



**Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"**

GABRIEL ALAN MADUREIRO GONÇALVES

**CRIAÇÃO FLEXÍVEL DE DESIGN DE WIDGETS PARA DASHBOARDS
INTERATIVOS**

Assis/SP

2019



**Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"**

GABRIEL ALAN MADUREIRO GONÇALVES

**CRIAÇÃO FLEXÍVEL DE DESIGN DE WIDGETS PARA DASHBOARDS
INTERATIVOS**

Trabalho de Conclusão de Curso apresentado ao Curso de Bacharelado em Ciência da Computação do Instituto Municipal de Ensino Superior de Assis – IMESA e a Fundação Educacional do Município de Assis – FEMA, como requisito parcial à obtenção do Certificado de Conclusão.

Orientando: Gabriel Alan Madureiro Gonçalves

Orientador: Douglas Sanches da Cunha

FICHA CATALOGRÁFICA

FICHA CATALOGRÁFICA

G635c GONÇALVES, Gabriel Alan Madureiro
Criação flexível de design de widgets para dashboards interativos
Gabriel Alan Madureiro Gonçalves. – Assis, 2019.

41p.

Trabalho de conclusão do curso (Ciência da Computação). –Fundação Educacional do Município de Assis-FEMA

Orientador: Me. Douglas Sanches da Cunha

1.Widget 2.Business intelligence 3.Dashboard interativo
CDD 005.74

**CRIAÇÃO FLEXÍVEL DE DESIGN DE WIDGETS PARA DASHBOARDS
INTERATIVOS**

GABRIEL ALAN MADUREIRO GONÇALVES

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, avaliado pela seguinte comissão examinadora:

Orientador: _____

Douglas Sanches da Cunha

Examinador: _____

Célio Desiró

Assis/SP

2019

DEDICATÓRIA

Dedico este trabalho aos meus pais, Alcir Ferreira Gonçalves e Andrea Regina Leite Madureiro Gonçalves, os quais sempre estiveram ao meu lado, educando, repreendendo e consolando. Dedico isto a eles, pois sempre se esforçaram em me dar o melhor e jamais desistiram de lutar por minha educação.

AGRADECIMENTOS

Agradeço a **Deus**, por imensa misericórdia, graça e consolo durante todo o tempo que me foi permitido estar entre meus colegas de estágio, classe, entre outros.

Agradeço aos meus avôs **Alcides Ferreira Gonçalves** e **Dorival Antonio Madureiro** pelo exemplo de vida e esforço e às minhas falecidas avós **Izaura Maria Felizardo Gonçalves** e **Jocelina Rosa Leite Madureiro**, por me marcarem como exemplo de humildade e amor intenso.

Agradeço aos meus tios **Adriana; Alberto; Alessandro; Almir, André; Marcos; Vera** e aos meus primos **Diego** e **Thiago** por todo o carinho e zelo que sempre tiveram para comigo.

À minha namorada **Emanuelle Haddad Smozinski**, que sempre esteve presente para me acalmar, alegrar e levantar perante meus obstáculos.

À minha melhor amiga **Victória Ribeiro Silvestre**, pela grande amizade que carrego há vários anos; por todos os conselhos; pelas engraçadas chamadas de vídeo; pelo ótimo sistema Poliedro.

À minha grande e mentalmente idosa amiga **Bruna Mara Barboza da Silva**, por todos os conselhos; risadas; broncas e mais broncas.

À uma das pessoas que mais influenciam minha formação, **Ivanilde da Silva Ramos**; pelas boas lembranças na Câmara Municipal; por todo o cuidado.

Aos meus amigos de coração **Gabriela Diniz Thomé; Luan Moraes Thomé; Viviane de Lucio Diniz Thomé** e ao recém-nascido **Guilherme Diniz Thomé**, uma família que se mostrou presente em me aconselhar e motivar durante minha mocidade.

Aos meus queridos amigos **Israel Soares D'Aurégio, João Victor Viel Pereira Pinto, Lucas Teixeira de Lemos, Matheus Cavalcanti dos Santos, Michel Gargel Nunes e Renato Virto Moreira** pelos memoráveis momentos alegres e cômicos durante o curso e estágio.

Ao meu memorável **Douglas**, por todo o conhecimento transmitido; pelas boas aulas de Arquitetura de Computadores e Sistemas Operacionais; pelos bons e cômicos momentos em estágio. Pela grande sinceridade em conselhos e paciência.

Por fim, agradeço a dois seres extremamente importantes que ganhei como irmãos, no exato início desta graduação, por mais que não possam falar e nem compreender o que aqui está escrito. Meu incrível gato **Leonardo** e sua mãe **Princesa**, pelo carinho; bons momentos e presença insubstituível.

“Caia sete vezes; levante-se oito.”

Provérbio Japonês

“Tudo posso naquele que me fortalece.”

(BÍBLIA, Filipenses, 4, 13)

RESUMO

O uso de *dashboards* para tomada de decisões, busca de padrões, gerenciamento de recursos corporativos e pessoais, teve grande ascensão como solução de dados para instituições dos mais variados gêneros. Entretanto, nem sempre tais *dashboards* possuem uma ergonomia coerente com o escopo de sua implementação ou de um usuário específico.

Além disso, a busca por profissionais com conhecimento em *design* de interface tornou-se importante no mercado de trabalho, devido a mudança de visão a respeito de *design*. Agora, as reações causadas pela facilidade de uso, eficiência e aplicações visualmente belas, geraram uma necessidade tão grande, quanto apenas o atributo de funcionalidade de um sistema.

Em consequência destes problemas de criação de *dashboards* dinâmicas e a deficiência de profissionais de *design* no mercado, este trabalho tem como função estudar os conceitos a respeito de *dashboards* e seus componentes derivados; Conceitos relacionados a IHC (Interface Humano-Computador) e por fim a implementação de uma ferramenta de tecnologia web para geração de *widgets* para *dashboards* dinâmicas.

Palavras-Chave: *Dashboard*, Interface Humano-Computador, *Design*, *User Experience*, *Web*.

ABSTRACT

The use of dashboards for decision-making, standards-seeking, enterprise and personal resource management are increasingly being implemented across a range of applications. However, such dashboards do not always have ergonomics consistent with the scope of their implementation or a specific user.

In addition, the search for professionals with knowledge in interface design has become important in the job market, due to the change of vision regarding interface design. Now the reactions caused by ease of use, efficiency, and visually beautiful applications have generated as much need as just the functionality attribute of a system.

Because of these problems and deficiencies, this work aims to study the concepts regarding dashboards and their derived components; Concepts related to IHC (Human-Computer Interface) and finally the implementation of a web technology tool to generate widgets for dynamic dashboards.

Keywords: Dashboard, Human-Computer Interface, Design, User Experience, Web.

LISTA DE ILUSTRAÇÕES

Figura 1: Quatro Perspectivas de Interação.....	21
Figura 2: Ciclo de Design.....	23
Figura 3: Arquitetura da Solução.....	31
Figura 4: Transformação de Dados para o Dashboard Dinâmico.....	31
Figura 5: Tela de Criação de Widgets antes da seleção de serviço.....	32
Figura 6: Janela de seleção de serviço.....	33
Figura 7: Tela de Criação de Widgets.....	33
Figura 8: Dashboard após a criação do novo widget.....	34
Figura 9: Janela de alteração de Widget.....	35
Figura 10: Tela de Configuração de Widget.....	35
Figura 11: Métodos estáticos emissores de eventos.....	36
Figura 12: Transformação de Dados para os Microserviços de BI.....	37

SUMÁRIO

1. INTRODUÇÃO	13
1.1 OBJETIVOS.....	14
1.2 JUSTIFICATIVAS.....	14
1.3 MOTIVAÇÃO	15
1.4 PERSPECTIVAS DE CONTRIBUIÇÃO.....	15
1.5 METODOLOGIA DE PESQUISA	16
2. DASHBOARDS.....	17
2.1 PROBLEMAS EM DASHBOARDS	18
2.2 COMPONENTES DE DASHBOARD	18
2.2.1 MÉTRICAS.....	18
2.2.2 KPIs.....	18
3. IHC – INTERFACE HUMANO - COMPUTADOR.....	20
3.1 INTERAÇÃO.....	20
3.2 INTERFACE.....	21
3.3 DESIGN	22
3.4 USABILIDADE.....	23
3.5 UX – USER EXPERIENCE	24
4. FERRAMENTAS.....	26
4.1 LINGUAGENS	26
4.1.1 HTML.....	26
4.1.2 JAVASCRIPT E TYPESCRIPT	26
4.1.3 CSS.....	27
4.2 FRAMEWORKS E BIBLIOTECAS AUXILIARES	27
4.2.1 ANGULAR	27
4.2.2 ANGULAR 2 GRIDSTER.....	28

