

# DIRECCIÓN DE TECNOLOGÍA DE INFORMACIÓN

## Estándares de Desarrollo de Aplicaciones FOD

2012

---

## Contenido

Historial de Revisiones .....	3
Objetivo.....	3
Desarrollo de Aplicaciones.....	3
Lenguajes de Programación .....	3
Aplicaciones Web (PHP): .....	3
Características .....	3
Inconvenientes .....	4
Lenguaje de Programación WEB Auxiliares (HTML, JavaScript, etc.).....	5
Aplicaciones de Escritorio (C# / Java).....	5
Lenguaje de Programación Escritorio Auxiliar (ActionScript, Visual BASIC) .....	6
Manejadores de Contenido (CMS/LMS) .....	6
Infraestructura .....	6
Base de Datos.....	6
Aplicaciones Web y de Escritorio .....	6
Páginas Web, Portales, Wikis, Aula Virtual, CMS, etc. ....	7
Estándares de diseño y programación Base de Datos .....	7
Objetivo.....	7
Codificación y Nomenclatura .....	7
Reglas generales:.....	8
Código PL/SQL .....	9
Estándares de diseño y programación de Aplicaciones .....	11
Objetivo.....	11
Estructura de Archivos .....	11
Codificación y Nomenclatura .....	12
Reglas Generales .....	12
Seguridad.....	13
Unificada y Centralizada.....	14
Interface de Usuario.....	14

## Historial de Revisiones

Fecha	Versión	Descripción	Responsables
24/08/2012	0.1	Inicio del Documento	Leonardo Correa Flores
5/10/2012	0.2	Modificaciones y Observaciones	Guiselle Víquez
26/12/2012	1.0	Modificaciones y Observaciones	Guiselle Víquez

## Objetivo

El objetivo del presente documento es presentar y plasmar los estándares de las herramientas, metodología e infraestructura básicas de desarrollo de aplicaciones que serán utilizadas por el área de Tecnologías de la Información de la Fundación Omar Dengo en la creación, modificación y estructuración de los distintos programas y aplicaciones desarrolladas a lo interno de la institución y en el seguimiento y control técnico de los desarrollos contratados a terceros.

## Desarrollo de Aplicaciones

Se tomará como desarrollo de aplicaciones aquella interacción que tenga el área de TI en la elaboración o modificación de un código fuente de una determinada aplicación ya sea Web o de Escritorio, que sea desarrollada internamente en la FOD o contratada a un tercero.

## Lenguajes de Programación

### *Aplicaciones Web (PHP):*

Se utilizará como lenguaje de Programación principal el PHP. La selección de este lenguaje corresponde a la difusión global y accesibilidad a documentación del mismo, apertura de código y potencial de desarrollo que ofrece.

Es un lenguaje de interpretación de lado del servidor, pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+.

El mismo puede ser desplegado por la mayoría de servidores Web y en casi todas las plataformas sin costo alguno.

### **Características**

- Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
- Es considerado un lenguaje fácil de aprender, ya que en su desarrollo se simplificaron distintas especificaciones, como es el caso de la definición de las variables primitivas, ejemplo que se hace evidente en el uso de php arrays.

- El código fuente escrito en PHP es invisible al navegador web y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea más segura y confiable que desarrollar directamente en HTML.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL, MSSQL y PostgreSQL.
- Capacidad de expandir su potencial utilizando módulos (llamados ext's o extensiones).
- Posee una amplia documentación en su sitio web oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre (sin costo), por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite aplicar técnicas de programación orientada a objetos. Incluso aplicaciones como Zend framework, empresa que desarrolla PHP, están totalmente desarrolladas mediante esta metodología.
- No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
- Tiene manejo de excepciones (desde PHP5).
- Si bien PHP no obliga a quien lo usa a seguir una determinada metodología a la hora de programar, aun haciéndolo, el programador puede aplicar en su trabajo cualquier técnica de programación o de desarrollo que le permita escribir código ordenado, estructurado y manejable. Un ejemplo de esto son los desarrollos que en PHP se han hecho del patrón de diseño Modelo Vista Controlador (MVC), que permiten separar el tratamiento y acceso a los datos, la lógica de control y la interfaz de usuario en tres componentes independientes.

### **Inconvenientes**

Se debe tener en cuenta los siguientes inconvenientes referentes al uso de PHP en la generación de Aplicaciones Web para evitar ataques o daños a los sistemas en desarrollo o producción o de igual forma sobre fallas posibles de lógica programable.

