



Universidade do Minho
Escola de Engenharia

Geração Automática de Interfaces de Utilizador para Aplicações Web

Dissertação

Catarina Araújo Machado – A81047

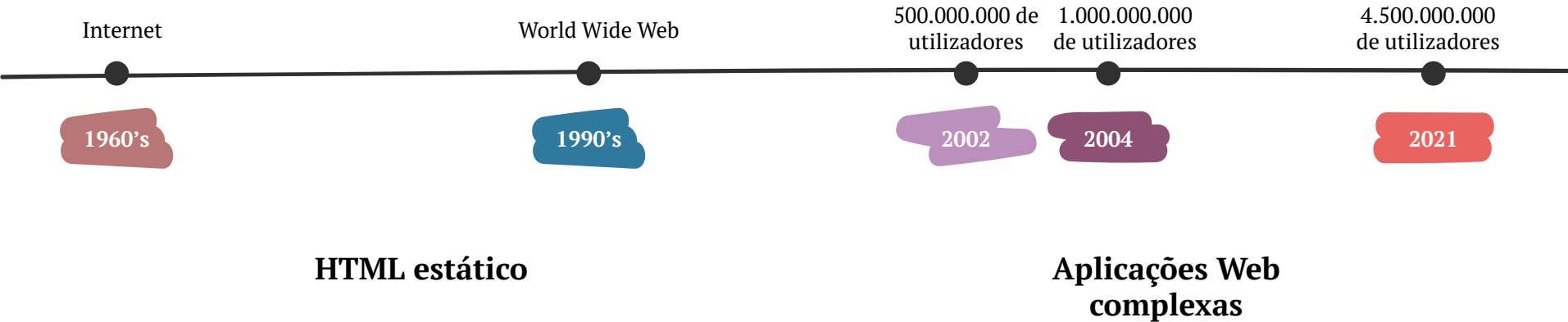
Orientador: José Creissac Campos

Agenda

- Contextualização
- Ferramentas
- Abordagem
- Implementação
- Conclusão

Contextualização

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



Motivação

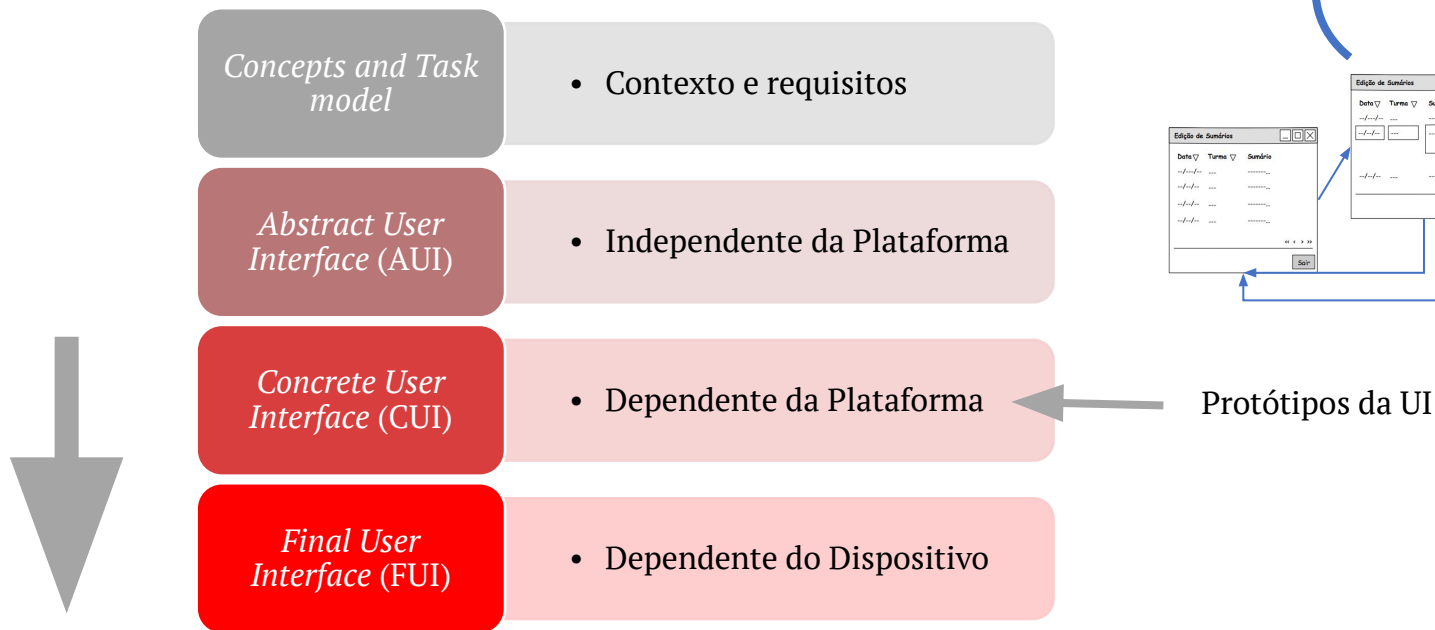
- As Interfaces de Utilizador são subsistemas complexos que representam uma parte relevante do esforço de *design* e desenvolvimento
- “The rapid pace of innovation and rather uncontrolled, organic evolution of the Web (...) put the **developers in a complex position.**” (Taivalsaar & Mikkonen, 2017)
- 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



