

GERADOR DE INTERFACES GRÁFICAS PARA IOS

GABRIEL SEBASTIAN RAMIREZ
JOYCE MARTINS

Roteiro

- Introdução
- Objetivos
- Fundamentação teórica
- Especificação
- Implementação
- Operacionalidade
- Resultados e discussão
- Conclusão
- Extensões

Introdução

-
- Nos últimos anos, os dispositivos móveis tornaram-se populares nas mais diversas áreas.
- A expectativa de vendas de tablets para 2013 no mercado nacional é de superar a marca de 7 milhões de unidades
- No entanto, o desenvolvimento requer bastante conhecimento e experiência do desenvolvedor.
- Titanium Studio, permite desenvolver aplicativos nativos com mais facilidade.



Objetivos

O objetivo deste trabalho é desenvolver uma ferramenta para gerar protótipos de interfaces gráficas de aplicativos móveis na plataforma iOS. Os objetivos específicos são:

- Disponibilizar uma ferramenta para iPad, na qual seja possível criar interfaces gráficas tanto para o iPhone e iPad.
- Gerar o XML e o código JavaScript correspondente às interfaces criadas.
- Exportar o XML e o código JavaScript para o ambiente do Titanium Studio.

Fundamentação Teórica

- Plataforma iOS
- HIG
- Titanium Studio
- Framework Alloy

Plataforma iOS

- O iOS é o sistema operacional utilizado pela Apple nos seus dispositivos móveis.
- Utilizado no iPhone, iPod Touch, iPad, iPad mini.
- A interface do usuário do iOS é baseado no conceito de manipulação direta, utilizando gestos em multi-toque.
- Atualmente se encontra na versão 7.0.4

HIG

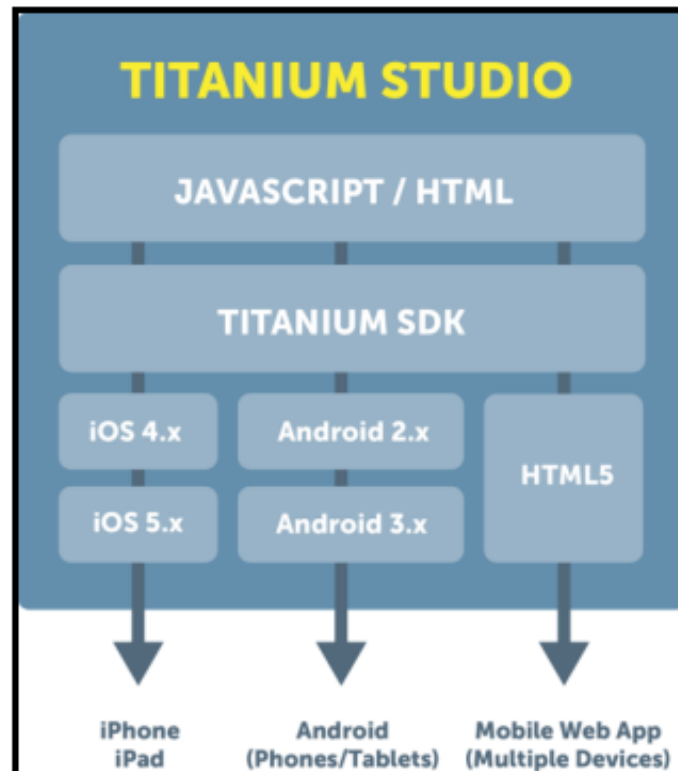
Principais recomendações:

- Utilizar componentes de interface com um tamanho mínimo de 44x44 pontos para não dificultar a manipulação dos mesmos;
- Atentar para a rotação do dispositivo.
- Desenvolver o aplicativo da maneira mais intuitiva.
- Procurar manipular os objetos diretamente.

Titanium Studio

- É uma plataforma *desktop* para desenvolvimento web e mobile criado pela AppCelerator em 2008.
- Possibilita criar aplicativos nativos para iOS e Android em JavaScript.
- Possibilita criar aplicativos web *mobile* em HTML5.
- Consiste em uma camada de *plug-ins* que, juntamente com o Aptana Studio 3.2.2, trabalha sobre o core do Eclipse versão 3.7.2.

