmentModel)
14   route.delete("/models/:modelId", authorization, isAdmin, deleteEquipmentModel)
15   app.use("", route)
16 }

```

## src/routes/requestsToChangePassword.route.ts

```

1 import { Router, Express } from "express"
2 import { changePassword, getOneRequest, listRequests, requestChange } from "../controllers/requestsToChangePassword.controller"
3 import { authorization, isAdmin, requestExists } from "../middlewares"
4 const route = Router()
5
6 export const passwordRoute = (app: Express) => {
7   route.get("/password/requests", requestChange)
8   route.post("/password/change", authorization, isAdmin, changePassword)
9   route.get("/password/requests/:id", authorization, isAdmin, requestExists, getOneRequest)
10  route.get("/password", authorization, isAdmin, listRequests)
11  app.use("", route)
12 }

```

## src/routes/user.route.ts

```

1 import { Router, Express } from "express"
2 import { userObjectives, createUser, listUserObjective, updateUser } from "../controllers/userObjective.controller"
3 import { validateCreate, emailAlreadyExists, isActive, authorization, isAdmin, userNotExists, userAlreadyActive } from "../middlewares"
4 const route = Router()
5
6 export const userRoute = (app: Express) => {
7   route.post("/users", validateCreate, emailAlreadyExists, createUser)
8   route.post("/login", isActive, loginUser)
9   route.get("/users", authorization, isAdmin, listUserObjective)
10  route.get("/users/accept/active", authorization, isAdmin, userNotExists, userAlreadyActive, acceptUserObjective)
11  app.use("", route)
12 }

```

Handwritten marks and signatures in blue ink, including a large 'X' and some illegible scribbles.



TAV Web

## src/services/call.service.ts

```
1 import { Between } from "typeorm"
2 import { AppDataSource } from "../database"
3 import { Call } from "../entity/Call"
4 import { Equipment } from "../entity/Equipment"
5 import { User } from "../entity/User"
6 import { ICallCreate } from "../types/call.types"
7
8 // TODO: COLOCAR OS DADOS DO USUARIO VINDO DO TOKEN
9 export const createCallService = async (data: ICallCreate): Promise<Call> => {
10   const callRepository = AppDataSource.getRepository(Call)
11   const equipmentRepository = AppDataSource.getRepository(Equipment)
12   const userRepository = AppDataSource.getRepository(User)
13
14   const userRequest = await userRepository.findOne({
15     where: {
16       id: data.userRequestId
17     }
18   })
19
20   const equipment = await equipmentRepository.findOne({
21     where: {
22       id: data.equipmentId
23     }
24   })
25
26   const call = callRepository.create({
27     description: data.description,
28     responsibleLocale: data.responsibleLocale,
29     status: "opening",
30     equipment: equipment,
31     equipmentCode: equipment.code,
32     equipmentName: equipment.name,
33     equipmentPlace: equipment.place,
34     subject: data.subject,
35     userRequester: userRequest,
36     userRequesterName: userRequest.firstName + " " + userRequest.lastName,
37     phoneUser: data.phoneUser
38   })
39
40   await callRepository.save(call)
41
42   return call
43 }
```

Handwritten signature and date: 10/05/23

TAV Web

```
90
91     const secretaryCalls = await callRepository.find({
92         where: {
93             date: Between(start, end),
94             equipmentPlace: "Secretaria"
95         }
96     })
97
98     const developmentCalls = await callRepository.find({
99         where: {
100             date: Between(start, end),
101             equipmentPlace: "Desenvolvimento"
102         }
103     })
104
105     const attendanceCalls = await callRepository.find({
106         where: {
107             date: Between(start, end),
108             equipmentPlace: "Atendimento"
109         }
110     })
111
112     const calls = {
113         office: officeCalls,
114         secretary: secretaryCalls,
115         development: developmentCalls,
116         attendance: attendanceCalls
117     }
118
119     return calls
120
121 }
```

Handwritten notes in blue ink, including a large 'G' and some illegible scribbles.

```
44
45 export const getOneCallService = (callId: number): Promise<Call> => {
46   const callRepository = AppDataSource.getRepository(Call)
47
48   const call = callRepository.findOne({
49     where: {
50       id: callId
51     }
52   })
53
54   return call
55 }
56
57 export const listCallsService = (): Promise<Call[]> => {
58   const callRepository = AppDataSource.getRepository(Call)
59   const calls = callRepository.find()
60   return calls
61 }
62
63 export const deleteCallService = (callId: number) => {}
64
65 export const updateCallService = async (callId: number) => {
66   const callRepository = AppDataSource.getRepository(Call)
67   const call = await callRepository.findOne({
68     where: {
69       id: callId
70     }
71   })
72
73   call.status = "closed"
74
75   await callRepository.save(call)
76 }
77
78 export const createReportService = async (dateStart: Date, dateEnd: Date) => {
79   const callRepository = AppDataSource.getRepository(Call)
80
81   const end = new Date(dateEnd)
82   const start = new Date(dateStart)
83
84   const officeCalls = await callRepository.find({
85     where: {
86       date: Between(start, end),
87       equipmentPlace: "Escritório"
88     }
89   })
```

Handwritten notes in blue ink, including a large number '9' and some illegible scribbles.

## src/services/equipment.service.ts

```

1 import { AppDataSource } from "typeorm"
2 import { Equipment } from "src/entity/Equipment"
3 import { EquipmentModel } from "src/entity/EquipmentModel"
4 import { IEquipmentData } from "src/types/equipment.types"
5
6 export const listEquipmentsService = async () => Promise<Equipment[]> => {
7   const equipmentRepository = AppDataSource.getRepository(Equipment)
8
9   const equipments = await equipmentRepository.find()
10
11   return equipments
12 }
13
14 export const getOneEquipmentService = async (equipmentId: number): Promise<Equipment> => {
15   const equipmentRepository = AppDataSource.getRepository(Equipment)
16
17   const equipment = await equipmentRepository.findOne({
18     where: {
19       id: equipmentId
20     }
21   })
22
23   return equipment
24 }
25
26 export const deleteEquipmentService = async (equipmentId: number) => {
27   const equipmentRepository = AppDataSource.getRepository(Equipment)
28   await equipmentRepository.delete(equipmentId)
29 }
30
31 export const updateEquipmentService = async (equipmentId: number, data: IEquipmentData): Promise<Equipment> => {
32   const equipmentRepository = AppDataSource.getRepository(Equipment)
33
34   await equipmentRepository.update(equipmentId, data)
35
36   const equipment = equipmentRepository.findOne({
37     where: {
38       id: equipmentId
39     }
40   })
41
42   return equipment
43 }
44
45 }
46
47 export const createEquipmentService = async (data: IEquipmentData) => {
48   const equipmentModelRepository = AppDataSource.getRepository(EquipmentModel)
49
50   const equipmentModel = await equipmentModelRepository.findOne({
51     where: {
52       id: data.equipmentModelId
53     }
54   })
55
56   const equipmentData = {
57     code: data.code,
58     name: data.name,
59     place: data.place,
60     equipmentModel: equipmentModel
61   }
62
63   const equipmentRepository = AppDataSource.getRepository(Equipment)
64   const equipment = equipmentRepository.create(equipmentData)
65   await equipmentRepository.save(equipment)
66
67   return equipment
68 }
69

```

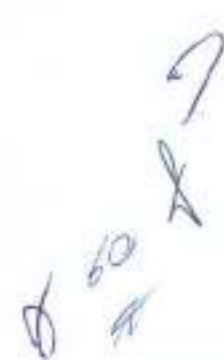
59

## src/services/equipmentModel.service.ts

```

1 import { AppDataSource } from "../../database"
2 import { EquipmentModel } from "../../entity/EquipmentModel"
3 import { IEquipmentModelData, IEquipmentModelCreate } from "../../types/equipmentModel.types"
4
5 export const createEquipmentModelService = async (data: IEquipmentModelData): Promise<EquipmentModel> => {
6     const equipmentModelRepository = AppDataSource.getRepository(EquipmentModel)
7
8     const equipmentModel = equipmentModelRepository.create(data)
9     await equipmentModelRepository.save(equipmentModel)
10
11     return equipmentModel
12 }
13
14 export const deleteEquipmentModelService = async (modelId: string) => {
15     const equipmentModelRepository = AppDataSource.getRepository(EquipmentModel)
16     await equipmentModelRepository.delete(modelId)
17 }
18
19 export const listEquipmentModelService = async (): Promise<EquipmentModel[]> => {
20     const equipmentModelRepository = AppDataSource.getRepository(EquipmentModel)
21
22     const equipmentModels = await equipmentModelRepository.find()
23
24     return equipmentModels
25 }
26
27 export const getOneEquipmentModelService = async (modelId: number): Promise<EquipmentModel> => {
28     const equipmentModelRepository = AppDataSource.getRepository(EquipmentModel)
29
30     const equipmentModel = await equipmentModelRepository.findOne({
31         where: {
32             id: modelId
33         }
34     })
35
36     return equipmentModel
37 }
38
39 export const updateEquipmentModelService = async (modelId: number, data: IEquipmentModelData): Promise<EquipmentModel> => {
40     const equipmentModelRepository = AppDataSource.getRepository(EquipmentModel)
41
42     const equipmentModels = await equipmentModelRepository.findOne({
43         where: {
44             id: modelId
45         }
46     })
47
48     const keys = Object.keys(data)
49     for (let i = 0; i < Object.keys(data).length; i++)
50         equipmentModels[keys[i]] = data[keys[i]]
51
52     await equipmentModelRepository.save(equipmentModels)
53
54     return equipmentModels
55 }
56 }
57 }

```



## src/services/requestsToChangePassword.service.ts

```

1 import { AppDataSource } from "../../database"
2 import { RequestToChangePassword } from "../../entity/ChangePassword"
3 import { User } from "../../entity/User"
4 import { IPasswordChanged, IRequestCreate } from "../../types/requestsToChangePassword.types"
5 import * as bcrypt from "bcryptjs"
6 import { ErrorHandler } from "../../utils/error"
7
8 export const createRequestService = async (email: string) => {
9     const userRepository = AppDataSource.getRepository(User)
10    const requestsToChangePasswordRepository = AppDataSource.getRepository(RequestToChangePassword)
11
12    const user = await userRepository.findOne({
13        where: {
14            email: email
15        }
16    })
17
18    const data: IRequestCreate = {
19        user: user
20    }
21
22    const requestToChangePassword = requestsToChangePasswordRepository.create(data)
23
24    await requestsToChangePasswordRepository.save(requestToChangePassword)
25
26    return requestToChangePassword
27 }
28
29
30 export const changePasswordService = async (data: IPasswordChanged): Promise<RequestToChangePassword> => {
31    const requestsToChangePasswordRepository = AppDataSource.getRepository(RequestToChangePassword)
32
33    const userRepository = AppDataSource.getRepository(User)
34
35    const user = await userRepository.findOne({
36        where: {
37            email: data.email
38        }
39    })
40
41    if(!user) {
42        throw new ErrorHandler(400, "Usuário inexistente")
43    }
44
45    const requestToChangePassword = await requestsToChangePasswordRepository.findOne({
46        where: {
47            user: user,
48            status: "pending"
49        }
50    })

```

Handwritten signatures and initials in blue ink, including a large stylized 'G' and other illegible marks.



