



**formosa.gov.ar**

el portal de nuestra gente

# Manual de Desarrollo de Portales Oficiales

**Unidad Provincial de Sistemas y  
Tecnologías de Información**

[www.formosa.gov.ar](http://www.formosa.gov.ar)



**Gobierno de la Provincia de Formosa  
República Argentina**

# ÍNDICE

<b>INTRODUCCIÓN</b> .....	3
<b>1. ESTÁNDARES</b> .....	5
<b>1.1. Base de Datos</b> .....	5
<b>1.2. Sistema de Archivo</b> .....	6
1.2.1 Estructura de módulos del Portal .....	6
1.2.2 Estándar y Normalización del Sistema de Archivo .....	7
1.2.3 Estructura de Módulos del Administrador de Sistemas .....	8
1.2.4 Estándar y Normalización del Administrador de Sistemas .....	9
<b>1.3 Variables de programa</b> .....	10
<b>1.4 Modelos</b> .....	10
<b>2. LÓGICA DE PROGRAMACIÓN</b> .....	12
<b>2.1. El sistema de Ingreso y Validación:</b> .....	12
<b>2.2. El sistema de Administración Central administra:</b> .....	12
<b>2.3. El Contenedor de Módulos</b> .....	12
<b>2.4. Los módulos</b> .....	13

## INTRODUCCIÓN

El objetivo principal de este documento es proporcionar a los profesionales de la Administración Pública Provincial (APP) una guía para el desarrollo de portales, sitios oficiales y sus sistemas administradores.

Está basado en los Estándares Tecnológicos Provinciales que determina el Manual de Desarrollo de Portales Oficiales de la Provincia de Formosa, cuyo desarrollo e implementación lo faculta la Resolución Nº 5359/05 (29 de Diciembre de 2005) del Ministerio de Economía, Hacienda y Finanzas de la Provincia de Formosa.

A continuación se describe la tecnología utilizada para el diseño y desarrollo del Portales Web Oficiales. Las mismas son de uso libre (open source) cumpliendo con los criterios establecidos en Plan Estratégico de Gobierno Electrónico Provincial.

- **Debian GNU/Linux: Sistema Operativo**

Debian GNU/Linux es un sistema operativo libre que utiliza el núcleo Linux y la mayor parte de las herramientas básicas vienen del Proyecto GNU. Este sistema operativo fue elegido por su robustez y seguridad para funcionar como un servidor WEB.

- **Apache HTTP Server: Servidor de paginas WEB**

Apache es un software/servidor libre HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP.

- **HTML: Creación de Páginas**

Es un lenguaje diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas WEB

- **CSS: Implementación de Hojas de Estilo en Cascada**

La utilización de Hojas de Estilo en Cascada (**Cascading Style Sheets, CSS**), definen la presentación de los documentos estructurados escritos en **HTML**.

- **PHP: Creación de Páginas dinámicas**

PHP es un lenguaje de programación usado para la creación de contenido para sitios WEB con el cual se puede crear páginas HTML a través de código de fuente.

- **AJAX:**

AJAX, es una técnica de desarrollo web para crear aplicaciones interactivas. Éstas se ejecutan en el cliente, es decir, en el navegador de los usuarios y mantiene comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla. Esto significa aumentar la interactividad, velocidad y usabilidad en la misma.

- **PostgreSQL: Bases de Datos**

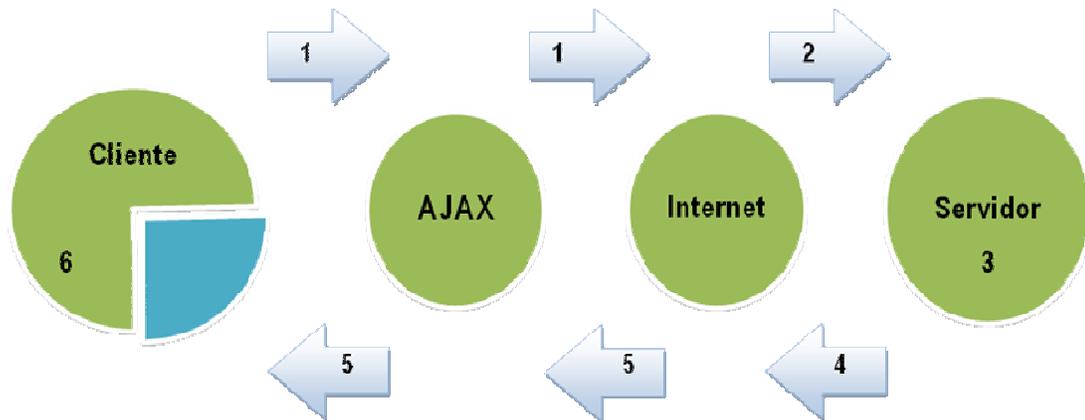
PostgreSQL es un servidor de base de datos relacional orientada a objetos de software libre, liberado bajo la licencia BSD.