Desenvolvimento no Titanium Studio



Framework Alloy

- Alloy é um *framework* da AppCelerator para o Titanium Studio que facilita o desenvolvimento de aplicativos móveis.
- Utiliza o MVC, a partir de XML e CSS, para separar a interface do usuário da lógica de negócio e do modelo de dados.
- É possível diminuir significativamente a quantidade de tempo e de código implementado.

Especificação

Requisitos Funcionais:

- RF01: permitir selecionar para qual dispositivo a interface será criada.
- RF02: exibir os elementos gráficos suportados.
- RF03: permitir escolher os elementos gráficos desejados.
- RF04: permitir informar as propriedades de cada elemento selecionado.

Especificação

Requisitos Funcionais:

- RF05: salvar localmente as interfaces gráficas criadas.
- RF06: gerar XML e código JavaScript correspondente à interface gráfica criada.
- RF07: exportar XML e código JavaScript gerados para uso no Titanium Studio.

Especificação

Requisitos não funcionais:

- RNF01: ser desenvolvida no ambiente do Titanium Studio.
- RNF02: ser desenvolvida para a plataforma iOS (iPad).
- RNF03: ser implementada utilizando a linguagem de programação JavaScript.

Especificação

Requisitos não funcionais:

- RNF04: utilizar XML para armazenamento dos elementos gráficos suportados.
- RNF05: utilizar o banco de dados SQLite para armazenar as interfaces criadas.

Elementos gráficos suportados

Os elementos gráficos suportados são: área de texto, barra de pesquisa, barra de progresso, botão, campo, seletor de valor, data e hora, imagem, interruptor, navegação, rótulo e visão web.

Exemplo:

Elemento	Código JavaScript	XML Alloy
área de texto	<pre>textArea = Ti.UI.createTextArea ({value: 'Texto' })</pre>	<pre><TextArea value='Texto'> </TextArea></pre>

