

Universidade do Minho Escola de Engenharia

Geração Automática de Interfaces de Utilizador para Aplicações Web

Dissertação

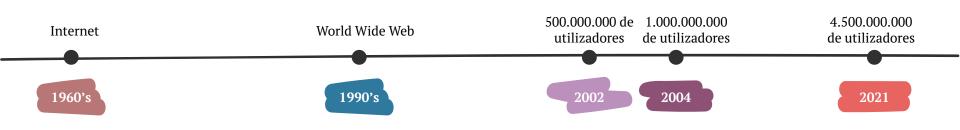
Catarina Araújo Machado - A81047 Orientador: José Creissac Campos

Agenda

- Contextualização
- Ferramentas
- Abordagem
- Implementação
- Conclusão

Contextualização

O **Desenvolvimento Web** é atualmente uma das principais áreas do **Desenvolvimento de Software**



HTML estático

Aplicações Web complexas

3

Motivação

- As Interfaces de Utilizador são subsistemas complexos que representam uma parte relevante do esforço de design e desenvolvimento
- *The rapid pace of innovation and rather uncontrolled, organic evolution of the Web (...) put the **developers in a complex position.**" (Taivalsaar & Mikkonen, 2017)
- (a) Conseguimos ajudar os designers e os programadores a lidar com esta complexidade?

Desenvolvimento de Interfaces de Utilizador baseado em Modelos

O Desenvolvimento baseado em Modelos tem como objetivo apoiar o Desenvolvimento de Software aumentando o nível de abstração do processo de desenvolvimento.

- Modelo captura o *design*;
- Implementação deriva do modelo (através de níveis de abstração/refinamento).

MBUID & UCD

Concepts and Task

• Contexto e requisitos

Abstract User *Interface* (AUI)

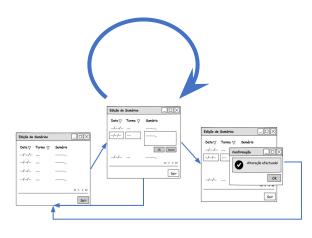
• Independente da Plataforma

Concrete User *Interface* (CUI)

• Dependente da Plataforma

Final User *Interface* (FUI)