```
31
32   if (!requestToChangePassword) {
33     throw new Error(translate(448, "NÃO existe registros abertos para esse usuário"));
34   }
35
36   const requestUpdate = {
37     status: "closed"
38   }
39
40   const newPassword = await bcrypt.hash(data.password, 10);
41
42   const userUpdate = {
43     password: newPassword
44   }
45
46   await userRepository.update(user, userUpdate);
47
48   await requestsToChangePasswordRepository.update(requestToChangePassword.id, requestUpdate);
49
50   const newRequest = await requestsToChangePasswordRepository.findOne({
51     where: {
52       id: requestToChangePassword.id
53     }
54   });
55
56   return newRequest;
57 }
58
59 export const listRequestService = async (): Promise<RequestToChangePassword[]> => {
60   const requestsToChangePasswordRepository = AppDataSource.getRepository(RequestToChangePassword);
61   const requests = await requestsToChangePasswordRepository.find();
62   return requests;
63 }
64
65 export const getOneRequestService = async (requestId: number): Promise<RequestToChangePassword> => {
66   const requestsToChangePasswordRepository = AppDataSource.getRepository(RequestToChangePassword);
67   const request = await requestsToChangePasswordRepository.findOne({
68     where: {
69       id: requestId
70     }
71   });
72   return request;
73 }
74 }
```

682

## src/services/user.service.ts

```
1 import { AppDataSource } from "../database";
2 import { User } from "../entity/User";
3 import { IUserCreate, ILoginData } from "../types/user.types";
4 import * as bcrypt from "bcryptjs";
5 import { ErrorHandler } from "../utils/error";
6 import jwt from "jsonwebtoken";
7
8 export const createUserService = async (data: IUserCreate) => {
9   const passwordHash = await bcrypt.hash(data.password, 10);
10  data.password = passwordHash;
11
12  const userRepository = AppDataSource.getRepository(User);
13  const user = userRepository.create(data);
14  await userRepository.save(user);
15
16  const { password, ...userWithoutPassword } = user;
17
18  return userWithoutPassword;
19 };
20
21 export const loginUserService = async (data: ILoginData) => {
22  const { email, password } = data;
23  const userRepository = AppDataSource.getRepository(User);
24  const user = await userRepository.findOne({
25    where: {
26      email: email,
27    },
28  });
29
30  const match = await bcrypt.compare(password, user.password);
31
32  if (!match) {
33    throw new ErrorHandler(422, "Senha inválida");
34  }
35
36  const config = {
37    secret: "igor",
38    expiresIn: "1h",
39  };
```

TAV Web

```

40
41 const token = jwt.sign(
42   { email: user.email, admin: user.isAdmin, firstName: user.firstName,
43     id: user.id, phone: user.phone, lastName: user.lastName },
44   config.secret,
45   { expiresIn: config.expiresIn }
46 );
47
48 return { accessToken: token };
49 };
50
51 export const listUsersNotActiveService = async (): Promise<User[]> => {
52   const userRepository = AppDataSource.getRepository(User);
53   const users = await userRepository.find({
54     where: {
55       isActive: false,
56     },
57   });
58
59   const usersShow = [];
60   for (let i = 0; i < users.length; i++) {
61     const { password, ...userShow } = users[i];
62     usersShow.push(userShow);
63   }
64
65   return usersShow;
66 };
67
68 export const acceptOrRejectUsersService = async (
69   userId: number,
70   isActive: boolean
71 ) => {
72   const userRepository = AppDataSource.getRepository(User);
73
74   const user = await userRepository.findOne({
75     where: {
76       id: userId,
77     },
78   });
79
80   if (typeof isActive !== "boolean") {
81     throw new ErrorHandler(400, "O campo 'is_active' é um valor booleano");
82   }
83
84   if (isActive) {
85     user.isActive = true;
86     userRepository.save(user);
87   }
88
89   await userRepository.delete(user);
90 };

```

## src/types/call.types.ts

```
1 export interface ICallCreate {
2     subject: string
3     description: string
4     responsibleLocale: string
5     userRequestId: number
6     equipmentId: number
7     phoneUser: string
8 }
```

## src/types/equipment.types.ts

```
1 export interface IEquipmentData {
2     name: string
3     code: string
4     place: string
5     equipmentModelId: number
6 }
7
8 export interface IEquipmentCreate {
9     name: string
10    code: string
11    place: string
12    equipmentModelId: number
13    id: number
14 }
15 |
```

## src/types/equipmentModel.types.ts

```
1  export interface IEquipmentModelData {
2      name: string,
3      description: string
4  }
5
6  export interface IEquipmentModelCreate {
7      id: number,
8      name: string,
9      description: string
10 }
11 |
```

## src/types/requestsToChangePassword.types.ts

```
1  import { User } from "../entity/User"
2
3  export interface IRequestCreate {
4      user: User
5  }
6
7  export interface IPasswordChanged {
8      email: string
9      password: string
10 }
```

9  
2  
66  
v

TAV Web

## src/types/user.types.ts

```
1  export interface IUserCreate {
2      firstName: string,
3      lastName: string,
4      phone: string,
5      email: string,
6      password: string,
7      terms_of_use: boolean
8  }
9
10 export interface ILoginData {
11     email: string
12     password: string
13 }
```

## src/utlis/error.ts

```
1  import { Response } from "express"
2
3  export class ErrorHandler extends Error{
4      statusCode: number;
5      message: string;
6
7      constructor(statusCode: number, message: string){
8          super()
9          this.statusCode = statusCode;
10         this.message = message;
11     }
12
13 }
14
15 export const handleError = (err: ErrorHandler, res: Response) => {
16     const {message, statusCode} = err;
17     res.status(err.statusCode).send(
18         {
19             status: "error",
20             statusCode,
21             message
22         }
23     )
24 }
```

6

8

8/27

## src/app.ts

```
1 import express from "express";
2 import { AppDataSource } from "../database";
3 import { routesApp } from "../routes";
4 import { handleError } from "../src/utis/error";
5 import cors from "cors"
6
7
8 const app = express();
9
10 app.use(express.json());
11 app.use(express.static("public"));
12
13 app.use((err: any, req: any, res: any, next: any) => {
14   handleError(err, res);
15 });
16
17 app.use(cors())
18
19 routesApp(app);
20
21 AppDataSource.initialize();
22
23 export default app;
24
```

## src/server.ts

```
1 import app from '../src/app';
2 import dotenv from 'dotenv';
3 dotenv.config();
4
5 const PORT = process.env.PORT || 5000;
6 app.listen(PORT, () => {
7   console.log('Runnin at http://localhost -> 5000');
8 });
9
```

TAV Web

**.gitignore**

```

1  .idea/
2  .vscode/
3  node_modules/
4  build/
5  tmp/
6  temp/
7  .env

```

**package.json**

```

1  {
2    "name": "help",
3    "version": "0.0.1",
4    "description": "Awesome project developed with TypeScript.",
5    "type": "commonjs",
6    "devDependencies": {
7      "@types/bcryptjs": "~2.4.2",
8      "@types/cors": "~2.8.12",
9      "@types/express": "~4.17.13",
10     "@types/jwtwebtoken": "~8.5.8",
11     "@types/node": "~16.11.10",
12     "@types/nodemailer": "~6.4.4",
13     "cors": "~2.8.5",
14     "ts-node": "10.7.0",
15     "ts-node-dev": "~2.0.0",
16     "typescript": "4.5.2"
17   },
18   "dependencies": {
19     "bcryptjs": "~2.4.3",
20     "dotenv": "~16.0.3",
21     "express": "~4.18.1",
22     "jsonwebtoken": "~8.5.1",
23     "jwt-decode": "~3.1.2",
24     "nodemailer": "~6.7.7",
25     "pg": "~8.4.0",
26     "reflect-metadata": "~0.1.13",
27     "typeorm": "0.3.7"
28   },
29   "scripts": {
30     "start": "ts-node-dev src/server.ts",
31     "typeorm": "typeorm-ts-node-commonjs"
32   }
33 }
34

```

9  
 8  
 69

## README.md

```
1
2 Steps to run this project:
3
4 1. Run 'npm i' command
5 2. Setup database settings inside 'data-source.ts' file
6 3. Run 'npm start' command
7 |
```

## tsconfig.json

```
1 {
2   "compilerOptions": {
3     "lib": [
4       "es5",
5       "es6"
6     ],
7     "target": "es5",
8     "module": "commonjs",
9     "moduleResolution": "node",
10    "outDir": "./build",
11    "emitDecoratorMetadata": true,
12    "experimentalDecorators": true,
13    "sourceMap": true,
14    "esModuleInterop": true,
15  }
16 }
```

Handwritten signature and date: 03/09/2023



# Front End

```

> node_modules
> public
> src
  > components
    > Footer
      > index.tsx
    > HeaderAdmin
      > index.css
      > index.tsx
    > HeaderUser
      > index.css
      > index.tsx
  > images
  > pages
    > createEquipment
      > index.tsx
    > createModelEquipment
      > index.tsx
    > createModel
      > index.tsx
    > detailPageEquipment
      > index.css
      > index.tsx
    > detailPageModelEquipment
      > index.css
      > index.tsx
    > editEquipment
      > index.tsx
    > editModelEquipment
      > index.tsx
    > landingPageAdmin
      > index.tsx
    > LandingPageEquipment
      > index.tsx
    > landingPageModelEquipment
      > index.tsx
    > landingPageTicket
      > index.tsx
  > sign
    > index.tsx
    > styles.ts
  
```




TAV Web



As pastas (Build, node\_modules e public) e o arquivo (package-lock.json) não foram adicionadas nesse arquivo por serem pastas criadas automaticamente ao criarmos um programa React Native, somadas elas apresentam milhares de linhas que fazem o algoritmo funcionar e as mesmas não necessitam estar neste arquivo. Os arquivos (yarn.lock) e (yarn-error.log) não foi adicionado por um arquivo gerado automaticamente ao instalarmos yarn no projeto.

*Handwritten signature and date:*  
 07/21/23

TAV Web

## src/components/Footer/index.tsx

```
1
2 export const Footer = () => {
3   return (
4     <div className="text-center copy-right">
5       &copy; Helpdesk todos os direitos reservados – 2022.
6     </div>
7   );
8 };
```

## src/components/HeaderAdmin/index.css

```
1 .div-container{
2   display: flex;
3   flex-direction: column;
4 }
5
6 .nav-container {
7   display: flex;
8   flex-direction: column;
9 }
10
11
12
13
14 @media (min-width:600px){
15   .nav-tags {
16     display: flex;
17     flex-direction: row;
18     padding-left: 180px;
19   }
20
21   .nav-container {
22     display: flex;
23     flex-direction: row;
24     justify-content: space-between;
25     width: 100%;
26     padding-right: 200px;
27   }
28
29   .div-container{
30     display: flex;
31     width: 100%;
32   }
33 }
```

## src/components/HeaderAdmin/index.tsx

```
1 import { Navbar, Nav } from "react-bootstrap";
2 import './index.css'
3
4 export const HeaderAdmin = () => {
5   const handleLogout = () => {
6     localStorage.removeItem("Access");
7     localStorage.removeItem("AccessEmail");
8   };
9   return (
10    <Navbar collapseOnSelect style={{backgroundColor: "black"}} variant="dark" expand="md">
11      <Navbar.Brand/></Navbar.Brand>
12      <Navbar.Toggle aria-controls="basic-navbar-nav" />
13      <Navbar.Collapse id="basic-navbar-nav">
14        <div className="div-container">
15          <Nav className="nav-container" activeKey={window.location.pathname}>
16            <div className="nav-tops">
17              <Nav.Link href="/admin">Chamados</Nav.Link>
18              <Nav.Link href="/modelsEquipment">Modelos de Equipamentos</Nav.Link>
19              <Nav.Link href="/equipments">Equipamentos</Nav.Link>
20              <Nav.Link href="/users">Usuários</Nav.Link>
21              <Nav.Link href="/report">Relatorio</Nav.Link>
22            </div>
23
24            <Nav.Link href="/" onClick={handleLogout}>
25              Logout
26            </Nav.Link>
27          </Nav>
28        </div>
29      </Navbar.Collapse>
30    </Navbar>
31  );
32 };
33
```

6  
X  
79  
BT

TAV Web

## src/components/HeaderUser/index.css