Casos de uso

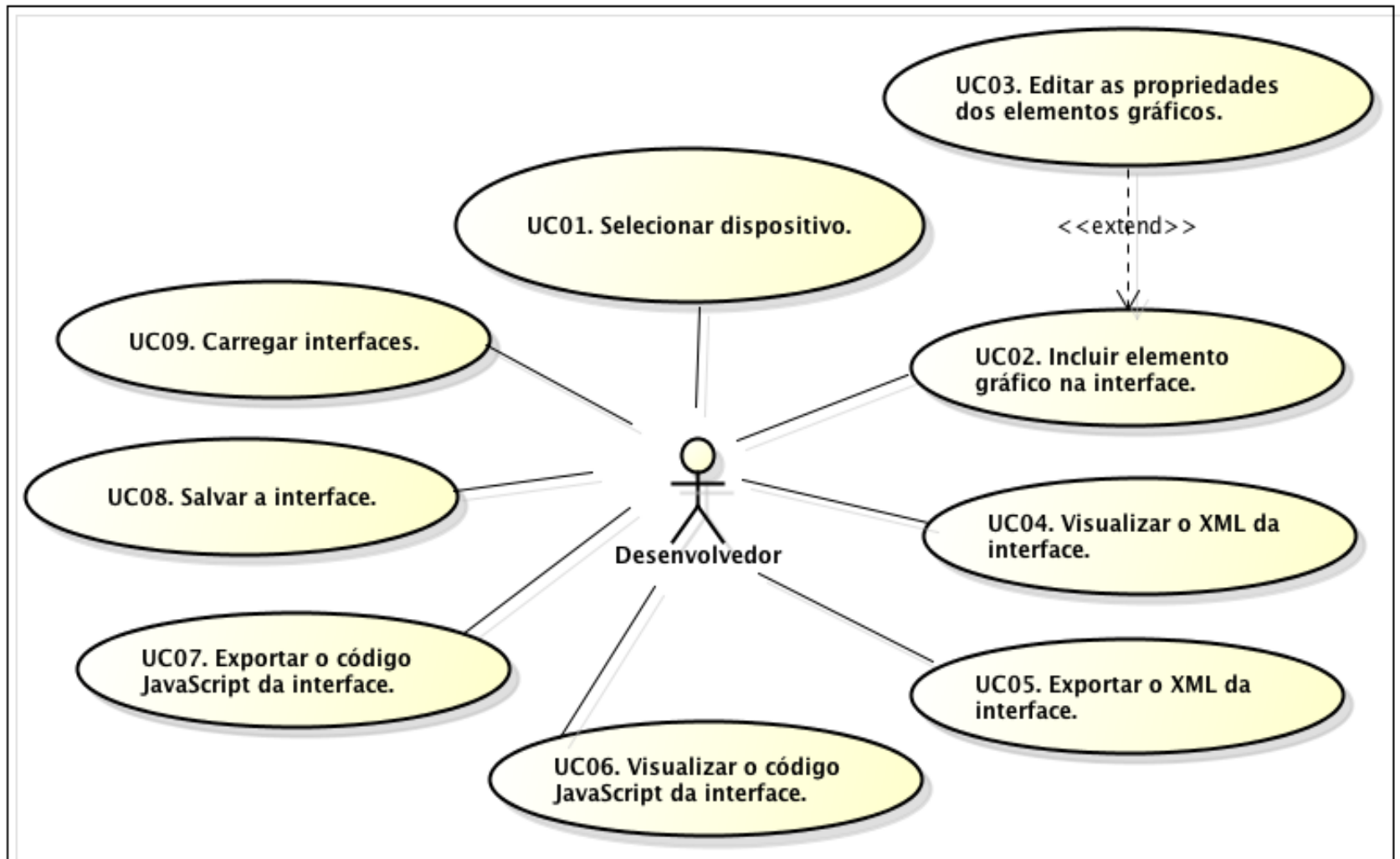


Diagrama de classes

Pacote Controlador

App.Controller

Gera_Codigo

```
- vr_codigo : String  
+ setCodigo(codigo : String) : void  
+ getCodigo() : String  
+ inicializaCodigo() : void  
+ geraCodigoElemento(par_elemento : Ti.UI.Element, par_id : int) : void  
+ finalizaCodigo() : void
```

Prop_Elementos

```
+ retornaArquivo() : XML  
+ retornaAltMinima(par_mElement : String, par_vlrAltura : int) : int  
+ retornaLarMinima(par_nmElement : String, par_vlrLargura : int) : int  
+ editaPropriedade(par_nmElement : String, par_nmPropriedade : String) : boolean  
+ retornaPropriedade(par_nmElemento : String, par_nmPropriedade : String) : String
```

Trata_Db

```
+ criaRegistro(par_nmlInterface : String, par_xmlInterface : String) : int  
+ buscaNovoId() : int  
+ updateRegistro(par_idInterface : int, par_xmlInterface : String) : void  
+ deletaInterfaces() : void  
+ buscaInterfaces() : Array[]  
+ executa(sql : String) : void
```

Limites

```
+ verificaY(par_vlrY : int, par_vlrAltura : int) : int  
+ verificaX(par_vlrX : int, par_vlrLargura : int) : int  
+ verificaAltura(par_vlrAltura : int, par_vlrY : int, par_elemento : Ti.UI.Element) : int  
+ verificaLargura(par_vlrLargura : int, par_vlrX : int, par_elemento : Ti.UI.Element) : int  
+ verificaOutros(par_arr_pontos_ele : Array(), par_arr_pontos_new : Array()) : boolean
```

Interface_Xml

```
- vr_arq_xml : File  
+ setArquivo(conteudo : String) : void  
+ getArquivo() : String  
+ criaNovoNo(par_nm_tag : String) : int  
+ criaNoPropriedades(par_nm_tag : String) : Node XML  
+ trazUltimoElemento(par_nm_tag : String) : int  
+ removeNo(par_nm_tag : String, par_id : int) : void  
+ posicoesElementos(par_elemento : Ti.UI.Element, par_vlrY : int, par_vlrX : int) : boolean  
+ alteraAtributo(par_nm_tag : String, par_nm_atributo : String, par_vl_atributo : String) : void  
+ mostraCodigo() : String  
+ montaCodigoSalvo(par_xml_interface : String) : void  
+ mostraCodigoGerado() : void
```

