
RELATÓRIO DE AVALIAÇÃO - BATTLESIDE

Denis Crispe Silva¹, Fernando Godoy de Oliveira¹

¹ Centro Federal de Educação Tecnológica Celso Suckow da Fonseca (CEFET/RJ)

Professor Diogo Mendonça
Teste de Software

1 OBJETIVO DO TRABALHO

Avaliar os requisitos não-funcionais (performance, segurança e usabilidade) da aplicação Battleside.

2 RESUMO DOS PROBLEMAS ENCONTRADOS

Nesta seção é apresentado um resumo dos problemas encontrados durante a avaliação feita neste trabalho que são posteriormente detalhados e evidenciados nas demais seções do trabalho.

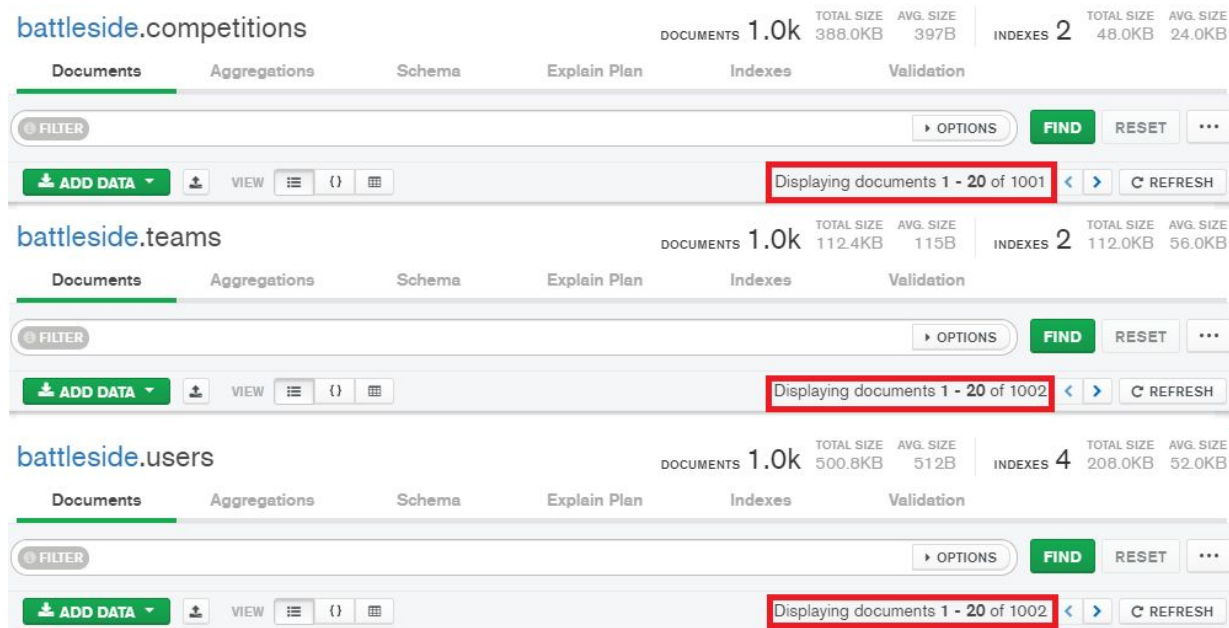
- **PERFORMANCE (TOTAL DE 1 PROBLEMA)**
 - Baixa pontuação no Lighthouse para Mobile
- **SEGURANÇA (TOTAL DE 5 PROBLEMAS)**
 - AccessToken não expira mesmo após o logout (o mesmo token consegue reutilizado);
 - Senha fraca, não é obrigatório o uso de caracteres especiais, números, letras maiúsculas e minúsculas;
 - Tráfego de dados sensíveis sem o uso de criptografia;
 - É possível alterar uma competição utilizando o token de acesso de um usuário diferente do criador/administrador da competição.
 - Vulnerabilidade no package ini, dependência do cypress e react-scripts
- **USABILIDADE (TOTAL DE 16 PROBLEMAS, 30 OCORRÊNCIAS INDIVIDUAIS E 1 OCORRÊNCIA PERSISTENTE NO SISTEMA INTEIRO)**
 - Heurística 1 - 4 ocorrências:
 - O sistema não possui algum tipo de aviso caso esteja carregando dados ou se os dados já foram carregados;
 - Não existe uma indicação de carregamento da página, como um spinner ou uma barra de carregamento;
 - Heurística 2 - 2 ocorrências:
 - Uso de jargões que podem não ser conhecidos pelo usuário;
 - Presença de botões com conveniências que podem confundir o usuário;
 - Heurística 3 - 3 ocorrências:
 - Falta de opções para dar autonomia ao usuário em campos de texto;
 - Heurística 4 - 4 ocorrências:
 - Falta de padronização de leiaute em funcionalidades iguais de itens diferentes;
 - Leiautes iguais em locais próximos para funções completamente distintas;

- Falta de padrão em datas: datas que devem obrigatoriamente vir depois de outras cronologicamente, se apresentam antes na tela;
- Heurística 5 - 7 ocorrências:
 - Campos de data propensos a erro ao permitir selecionar datas impossíveis ou inválidas para a função;
 - Botões com funcionalidades que podem causar um retrabalho ao usuário caso clicados erroneamente não possuem botão de confirmação;
- Heurística 6 - 2 ocorrências:
 - Dados não permanecem em caso de troca de página, sem confirmação de reenvio de formulário;
- Heurística 7:
 - O sistema é simples e não precisa de atalhos para ferramentas, porém atalhos básicos são presentes no sistema;
- Heurística 8 - 4 ocorrências e um problema persistente no sistema todo (campos de texto):
 - Botões com pouco contraste com o fundo que estão inseridos, se tornando difíceis de ver;
 - Fontes praticamente invisíveis por falta de contraste com o fundo;
 - Campos de texto sem tamanho flexível e sem maneiras de visualizar todo o campo, o que prejudica o usuário em campos que tendem a receber um alto número de dados;
- Heurística 9 - 3 ocorrências:
 - Mensagens de erro pouco informativas sobre o motivo do problema;
- Heurística 10 - 1 ocorrência:
 - Funções do sistema são simples, não necessitando de documentação adicional, porém, ocorrência de documentação ambígua em botão.

3 PERFORMANCE

3.1 COMPORTAMENTO DO SISTEMA COM MAIS DE 1000 REGISTROS

O sistema foi testado com todas as suas *collections* com pelo menos 1001 registros. Para esta tarefa, foi criado um script em Python que varia informações e gera a massa de dados no formato de arquivos JSON que posteriormente são importados no MongoDB. Os prints nas imagens abaixo, mostram a quantidade de registros em cada *collection*.



Após a criação e importação da massa de dados, o sistema foi exercitado e diversas funcionalidades foram testadas. Inserções, buscas, exclusões e atualizações foram realizadas. O sistema não apresentou nenhuma lentidão perceptível em nenhuma funcionalidade, o funcionamento do sistema se manteve dentro da normalidade.

Este teste com mais de 1000 registros não foi suficiente para gerar algum tipo de anomalia na execução da aplicação. É possível concluir que o sistema suporta 1000 registros em todas as suas *collections*.

3.2 PERFORMANCE DE CARGA - LIGHTHOUSE

O plugin Lighthouse audita o sistema de forma automatizada e no final apresenta um relatório acerca da qualidade das páginas *web*, são observados fatores como desempenho, acessibilidade e boas práticas. Neste trabalho foi utilizado apenas o indicador de performance(desempenho).

