



WORDCAMP
MANAGUA 2017

HEADLESS WORDPRESS

DANNY MORA @ARGOSBASS

Sobre mí



Danny Mora

Web Developer

+7 Años Experiencia

Twitter: @argosbass



Headless ¿Qué es?



Headless WordPress

- WordPress sin su sistema front end nativo. Siendo utilizado como backend y sirviendo como modelo de datos a una aplicación front end.
- Existen varios tipos de aplicaciones que se pueden considerar Headless WordPress, entre ellos aplicaciones móviles, frameworks front end*, WP-CLI, WP-Console.
- Headless WordPress no es la nueva manera de hacer WordPress. Es una de ellas.

Ventajas

- Mayor control sobre el front end.
- Permite utilizar tecnologías front end de vanguardia.
- Nos permite utilizar WordPress como una parte del stack solamente y no como todas las capas de este.
- Podemos utilizar las herramientas de modelado de contenido que WordPress ofrece.

Desventajas

- Perdemos las funcionalidades y facilidades que nos ofrece WordPress para el front end (funciones de theming, plugins de renderización, etc).
- Debemos crear el modelo de datos dos veces: en WordPress y en la aplicación.
- Necesitamos equipos de trabajo muy especializados para cada parte del stack.



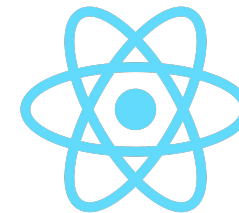
Sitio WordPress

- Instalar WordPress. (<https://wordpress.org/download/>)
- REST API Plugin. (<https://wordpress.org/plugins/rest-api/>)
- Fakerpress Plugin. (<https://wordpress.org/plugins/fakerpress/>)

Ecosistema actual...

Existen varias librerías/frameworks para front end que se encargan de:

- Gestión de modelo
- Enrutamiento
- Comunicación con el servidor
- Composición app y arquitectura
- **Rendering UI ***
- **Eventos UI ***



¿Qué es React?

- React es una **librería de código abierto** escrita en JavaScript, **mantenida por Facebook**.
- Sirve para crear **aplicaciones** de tipo SPA cuyos **datos cambian a través del tiempo**.
- Su objetivo es ser **sencilla, declarativa, y fácil de combinar**.

¿Por qué utilizar React?

- Ofrece grandes beneficios en **modularidad** y **desempeño** gracias al **DOM Virtual**.
- Beneficia el desarrollo orientado a **componentes reutilizables**.
- Tiene **gran aceptación por parte de la comunidad** de desarrolladores.

DOM Virtual

- Es una representación del DOM ubicada en memoria.
- Previene manipular innecesariamente el DOM, ya que es muy costoso.
- React hace una comparativa (diff) entre el DOM actual y el DOM Virtual para conocer qué partes cambiaron y solamente actualizar estas partes.
- Comparar el DOM con el DOM virtual no es costoso y permite realizar el menor número de operaciones necesarias para actualizar el DOM real.

JSX

- Es una extensión de JavaScript similar a XML.
- Es propia de React.
- Permite utilizar etiquetas dentro del código JavaScript. Su uso no es obligatorio pero es recomendable.
- Permite definir bindings ({}), unidireccionales.
- React NO soporta bindings bidireccionales.

JSX vs JS puro

```
const App = function( ){  
  return <div> Hello World! </div>;  
}
```

```
“use strict”  
var App = function App( ){  
  return React.createElement(  
    “div”,  
    null,  
    “Hello World!”  
  );  
};
```

¿Cómo pensar al estilo React?

- Definir un bosquejo.
- Dividir en una jerarquía de componentes.
- Construir una versión estática en React.
- Definir el estado mínimo inicial del estado de la interface.
Ejemplo: la lista original de productos, los valores de los *checkbox* marcados por defecto.
- Definir dónde debe vivir el estado.
- Definir el flujo de datos inverso (`setState()`).

☐ Only show products in stock

Name	Price
Sporting Goods	
Football	\$49.99
Baseball	\$9.99
Basketball	\$29.99
Electronics	
iPod Touch	\$99.99
iPhone 5	\$399.99
Nexus 7	\$199.99

Componentes

- Idea central de React
 - **Método Render:** debe devolver un nodo.
 - **Propiedades:** valores inmutables asociados a un componente
- El método render regenera todo el DOM del componente

Propiedades

- Parte inmutable de un componente.
- Se pasan al componente al usar el tags JSX para instanciarlo
 - `<Mycomponente prop1="" prop2={} />`

Validación de propiedades

- Por tipo.
- Por obligatoriedad:
 - `React.PropTypes.string.isRequired`
- Por valores:
 - `React.PropTypes.oneOf(['valor1' , 'valor2'])`
- Validación personalizada.

Estado

- Parte mutable de un componente
- El componente accede mediante `this.state`
- **REGLA DE ORO**
 - Intentar hacer siempre tus componentes sin estado.

Composición

- Un componente puede instanciar en su función **render** a otro componente.
- El componente instanciado pertenece al componente que le instancia.
- El estado de un componente padre puede ser recibido como propiedades en el componente hijo.

GRACIAS A

RAIN

tigo business
Una solución para cada negocio

 **BOLDGRID**

 **movistar**

 **KRONOSCODE**

 **JETPACK**

 **Woo** **COMMERCE**

 **bluehost**

 **monchi.to**
DISEÑO & DESAROLLO WEB

 **MODERN TRIBE**

 **SiteGround**

 **dot|creek**

 **accedo**
DESIGN - DEVELOP - DESIGN

 **TOPFLOOR**
MARKETING

 **DreamHost**

 **WPML.ORG**

 **SiteLock.**



UCA
UNIVERSIDAD
CENTROAMERICANA

 **dinterweb**
Inbound Marketing Agency

SUCURI

 Hotel **Mansión** **Leodolinda**

 **Bacanalnica**
.com

Preguntas
Comentarios.



Headless WP
@argosbass