4.2.4 PRIMENG.....	28
4.3 JSON.....	29
5. PROPOSTA E DESENVOLVIMENTO DO TRABALHO.....	30
5.1 DEFINIÇÃO DO PROBLEMA.....	30
5.2 ARQUITETURA DA SOLUÇÃO.....	30
5.2.1 RECUPERAÇÃO DE DADOS PERSISTIDOS.....	31
5.2.2 GERENCIAMENTO DE WIDGETS.....	32
CONCLUSÃO.....	38
REFERÊNCIAS.....	39

1. INTRODUÇÃO

Uso de dados na gestão e tomada de decisões tornou-se uma grande tendência no Brasil e no mundo. Através da informação é possível ter um controle sobre dados pessoais, empresariais, e de como uma plataforma tem sido utilizada por um específico usuário. Segundo a Serasa Experian, em 2018, 91% das empresas brasileiras consideraram dados na definição da estratégia de negócios, número maior que o apresentado em 2017 (86%).

Na utilização dos dados, em diversos tipos de softwares, é implementado um conceito conhecido por “*Dashboard*”. *Dashboard* caracteriza-se por um painel com um conjunto de *widgets* KPIs (*Key Performance Indicator*) que indicam valores, métricas, gráficos ou qualquer tipo de informação relevante ao usuário ou seu negócio. “O *Dashboard* é uma ferramenta de visualização de dados que exibe o status atual das métricas e indicadores chave de desempenho (KPIs) de uma empresa.” (PITON).

O conceito tem como objetivo proporcionar uma interface de monitoramento, visual e enxuta, com base em diversos tipos de dados específicos e gerais, tais como: Quantidade de Vendas em um período; Porcentagem de lucros no mês; entre outros. Podem também ser utilizados como ícones para acesso à visualização mais detalhada de um uma fonte de dados.

A facilidade no monitoramento dos dados pode se tornar um problema, caso existam inconsistências nos dados, resultando em informação falsa. Além de que determinados *dashboards* podem não corresponder as necessidades de um usuário, exibindo um *dashboard* com informações pouco relevantes ou, em alguns casos, com componentes que poderiam usufruir de técnicas visuais de maneira mais eficiente proporcionando uma melhor experiência ao usuário. “A questão é: muitas vezes, quando tentamos entender as informações disponíveis, precisamos de muito mais do que um simples recorte, algo específico de um dado ou uma única perspectiva.” (GRIGO-MCMAHON; MURRAY, 2017).

Os conceitos de desenvolvimento de *dashboards* abrangem algumas áreas do conhecimento com um certo contraste entre si, como: *Data Science*; BI (*Business Intelligence*); UI (*User Interface*); UX (*User Experience*). Atualmente diversos sistemas e aplicativos usufruem do uso dos dados e do conhecimento em experiência do usuário, tais como a linguagem de *design Metro*, criada pela Microsoft que proporcionou uma grande mudança no desenvolvimento dos próximos sistemas operacionais, devido a sua facilidade de manipulação pelo usuário e retorno de informação útil, de forma sutil e minimalista. Além de diversos aplicativos de “gestão pessoal”, que realizam cálculos para mostrar ao usuário informações a respeito de um avanço em alguma atividade ou meta.

Este trabalho tem como objetivo, a implementação de uma ferramenta de criação de *dashboards* (Blocos de KPI), de maneira flexível em todo o espaço disponível do *dashboard*, o que possibilitará ao usuário a customização visual de seu componente, como estilo e cores. E a implementação da customização de funcionalidade, podendo escolher que tipo de dado e de que maneira necessita ser exibido, através de serviços com protocolo HTTP.

1.1 OBJETIVOS

O objetivo deste trabalho constitui na implementação de um “gerador” de *widgets* em um ambiente (*Dashboard*) interativo, o mesmo que deverá fornecer diversos serviços de BI, os quais serão utilizados na configuração de dados de um *dashboard* criado pelo usuário.

Por fim, este trabalho visa a criação de uma ferramenta que possibilita a geração de *dashboards* de forma personalizada (seguindo os padrões de estilo), o que proporcionará uma alta produtividade a qualquer usuário de um determinado sistema.

1.2 JUSTIFICATIVAS

A demanda por *dashboards* com componentes visualmente e operacionalmente eficientes é grande. Alguns *dashboards* atualmente no mercado buscam o melhor desempenho em mostrar informações aos usuários, entretanto, acabando excedendo em quantidade, deixando o ambiente um pouco confuso, além de haver muita limitação de opções de personalização. Para que uma possível alteração no ambiente possa ser realizada, é necessário um determinado tempo para programação de serviços adequados a necessidade do usuário, tão bem como ao desenvolvimento visual de mais um componente.

A possibilidade de uma aplicação que permita reduzir o tempo gasto em desenvolvimento visual torna bastante interessante, pois, além de aumentar a produtividade e eficiência de uso do usuário, poderá proporcionar uma economia de tempo em desenvolvimento visual, deixando apenas, um padrão a ser seguido.

Assim, a implementação deste trabalho tem como justificativa a necessidade de suprir algumas lacunas no desenvolvimento de *dashboards* que ainda hoje, não satisfazem todas as necessidades dos usuários.

1.3 MOTIVAÇÃO

Este trabalho tem como motivação a implementação de uma ideia capaz de suprir as necessidades do mercado, como uma extensão ao conceito já consolidado de *dashboard*, apresentando uma nova maneira de desenvolvimento de *widjets* em *dashboards*, onde o usuário, mediante aos escopos de ferramentas visuais e de serviços disponíveis, desenvolve seus próprios componentes, oferecendo uma maior sensação de controle ao usuário.

Também é evidente que ao decorrer deste desenvolvimento, uma grande quantidade de informações de extrema importância será recuperada, devido um grande leque de tópicos de conhecimento abordados como *Design UX* e *UI*; *Business Intelligence*, entre outros.

1.4 PERSPECTIVAS DE CONTRIBUIÇÃO

No decorrer do desenvolvimento deste trabalho, haverá possíveis contribuições, sejam elas contribuições ao conhecimento ou contribuições aos usuários e gerentes de negócios.

A cerca do conhecimento obtido, pode ser destacado a necessidade do estudo dos conceitos de experiência de usuário, *design* de interface, tecnologias e linguagens para o *desenvolvimento web*. Para os usuários e gerentes de negócios, o desenvolvimento deste trabalho poderá contribuir para uma melhor experiência em um ambiente acadêmico e corporativo, através de *widjets* que realmente possam satisfazer as necessidades de seus usuários. Portanto, realizar um estudo e implementação unindo estes conceitos e ferramentas trará grandes contribuições para o desenvolvimento de novos ou já conhecidos *dashboards*.

1.5 METODOLOGIA DE PESQUISA

Devido a quantidade de temas distintos relacionados ao desenvolvimento de *widgets* que deverão ser abordados, esta pesquisa caracteriza-se como uma Pesquisa Exploratória. Tendo como base o procedimento de Pesquisa Bibliográfica, o referencial teórico deste trabalho será obtido por meio de artigos científicos, livros e *web sites*.

A pesquisa bibliográfica foi feita a partir do levantamento de referências teóricas já analisadas, e publicadas por meios escritos e eletrônicos, como livros, artigos científicos, páginas de *web sites*. Qualquer trabalho científico inicia-se com uma pesquisa bibliográfica, que permite ao pesquisador conhecer o que já estudou sobre o assunto. Porém, pesquisas científicas que se baseiam unicamente na pesquisa bibliográfica, procurando referências teóricas publicadas com o objetivo de recolher informações ou conhecimentos prévios sobre o problema a respeito do qual se procura a resposta (FONSECA, 2002, p. 32).

1.6 ESTRUTURA DO TRABALHO

Este trabalho tem sua estrutura definida em cinco capítulos, onde os três iniciais abordam todos os conceitos utilizados neste trabalho e os capítulos restantes abordam todo o escopo de implementação.