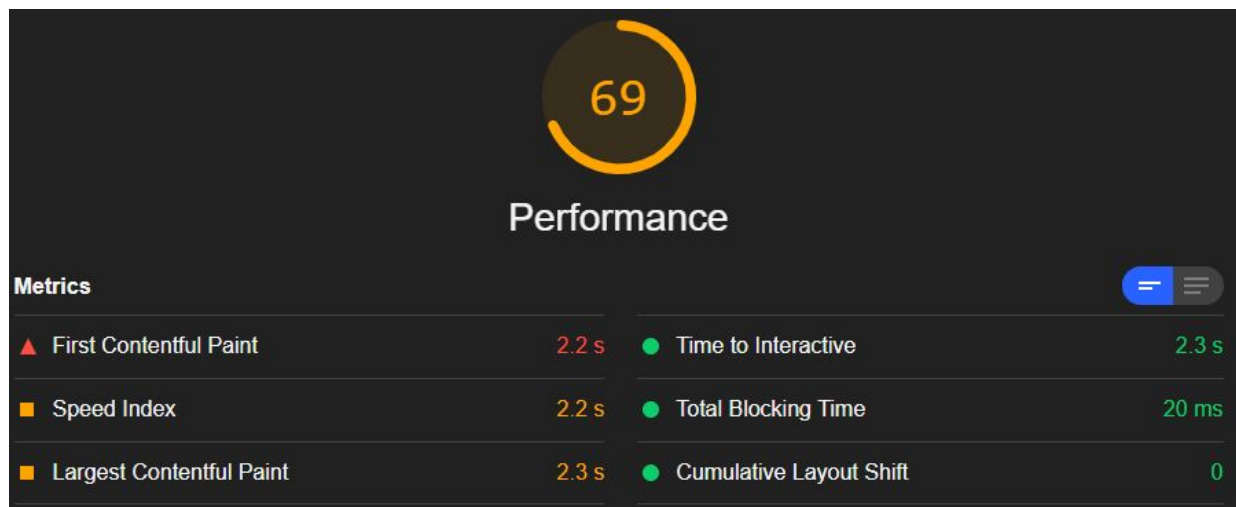
Para a composição da performance são observados algumas métricas, como:

- a quantidade de tempo que leva para a página se tornar totalmente interativa;
- a velocidade com que o conteúdo de uma página é preenchido visivelmente;
- o tempo que leva até o momento em que o primeiro texto ou imagem é apresentada;
- quantidade total de tempo que uma página fica bloqueada para responder à entrada do usuário;

Como resultado tem uma pontuação final, a contabilização de tempo das métricas e sugestões de melhorias que influenciam diretamente na performance.

3.2.1 DESKTOP

O sistema obteve a pontuação 69 no plugin Lighthouse para Desktop, como mostrado na imagem abaixo:



Levando 2.3 segundos até que a página esteja totalmente interativa. Com 2.2 segundos o primeiro conteúdo é apresentado na tela e também preenchido visualmente. Com 2.3 segundos o maior conteúdo é totalmente carregado. O tempo total de bloqueio foi de 20 milissegundos.

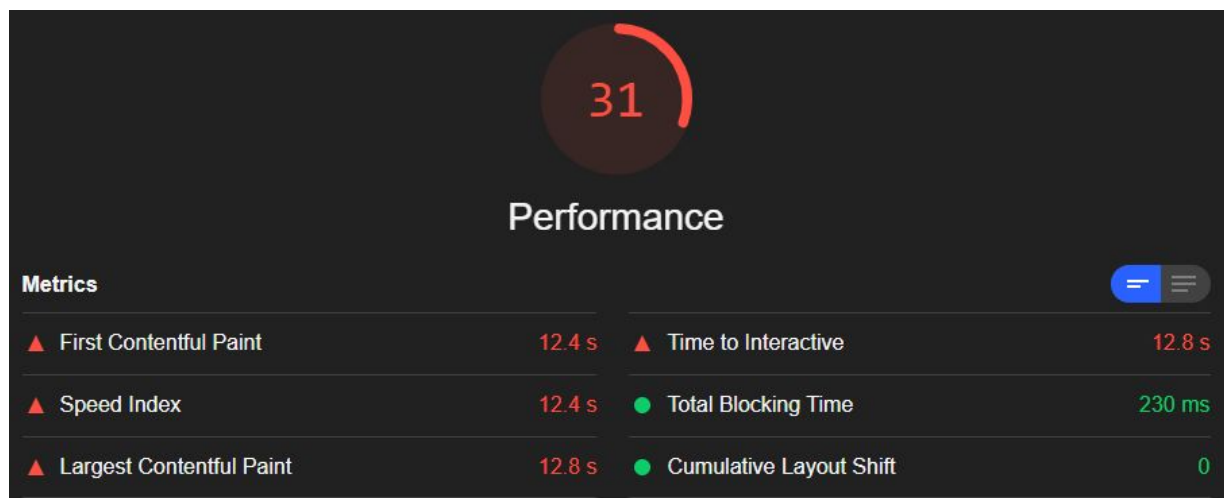
O tempo de carregamento de 2 segundos é um bom tempo, não é instantâneo mas a percepção de tempo como usuário é igual ao carregamento de outros sites. A transição de telas e a execução de comandos tem um tempo de resposta dentro do habitual, o uso da aplicação não apresentou nenhum tipo de gargalo ou travamento.

Foram dadas 3 sugestões de melhoria que afetam diretamente na pontuação da performance, foram elas:

- Remover o JavaScript não utilizado;
- Ativar compressão de texto
- Minificar o JavaScript;

3.2.2 MOBILE

O sistema obteve a pontuação 31 no plugin Lighthouse para Mobile, como mostrado na imagem abaixo:



A aplicação apresentou uma performance bem inferior para a plataforma mobile. As métricas apresentaram um tempo de 10 segundos maior em relação aos tempos para desktop. A página levou 12.8 segundos para ficar totalmente interativa. Levou 12.4 segundos para que o primeiro elemento (texto ou imagem) fosse apresentado e totalmente preenchido visualmente. O tempo total de bloqueio foi de 230 milissegundos.

Por questões técnicas e por só estar rodando localmente no computador, não foi possível testar de fato o sistema em um dispositivo mobile. Porém, analisando levando em consideração o tempo dado pelo Lighthouse,

12 segundos para o carregamento da página é um alto tempo de espera. O tempo destoa muito em comparação ao mesmo para desktop,

As sugestões de melhorias foram as mesmas que para desktop, também adicionando outras. Uma curiosidade é que apenas duas sugestões (que também foram sugeridas para desktop), representam quase 5 segundos dos tempos totais, como mostrado na figura abaixo.

Opportunities — These suggestions can help your page load faster. They don't [directly affect](#) the Performance score.

Opportunity	Estimated Savings
▲ Remove unused JavaScript	<div></div> 3.15 s ▼
▲ Minify JavaScript	<div></div> 1.8 s ▼

Fora as sugestões que também foram feitas para desktop, também foi sugerido:

- Remover o CSS não utilizado;
- Eliminar recursos de bloqueio de renderização;
- Pré-conectar às origens necessárias (adição de “preconnect” ou “dns-prefetch”)

4 SEGURANÇA

4.1 BROKEN AUTHENTICATION

4.1.1 EXPIRAÇÃO DO ACCESS_TOKEN

No *local storage* do navegador é possível extrair o *accessToken* (como apresentando na figura abaixo), token este que permite acesso a aplicação, o correto é este *token* expirar após um período de inatividade ou o *logout* do usuário. Porém em um simples teste foi possível verificar que o *token* não está sendo finalizado, as informações foram copiadas e reutilizadas para login em outro navegador após o *logout* do primeiro navegador onde o *token* era armazenado.