- Como es un lenguaje que se interpreta en ejecución, para ciertos usos puede resultar un inconveniente que el código fuente no pueda ser ocultado. La ofuscación es una técnica que puede dificultar la lectura del código pero no necesariamente impide que el código sea examinado.
- Debido a que es un lenguaje interpretado, un script en PHP suele funcionar considerablemente más lento que su equivalente en un lenguaje de bajo nivel, sin embargo este inconveniente se puede minimizar con técnicas de cache tanto en archivos como en memoria.
- Las variables al no ser tipadas dificulta a los diferentes IDE's para ofrecer asistencias para el tipeado del código, aunque esto no es realmente un inconveniente del lenguaje en sí.

Esto es solventado por Zend Studio añadiendo un comentario con el tipo a la declaración de la variable.

### ***Lenguaje de Programación WEB Auxiliares (HTML, JavaScript, etc.)***

Para el desarrollo de aplicaciones Web evidentemente es necesario el uso de código HTML, el cual deberá respetar lo estipulado en la Guía para Desarrollo de Aplicaciones Web (**DTI-M-003**).

De igual forma para el contenido ya sea animado o funcional y visual de la página que contendrá el código PHP, se utilizarán los lenguajes de programación y herramientas existentes en el mercado y a las cuales la FOD tiene acceso y permiso de utilizar, tales como HTML, JavaScript, CSS, JQuery, etc.

#### **HTML**

Se utilizará en la construcción de Interfaces, y únicamente puede ser utilizado en la capa de VISTA.

#### **CSS**

Utilizado para darle una mejor apariencia al diseño. El mismo debe ser estándar y debe apegarse a lo que indique el presente documento. **No se podrán utilizar css** que no se encuentren en la carpeta contenedora principal, esto para no romper con los estándares del Diseño. Solo se utilizará en la capa de VISTA en la carpeta llamada **“css”**.

#### **Java Script**

Utilizado para agregar efectos y transiciones de datos en pantalla a la hora de desplegar información. Deben estar contenidos en archivos **“\*.js”** y ubicados en la capa de VISTA en la carpeta llamada **“js”**, sin embargo estos pueden utilizar la capa de CONTROLADOR para tener acceso a datos.

#### **ActionScript**

Es el lenguaje de programación de la Plataforma Adobe Flash y será utilizado en la parte de Aplicaciones Web cuando se requiera el uso de un aplicativo Flash. Estará contenido en la capa de VISTA.

### ***Aplicaciones de Escritorio (C# / Java)***

Se tomarán como aplicaciones Escritorio todos los desarrollos que tengan interacción con la Base de Datos General y que se ejecuten de manera local en el computador, independientemente del Sistema Operativo en el que se trabaje.

Para el desarrollo de aplicaciones de escritorio se utilizarán dos lenguajes de programación como base el C# de Visual Studio .Net con el cual el desarrollo de aplicaciones específicas para el entorno Microsoft Windows es más simple. Sin embargo si se requiere hacer una aplicación Multiplataforma se podrá disponer del lenguaje de programación JAVA.

## ***Lenguaje de Programación Escritorio Auxiliar (ActionScript, Visual BASIC)***

### **ActionScript**

Es el lenguaje de programación de la Plataforma Adobe Flash y será utilizado en la parte de Aplicaciones Web cuando se requiera el uso de un aplicativo Flash. Estará contenido en la capa de VISTA.

### **Visual BASIC**

Se utilizará dentro de las aplicaciones de Escritorio solo si mejora la utilidad, rendimiento o tiempo de programación de alguna utilidad.

## ***Manejadores de Contenido (CMS/LMS)***

Los manejadores estándar de contenidos que se podrán utilizar dentro de la FOD serán **Joomla o WordPress**. La utilización de uno u otro se evaluará según los requerimientos del área o cliente solicitante del proyecto.

Cada arquitectura de Manejador tendrá un control específico, en base a su diseño. Si se necesita modificar alguna funcionalidad o plantilla de un manejador utilizado se hará de acuerdo a lo especificado por cada MS.

## **Infraestructura**

La infraestructura que contendrá las aplicaciones de la FOD estará regida por el área de Infraestructura Tecnológica de la Institución.

## **Base de Datos**

### ***Aplicaciones Web y de Escritorio***

Debido a que dos grandes áreas medulares de la institución utilizan como servidor de Base de datos MS SQL Server 2008 R2, los desarrollos elaborados en el área deberán hacerse utilizando estos mismos como referencia en la parte de base de datos, de tal forma que si deben interactuar dentro de un entorno como un CMS el mismo será dentro de un iframe mediante la llamada a la aplicación local desarrollada, esto para no perder la estandarización que se estructura en este manual.