Diagrama de classes

Pacote Modelo

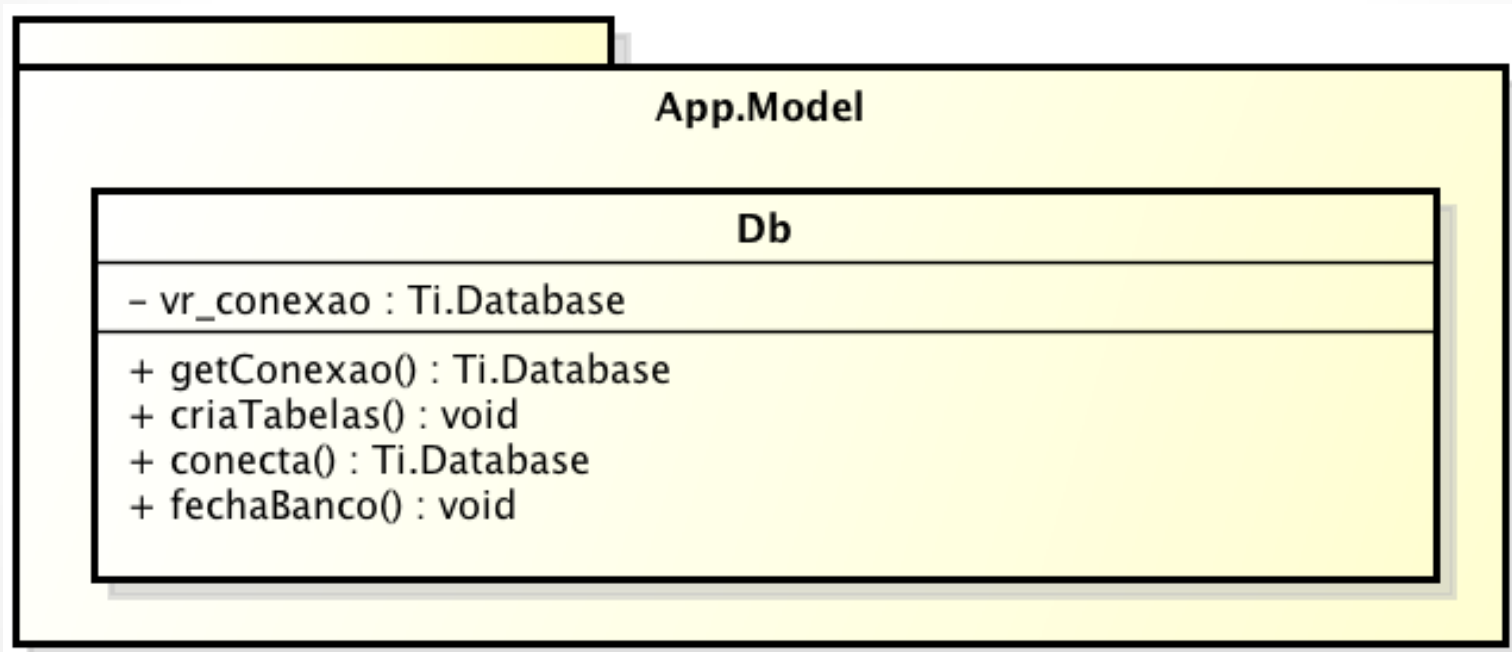