• Dependente do Dispositivo



Protótipos da UI

Ferramentas

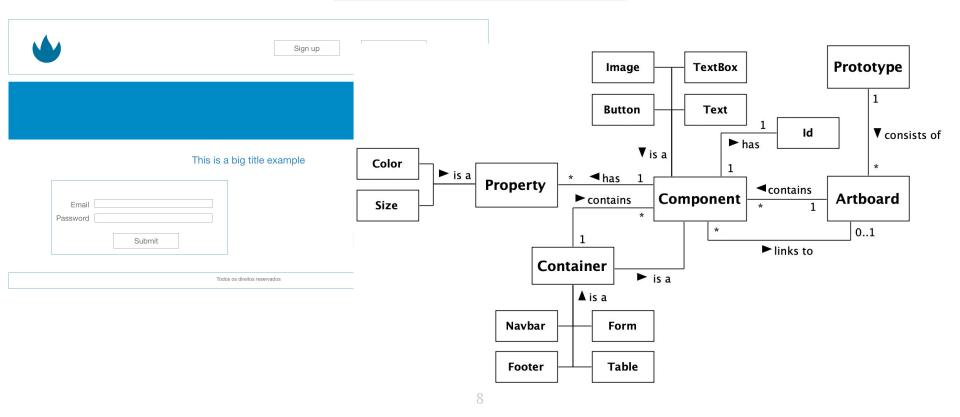
Desenvolvimento Protótipos



Framework Web

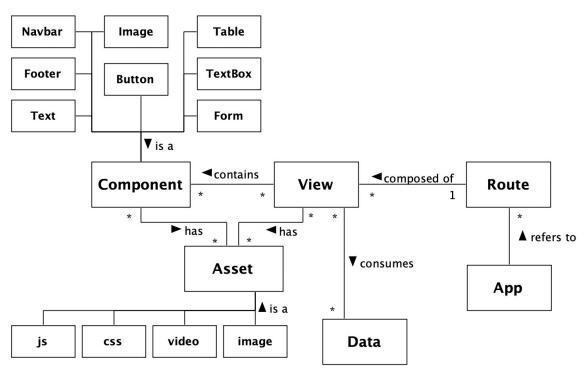


Estrutura de um Protótipo

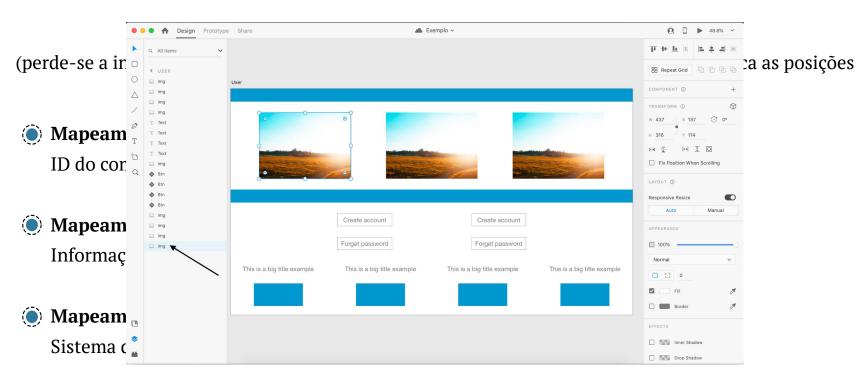


3. Ferramentas

Metamodelo da Arquitetura de Componentes Vue.js



De Protótipos a Modelos CUI



De Protótipos a Modelos CUI

- (in a componentes simples (texto, imagem, botão, caixa de texto)
- **© Componentes compostos** (*navbar*, *footer*, formulário)

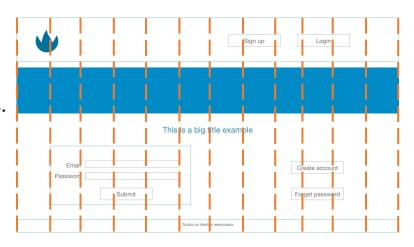
Requisitos

- 1 só projeto com as várias páginas, compostas por uma navbar e um footer comum a todas elas (não sendo obrigatório ter ambos) e com a respetiva navegação entre as páginas. Oportunidade de escolher qual das páginas é a homepage;
- Navbar e Footer em ficheiros à parte e reaproveitados para todas as páginas do projeto;
- © Componentes simples e componentes compostos (o designer terá que delimitar);
- © Componentes personalizados com cores (cor do texto, cor de fundo e cor do contorno de um componente simples ou composto). Informações geradas para um ficheiro .css;
- Posições relativas;
- Oportunidade de personalizar a largura do texto (colocando a informação no ID, no Adobe XD).

Algoritmo - CUI para FUI

Posicionamento:

- Dividir o protótipo em secções (linhas);
- Dividir o protótipo em colunas;
- Fazer o esqueleto das secções e colunas em Vue.js.



Componentes:

- Corresponder o ID do componente e o código do componente;
- Fazer as alterações necessárias à cor e possível navegação do componente.

Implementação

(Python)



"Parsing"



Traduzir um e um só ficheiro SVG para um ficheiro Vue.js

- Criar projeto Vue.js ("vue create");
- Interpretar ficheiro externo de *routing*;
- Correr o *script* "parsing" para todos os ficheiros SVG;
- Criar ficheiros de configuração.

- Percorrer o ficheiro SVG e colocar todas as informações em estruturas de dados;
- Criar o Sistema de *Grid*;
- Gerar os Componentes.

Implementação - "Script"

Argumentos necessários:

- Nome do projeto;
- Lista dos nomes dos ficheiros SVG que irão constituir o projeto;
- Nome do ficheiro SVG que será a homepage da aplicação web;
- Nome do ficheiro com as informações de routing.

Implementação - "Parsing"

SVG

```
<g id="Btn-3" data-name="Btn" transform="translate(829)</pre>
1041)">
   <g id="Btn-4" data-name="Btn" fill="#dd983c"
stroke="#dd983c" stroke-width="1">
    <rect width="261" height="64" stroke="none"/>
    <rect x="0.5" y="0.5" width="260" height="63" fill="none"/>
   </g>
   <text id="Text-4" data-name="Text"
transform="translate(40 41)" fill="#fff" font-size="24"
font-family="HelveticaNeue, Helvetica Neue"><tspan x="0"
y="0">Aulas Individuais</tspan></text>
  </g>
```

Estrutura de Dados

```
"id": "Btn-3",
 "data-name": "Btn",
 "transform": "translate(829 1041)",
 "fill": "#dd983c",
 "stroke": "#dd983c",
 "text": "Aulas Individuais",
 "width": "261",
 "height": "64",
 "text color": "#fff"
},
```