Se utilizará una sola base de datos llamada **BDSistemaFOD** la cual manejará el modelo global de la base de datos Institucional. Centrando en ella los datos necesarios para el desarrollo normal de todos los módulos deseados en el sistema.

El Analista o diseñador debe realizar los ajustes locales en cada uno de sus equipos a las tablas que no afecten la base central del modelo, es decir, solo deben trabajar sobre las tablas que no sean tablas base del sistema, si se requiere una modificación a esta tabla base del sistema debe coordinarse la modificación del mismo con la Dirección de TI, para que el cambio sea analizado y tomado en cuenta en el mantenimiento del Core Institucional.

La generación de los scripts de la base de datos debe ser lo más portable posible esto debido a que los mismos deben ser ejecutados en el servidor de pruebas cuando se desee colocar una versión como BETA y luego ser replicables a la base de datos en Producción por parte del área de infraestructura sin que afecte el comportamiento normal de los módulos ya realizados.

*Sin embargo, si una aplicación web desarrollada por el área, tuviese que alimentarse de otro motor de base de datos podrá hacerlo mediante las conexiones adecuadas y respetando los lineamientos de seguridad de conexión y extracción de datos que presenta este manual.*

### ***Páginas Web, Portales, Wikis, Aula Virtual, CMS, etc.***

Las bases de datos de estos contenidos serán administradas en el motor de base de datos MySQL, esto debido a las restricciones que presentan las diferentes herramientas utilizadas en su desarrollo. El motor central de las mismas o común en ellas es el MySQL Server.

De ser necesario y para evitar la duplicidad de diseños de interface, estos distintos manejadores de interfaces, llamarán a los módulos de las aplicaciones Web por medio de iframes creados para dicho fin.

### **Estándares de diseño y programación Base de Datos**

Para el desarrollo, diseño y presentación de la base de datos y sus componentes se deben respetar las normas establecidas en el Manual de Diseño de base de Datos, y para el diseño de las interfaces de usuario, tanto reportes como formularios el Manual de Diseño de Interfaces.

El presente manual especifica las utilidades y estructuras auxiliares que deben utilizar en todos los desarrollos de aplicaciones Web realizados por el área de TI de la FOD.

### **Objetivo**

1. Asegurar la legibilidad del modelo de datos, inclusive para personas que no están relacionadas con el ambiente informático, en etapas de análisis y diseño.
2. Facilitar la portabilidad entre motores de bases de datos, plataformas y aplicaciones.
3. Facilitar la tarea de los programadores en el desarrollo de los sistemas.

### **Codificación y Nomenclatura**

La codificación de las tablas de las bases de datos a desarrollar debe cumplir ciertos requisitos, detallados en el presente documento. Estos requisitos pueden aplicarse a cualquier motor de bases de datos.

## Reglas generales:

Únicamente se utilizarán caracteres alfabéticos, salvo que por la naturaleza del nombre se necesiten dígitos numéricos. Se prohíbe el uso de caracteres de puntuación o símbolos.

- **Nombre de la base de dato:** debe comenzar con el prefijo **BD** seguido del nombre de la base de datos comenzando con la primera letra mayúscula.

Ejemplo: BDProduccion

- **Nombres de las tablas:** deben comenzar con el prefijo **T** seguido del nombre de la tabla con la primera letra en mayúscula, y en caso de ser nombre compuesto cada palabra debe empezar con mayúscula (Camel Case), en ningún caso debe exceder 20 caracteres de longitud y debe ser un nombre descriptivo de la información que almacenará.

Ejemplo: TInformacionLaboral

- **Nombre de tablas intermedias:** deben comenzar con el prefijo **R**, con la primer letra en mayúscula, y en caso de ser nombre compuesto cada palabra debe empezar con mayúscula( Camel Case), en ningún caso debe exceder 20 caracteres y el nombre debe ser relacionado a las tablas que une o el proceso que alimenta.

Ejemplo: RMatriculaProfesor

- **Nombres de los campos o atributos:** deben especificarse bajo el estándar CamelCase. Este estándar especifica escribir las palabras compuestas eliminando los espacios y poniendo en mayúscula la primera letra de cada palabra.

Ejemplo: PrimerApellido

- **Llaves Primarias:** deben comenzar con el prefijo **id** en minúscula, seguido del nombre de la tabla (sin el sufijo T). El nombre del constraint en la base de datos debe ir acompañado del sufijo **PK** junto al nombre de la tabla.

