

# SmartNode Dashboard: um framework front-end baseado em Node-RED para criação de City Dashboards

Cesimar X. S. Dias<sup>1</sup>, Frederico Lopes<sup>2</sup>, Jair C. Leite<sup>2</sup>

<sup>1</sup>*Campus Natal Zona-Norte*  
Instituto Federal do Rio Grande do Norte (IFRN) – Natal – RN – Brazil

<sup>2</sup>Instituto Metr pole Digital  
Universidade Federal do Rio Grande do Norte (UFRN) – Natal, RN – Brazil

cesimar.xavier@ifrn.edu.br, fred@imd.ufrn.br, jair@dimap.ufrn.br

**Abstract.** *This work presents the SmartNode Dashboard (SND), a tool based on front-end technologies for creation of intelligent dashboards. SND has implemented design and interface web standards focused on content with reusable structures. In addition, a case study was created using Node-RED as an execution platform. As a result of this work, SmartNode Dashboard was developed, a framework for creating standardized and extensible interfaces. In addition, it offers dashboard developers a methodology for using the framework with Node-RED to facilitate and extend teams' ability to perform performance, time and quality in creating dashboards.*

**Resumo.** *Este trabalho apresenta o SmartNode Dashboard (SND), uma ferramenta baseada em tecnologias front-end para cria o de dashboards para cidades inteligentes. O SND implementou padr es de projetos e de interface web focados em conte dos com estruturas reaproveit veis. Al m disso, foi criado um caso de estudo utilizando o Node-RED como plataforma de execu o. Como resultados deste trabalho, foi desenvolvido o SmartNode Dashboard, um framework para cria o de interfaces padronizadas e extens veis. Al m disso, ele oferece aos desenvolvedores de dashboards uma metodologia de implanta o do framework com o Node-RED para facilitar e ampliar a capacidade das equipes no tocante ao desempenho, tempo e qualidade na cria o de dashboards.*

## 1. Introdu o

A Uni o das Na es Unidas (ONU) estima que at  2050 a popula o mundial ter  alcan ado o patamar de 9 bilh es de habitantes. Um n mero alarmante quando considerada a grande demanda de insumos que uma popula o t o grande despender  para se manter. Da mesma maneira, estima-se que mais de 50% da popula o mundial viva em  reas urbanas, atualmente. No leste europeu, por exemplo, enfrentar  uma situa o mutuamente dram tica, pois j  se sabe que 80% da popula o dessa regi o viver  nas  reas urbanas at  o ano de 2020 [Lopes and Oliveira 2017]. Esse panorama traz novos desafios  s cidades, traz consigo a necessidade delas se reinventarem para enfrentar esses desafios que   transformar espa os urbanos em lugares mais inovadores, participativos, conectados e sustent veis, sem negligenciar a qualidade de vida da popula o.

O panorama defronte a dram tica massa urbana representa desafios cruciais para a log stica da gest o p blica. Com o avan o das tecnologias e a urg ncia de refletir sobre

como estão sendo usados os espaços urbanos, sem negligenciar a sustentabilidade, várias cidades ao redor do mundo estão trabalhando continuamente em pesquisas com foco em soluções eficientes. Como reflexo disso, o conceito de cidade inteligente vem ganhando cada vez mais notoriedade no cenário atual. [Macke et al. 2018] afirma que até o ano de 2020 40 cidades serão consideradas inteligentes, e que em 2025 esse número passará para 88 cidades. Dentre essas cidades, duas capitais brasileiras - Rio de Janeiro e Curitiba, deverão compor o grupo das 8 cidades mais inteligentes da América Latina.