No primeiro capítulo deste trabalho está a introdução, onde é apresentada a área de estudo e os respectivos objetivos, justificativas, motivações, perspectivas de contribuição e metodologia de pesquisa a respeito deste trabalho. No capítulo seguinte é abordado o conceito de *dashboard*, algumas de suas deficiências e seus componentes. No terceiro capítulo é apresentado o conceito de IHC, sua importância e em seguida, outros conceitos indispensáveis para o desenvolvimento de interfaces para usuários.

No quarto capítulo são apresentadas as linguagens e ferramentas que serão utilizadas durante a implementação deste trabalho e por fim, no quinto capítulo são definidos o problema, proposta de solução e arquitetura implementada.

2. DASHBOARDS

A utilização de *dashboards* é um grande atrativo para empresas e negócios devido ao seu fácil modelo de apresentação de informação. O termo *dashboard* tem como conceito histórico: “Uma tábua de madeira ou couro na frente de uma carruagem, para impedir a entrada de lama.” (OXFORD, 2019, traduzido).

Após algumas décadas após a Revolução Industrial, com a evolução das máquinas, o termo *dashboard* foi começando a tomar forma até se transformar no que é comumente usado hoje, uma grande referência ao original termo de “tábua de madeira”. O que era utilizado apenas como mero artefato de controle/informação de máquina em tempo real (medidores de temperatura, água, combustível, velocidade), começa a ser utilizado para a gestão de sistemas, empresas e recursos pessoais.

“Painéis de controle e visualização são ferramentas cognitivas que melhoram seu "espaço de controle" sobre muitos dados de negócios. Essas ferramentas ajudam as pessoas a identificar visualmente tendências, padrões e anomalias, auxiliando as sobre o que veem e ajudando a guiá-las em direção a decisões eficazes.” (BRATH, Richard. PETERS, Michael. 2004, p. 1, traduzido).

Por meio de *dashboards* é possível ter uma ampla visão de tudo o que ocorre em um sistema, empresa ou pessoa. No contexto de negócios, a sua implementação resume na exibição de informação refinada, oriunda de pequena, média ou uma grande escala de dados, proporcionando aos seus usuários a descoberta de padrões e informação relevante para a tomada de decisões de uma empresa.

A visualização da informação de alta qualidade acerca do desempenho ou situação de um artefato é muito apreciada, entretanto, a obtenção de informação através de tantos dados dispersos e desestruturados traz ao desenvolvimento de *dashboards* duas questões:

- 1) Como desenvolver *dashboards* que satisfaçam as necessidades de uma corporação ou usuário?
- 2) As informações exibidas aos usuários (sejam finais ou secundários) são realmente necessárias, acerca de suas respectivas funções em uma corporação, ou até mesmo, úteis para o seu gerenciamento pessoal?

2.1 PROBLEMAS EM DASHBOARDS

Para que um *dashboard* seja eficiente para um usuário específico, é necessário que os dados exibidos ao mesmo sejam coerentes as perante a suas necessidades. Uma grande deficiência frequente em diversas aplicações é a má implementação de *dashboards*, onde muitos dados são exibidos de uma só vez, mostrando mais que o necessário e dando um aspecto semelhante a um *cockpit* de aviação, devido a quantidade de componentes informativos (Blocos, itens ou cartões, presentes em um *dashboard* que exibe informação por meio de números e gráficos) presentes em apenas uma interface.

Um dashboard pode fazer todo o sentido para seu criador, entretanto, pode confundir o usuário final que consome as informações. (AD Digital).

2.2 COMPONENTES DE DASHBOARD

Acerca do conceito de *dashboards*, é relevante o conhecimento de alguns de seus componentes e suas características. Existem as Métricas e os Indicadores-Chave de Desempenho, conhecidos como *KPIs* (*Key Performance Indicator*).

2.2.1 Métricas

Métricas são artefatos que exibem uma estatística (de maior parte numérica) quantitativa a respeito de alguma necessidade, com a função de “medir” as variáveis de um escopo. Exemplo: Quantidade de visitas em um *website*; Quantidade de vendas; etc.

Métricas, são medidas que servem de base para a constituição de um indicador, normalmente associadas ao comportamento do usuário e sem uma meta definida. (MOTTA, 2015).

2.2.2 KPIs

KPI é um artefato que exibe o progresso para atingir determinado resultado. É criado a partir das métricas, que tem o objetivo de mostrar o progresso de atingimento de determinado resultado ou meta. Este tipo de artefato é recomendado para a medição de progressos buscados por uma organização. Além de avaliar a performance organizacional, os indicadores podem ser úteis para analisar tendências do usuário.

KPI é a sigla para o termo em inglês Key Performance Indicator, que significa indicador-chave de Desempenho. Esse indicador é utilizado para medir o desempenho dos processos de uma empresa e, com essas informações, colaborar para que alcance seus objetivos (INTERNET INNOVATION).

3. IHC – INTERFACE HUMANO - COMPUTADOR

A base de conhecimento em IHC (Interface Humano-Computador) é essencial na construção de interfaces eficientes e que proporcionem um bom nível de usabilidade para seus usuários, além do conceito muito estudado, chamado Experiência de Usuário (*UX*). IHC é uma disciplina interessada no projeto, implementação e avaliação de sistemas computacionais interativos para uso humano, juntamente com os fenômenos relacionados a esse uso. (Hewett *et al.*, 1992).

O conceito que já era objeto de estudo durante a década de 80, tornou-se mais forte ainda com o surgimento da *World Wide Web* na década seguinte. Após a chegada de dispositivos móveis e seus diversos tipos de *displays* unida à liberdade de desenvolvimento, a necessidade de *interfaces* mais “sedutoras” começou a ter relevância semelhante às necessidades de funcionalidade. *Designers* de interação tornaram-se necessários, assim com os engenheiros de software, para criar experiências interessantes (BEYON, David).

IHC pode ser considerada como um conjunto de métodos, princípios, conceitos e regras que são importantes para garantir que os sistemas sejam fáceis de usar e aprender, indo além do escopo de funcionalidade, provocando satisfação aos seus usuários.

3.1 INTERAÇÃO

Interação é o termo utilizado para descrever o contato, conexão entre os objetos através de uma sequência de ações e reações através de estímulos como em relações humanas, por exemplo. As máquinas trouxeram previamente o conceito de operação, entretanto, com o avanço em tecnologias e áreas como a cognição, começaram a ser desenvolvidos sistemas que permitiam que houvesse uma conexão entre o usuário e sua máquina para um fim.

Em geral, a interação usuário-sistema pode ser considerada como tudo o que acontece quando uma pessoa e um sistema computacional se unem para realizar tarefas, visando a um objetivo (Hix e Hartson *et al.*, 1993).

O livro *Interação Humano-Computador* explana alguns exemplos de interações, identificadas por Kammergaard em 1988, resultando em 4 perspectivas que são:

- **Sistema:** Quando o usuário realiza operações muito específicas em uma interface, fazendo com que o comportamento do usuário seja parecido com um computador, realizando ações muito bem definidas pelo *designer*.
- **Parceiro de Discurso:** Quando uma interface é programada com ênfase em proporcionar ao usuário a possibilidade de usufruir da ajuda da máquina para a realização de suas tarefas, entretanto, por meio de linguagem natural, seja visual ou auditiva.
- **Mídia:** Nesta perspectiva a interação com uma máquina tem como objetivo de ser um meio de comunicação entre dois ou mais usuários de um sistema.
- **Ferramenta:** A perspectiva de ferramenta remete aos sistemas que têm como objetivo a geração de algum resultado. O seu contexto de interação está contido em utilizar a máquina com um fim de gerar um resultado, trabalho ou produto.

