



Ministerio de Economía,
Hacienda y Finanzas
PROVINCIA DE FORMOSA



formosa.gov.ar

el portal de nuestra gente

Manual de Desarrollo de Aplicaciones Web

Unidad Provincial de Sistemas y

Tecnologías de Información

www.formosa.gov.ar



Gobierno de la Provincia de Formosa
República Argentina

Belgrano 8
Formosa
Tel. (0371)
www.formosa.gov.ar
upsti@formosa.gov.ar

ÍNDICE

INTRODUCCIÓN	3
1. ESTÁNDARES	4
1.1 Framework de desarrollo	4
1.2 Base de Datos	4
1.3 Sistema de Archivo	5
1.4 Variables de programa	7
1.5 Modelos	7
2. LÓGICA DE PROGRAMACIÓN	8
2.1. El sistema de Ingreso y Validación:	8
2.2. El Sistema de Administración Central administra:	8
2.3. El Contenedor de Módulos	8
2.4. Los Módulos	8
2.5 Componentes	9

INTRODUCCIÓN

El objetivo principal de este documento es proporcionar a los profesionales de la Administración Pública Provincial (APP) una guía para el desarrollo de aplicaciones web.

A continuación se describe la tecnología utilizada para el diseño y desarrollo del Aplicaciones Web. 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:**

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.

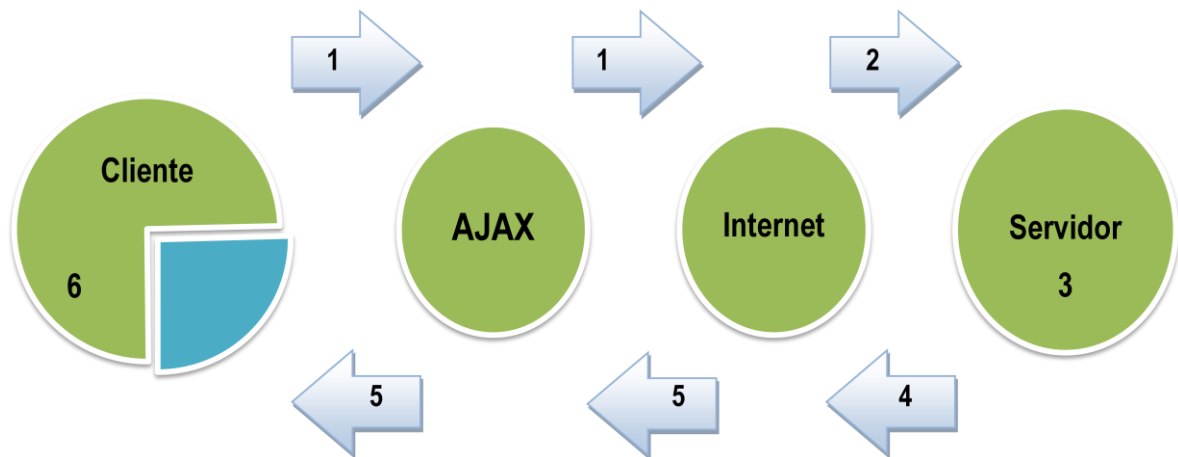
- **JQuery**

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**

El sistema posee la siguiente arquitectura básica:



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

1. ESTÁNDARES

1.1 Framework de desarrollo

El framework utilizado para el desarrollo de aplicaciones web introduce un nuevo paradigma en programación basado en componentes que crean estructuras complejas en tiempo de ejecución.

El mismo no solo facilita la programación de sistemas sino que también ofrece una interfaz amigable, estandarizada y sencilla al usuario final.

Compatible con Mysql, Postgresql y Oracle (fase de prueba).

Toda la interfaz gráfica y la comunicación entre el cliente y el servidor es manejada por la librería jquery, encargada de armar los componentes del sistema.

Utiliza a su vez la librería “templates” (php), que permite separar el código php del código html

1.2 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>

1.3 Sistema de Archivo

Uno de los pilares del Framework es la estandarización de la estructura de directorios. El mismo permite una programación más comprensible y ordenada. Todos los módulos del sistema se almacenan en una carpeta denominada “modulos”. sistema.

