

**seaside** 

# Navegación



# Navegación

- #call: y #answer

## Server Code

```
ContactListView>>editContact:
```

```
...  
self call:  
...
```

Contact-  
View

```
ContactView>>save  
self answer
```

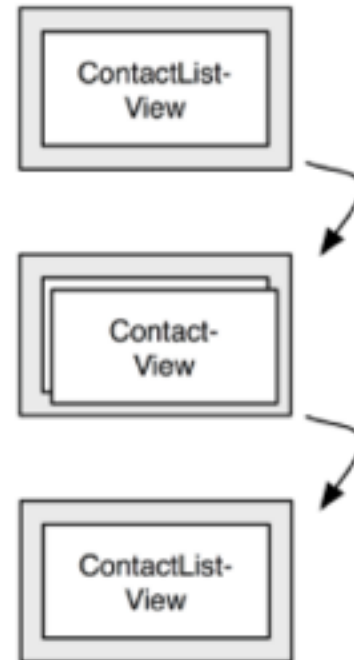
```
ContactListView>>editContact:
```

```
...
```

```
self call: Contact-  
View
```

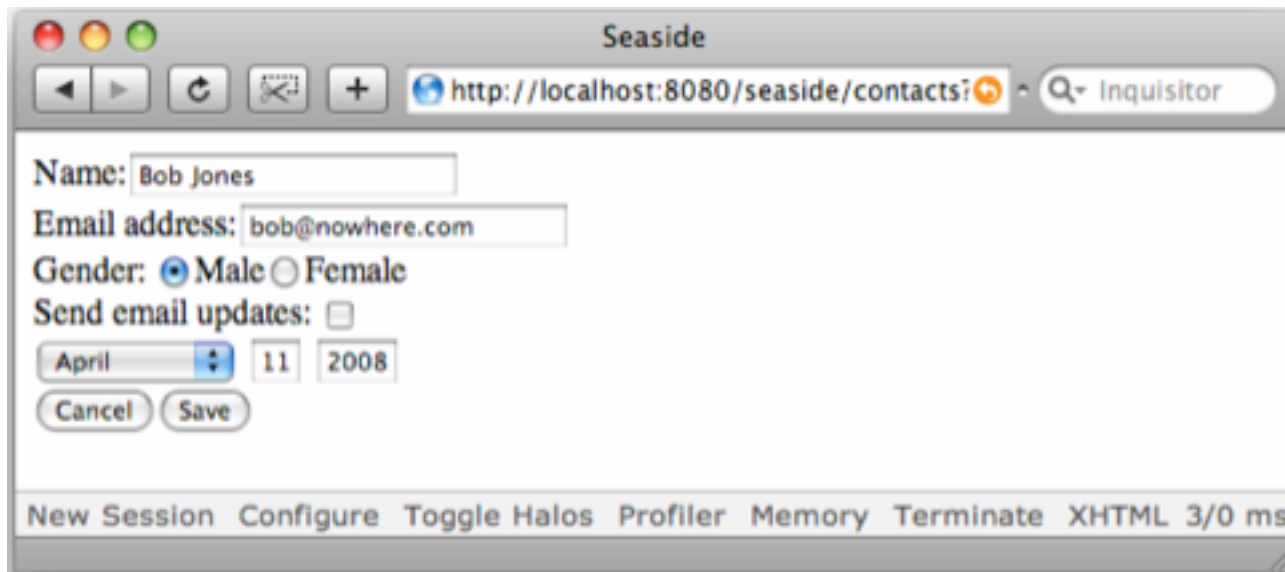
```
"continue here"
```

## Browser



# Navegación

- #answer vs. #answer:



```
ContactView>>save
```

```
self answer: true
```

```
ContactView>>cancel
```

```
self answer: false
```

# Navegación

- #answer vs. #answer:



```
ContactListView>>editContact: aContact
| view |
view := ContactView new.
view contact: aContact.
(self call: view)
  ifFalse: [ ^ nil ].
self inform: 'Thanks for editing ' , aContact name
```

# Navegación

- Cuando el componente al que delegamos necesita un objeto del dominio, podemos optar entre:
  - Enviar el propio objeto
  - Enviar una copia

```
ContactListView>>editContact: aContact
| view copy |
view := ContactView new.
copy := aContact copy.
view contact: copy.
(self call: view)
  ifTrue: [ Contact removeContact: aContact; addContact: copy ]
```

# Navegación

- Don't call while rendering:
  - <http://book.seaside.st/book/components/calling/dont-call-while-rendering>
- No usar #call: en el método #renderContentOn:
  - Solo en callbacks.