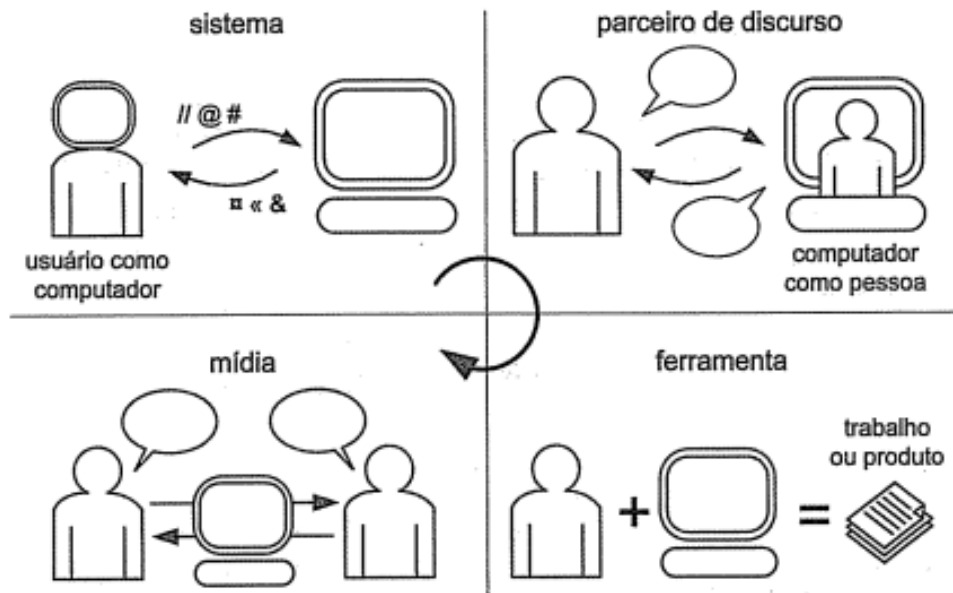


Figura 1: Quatro Perspectivas de Interação

3.2 INTERFACE

A interface é o meio de contato entre o computador e um usuário, por onde ocorre todo o contato físico e conceitual durante uma interação. Por meio da interface, “sistemas” diferentes podem compartilhar informação entre si de inúmeras maneiras (recursos visuais, através de gestos, ações motoras ou sons), que são as interações.

O conceito de interface é definido pelo dicionário Michaelis como um “Dispositivo que causa uma ligação entre dois sistemas” e “Aquilo que ocasiona uma união física ou lógica entre dois sistemas

que, diretamente, não poderiam estar conectados”, dando ênfase à ideia de uma camada intermediária entre dois sistemas.

O conceito de Contato Físico e Conceitual, que ocorre nas interações entre dois sistemas, diz respeito a realização da interação, seja através de entrada de dados ou a interpretação de uma informação. O livro *Interação Humano-Computador* aborda os conceitos de Contato Físico e Conceitual como:

- **Contato Físico:** O contato físico ocorre através do hardware e do software utilizados durante a interação. Dispositivos de entrada, como teclado, mouse, joystick, microfone, caneta (que escreve sobre a tela) e câmera (webcam), permitem ao usuário agir sobre a interface do sistema e participar ativamente da interação. Já os dispositivos de saída, como monitor, impressora e alto-falante, permitem ao usuário perceber as reações do sistema e participar passivamente da interação.
- **Contato Conceitual:** O contato conceitual com a interface envolve a interpretação do usuário daquilo que ele percebe por meio do contato físico com os dispositivos de entrada e de saída durante o uso do sistema. Essa interpretação permite ao usuário compreender as respostas do sistema e planejar os próximos caminhos de interação.

3.3 DESIGN

Conceito de *design* não é fácil de ser definido como os outros anteriormente explanados. As vezes pode ser entendido como o visual de um objeto.

Objetos como um carro, celular e câmera fotográfica foram criados com a premissa de serem uma intervenção em um ambiente, promovendo uma solução para uma deficiência. O simples fato da implementação de uma solução pode ser considerado *design*.

Quando é realizado o *design* de uma solução, todo ambiente de implementado muda. Por exemplo: é possível trabalhar e organizar documentos, papéis, calendários e outros artefatos em uma mesa de trabalho. Porém, se esta mesa se ausentar do design e implementação de uma cadeira, será apenas uma mesa comum, sem a função específica de ser ergonomicamente uma mesa de trabalho.

Em termos bem gerais (Lawson *et al*, 2006; Löwgren e Stolterman *et al*, 2004) e através da figura a seguir, pode-se caracterizar a atividade de *design* como sendo:

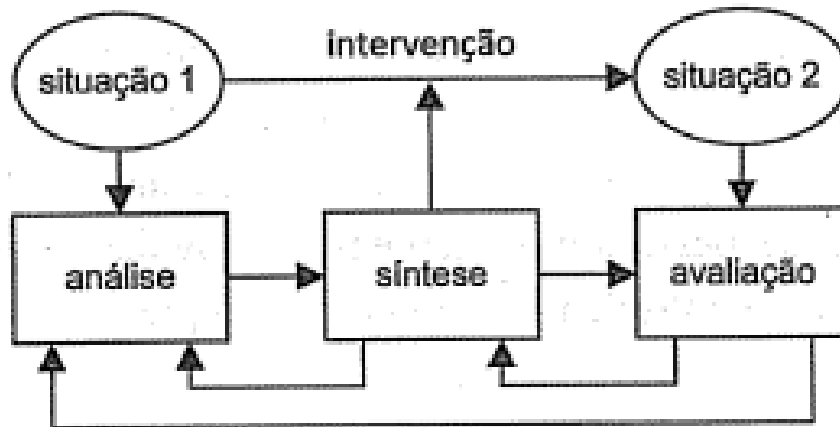


Figura 2: Ciclo de Design

- A **análise** da situação atual: estudar e interpretar a situação atual;
- A **síntese** de uma intervenção: planejar e executar uma intervenção na situação atual;
- A **avaliação** da nova situação: verificar o efeito da intervenção, comparando a situação analisada anteriormente com a nova situação, atingida após a intervenção.

3.4 USABILIDADE

Para que um sistema possa cumprir seu objetivo perante seu contexto é necessário que a interface e interação estejam adequadas para prover o máximo de suporte aos seus usuários.

O Conceito de usabilidade no desenvolvimento de interfaces é formado por diversas características ou atributos de qualidade, que são: facilidade de uso, clareza, eficiência e outras técnicas de *design* que acaba agregando valor a usar habilidade de uma aplicação. De maneira clara, o quanto bem um desenvolver de interfaces utilizar as técnicas de desenvolvimento dando foco aos atributos de qualidade, melhor será a usabilidade e a possível experiência do usuário.

A usabilidade está relacionada com a facilidade de aprendizado e uso da interface, bem como a satisfação do usuário em decorrência desse uso (Nielsen *et al*, 1993).

Um conjunto de atributos relacionados com o esforço necessário para o uso de um sistema interativo, e relacionados com a avaliação de tal uso, por um conjunto específico de usuários (ISO/IEC 9126 *et al*, 1991).

Caso alguns atributos sejam bem implementados durante o desenvolvimento de interfaces, o nível de usabilidade será maior. Nielsen (1993) descreve a composição da usabilidade em cinco elementos:

- **Eficiência (*efficiency*):** O quão eficiente for a interface no cumprimento das tarefas a si competidas, maior será a usabilidade;
- **Facilidade de Aprendizado (*learnability*):** O quão fácil for a aprendizagem de uma aplicação, maior será a usabilidade;
- **Facilidade de Recordação (*memorability*):** O quão memorizável forem as etapas de interação da interface (sem a necessidade de repetitivos treinamentos), maior será a usabilidade.
- **Satisfação do Usuário (*satisfaction*):** O quão prazeroso e agradável for o uso de um sistema (apesar da subjetividade deste conceito, maior será a usabilidade. Este conceito é muitas vezes associado a UX (*user experience*)).
- **Segurança no uso (*safety*):** O quão segura for a interação de um sistema para que não ocorram operações inesperadas ou erros, maior será a usabilidade. Exemplo: Evitar posicionar botões críticos como “excluir” próximos a opções como sair e gravar.

3.5 UX – USER EXPERIENCE

Após a abertura de desenvolvimento de sistemas para inúmeras áreas e atividades do conhecimento humano (saúde, educação, entretenimento), aumentaram a importância de considerar as emoções e sensações durante o uso de uma interface. A experiência do usuário pode ser considerada como um dos atributos de usabilidade, entretanto seu conceito não diz respeito ao lado de *design* de interfaces, mas à abstrata e subjetiva experiência de quem participa das interações com elas.