```
1  .div-container{
2    display: flex;
3    flex-direction: column;
4  }
5
6  .nav-container {
7    width: 100%;
8    padding-right: 200px;
9  }
10
11
12  @media (min-width:600px){
13
14    .nav-container {
15      width: 100%;
16      padding-right: 200px;
17    }
18
19    .div-container{
20      display: flex;
21      flex-direction:row;
22      width: 100%;
23      justify-content: space-between;
24    }
25
26
27
28
29  }
```

9  
X  
45  
X



## src/components/HeaderUser/index.tsx

```
1 import { Navbar, Nav } from "react-bootstrap";
2 import "../index.css";
3
4 export const HeaderUser = () => {
5   const handleLogout = () => {
6     localStorage.removeItem("Access");
7     localStorage.removeItem("AccessEmail");
8   };
9   const accessEmail = localStorage.getItem("AccessEmail");
10
11   return (
12     <Navbar
13       collapseOnSelect
14       style={{ backgroundColor: "black" }}
15       variant="dark"
16       expand="md"
17     >
18       <Navbar.Brand /> </Navbar.Brand>
19       <Navbar.Toggle aria-controls="basic-navbar-nav" />
20       <Navbar.Collapse id="basic-navbar-nav">
21         <div className="nav-container" activeKey={window.location.pathname}>
22
23           <div className="div-container">
24             <Nav.Link style={{paddingLeft: "100px"}} href="/tickets">Chamados</Nav.Link>
25             <Nav.Link href="/" onClick={handleLogout}>
26               Logout
27             </Nav.Link>
28           </div>
29
30         </Nav>
31       </Navbar.Collapse>
32     </Navbar>
33   );
34 };
35
```

## src/pages/createEquipment/index.tsx

```

1 //import { Axios } from "axios";
2 import { useContext, useState } from "react";
3 import { Form, Button, Row, Col } from "react-bootstrap";
4 import { HeaderAdmin } from "../../../components/HeaderAdmin";
5 import { Footer } from "../../../components/Footer";
6 import { EquipmentContext } from "../../../providers/equipment";
7 import { EquipmentModelContext } from "../../../providers/equipmentModel";
8 import { useNavigate, redirect } from "react-router-dom";
9 import { AuthContext } from "../../../providers/user";
10
11 const CreateEquipment = () => {
12   const { equipments, createEquipment } = useContext(EquipmentContext);
13   const { equipmentModels } = useContext(EquipmentModelContext);
14
15   const { isLoggedIn } = useContext(AuthContext);
16
17   const [place, setPlace] = useState("");
18   const [equipmentModelId, setEquipmentModelId] = useState("");
19   const [code, setCode] = useState("");
20
21   const navigate = useNavigate();
22
23   const user = JSON.parse(localStorage.getItem("user") || "{}(admin: false, email: '')");
24
25   if (user.admin == false) {
26     if (user.email != "") {
27       navigate("/TICKETS");
28     } else {
29       navigate("/");
30     }
31     return <div>
32   }
33
34   const onSubmit = (e: any) => {
35     e.preventDefault();
36
37     const name = equipmentModels.filter((eq) => eq.id == parseInt(equipmentModelId))[0].name;
38
39     const data = {
40       name: name,
41       place: place,
42       equipmentModelId: parseInt(equipmentModelId),
43       code: code
44     };
45
46     navigate("/equipments", {
47       state: -1,
48     });
49
50     createEquipment(data);
51   };

```

Handwritten notes and signatures in the bottom right corner of the page, including a large stylized 'G' and other illegible marks.

```

52
53     return (
54         <div>
55             <header className="header mb-2">
56                 <headerAdmin />
57             </header>
58
59             <main className="main">
60                 <div className="form-box p-5">
61                     <form autoComplete="off">
62                         <h1 className="text-center U-100, fs-3, fw-normal">Equipamento</h1>
63                         <br />
64
65                         <Form.Group className="mb-3" as={Row}>
66                             <Form.Label>Modelo do Equipamento:</Form.Label>
67                             <Col sm={12}>
68                                 <Form.Control
69                                     as="select"
70                                     name="nameModel"
71                                     placeholder="Modelo"
72                                     value={equipmentModelId}
73                                     onChange={(e) => setEquipmentModelId(e.target.value)}
74                                 >
75                                     <option>Selecione o Modelo</option>
76                                     {equipmentModels.map(equipment) => (
77                                         return (
78                                             <option value={equipment.id}> {equipment.name}</option>
79                                         )
80                                     )}
81                                 </Form.Control>
82                             </Col>
83                         </Form.Group>
84
85                         <Form.Group className="mb-3" as={Row}>
86                             <Form.Label>Lugar Onde Esta Instalado:</Form.Label>
87                             <Col sm={12}>
88                                 <Form.Control
89                                     as="select"
90                                     name="place"
91                                     value={place}
92                                     onChange={(e) => setPlace(e.target.value)}
93                                 >
94                                     <option>Selecione o lugar</option>
95                                     <option value={"Atendimento"}>Atendimento</option>
96                                     <option value={"Secretaria"}>Secretaria</option>
97                                     <option value={"Desenvolvimento"}>Desenvolvimento</option>
98                                     <option value={"Escritório"}>Escritório</option>
99                                 </Form.Control>
100                             </Col>
101                         </Form.Group>

```

Handwritten notes and signatures in blue ink, including a large stylized signature and the numbers 78 and 87.

```
102
103     <Form.Group className="mb-3" as={Row}>
104       <Form.Label>Codigo do Equipamento:</Form.Label>
105       <Col sm={12}>
106         <Form.Control
107           type="text"
108           name="equipmentId"
109           placeholder="Código"
110           value={code}
111           onChange={(e) => setCode(e.target.value)}
112         />
113       </Col>
114     </Form.Group>
115
116     <Button variant="info" type="submit" onClick={onSubmit}>
117       Enviar
118     </Button>
119   </>
120 </Form>
121 </div>
122 </main>
123 <footer className="footer">
124   <Footer />
125 </footer>
126 </div>
127 };
128 };
129 export default CreatEquipment;
130
```

Handwritten notes and signatures in the bottom right corner, including a large stylized signature and some illegible scribbles.

## src/pages/createModelEquipment/index.tsx

```
1 import { useContext, useState } from "react";
2 import { Form, Button, Row, Col } from "react-bootstrap";
3 import { useNavigate } from "react-router-dom";
4 import { Footer } from "../../../components/Footer";
5 import { HeaderAdmin } from "../../../components/HeaderAdmin";
6 import { EquipmentModelContext } from "../../../providers/equipmentModel";
7
8 const CreateModelEquipment = () => {
9   const { equipmentModels, createEquipmentModel } = useContext(
10     EquipmentModelContext
11   );
12
13   const [name, setName] = useState("");
14   const [description, setDescription] = useState("");
15
16   const onSubmit = (e: any) => {
17     e.preventDefault();
18
19     const data = {
20       name: name,
21       description: description,
22     };
23
24     setName("");
25     setDescription("");
26
27     createEquipmentModel(data);
28   };
29
30   const navigate = useNavigate();
31
32   const user = JSON.parse(localStorage.getItem("user") || "{}");
33
34   if (user.admin === false) {
35     if (user.email != "") {
36       navigate("/tickets");
37     } else {
38       navigate("/");
39     }
40   }
41   return <></>
42 }
```

9  
30  
4

```

42
43 return (
44   <div>
45     <header className="header mb-2">
46       <HeaderAdmin />
47     </header>
48
49     <main className="main">
50       <div className="form-box p-5">
51         <Form autoComplete="off">
52           <h1 className="text-center w-100, fs-3 fw-normal">
53             Modelo do Equipamento
54           </h1>
55           <br />
56
57           <Form.Group className="mb-1" as={Row}>
58             <Form.Label>Modelo do Equipamento:</Form.Label>
59             <Col sm={12}>
60               <Form.Control
61                 type="text"
62                 name="equipmentModel"
63                 placeholder="Modelo"
64                 value={name}
65                 onChange={(e) => setName(e.target.value)}
66               />
67             </Col>
68           </Form.Group>
69
70           <Form.Group className="mb-3" as={Row}>
71             <Form.Label column sm={3}>
72               {" "}
73             <strong>Descrição:</strong>
74             </Form.Label>
75             <Col sm={12}>
76               <Form.Control
77                 as="textarea"
78                 name="description"
79                 placeholder="Descrição"
80                 value={description}
81                 onChange={(e) => setDescription(e.target.value)}
82                 rows={5}
83               />
84             </Col>
85           </Form.Group>
86
87           <Button variant="info" type="submit" onClick={onSubmit}>
88             Enviar
89           </Button>
90           <br />
91         </Form>
92       </div>
93     </main>

```


  
 8/1

TAV Web

```
94     <footer className="footer">
95         <Footer />
96     </footer>
97 </div>
98 );
99 };
100
101 export default CreateModelEquipment;
102
```

Handwritten notes in blue ink, including a large '9' at the top, a checkmark, and the number '32'.

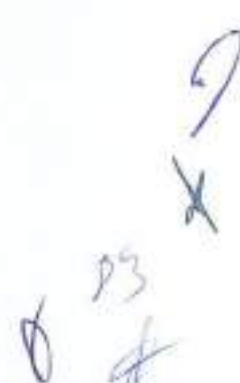


## src/pages/createTicket/index.tsx

```

1  import React, { Component, useContext, useEffect, useState } from 'react';
2  import { Form, Button, Row, Col } from 'react-bootstrap';
3  import { useNavigate } from 'react-router-dom';
4  import { Footer } from '../../../components/Footer';
5  import { HeaderUser } from '../../../components/HeaderUser';
6  import { CallsContext } from '../../../providers/call';
7  import { EquipmentContext } from '../../../providers/equipment';
8  import { EquipmentModelContext } from '../../../providers/equipmentModel';
9  import { AuthContext } from '../../../providers/user';
10
11  interface ICallCreate {
12    subject: string;
13    description: string;
14    responsibleLocale: string;
15    equipmentId: number;
16    userRequestId: number;
17    phoneUsers: string;
18  }
19
20  interface IUserLogged {
21    firstName: string;
22    email: string;
23    id: number;
24    phone: string;
25    lastName: string;
26  }
27
28  const CreateTicket = () => {
29    const { calls, createCall } = useContext(CallsContext);
30    const { equipments, listEquipments } = useContext(EquipmentContext);
31    const { equipmentModels } = useContext(EquipmentModelContext);
32
33    const [description, setDescription] = useState('');
34    const [responsibleLocale, setResponsibleLocale] = useState('');
35    const [subject, setSubject] = useState('');
36    const [equipment, setEquipment] = useState('');
37
38    const [nameModel, setNameModel] = useState('');
39    const [currentDate] = useState(new Date().toLocaleDateString('en-US'));
40    const [place, setPlace] = useState('');
41
42    const userLogged = JSON.parse(localStorage.getItem('user') || '{}') as IUserLogged;
43
44    const [nameUser] = useState(userLogged.firstName + " " + userLogged.lastName);
45    const [phone] = useState(userLogged.phone);
46
47    useEffect(() => {
48      listEquipments();
49    }, []);

```


  
 0 93