Hoje, um grande número de cidades desenvolvem e mantêm projetos, com ferramentas próprias ou compartilhadas, na tentativa de minimizar os esforços despendidos com os problemas enfrentados nos grandes centros. O governo de Amsterdã, por exemplo, se empenhou em contribuir no desenvolvimento do *City Service Development Kit (CitySDK)*<sup>1</sup>. Paralelamente a isso, a *University of New South Wales* (Universidade de Nova Gales do Sul) - UNSW, na Faculdade de Ambientes Construídos, foi desenvolvido o *City Futures Research Centre* (Centro de Pesquisa para Cidades do Futuro). Este centro é atualmente referência nacional pelas suas contribuições acerca das pesquisas urbanas aplicadas. No mesmo sentido, a cidade de Glasgow, na Inglaterra, desenvolve o projeto *Future City Glasgow*. Um projeto com foco em apresentar o potencial de se trabalhar com dados abertos gerados pelas cidades, explorando a capacidade do seu uso na criação de ideias inovadoras. Projetos como esses revelam que as cidades estão engajadas na busca por soluções mais eficazes para demandas recorrentes e cada vez mais latentes no ambiente de *smart cities*.

As cidades supracitadas disponibilizam *dashboards* para apresentar informações aos cidadãos, com cada solução possuindo suas particularidades, mas não compartilhando seus métodos de construção. Nesse cenário, este trabalho apresenta o *SmartNode Dashboard (SND)*, um *framework* que facilita a criação de *dashboards* de informações. Ele foi projetado a partir da junção de tecnologias *front-end* e uma ferramenta de programação por fluxos, o *Node-RED*<sup>2</sup>. Por focar em cidades, foram criados alguns *cards* com dados para parâmetros urbanos. Seu intuito é oferecer às equipes de desenvolvimento de software, mais eficiência, padronização, flexibilidade, agilidade e reutilização de códigos para construção de novos projetos de *city dashboards*. Seu uso não se restringe às cidades, pois sua codificação é bastante simples, possibilitando programadores customizar de acordo com a necessidade específica.

O restante deste artigo está estruturado da seguinte forma. A Seção 2 apresenta o embasamento teórico necessário para o desenvolvimento do presente trabalho. A Seção 3 apresenta o *SmartNode Dashboard* e suas funcionalidades. Na Seção 4 é exposta como foi realizado o caso de estudo utilizando o *Node-RED*, Na Seção 5 apresenta como foi realizada a validação do trabalho. Por fim, a Seção 6 traz algumas considerações finais.

## 2. Embasamento Teórico

Essa seção apresenta alguns conceitos necessários para o entendimento desse artigo.

---

<sup>1</sup>Coleta dados abertos e fornece uma maneira uniforme de apresentar esses dados

<sup>2</sup>O *Node-RED* é uma ferramenta para conectar virtualmente os dispositivos, APIs e serviços online.

## 2.1. Smart City

O termo "Cidade Inteligente" teve seus primeiros registros no início da década de 90, porém aplicado geralmente no contexto do planejamento urbano, construção civil e arquitetura. Apenas nos últimos anos é que as tecnologias estão sendo empregadas à sistemas de informação na integração das operações de serviços e infraestrutura urbana [Falcao et al. 2012]. Esse termo foi cunhado a fim de conceituar o fenômeno de desenvolvimento das cidades, que cada vez mais se tornavam dependentes de tecnologia, inovação e globalização, principalmente em uma perspectiva econômica [Rizzon et al. 2017]. Para [Leem and Kim 2013], o fator central para mudança dos paradigmas das cidades está relacionado diretamente com as Tecnologias de Informação e Comunicação (TIC). A grande utilização das TICs nas cidades, na busca por promover melhorias estruturais e que consiga entregar, de forma precisa e dinâmica, comodidade aos cidadãos, vários autores indicam que uma *Smart City* pode ser definida principalmente, pelo amplo emprego de TICs em infraestruturas tradicionais, bem como para melhorar a participação ativa de capital humano e social [Rizzon et al. 2017]. Essa noção está intimamente ligada à economia do conhecimento, às mudanças na aglomeração espacial e ao desenvolvimento urbano baseado no conhecimento [Macke et al. 2018]. De uma forma simples, o conceito de *smart cities*, ou cidades inteligentes, pode ser definido pelo amplo uso das tecnologias com intuito de melhorar a infraestrutura das cidades e transformar os centros urbanos em locais mais eficientes e melhores de se viver.