Alguns interpretam a preocupação com emoções e sentimentos dos usuários como uma atenção maior à satisfação do usuário como parte do critério de usabilidade (Bevan *et al*, 2009). Outros, no entanto, consideram essa preocupação como um critério de qualidade distinto, chamado de experiência do usuário (*user experience* – Sharp *et al*, 2007).

A Satisfação é relacionada com frequência à *user experience*, entretanto, há outras emoções e sensações sendo negativas ou positivas como: Atração, Cansaço, Desafio, Diversão, Entretenimento, Frustração, Interesse, Motivação, Ofensa, Prazer, Provocação (Sharp *et al*, 2007).

A experiência de uso é algo subjetivo, pessoal. Entretanto, podemos projetar sistemas interativos visando promover uma boa experiência de uso, incorporando características que promovam boas emoções nos usuários e que evitem provocar sensações desagradáveis, sempre respeitando as limitações dos usuários (Barbosa, 2010).

Um bom envolvimento emocional dos usuários durante a interação agrega valor ao sistema interativo (Barbosa, 2010).

4. FERRAMENTAS

Para o desenvolvimento deste trabalho, foram utilizadas linguagens e ferramentas relacionadas ao desenvolvimento web. Neste capítulo são abordados seus conceitos, área de implementação?

4.1 LINGUAGENS

Durante a implementação desse projeto foram utilizadas algumas linguagens, não somente de programação, mas como de marcação, por exemplo. Essas linguagens são frequentemente utilizadas na criação de sites e aplicações híbridas, e serão igualmente utilizadas em todo o desenvolvimento do projeto.

4.1.1 HTML

HTML (*Hypertext Markup Language*) é uma linguagem de marcação criada por Tim Berners-Lee em 1991(primeira publicação) no centro de pesquisas CERN, Suíça. Esta linguagem possibilita a renderização de diversos itens gráficos por meio de um navegador de internet (*browser*), como parágrafos, imagens, entre outros. Por meio dela é possível realizar diversas interações com os usuários, tais como a apresentação e leitura de dados (Hostinger, 2019).

Hipertexto significa um texto que possui referências (links) para outros textos que podem ser acessados imediatamente. Ele publicou a primeira versão do HTML em 1991, consistindo em 18 *tags*. Desde então cada versão do HTML vem com novas *tags* e atributos (modificadores de *tags* – Hostinger, 2019).

4.1.2 Javascript e Typescript

Javascript é uma linguagem de programação de comportamento que permite a criação de conteúdos dinâmicos, controle de mídias e animações para deixar um site mais interativo e interessante (Hostinger).

O JavaScript foi originalmente desenvolvido sob o nome de Mocha. Posteriormente teve seu nome modificado para LiveScript e, por fim, JavaScript. LiveScript foi o nome oficial da linguagem quando ela foi lançada pela primeira vez na versão beta do navegador Netscape 2.0, em setembro de 1995, mas teve seu nome alterado em um anúncio conjunto com a Sun Microsystems, em

dezembro do mesmo ano, quando foi implementado no navegador Netscape, versão 2.0B3 (IMasters).

Typescript é um *superset* criado pela Microsoft que fornece recursos que melhor suportam o uso da Programação Orientada a Objetos, que tem como base quatro princípios fundamentais: encapsulamento, herança, abstração e polimorfismo. A POO sempre foi um problema ao ser aplicada em JavaScript, devido a sua sintaxe não permitir escrever classes, por exemplo, de forma tão clara, além da fraca tipagem de dados. O *Typescript* oferece então uma forma de corrigir ou contornar esses problemas, adicionando funcionalidades que, quando compiladas, resultarão em código JavaScript novamente (Devmedia).

4.1.3 CSS

Após a chegada do HTML e sua adesão em novas áreas além do mundo acadêmico e científico, a linguagem ganhou a possibilidade de criar algumas modificações de tamanho de texto e cores. Entretanto, com o uso abusivo de técnicas que embelezavam uma *webpage*, criaram-se problemas relacionados as personalizações, pois era necessário alterar linha após linhas para que uma página pudesse se adequar às necessidades de um *designer*, e isto não era nada produtivo (Devmedia).

Misturar estilo e estrutura não era mais interessante, e foi assim que em 1995, Håkon Wium Lie e Bert Bos apresentaram a proposta do CSS (*Cascading Style Sheets*, ou Folha de Estilo em Cascata) que logo foi apoiada pela W3C (Devmedia).

4.2 FRAMEWORKS E BIBLIOTECAS AUXILIARES

Assim como linguagens, neste trabalho serão utilizados alguns *frameworks* e bibliotecas de componentes, artefatos muito utilizados que aumentaram a produtividade no desenvolvimento de sistemas.

4.2.1 Angular

Angular é uma plataforma e framework para construção da interface de aplicações usando HTML, CSS e, principalmente, JavaScript, criada pelos desenvolvedores da Google (Algaworks).

O Angular é uma plataforma que facilita a criação de aplicativos com a web. Angular combina modelos declarativos, injeção de dependência, ferramentas de ponta a ponta e as melhores práticas recomendadas para resolver desafios de desenvolvimento. O Angular capacita os desenvolvedores a criar aplicativos que estejam na Web, em dispositivos móveis ou na área de trabalho (Angular, traduzido).

4.2.2 Angular 2 Gridster

Angular 2 Gridster é um componente feito para o *framework* Angular que possibilita a criação de um sistema de grids flexíveis, permitindo que um usuário possa redimensionar e modificar as posições de cada componente HTML. É um projeto open source com código mantido no site Github.

4.2.3 BOOTSTRAP

Bootstrap é um *framework* para o desenvolvimento web que fornece padrões de estilo, mantendo uma aplicação com aparência uniforme. Uma característica importante do bootstrap é que foi feito para suprir a necessidade de sistemas que necessitavam adquirir páginas responsivas, que são interfaces que se adequam às diversas resoluções protegendo a facilidade de leitura e visualização de um site (CIA WEBSITES, 2017).

Foi desenvolvido por Jacob Thorton e Mark Otto, engenheiros do Twitter, como uma tentativa de resolver incompatibilidades dentro da própria equipe. O intuito era otimizar o desenvolvimento de sua plataforma através da adoção de uma estrutura única. Isto reduziria inconsistências entre as diversas formas de se codificar, que variam de profissional para profissional. E a tentativa deu tão certo que eles perceberam o grande potencial da ferramenta, lançando-a no GitHub como um software livre (CIA WEBSITES, 2017).

4.2.4 PRIMENG

PrimeNG é uma rica coleção de componentes de UI para Angular. Todos os *widgets* são *open source* e gratuitos para uso sob licença MIT. O PrimeNG é desenvolvido pela PrimeTek Informatics, um fornecedor com anos de experiência no desenvolvimento de soluções de UI de código aberto (PrimeTek).

Prime NG é uma biblioteca de componentes angular, responsável por fornecer diversas facilidades no desenvolvimento com o *framework Javascript*. Por meio desta biblioteca é possível ter acesso tanto a padrões de estilo, quanto a componentes avançados que se comunicam com o *framework Angular*, como tabelas, menus, avisos personalizados e diversos outros componentes para desenvolvimento web. (PrimeTek)

4.3 JSON

Para que os dados pudessem ser trafegados, foi utilizado o formato de arquivos JSON (*JavaScript Object Notation*). Por meio deste formato os dados são estruturados em pares chave/valor onde podem ser armazenados (de maneira textual) números, letras, vetores e objetos complexos (que possam ser representados através da estrutura desta notação).

JSON é em formato texto e completamente independente de linguagem, pois usa convenções que são familiares às linguagens C e familiares, incluindo C++, C#, Java, JavaScript, Perl, Python e muitas outras. (JSON ORG)

5. PROPOSTA E DESENVOLVIMENTO DO TRABALHO

Neste capítulo será apresentado o problema a ser resolvido, a arquitetura do projeto desenvolvido, tecnologias dependentes e como a união destas podem solucionar um problema específico. A solução deste problema constitui-se na implementação de uma ferramenta de geração de *widgets* personalizados, os quais permitem que um usuário pleno domínio sobre quais e como os dados serão apresentados em um *dashboard* dinâmico.

5.1 DEFINIÇÃO DO PROBLEMA

A definição do problema constituiu-se sobre uma implementação baseado nas tecnologias *WEB* (HTML, Javascript e CSS) para a construção do *design*, estrutura e comunicação dos dados entre o próprio *dashboard*. Para a entrada e saída dos dados, será utilizada uma implementação de microsserviços de *business intelligence*, que fornecerá e receberá via protocolo HTTP, toda a informação necessária para ser processada e persistida após o uso desta ferramenta.