```
50
51 const navigate = useNavigate();
52
53 const user = JSON.parse(
54   localStorage.getItem("user") || '{"admin: true, email: ""}'
55 );
56
57 if (user.admin == true) {
58   if (user.email != "") {
59     navigate("/equipments");
60   } else {
61     navigate("/");
62   }
63   return <</>;
64 }
65
66 const onSubmit = (e: any) => {
67   e.preventDefault();
68
69   const equipmentProduct = equipments.filter(
70     (eq) => eq.id == parseInt(equipment)
71   )[0];
72
73   const user: IUserLogged = JSON.parse(localStorage.getItem("user") || "{}");
74
75   const data: ICallCreate = {
76     description: description,
77     equipmentId: equipmentProduct.id,
78     responsibleLocale: responsibleLocale,
79     subject: subject,
80     userRequestId: user.id,
81     phoneUser: user.phone
82   };
83
84   setDescription("");
85   setSubject("");
86   setResponsibleLocale("");
87   setEquipment("");
88
89   createCall(data);
90 };
91
92 return (
93   <div>
94     <header className="header mb-2">
95       <HeaderUser />
96     </header>
```

6  
x  
84  
a

TAV Web

```

97
98 <main className="main">
99 <div className="form-box p-5">
100 <form autoComplete="off">
101 <h1 className="text-center w-100, fs-3 fw-normal">Chamado</h1>
102 </form>
103 <div>
104 <Form.Group className="mb-3" as={Row}>
105 <Form.Label coluna sm={3}>
106 Assunto:
107 </Form.Label>
108 <Col sm={12}>
109 <Form.Control
110 type="text"
111 name="title"
112 placeholder="Assunto"
113 value={subject}
114 onChange={e => setSubject(e.target.value)}
115 />
116 </Col>
117 </Form.Group>
118
119 <Form.Group className="mb-3" as={Row}>
120 <Form.Label> Equipamento:</Form.Label>
121 <Col sm={12}>
122 <Form.Control
123 as="select"
124 name="equipmentId"
125 //placeholder="Codigo do Equipamento"
126 value={equipment}
127 onChange={e => {
128 const eq = equipments.filter(x => x.id == parseInt(e.target.value)) [0];
129 setEquipment(e.target.value);
130 setPlace(eq.place);
131 setNameModel(eq.name);
132 }
133 }
134 </Form.Control>
135 </Col>
136 <Form.Group>
137 <Form.Label> Equipamento:</Form.Label>
138 <Form.Control
139 as="select"
140 name="equipmentId"
141 value={equipment}
142 onChange={e => {
143 const eq = equipments.filter(x => x.id == parseInt(e.target.value)) [0];
144 setEquipment(e.target.value);
145 setPlace(eq.place);
146 setNameModel(eq.name);
147 }
148 }
149 </Form.Control>
150 </Form.Group>

```

85  
9  
X

TAV Web

```

145
146 <Form.Group className="mb-3" as={Row}>
147   <Form.Label column sm={3}>
148     Equipamento:
149   </Form.Label>
150   <Col sm={12}>
151     <Form.Control
152       type="text"
153       name="title"
154       placeholder="Assunto"
155       value={nameModel}
156       disabled
157     />
158   </Col>
159 </Form.Group>
160
161 <Form.Group className="mb-3" as={Row}>
162 <Form.Label> Lugar de instalação:</Form.Label>
163   <Col sm={12}>
164     <Form.Control
165       type="text"
166       name="title"
167       placeholder="Assunto"
168       value={place}
169       disabled
170     />
171   </Col>
172 </Form.Group>
173
174 <Form.Group className="mb-3" as={Row}>
175 <Form.Label>Data:</Form.Label>
176   <Col sm={12}>
177     <Form.Control
178       type="text"
179       name="title"
180       placeholder="Assunto"
181       value={currentDate}
182       disabled
183     />
184   </Col>
185 </Form.Group>
186
187 <Form.Group className="mb-3" as={Row}>
188   <Form.Label column> Ocorrência Observada:</Form.Label>
189   <Col sm={12}>
190     <Form.Control
191       as="textarea"
192       name="description"
193       value={description}
194       onChange={(e) => setDescription(e.target.value)}
195     />
196   </Col>
197 </Form.Group>

```



TAV Web

```

196         />
197     </Col>
198 </Form.Group>
199 </div>
200
201 <div>
202 <Form.Group className="mb-3" as={Row}>
203   <Form.Label> Nome Completo:</Form.Label>
204   <Col sm={12}>
205     <Form.Control
206       type="text"
207       name="bossUser"
208       placeholder="Nome completo"
209       value={nomeUser}
210       disabled
211     />
212   </Col>
213 </Form.Group>
214
215 <Form.Group className="mb-3" as={Row}>
216   <Form.Label> Telefone:</Form.Label>
217   <Col sm={12}>
218     <Form.Control
219       type="text"
220       name="bossUser"
221       placeholder="Nome do Responsável Local"
222       value={phone}
223       disabled
224     />
225   </Col>
226 </Form.Group>
227
228 <Form.Group className="mb-3" as={Row}>
229   <Form.Label> Nome do Responsável Local:</Form.Label>
230   <Col sm={12}>
231     <Form.Control
232       type="text"
233       name="bossUser"
234       placeholder="Nome do Responsável Local"
235       value={responsivelLocal}
236       onChange={(e) => setResponsivelLocal(e.target.value)}
237     />
238   </Col>
239 </Form.Group>
240 </div>
241 </div>
242
243 <Button variante="info" type="submit" onClick={onSubmit}>
244   Enviar
245 </Button>
246 <br />
247 </Form>

```

6  
87  
88



TAV Web

```
248     </div>
249   </main>
250   <footer className="footer">
251     <Footer />
252   </footer>
253 </div>
254 );
255 };
256
257 export default CreateTicket;
258
```

9  
X  
88  
/



## src/pages/detailPageEquipment/index.css

```
1  .entry-page {
2    height: 100vh;
3    display: flex;
4    justify-content: center;
5    align-items: center;
6  }
7
8  .show-box {
9    padding: 4rem 2rem;
10   margin-bottom: 3rem;
11   background-color: var(--bs-light);
12   border-radius: .3rem;
13   box-shadow: 0px 0px 15px -5px black;
14 }
15
16 .show-box2 {
17   height: 80vh;
18   padding: 4rem 2rem;
19   margin-bottom: 3rem;
20   background-color: var(--bs-light);
21   border-radius: .3rem;
22   box-shadow: 0px 0px 15px -5px black;
23 }
24 .form-box {
25   max-width: 550px;
26   margin-left: auto;
27   margin-right: auto;
28   padding: 4rem 2rem;
29   margin-bottom: 3rem;
30   background-color: var(--bs-light);
31   border-radius: .3rem;
32   box-shadow: 0px 0px 15px -5px black;
33 }
```

Handwritten notes in blue ink, including a large number '9' and some illegible scribbles.

## src/pages/detailPageEquipment/index.tsx

```

1 import { Footer } from "../../../components/Footer";
2 import { HeaderAdmin } from "../../../components/HeaderAdmin";
3 import { useNavigate, useParams } from "react-router-dom";
4 import { useContext, useEffect, useState } from "react";
5 import { EquipmentContext } from "../../../providers/equipment";
6 import "./index.css";
7 import api from "../../../services/api";
8
9 const DetailPageEquipment = () => {
10   const { id } = useParams();
11
12   const [place, setPlace] = useState("");
13   const [name, setName] = useState("");
14   const [code, setCode] = useState("");
15
16   let token = localStorage.getItem("authToken") || "";
17
18   useEffect(() => {
19     api.get(`/equipments/${parseInt(id as string)}`, {
20       headers: { Authorization: `Bearer ${token}` },
21     }).then((res) => {
22       setPlace(res.data.place as string);
23       setName(res.data.name as string);
24       setCode(res.data.code as string);
25     });
26   }, []);
27
28   const navigate = useNavigate();
29
30   const user = JSON.parse(localStorage.getItem("user") || "{}");
31
32   if (user.admin == false) {
33     if (user.email != "") {
34       navigate("/tickets");
35     } else {
36       navigate("/");
37     }
38   }
39   return <</>
40 }
41
42 return (
43   <div>
44     <header className="header mb-2">
45       <HeaderAdmin />
46     </header>
47
48     <main className="main">
49       <div className="show-bus">
50         <div className="row">
51           <h4 className="text-center">Dados do Equipamento:</h4>
52           <br />
53           <br />
54           <div className="row">Modelo do Equipamento:</div>
55           <div className="row">
56             {name}
57           </div>
58         </div>
59       </div>
60     </main>
61   </div>
62 );

```

9  
x

o  
aa  
//

```
56         </dd>
57
58         <dt className="row">Codigo do Equipamento:</dt>
59         <dd className="row">{code}</dd>
60
61         <dt className="row">Lugar de Instalação do Equipamento:</dt>
62         <dd className="row">{place}</dd>
63
64         <br />
65         <br />
66     </dl>
67 </div>
68 </main>
69 <footer className="footer">
70     <Footer />
71 </footer>
72 </div>
73 );
74 };
75
76 export default DetailPageEquipment;
77
78
79
```

6  
16  
18



TAV Web

## src/pages/detailPageModelEquipment/index.css

```
1 .entry-page {
2   height: 100vh;
3   display: flex;
4   justify-content: center;
5   align-items: center;
6 }
7
8 .show-box {
9   padding: 4rem 2rem;
10  margin-bottom: 3rem;
11  background-color: var(--bs-light);
12  border-radius: .3rem;
13  box-shadow: 0px 0px 15px -5px #black;
14 }
15
16 .show-box2 {
17   height: 80vh;
18   padding: 4rem 2rem;
19   margin-bottom: 3rem;
20   background-color: var(--bs-light);
21   border-radius: .3rem;
22   box-shadow: 0px 0px 15px -5px #black;
23 }
24 .form-box {
25   max-width: 550px;
26   margin-left: auto;
27   margin-right: auto;
28   padding: 4rem 2rem;
29   margin-bottom: 3rem;
30   background-color: var(--bs-light);
31   border-radius: .3rem;
32   box-shadow: 0px 0px 15px -5px #black;
33 }
```


9  
✓  
✓

## src/pages/detailPageModelEquipment/index.tsx

```

1  import React, { Component, useContext, useEffect, useState } from "react";
2  import axios from "axios";
3  import { Footer } from "../../../components/footer";
4  import { HeaderAdmin } from "../../../components/HeaderAdmin";
5  import { Col, Row } from "react-bootstrap";
6  import { EquipmentModelContext } from "../../../providers/equipmentModel";
7  import "../index.css";
8  import { useParams } from "react-router-dom";
9  import api from "../../../Services/api";
10
11  const DetailPageModelEquipment = () => {
12    const { equipmentModel } = useContext(EquipmentModelContext);
13    const { id } = useParams();
14
15    const idParams = parseInt(id as string);
16
17    const [name, setName] = useState("");
18    const [description, setDescription] = useState("");
19
20    let token = localStorage.getItem("authToken") || "";
21
22    useEffect(() => {
23      api.get(`/models/${parseInt(id as string)}`, {
24        headers: { Authorization: `Bearer ${token}` },
25      }).then((res) => {
26        setName(res.data.name as string);
27        setDescription(res.data.description as string);
28      });
29    }, []);
30
31    return (
32      <div>
33        <header className="header mb-2">
34          <HeaderAdmin />
35        </header>
36
37        <main className="main">
38          <div className="show-box">
39            <div className="row">
40              <div className="text-center">Dados do Modelo de Equipamento:</div>
41            </div>
42            <div>
43              <div className="row">Modelo do Equipamento:</div>
44              <div className="row">
45                <p>{name}</p>
46              </div>
47            </div>
48            <div className="row">Descrição:</div>
49            <div>
50              <div>
51                <div className="row-10">{description}</div>
52              </div>
53            </div>
54          </div>
55        </main>
56      </div>
57    );
58  };

```





TAV Web