- **JQuery**

JQuery es un una biblioteca o framework de Javascript que permite simplificar la manera de interactuar con los documentos HTML, permitiendo manejar eventos, desarrollar animaciones y agregar interacción con la tecnología AJAX a páginas web. jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en Javascript que de otra manera requerirían de mucho más código. Es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

## ▪ Arquitectura del Sistema

El sistema posee la siguiente arquitectura:



1. El cliente envía la petición a través de AJAX.
2. El servidor recibe la petición.
3. El servidor arma la página con PHP (codificación) y templates (Diseño)
4. El servidor envía la respuesta (solo la porción de html a actualizar).
5. El cliente recibe la respuesta a través de AJAX
6. El cliente recarga solo una porción de la página.

## 1. ESTÁNDARES

### 1.1. Base de Datos

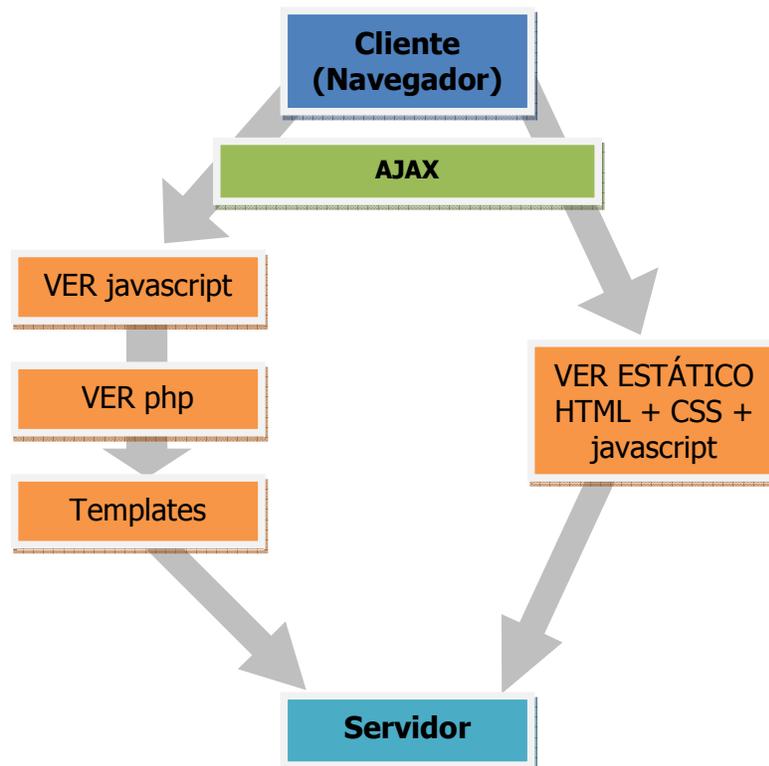
El estándar de la Base de Datos se corresponde con los estándares tecnológicos adoptados por el Gobierno Provincial y definido en la sección 1 del Manual de Desarrollo Framework Grial, disponible en el Portal Oficial del Gobierno

<http://www.formosa.gov.ar/upsti/estandarestecnicos/documentos>

## 1.2. Sistema de Archivo

### 1.2.1 Estructura de módulos del Portal

- Ajax: Interfaz con la tecnología ajax (ejecutar y cargar).
- VER javascript: Se encarga de enviar el pedido del cliente.
- VER php + templates + css: Armar la respuesta del cliente en el servidor.
- VER ESTÁTICO: se encarga de armar el contenido estático del portal.



Dos brazos bien definidos surgen de la estructura de módulos, los cuales dan inicio a la siguiente clasificación:

### *Módulos Dinámicos*

Son los módulos encargados de recuperar información almacenada en la base de datos del sistema. Estos módulos agregan dinamismo al Portal permitiendo a los administradores actualizar la información sin recurrir a desarrolladores web.