O desenvolvimento desta ferramenta tem como objetivo fornecer facilidades de acesso e visualização a dados para nossa instituição acadêmica, podendo ser utilizado nos mais variados setores. O Foco principal deste projeto é implementar uma ferramenta que permita a criação de *widgets* customizáveis, tanto a nível de estética quanto funcionalidade e propósito.

5.2 ARQUITETURA DA SOLUÇÃO

A modelagem de arquitetura desta solução é ilustrada na figura 3. Nesta figura são exibidos todos os membros e suas relações. Após a imagem serão também descritas as responsabilidades de cada membro e os dois principais fluxos de processamento que são: Recuperação de Dados Persistidos e o Gerenciamento de Widgets.

Um usuário acessa a ferramenta de criação de *widgets* e assim que as rotinas de criação ou alteração se encerram, uma requisição HTTP é feita para um servidor de *Business Intelligence*, onde os dados do *widget* do usuário são persistidos. A partir deste ponto, assim que o *Dashboard* Dinâmico for acessado, uma nova requisição será feita buscando todos os *widgets* salvos anteriormente.

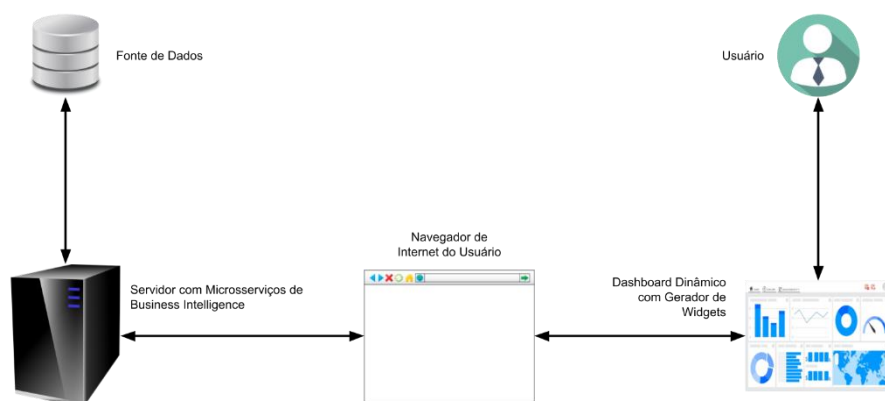


Figura 3: Arquitetura da Solução

5.2.1 Recuperação De Dados Persistidos

O Usuário acessa uma página em uma URL onde a aplicação de *Dashboard Dinâmico* e o Gerador de *Widgets* estão localizados. Assim que a página estiver sendo carregada, através de Javascript, uma requisição HTTP GET será feita à um servidor de microserviços de *business intelligence*, que retornará dados do *dashboard*, obtidos de uma fonte de dados, como por exemplo, um SGDB.

Após o êxito da requisição GET e a obtenção dos dados pelo navegador do usuário, as informações passarão por uma etapa de processamento onde os dados recebidos através de um arquivo JSON (*Javascript Object Notation*) serão reestruturadas para que possam funcionar ao ambiente do componente Angular 2 Gridster.

Na figura 4 é exibido o método responsável pela conversão de um objeto oriundo de um servidor HTTP para um objeto de outra classe, para que ocorra o bom funcionamento de todos os recursos do *Dashboard Dinâmico*.

```

public desempacotarWidget(item: GridsterItemVO): GridsterItem {
  let objetoGridsterItem: GridsterItem;
  objetoGridsterItem = {
    id: item.id,
    dashboard: item.dashboard,
    dragEnabled: item.dragEnabled,
    resizeEnabled: item.dragEnabled,
    cols: item.cols,
    rows: item.rows,
    x: item.x,
    y: item.y,
    titulo: item.titulo,
    tipoGrafico: item.tipoGrafico,
    view: item.view,
    cor: item.cor
  };
  return objetoGridsterItem;
}

```

Figura 4: Transformação de Dados para o Dashboard Dinâmico

5.2.2 Gerenciamento De Widgets

Assim que todos os dados dos *widgets* (já processados) são carregados, o gerador de *widget* passa as informações para o *dashboard dinâmico* e então todos os *widgets* são exibidos de acordo com as características específicas de cada *widget*. Assim que todas as rotinas de carregamento e renderização forem finalizadas, o usuário poderá realizar 3 tipos de operações: Criação de *Widgets*, Alteração de *Widgets* e Remoção de *Widgets*.

O Gerador de *widgets* proverá diversos recursos de personalização tais como: A informação que será exibida através do *widget*; o tipo de gráfico pelo qual a informação será exibida; o título e as posições do *widget* no *dashboard*; as cores de fundo e dos recursos das métricas de cada *widget*.

5.2.2.1 Criação De Widget

Na tela inicial do *dashboard dinâmico* é implementado um botão de criação de *widget* que por meio de um clique, abrirá uma janela para a organização dos dados de entrada, tais como título; tipo de gráfico/métrica; serviço de informação a ser utilizado; cores que serão utilizados para *widget* e gráfico. A representação desse a tela é exibida na figura 5.



A imagem mostra uma interface de usuário para a criação de um widget. O formulário é dividido em seções:

- Criação de Widget**: Título da seção.
- Título do Widget**: Campo de texto para o título.
- Tipo de Métrica**: Seção com quatro opções de gráfico/métrica:
 - Numérico**: Representado por um ícone de uma seta apontando para baixo com o número 1/9.
 - Barra**: Representado por um ícone de gráfico de barras.
 - Pizza**: Representado por um ícone de gráfico de pizza.
 - Linhas**: Representado por um ícone de gráfico de linhas.
- Serviço**: Campo de texto para o serviço.
- Cor do Widget**: Campo de cor com um pequeno quadrado branco para seleção.
- Criar**: Botão azul para finalizar a criação.

Figura 5: Tela de Criação de Widgets antes da seleção de serviço

Assim que houver um clique no tipo de gráfico, uma nova janela será aberta, exibindo todos os serviços de informação disponíveis para aquele tipo de gráfico. Esta janela de seleção de serviço está exibida na figura 6.

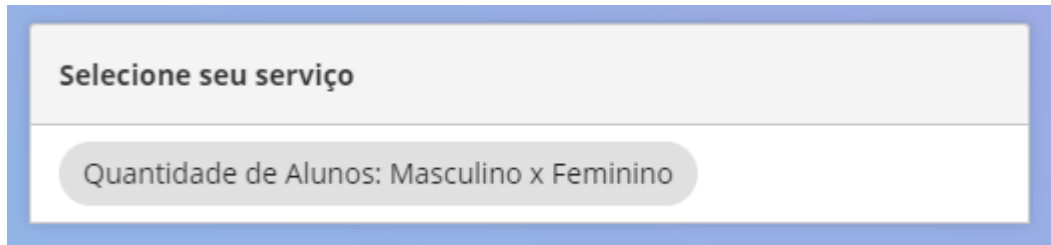


Figura 6: Janela de seleção de serviço

Após usuário selecionar o tipo de informação, esta janela será fechada e alguns campos para seleção de cores serão adicionadas, dependendo do tipo de gráfico escolhido como exibido na figura 7.