## 2.2. Dashboard

[Kourtit and Nijkamp 2018] explicam que um *dashboard* é uma ferramenta de gerenciamento estratégico que usa fatores críticos de sucesso e indicadores-chave de desempenho para traduzir a missão e a estratégia de uma organização em um conjunto equilibrado de medidas de desempenho integradas à ações concretas relacionadas. O atual volume de dados e informações coletadas, armazenadas, compartilhadas e usadas nos centros urbanos, são quase ilimitados [Kourtit and Nijkamp 2018]. Os dados oriundos do monitoramento das cidades podem ser sumarizados numa estrutura e disponibilizados para a visualização, servindo como ferramenta de tomada de decisões. Esses dados podem ser alimentados em diversas formas de fontes, espalhadas por uma cidade, que podem registrar informação e enviar para uma base concentradora de dados. A informação sobre previsão do tempo, poluição do ar, o nível das águas dos rios, indicador econômico, até mesmo as condições do tráfego, podem ser exibidas num único painel [Suakanto et al. 2013].

### 2.2.1. Tipos de Dashboards

Com a grande diversificação de contextos, inúmeros tipos diferentes de *dashboards* se tornaram populares. Dentre eles existem três tipos principais que são os mais utilizados.

**Dashboards operacionais:** exibem informações que facilitam as operações do dia a dia de uma empresa. Os objetivos comuns podem ser monitorar o tempo de atividade do servidor, as vendas diárias, as chamadas diárias feitas ou os compromissos reservados.

**Dashboards estratégicos e executivos:** o painel estratégico se concentra em fornecer uma visão geral de alto nível do estado do negócio e aborda as principais mudanças que o negócio cria.

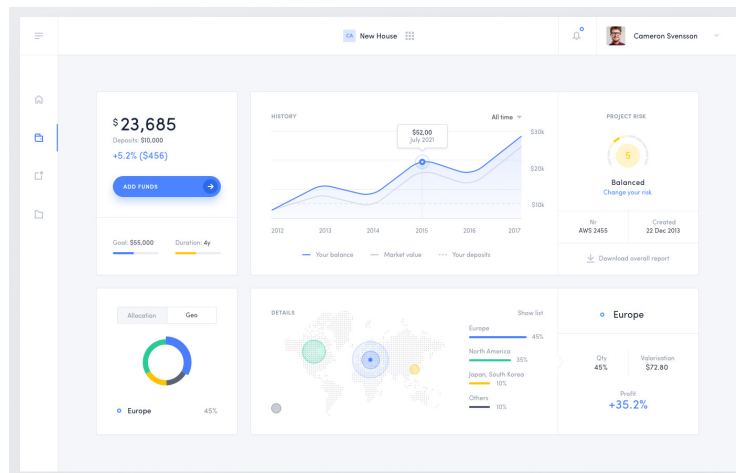


Figura 1. *Dashboard* com resumo de informações

**Dashboards analíticos:** exibe dados operacionais ou estratégicos - ou ambos. O painel analítico oferece funcionalidades de detalhamento, permitindo que os usuários explorem dados em maiores detalhes.

### 2.3. Padrões de Projeto

Para [Cybis et al. 2015], "padrões se referem aos problemas mais comuns e às boas soluções para esses problemas, que são capturadas, compartilhadas e reutilizadas por profissionais atuando no projeto de diversos tipos de produtos e serviços". Seu uso aumenta a eficiência não somente na etapa de programação, mas também na utilização, pois oferece aos usuários usabilidade na interação com as interfaces. O arquiteto Christopher Alexander afirmou que um padrão descreve um problema e norteia a solução de modo que se possa utilizar esta solução inúmeras vezes sem nunca fazê-la da mesma forma [Gamma et al. 2000]. [Gamma et al. 2000] corroboram a asserção do arquiteto e explica que, apesar de Alexander relacionar suas proposições ao contexto da arquitetura, o cerne para ambos os tipos de padrões está a solução para um problema num determinado contexto.