The screenshot shows the Chrome DevTools Storage tab. On the left, the 'Storage' section is expanded, showing 'Local Storage' for the URL 'http://localhost:3000'. The main pane displays a table of stored items:

Key	Value
accessToken	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXLTJ5
state	{"user":{"name":"Denis","token":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXLTJ5","id":1}}

Basta no console do navegador colocar a chave *accessToken* com o valor do token guardado e reutilizá-lo. Este mesmo dado trafega no *header* das requisições, logo caso alguma requisição seja interceptada por um usuário malicioso, pode ser usada para fazer autenticação na aplicação sem necessitar de senha. A imagem abaixo mostra como é feito para alterar o valor de uma chave no *local storage*.

```
> localStorage.setItem('accessToken', 'Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjVmZWQyYTJhZjdlOiI6IjodHRWoi8ybG91YXxob3N0OjMwMDA1LCJpc3MiOiIodHRWoi
```

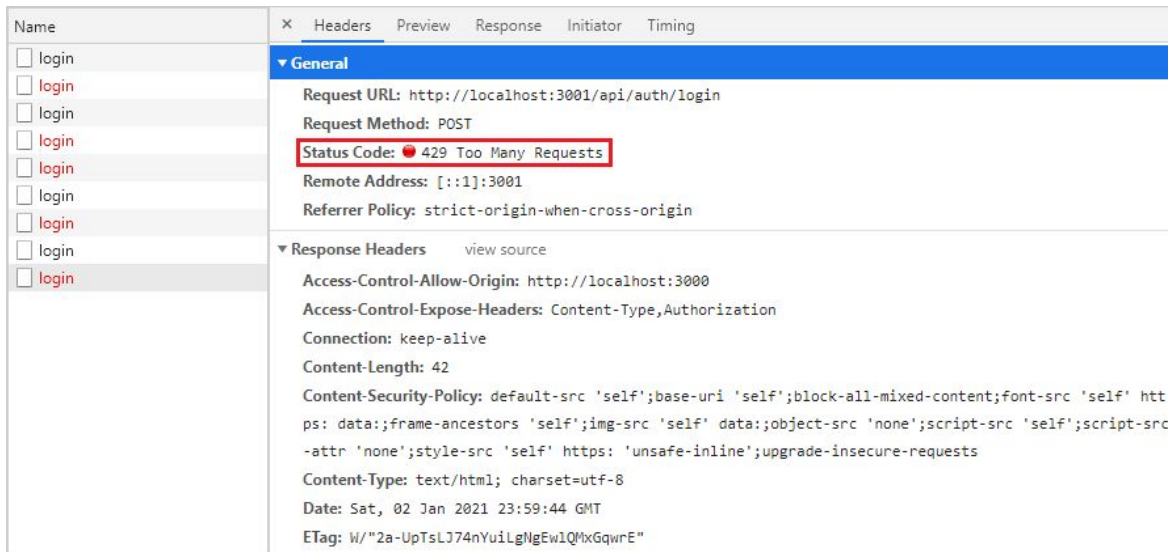
4.1.2 SENHA FRACA

Durante o cadastro do usuário é possível cadastrar uma senha fraca, o sistema não requer o uso de caracteres especiais, letras minúsculas, letras maiúsculas e números. A única verificação feita é em relação ao

tamanho da senha, que deve conter no mínimo 8 caracteres. Uma senha fraca facilita a quebra de senha em métodos como a força bruta.

4.1.3 FORÇA BRUTA

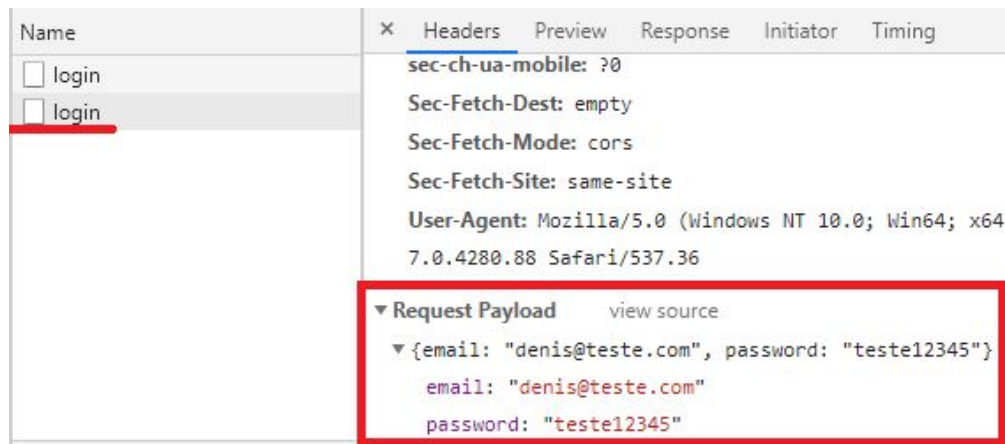
O sistema possui uma funcionalidade que reconhece múltiplas requisições vindas do mesmo cliente, dificultando o uso de força bruta com apenas um computador. É importante este tipo de verificação para atrapalhar o uso de ferramentas automatizadas que fazem diversas requisições para o servidor. A imagem abaixo mostra o comportamento do servidor que após 5 requisições seguidas para o serviço de login, responde com código 429 “Too Many Requests” e bloqueia temporariamente o cliente.



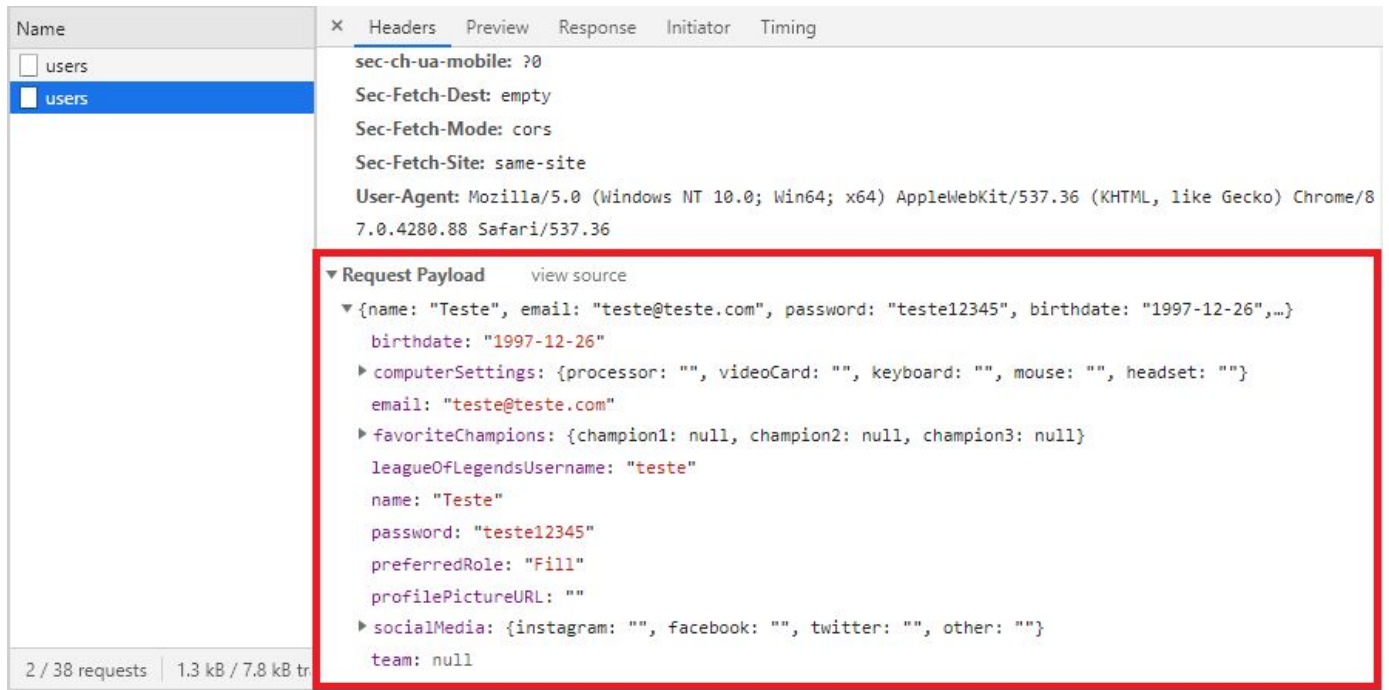
4.2 SENSITIVE DATA EXPOSURE

Alguns sistemas podem armazenar dados sensíveis dos usuários, como: senhas, número de cartão de crédito, informações pessoais e etc. É necessário proteger este tipo de dados para não expor dados pessoais dos usuários. Os dados devem ser protegidos tanto quando em tráfego (request e response) ou armazenados (banco de dados), sempre de forma criptografada para impedir que mesmo caso seja interceptado, não possa ser lido.

A senha do usuário é um dado sensível da aplicação que é trafegado em texto não criptografado. No momento do login a requisição que é feita para o backend, envia os dados do usuário sem qualquer criptografia. A imagem abaixo evidencia a forma com que a requisição é feita.



