

Organización de aplicaciones con Backbone.js



Diego Cardozo

github.com/diegocard/backbone-presentation

Objetivos

- Esta charla no es un tutorial de Backbone
- Se centra en el diseño de clientes web inteligentes
- Vamos a usar Backbone como ejemplo
- Vemos conceptos independientes de la tecnología
- Aplicables a otras herramientas
- Y de paso aprendemos algo de Backbone

¿Que espero que se lleven?

- Conceptos útiles para diseñar clientes complejos
- Conocimientos básicos de Backbone
- Ganas de aprender más

Para los que conocen Backbone

- Buenas prácticas
- Combinación con otras herramientas

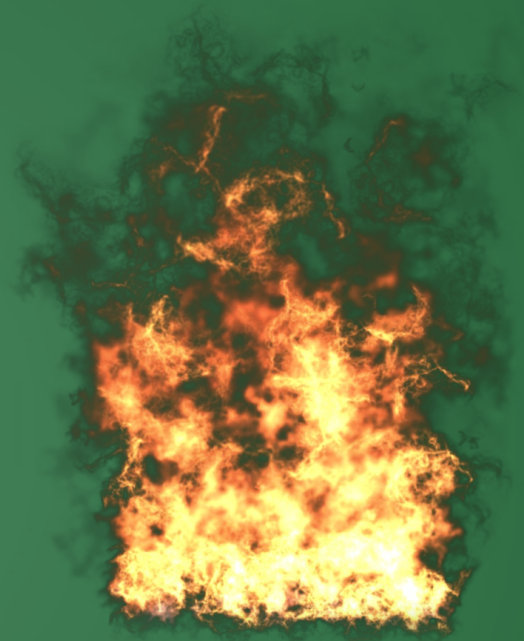
Agenda

1. Introducción
2. Arquitectura
3. Ejemplo
4. Componentes de Backbone
5. Estructurar una aplicación

Introducción (1)

- Los clientes web cuentan cada vez con mas recursos
- Permite construir clientes mas inteligentes
- Pero los clientes complejos con jQuery...
 - Son difíciles de construir
 - Carecen de estructura
 - Dificultan la reutilización
 - Crear tu propia estructura sería reinventar la rueda

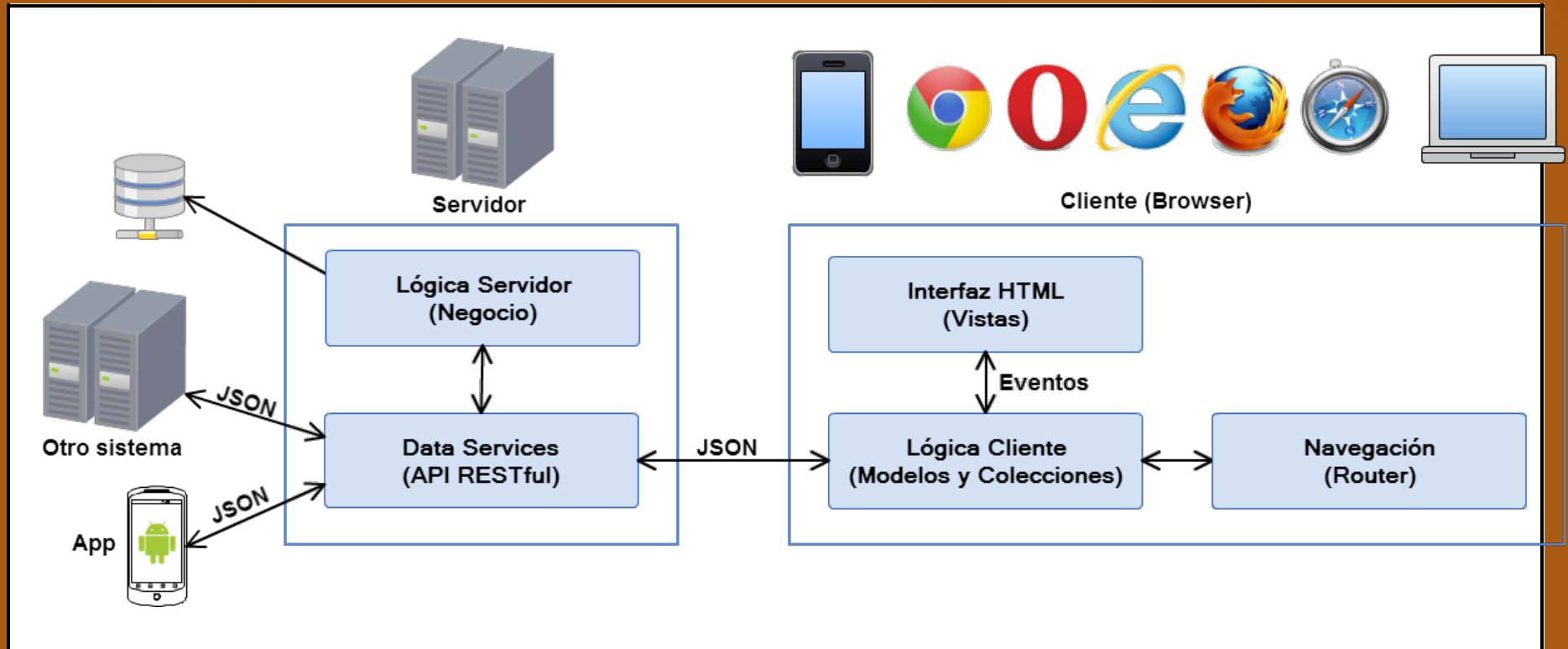
Introducción (2)



Introducción (3)