# Navegación

- Built-in Dialogs
  - WAComponent>>inform:
  - WAComponent>>confirm:
  - WAComponent>>request:
  - WAComponent>>chooseFrom:



# Composición

- ¿Que pasa si queremos mostrar varios componentes en la misma página?



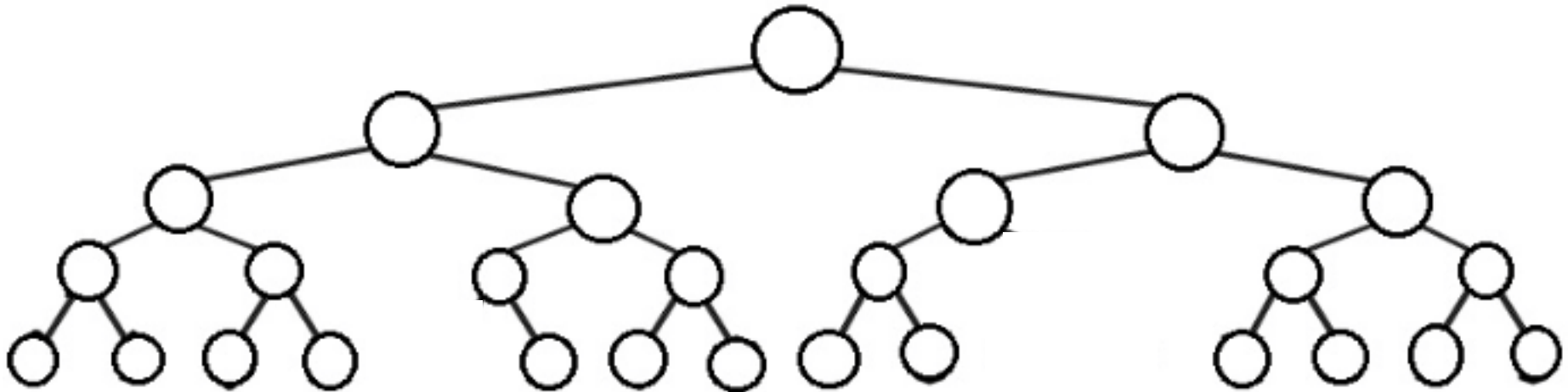
# Composición

- ¿Que pasa si queremos mostrar varios componentes en la misma página?



# Composición

- Simplemente hacemos que un componente “conozca” a otros
  - Pero bajo cierto “formato” establecido por el framework



# Composición

- Simplemente hacemos que un componente “conozca” a otros
  - Pero bajo cierto “formato” establecido por el framework
    - `#children`
    - en el método `#renderContentOn:` del “padre”:
      - enviar mensaje `#render:` a `HTMLCanvas` con el componente hijo que se necesite como parámetro

# Composición

Seaside

http://localhost:8080/seaside/iAddress?\_s:

**iAddress**

[Add contact](#)

Name	Address
<a href="#">Bob Jones</a>	<a href="#">bob@nowhere.com</a> ==
<a href="#">Steve Smith</a>	<a href="#">sm@somewhere.com</a> ==

Name:

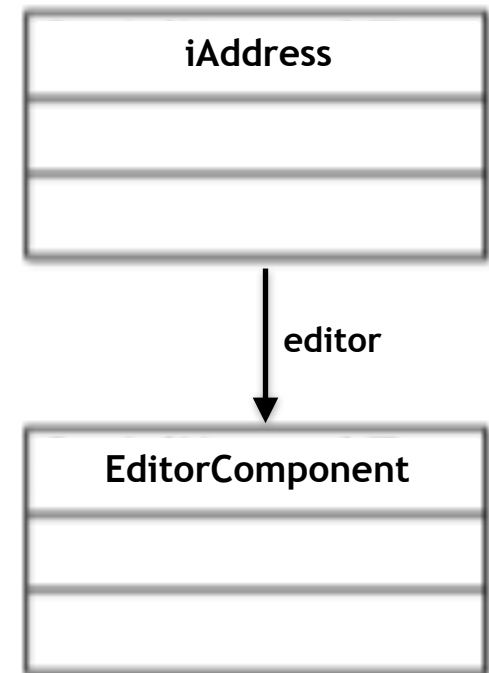
Email address:

Gender: ☒ Male ☐ Female

Send email updates: ☐

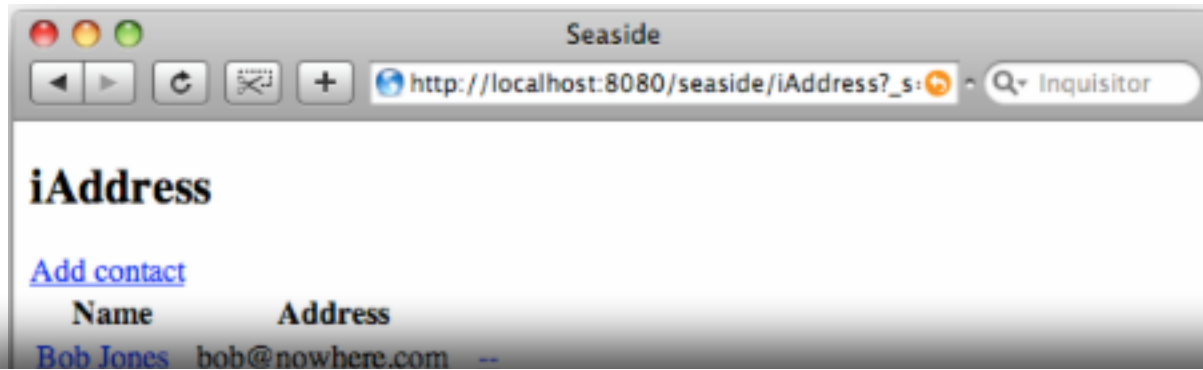
October 15 1968

New Session Configure Toggle Halos Profiler Memory Terminate XHTML 3/7 ms



```
IAddress>>children  
  ^ Array with: editor
```

# Composición



iAddress

```
IAddress>>renderContentOn: html
  html form: [
    self renderTitleOn: html.
    self renderBarOn: html.
    html table: [
      html tableRow: [
        html tableHeading: 'Name'.
        html tableHeading: 'Address' ].
      self renderDatabaseRowsOn: html ].
    html horizontalRule.
    html render: editor ]
```

# Composición

```
IAddress>>editContact: aContact  
    editor contact: aContact
```

iAddress

iAddress

```
IAddress>>renderDatabaseRowsOn: html  
    self contacts do: [ :contact |  
        html tableRow: [ self renderContact: contact on: html ] ]
```

```
IAddress>>renderContact: aContact on: html  
    html tableData: [  
        html anchor  
            callback: [ self editContact: aContact ];  
            with: aContact name].  
    html tableData: aContact emailAddress.  
    html tableData: [  
        html anchor  
            callback: [ self removeContact: aContact ];  
            with: '--' ]
```

# Composición

Seaside

http://localhost:8080/seaside/iAddress?\_s: Inquisitor

## iAddress

[Add contact](#)

Name	Address
<a href="#">Bob Jones</a>	bob@nowhere.com ==
<a href="#">Steve Smith</a>	sm@somewhere.com ==

---

Name:

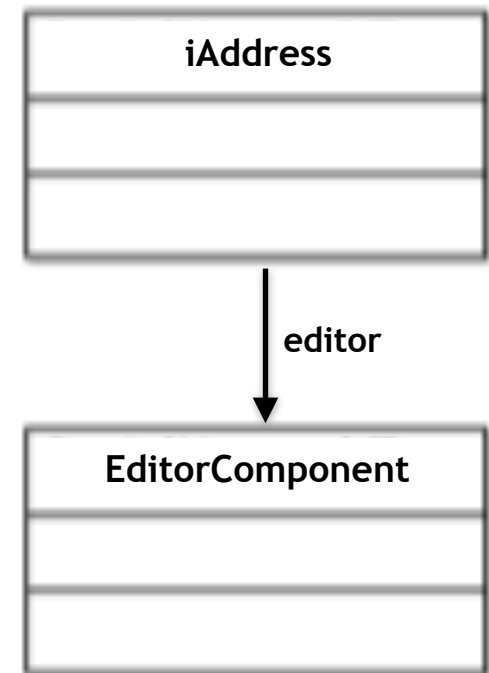
Email address:

Gender: ☒ Male ☐ Female

Send email updates: ☐

October 15 1968

New Session Configure Toggle Halos Profiler Memory Terminate XHTML 3/7 ms



```
IAddress>>children
^ Array with: editor
```