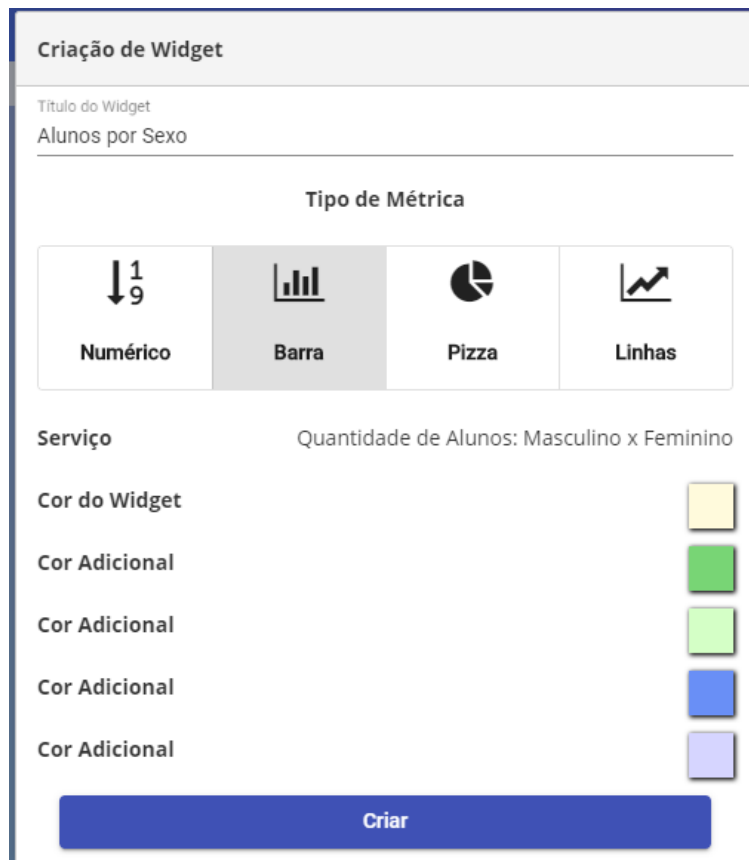


Figura 7: Tela de Criação de Widgets

Durante a criação do *widget* duas operações serão realizadas, através de uma emissão de evento de *Javascript* que será capturado pelo *dashboard dinâmico*, que são a inserção dos dados no vetor de *widgets* do *dashboard*, e uma requisição POST ao servidor de microsserviços com os novos dados. Após todos os campos e requisitos serem preenchidos o *widget* poderá ser criado e o *dashboard* receberá um novo item como exibido na figura 8.

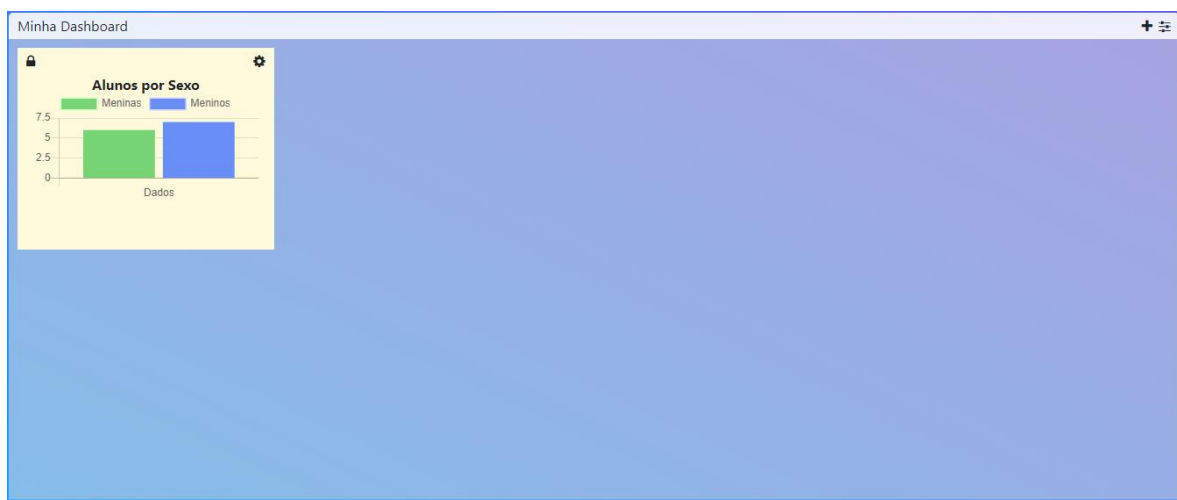


Figura 8: Dashboard após a criação do novo widget

5.2.2.2 Alteração De Widget

Para realizar alteração de um *widget*, o usuário deverá clicar no ícone de opção no próprio *widget* e selecionar a paleta de alteração. Após isso uma janela de alteração será exibida, mostrando os campos de título e as cores que poderão ser alteradas de acordo com a preferência do usuário. Assim que os dados forem alterados, a operação poderá ser concluída, gerando uma emissão de evento de *Javascript*, requisitando ao *dashboard*, que realize uma requisição PUT para concluir a atualização do *widget*.

Caso usuário deseje alterar a informação da métrica exibida, será necessário que ele remova o *widget* do *dashboard*, através da opção de exclusão na tela de configuração e crie um *widget* com o serviço desejado. A figura a seguir demonstra a janela de alteração.

Alteração de Widget
Confirmar Alteração

Título do Widget
Alunos por Sexo

Cor do Widget

Cor Adicional

Cor Adicional

Cor Adicional

Cor Adicional

Figura 9: Janela de alteração de Widget

5.2.2.3 Remoção De Widget

Para realizar a remoção de um *widget*, é necessário que usuário acesse a tela de configuração do *widget* e selecione a opção de remoção, localizada através de um ícone no canto superior direito. Após o clique será exibido uma janela para confirmação e após isso o *widget* será removido. Após a remoção visual do *widget*, um evento de *Javascript* será emitido ao *dashboard* para que uma requisição DELETE seja realizada, finalizando assim, a rotina de remoção.

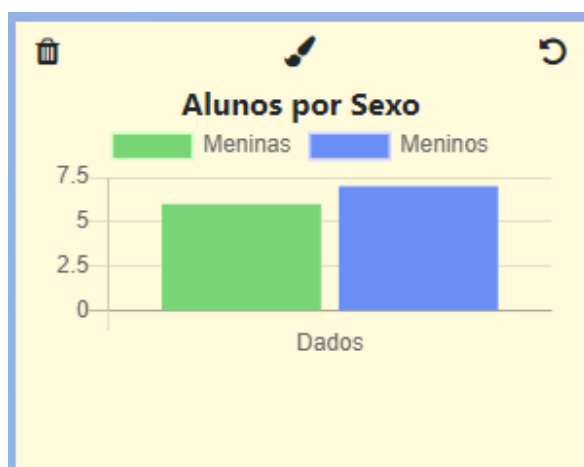


Figura 10: Tela de Configuração de Widget

5.2.2.4 Callbacks

O *dashboard dinâmico*, por meio do componente Angular 2 Gridster, fornece a possibilidade de escutar alterações no campo de *widgets*, como por exemplo: Criação de novos *widgets*; remoção de *widgets*; redimensionamento e movimentação de *widgets* em uma “espécie” de plano cartesiano. Para que todas as funcionalidades desta solução possam funcionar com êxito, é necessário que algumas funções estáticas sejam injetadas no objeto de configuração do *dashboard dinâmico*. O desenvolvimento de métodos estáticos pode causar algumas limitações relacionadas à injeção de dependência, recurso necessário na construção de classes de serviços para requisições HTTP. Para contornar o problema, uma classe de eventos estáticos foi implementada, possibilitando assim a comunicação de um escopo estático para um método de uma instância de classe, através de um receptor (*listener*) codificado dentro do trecho de código implementado da interface *onInit* do *framework* Angular.

```
public static itemAlterado(item: GridsterItem) {  
    Eventos.atualizarWidget.emit(item);  
}  
public static itemRemovido(item: GridsterItem) {  
    Eventos.removerWidget.emit(item);  
}
```

Figura 11: Métodos estáticos emissores de eventos

5.2.2.5 Transformação De Dados

De forma semelhante a transformação dos dados anterior, uma outra rotina foi implementada para as alterações dos dados de cada *widget*. É necessário que os dados estruturados para o funcionamento do *dashboard dinâmico* sejam reestruturados de forma que possam ser lidos e persistidos de maneira correta. A figura a seguir mostra este trecho de código responsável pela conversão de dados no formato do *dashboard dinâmico* para o formato utilizado pelos microsserviços de *business intelligence*.

```
public empacotarWidget(item: GridsterItem): GridsterItemVO {
  const objetoGridsterItemVO = new GridsterItemVO();
  objetoGridsterItemVO.id = item.id;
  objetoGridsterItemVO.dashboard = item.dashboard;
  objetoGridsterItemVO.cols = item.cols;
  objetoGridsterItemVO.rows = item.rows;
  objetoGridsterItemVO.x = item.x;
  objetoGridsterItemVO.y = item.y;
  objetoGridsterItemVO.dragEnabled = item.dragEnabled;
  objetoGridsterItemVO.resizeEnabled = item.resizeEnabled;
  objetoGridsterItemVO.tipoGrafico = item.tipoGrafico;
  objetoGridsterItemVO.titulo = item.titulo;
  objetoGridsterItemVO.view = item.view;
  objetoGridsterItemVO.cor = item.cor;
  return objetoGridsterItemVO;
}
```

Figura 12: Transformação de Dados para os Microsserviços de Business Intelligence

CONCLUSÃO

Mediante a conclusão das principais rotinas de processamento deste projeto, o conceito de “criação flexível” torna-se realmente tangível, proporcionando grande liberdade ao usuário para organizar seus *widgets* em um *dashboard* da maneira como bem entender, uma real experiência de criação de *design*. Foi possível, através de fontes de serviços de persistência de *widgets*, salvar todas as criações e alterações visuais feitas por um usuário, bem como os dados específicos para a geração das métricas. A linguagem *Javascript* mostrou um grande desempenho durante todo o desenvolvimento deste trabalho e teve grande importância na estruturação dos dados dos componentes de criação, alteração e renderização de métricas.

