

Ease 3: Componentes Dinámicos

Componentes Dinámicos

Partimos de...

- Componentes (estáticos) extraídos
- Divididos según *atomic design*
- Storybook con catálogo completo de componentes
- Todos los componentes tienen tests

Componentes Dinámicos

Queremos...

- Componentes que acepten *props*
 - contenedores
 - *contenido* variable (reutilizables)
- Padres *configuran* el *contenido* de sus hijos

```
import AppLayout from '../layouts/AppLayout'

const ProductDetails = () => (
  <AppLayout className="product-details">
    <ProductInfo/>
    <RelatedProducts/>
  </AppLayout>
)

export default ProductDetails
```

```
import PropTypes from 'prop-types'

const AppLayout = ({ children, className }) => (
  <Fragment>
    <AppHeader/>
    <div className="main-content wrapper">
      <div className={className}>
        { children }
      </div>
    </div>
    <AppFooter/>
  </Fragment>
)

AppLayout.propTypes = {
  children: PropTypes.node.isRequired,
  className: PropTypes.string
}

AppLayout.defaultProps = {
  className: ''
}
```

```
import React from 'react'
import { mount } from 'enzyme'
import AppLayout from '../../../src/ui/layouts/AppLayout'

const wrapper = mount(
  <AppLayout className="test-page">
    <span className="test-content">content</span>
  </AppLayout>
)

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

  it('contains a header and a footer', () => {
    expect(wrapper.exists('AppLayout')).toBe(true)
    expect(wrapper.exists('AppFooter')).toBe(true)
  })

  it('wraps the content in a div with "className"', () => {
    const content = wrapper.find('.test-page > .test-content')
    expect(content.text()).toBe('content')
  })

})
```

Componentes Dinámicos

Componentes dinámicos y storybook

- Vamos a instalar *@storybook/addon-knobs*
 - un plugin de storybook
 - ver y modificar el valor de las props
 - <https://npmjs.com/package/@storybook/addon-knobs>