### *Módulos Estáticos*

Algunas secciones no requieren actualización permanente. Para cada una de ellas, existen módulos estáticos que publican información (relativamente estable) almacenada en archivo(s) HTML que no tienen ninguna vinculación con la base de datos. Estos módulos deben ser actualizados por los desarrolladores web.

## 1.2.2 Estándar y Normalización del Sistema de Archivo

La estandarización del sistema de archivo es importante para la comprensión del sistema.

### 1.2.2.1 Archivos

Los archivos PHP dentro de la carpeta /nombre\_del\_modulo/php deben tener la siguiente nomenclatura:

ver\_[modulo]\_home.php  
ver\_[modulo]\_busqueda.php  
ver\_[modulo]\_resultado\_busqueda.php  
ver\_[modulo]\_detalle.php  
ver\_[modulo]\_acceso\_directo.php

Los archivos HTML dentro de la carpeta /nombre\_del\_modulo/templates deben tener la siguiente nomenclatura:

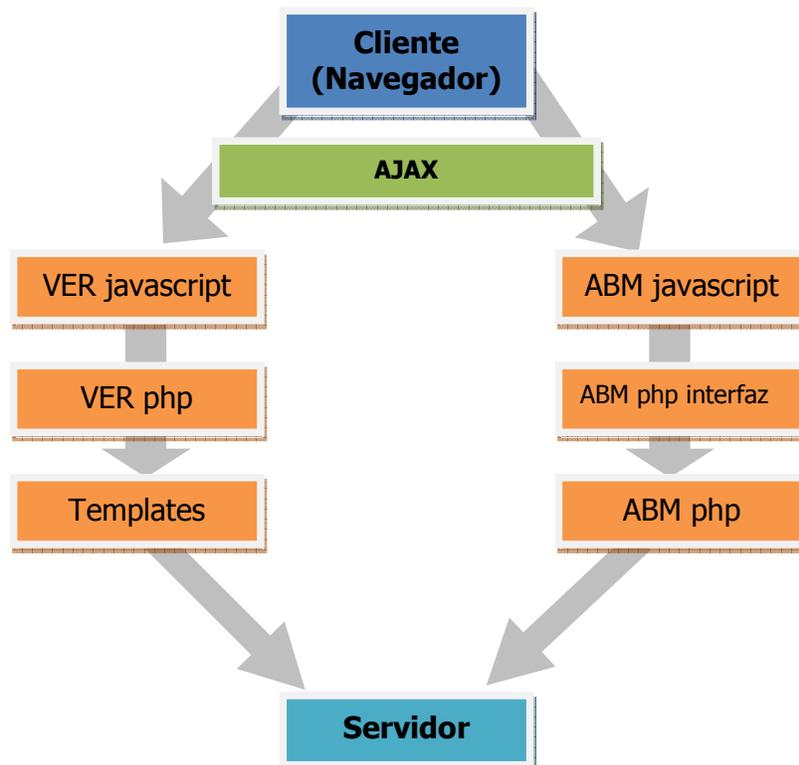
ver\_[modulo]\_home.html  
ver\_[modulo]\_busqueda.html  
ver\_[modulo]\_resultado\_busqueda.html

un\_[modulo].html  
ver\_[modulo]\_detalle. html  
un\_[atributo]\_[modulo].html  
ver\_[modulo]\_acceso\_directo. Html

### 1.2.3 Estructura de Módulos del Administrador de Sistemas

El administrador de sistemas cuenta con una estructura de módulos organizada cuidadosamente según cada capa de la arquitectura (ajax, templates, php, etc.).

- **Ajax:** Interfaz con la tecnología ajax (ejecutar y cargar).
- **VER javascript:** Se encarga de enviar la página al cliente.
- **VER php + templates (css):** Armar la respuesta del cliente en el servidor.
- **ABM Javascript:** se encarga de llamar a las funciones del Módulo **ABM php interfaz**.
- **ABM php interfaz:** define las funciones para interactuar con **javascript**.
- **ABM php:** Conjunto de funciones para dar de alta, baja o modificar la base de datos.



Existen dos tipos de peticiones del cliente, quien en este caso es el usuario encargado de manejar el Administrador de Sistemas. Si el usuario solo quiere recibir información, el flujo de módulos se realiza por el camino izquierdo. Si el usuario agrega, modifica o elimina información del sistema, entonces el flujo se realiza por el camino derecho. Esto permite tener una organización adecuada en el sistema de archivos para optimizar la lectura del código, como se detalla a continuación.

#### 1.2.4 Estándar y Normalización del Administrador de Sistemas

La estandarización del sistema de archivo es importante para la comprensión del sistema.

##### 1.2.4.1 Nombres de archivos

Los archivos PHP dentro de la carpeta /nombre\_del\_modulo/php deben tener la siguiente nomenclatura:

```
abm_[modulo].php  
abm_[modulo]_interfaz.php  
ver_[modulo]_home.php  
ver_[modulo]_busqueda.php  
ver_[modulo]_listado.php  
ver_[modulo]_abm.php  
cargar_archivo_[modulo].php
```

Los archivos HTML dentro de la carpeta /nombre\_del\_modulo/templates deben tener la siguiente nomenclatura:

```
ver_[modulo]_home. Html  
ver_[modulo]_busqueda. html  
un_[modulo].html  
ver_[modulo]_abm. html
```

Los archivos Javascripts dentro de la carpeta /nombre\_del\_modulo/scripts deben tener la siguiente nomenclatura:

```
loader.js
```

ver\_[modulo].js  
abm\_[modulo].js

### 1.3 Variables de programa

#### PHP

Es importante respetar los nombres de las variables de programas. El mismo debe tener el mismo nombre de la columna en la Base de Datos.

Según el ámbito de la variable, la misma debe ser deben tener la siguiente nomenclatura:

- Variables de sesión

`$_SESSION['session_{atributo}']`

Ej: `$_SESSION['session_syscata01_descripcion']`

- Variables local

`#{atributo}`

Ej: `$_syscata01_descripcion`

- Variable de URL

`$_POST["{atributo}"]`

Ej: `$_POST["syscata01_descripcion "]`

#### Templates

Las variables en los templates HTML también tienen que respetar el nombre de la columna (atributo). Las mismas siempre se encuentran encerradas entre llaves.

`<td>{atributo}</td>`

Ej: `<td>{syscata01_descripcion}</td>`

### 1.4 Modelos

El corazón del desarrollo de una aplicación son los modelos. Existen varios modelos predefinidos los cuales se adecuan a los casos más generales. Para los casos particulares se deben tener otros cuidados respetando los estándares tecnológicos de la APP.

Los modelos son:

- Módulos ABM

- Módulos ABM con archivos adjuntos
- Módulos VER
- Módulos VER con paginación
- Módulos VER con búsqueda
- Módulo Pipe Filter

## 2. LÓGICA DE PROGRAMACIÓN

A nivel programación al mismo lo podemos separar en cuatro grandes componente:

- El sistema de Ingreso y Validación,
- El sistema de administración central
- El contenedor de módulos y
- Los módulos

Los tres primeros son componentes indispensables y totalmente reutilizados, ya que son el corazón de toda la ingeniería.

### 2.1. El sistema de Ingreso y Validación:

- controla y valida el acceso al sistema,
- recupera los privilegios del usuario que ingresa,
- carga los módulos a los cuales el usuario tiene acceso
- y habilita las funciones disponibles para dicho usuario en cada módulo

### 2.2. El Sistema de Administración Central administra:

- los módulos,
- los perfiles,
- los usuarios,
- los perfiles de acceso,
- y el organismo de cada usuario

### 2.3. El Contenedor de Módulos

Tiene la función de mostrar cada uno de los módulos que el Sistema de Ingreso y Validación establece según el perfil y organismo del usuario que ingreso al Administrador de Sistemas.

## 2.4. Los Módulos

Son cada una de las funcionalidades del Administrador de Sistemas que, categorizados según requerimientos específicos, tienen el objetivo de organizar y actualizar la información del portal oficial, facilitando el desarrollo y programación del mismo.

Para desarrollar un módulo estándar se debe seguir los siguientes pasos:

- Documentar los requerimientos funcionales del módulo
- Documentar el impacto global en el todo el sistema.
- Definir que TIPO de módulo es (ABM, ABM con archivo adjunto, ABM con paginación, Pipe Filter, etc)
- Definir el alcance del módulo (MULTIUSUARIO, MULTIORGANISMO)
- Diseñar el diagrama relacional del módulo resaltando las relaciones con otras tablas del sistema
- Desarrollar el módulo
- Prueba del módulo
- Implementación del módulo