Ferramentas

Desenvolvimento Protótipos



Figma



Sketch

Framework Web



ANGULAR



React

Estrutura de um Protótipo

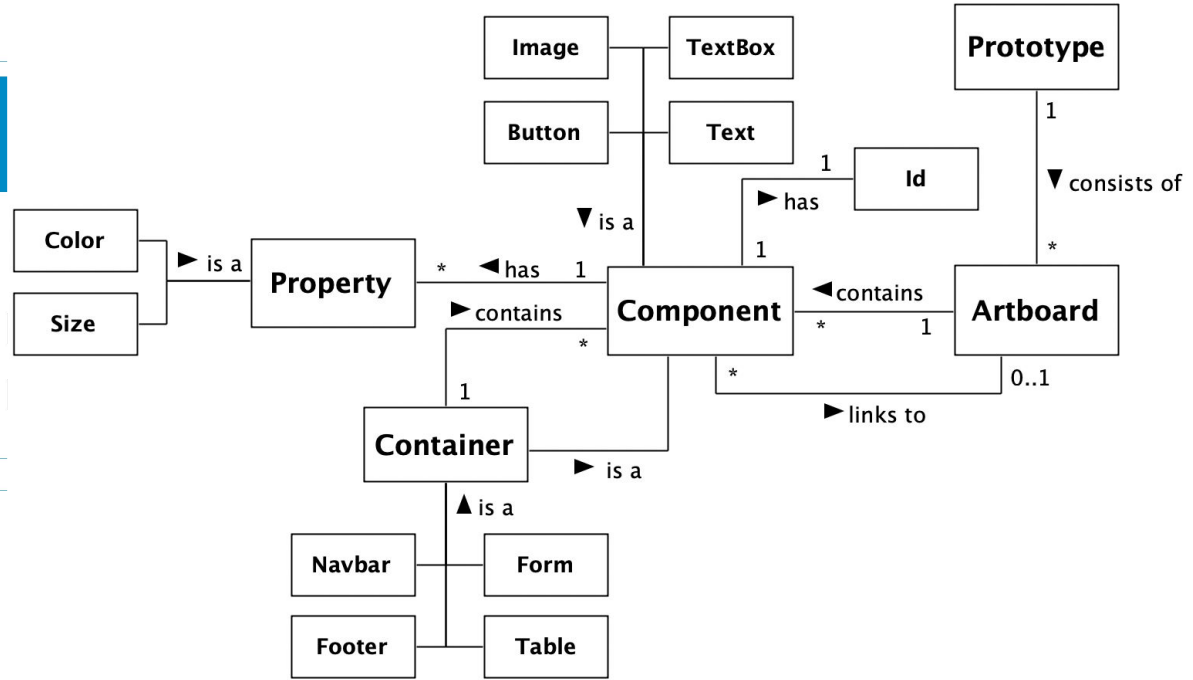


Sign up

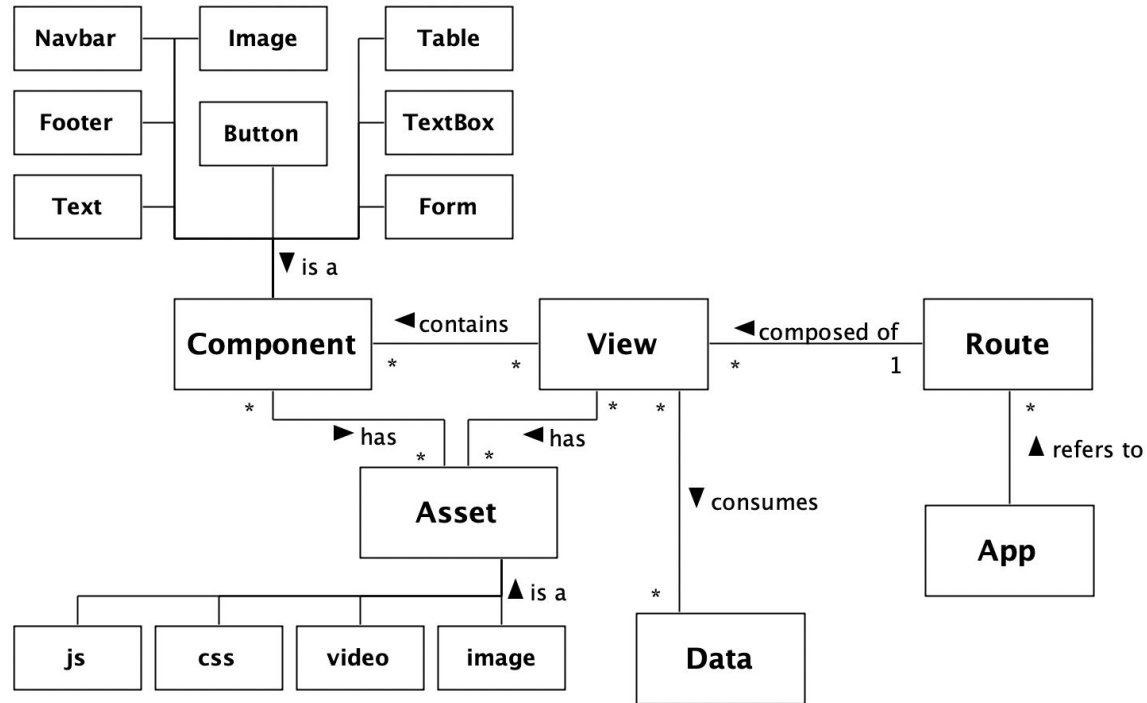
This is a big title example

Email
Password

Todos os direitos reservados



Metamodelo da Arquitetura de Componentes Vue.js



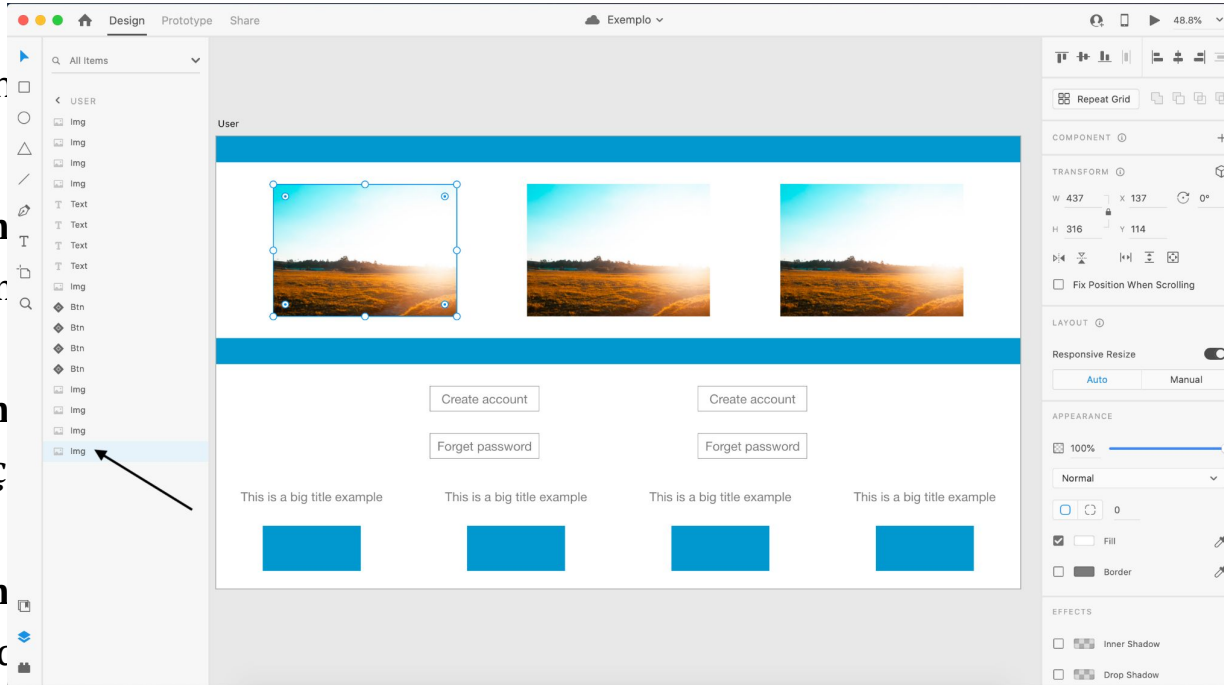
De Protótipos a Modelos CUI

(perde-se a in

● Mapeam
ID do cor

● Mapeam
Informaç

● Mapeam
Sistema c



ca as posições

De Protótipos a Modelos CUI

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

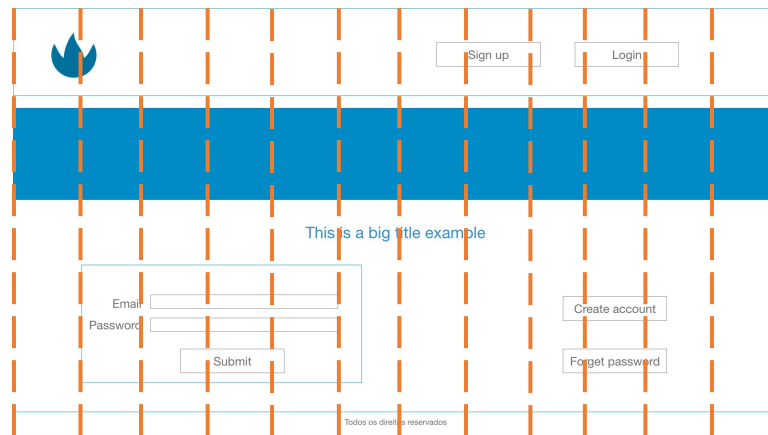
Requisitos

- 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)

“Script”



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

“Parsing”



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