Diagrama de classes

Pacote Visão

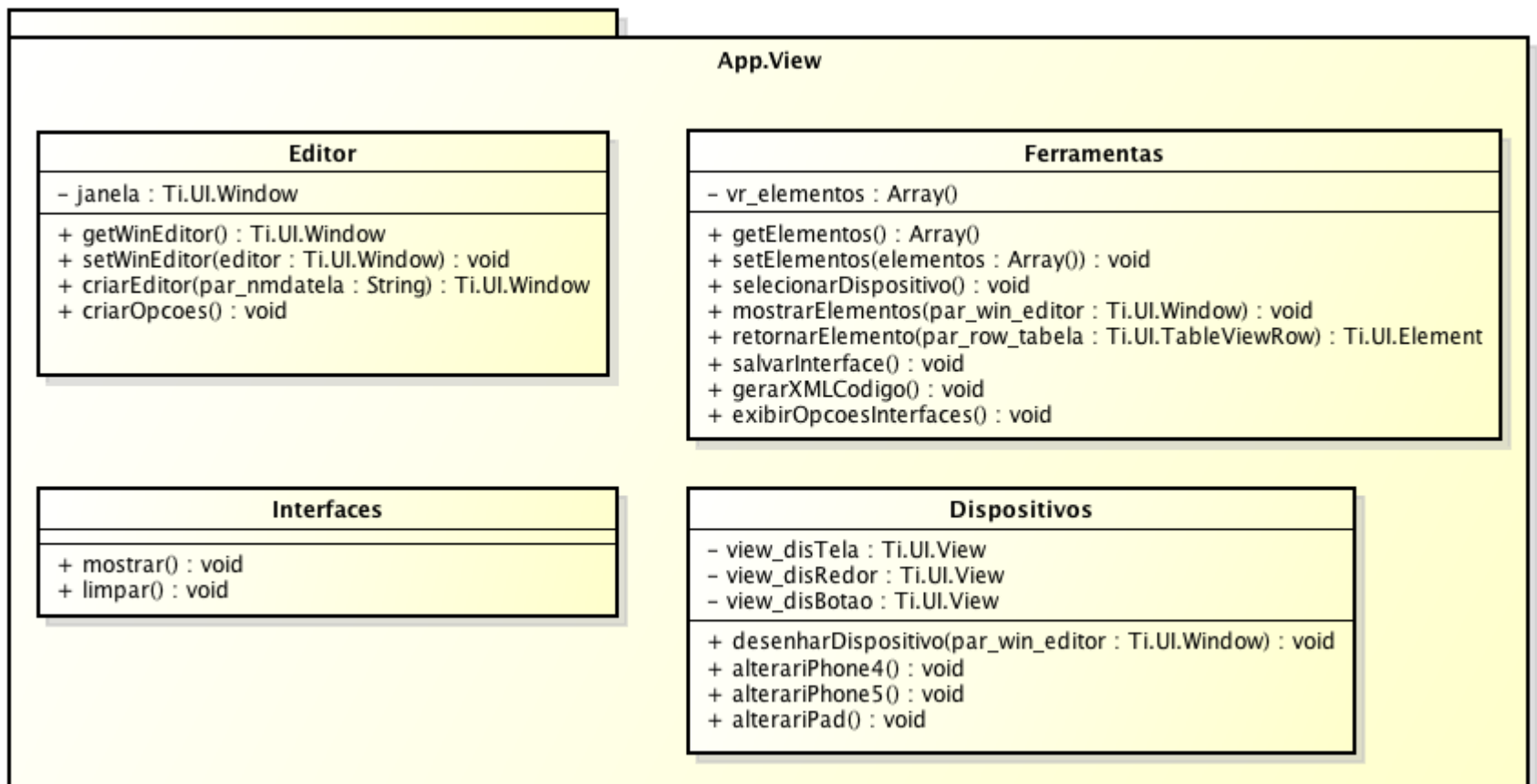
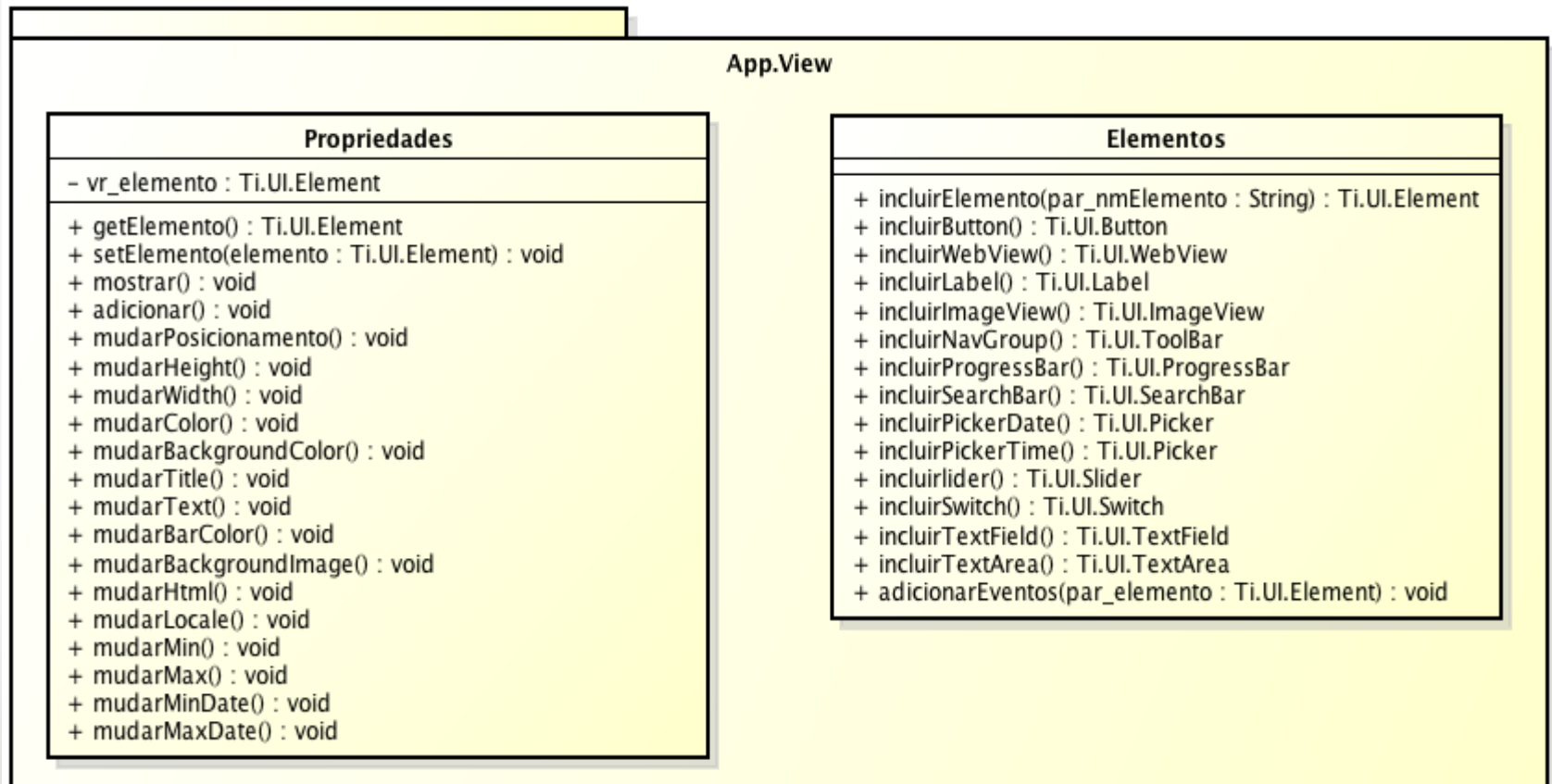