```
54
55         <br />
56         <br />
57     </dl>
58 </div>
59 </main>
60 <footer className="footer">
61     <Footer />
62 </footer>
63 </div>
64 ;
65 };
66 export default DetailPageModelEquipment;
67
```



## src/pages/editEquipment/index.tsx

```

1  import { Form, Button, Row, Col } from "react-bootstrap";
2  import { Footer } from "../../../components/Footer";
3  import { HeaderAdmin } from "../../../components/HeaderAdmin";
4  import { useContext, useEffect, useState } from "react";
5  import { useNavigate, useParams } from "react-router-dom";
6  import { EquipmentContext } from "../../../providers/equipment";
7
8  import api from "../../../services/api";
9  import { EquipmentModelContext } from "../../../providers/equipmentModel";
10
11 interface IEquipment {
12   id: number;
13   name: string;
14   code: string;
15   place: string;
16   equipmentModelId: number;
17 }
18
19 const EditEquipment = () => {
20   const { id } = useParams();
21
22   const { equipments, updateEquipment, getOneEquipment } = useContext(EquipmentContext);
23   const { equipmentModels } = useContext(EquipmentModelContext);
24   const navigate = useNavigate();
25
26   const equipmentId = parseInt(id as string);
27   const [place, setPlace] = useState("");
28   const [name, setName] = useState("");
29   const [code, setCode] = useState("");
30
31
32   let token = localStorage.getItem("authToken") || ""
33
34   useEffect(() => {
35     api.get(`/equipments/${parseInt(id as string)}`, {
36       headers: { Authorization: `Bearer ${token}` },
37     }).then(res => {
38       setPlace(res.data.place as string);
39       setName(res.data.name as string);
40       setCode(res.data.code as string);
41     });
42   }, []);
43
44   const user = JSON.parse(localStorage.getItem("user") || "{}(admin: false, email: '')");
45
46   if (user.admin == false) {
47     if (user.email != "") {
48       navigate("/tickets");
49     } else {
50       navigate("/");
51     }
52   }
53   return <</>

```

95  
A 56  
7

TAV Web

```

184     <Form.Control
185       as="select"
186       name="place"
187       value={place}
188       onChange={(e) => setPlace(e.target.value)}
189     >
190     <option>Selecione o lugar</option>
191     <option value={"Atendimento"}>Atendimento</option>
192     <option value={"Secretaria"}>Secretaria</option>
193     <option value={"Desenvolvimento"}>Desenvolvimento</option>
194     <option value={"Escritório"}>Escritório</option>
195   </Form.Control>
196 </Col>
197 </Form.Group>
198
199 <Form.Group className="mb-3" as={Row}>
200   <Form.Label>Codigo do Equipamento:</Form.Label>
201   <Col sm={12}>
202     <Form.Control
203       type="text"
204       name="equipmentId"
205       placeholder="Codigo"
206       value={code}
207       onChange={(e) => setCode(e.target.value)}
208     />
209   </Col>
210 </Form.Group>
211
212   <Button variant="info" type="submit" onClick={onSubmit}>
213     Editar
214   </Button>
215   <br />
216 </Form>
217 </div>
218 </main>
219 <footer className="footer">
220   <Footer />
221 </footer>
222 </div>
223 |:
224 };
225
226 export default EditEquipment;
227

```

6  
90  
4



```

54
55
56 const onSubmit = (e: any) => {
57   e.preventDefault();
58
59   const data = {
60     name: name,
61     place: place,
62     code: code,
63   };
64
65   updateEquipment(equipmentId, data);
66   navigate("/equipments");
67 }
68
69 return (
70   <div>
71     <header className="header mb-2">
72       <Header/Admin />
73     </header>
74
75     <main className="main">
76       <div className="form-box p-0">
77         <Form autoComplete="off">
78           <div className="text-center w-100, fa-3 fa-normal"> Editor Equipamentos</div>
79           <hr />
80
81           <Form.Group className="mb-3" as={Row}>
82             <Form.Label Modelo do Equipamento:</Form.Label>
83             <Col sm={12}>
84               <Form.Control
85                 as="select"
86                 name="nomeModelo"
87                 placeholder="Modelo"
88                 value={nome}
89                 onChange={(e) => setName(e.target.value)}
90               >
91                 <option>Selecione o Modelo</option>
92                 {equipmentModels.map((equipment) => {
93                   return (
94                     <option value={equipment.name}> {equipment.name}</option>
95                   );
96                 })}
97               </Form.Control>
98             </Col>
99           </Form.Group>
100
101           <Form.Group className="mb-3" as={Row}>
102             <Form.Label Lugar Onde esta Instalado:</Form.Label>
103             <Col sm={12}>

```



TAV Web

```

255     </Col>
256   </Row>
257   <Table striped bordered hover>
258     <thead>
259       <tr>
260         <th>Modelo de Equipamento</th>
261         <th>Chamado Encerrados</th>
262         <th>Chamados em Aberto</th>
263         <th>Total</th>
264       </tr>
265     </thead>
266
267     <tbody>
268       {groupByCall(reports?.development).map((report) => {
269         <tr>
270           <td style={{width: "360px"}}>{report.name}</td>
271           <td>{report.open}</td>
272           <td>{report.close}</td>
273           <td>{report.total}</td>
274         </tr>
275       )}}
276     </tbody>
277
278     <thead>
279       <tr>
280         <th>Total</th>
281         <th>
282           {
283             reports?.development.filter(
284               (r) => r.status == "opening"
285             ).length
286           }
287         </th>
288         <th>
289           {
290             reports?.development.filter(
291               (r) => r.status == "closed"
292             ).length
293           }
294         </th>
295         <th>{reports?.development.length}</th>
296       </tr>
297     </thead>
298   </Table>
299 </>
300 | : (
301 <</>
302 )}
303
304 (reports.office.length > 0 ? (
305   <

```

123  
L 9 C

TAV Web

```

306 <Row style={{ marginTop: "10px" }}>
307 <Col className="text-center">
308 <h4>Escritório</h4>
309 </Col>
310 </Row>
311 <Table striped bordered hover>
312 <thead>
313 <tr>
314 <th>Modelo de Equipamento</th>
315 <th>Chamado Encerrados</th>
316 <th>Chamados em Aberto</th>
317 <th>Total</th>
318 </tr>
319 </thead>
320
321 <tbody>
322 {groupByCall(reports?.office).map((report) => (
323 <tr>
324 <td style={{ width: "350px" }}>{report.name}</td>
325 <td>{report.close}</td>
326 <td>{report.open}</td>
327 <td>{report.total}</td>
328 </tr>
329 )]}
330 </tbody>
331
332 <thead>
333 <tr>
334 <th>Total</th>
335 <th>
336 {
337 reports?.office.filter((r) => r.status == "closed")
338 .length
339 }
340 </th>
341 <th>
342 {
343 reports?.office.filter((r) => r.status == "opening")
344 .length
345 }
346 </th>
347 <th>{reports?.office.length}</th>
348 </tr>
349 </thead>
350 </Table>
351 </>
352 ) = (
353 <<>
354 )}
355 </div>

```

129 ✖  
 4  
 ↷

TAV Web

```
354     </div>
355   ): {
356     </div>
357   }
358   <main>
359     <footer className="footer">
360       <Footer />
361     </footer>
362   </div>
363 </div>
364 </div>
365 }
366 }
367
```

125  
A  
D  
P



TAV Web

## src/pages/signup/index.tsx

```

1  import { useContext, useState } from "react";
2  import axios from "axios";
3  import styles from "../styles.module.css";
4  import { AuthContext } from "../../../providers/user";
5
6  const Signup = () => {
7
8      const { signUp } = useContext(AuthContext)
9
10     const [firstName, setFirstName] = useState("");
11     const [lastName, setLastName] = useState("");
12     const [email, setEmail] = useState("");
13     const [password, setPassword] = useState("");
14     const [phone, setPhone] = useState("");
15
16     const handleSubmit = async (e: any) => {
17         e.preventDefault();
18         const data = {
19             firstName: firstName,
20             lastName: lastName,
21             email: email,
22             password: password,
23             phone: phone,
24             terms_of_use: true
25         }
26         signUp(data)
27     };
28
29     return (
30         <div className={styles.signup_container}>
31             <div className={styles.signup_form_container}>
32                 <div className={styles.left}>
33                     <h1>Bem Vindo</h1>
34                     <a href="/">
35                         <button type="button" className={styles.white_btn}>
36                             Login
37                         </button>
38                     </a>
39                 </div>
40                 <div className={styles.right}>
41                     <form className={styles.form_container} onSubmit={handleSubmit}>
42                         <h1>Criar Conta</h1>
43                         <input
44                             type="text"
45                             placeholder="Primeiro Nome"
46                             name="firstName"
47                             onChange={(e) => setFirstName(e.target.value)}
48                             value={firstName}
49                             required
50                             className={styles.input}
51                         />

```

126  


TAV Web

```
52     <input
53       type="text"
54       placeholder="Informe Nome"
55       name="lastName"
56       onChange={(e) => setLastName(e.target.value)}
57       value={lastName}
58       required
59       className={styles.input}
60     />
61     <input
62       type="email"
63       placeholder="Email"
64       name="email"
65       onChange={(e) => setEmail(e.target.value)}
66       value={email}
67       required
68       className={styles.input}
69     />
70     <input
71       type="text"
72       placeholder="Telefone"
73       name="telefone"
74       onChange={(e) => setPhone(e.target.value)}
75       value={phone}
76       required
77       className={styles.input}
78     />
79     <input
80       type="password"
81       placeholder="Senha"
82       name="password"
83       onChange={(e) => setPassword(e.target.value)}
84       value={password}
85       required
86       className={styles.input}
87     />
88     <button type="submit" className={styles.green_btn} onClick={handleSubmit}>
89       Registrar
90     </button>
91   </form>
92 </div>
93 </div>
94 </div>
95 </div>
96 !;
97 }
98
99 export default Signup;
100
```

127



## src/pages/signup/styles.module.css

```
1  .signup_container {
2    width: 100%;
3    min-height: 100vh;
4    background-color: #f7f7f7;
5    display: flex;
6    align-items: center;
7    justify-content: center;
8  }
9
10 .signup_form_container {
11   width: 900px;
12   height: 300px;
13   display: flex;
14   border-radius: 10px;
15   box-shadow: 0px 3px 3px -2px #000000 / 30%,
16             0px 3px 4px 0px #000000 / 14%, 0px 1px 8px 0px #000000 / 12%;
17 }
18
19 .left {
20   flex: 1;
21   display: flex;
22   flex-direction: column;
23   align-items: center;
24   justify-content: center;
25   background-color: #3bb19e;
26   border-top-left-radius: 10px;
27   border-bottom-left-radius: 10px;
28 }
29
30 .left h1 {
31   margin-top: 0;
32   color: #ffffff;
33   font-size: 35px;
34   align-self: center;
35 }
36
37 .right {
38   flex: 2;
39   display: flex;
40   flex-direction: column;
41   align-items: center;
42   justify-content: center;
43   background-color: #ffffff;
44   border-top-right-radius: 10px;
45   border-bottom-right-radius: 10px;
46 }
47
48 .form_container {
49   display: flex;
50   flex-direction: column;
51   align-items: center;
52 }
```

Handwritten notes and signatures in blue ink, including the letters "R.B." and several illegible scribbles.