## 3. O Framework SmartNode Dashboard

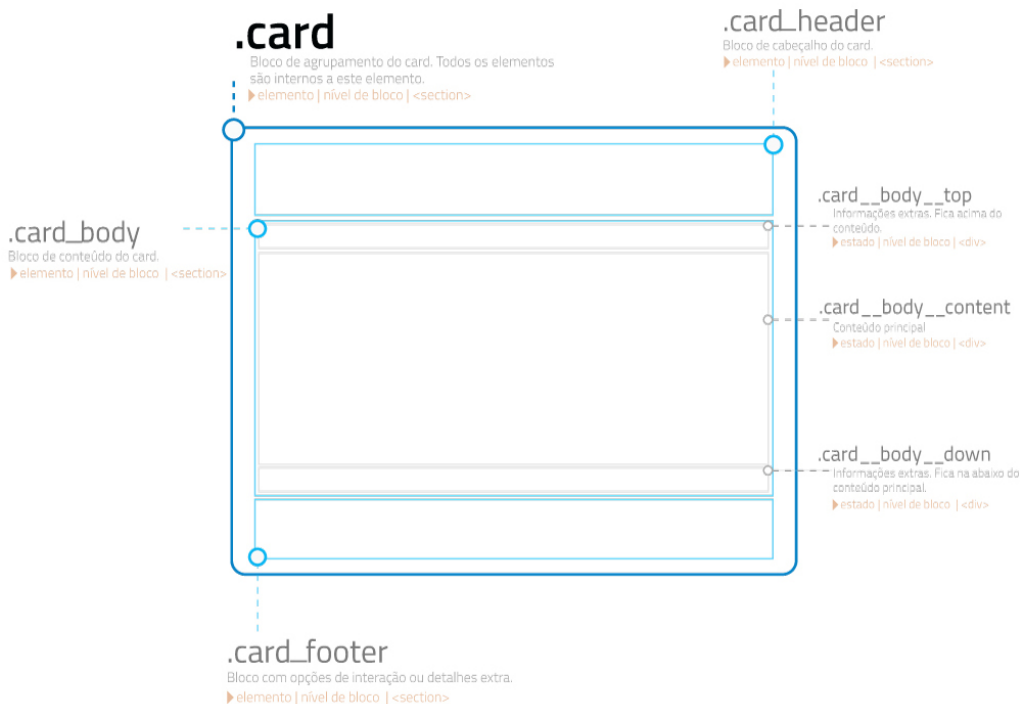
O *SmartNode Dashboard* (SND) consiste num *framework* CSS que é responsável pela padronização dos elementos de interfaces do *dashboard*. Por meio de sua estrutura customizada de código, fornece aos programadores a criação de interfaces mais eficazes e num curto espaço de tempo. O SND foi desenvolvido sobre a base de outro *framework front-end*, o *Bootstrap Framework*<sup>3</sup>.

### 3.1. Arquitetura do Framework

O SDN adota a arquitetura de formatação BEM<sup>4</sup>. Essa arquitetura organiza o código em três níveis: (i) bloco, (ii) elemento e (iii) modificador. O componente básico do *SmartNode Dashboard* é o *card* e dessa forma ele segue a estrutura mostrada na Figura 2.

<sup>3</sup>Bootstrap é uma biblioteca de componentes

<sup>4</sup>BEM significa Bloco, Elemento, Modificador. Metodologia de nomenclaturas de classes em CSS, para criar código reutilizável.



**Figura 2. Estrutura do Card utilizando a metodologia BEM**

Cada bloco do *card* tem uma função estrutural, que é representada por uma classe. **.card**: classe básica do componente *card*. É responsável pela formatação do bloco principal; **.card\_header**: classe responsável por formatar o bloco de cabeçalho; **.card\_body**: classe que formata os elementos de conteúdo. O bloco de conteúdo que é formatado por esta classe, contém três sub-blocos, que dividem o espaço de conteúdo e que pode ser formatado livremente. Os blocos internos são: **.card\_body\_top**, um bloco para conteúdo acima; **.card\_body\_content**, que é organizado para ser o elemento do conteúdo principal e **.card\_body\_down**, bloco para conteúdo complementar; **.card\_footer**: classe responsável pela formatação do elemento de rodapé. Esse bloco é voltado para incluir elementos de interação, como links.