Ejemplo: idInformacionLaboral

Ejemplo: PKInformacionLaboral

- **Llaves secundarias:** los campos relacionados con otras tablas deben tener el mismo nombre del campo al que quieren hacer referencia. El nombre del constraint en la base de datos debe ir acompañado del sufijo **FK** junto con el nombre de la tabla más un consecutivo numérico de dos dígitos que inicie en 01 y se incremente cuantas veces sea necesario para hacer diferencia entre cada uno.

Ejemplo: FKInformacionLaboral01, FKInformacionLaboral02



## Código PL/SQL

### Estilo:

La codificación de cualquier sentencia DML (SELECT, UPDATE, DELETE o INSERT) debe emplear la indentación para su clarificación, facilitando la búsqueda de las tablas implicadas (FROM) y de las condiciones impuestas (WHERE).

Cuando están implicadas más de una tabla, deben emplearse, asimismo, las abreviaturas asignadas a cada tabla, vista o secuencia para cualificar los campos ayudando de este modo, al optimizador a realizar el parse de la sentencia.

Dentro del código PL/SQL que se digite, las palabras reservadas deben ser en mayúscula por completo para poder hacer una diferencia visual a la hora de programar.

No se debe incluir código anónimo o que no se encuentre referenciado en una función específica contenido dentro de un BEGIN y un END.

Incluir comentarios de forma obligatoria y en bloque, divididos entre `/* ... */` antes de cada procedimiento o sentencia creada que indique el motivo por el cual se hace, para que sirve, fecha de creación y procedimiento de origen

### Variables de trabajo

Los criterios para nombrar las variables dentro del código son los mismos que para nombrar una tabla, se antepone el prefijo específico para cada tipo de variable (ver tabla a continuación), según el tipo de uso que se le dará en la codificación.

USO	PREFIJO
Constante	C
Contador	I
Variable	V
Global	G
Parametros	P

### Consultas (Queries)

Las consultas se nombrarán anteponiendo el prefijo **Q\_** a un identificador de no más de 20 caracteres. El identificador se derivará de la tabla sobre la que se realiza la consulta. Para consultas complejas, esto es, joins de varias tablas, el identificador se derivará de la tabla más significativa. Si la elección no estuviese clara, se procurará asignar nombres de forma justificada.

### **Funciones**

Los nombres de funciones deben contener el prefijo **FNC** y tener un nombre significativo que identifique a la función.

### **Store Procedures**

Los nombres de los procedimientos almacenados deben contener el sufijo **SP\_** y deben tener un nombre significativo que identifique al procedimiento.

### **Transacciones**

Los nombres de funciones deben contener el prefijo **TNX** y tener un nombre significativo que identifique a la transacción.

## Estándares de diseño y programación de Aplicaciones

### *Objetivo*

Estandarizar la forma en que se desarrollan aplicaciones en la FOD, al menos en forma visual, para que los proyectos en desarrollo sean comprendidos por todos los miembros técnicos del equipo y poder dar continuidad al mismo de una manera fluida y sin curva previa de aprendizaje.

### *Estructura de Archivos*

Los archivos estarán clasificados en 7 carpetas distintas, las cuales se utilizarán de la siguiente manera:

- a. **BASE DE DATOS:** en esta carpeta estarán ubicados todos los scripts de generación o modificación de base de datos, así como los scripts de migración y mini-programas diseñados para la minería de datos y migración de datos en el desarrollo de una aplicación.
- b. **CONTROLADOR:** en esta carpeta se ubicarán los archivos que se relacionan a la capa de controlador del MVC que se está desarrollando.
- c. **DOCUMENTACION:** en esta carpeta se irán colocando todos los archivos de documentación que se van desarrollando a lo largo del desarrollo de los proyectos.
- d. **ENTIDADES:** si en el desarrollo del sistema se necesita crear clases entidades para el funcionamiento del mismo, éstas se colocan en esta carpeta.
- e. **MODELO:** en esta carpeta se ubicarán los archivos de interacción con la base de datos, relacionados a la capa modelo del MVC que se está utilizando.
- f. **SEGURIDAD:** las librerías propias de desarrollo que se utilizan en la aplicación se ubican en esta carpeta, en ella se encuentran los archivos que controlan la capa general de seguridad y comunicación con los web services utilizados para el control de la seguridad del sistema
- g. **VISTA:** en esta carpeta se colocan los archivos de control de interface de usuario, adicionalmente cuenta con librerías y carpetas específicas para los complementos a dicha interface gráfica.
  1. CSS: se almacenan todas las librerías gráficas de control \*.css.
  2. JS: repositorio para los Java Scripts que se utilizan en el desarrollo.
  3. LIB: librerías propias para el control de interface, son parte del FRAMEWORK propuesto de desarrollo.