El framework almacena los archivos adjuntos en la carpeta “media/uploads”. Dicha carpeta está reservada para todos los módulos que necesitan almacenar adjuntos, ya sea imágenes o archivos. Dentro de esta carpeta se irán creando tantas subcarpetas como sea necesaria (en general, una por módulo). Estas carpetas también son creadas automáticamente por el framework

Por normativa, cada módulo debe tener una carpeta con un nombre representativo de la misma. Dentro, debe contener dos carpetas imprescindibles:

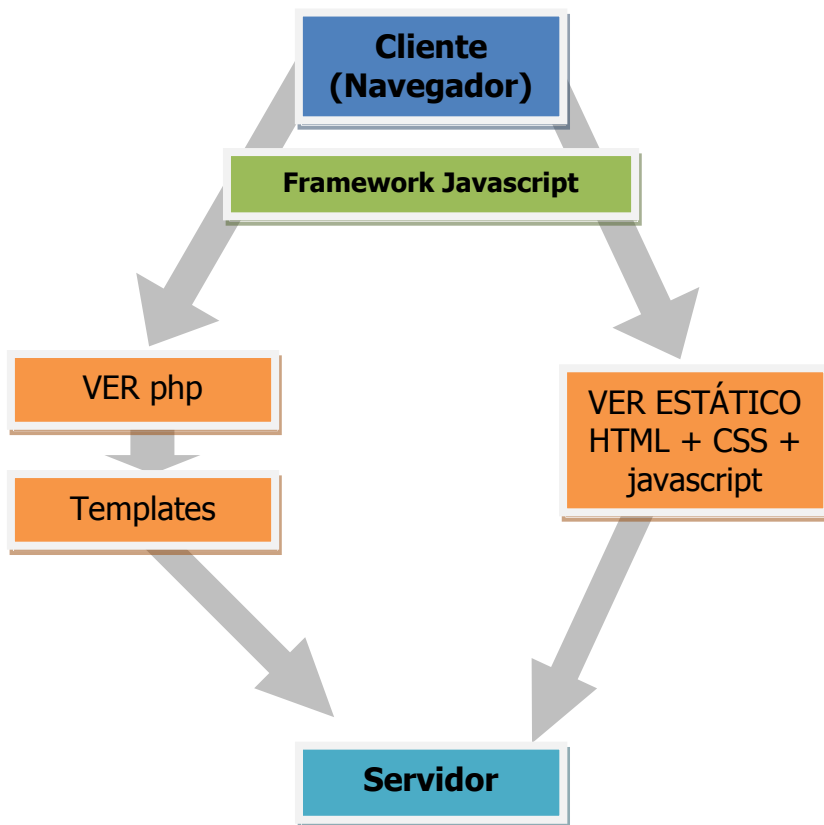
1. “php” para almacenar archivos con extensión php
2. “templates” para almacenar archivos con extensión html

Dentro de la carpeta “php” podemos encontrar (por lo general) los siguientes archivos:

1. ver_[modulo].php: archivo encargado de armar la vista de los datos del módulo
2. ver_[modulo]_abm.php: archivo encargado de armar el formulario (alta/modificación) del módulo.
3. valida_[modulo].php: archivo encargado de las validaciones u operaciones del sistema antes que se inicie la transacción a la base de datos del Formulario.
4. onSubmit_[modulo].php: archivo encargado de las validaciones y operaciones antes de que termine la transacción a la base de datos del Formulario

En la carpeta templates podemos encontrar los siguientes archivos:

5. ver_grid.html: template encargado de armar la vista de datos del módulo. El mismo contiene código que generalmente no es tocado por el programador.
6. ver_form.html: template encargado de armar el formulario ABM del módulo.
7. ver_reporte_[modulo].html: estos archivos son usado para reportes. El mismo contiene la información de cabecera de una tabla.
8. un_[detalle]_[modulo].html: este es usado como template para el detalle de un reporte.



1.4 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>`

1.5 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

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

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 programadas de un sistema, categorizados según requerimientos específicos, tienen el objetivo de organizar y actualizar la información.

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 el TIPO de módulo

- Definir el alcance del módulo
- Diseñar el diagrama relacional del módulo resaltando las relaciones con otras tablas del sistema
- Desarrollar el módulo
- Prueba del módulo
- Implementación del módulo

2.5 Componentes

Todos los componentes del flexmind son objetos javascript, o más precisamente funciones jquery, con una gran cantidad de parámetros a setear. Los componentes son:

- **flexGrid**: Grid, para visualizar información de BD
- **flexForm**: Componente para generar un formulario dinámico
- **flexCombo**: Componente para armar un menú desplegable desde la BD
- **flexTree**: Componente para armar un árbol de registros de la BD
- **flexBox**: Ventana popup del sistema
- **flexMap**: Componente para realizar reportes en google maps
- **flexCalendar**: componente para armar un calendario mensual

La normativa principal es que el **Javascript** siempre envía código en formato **JSON** y el **PHP** siempre devuelve **XML**.