A evolução das *Tecnologias WEB*, proporcionaram possibilidades para a construção de sistemas mais complexos e visualmente agradáveis aos usuários. Unir estes conceitos à capacidade de gerar um *UX* e *UI* específico e íntimo será de grande valor ao usuário, que ainda carece de tal liberdade em suas plataformas de gestão.

Com o estudo sobre a interação humano-computador, *design*, experiência de usuário e *design de interfaces*, foi possível perceber a importância de temas antes não muito explorados e sutis que, no entanto, exercem papel fundamental na construção de soluções que realmente chamem atenção ou proporcionem uma boa experiência de uso aos seus usuários.

A partir do estado deste projeto é possível que versões mais novas e mais otimizadas sejam implementadas, bem como novos recursos como: Alertas para avisar ao usuário sobre um novo serviço disponível; Novos tipos de métricas; novas possibilidades de customização de estilos como, cores de título e novas fontes.

Através desta pesquisa e implementação é esperado que o projeto continue em desenvolvimento, através de implantações de novos módulos de customização e refatorações, para que o código-fonte continue claro e limpo para novas modificações.

REFERÊNCIAS

AD-DIGITAL. **Superando as Deficiências do Dashboard: Usando a automação para obter insights claros e acionáveis.** Disponível em <<http://ad-digitaltecnologia.com.br/materials/Business-Intelligence-Whitepaper-Ai.pdf>>. Acesso em 12 mar. 2019.

ALGAWORKS. **Bate-papo sobre Angular 2 e PrimeNG com Çağatay Çivici.** Disponível em <<https://blog.algaworks.com/angular2-primeng/>>. Acesso em 18 de mar. 2018.

ALGAWORKS. **O que é Angular?** Disponível em <<https://blog.algaworks.com/o-que-e-angular/>>. Acesso em 18 de mar. 2018.

ANGULAR. **What is Angular?** Disponível em <<https://angular.io/docs>>. Acesso em 18 mar. 2019.

BARBOSA, Simone Diniz Junqueira. SILVA, Bruno Santana da Silva. **Interação Humano-Computador.** 1ª Edição. 2010

BEVAN, Nigel. **Extending Quality in Use to Provide a Framework for Usability Measurement.** Human-Computer Interaction 10, pp. 13-22, 2009.

BEYON, David. **Interação Humano-Computador,** 2ª Edição. Tradução de Heloisa Coimbra de Souza. Editora Addison Wesley, 2011.

BRATH, Richard. PETERS, Michael. **Dashboard Design: Why Design is Important.** Disponível em <http://cs.furman.edu/~pbatchelor/csc105/articles/TUN_DM_ONLINE.pdf>. Acesso em 11 mar. 2019.

CIA WEBSITES. **O que é Bootstrap e Para Que Serve?** Disponível em <<https://www.ciawebsites.com.br/dicas-e-tutoriais/o-que-e-bootstrap/>>. Acesso em 18 de mar. 2019.

DEVMEDIA. **A origem do CSS, um pouco da história.** Disponível em <<https://www.devmedia.com.br/a-origem-do-css-um-pouco-da-historia/15195>>. Acesso em 18 de mar. 2019.

FACULDADES INTEGRADAS DE OURINHOS. **Cronograma.** Disponível em <http://fio.edu.br/manualtcc/co/8_Cronograma.html>. Acesso em 29 out. 2018.

FONSECA, J. J. S. **Metodologia da pesquisa científica.** Fortaleza: UEC, 2002. Apostila.

GERHARDT, Tatiana Engel; SILVEIRA, Denise Tolfo. **Métodos de Pesquisa**. 1ª Edição. UFMG.

GRIGO-MCMAHON, Murray. **Dashboards, Modelos e um pouco de lama**. Disponível em <<https://ecommercenews.com.br/artigos/dicas-artigos/dashboards-modelos-e-um-pouco-de-lama/>>. Acesso em 15 out. 2018.

HEWETT, Thomas. Baecker Ronald. CARD, Stuart. CAREY, Tom. GASEN, Jean. MANTEI, Marilyn. PERLMAN, Gary. STRONG, Gary. VERPLANK, William. **Curricula for Human-Computer Interaction**. New York, NY, USA ©1992.

HIX, Deborah. HARTSON, H. Rex. **Developing User Interfaces: Ensuring Usability Through Product & Process (Wiley Professional Computing)**. New York, NY, USA 1993.

HOSTINGER. **O que é HTML? Guia Básico para Iniciantes**. Disponível em <<https://www.hostinger.com.br/tutoriais/o-que-e-html-conceitos-basicos/#A-Historia-do-HTML>>. Acesso em 18 de mar. 2018.

HOSTINGER. **O que é Javascript?** Disponível em <<https://www.hostinger.com.br/tutoriais/o-que-e-javascript/>>. Acesso em 18 de mar. 2018.

IMASTERS. **Javascript – 20 anos de história e construção da web**. Disponível em <<https://imasters.com.br/front-end/javascript-20-anos-de-historia-e-construcao-da-web>>. Acesso em 18 de mar. 2018.

INTERNET INNOVATION. **Saiba o que é KPI e entenda sua importância para análises em marketing digital**. Disponível em <<https://www.internetinnovation.com.br/blog/saiba-o-que-e-kpi-e-entenda-sua-importancia-para-analises-em-marketing-digital/>>. Acesso em 13 mar. 2019.

JSON ORG. **Introdução ao JSON**. Disponível em <<https://www.json.org/json-pt.html>>. Acesso em 22 de ago. 2019.

LAWSON, Bryan. **How Designer Think: The Design Process Demystified**, 4ª Edição, Oxford UK: Architectural Press, 2006.

LÖWGREN, Jonas. STOLTERMAN, Erik. **Thoughtful Interaction Design: A Design Perspective on Information Technology**. Cambridge, MA. The MIT Press, 2004.

MOTTA, Gabriel. **Métricas e KPIs: entenda a diferença**. Disponível em <<https://blog.ingagedigital.com.br/metricas-e-kpis-entenda-a-diferenca/>>. Acesso em 13 mar. 2019

NILSEN, Jakob. **Usability Engineering**. New York, NY: Academic Press, 1993.

PITON, Rafael. **O que é Dashboard?** Disponível em <<https://rafaelpiton.com.br/blog/o-que-e-dashboard/>>. Acesso em 15 out. 2018.

PRIMETEK. **Why PrimeNG?** Disponível em <<https://www.primefaces.org/primeng/#/>>. Acesso em 18 de mar. 2018.

SERASA EXPERIAN. **Mais de 90% das empresas brasileiras consideram dados na definição da estratégia de negócios, aponta Serasa Experian.** Disponível em <<https://www.serasaexperian.com.br/sala-de-imprensa/mais-de-90-das-empresas-brasileiras-consideram-dados-na-definicao-da-estrategia-de-negocios-aponta-serasa-experian>>. Acesso em 15 out. 2018.

SHARP, Helen. ROGERS, Yvonne. PREECE, Jenny. **Interaction Design: Beyond Human-Computer Interaction**, 2ª Edição. New York, NY: John Wiley & Sons, 2007.

VENKI. **Conheça 10 Indicadores de desempenho de processos (KPI).** Disponível em <<http://www.venki.com.br/blog/indicadores-de-desempenho-de-processos/>>. Acesso em 15 out. 2018.