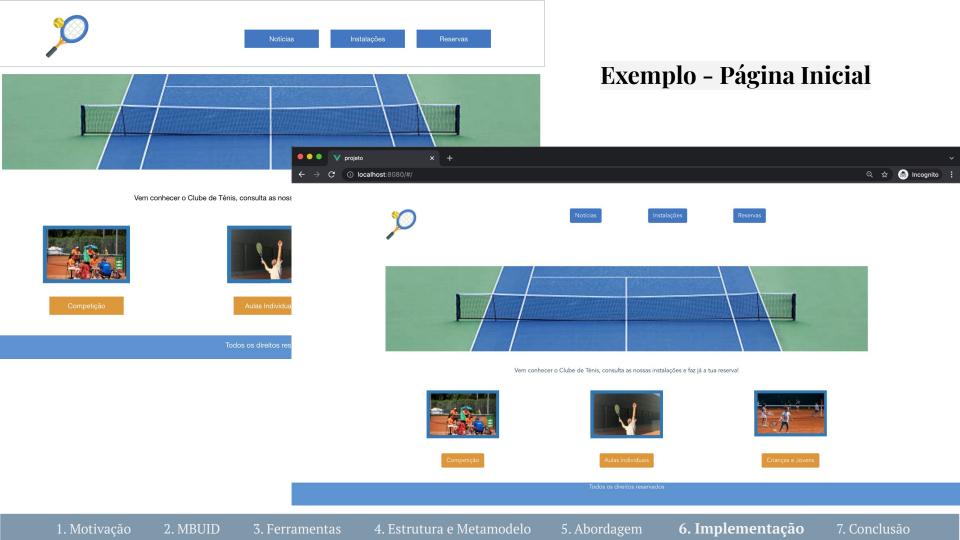
Estrutura de Dados

```
"5": {
                 %linha 5
 "2":[
                 %linha 5, coluna 2
  ...,
 ["Btn-3",[
    {"x": 829},
    {"y": 1041},
    {"width": 261},
    {"height": 64},
    {"x final": 1090},
    {"y final": 1105},
    {"fill": "#dd983c"},
    {"stroke": "#dd983c"},
     {"col": 2} ...
```

Implementação - "Parsing"

```
Web
                            (.Vue)
<div class="col-2">
 <button class="btn btn-outline-secondary Btn-3-id-Inicio"</pre>
type="button">
     Aulas Individuais
 </button>
</div>
                            (.css)
.Btn-3-id-Inicio {
  color: #fff !important;
  background-color: #dd983c !important;
  border-color: #dd983c !important;
```

17





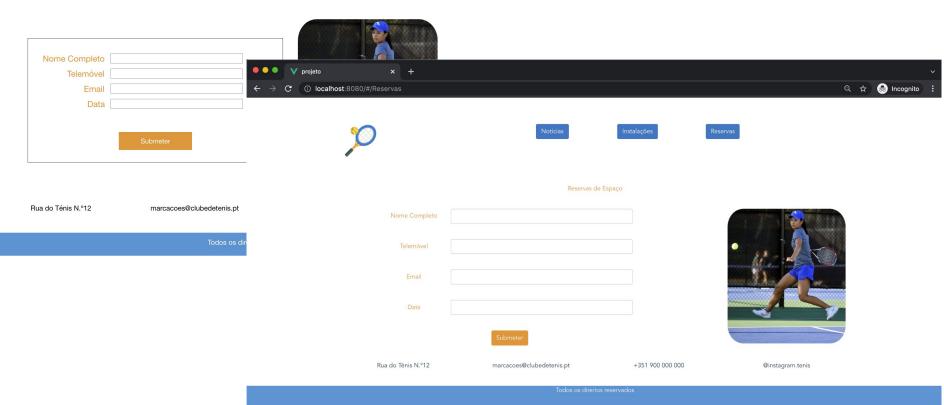
Notícias

Instalações

Reservas

Reservas de Espaço

Exemplo - Página Reservas



1. Motivação

2. MBUID

3. Ferramentas

4. Estrutura e Metamodelo

5. Abordagen

6. Implementação

7. Conclusão

Trabalho Futuro

- Flexibilidade do *design* dos componentes pode ser melhorada (componente *default*, retirado diretamente do Vue.js/Bootstrap);
- Valor por omissão para o espaçamento das linhas (40 píxeis);
- Valor por omissão para o tamanho da letra do texto;
- Valor por omissão no formulário, assume-se que o "Texto" vai ocupar sempre 3 colunas e a "Text box" vai ocupar 9 colunas;
- Formulário simples (apenas com o texto, caixa de texto e botões; ignora imagens, por exemplo);
- Recursividade dentro de uma coluna (uma coluna também se pode dividir em linhas e em colunas, e assim sucessivamente).