Diagrama de classes

Pacote Visão



Implementação

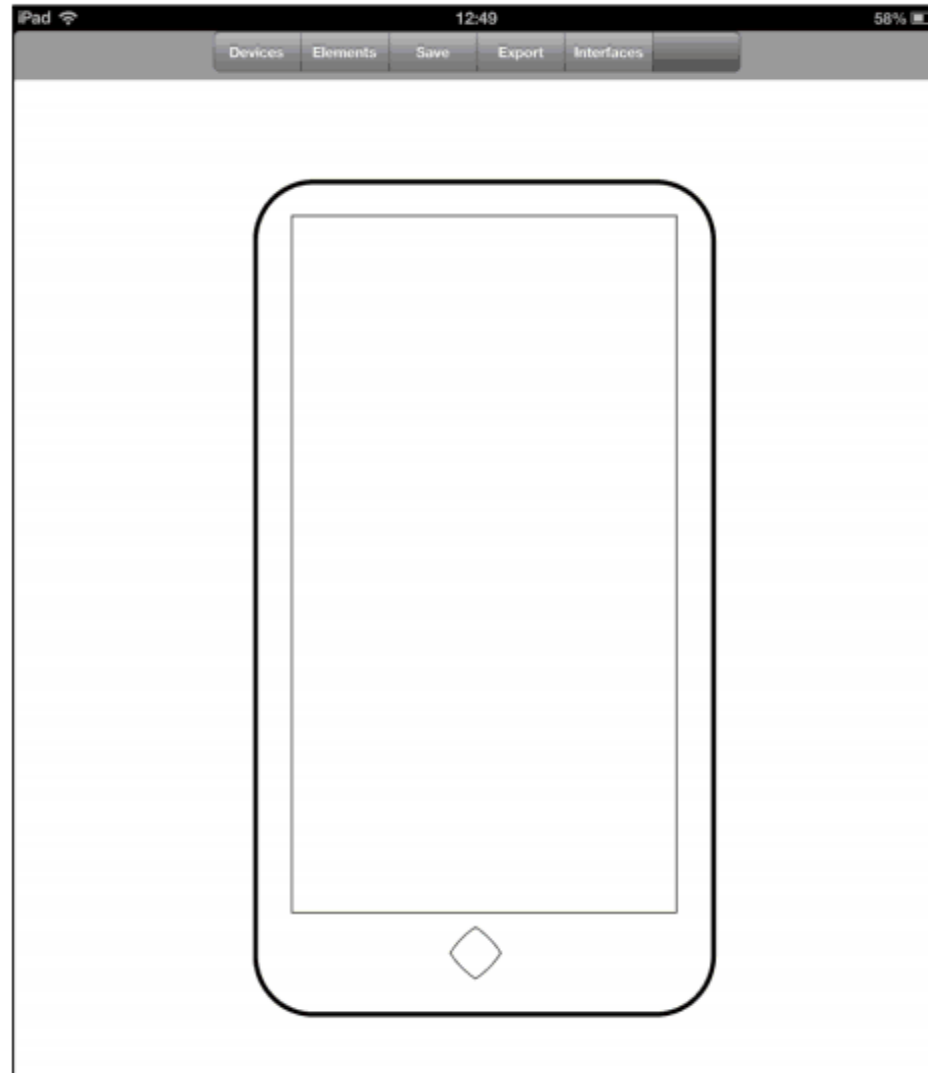
Técnicas e ferramentas:

- Titanium Studio, versão 3.1.1.
- Xcode SDK, versão 5.0.
- Banco de dados SQLite.

Dispositivos para testes:

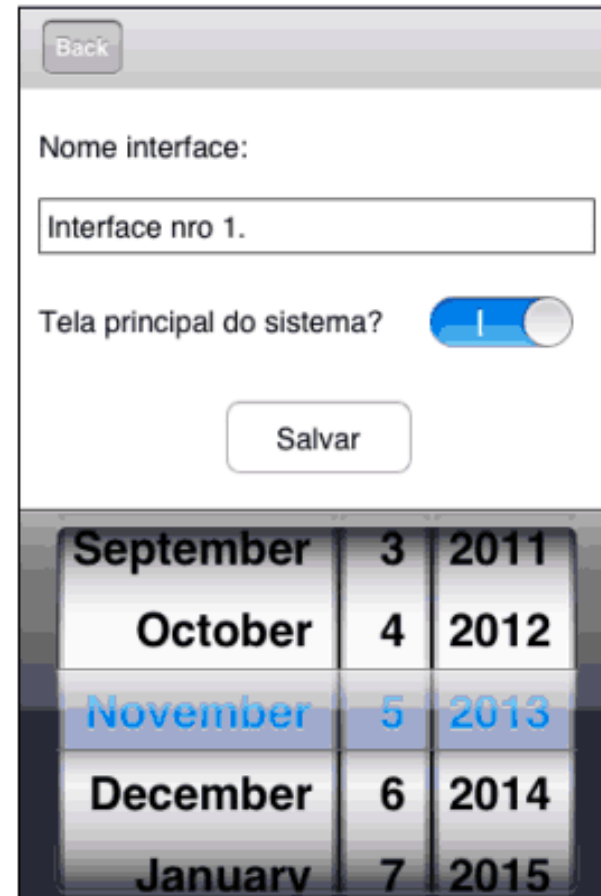
- iPhone 4 de 16 GB.
- iPad 4 de 16 GB.
- Emulador do iPhone e do iPad.