```
53
54 .form_container h1 {
55     font-size: 40px;
56     margin-top: 0;
57 }
58
59 .input {
60     outline: none;
61     border: none;
62     width: 370px;
63     padding: 15px;
64     border-radius: 10px;
65     background-color: #edf5f3;
66     margin: 5px 0;
67     font-size: 14px;
68 }
69
70 .error_msg {
71     width: 370px;
72     padding: 15px;
73     margin: 5px 0;
74     font-size: 14px;
75     background-color: #f44646;
76     color: white;
77     border-radius: 5px;
78     text-align: center;
79 }
80
81 .white_btn,
82 .green_btn {
83     border: none;
84     outline: none;
85     padding: 12px 0;
86     background-color: white;
87     border-radius: 20px;
88     width: 100px;
89     font-weight: bold;
90     font-size: 14px;
91     cursor: pointer;
92 }
93
94 .green_btn {
95     background-color: #30b196;
96     color: white;
97     margin: 10px;
98 }
```

TAV Web

## src/pages/ticket/index.css

```
1  .entry-page {
2    height: 100vh;
3    display: flex;
4    justify-content: center;
5    align-items: center;
6  }
7
8  .show-box {
9    padding: 4rem 2rem;
10   margin-bottom: 3rem;
11   background-color: var(--bs-light);
12   border-radius: .3rem;
13   box-shadow: 0px 0px 15px -5px black;
14 }
15
16 .show-box2 {
17   height: 80vh;
18   padding: 4rem 2rem;
19   margin-bottom: 3rem;
20   background-color: var(--bs-light);
21   border-radius: .3rem;
22   box-shadow: 0px 0px 15px -5px black;
23 }
24 .form-box {
25   max-width: 550px;
26   margin-left: auto;
27   margin-right: auto;
28   padding: 4rem 2rem;
29   margin-bottom: 3rem;
30   background-color: var(--bs-light);
31   border-radius: .3rem;
32   box-shadow: 0px 0px 15px -5px black;
33 }
```

BO  
↓  
↓  
↓

TAV Web

## src/pages/ticket/index.tsx

```

1  import { useContext, useEffect, useState } from "react";
2  import { Footer } from "../../../components/footer";
3  import { HeaderUser } from "../../../components/HeaderUser";
4  import { CallsContext } from "../../../providers/call";
5  import { useNavigate, useParams } from "react-router-dom";
6  import api from "../../../services/api";
7  import "../index.css";
8  import { HeaderAdmin } from "../../../components/HeaderAdmin";
9
10 const DetailPageTicket = () => {
11   const { id } = useParams();
12
13   const [userRequesterName, setUserRequesterName] = useState("");
14   const [responsibleLocale, setResponsibleLocale] = useState("");
15   const [status, setStatus] = useState("");
16   const [subject, setSubject] = useState("");
17   const [equipmentName, setEquipmentName] = useState("");
18   const [equipmentCode, setEquipmentCode] = useState("");
19   const [equipmentPlace, setEquipmentPlace] = useState("");
20   const [description, setDescription] = useState("");
21   const [phone, setPhone] = useState("");
22
23   const navigate = useNavigate();
24
25   const user = JSON.parse(
26     localStorage.getItem("user") || '{"admin: false, email: ""}'
27   );
28
29   let token = localStorage.getItem("authToken") || "";
30
31   useEffect(() => {
32     api
33       .get(`/calls/${parseInt(id as string)}`, {
34         headers: { Authorization: `Bearer ${token}` },
35       })
36       .then((res) => {
37         setEquipmentCode(res.data.equipmentCode as string);
38         setEquipmentName(res.data.equipmentName as string);
39         setSubject(res.data.subject as string);
40         setStatus(res.data.status as string);
41         setResponsibleLocale(res.data.responsibleLocale as string);
42         setUserRequesterName(res.data.userRequesterName as string);
43         setDescription(res.data.description as string);
44         setEquipmentPlace(res.data.equipmentPlace as string);
45         setPhone(user.phone);
46       });
47   }, []);
48
49   if (user.email == "") {
50     navigate("/");
51     return <</>;
52   }

```



TAV Web

```

33
34     return |
35         <div>
36             <header className="header sb-2">
37                 {user.admin == true ? <headerAdmin /> : <headerUser />}
38             </header>
39
40             <main className="main">
41                 <div className="show-box">
42                     <h4 className="text-center">
43                         Status: { " " }
44                         {status == "opening" ? "Chamado aberto" : "Chamado encerrado"}
45                     </h4>
46                     <table className="row">
47                         <tr>
48                             <td className="text-left">Dados do Chamado:</td>
49                             <td />
50                             <td className="col-sm-3">Assunto:</td>
51                             <td className="col-sm-9">{subject}</td>
52                             <td className="col-sm-3">Equipamento:</td>
53                             <td className="col-sm-9">
54                                 <p>{equipmentName}</p>
55                             </td>
56                             <td className="col-sm-3">Codigo do Equipamento:</td>
57                             <td className="col-sm-9">
58                                 <p>{equipmentCode}</p>
59                             </td>
60                             <td className="col-sm-3">Local de Instalação:</td>
61                             <td className="col-sm-9">
62                                 <p>{equipmentPlace}</p>
63                             </td>
64                             <td className="col-sm-3">Descrição:</td>
65                             <td className="col-sm-9">{description}</td>
66
67                             <tr>
68                                 <td />
69                                 <td />
70                                 <td />
71                                 <td className="text-left">Dados do Usuário:</td>
72                                 <td />
73                                 <td className="col-sm-3">Nome:</td>
74                                 <td className="col-sm-9">{userRequestorName}</td>
75                                 <td className="col-sm-3">Telefones:</td>
76                                 <td className="col-sm-9">{phone}</td>
77                                 <td className="col-sm-3">Responsavel do Local:</td>
78                                 <td className="col-sm-9">{responsibileLocal}</td>
79
80                             </tr>
81                         </table>
82                     </div>
83                 </main>
84             </div>
85         </div>
86
87         <footer className="footer">
88             <Footer />
89         </footer>
90     </div>
91
92     export default DetailPageTicket;
93

```

132

ST

A

TAV Web

## src/pages/tickets/index.tsx

```

1 import { useEffect, useState } from "react";
2 import { Table, Button, Row, Col, Form, Container } from "react-bootstrap";
3 import { Footer } from "../../components/Footer";
4 import { HeaderComponent } from "../../components/Header";
5
6 import { useContext } from "react";
7 import { callsContext } from "../../../providers/calls";
8 import { useNavigate } from "react-router-dom";
9
10
11 interface ICall {
12   id: number;
13   subject: string;
14   data: string;
15   description: string;
16   responsibleLocal: string;
17   external: string;
18   userRequesterId: number;
19   userSolvedId: number;
20   userRequesterName: string;
21   equipmentId: number;
22   equipmentCode: string;
23   equipmentPlace: string;
24   equipmentMass: string;
25 }
26
27 const LandingPageCall = () => {
28   const { calls, updateCall, setCalls, listCalls } = useContext(callsContext);
29   const [searchTerm, setSearchTerm] = useState("");
30
31   const filterCalls = (searchTerm: string) => {
32     const result = calls.filter((call) => {
33       call.subject.toLowerCase().includes(searchTerm.toLowerCase());
34     });
35     setSearchTerm(searchTerm);
36   };
37
38   useEffect(() => {
39     listCalls();
40   }, []);
41
42   const navigate = useNavigate();
43
44   const user = JSON.parse(localStorage.getItem("user")) || { name: "User", email: "" };
45
46   if (user.email === "" || user.name === "") {
47     navigate("/login");
48     return;
49   }
50
51   return (
52     <div>
53       <Header className="header" id="2" />
54       <Header id="1" />
55     </div>
56   );
57
58   const resolveCall = (id: number) => {
59     updateCall(id);
60     const newCalls = calls.filter((call) => call.id !== id);
61     console.log("Resolvido");
62     setCalls(newCalls);
63   };
64
65   return (
66     <div>
67       <div style="display: flex; justify-content: space-between; align-items: center; margin-bottom: 10px;">
68         <div>
69           <h2>Chamados</h2>
70         </div>
71         <div>
72           <input type="text" value={searchTerm} />
73           <button type="button" value="Pesquisar" />
74         </div>
75       </div>
76       <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;">
77         <thead>
78           <tr>
79             <th>ID</th>
80             <th>Assunto</th>
81             <th>Data</th>
82             <th>Descrição</th>
83             <th>Responsável Local</th>
84             <th>Equipamento</th>
85             <th>Status</th>
86             <th>Ações</th>
87           </tr>
88         </thead>
89         <tbody>
90           <tr>
91             <td>1</td>
92             <td>Chamado de teste</td>
93             <td>2023-09-15</td>
94             <td>Teste de chamados</td>
95             <td>Administrador</td>
96             <td>Equipamento 1</td>
97             <td>Aberto</td>
98             <td>
99               <button type="button" value="Atualizar" />
100               <button type="button" value="Resolver" />
101             </td>
102           </tr>
103         </tbody>
104       </table>
105     </div>
106   );
107 };

```

133

A  
9



TAV Web

```

129         <button
130         </button
131         className="btn-warning"
132         onClick={() => reservecall.call()}
133         >
134         <input type="text" value={Encerrar Chasado} />
135         </button>
136     </div>
137 </div>
138 </tbody>
139 </table>
140 </div>
141 </div>
142 </div>
143 </div>
144 </div>
145 </div>
146 </div>
147 </div>
148 </div>
149 </div>
150 </div>

```

## src/pages/users/index.css

```

1 .form-users {
2   max-width: 550px;
3   margin-left: auto;
4   margin-right: auto;
5   padding: 4rem 2rem;
6   margin-bottom: 3rem;
7   background-color: var(--bs-light);
8   border-radius: 0.3rem;
9   box-shadow: 0px 0px 15px -5px #000;
10 }

```

BS  




TAV Web

## src/pages/users/index.tsx

```

1 import * as React from "react";
2 import Table from "mui/material/Table";
3 import TableBody from "mui/material/TableBody";
4 import TableCell from "mui/material/TableCell";
5 import TableContainer from "mui/material/TableContainer";
6 import TableHead from "mui/material/TableHead";
7 import TableRow from "mui/material/TableRow";
8 import Paper from "mui/material/Paper";
9 export { HeaderAdmin } from "../../../components/HeaderAdmin";
10 export { AuthContext } from "../../../providers/user";
11 import { useContext, useEffect, useState } from "react";
12 import "../../../index.css";
13
14
15 import CheckIcon from "@material-ui/icons/Check";
16 import CloseIcon from "@material-ui/icons/Close";
17 export { Footer } from "../../../components/footer";
18
19 interface IUserInterface {
20   id: number;
21   firstName: string;
22   lastName: string;
23   phone: string;
24   email: string;
25   isAdmin: boolean;
26   terms_of_user: boolean;
27   isActive: boolean;
28 }
29
30 interface ITableUsers {
31   users: IUserInterface[];
32 }
33
34 const TableUsers = () => {
35   const { getUsers, users, acceptOrRejectUsers, setUsers } = useContext(AuthContext);
36
37   useEffect(() => {
38     getUsers();
39   }, []);
40
41   const deleteUser = (id: number) => {
42     acceptOrRejectUsers(false, id);
43
44     const newUsers = users.filter(user => user.id != id);
45     setUsers(newUsers);
46   };
47
48   const acceptUser = (id: number) => {
49     acceptOrRejectUsers(true, id);
50     const newUsers = users.filter(user => user.id != id);
51     setUsers(newUsers);
52   };
53
54   return (
55     <TableContainer component={Paper}>
56       <Table sx={{ minWidth: 400 }} size="small" aria-label="a simple table">
57         <TableHead>
58           <TableRow>
59             <TableCell>Email</TableCell>
60             <TableCell align="right">Ações</TableCell>
61           </TableRow>
62         </TableHead>
63         <TableBody>

```

136



TAV Web

## src/providers/call.tsx