- 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 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)

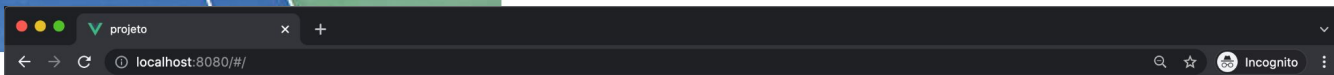
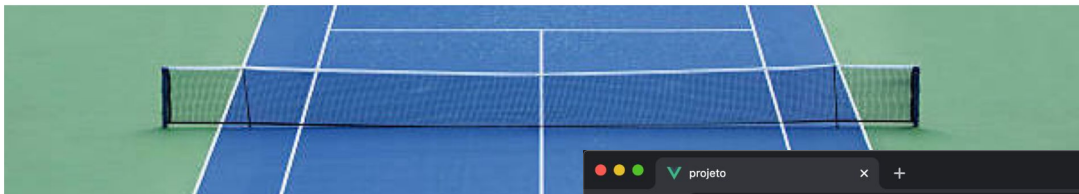
```
<div class="col-2">  
  <button class="btn btn-outline-secondary Btn-3-id-Inicio"  
    type="button">  
    Aulas Individuais  
  </button>  
</div>
```

(.css)

```
.Btn-3-id-Inicio {  
  color: #fff !important;  
  background-color: #dd983c !important;  
  border-color: #dd983c !important;  
}
```

[Notícias](#)[Instalações](#)[Reservas](#)

Exemplo - Página Inicial



Vem conhecer o Clube de Ténis, consulta as nos



Competição



Aulas Individua

Todos os direitos res

[Notícias](#)[Instalações](#)[Reservas](#)

Vem conhecer o Clube de Ténis, consulta as nossas instalações e faz já a tua reserva!



Competição



Aulas Individuais



Crianças e Jovens

Todos os direitos reservados

[Notícias](#)[Instalações](#)[Reservas](#)

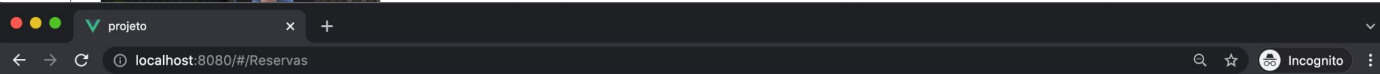
Reservas de Espaço

Nome Completo

Telemóvel

Email

Data

[Notícias](#)[Instalações](#)[Reservas](#)

Rua do Ténis N.º12

marcacoes@clubedetenis.pt

Todos os dir

Reservas de Espaço

Nome Completo

Telemóvel

Email

Data



Rua do Ténis N.º12

marcacoes@clubedetenis.pt

+351 900 000 000

@instagram.tenis

Todos os direitos reservados

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;
- Flexibilidade do código gerado (possibilidade de evolução do próprio código);
- O objetivo é acelerar a geração de código, não automatizar totalmente o desenvolvimento.

ICGI - International Conference on Graphics and Interaction

Novembro 2021





Universidade do Minho
Escola de Engenharia

Geração Automática de Interfaces de Utilizador para Aplicações Web

Dissertação

Catarina Araújo Machado – A81047

Orientador: José Creissac Campos