Para estender as funcionalidades o bloco principal recebe uma classe adicional. Essa classe deve manter a padronização nos nomes e adicionar um modificador, conforme especificação do BEM.

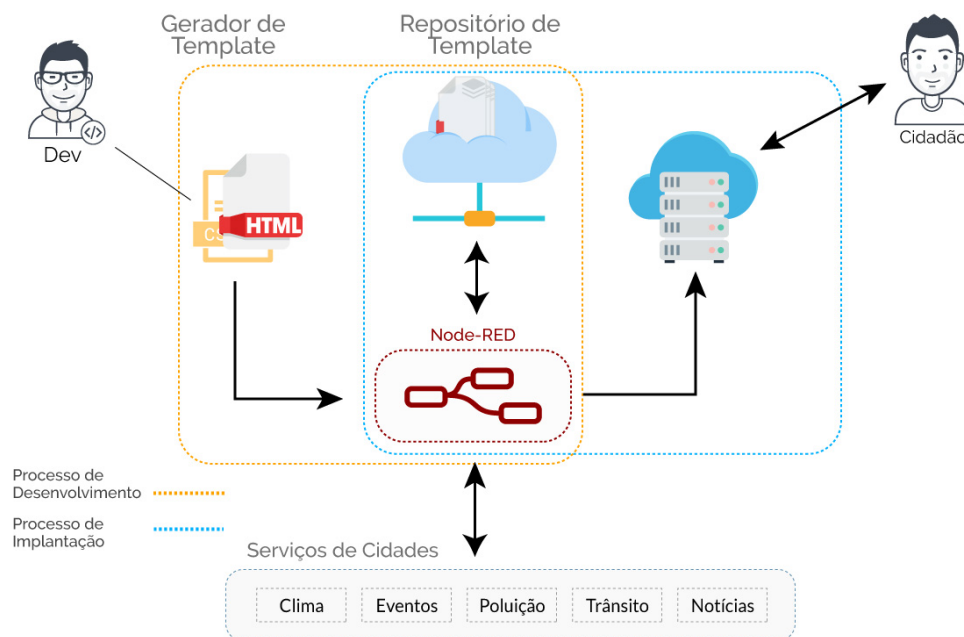
## 4. Metodologia de Uso do Framework

A Figura 3 mostra uma visão geral do funcionamento do processo necessário para atingir a organização da interface do *dashboard*. Existem dois processos: (i) processo de desenvolvimento e (ii) processo de implantação.

### 4.1. Processo de Desenvolvimento

Esse processo corresponde aos métodos de implementação de uma estrutura de código para o *dashboard*. Ela funciona em três etapas, sendo a terceira etapa opcional.

**Etapa 1. Criação do modelo de renderização:** no primeiro momento o desenvolvedor deve criar uma estrutura de código em HTML para servir de *template*. Esse



**Figura 3. Visão geral do funcionamento do SmartNode Dashboard**

código já utiliza o código CSS do *SmartNode Dashboard*. Ele é um código base para a organização da interface, sendo considerado uma prévia da estrutura final;

**Etapa 2. Implementação do fluxo:** em seguida é necessário utilizar o *node-RED* para criar um fluxo para funcionamento do *template* gerado na Etapa 1. Esse fluxo é composto pelo fluxo principal e diversos sub-fluxos. Cada sub-fluxo, contido nesse fluxo principal, representará um parâmetro (*card*) de visualização no *dashboard* (Figura 4). Cada sub-fluxo funciona com um par de nós, um para se conectar às fontes de dados e um segundo para receber e renderizar os dados. Dessa forma, o fluxo principal será dividido em vários fluxos menores e no fim eles se ligarão no último nó, que é um nó de configuração do layout do *dashboard*;

**Etapa 3. Compartilhamento de fluxo:** o compartilhamento do código do *template* é realizado através do repositório oficial do *Node-RED*. Onde o desenvolver poderá contribuir com a biblioteca de *flows* e disponibilizar seu código para outros programadores utilizarem ou ajudar na melhoria.

#### 4.2. Processo de Implantação

Esse processo é responsável por utilizar um fluxo compartilhado na biblioteca de fluxos do *Node-RED*. Para isso é necessário seguir três passos.