O mesmo ocorre no cadastro, a senha também é enviada sem criptografia, como mostrado na imagem abaixo.



Um ponto positivo é que apesar de ser trafegada em texto livre sem criptografia, quando armazenada no banco de dados, a senha é guardada de forma criptografada. Esta é a maneira de tratar dados sensíveis, sempre armazenados de forma criptografada. A imagem abaixo evidencia como é o formato de um registro da *collection* user (usuário).

```

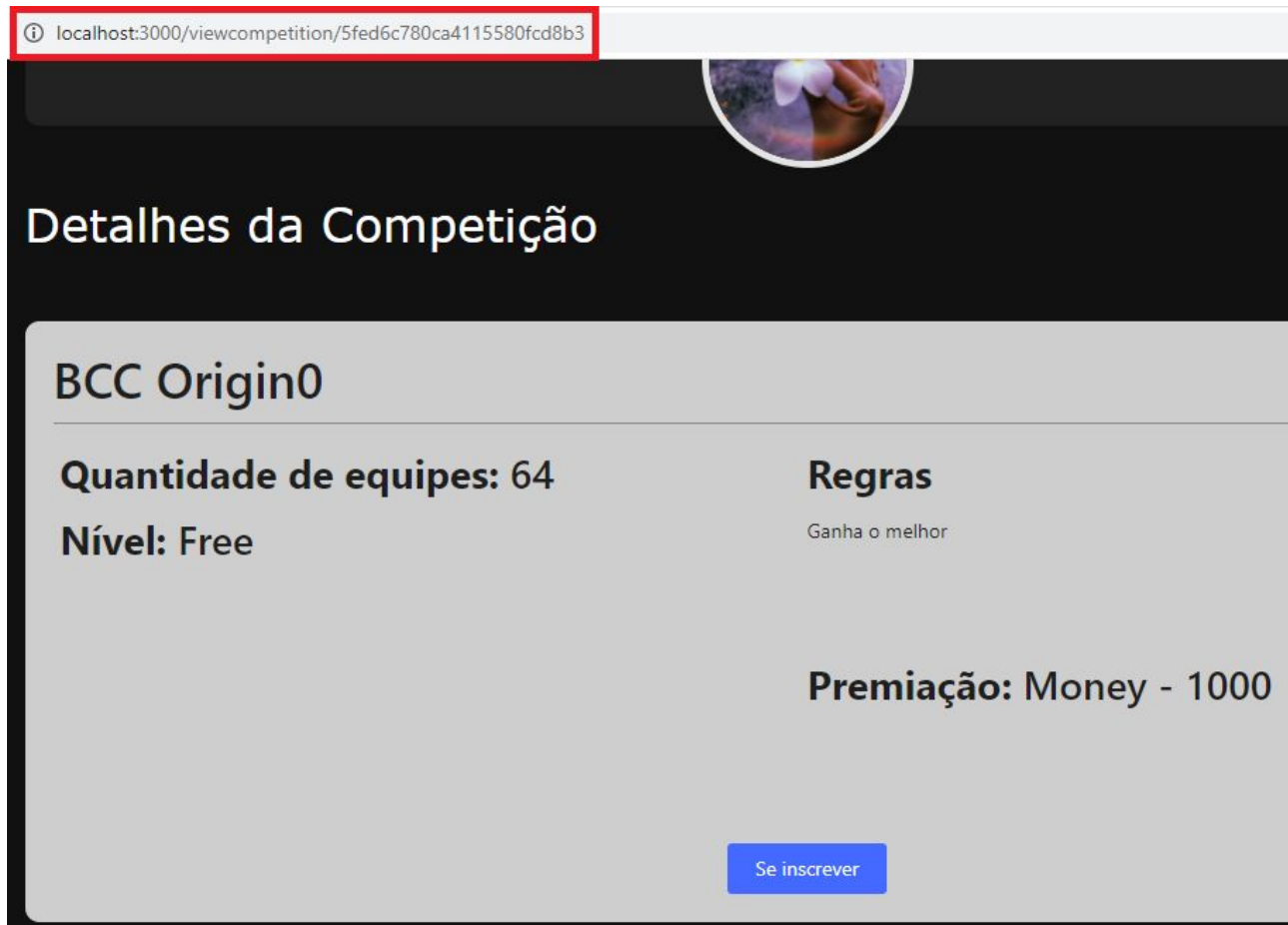
_id: ObjectId("5ff0d2d2f7ae962bb820d7b1")
preferredRole: "Fill"
> favoriteChampions: Object
> socialMedia: Object
> computerSettings: Object
team: null
leagueOfLegendsUsername: "testeDenis"
profilePictureURL: ""
birthdate: 1997-12-26T02:00:00.000+00:00
password: "$2b$10$b1SvvtgazL1FNSzX7i8ee5pAMhZ9878b6VDVd3GjDvx4.owleNqG"
email: "denis@teste.com"
name: "Denis"
__v: 0

```

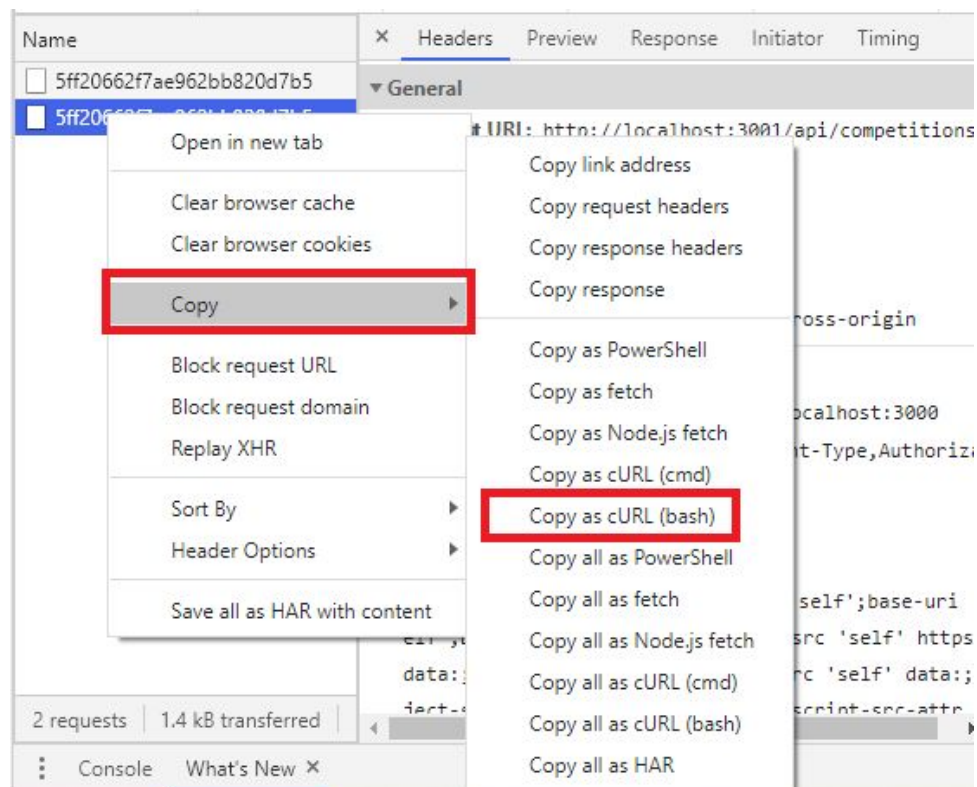
4.3 BROKEN ACCESS CONTROL

O sistema possui uma falha de segurança no controle de acesso, utilizando o token de um usuário é possível editar a competição de outro usuário. Os prints presentes nas imagens a seguir mostram como é feito este processo.

Primeiramente é necessário ter o id da competição alvo da edição, para isso basta entrar na página de visualização da competição, a mesma página que possibilita a inscrição naquela competição. Como mostrado na imagem abaixo, o id da competição fica exposto na URL quando a página é acessada.

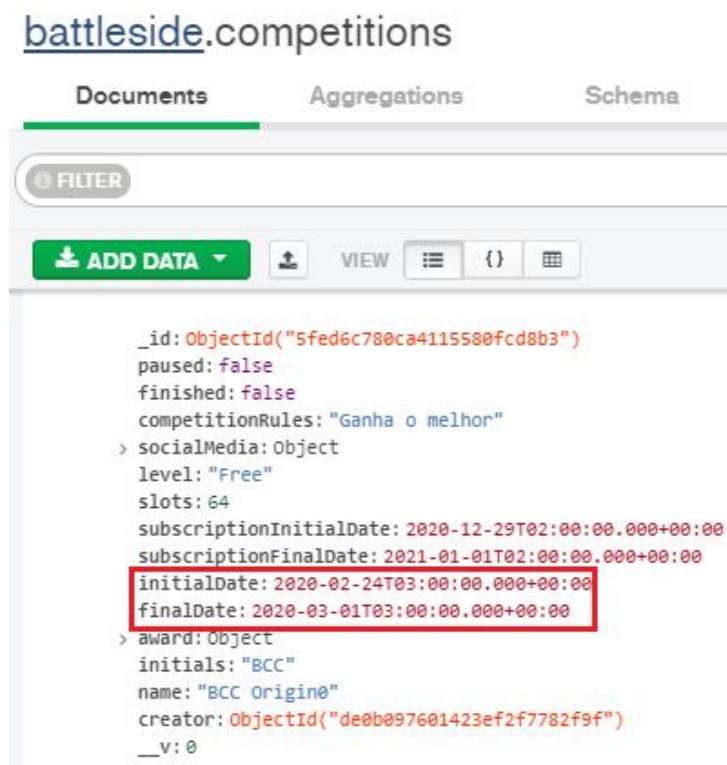
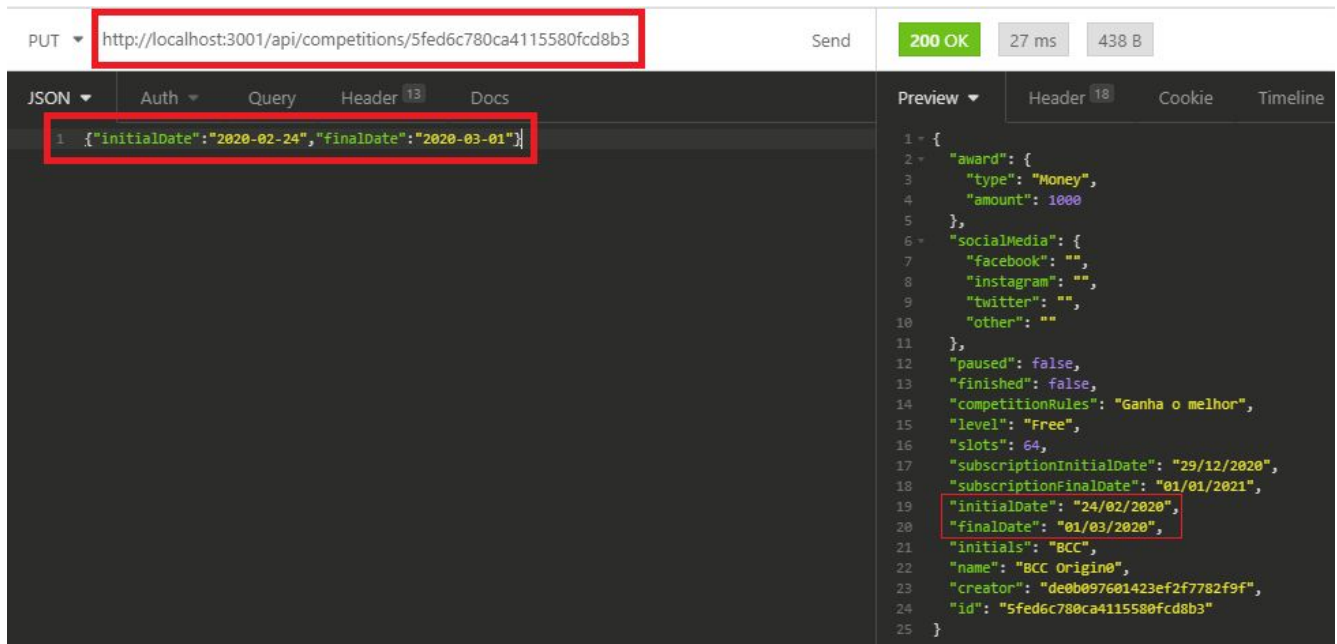


Após obter este dado, basta o usuário criar uma competição, fazer uma edição e copiar a requisição que foi feita desta edição.



Após copiar a requisição, basta colá-la em um programa que faça requisições REST (no teste foi utilizada a ferramenta Insonmia). Da requisição é necessário editar o id da competição na URL e as

informações da competição que serão alteradas (no teste foi alterada a data de início e fim da competição). As imagens abaixo mostram que a requisição foi aceita e alterações realizadas.



4.4 USING COMPONENTS WITH KNOWN VULNERABILITIES

Utilizando o comando “yarn audit”, é possível obter um relatório acerca das vulnerabilidades presentes nas bibliotecas e componentes utilizados na aplicação. Foram encontradas 2 vulnerabilidades de severidade baixa no *frontend* e nenhuma no *backend*, as 2 encontradas são na verdade o mesmo problema porém presente em 2 dependências distintas. A imagem abaixo mostra a execução do comando audit.

```
Denis@DESKTOP-8NU03IH MINGW64 /c/opt/Projects/trabalho-integrado-2020-1-grupo-1/
frontend (main)
$ yarn audit
yarn audit v1.22.10
```

low	Prototype Pollution
Package	ini
Patched in	>1.3.6
Dependency of	cypress
Path	cypress > is-installed-globally > global-dirs > ini
More info	https://www.npmjs.com/advisories/1589

low	Prototype Pollution
Package	ini
Patched in	>1.3.6
Dependency of	react-scripts
Path	react-scripts > react-dev-utils > global-modules > global-prefix > ini
More info	https://www.npmjs.com/advisories/1589

```
2 vulnerabilities found - Packages audited: 2160
Severity: 2 Low
Done in 7.89s.
```

A vulnerabilidade se trata de “Poluição de Protótipo”, segundo o próprio site do NPM, se um invasor enviar um arquivo INI malicioso para um aplicativo que o analisa com o “*ini.parse*”, ele poluirá o protótipo do aplicativo. Isso também pode ser explorado ainda mais, dependendo do contexto, Para resolver esta vulnerabilidade basta atualizar o package ini para versão 1.3.6 ou superior.

5 USABILIDADE

Nesta seção será analisada a questão da usabilidade do sistema, de acordo com os 10 princípios gerais de Jakob Nielsen para design de interação. O sistema inteiro foi analisado e serão elucidados os pontos positivos e negativos do mesmo em relação às heurísticas apresentadas nas subseções a seguir.

5.1 VISIBILIDADE DO STATUS DO SISTEMA

Esta heurística mede a capacidade do sistema sempre manter o usuário informado do que está acontecendo. No sistema, foi observado que em determinadas telas era impossível saber se o sistema estava carregando os dados, ou se os dados eram inexistentes, por exemplo.

Na tela demonstrada na evidência a seguir, na área de visualizar as competições, foram definidos filtros para a busca, porém, não é informado ao usuário se o sistema ainda está carregando as competições, ou se não existem competições, a tela apenas permanece a mesma de antes dos filtros serem selecionados.

Competições

Use os filtros!

