

## Problema 1 - Problemas de responsividade

### Descrição:

Diversas páginas da plataforma apresentaram falhas de responsividade ao simular uso em dispositivos móveis. Elementos ficam sobrepostos ou parcialmente ocultos, comprometendo a navegação. Dois exemplos disso são as informações contidas em alguns cards no dashboard e as listas de sessões na página de sessões em “Minha conta”.

### **Páginas afetadas:**

- Aba de **Sessões** - lista de dispositivos parcialmente oculta.
- Tela de **Agenda** - botões para mudar o layout da agenda parcialmente ocultos.
- Tela de **Cliente** - tabela com as informações de endereço fica parcialmente oculta.
- Tela de **Recursos** - ao clicar em filtros adicionais, o botão exibir fica quase oculto.
- Tela de **Contratos** - botões ocultos e tabela com colunas sobrepostas.
- Abas de **Recurso** e **Cliente e solicitante** - ambas com tabela parcialmente oculta.
- Tela de **Atividades agendadas** - header quebrado.

### Passos para reproduzir:

1. Acessar a plataforma Tiflux.
2. Reduzir a largura do navegador até simular um dispositivo móvel (usar o toggle device toolbar).
3. Abrir as telas listadas acima.

Obs: Os testes foram realizados usando a dimensão **iPhone 14 Pro Max** e **Samsung Galaxy A51/71** no navegador **Google Chrome**.

### Evidência:

Os vídeos abaixo mostram que ao reduzir a largura da tela, elementos ficam sobrepostos, menus cortados ou botões escondidos:

- [Aba de sessões](#)
- [Tela de Agenda](#)
- [Tela de Cliente](#)
- [Tela de Recursos](#)
- [Tela de Contratos](#)
- [Abas de Recurso e Cliente e solicitante](#)

Demais evidências em imagens:

- [Mensagem de boas vindas](#)
- [Card dashboard](#)
- [Atividades agendadas](#)
- [Apontamentos - Modelos](#)

**Severidade:**

**Média** / embora não bloqueie o uso da aplicação, pode prejudicar significativamente a experiência do usuário em dispositivos móveis.

**Impacto:**

Usuários acessando pelo celular podem ter dificuldade de usar algumas funcionalidades e podem abandonar o fluxo por falta de legibilidade ou botões escondidos.

**Sugestão de melhoria:**

Seria interessante implementar ajustes de CSS e media queries para garantir que todos os componentes se adaptem corretamente a resoluções menores.

**Problema 2 - Caractere especial no nome de usuário****Descrição:**

O campo “Nome” nas configurações de usuário permite a inserção de caracteres especiais e números, mesmo sendo um campo destinado a nomes reais. Isso pode gerar inconsistência nos dados e problemas em integrações ou relatórios.

**Passos para reproduzir:**

1. Acessar a tela Minha conta.
2. No campo “Nome”, digitar caracteres especiais ou números, por exemplo: @matheus!2.
3. Clicar em “Salvar”.
4. Observe que o sistema aceita os caracteres e salva o registro sem alertas.

Obs: Os testes foram realizados usando o navegador **Google Chrome**.

**Evidência:**

- [Caractere especial no nome de usuário](#)
- [Injeção de comandos sql](#)

**Severidade:**

**Média** / não bloqueia o fluxo do usuário, mas gera dados inconsistentes que podem afetar relatórios ou integrações futuras.

**Impacto:**

Usuários podem inserir dados inválidos nos campos de nome, prejudicando a qualidade dos dados armazenados, causando possíveis erros na hora de sincronizar os dados ou enviar relatórios, além de poder quebrar queries SQL se não forem tratados corretamente, gerando vulnerabilidades como SQL Injection.

### **Sugestão de melhoria:**

Adicionar validação no front-end e back-end para permitir apenas letras, espaços e caracteres normalmente usados em nomes. Além disso, mostrar mensagem de erro clara caso o usuário tente inserir caracteres inválidos.

## **Problema 3 - Campo CEP permite mais dígitos que o formato padrão**

### **Descrição:**

O campo “CEP” no cadastro de endereço de cliente aplica máscara automática no formato #####-###, mas permite inserir mais de 8 dígitos sem exibir mensagem de erro.

### **Passos para reproduzir:**

1. Acessar no menu **Clientes > Novo Cliente**.
2. No campo CEP, inserir vários números (exemplo: 88888888888888).
3. Clicar em “Salvar” no caso de edição ou se estiver criando um novo cliente, clicar em “Próximo” e continuar o fluxo.
4. Observar que a máscara é aplicada, mas o campo aceita valores além do limite padrão.

Obs: Os testes foram realizados usando o navegador **Google Chrome**.

### **Evidência:**

- [Campo CEP permite mais dígitos que o formato padrão](#)

### **Severidade:**

**Baixa** / o sistema mantém a máscara e aceita apenas números evitando caracteres especiais, mas não impede a digitação de CEPs extremamente longos, o que pode causar inconsistência de dados.

### **Impacto:**

CEP inválido pode gerar erro em integrações externas e prejudicar a confiabilidade dos dados.

### **Sugestão de melhoria:**

Implementar validação de comprimento máximo (8 dígitos) e impedir digitação além do limite.

## Observação técnica - Erro "Uncaught (in promise)" no console

**ATENÇÃO:** Percebi que esse é um padrão utilizado para validar os campos obrigatórios, já que as mensagens de erro e marcação dos campos inválidos são feitas a partir dessa exceção. Mas vale a pena discutir se essa é a melhor opção de tratamento, pois o uso de exceções não tratadas pode dificultar a identificação de erros reais no futuro.

### Descrição:

Ao tentar salvar informações do usuário sem preencher um campo obrigatório, o console do navegador mostra um erro JavaScript "Uncaught (in promise)", embora o fluxo funcione corretamente e o campo obrigatório seja destacado em vermelho.

### Passos para reproduzir:

1. Acessar a tela de edição de usuário.
2. Deixar um **campo obrigatório vazio** (ex: "Nome" ou "Email").
3. Clicar em "Salvar".
4. Abrir o console do navegador (F12) e observar o erro.

Obs: Os testes foram realizados usando o navegador **Google Chrome**.

### Evidência:

- [Erro no console](#)

### Severidade:

**Baixa** / o fluxo do usuário não é interrompido e a validação visual funciona da maneira correta.

### Impacto:

Não afeta diretamente o usuário, mas demonstra que o código poderia ser melhor estruturado para evitar erros no console tratando exceções.

### Sugestão de melhoria:

Tratar a promise no JavaScript corretamente para que não apareça erro no console.