Operacionalidade



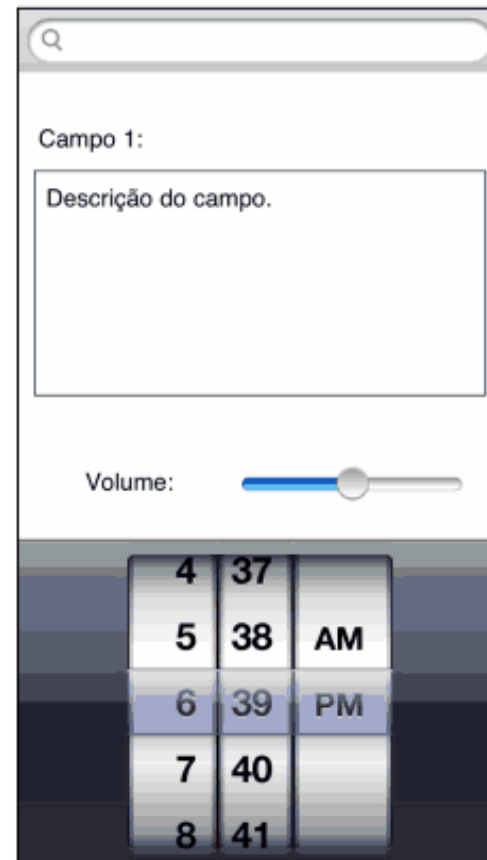
Resultados e discussão

iPhone 4



Resultado e discussão

iPhone 5



Resultados e discussão

iPad



Comparativo dos trabalhos correlatos

funcionalidades / ferramenta	Ramirez	Interface 2	jQuery Mobile	Forgued UI
geração de código para Android	NÃO	NÃO	SIM	SIM
geração de código para iOS	SIM	SIM	SIM	SIM
geração de código para Titanium Studio	SIM	NÃO	NÃO	SIM
geração de código para Xcode	NÃO	SIM	NÃO	NÃO
ferramenta disponível em ambiente desktop	NÃO	NÃO	SIM	SIM
ferramenta disponível para Android	NÃO	NÃO	SIM	NÃO
ferramenta disponível para iOS	SIM	SIM	SIM	NÃO
persistência das interfaces	SIM	SIM	SIM	SIM

Conclusão

- A ferramenta desenvolvida neste trabalho atingiu todos os objetivos propostos.
- As interfaces geradas para o iPhone 4 e iPhone 5 mantiveram-se fiel as criadas na ferramenta.
- Para o iPad, alguns elementos apresentaram falta de precisão no posicionamento.
- Certos elementos gráficos do iOS acabaram não sendo contemplados, como por exemplo: Table view, Toolbar, Popover, Split view, entre outros.

Conclusão

- A maior dificuldade encontrada foi exibir as interfaces salvas no banco local.
- O *framework* Titanium mostrou-se bastante útil.

Extensões

- Aumentar a quantidade de elementos gráficos suportados.
- Aumentar a quantidade de propriedades gráficas suportadas.
- Ter suporte a nova versão do iOS 7.0.
- Gerar código para o iPad mini.
- Melhorar a precisão do posicionamento dos elementos na geração do código para o iPad.

Extensões

- Disponibilizar a ferramenta no iPhone e/ou iPad mini.
- Gerar o código das interfaces na linguagem nativa do iOS, o Objective C.
- Permitir a ligação entre determinadas interfaces, criando a ideia de um projeto.
- Criar eventos para os elementos gráficos.
- Permitir aumentar e diminuir o tamanho dos elementos de uma maneira mais intuitiva.



Apresentação da ferramenta