**Passo 1. Importação de fluxo:** executar a importação da estrutura compartilhada, a partir do repositório, utilizando a Gerência de Paletas (*Manage Palette*) da plataforma;

**Passo 2. Configuração dos serviços:** configurar as credenciais de acesso às APIs que disponibilizam os dados (APIs de previsão do tempo, por exemplo);

**Passo 3. Deploy e publicação:** esse passo corresponde a última atividade antes da disponibilização do *dashboard*. Para isso basta realizar a implantação (*deploy*) direto na plataforma.

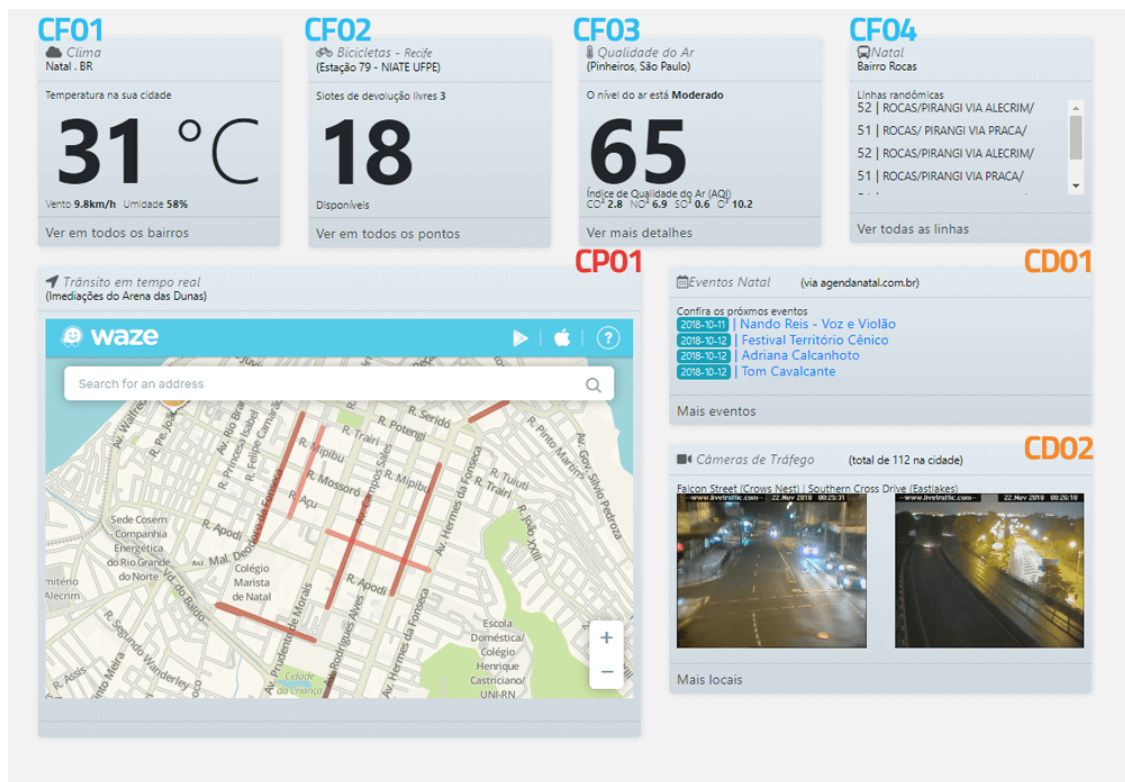


Figura 4. Protótipo a partir da implementação do *template* utilizando o *Smart-Node Dashboard*