Platina



Dinheiro



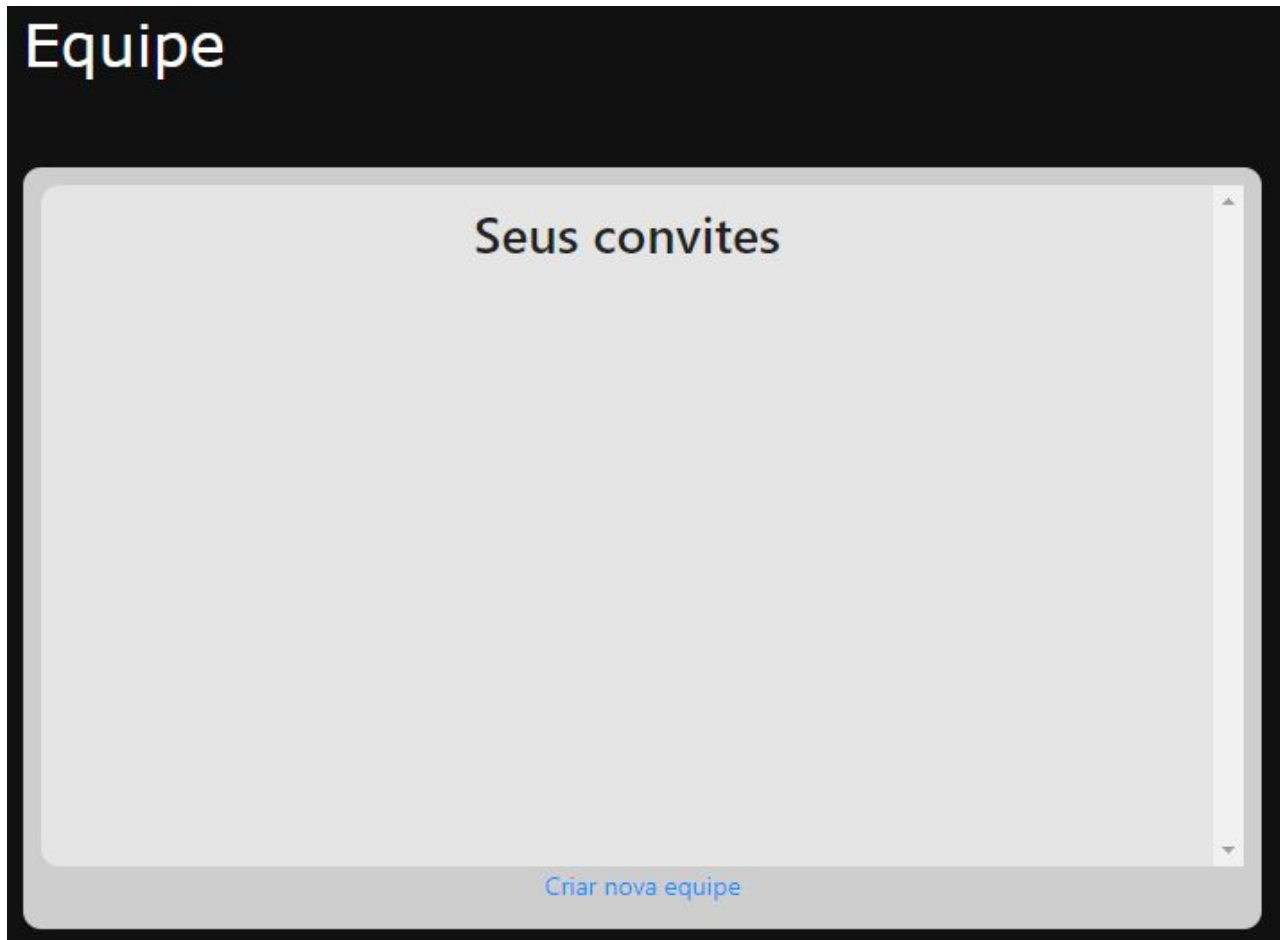
Na aba “Sua competição” também não existem mensagens informativas de que não existem prints recebidos caso a lista esteja vazia, para não dar a impressão ao usuário de que o sistema ainda está carregando os prints.

Sua competição

testeteste

Prints recebidos

O mesmo ocorre na tela de “Equipe”, na área de “Seus convites”.

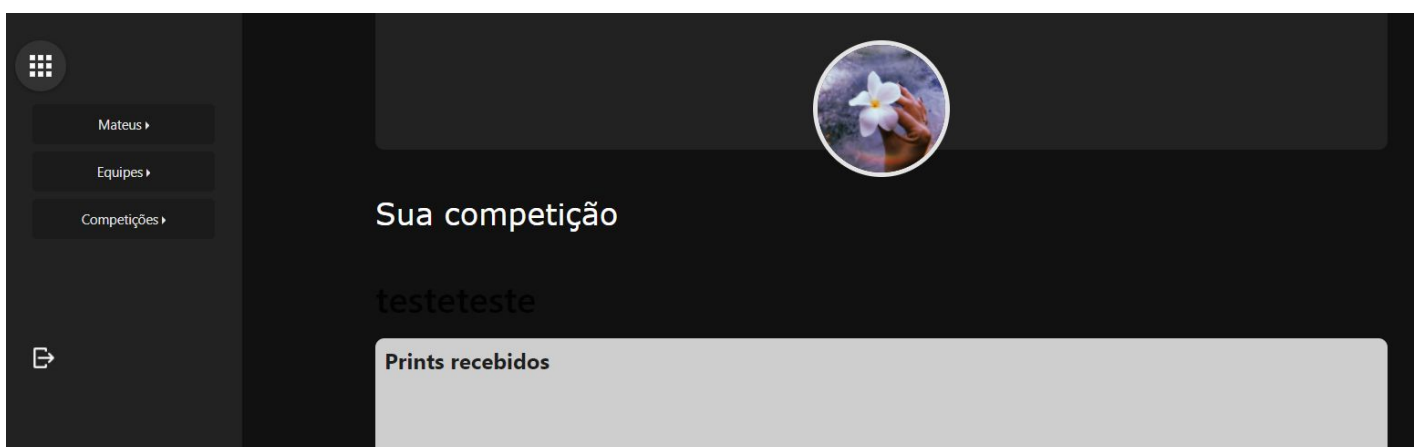


Ao trocar de telas, também foi observada a falta de uma indicação de carregamento da página, como um spinner ou uma barra de carregamento indicando o progresso.

5.2 CORRESPONDÊNCIA ENTRE O SISTEMA E O MUNDO REAL

De acordo com esta heurística, o sistema deverá se comunicar e informar o usuário com a linguagem do último, ou seja, com termos, conceitos e convenções conhecidas pelo mesmo, tanto em frases quanto em ícones e imagens do sistema.

Na evidência a seguir, pode ser observado o termo “*Print*”, o que significa uma captura de tela, porém, para usuários mais leigos, isso pode não remeter imediatamente a este conceito, pelo motivo do mesmo não estar familiarizado com esse tipo de jargão.



Além disso, o botão de sair do sistema, demonstrado pelo ícone com a seta na imagem acima, pode levar o usuário a pensar que a funcionalidade deste botão é de sair da página e retornar a alguma anterior.

5.3 CONTROLE DO USUÁRIO E LIBERDADE

É denotado por esta heurística que o usuário pode cometer erros, porém, ele não deve ser punido firmemente por isso, o sistema deve ter um tipo de “válvula de escape” para sair da desfazer ou refazer um processo, sem a necessidade de um processo custoso e demorado para o realizar. Estes tipos de saída dão ao usuário mais autonomia e liberdade para realizar suas ações, por estarem mais no controle do sistema.

No sistema, em telas que contém formulários, não existem muitas opções que dão maior controle ao usuário.

Isso ocorre em todas as telas de formulário. Tendo a evidência da tela de criar competição, não existe a opção de limpar todos os campos, o que pode ser útil em casos que o usuário preencheu todos ou múltiplos dados de maneira equivocada. Também não existe a opção de limpar certos campos em específico. Em áreas de texto que podem receber um grande número de dados, como no exemplo do campo de “Regras”, pode ser uma funcionalidade útil, pois pode se tornar custoso para um usuário apagar tudo, visto que, somado a isso, este campo em específico não mostra todo o conteúdo inserido nele a todo o momento, logo, o usuário teria que marcar todo o campo com algum atalho do teclado para apagá-lo por completo.