Conclusão

- Adobe XD + Vue.js/Bootstrap;
- SVG permite independência das ferramentas de prototipagem (mas complicado de interpretar);
- Resultado visual próximo da realidade pretendida;
- (i) Flexibilidade do código gerado (possibilidade de evolução do próprio código);
- (i) O objetivo é acelerar a geração de código, não automatizar totalmente o desenvolvimento.

Artigo Científico

ICGI - International Conference on Graphics and Interaction

Novembro 2021

Towards the integration of user interface prototyping and model-based development

Catarina Machado HASLab/INESC TEC & University of Minho Braga, Portugal a81047@alunos.uminho.pt

José Creissac Campos HASLab/INESC TEC & University of Minho Braga, Portugal jose.campos@di.uminho.pt

Advance—The main objective of this paper is in make a consideration has automation of sub-paper in in make a consideration in the automation of sub-paper in in make a consideration from server-side to clear-side starring from principles of their supplies their paper in the paper in make a consideration of from server-side to clear-side to clear-side confidence of from server-side confidence of fro

of Web appears we same the strengthness and minimum of the appear we same that integrated and the paper we same that integrated and the consequence of the appears of the appears of the appears of the appearance the interpretation of the O'Vs for that Atome An exports.

Index Terms—Web Development, Prototyping tools, Web frameworks, Code generation.

I. INTRODUCTION

some teresupment, covering me development of websites, we be services and web applications [1]. In the early 90's, Raising the level of abstraction of the development process, web services and web applications [1]. In the early 90's,
Kassing the level of abstraction of the development process,
however, seems a good solution to the current technological most weep pages were state 111stl, documents [2], weenstes nowever, sterms a good sommon have since become complex web applications that perform nave size become compact wor applications mis perform innauractions, present real-time data, and provide interactive.

This paper this arises with the general objective of studying cranaccom, person reasume data, and provide interactive and paper man arms with the general objective of multiple experiences to users [3]. Due to this complexity, development where the experience of the complexity development can take advantage of modelsuperiors to unest 153, but to mis complexity, development on one user interacte development can take advantage of model-support has evolved though the availability of frameworks based approaches, with a particular focus on web applications apport an everteet messign the availability of transversa.

Similar automatic recurring implementation patterns, thus reducing user interfaces, and on the role that user interface prototypes user amountance recogning inspermentations panetries, unto retaining the need for hosterplate code [4]. More recently, low code (or might play, in summary, the main contribution is in the stear non notes not notarphatae cone [4], some recensity, now code (or unigat pass), in summary, the main contributions is in the area even, no code) development platforms aim to further automate of web development by proposing an approach to automate vec, no course un repairement plantoring and to interface automatic the programming of user interfaces [5]. Automation, however, code generation from user interface prototypes. toe programming of user intertaces [5]. Automatore, however, cook generation aron user intertace prototypes.

The remainder of the paper is structured as follows. Sec-

This work is financed by National Funds through the Petitiguese funding agency, ECT. Paradação para a Ciência e al Fermingia, within project UIDB/S001470000.

978-1-6654-8343-8/21/831.00 (02021 1828)

But the ground learning the gas of intermethance of the ground learning the ground learning the ground learning the ground learning the ground ground

from the requirements phase to the production phase, with no need to change the system's description [8]. Additionally, the models can be used to (at least partially) automate the Web Development has become one of the main areas of been slow [9]. Prototyping-based approaches are favoured due generation of the source code. However, the adoption of not occorponent an occome one or me man areas or sects alow py, rroscyping-tusco approaches are favouree due to the flexibility they afford during design and development.

the rangement process.

The first web development frameworks were developed based user interface development and user-centred design prothe arts were development transmoorks were developed under usern meranee development and user-centred design, pre-tage in the late 1995's, and since then over five thousand have supplied tools and Web development frameworks. Section III as no see 1990), and succe then over ave moustand have supplied used and over-originate transcences. Section III been released [1]. Although all of them target (multi-dered) then presents the approach proposed to go from user interface prototypes to source code. To this end, the structure of a prototype and a meta-model of a web framework are presented and the mapping between them discussed. In Section IV a short example is used to illustrate how this approach is applied. In the last two Sections, the discussion and conclusions of the

6. Implementação



Universidade do Minho

Escola de Engenharia

Geração Automática de Interfaces de Utilizador para Aplicações Web

Dissertação

Catarina Araújo Machado - A81047 Orientador: José Creissac Campos