Es recomendable tener conocimientos básicos sobre Ajax, Jquery, JSON y XML.

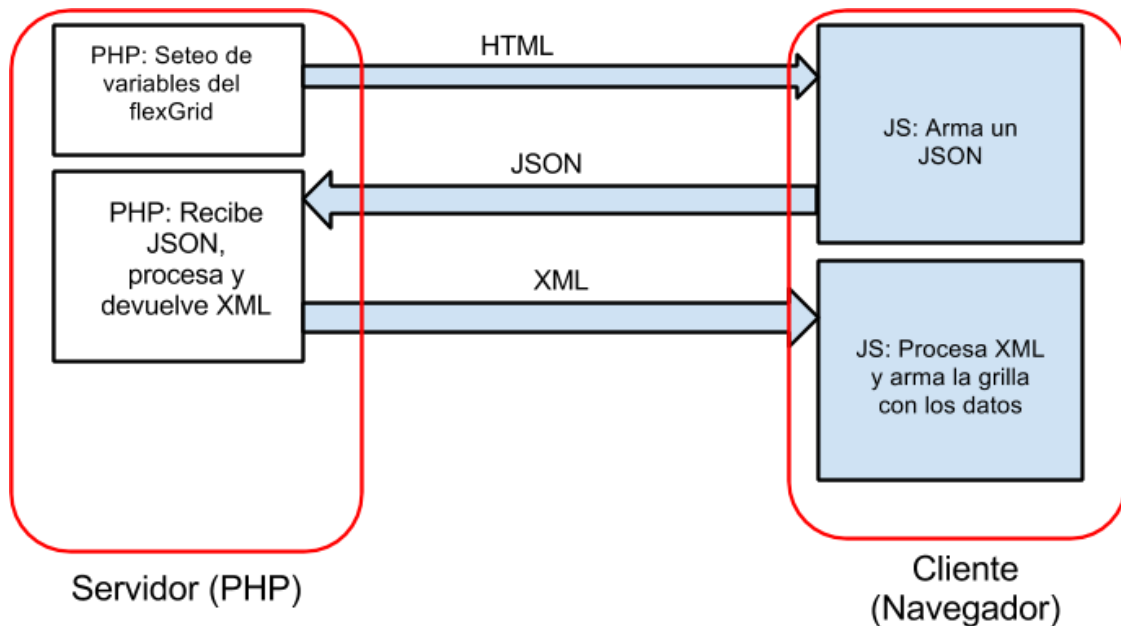
El componente flexGrid y flexForm utilizan (en forma embebida) los otros componentes (a excepción de flexMap y flexCalendar, utilizados a modo de reportes). A continuación se detalla el funcionamiento del flexGrid y flexForm.

Funcionamiento del flexGrid

El flexGrid ejecuta una función javascript con un determinado grupo de atributos. Dichos atributos son seteados por un archivo PHP.

Para visualizar un flexgrid, se ejecuta a un archivo php que inserta código html y javascript en el “**contenedor de módulos**”. El cliente procesa el javascript y ejecuta la función, enviando variables en formato JSON al archivo **flexGridXml.php**, el cual recibe los parámetros, procesa la información, ejecuta la consulta a la base de datos y devuelve un XML. La función javascript que inicializó el proceso recibe este xml y “genera” la tabla de contenidos.

A nivel esquema, el proceso sería el siguiente.



Funcionamiento del flexForm

Este componente es el encargado de las funciones de alta o modificación sobre la base de datos. El funcionamiento sucede en dos etapas:

- Construir el formulario en tiempo de ejecución
- Ejecutar la acción (agregar o modificar)

El armado del formulario consiste en un ejecutar un archivo php que setea las variables del flexForm y envía código HTML y javascript al cliente. El cliente ejecuta la función y arma cada uno de los elementos del formulario junto con los botones, tabs, subforms, etc.

Cuando se presiona el botón guardar ocurre la segunda etapa del componente: **ejecutar la acción**, en ese momento se envía código JSON al archivo **flexFormSubmit.php** y el mismo devuelve un XML informando del éxito o el error de la acción.

En el caso que el archivo **flexFormSubmit.php** devuelve un error de validación del formulario, el XML incluye información adicional de manera que el cliente pueda marcar los errores (marcando en rojo el campo con error o con un mensaje arriba del formulario si es un error global)

A nivel esquema, el proceso es como lo muestra el siguiente gráfico:

