# Arquitectura de software Corporación municipal de Valparaíso

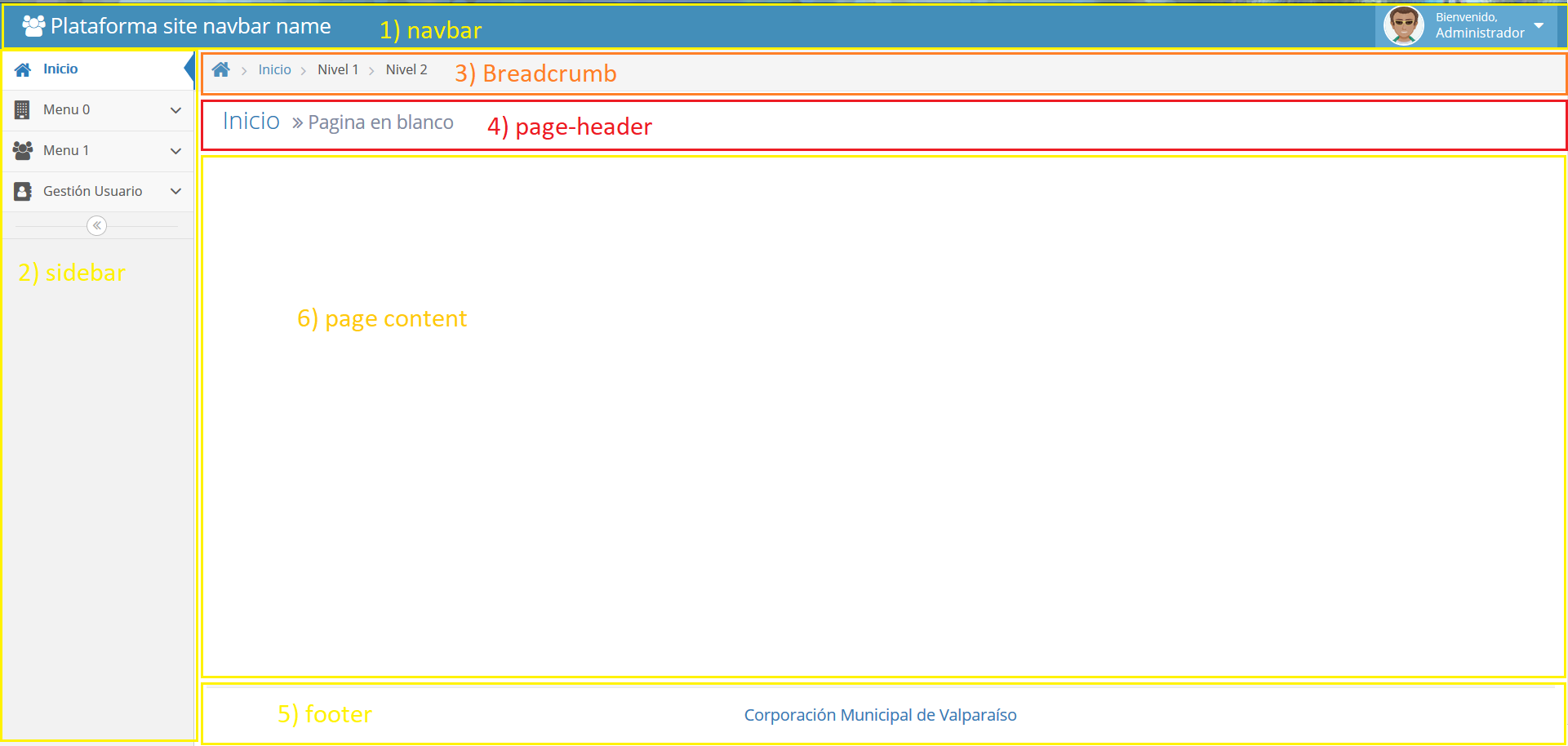
El proyecto base usado en la CMV se basa en CodeIgniter, que implementa el modelo MVC. A este se integra un témplate web basado en Bootstrap y JS, ace demo-template de wapbootstrap. Para el despliegue de datos en tablas estructuradas se integra la librería JS, Datatable. Iconos para menús y botones, Font Awesome 4.7. El sistema de autenticación se basa en la librería Ion-Auth 3.

A continuación, se detallan la lista de dependencias, su documentación y los links para obtener los recursos de forma individual y el Proyecto base que integra todos los elementos.

* CodeIgniter 3.1.10 : <https://www.codeigniter.com/> (Usar manual de la versión 3.1.x)
* Ion-Auth 3: <http://benedmunds.com/ion_auth/>
* DataTable: <https://datatables.net/manual/>
* Frond template: <https://github.com/bopoda/ace>
* Librería Ajax y Jquery: jquery-2.1.4.
* Font Awesome 4.7: <https://fontawesome.com/v4.7.0/icons/>
* CI Proyecto base: <https://gitlab.com/jortizd/ci-proyecto-base> (Solicitar permisos)

## Elementos visuales y construcción de vista

El proyecto base construye una vista tomando diferentes componentes y organizando la información en su respectivo sitio gracias a la estructura del archivo global.php que tiene toda la información base para la vista final (librerías js, css, orden de los componentes) este archivo se encuentra en application/view/global.php, cada componente se administra de forma separada. Los componentes que se describen son:



1. Nadbar: application/views/navbar.php, Se encuentra el menú del perfil de usuario y nombre corporativo.
2. Sidebar: application/views/sidebar.php, Se encuentra el menú de navegación del sistema
3. Breadcrumb: application/views/ breadcrumb.php, seguidilla de las paginas anteriores para llegar a la actual, su función es tener un retorno fácil a las paginas cuando requieran datos adicionales no contemplados en el menú de navegación (Link personalizados).
4. Page-header: application/views/pageheader.php, título y subtitulo de contenido.
5. Footer: application/view/footer.php, bajada de pagina
6. Page Content: Contenido de la página a mostrar, dependiendo de su ubicación normalmente lo encontramos en aplication/views/components dentro de una carpeta con el nombre del controlador que despliega la página.

La construcción de la vista queda en responsabilidad de la clase view\_handler ubicada en application/libraries/view\_handler.php.

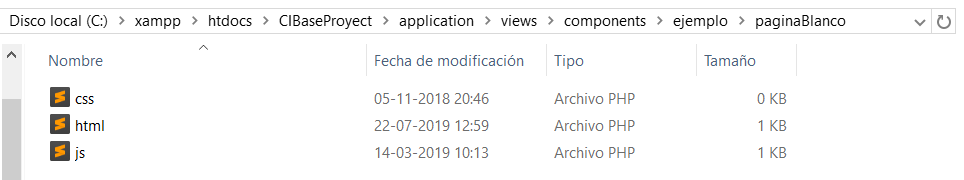
### Ejemplo práctico de una vista

Se crea una clase controllers dentro de la carpeta application/controllers extendiendo de la clase base CI\_Controller

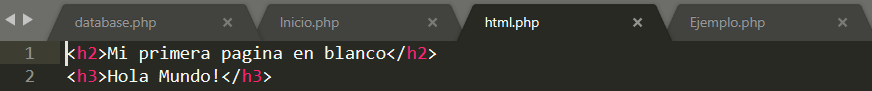


Linea 33: se llama a la función constructora de la vista. Primer y segundo parámetro son los nombres de la carpeta donde están los archivos html, js y css del page content(6).

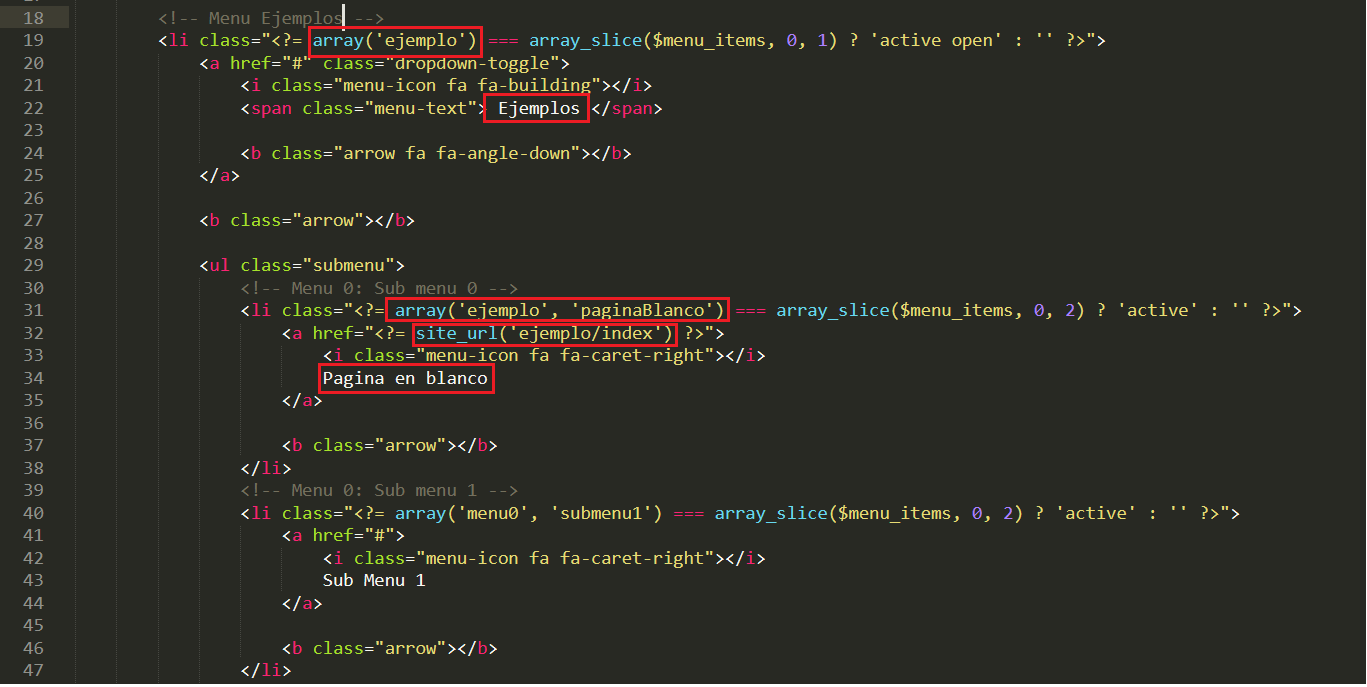
Se crea las carpetas y archivos en blanco del page contentm los cuales tienen que tener los nombres pasados como parámetros al view\_handler->view(carpeta 1, carpeta2, datos). Si la estructura de archivos requiere más de 2 carpeta simplemente incluya el path en el segundo parámetro view\_handler->view(carpeta 1, carpeta2/carpeta3/etc, datos).



En el archivo HTML escribimos un Hola mundo clásico



Modificamos el sidebar(2) para acceder por el menú de navegación



* Linea 19: Reemplazar por el identificador del menú de navegación.
* Linea 22: Reemplazar por el nombre que tendrá el menú de navegación.
* Linea 31: Reemplazar por el identificador del Menú y sub menú de navegación.
* Linea 32: URL de página usando función site\_url(‘costructorName/methodName’).
* Linea 34: Nombre que tendrá sub menú.

Como resultado se obtiene una página en blanco con el contenido y el menú de navegación.