## Codificación y Nomenclatura

### Reglas Generales

**Nombre del archivo:** el archivo se nombrará de acuerdo al tipo de archivo que se esté generando, de la siguiente manera:

- **Script de Base de Datos:** indicar en la primer parte del nombre que tipo de Script es de los siguientes tipos:
  1. **Estructura:** aquellos archivos utilizados para crear o modificar la estructura de una base de datos, solo contienen los comandos CREATE TABLE, ALTER TABLE, CREATE DATABASE, ALTER DATABASE, ALTER ROLE, ADD USER. El nombre se le antepone el prefijo **“EST\_”** y un titulo representativo de la estructura que está generando.  
*Ejemplo: EST\_BaseDatosClientes.sql*
  2. **Store Procedures:** archivos que contienen la estructura de creación de los Store Procedures que se utilizan en la base de datos. Se le antepone el prefijo **“STP\_”** y un titulo representativo de los procedimientos que contiene.  
*Ejemplo: STP\_BaseDatosCleintes.sql*
  3. **Migración de Datos:** son los archivos que contienen scripts de base de datos para inserción o migración de datos. Se le antepone el prefijo **“MGD\_”** y un título representativo de los datos que afecta.  
*Ejemplo: MGD\_BaseDatosClietetnes.sql*
- **Archivos de desarrollo PHP, C#, Java:** estos archivos contienen el código fuente de desarrollo de aplicaciones, sin embargo se deben respetar los siguientes lineamientos para nombrar el archivo que contiene estos fuentes:
  1. **CONTROLADOR:** son aquellos archivos relacionados con la capa controlador del MVC, se antepondrá como sufijo a estos archivos la letra **“C”** y el nombre del módulo en desarrollo.  
*Ejemplo: VClientes.php*
  2. **MODELO:** son aquellos archivos relacionados con la capa modelo del MVC, se antepondrá como sufijo a estos archivos la letra **“M”** y el nombre del módulo en desarrollo.  
*Ejemplo: MClientes.php*

3. VISTA: son aquellos archivos relacionados con la capa vista del MVC, se antepone como sufijo a estos archivos la letra “V” y el nombre del módulo en desarrollo.

*Ejemplo: VClientes.php*

4. ENTIDAD: no tendrán ningún sufijo específico, solo el nombre significativo de lo que esta entidad controla.

*Ejemplo: Clientes.php*

- **Nomenclatura General:** reglas que se deben seguir al establecer las distintas funcionalidades en el código fuente desarrollado.

1. CLASES: al desarrollar la codificación para las clases estas deben:

- Primer Letra con Mayúscula
- Nombre en Singular

2. FUNCIONES

- CamelCase
- Significar algo

3. INTERFACES

- Inicio del nombre con “I”

4. VARIABLES

- CamelCase
- En los contadores usar algo significativo

## Seguridad

El tema de seguridad es uno de los más delicados en el desarrollo de software, y específicamente para la FOD se debe estructurar el sistema de Seguridad de acceso y datos muy bien.

La característica principal de la Fundación es que tienen una gran gama de tipos de usuario, la idea es unificar a todos en una misma tabla y hacer modular el sistema.

Los tipos de usuarios que se han identificado son los siguientes:

- Administradores de Sistema
- Desarrolladores y Diseñadores
- Usuarios de Prueba
- Docentes
- Asesores

- Área administrativa PRONIE
- Empresa Privada
- Estudiantes

### ***Unificada y Centralizada***

Todos los accesos al sistema deben ser canalizados por una sola tabla que administre los usuarios, esta tabla estará relacionada a una tabla General llamada Persona, la cual esta relacionada a uno o varios tipos de usuario.

### **Interface de Usuario**

En el caso de la interface de usuario el diseño debe basarse en los estándares que defina la FOD, basados en los lineamientos que indique el libro de Marca Institucional y de igual forma es competencia del área de diseño especificar la forma de aplicar dicha interface.

Sin embargo se va a tener una guía básica en el momento de desarrollo del sistema donde se indica el tamaño y forma de los distintos elementos que se utilizarán en el desarrollo.