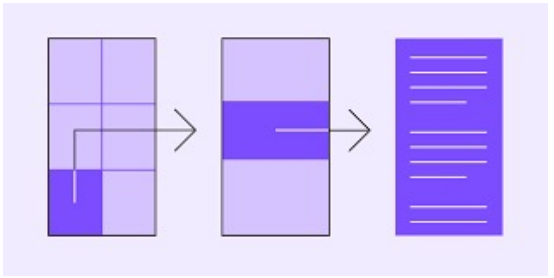
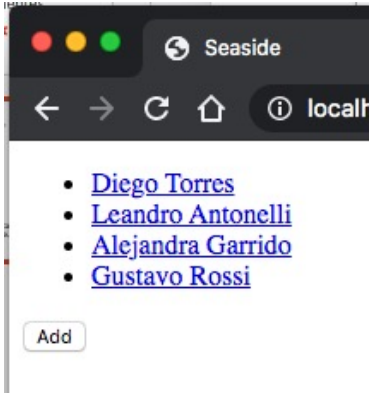
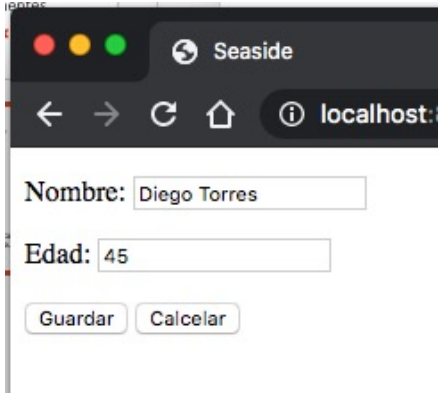


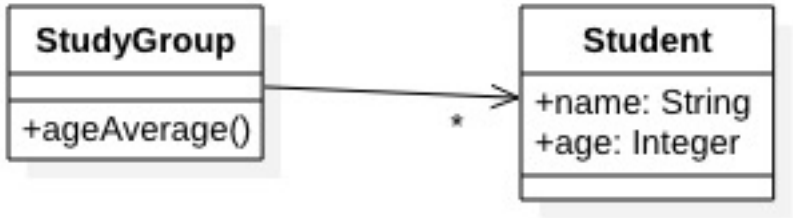
Formularios y navegación entre componentes

seaside

Listar, agregar, mostrar/editar …







Pinceles que vamos a utilizar:

unorderedList (<ul>), listItem (<li>), anchor (<a>), paragraph (<p>), form (<form>), label (<label>), textInput (<input>), button (<button>)

Forms (en general)

- En la web, utilizamos los forms para que el usuario ingrese datos.
- Los forms tienen inputs (textInput, textArea, select, etc.)
- Los inputs se configuran con la información a mostrar
- Los botones envían el formulario y con él los nuevos datos de los inputs

Forms (Seaside)

- Los inputs necesitan:
  - Obtener el valor para mostrar
  - Guardar el nuevo valor ingresado
- Cuando se oprime un botón, se guardan los nuevos valores de los inputs, y se ejecuta el callback del botón
- El formulario conoce al objeto que edita, y al contenido de los inputs (en variables de instancia)

- El callback del botón guardar, modifica al objeto editado
- El callback del botón cancelar deja todo como está

Input (dos formas de configurarlos)

- Con un selector (mensaje) que se usa como getter y setter (en este caso a self, el componente)

```
aCanvas textInput on: #newName of: self
```

- Con #with: y #callback:

```
aCanvas textInput
    callback: [ :newValue | self newAge: newValue ];
    with: self newAge ].
```

Conectando componentes

- El mensaje #call:, que entienden los componentes, pasa el control al componente que viene como parámetro (es bloqueante)
- El mensaje #answer: , que entienden los componentes, retorna el control al componente que lo llamó