

Ease 4: Flujo de datos

Fujo de Datos

Partimos de...

- Casi todos los componentes extraídos
- Componentes dinámicos (props)
- Storybook con catálogo completo de componentes
- Todos los componentes tienen tests

Fujo de Datos

Queremos...

- Utilizar *datos reales*
 - volcados JSON de la data que devolverá el servidor
 - *todo el contenido* se lee de este JSON
- Las *vistas* reciben (props) y distribuyen la data
 - Views → Organisms → Molecules → Atoms

Fujo de Datos

En el repo hay múltiples ficheros JSON

ProductList \Leftarrow product-list.json + shopping-cart.json

ProductDetails \Leftarrow product-details.json + shopping-cart.json

ShoppingList \Leftarrow shopping-cart.json

Checkout \Leftarrow shopping-cart.json

Fujo de Datos

Queremos...

- Utilizar *datos reales*
 - volcados JSON de la data que devolverá el servidor
 - *todo el contenido* se lee de este JSON
- Las *vistas* reciben (props) y distribuyen la data
 - Views → Organisms → Molecules → Atoms

```
import AppLayout from '../layouts/AppLayout'
import ProductGrid from '../organisms/ProductGrid'
import Pagination from '../molecules/Pagination'
import IntroTitle from '../molecules/IntroTitle'

const ProductList = ({ data, shoppingCart, intl }) => {
  const { results, totalPages, page } = data
  return (
    <AppLayout shoppingCart={shoppingCart}>
      <IntroTitle title="Products" subtitle="Full catalog"/>
      <ProductGrid products={results} />
      <Pagination pages={totalPages} currentPage={page}/>
    </AppLayout>
  )
}

ProductList.propTypes = {
  data: PropTypes.object.isRequired,
  shoppingCart: PropTypes.object.isRequired
}
```

```
import React from 'react'
import ProductList from '../src/ui/views/ProductList'
import data from '../data/product-list.json'
import shoppingCart from '../data/shopping-cart.json'

describe('<ProductList/>', () => {

  const wrapper = mount(
    <ProductList data={data} shoppingCart={shoppingCart}/>
  )

  it('distributes the data among its organisms', () => {
    expect(
      wrapper.find('AppLayout').props().shoppingCart
    ).toEqual(shoppingCart)
  })
})
```

```
import React from 'react'
import { storiesOf } from '@storybook/react'
// components
import ProductList from './ProductList'
// data
import productListData from '../../../.storybook/data/product-list.json'
import shoppingCartData from '../../../.storybook/data/shopping-cart.json'

storiesOf('Views', module)

  .add('ProductList', () => (
    <ProductList data={productListData}
                shoppingCart={shoppingCartData} />
  ))
```


Fujo de Datos

Los *organismos* reciben la data desde la vista

- Actualizar sus *props*
 - Generar el *contenido* dinámicamente
 - Declarar sus *PropTypes*
 - Actualizar *las historias* para que reciban la data
 - Actualizar *los tests* con la nueva funcionalidad

```
import React from 'react'
import { storiesOf } from '@storybook/react'
import { withKnobs, number } from '@storybook/addon-knobs'
import Pagination from '../molecules/Pagination'

storiesOf('Organisms', module)
  .addDecorator(withKnobs)

  .add('Pagination', () => (
    <Pagination pages={number('pages', 5)}
               currentPage={number('currentPage', 2)} />
  ))
```

```
import React from 'react'
import { mount } from 'enzyme'
import Pagination from '../src/ui/molecules/Pagination'
```

```
describe('<Pagination />', () => {
```

```
  const wrapper = mount(
    <Pagination pages={5}
      currentPage={2} />
  )
```

```
  it('renders the correct number of pages', () => {
    expect(
      wrapper.find('PaginationItem').length
    ).toBe(5)
  })
```

```
  it('mark the correct page as active', () => {
    expect(
      wrapper.find('PaginationItem').at(1).props()
    ).toEqual({ page: 2, active: true })
  })
```

```
})
```

Fujo de Datos

Metodología de trabajo

1. Empieza con una **vista**

- ¿Qué **data** necesita cada **organismo**?
- Actualiza implementación, historia y test

2. Para cada **organismo**

- ¿Qué **data** necesita cada **molécula**?
- Actualiza implementación, historia y test

3. Para cada **molécula**

- ...

Ejercicio

Modifica la app para que lea el contenido de los JSON

- Desde las vistas hacia abajo
- Actualiza los tests y las historias
 - Añade *knobs* siempre que sea relevante
- Es posible que tengas que reestructurar componentes
 - Sobre todo: crear nuevos átomos

Componentes Sugeridos

Vistas

ProductList

- props
 - data
 - shoppingCart
- contiene:
 - AppLayout
 - IntroTitle
 - ProductGrid
 - Pagination

Vistas

ProductDetails

- props
 - data
 - shoppingCart
- contiene:
 - AppLayout
 - ProductInfo
 - RelatedProducts

Vistas

ShoppingCart

- props
 - `shoppingCart`
- contiene:
 - `AppLayout`
 - `IntroTitle`
 - `ShoppingCartItems`
 - `OrderSummary`
 - `Button`

Vistas

Checkout

- props
 - shoppingCart
- contiene:
 - AppLayout
 - VerticalSeparator
 - IntroTitle
 - AddressForm
 - Card
 - DeliveryMethodSelector
 - OrderSummary
 - Button

Layouts

AppLayout

- props
 - children
 - className
 - shoppingCart
- contiene:
 - AppHeader
 - AppFooter

Organismos

Pagination

- props
 - pages
 - currentPage
- contiene:
 - PaginationItem

Organismos

OrderSummary

- props
 - title
 - subtitle
 - shoppingCart
- contiene:
 - Card
 - CardRow
 - IntroTitle
 - VerticalSeparator
 - Money

Organismos

Product Grid

- props
 - products
- contiene:
 - Grid
 - GridItem
 - ProductSummary

Organismos

Product Info

- props
 - product
- contiene:
 - ProductTitle
 - ProductPicture
 - ProductPrice
 - ProductAvailability
 - ProductDescription
 - ProductQuantity
 - Button

Organismos

Related Products

- props
 - products
- contiene:
 - BetaHeader
 - ProductGrid

Organismos

ShoppingCartItems

- props
 - items
- contiene:
 - ShoppingCartItem

Moléculas

AppHeader

- props
 - shoppingCart

Moléculas

ProductQuantity

- props
 - quantity

Moléculas

ProductSummary

- props
 - product
- contiene:
 - ProductTitle
 - ProductPrice
 - ProductThumbnail

Moléculas

ShoppingCartItem

- props
 - item
- contiene:
 - ProductThumbnail
 - ProductInfo
 - ProductPrice
 - ProductQuantity
 - ProductTotal
 - Button

Moléculas

FormField

- props
 - value
 - label
 - name
 - half
 - error

Moléculas

ProductInfo

- props
 - product
- contiene:
 - ProductTitle
 - Subtitle

Átomos

Money

- props
 - price
 - quantity
 - total

Átomos

PaginationItem

- props
 - page
 - active

Átomos

ProductPrice
ProductTotal

- props
 - product
- contiene:
 - Money

Átomos

ProductAvailability

ProductPrice

ProductThumbnail

ProductTitle

- props
 - product