Componentes Dinámicos

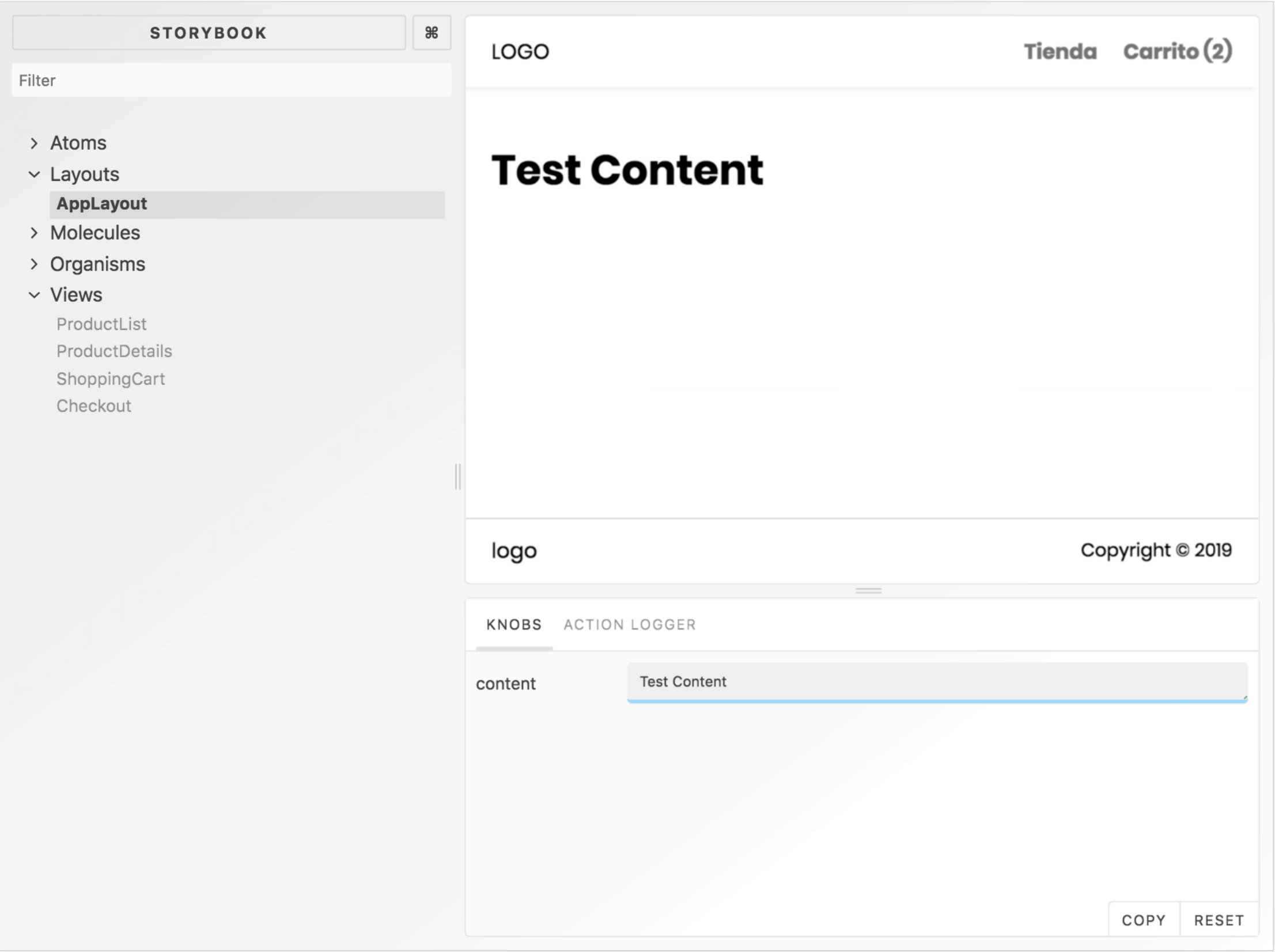
crear *.storybook/addons.js*

import '@storybook/addon-knobs/register'


```
import React from 'react'
import { storiesOf } from '@storybook/react'
import { withKnobs, text } from '@storybook/addon-knobs'
import AppLayout from './AppLayout'

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

  .add('AppLayout', () => {
    return (
      <AppLayout>
        text('content', 'Test Content')
      </AppLayout>
    )
  })
```



Ejercicio

Implementa los componentes dinámicos

- Contenedores
- Saca props en los componentes con contenido variable
 - *documentálas con PropTypes*
- Añade knobs a las historias de los componentes con props
- Testea diferentes valores de las props

Componentes Sugeridos

Componentes Dinámicos

AppLayout

- contenedor
- utilizado en:
 - las 4 vistas
- contiene:
 - todo el contenido de la vista
- props
 - children
 - className

Componentes Dinámicos

Grid

- contenedor
- utilizado en:
 - `ProductGrid`
- contiene:
 - `GridItem` (múltiples)
- props
 - `children`

Componentes Dinámicos

GridItem

- contenedor
- utilizado en:
 - ProductGrid
- contiene:
 - ProductSummary
- props
 - children

Componentes Dinámicos

RadioGroup

- contenedor
- utilizado en:
 - `DeliveryMethodSelector`
- contiene:
 - `Radio` (múltiples)
- props
 - `children`

Componentes Dinámicos

Radio

- utilizado en:
 - `DeliveryMethodSelector` (dentro de `RadioGroup`)
- props
 - `name`
 - `value`
 - `label`

Componentes Dinámicos

FormGroup

- contenedor
- utilizado en:
 - `AddressForm`
- contiene:
 - `FormFields` (múltiples)
- props
 - `children`

Componentes Dinámicos

FormField

- utilizado en:
 - `AddressForm` (dentro de `FormGroup`)
- props
 - `value`
 - `label`
 - `name`
 - `half` (boolean)
 - `error` (string)

Componentes Dinámicos

Card

- contenedor
- utilizado en:
 - `ShippingCart`
 - `Checkout`
- contiene:
 - `CardRow` (múltiples)
- props
 - `children`
 - `transparent`

Componentes Dinámicos

CardRow

- contenedor
- utilizado en:
 - ShippingCart
 - Checkout
- props
 - label
 - children
 - bold (boolean)

Componentes Dinámicos

IntroTitle

- utilizado en:
 - todas las vistas
- props
 - title
 - subtitle
- compuesto por:
 - BetaHeader
 - Subtitle

Componentes Dinámicos

BetaHeader

- contenedor
- utilizado en:
 - IntroTitle
- props
 - children

Componentes Dinámicos

Subtitle

- contenedor
- utilizado en:
 - IntroTitle
- props
 - children

Componentes Dinámicos

Button

- contenedor
- utilizado en:
 - Checkout
 - ShoppingCart
 - ShoppingCartItem
- props
 - children
 - className
 - fullwidth

Componentes Dinámicos

OrderSummary

- utilizado en:
 - Checkout
 - ShoppingCart
- props
 - title
 - subtitle