
Aplicaciones solidarias basadas en sistemas de gestión de contenido

CMS / Introducción a Wordpress

Agenda

- Introducción a CMS
 - Caso de estudio: Wordpress
 - Plataformas: **solidaridad.uy, trabajandodesdeelbarrio**
 - Proyectos
 - Desarrollo de software en Wordpress
-

Definición (formal)

A content management system (CMS) is a software package that provides some level of automation for the tasks required to effectively manage content.

- Software
 - Contenido
 - Automatización
-

Funcionalidades típicas

- Control de acceso
 - Manejo de estado y *workflow*
 - Manejo de versiones
 - Control de dependencia
 - Persistencia
-

Wordpress como CMS

- Origen 2003 (www.wordpress.org)
 - PHP, MySQL (LAMP stack)
 - Generación de *blogs* hospedados directamente en *wordpress.org* manejable sin conocimientos de computación
 - Ecosistema de temas y plugins gratuitos y pagos
-

Conceptos Principales

- Tema
 - Plugin
 - Usuario
 - Rol
 - Blog Post
 - Page
-

Trabajando desde el Barrio

Buscar

Barrio

Categoría



www.trabajandodesdeelbarrio.edu.uy

Arquitectura

- HTML, CSS, Vanilla JS (ES6)
 - Wordpress
 - *Theme* propio
 - *Plugin GeoDirectory* para ABM oficios y persistencia en base de datos
 - *Role* de colaborador para administrar información de un barrio específico
 - *Posts* para información adicional (derechos laborales, condiciones, etc)
 - API Rest búsqueda
-

SolidaridadUY

Plataforma colaborativa de
iniciativas solidarias en Uruguay

Nombre, Barrio, Ciudad...



www.solidaridad.uy

Arquitectura

- React JS
 - Backend *planillas*
-

Panel de administración

Trabajando desde el barrio

Actualmente, la subida de información a la plataforma se realiza con un formulario provisto por el plugin *GeoDirectory*. Este proyecto consiste en crear una nueva herramienta en www.trabajandodesdeelbarrio.edu.uy para el alta, baja y modificación de oficios.

Tecnologías:

- PHP, MySQL
 - HTML, CSS, Javascript
-

Georeferenciación en Trabajando desde el barrio

Se desea agregar la funcionalidad de buscar trabajadores cercanos a la posición geográfica del usuario que realiza la búsqueda. Se debe realizar la modificación de la base de datos y endpoints a fin de contemplar información geográfica, así como enviar la geolocalización del usuario en los requests.

Tecnologías

- Javascript
 - PHP
 - MySQL
-

Base de datos solidaridad.uy

La base de datos de solidaridad.uy consiste en la actualidad de un conjunto de planillas en *Google Drive*. Este proyecto consiste en analizar el modelo de datos utilizado actualmente, diseñar un esquema apropiado y por último crear y cargar la base de datos en algún DBMS a definir.

Tecnologías:

- Un DBMS apropiado para la tarea (a definir)
-

Backend solidaridad.uy

La aplicación solidaridad.uy se nutre de *endpoints* generados automáticamente por *google drive*, donde se expone la información pública en formato *json* ([ejemplo](#) y [ejemplo](#)). Este proyecto consiste en elegir una tecnología apropiada para desarrollar el backend de solidaridad.uy, y programar los *endpoints* utilizados por la aplicación de react.js.

Tecnologías:

- Servidor de aplicaciones apropiado para la tarea (a definir)
 - React.js
-

—
Desarrollo
En
Wordpress
(1/6)

Directorio de instalación

/www_root/

wp-content

↳ themes

↳ plugins

wp-admin

wp-includes

index.php

wp-config.php

.htaccess

Engine Wordpress

- En esencia un gran *template*
 - Interacción a través de *actions* y *filters*
 - **Action:** Evento disparado por wordpress al ejecutar cierta parte de su *template*
 - Al recibir un request
 - En el lifecycle de los plugins
 - Al publicar posts
 - Al crear o eliminar usuarios
 - ...
 - **Filter:** Evento disparado por wordpress que permite modificar cierto contenido:
 - HTML, CSS, permisos, etc
-

—
Desarrollo
En
Wordpress
(3/6)

Crear nuevo tema

wp-content

↳ themes

↳ theme_ejemplo

↳ style.css

↳ index.php

↳ front-page.php

↳ header.php

↳ footer.php

↳ single.php

↳ functions.php

↳ ...

Crear nuevo plugin

wp-content

↳ plugins

↳ plugin_ejemplo

↳ Plugin_ejemplo.php

- Interactuar con *actions* y *filters*
 - Crear endpoints
 - Interactuar con la base de datos
 - Prover funciones a otros plugins y páginas del sitio
-

API Actions y Filters

https://codex.wordpress.org/Plugin_API

Filter Functions

- `has_filter()`
- `add_filter()`
- `apply_filters()`
- `apply_filters_ref_array()`
- `current_filter()`
- `remove_filter()`
- `remove_all_filters()`
- `doing_filter()`

Actions Functions

- `has_action()`
 - `add_action()`
 - `do_action()`
 - `do_action_ref_array()`
 - `did_action()`
 - `remove_action()`
 - `remove_all_actions()`
 - `doing_action()`
-

Ejemplo Actions y Filters

```
1 function my_the_post_action( $post_object ) {  
2 // modify post object here  
3 }  
4 add_action( 'the_post', 'my_the_post_action' );
```

```
1 add_filter( 'the_content', 'filter_the_content_in_the_main_loop', 1 );  
2  
3 function filter_the_content_in_the_main_loop( $content ) {  
4  
5 // Check if we're inside the main loop in a single Post.  
6 if ( is_singular() && in_the_loop() && is_main_query() ) {  
7 return $content . esc_html__( 'I'm filtering the content inside the main loop', 'wporg' );  
8 }  
9 return $content;  
10 }
```

¿Preguntas?