[illegible]

Isso pode ser observado também na tela de edição de perfil:

Editar perfil

Nome *

Mateus Rodrigues

Username do League of Legends *

UnoDeFirmaTunado

Email *

mateus@gmail.com

Role preferida *

Top

Data de nascimento *

09/07/2021



Campeão 1 *



Campeão 2 *



Campeão 3 *



Processador

ddsadad123

Placa de Video

dasda1231

Teclado

adsad

Mouse

1313

Headset

dasdad

Facebook

facebook.com/

Instagram

instagram.com/

Twitter

twitter.com/

Outro link

Outras

Salvar

Ocorre também na tela de nova equipe:

Nova equipe

Nome da equipe *

Sigla da equipe *

Criar

5.4 CONSISTÊNCIA E PADRÕES

Nesta heurística é dito que o usuário não deve indagar se ações, palavras e situações diferentes se referem às mesmas coisas. O sistema deve ter consistência em suas funcionalidades, ações com funcionalidades iguais para itens diferentes precisam seguir um certo padrão.

No sistema, alguns botões possuem a mesma cor para atividades completamente distintas.

A evidência a seguir na tela “Minha Equipe” quebra o padrão, no sentido que os botões de visualizar e sair da equipe apresentam o mesmo leiaute, mesmo esquema de cores e se encontram em lugares bem semelhantes da tela, porém, possuem funcionalidades bem distintas.

Equipe



teste teste

tst

Visualizar equipe

Sair da equipe

Esta inconsistência também pode ser visualizada na tela “Sua competição”, com os botões “Editar data e horário” e “Deletar competição”.

Sua competição

testeteste

Prints recebidos

Ações do administrador

Editar data e horário

Deletar competição

Além disso, alguns botões com o mesmo tipo de funcionalidade para itens diferentes apresentam layouts diferentes, o que pode ser observado nos botões “Criar” das telas de “Nova competição” e “Nova equipe”, ou seja, não há um padrão para as funcionalidades.

Nova competição

Nome *

Sigla *

Nível *

Data do início da competição *

Data do fim da competição *

Data do início da inscrição *

Data do fim da inscrição *

Quantidade de equipes *

Tipo de premiação *

Valor de premiação *

Regras *

Criar

Nova equipe

Nome da equipe *

Sigla da equipe *

Criar

Além disso, datas que devem obrigatoriamente vir antes de outras devem seguir um padrão, e se apresentarem na sequência correta. Pode-se observar na tela de “Nova competição” que um padrão não é seguido, e as datas de inscrição se apresentam posteriormente às datas de início e fim da competição.

Nova competição

Nome *

teste

Nível *

Livre

Data do início da competição *

01/12/2020

Data do fim da competição *

02/12/2020

Data do início da inscrição *

03/01/2021

Data do fim da inscrição *

04/01/2021



Ocorreu um erro inesperado!

Data de início da competição deve ser posterior a 2021-01-04T02:00:00.000Z

OK

5.5 PREVENÇÃO DE ERROS

Esta heurística dita que os melhores designs previnem erros de ocorrerem, eliminando condições que são cabíveis a erros, ou pelo menos confirmando com o usuário se deseja seguir com a opção errônea, apresentando o erro e demonstrando possíveis soluções para o mesmo.

No sistema, os campos de datas são propensos a erros, pela falta de algumas validações nos mesmos, com as validações ocorrendo em sua totalidade ao tentar completar a ação.

Na evidência a seguir, na tela de “Editar perfil”, o usuário pode selecionar datas de nascimento com datas que sequer chegaram. O sistema deveria prevenir que essas datas fossem escolhidas para evitar erros do usuário, ao invés de somente realizar a validação ao salvar a edição do perfil.

Editar perfil

Nome *

Mateus Rodrigues

Username do League of Legends *

UnoDeFirmaTunado

Email *

mateus@gmail.com

Role preferida *

Top

Data de nascimento *

09/07/2021

Um caso semelhante acontece na tela de “Nova competição”, onde as datas de início e fim da competição podem ser selecionadas anteriormente ao dia corrente.

Nova competição

Nome *

Sigla *

Nível *

Livre

Data do início da competição *

09/12/2020

Data do fim da competição *

11/12/2020

Data do início da inscrição *

03/02/2021

Quantidade de equipes *

4 equipes

Tipo de premiação *

RP

dezembro de 2020

D	S	T	Q	Q	S	S
29	30	1	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
27	28	29	30	31	1	2
3	4	5	6	7	8	9

Hoje

Em caso de "nenhum", coloque 0

O mesmo ocorre nas datas de inscrição.

Nova competição

Nome *

Sigla *

Nível *

Livre

Data do início da competição *

09/12/2020

Data do fim da competição *

11/12/2020

Data do início da inscrição *

05/11/2020

Data do fim da inscrição *

28/11/2020

Quantidade de equipes *

4 equipes

Tipo de premiação *

RP

Regras *

novembro de 2020

D	S	T	Q	Q	S	S
1	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	27	28
29	30	1	2	3	4	5
6	7	8	9	10	11	12

Hoje

e 0

Na página de “Nova competição” os campos referentes à intervalo de datas (decorrer da competição e decorrer da inscrição) permite que o usuário selecione datas de fim anteriores às datas de início, ocorrendo a validação somente quando o usuário clica no botão “Criar”. O sistema deveria limitar o usuário a escolher datas de término posteriores às datas de início para prevenir condições de erro.

Nova competição

Nome *

Sigla *

Nível *

Livre

Data do início da competição *

03/01/2021

Data do fim da competição *

01/01/2021

Data do início da inscrição *

dd/mm/aaaa

Quantidade de equipes *

4 equipes

Tipo de premiação *

RD

janeiro de 2021

D	S	T	Q	Q	S	S
27	28	29	30	31	1	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	27	28	29	30
31	1	2	3	4	5	6

Hoje

Em caso de "nenhum" coloque 0

Isso também ocorre nos campos de datas de inscrição.

Nova competição

Nome *

Sigla *

Nível *

Livre

Data do início da competição *

09/12/2020

Data do fim da competição *

11/12/2020

Data do início da inscrição *

09/01/2021

Data do fim da inscrição *

02/01/2021

Quantidade de equipes *

4 equipes

Tipo de premiação *

RP

Regras *

janeiro de 2021

D	S	T	Q	Q	S	S
27	28	29	30	31	1	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	27	28	29	30
31	1	2	3	4	5	6

Hoje

e 0

Além disso, nas datas de início e fim da inscrição, podem ser selecionadas datas posteriores às datas de início e fim da competição e vice-versa, com a validação ocorrendo apenas quando o usuário clicar em “Criar” ao invés de ocorrer também no campo em si, permitindo o usuário a criar esta situação que causará um erro.

Nova competição

Nome *	Sigla *
<input type="text"/>	<input type="text"/>
Nível *	
<div>Livre</div>	
Data do início da competição *	Data do fim da competição *
<div>09/12/2020</div>	<div>11/12/2020</div>
Data do início da inscrição *	Data do fim da inscrição *
<div>09/01/2021</div>	<div>02/01/2021</div>

O usuário também pode clicar no botão de sair do sistema sem algum tipo de confirmação por parte do sistema, podendo perder dados preenchidos em formulários e causar alguns tipos de inconveniência por conta de um erro humano.

Mateus ▶

Equipes ▶

Competições ▶

Nova competição

Nome *

teste

Sigla *

tes

Nível *

Livre

Data do início da competição *

09/12/2020

Data do fim da competição *

11/12/2020

Data do início da inscrição *

09/01/2021

Data do fim da inscrição *

02/01/2021

5.6 RECONHECIMENTO AO INVÉS DE MEMÓRIA

Esta heurística dita que o usuário deve utilizar a menor quantidade de sua memória possível. O sistema deve fornecer as informações necessárias de outras partes da interface para o usuário realizar suas atividades.

No sistema, em telas de edição os dados já vêm preenchidos com as informações correntes. Porém, em telas de criação, ao digitar os dados, sair daquela página e retornar à mesma, os dados são apagados e o usuário deve digitá-los novamente, tendo que usar de sua memória para lembrar dos dados preenchidos.