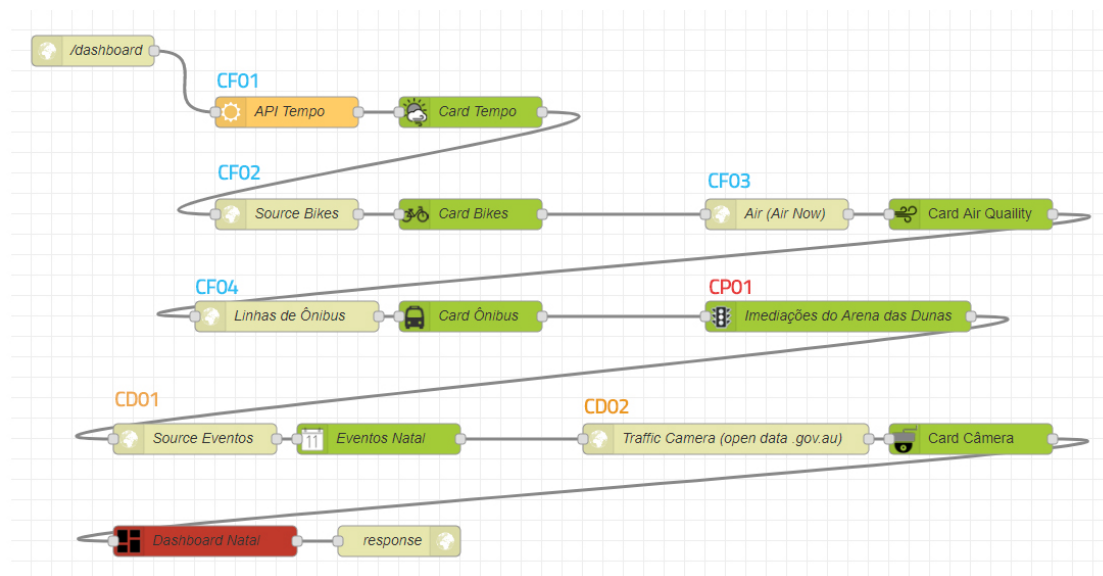
### 4.3. Integração ao *Node-RED*

A principal função do *Node-RED* no contexto desse projeto é abstrair a complexidade de uma infraestrutura robusta e de grande porte para executar o projeto de *city dashboard*. Isso é possível pela capacidade que essa infraestrutura oferece, além do seu padrão de programação por fluxos. Assim, é possível, de forma simples e rápida, a disponibilização do *template* em execução.

A Figura 5 apresenta um fluxo com alguns sub-fluxos do Processo de Implantação (Seção 4.2), que define a metodologia de implantação de um *template* lançando mão do *Node-RED*. A ideia é mostrar como são criados os fluxos para consumir os serviços de dados.

Cada sub-fluxo representa uma saída no *dashboard*. Cada saída é demarcada por um par de nós (nó de dados e nó de renderização) que origina um *card* de saída, conforme a Figura 4. Essa figura é composta por *cards* e estes apresentam informações sobre a cidade. Como é possível perceber, praticamente todos os nós de saída (identificados pela cor verde) são antecedidos de um nó que tem a função de ser a conexão com os dados, ou seja, são responsáveis por fornecer um meio de comunicação com a fonte de dados e tratá-los para que o nó de saída possa realizar a renderização.

O fluxo exibido na Figura 5, com os sub-fluxos **CF01**, **CF02**, **CF03**, **CF04**, **CP01** e **CD01**, **CD02** são responsáveis pelas saídas dos *cards* **CF01**, **CF02**, **CF03**, **CF04**, **CP01**, **CD01** e **CD02**, respectivamente, como apresentado na Figura 4.



**Figura 5. Fluxo para o Template**

## 5. Validação

Na validação deste trabalho foram utilizados os métodos de questionários e observação. Ela teve intenção de avaliar a experiência dos participantes ao construir um *dashboard* utilizando o SND. Os critérios avaliados foram a eficiência (quantidade de passos utilizados) e a eficácia (nível de conclusão alcançado). Desse modo, para validar o objetivo deste trabalho, a escolha dos participantes foi realizada na perspectiva de utilizar pessoas com experiências em desenvolvimento de software. Foram avaliados cinco pessoas, com experiência entre 5 e 30 anos de experiência. Estes foram submetidos ao processo de avaliação que consistiu em: responder um questionário pré-teste (com oito perguntas), realizar tarefas de construção de *dashboard* e responder um questionário pós-teste (com nove perguntas).