```

1  import {withDecode} from "jwt-decode";
2  import {
3    createContext,
4    Dispatch,
5    ReactNode,
6    SetStateAction,
7    useEffect,
8    useState,
9  } from "react";
10 import api from "../../services/api";
11
12 interface ICall {
13   id: number;
14   subject: string;
15   date: string;
16   description: string;
17   responsibleLocale: string;
18   status: string;
19   userRequesterId: number;
20   userSolvedId: number;
21   userRequesterName: string;
22   equipmentId: number;
23   equipmentCode: string;
24   equipmentPlace: string;
25   equipmentName: string;
26 }
27
28 interface IUser {
29   id: number;
30   firstName: string;
31   lastName: string;
32   phone: string;
33   email: string;
34   password: string;
35   isActive: boolean;
36   terms_of_use: boolean;
37   isActive: boolean;
38 }
39
40 interface ICallCreate {
41   subject: string;
42   description: string;
43   responsibleLocale: string;
44   equipmentId: number;
45 }
46
47 interface IReport {
48   office: ICall[];
49   secretary: ICall[];
50   development: ICall[];
51   attendance: ICall[];
52 }
53
54 interface ICallProvider {
55   children: ReactNode;
56 }
57
58 interface ICallContext {
59   calls: ICall[];
60   createCall: (call: ICallCreate) => void;
61   listCalls: () => void;
62   getOneCall: (callId: number) => void;
63   deleteCall: (callId: number) => void;
64   updateCall: (callId: number) => void;
65   createReport: (start: string, end: string) => Promise<IReport>;
66   setCalls: Dispatch<SetStateAction<ICall[]>>;

```

138  
 A  
 P  
 S

TAV Web

```

07 }
08
09 const defaultValues: ICallContext = {
10   calls: [],
11   createCall: (ideta: ICallCreate) => {},
12   listCall: () => {},
13   getOneCall: (callId: number) => {},
14   deleteCall: (callId: number) => {},
15   updateCall: (callId: number) => {},
16   createReport: (start: string, end: string) => {
17     return Promise.resolve({
18       office: [],
19       secretary: [],
20       development: [],
21       attendance: [],
22     });
23   },
24   setCalls: () => {},
25 };
26
27 export const CallContext = createContext<ICallContext>(defaultValues);
28
29 export const CallsProvider = ({ children }: ICallProvider) => {
30   const [calls, setCalls] = useState<ICall[]>([]);
31   const [reports, setReports] = useState<IReport>({
32     office: [],
33     secretary: [],
34     development: [],
35     attendance: [],
36   });
37
38   let token = "";
39   if (typeof window !== "undefined") {
40     token = localStorage.getItem("authToken") || "";
41   } else {
42     //const user: User = getCookie(token);
43   }
44
45   const listCall = () => {
46     api
47       .get("/calls", { headers: { Authorization: `Bearer ${token}` } })
48       .then((resp) => {
49         setCalls(resp.data);
50       });
51   };
52
53   useEffect(() => {
54     listCall();
55   }, []);
56
57   const createCall = (data: ICallCreate) => {
58     api
59       .post("/calls", data, { headers: { Authorization: `Bearer ${token}` } })
60       .then((resp) => {
61         setCalls([...calls, resp.data]);
62       });
63   };
64
65   const getOneCall = (callId: number) => {
66     api.get(`/calls/${callId}`, {
67       headers: { Authorization: `Bearer ${token}` },
68     });
69   };
70 }

```

139






TAV Web

```

138
139 const deleteCall = (callId: number) => {
140   api.delete(`/calls/${callId}`, {
141     headers: { Authorization: `Bearer ${token}` },
142   });
143 }
144
145 const updateCall = (callId: number) => {
146   api
147     .patch(
148       `/calls/${callId}`,
149       {},
150       {
151         headers: { Authorization: `Bearer ${token}` },
152       }
153     )
154     .catch((err) => console.log("Err -> ", err));
155 }
156
157 const createReport = async (start: string, end: string) => {
158   const res = await api.post("/report", {
159     "date_start": start,
160     "date_end": end
161   }, {
162     headers: { Authorization: `Bearer ${token}` },
163   });
164
165   return res.data as IReport;
166 }
167
168 return {
169   <CallsContext.Provider
170     value={{
171       calls,
172       createCall,
173       createReport,
174       deleteCall,
175       getOneCall,
176       listCalls,
177       updateCall,
178       setCalls,
179     }}
180   >
181     {children}
182   </CallsContext.Provider>
183   </>
184 };
185 }
186 }
187

```

TAV Web

## src/providers/equipment.tsx

```

1 import { useState } from "react";
2 import { createContext, ReactNode, useEffect, useReducer } from "react";
3 import api from "../services/api";
4
5 interface IEquipment {
6   id: number;
7   name: string;
8   code: string;
9   place: string;
10  equipmentModelId: number;
11 }
12
13 interface IEquipmentUpdate {
14   name: string;
15   code: string;
16   place: string;
17 }
18
19 interface IEquipmentCreate {
20   name: string;
21   place: string;
22   equipmentModelId: number;
23   code: string;
24 }
25
26
27 interface IUser {
28   id: number;
29   firstName: string;
30   lastName: string;
31   phone: string;
32   email: string;
33   password: string;
34   isAdmin: boolean;
35   terms_of_use: boolean;
36   isActive: boolean;
37 }
38
39 interface IEquipmentProvider {
40   children: ReactNode;
41 }
42
43 interface IEquipmentContext {
44   equipments: IEquipment[];
45   createEquipment: (equipment: IEquipmentCreate) => void;
46   listEquipments: () => void;
47   getOneEquipment: (equipmentId: number) => void;
48   deleteEquipment: (equipmentId: number) => void;
49   updateEquipment: (equipmentId: number, data: IEquipmentUpdate) => void;
50 }
51
52 const defaultValues: IEquipmentContext = {
53   equipments: [],
54   createEquipment: (equipment: IEquipmentCreate) => {},
55   listEquipments: () => {},
56   getOneEquipment: (equipmentId: number) => {},
57   deleteEquipment: (equipmentId: number) => {},
58   updateEquipment: (equipmentId: number, data: IEquipmentUpdate) => {},
59 };
60
61 export const EquipmentsContext = createContext<IEquipmentContext>(defaultValues);
62
63 export const EquipmentsProvider = ({ children }: IEquipmentProvider) => {
64   const [equipments, setEquipments] = useState<IEquipment[]>([]);
65
66   let token = localStorage.getItem("authToken") || ""

```

12/1  
X  
X  
7



TAV Web

## src/providers/equipmentModel.tsx

```

1 import { decode } from "jwt-decode";
2 import { createContext, Dispatch, ReactNode, useEffect, useState } from "react";
3 import api from "../services/api";
4
5 interface IEquipmentModel {
6   id: number;
7   name: string;
8   description: string;
9 }
10
11 interface IEquipmentModelCreate {
12   name: string;
13   description: string;
14 }
15
16 interface IEquipmentModelUpdate {
17   name: string;
18   description: string;
19 }
20
21 interface IUser {
22   id: number;
23   firstName: string;
24   lastName: string;
25   phone: string;
26   email: string;
27   password: string;
28   isAdmin: boolean;
29   terms_of_user: boolean;
30   isActive: boolean;
31 }
32
33 interface IEquipmentModelProvider {
34   children: ReactNode;
35 }
36
37 interface IEquipmentModelContext {
38   equipmentModels: IEquipmentModel[];
39   createEquipmentModel: (equipmentModel: IEquipmentModelCreate) => void;
40   listEquipmentModels: () => void;
41   getOneEquipmentModel: (equipmentModelId: number) => void;
42   deleteEquipmentModel: (equipmentModelId: number) => void;
43   updateEquipmentModel: (
44     equipmentModelId: number,
45     data: IEquipmentModelUpdate
46   ) => void;
47   setEquipmentModels: Dispatch<React.SetStateAction<IEquipmentModel[]>> => void;
48 }
49
50 const defaultValues: IEquipmentModelContext = {
51   equipmentModels: [],
52   setEquipmentModels: () => {},
53   createEquipmentModel: (equipmentModel: IEquipmentModelCreate) => {},
54   listEquipmentModels: () => {},
55   getOneEquipmentModel: (equipmentModelId: number) => {},
56   deleteEquipmentModel: (equipmentModelId: number) => {},
57   updateEquipmentModel: (equipmentModelId: number, data: IEquipmentModelUpdate) => {},
58 };
59
60 export const EquipmentModelContext =
61   createContext<IEquipmentModelContext>(defaultValues);
62

```

143  
 0 143 A  
 0 143

```

63 export const EquipmentModelProvider = ({
64   children,
65 }) => {EquipmentModelProvider} => {
66   const [equipmentModels, setEquipmentModels] = useState<IEquipmentModel[]>([]);
67
68   let token = "";
69   if (typeof window !== "undefined") {
70     token = localStorage.getItem("authToken") || "";
71   } else {
72   }
73   //const user: IUser = jwtDecode(token);
74
75   const listEquipmentModels = () => {
76     api
77       .get("/models", { headers: { Authorization: `Bearer ${token}` } })
78       .then((resp) => {
79         setEquipmentModels(resp.data);
80       });
81   };
82
83   useEffect(() => {
84     listEquipmentModels();
85     // [equipmentModels]
86
87     const createEquipmentModel = (data: IEquipmentModelCreate) => {
88       api.post("/models", data, {
89         headers: { Authorization: `Bearer ${token}` },
90       });
91     };
92
93     const getOneEquipmentModel = (equipmentModelId: number) => {
94       api.get(`/models/${equipmentModelId}`, {
95         headers: { Authorization: `Bearer ${token}` },
96       });
97     };
98
99     const deleteEquipmentModel = (equipmentModelId: number) => {
100       api.delete(`/models/${equipmentModelId}`, {
101         headers: { Authorization: `Bearer ${token}` },
102       });
103     };
104
105     const updateEquipmentModel = {
106       equipmentModelId: number,
107       data: IEquipmentModelUpdate
108     } => {
109       api.patch(`/models/${equipmentModelId}`, data, {
110         headers: { Authorization: `Bearer ${token}` },
111       });
112     };
113
114     return {
115       <EquipmentModelContext.Provider>
116         value={{
117           createEquipmentModel,
118           deleteEquipmentModel,
119           equipmentModels,
120           getOneEquipmentModel,
121           listEquipmentModels,
122           updateEquipmentModel,
123           setEquipmentModels
124         }}
125       </EquipmentModelContext.Provider>
126     </children>
127   } </EquipmentModelContext.Provider>
128 }
129 }

```

104  
 X 9 A  
 7

TAV Web

## src/providers/index.tsx

```
1 import { ReactNode } from "react";
2 import { CallsProvider } from "../call";
3 import { EquipmentProvider } from "../equipment";
4 import { EquipmentModelProvider } from "../equipmentModel";
5 import { AuthProvider } from "../user";
6
7 interface IProviders {
8   children: ReactNode;
9 }
10
11 const Providers = ({ children }: IProviders) => {
12   return (
13     <AuthProvider>
14       <CallsProvider>
15         <EquipmentProvider>
16           <EquipmentModelProvider>
17             {children}
18           </EquipmentModelProvider>
19         </EquipmentProvider>
20       </CallsProvider>
21     </AuthProvider>
22   );
23 };
24
25 export default Providers;
26
```

143  
E  
A  
D

TAV Web

## src/providers/user.tsx