Isso ocorre na tela de “Nova equipe”:

Nova equipe

Nome da equipe *

Sigla da equipe *

Criar

Ocorre também na tela de “Nova competição”:

Nova competição

Nome *

Sigla *

Nível *

Livre

Data do início da competição *

dd/mm/aaaa

Data do fim da competição *

dd/mm/aaaa

Data do início da inscrição *

dd/mm/aaaa

Data do fim da inscrição *

dd/mm/aaaa

Quantidade de equipes *

4 equipes

Tipo de premiação *

RP

Valor de premiação *

Em caso de 'nenhum', coloque 0

Regras *

Criar

5.7 FLEXIBILIDADE E EFICIÊNCIA DE USO

Nesta heurística, é dito que atalhos para usuários mais experientes podem acelerar e facilitar o processo. Como o sistema se trata de um sistema básico de *CRUD* (criação, consulta, alteração e remoção), e não uma ferramenta com funcionalidades próprias e peculiares, não são necessários muitos atalhos. Foi observado no entanto que atalhos básicos de campos de texto (por exemplo, atalhos com o botão “CTRL”, como CTRL+A para selecionar todo o campo, CTRL+Z para desfazer a última alteração no campo etc.) funcionam normalmente. E atalhos que são comuns para diversos sistemas, como “ENTER” para avançar o processo, também funcionam normalmente.

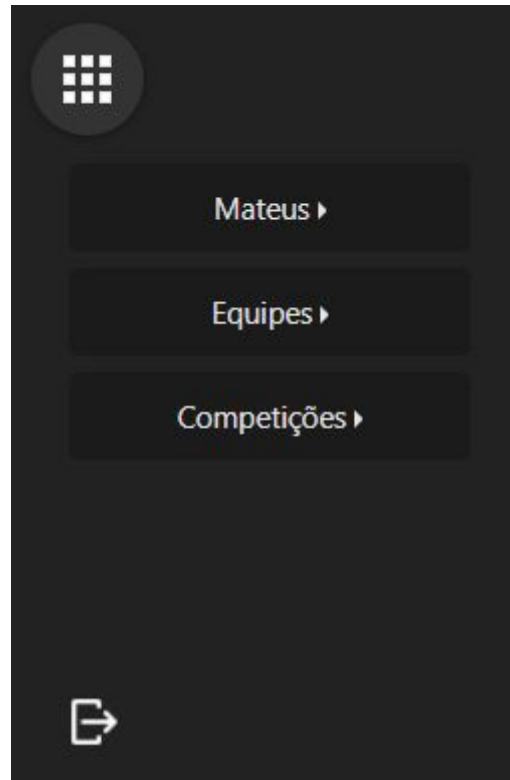
5.8 ESTÉTICA E DESIGN MINIMALISTA

É dito nesta heurística que as interfaces devem ser simplistas no sentido que devem oferecer apenas informações necessárias, e não conter informações que serão irrelevantes e raramente utilizadas. Porém, isso não significa que o design da interface deve ser monótono, significa apenas que os elementos essenciais são o foco da interface.

No sistema, não foram observadas informações desnecessárias na tela, todas as informações e funcionalidades pareciam ter um uso dentro da interface que eram inseridos.

Porém, na parte estética foram analisados alguns problemas. Primeiramente, alguns botões possuem pouco contraste com o fundo da área que se encontram, podendo causar certa dificuldade na visualização dos mesmos.

Este problema foi observado no menu lateral:



Algumas fontes se encontram praticamente invisíveis por não possuírem um contraste com o fundo. Este problema foi encontrado na tela de “Sua competição”, no nome da competição

Sua competição

testeteste

Prints recebidos

No sistema, os campos de texto não possuem um tamanho flexível que se ajusta de acordo com o tamanho do texto inserido, e nem uma maneira de visualizar todo o campo, como uma barra de rolagem no campo de texto. Isso pode ter sido feito visando manter a estética da interface, porém, não é uma boa escolha tendo em vista a usabilidade por parte do usuário, pois em campos que são necessárias muitas informações (como por exemplo o campo “Regras”, ilustrado na evidência a seguir) o usuário é forçado a utilizar comandos no teclado para poder verificar todos os dados escritos no campo.

Isso pode ser evidenciado em todos os campos de texto no sistema. Abaixo segue uma evidência na área de “Nova competição”:

Nova competição

Nome *

teste teste teste teste teste teste teste teste

Sigla *

teste teste teste teste teste teste teste teste

Nível *

Livre

Data do início da competição *

dd/mm/aaaa



Data do fim da competição *

dd/mm/aaaa



Data do início da inscrição *

dd/mm/aaaa



Data do fim da inscrição *

dd/mm/aaaa



Quantidade de equipes *

4 equipes



Tipo de premiação *

RP



Valor de premiação *

Em caso de 'nenhum', coloque 0

Regras *

teste Regras teste Regras teste Regras teste Regras teste Regras teste Regras teste Regras teste

Criar

5.9 ASSISTÊNCIA AO USUÁRIO RECONHECER, DIAGNOSTICAR E SE RECUPERAR DE ERROS

Esta heurística indica que as mensagens de erro devem ser ilustradas de uma maneira entendível ao usuário, indicando o problema e possivelmente a solução.

No sistema, as mensagens de erro são customizadas de acordo com o erro encontrado, ilustrando corretamente o campo e dando soluções plausíveis para o problema. Porém, as mensagens podem ser pouco informativas do motivo do problema. Isso pode ser observado em campos de data, onde o sistema informa ao usuário o intervalo possível para a data, porém, não indica o motivo deste intervalo.

Isso pode ser observado nos campos de data, na tela de “Nova competição”. A evidência abaixo ilustra uma mensagem de erro com a data de início da competição. A data do início da competição deve ser posterior a “2021-01-08” pois a data do fim da inscrição é neste mesmo dia, e isso não é especificado na mensagem de erro, o que é uma informação importante para o usuário decidir como abordará o erro.

The screenshot shows a web form titled "Nova competição" with a dark background. A white modal dialog box is centered on the screen, displaying an error message. The message has a red 'X' icon and reads: "Ocorreu um erro inesperado! Data de início da competição deve ser posterior a 2021-01-08T02:00:00.000Z". Below the message is a blue "OK" button. In the background, the form fields are visible: "Nome *" with the value "dadadadas", "Nível *" with the value "Livre", "Data do início da competição *" with the value "05/01/2021", "Data do fim da competição *" with the value "06/01/2021", "Data do início da inscrição *" with the value "07/01/2021", and "Data do fim da inscrição *" with the value "08/01/2021". Each date field has a calendar icon to its right.

Essa falta de especificação na mensagem de erro também ocorre em datas com a data final menor que a data inicial de algum intervalo. Isto pode ser observado nos intervalos de data de competição e inscrição na tela de “Nova competição”:

The screenshot shows the same "Nova competição" form as the previous one, but with a different error message. The modal dialog box displays: "Ocorreu um erro inesperado! Data do fim da competição deve ser posterior a 2021-01-07T02:00:00.000Z". The background form fields are: "Nome *" with "dadadadas", "Nível *" with "Livre", "Data do início da competição *" with "07/01/2021", "Data do fim da competição *" with "06/01/2021", and the "Data do início da inscrição *" and "Data do fim da inscrição *" fields are not visible in this view.

Nova competição

Nome *

dadadadas

Nível *

Livre

Data do início da competição *

07/01/2021

Data do fim da competição *

08/01/2021

Data do início da inscrição *

04/01/2021

Data do fim da inscrição *

03/01/2021

Ocorreu um erro inesperado!

Data do fim da inscrição deve ser posterior a 2021-01-04T02:00:00.000Z

OK

Não foi possível verificar o erro no campo de “Data de nascimento” na edição do perfil do usuário, pois o mesmo possui um erro de validação, onde datas de nascimento podem ser salvas com datas posteriores ao dia corrente.

5.10 AJUDA E DOCUMENTAÇÃO

Esta heurística dita que é melhor que o sistema não necessite de alguma explicação adicional, porém, caso necessite, o sistema deve oferecer uma documentação apropriada para auxiliar o usuário a completar a ação desejada.

As funções do sistema são bem simples, e os botões, campos de texto, filtros e afins apresentam textos bem informativos de suas funções. Porém, o botão de sair do sistema possui uma documentação ambígua, contendo apenas o texto “Sair”, o que pode vir a confundir o usuário, que pode inferir que o botão possui a função de sair da página atual.