### 5.1. Resultados

No critério eficiência, onde se mensurou a quantidade de esforço que um participante levaria para concluir a tarefa, mostrou que o participante mais rápido levou 11 minutos para concluir e o mais demorado concluiu em 35 minutos. A média entre eles foi de 23 minutos na conclusão das tarefas.

No critério eficácia, foi avaliado o quanto os participantes se aproximaram da conclusão das tarefas. Dentre todos participantes, 40% não concluíram todas as atividades propostas.

### 5.2. Discussão

Apesar de aproximadamente 30% dos participantes terem enfrentado dificuldades ao realizar as tarefas, eles apontaram que a metodologia, aliada ao *SmartNode Dashboard*, era fácil de ser realizada. Na concepção dos avaliados, a dificuldade se deu por ser uma nova forma de pensar na montagem, por meio de montagem de fluxos. Esse conceito não ficou claro, apesar de ser simples.



## 6. Considerações Finais

Este trabalho apresentou a união do estado da arte nas tecnologias *front-end* para criar o *SmartNode Dashboard*, e alinhando a isso uma metodologia de implantação que pode lançar mão de outras ferramentas que simplificam o desenvolvimento de *dashboards*. Apesar de ter constructos finitos, esse trabalho apresenta a riqueza que é olhar para novas tecnologias que surgem constantemente, e que podem auxiliar e ampliar a capacidade que as equipes de desenvolvimento de software têm para solucionar problemas. Outros projetos com finalidade análoga a este trabalho, como os projetos de *dashboards* de Amsterdã, cidades do Reino Unido e Sidney, serviram de comparativo para o desenvolvimento deste. Sendo que aqueles projetos não mantinham um padrão, seja em tecnologias utilizadas ou padrão de desenvolvimento. Esses projetos foram criados apenas com intuito de exibir as informações de seus sistemas e cidades, mas não de criar um padrão de desenvolvimento ou que fosse capaz de serem reproduzidos. O SND foi construído para ser compartilhado e receber contribuições, por esse motivo encontra-se disponível através de um repositório, <https://github.com/cesimar/smart-node-dashboard>.

## Referências

- Cybis, W., Betiol, A. H., and Faust, R. (2015). *ERGONOMIA E USABILIDADE: CONHECIMENTOS, MÉTODOS E APLICAÇÕES*. Novatec, São Paulo, 3th edition.
- Falcao, A. G. R., Baptista, C. d. S., and Menezes, L. C. d. (2012). Crowd4city: Utilizando sensores humanos como fonte de dados em cidades inteligentes. In *VIII Simpósio Brasileiro de Sistemas de Informação*, pages 144–149.
- Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (2000). *Padrões de projeto: soluções reutilizáveis de software orientado a objetos*. Bookman, Porto Alegre.
- Kourtit, K. and Nijkamp, P. (2018). Big data dashboards as smart decision support tools for i-cities—an experiment on stockholm. *Land Use Policy*, 71:24–35.
- Leem, C. and Kim, B. P. (2013). Taxonomy of ubiquitous computing service for city development. *Personal and Ubiquitous Computing*, 17:1475 – 1483.
- Lopes, I. M. and Oliveira, P. (2017). Can a small city be considered a smart city? *Procedia Computer Science*, 121:617 – 624. CENTERIS 2017 - International Conference on ENTERprise Information Systems / ProjMAN 2017 - International Conference on Project MANagement / HCist 2017 - International Conference on Health and Social Care Information Systems and Technologies, CENTERIS/ProjMAN/HCist 2017.
- Macke, J., Casagrande, R. M., Sarate, J. A. R., and Silva, K. A. (2018). Smart city and quality of life: Citizens' perception in a brazilian case study. *Journal of Cleaner Production*, 182:717 – 726.
- Rizzon, F., Bertelli, J., Matte, J., Graebin, R., and Macke, J. (2017). Smart city: Um conceito em construção. In *Revista Metropolitana de Sustentabilidade*, pages 144–149.
- Suakanto, S., Supangkat, S. H., Saragih, R., et al. (2013). Smart city dashboard for integrating various data of sensor networks. In *ICT for Smart Society (ICISS), 2013 International Conference on*, pages 1–5. IEEE.