- Backbone brinda
 - Estructura al código JavaScript en el cliente
 - Varias utilidades reutilizables
- Básicamente es un framework MV*
- Organizamos el código en distintos componentes
 - Modelos
 - Colecciones
 - Vistas
 - Templates
 - Routers

Arquitectura (1)



Arquitectura (2)

Ventajas

- Mantenimiento más sencillo
- Distribución de carga
- Comienzo del desarrollo más ágil
- La interfaz es simplemente otro cliente
- Se presta muy bien para testing
- Perfecto para combinar con aplicaciones móviles

Ejemplo

github.com/diegocard/backbone-presentation/demo

Componentes (1)

Modelo

```
var User = Backbone.Model.extend({  
  urlRoot: '/users'  
});
```

Componentes (2)

Colección

```
var Users = Backbone.Collection.extend({  
  url: '/users'  
});
```

Componentes (3)

Vista

```
var UserListView = Backbone.View.extend({
  el: '.page',
  render: function () {
    var that = this;
    var users = new Users();
    users.fetch({
      success: function (users) {
        var template = _.template(
          $('#user-list-template').html(),
          {users: users.models}
        );
        that.$el.html(template);
      }
    })
  }
});
```

Componentes (4)

Manejo de eventos

```
var UserEditView = Backbone.View.extend({
  el: '.page',
  events: {
    'submit .edit-user-form': 'saveUser',
    'click .delete': 'deleteUser'
  },
  saveUser: function (ev) {
    var userDetails = $(ev.currentTarget).serializeObject();
    var user = new User();
    user.save(userDetails, {
      success: function (user) {
        router.navigate('', {trigger: true});
      }
    });
  }
});
```

Componentes (5)

Template

```
<script type="text/template" id="user-list-template">
  <a href="#/new" class="btn btn-primary">New</a>
  <hr />
  <table class="table striped">
    <thead>
      <tr>
        <th>First Name</th><th>Last Name</th><th>Age</th><th></th>
      </tr>
    </thead>
    <tbody>
      <% _.each(users, function(user) { %>
        <tr>
          <td><%= htmlEncode(user.get('firstname')) %></td>
          <td><%= htmlEncode(user.get('lastname')) %></td>
          <td><%= htmlEncode(user.get('age')) %></td>
          <td><a class="btn" href="#/edit/<%= user.id %>">Edit</a></td>
        </tr>
```

Componentes (6)

Router

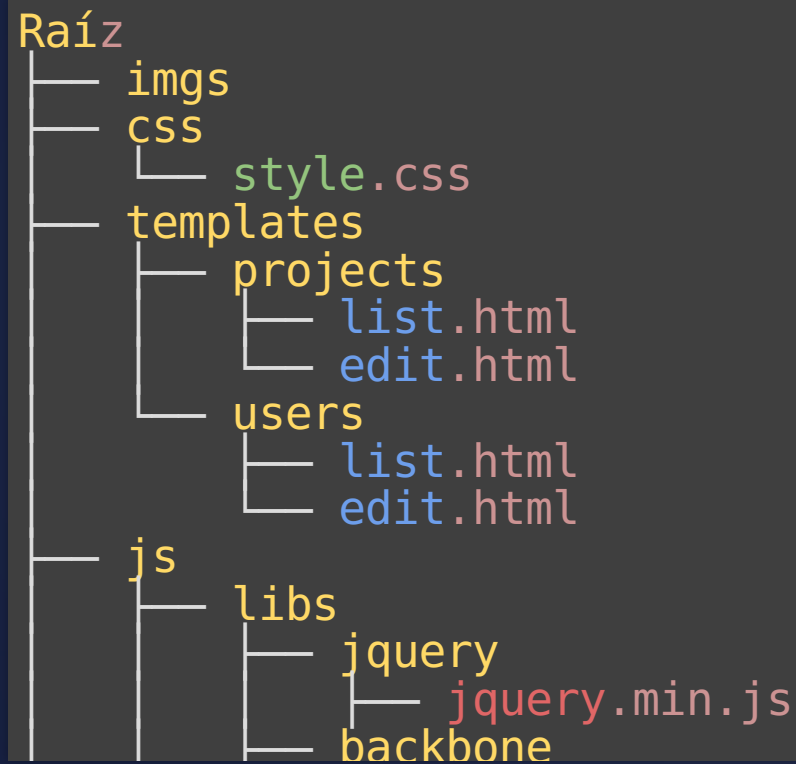
```
var Router = Backbone.Router.extend({  
  routes: {  
    "": "home",  
    "edit/:id": "edit",  
    "new": "edit",  
  }  
});
```


Estructura (1)

- Usar Backbone no es garantía de prolijidad
- Se precisa estructurar y modularizar la aplicación
- Para esto utilizamos Require.js
- Encontré un excelente ejemplo en:
 - backbonetutorials.com/organizing-backbone-using-modules

Estructura (2)

Estructura sugerida



Estructura (3)

Ejemplo: Lista de proyectos

```
define([
  'jquery',
  'underscore',
  'backbone',
  // Usando el plugin text! de Require.js,
  // cargamos las templates como texto plano
  'text!templates/project/list.html'
], function($, _, Backbone, projectListTemplate){
  var ProjectListView = Backbone.View.extend({
    el: $('#container'),
    render: function(){
      // Compilamos la template usando Underscore
      // y agregar la plantilla al elemento de la vista
      var data = {};
      var compiledTemplate = _.template(projectListTemplate, data);
      this.$el.append(compiledTemplate);
    }
  });
  return ProjectListView; // Retornar la vista
});
```

Recursos

- backbonejs.org
- backbonetutorials.com
- [addyosmani.github.io/backbone-fundamentals](https://github.com/addyosmani/backbone-fundamentals)
- github.com/diegocard/backbone-presentation

¿Preguntas?