```

1 import { createContext, Dispatch, ReactNode, SetStateAction, useContext, useState } from 'react'
2 import api from '~/services/api'
3 import { Success } from 'react-toastify'
4 import { useNavigate } from 'react-router-dom'
5
6 interface StaffProvider {
7   children: ReactNode
8 }
9
10 interface Staff {
11   password: string
12   email: string
13 }
14
15 interface Usuario {
16   firstName: string
17   password: string
18   lastName: string
19   email: string
20   phone: string
21 }
22
23 interface Login {
24   email: string
25   password: string
26   firstName: string
27   id: string
28 }
29
30 interface UserInactivation {
31   id: number
32   firstName: string
33   lastName: string
34   phone: string
35   email: string
36   isActive: boolean
37   terms_of_use: boolean
38   isActive: boolean
39 }
40
41 interface UserLogged {
42   firstName: string
43   email: string
44   id: number
45 }
46
47 interface StaffContext {
48   signIn: (data: Staff) => void
49   signUp: () => void
50   signIn: (data: Usuario) => void
51   isLogged: boolean
52   users: UserInactivation[]
53   getUsers: () => void
54   createNewUser: (active: boolean, id: number) => void
55   updateUser: (dispatch: Dispatch<SetStateAction<UserInactivation[]>>) => void
56   userLogged: UserLogged
57   setLogged: (dispatch: Dispatch<SetStateAction<UserLogged>>) => void
58 }
59
60 const defaultValues: StaffContext = {
61   signIn: (data: Staff) => console.log('hi'),
62   signUp: () => console.log('hi'),
63   signIn: (data: Usuario) => console.log('hi'),
64   isLogged: true,
65   users: [],
66 }

```

146



TAV Web

```

46   getUsers: () => {},
47   acceptOrRejectUsers: (accept: boolean, id: number) => {},
48   setUsers: () => {},
49   userLogged: (email: "", firstName: "", id: 0),
50   setUserLogged: () => {}
51 }
52
53 export const AuthContext = createContext<AuthContext>(defaultValues);
54
55 export const AuthProvider = ({ children }: AuthProvider) => {
56
57   const localToken = localStorage.getItem("authToken") || ""
58
59   const navigate = useNavigate();
60
61   const [isLoggedIn, setIsLoggedIn] = useState(
62     localToken != "" ? true : false
63   );
64
65   const [users, setUsers] = useState<UserInertia[]>([])
66   const [userLogged, setUserLogged] = useState<UserLogged>({
67     email: "", firstName: "", id: 0
68   })
69
70   const signIn = (data: IAuth) => {
71     api
72       .post("/login", data)
73       .then((response) => {
74
75         const user: User = jwtDecode(response.data.accessToken)
76         localStorage.setItem("user", JSON.stringify(user))
77         localStorage.setItem("authToken", response.data.accessToken);
78         console.log("USER TAKEN ", user)
79         setIsLoggedIn(true);
80         setUserLogged({email: user.email, firstName: user.firstName, id: parseInt(user.id)});
81         //user.admin == true?
82         navigate("/admin");
83       } else {
84         navigate("/signin");
85       }
86     }
87   }
88   .catch((err) => {
89     console.log("Error -> ", err)
90   })
91 }
92
93 const signOut = () => {
94   localStorage.clear();
95   navigate("/", {
96     state: -1
97   })
98   return window.location.reload();
99 }
100
101 const signUp = (data: IUser) => {
102   console.log("User -> ", data)
103   api
104     .post("/users", data)
105     .then((response) => {
106       console.log("Criado")
107       navigate("/", {
108         state: -1

```

132

```

128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

148  
 ✓  
 ✓



TAV Web

## src/routes/index.tsx

```

1 import { Routes, Route } from "react-router-dom";
2
3 import Login from "../pages/login";
4 import CreateEquipment from "../pages/createEquipment";
5 import CreateModelEquipment from "../pages/createModelEquipment";
6 import CreateTicket from "../pages/createTicket";
7 import DetailPageEquipment from "../pages/detailPageEquipment";
8 import DetailPageModelEquipment from "../pages/detailPageModelEquipment";
9 import EditEquipment from "../pages/editEquipment";
10 import EditModelEquipment from "../pages/editModelEquipment";
11 import LandingPageEquipment from "../pages/LandingPageEquipment";
12 import LandingPageModelEquipment from "../pages/LandingPageModelEquipment";
13 import Signup from "../pages/signup";
14 import DetailPageTicket from "../pages/ticket";
15 import LandPageAdmin from "../pages/LandingPageAdmin";
16 import Users from "../pages/users";
17 import { ReportTicket } from "../pages/reportTicket";
18 import LandingPageTicket from "../pages/LandingPageTicket";
19
20 const RoutesApp = () => {
21   return (
22     <div className="App">
23       <Routes>
24         <Route index element={<Login />} />
25         <Route path="/signup" element={<Signup />} />
26         <Route path="/users" element={<Users />} />
27         <Route path="/report" element={<ReportTicket />} />
28
29         <Route path="/equipments" element={<LandingPageEquipment />} />
30         <Route path="/equipments/:id" element={<DetailPageEquipment />} />
31         <Route path="/addEquipment" element={<CreateEquipment />} />
32         <Route path="/editEquipment/:id" element={<EditEquipment />} />
33
34         <Route path="/modelEquipment" element={<LandingPageModelEquipment />} />
35         <Route path="/modelEquipment/:id" element={<DetailPageModelEquipment />} />
36         <Route path="/addModelEquipment" element={<CreateModelEquipment />} />
37         <Route path="/editModelEquipment/:id" element={<EditModelEquipment />} />
38
39         <Route path="/admin" element={<LandPageAdmin />} />
40         <Route path="/tickets" element={<LandingPageTicket />} />
41         <Route path="/tickets/:id" element={<DetailPageTicket />} />
42         <Route path="/add" element={<CreateTicket />} />
43
44       </Routes>
45     </div>
46   );
47 }
48
49 export default RoutesApp;

```

149  
  


TAV Web

## src/services/api.ts

```
1 import axios from "axios";
2
3 const api = axios.create({
4   baseURL: "http://localhost:5000"
5 });
6
7 export default api;
8
```

## src/App.css

```
1 .default-layout{
2   display: grid;
3   grid-template-rows: 80px auto 3 rem;
4   grid-template-areas: "header" "main" "footer";
5 }
6
7 .header{
8   grid-area: header;
9 }
10
11 .main{
12   grid-area: main;
13   min-height: 88vh;
14 }
15
16 .footer{
17   grid-area: footer;
18   background: black;
19   padding: 1rem;
20   color: white;
21 }
```

## src/App.test.tsx

```
1 import React from 'react';
2 import { render, screen } from '@testing-library/react';
3 import App from './App';
4
5 test('renders learn react link', () => {
6   render(<App />);
7   const linkElement = screen.getByText(/learn react/i);
8   expect(linkElement).toBeInTheDocument();
9 });
10
```

130  
#  
#  
#



TAV Web

## src/App.tsx

```

1 import './App.css';
2 import RoutesApp from './routes';
3 import 'bootstrap/dist/css/bootstrap.min.css';
4
5 function App() {
6   return (
7     <RoutesApp />
8   );
9 }
10
11 export default App;
12

```

## src/index.css

```

1 body {
2   margin: 0;
3   font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',
4     'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',
5     sans-serif;
6   -webkit-font-smoothing: antialiased;
7   -moz-osx-font-smoothing: grayscale;
8 }
9
10 code {
11   font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',
12     monospace;
13 }
14

```

## src/index.tsx

```

1 import React from "react";
2 import ReactDOM from "react-dom/client";
3 import './index.css';
4 import App from './App';
5 import reportWebVitals from './reportWebVitals';
6 import Providers from './providers';
7 import { BrowserRouter } from "react-router-dom";
8
9 const root = ReactDOM.createRoot(
10   document.getElementById("root") as HTMLElement
11 );
12
13 root.render(
14   <React.StrictMode>
15     <BrowserRouter>
16       <Providers>
17         <App />
18       </Providers>
19     </BrowserRouter>
20   </React.StrictMode>
21 );
22
23 // If you want to start measuring performance in your app, pass a function
24 // to log results (for example: reportWebVitals(console.log))
25 // or a number of milliseconds (e.g. reportWebVitals(5000))
26 reportWebVitals();

```

1312  
 b o c

TAV Web

src/logo.svg

```

1 <img alt="Logo" data-bbox="145 130 885 415"/>
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26

```

src/react-app-env.d.ts

```

1 // reference types/react-scripts"
2

```

src/reportWebVitals.ts

```

1 import { ReportHandler } from 'web-vitals';
2
3 const reportWebVitals = (onPerfEntry?: ReportHandler) => {
4   if (onPerfEntry && onPerfEntry instanceof Function) {
5     import('web-vitals').then(({ getCLS, getFID, getFCP, getLCP, getTTFB }) => {
6       getCLS(onPerfEntry);
7       getFID(onPerfEntry);
8       getFCP(onPerfEntry);
9       getLCP(onPerfEntry);
10      getTTFB(onPerfEntry);
11    });
12   }
13 };
14
15 export default reportWebVitals;
16

```

130  
 9  
 B  
 \*



TAV Web

**src/setupTests.ts**

```
1 // jest-dom adds custom jest matchers for asserting on DOM nodes.
2 // allows you to do things like:
3 // expect(element).toHaveTextContent(/react/i)
4 // learn more: https://github.com/testing-library/jest-dom
5 import '@testing-library/jest-dom';
6
```

**.gitignore**

```
1 # See https://help.github.com/articles/ignoring-files/ for more about ignoring files.
2
3 # dependencies
4 /node_modules
5 /.pnpm
6 .pnpm.js
7
8 # testing
9 /coverage
10
11 # production
12 /build
13
14 # misc
15 .DS_Store
16 .env.local
17 .env.development.local
18 .env.test.local
19 .env.production.local
20
21 rpm-debug.log*
22 yarn-debug.log*
23 yarn-error.log*
24
```

133

~~133~~

133

TAV Web

## package.json

```

1  {
2    "name": "react-bootstrap",
3    "version": "1.4.0",
4    "private": true,
5    "dependencies": {
6      "classnames/react": "11.1.0",
7      "classnames/styled": "11.1.0",
8      "material-ui/core": "4.11.4",
9      "material-ui/icons": "4.11.3",
10     "mui-material": "1.10.17",
11     "testing-library/jest-dom": "7.16.0",
12     "testing-library/react": "11.6.0",
13     "testing-library/user-event": "13.5.0",
14     "types/jest": "27.5.2",
15     "types/node": "16.16.0",
16     "types/react": "18.0.26",
17     "types/react-dom": "18.0.0",
18     "sass": "1.2.0",
19     "bootstrap": "5.1.3",
20     "dayjs": "1.11.0",
21     "jwt-decode": "3.1.0",
22     "react": "18.2.0",
23     "react-bootstrap": "1.7.0",
24     "react-dom": "18.2.0",
25     "react-router-dom": "6.4.0",
26     "react-scripts": "5.0.1",
27     "styled-components": "5.3.0",
28     "typescript": "4.9.3",
29     "web-itals": "2.1.0"
30   },
31   "scripts": {
32     "start": "react-scripts start",
33     "build": "react-scripts build",
34     "test": "react-scripts test",
35     "eject": "react-scripts eject"
36   },
37   "eslintConfig": {
38     "extends": [
39       "react-app",
40       "react-app-strict"
41     ]
42   },
43   "browserslist": {
44     "production": [
45       ">0.2%",
46       "not dead",
47       "not ie < 11"
48     ],
49     "development": [
50       "last 1 chrome version",
51       "last 1 firefox version",
52       "last 1 safari version"
53     ]
54   },
55   "devDependencies": {
56     "types/styled-components": "5.1.26"
57   }
58 }

```

1361

9

9



