



**MINISTERIO DE EDUCACIÓN PÚBLICA
DIRECCIÓN DE INFORMÁTICA DE GESTIÓN
DEPARTAMENTO DE SISTEMAS DE INFORMACIÓN**

**MANUAL
ESTANDAR DE DESARROLLO y
MANTENIMIENTO DE SOFTWARE**

**CÓDIGO:
DIG-DSI-MAN-01**

SETIEMBRE, 2021

TABLA DE CONTENIDO

1	HISTORIAL DE REVISIONES	4
2	INTRODUCCIÓN	6
3	OBJETIVOS DEL MANUAL	7
3.1	Objetivo General	7
3.2	Objetivos Específicos	7
4	ALCANCE	7
5	ABREVIATURAS Y DEFINICIONES	8
5.1	Abreviaturas	8
5.2	Definiciones	8
6	AUTORES:	9
7	ACTUALIZADO POR:	10
8	DIRECTRICES	10
9	RESPONSABILIDAD	11
10	DESARROLLO DE SOFTWARE	11
10.1	Generalidades del Desarrollo de Software en el MEP	11
10.2	Lenguaje de Desarrollo	12
10.3	Conformación de los proyectos y sus capas	13
10.3.1	Capa de acceso a Datos	14
10.3.2	Capa lógica	18
10.3.3	Capa de presentación	20
10.4	Diseño de las Pantallas	21
10.5	Reglas para Codificar:	22
10.6	Sistema de Control de Accesos y Permisos.	27
10.7	Implementación de Aplicaciones en la DMZ	30
10.8	Reportes	30
10.9	Manuales e Instructivos	31
10.9.1	Manual Técnico	32

10.9.2	Manual de Usuario	33
11	DOCUMENTOS DE REFERENCIA	33
12	ANEXOS	33
13	HOJA DE FIRMAS	33

1 HISTORIAL DE REVISIONES

Fecha actualización	Versión	Descripción	Responsables
Setiembre, 2009.	1.0	Creación	MARI. Kattia Paniagua Alfaro, Departamento de Innovación y Control Informático. Lic. John Mehlbaum Ucañan, Departamento de Soporte Técnico. Licda. Caroline Gutiérrez, Departamento de Sistemas de Información. Licda. Jacqueline Sequeira Torres, Departamento de Sistemas de Información. Licda. Marcela Ramírez Rojas, Departamento de Innovación y Control Informático. Ing. Fabio López Alfaro, Departamento de Sistemas de Información.
Abril, 2011.	1.1	Actualización	Licda. Grettel Sánchez Calvo, Departamento de Sistemas de Información. Lic. José Martín Sanchún Macín, Departamento de Innovación y Control Informático. Ing. Shirley Calvo Bolívar, Departamento de Innovación y Control Informático.

Febrero, 2012.	1.2	Se le aplica formato y contenido al manual de acuerdo con el Estándar de Documentación de la Dirección de Informática de Gestión.	Lic. José Martín Sanchún Macín, Departamento de Innovación y Control Informático.
Febrero, 2017.	2.0	Actualización de contenido del manual.	MATI. Mayra Torres Aguilar, Departamento de Sistemas de Información.
Setiembre, 2021.	2.1	Eliminación de información obsoleta y entrega de nueva versión para actualización de fondo.	MATI. Mayra Torres Aguilar, Departamento de Sistemas de Información.
		Consolidación de información y revisión de formato en cumplimiento al Estándar de Documentación de la Dirección de Informática de Gestión.	Tec. Jendry Cascante Calvo, Departamento de Gestión y Control Informático.

2 INTRODUCCIÓN

Cada ser humano es diferente y único, lo que nos lleva a un universo de formas de pensar y resolver cada una de las situaciones con la que nos enfrentamos. Que difícil si en una misma sala, introducimos a una persona de cada uno de los idiomas existentes en el mundo para resolver un problema en común. Definitivamente el establecer un dialogo llevaría mucho tiempo y probablemente el problema se agravaría.

El desarrollo de sistemas comprende la misma dinámica, miles de formas de pensar para automatizar y optimizar de la mejor forma el trabajo del día a día de una institución. De ahí la necesidad de buscar y establecer un canal común que nos permita comunicarnos y crear las soluciones tecnológicas en un menor tiempo y con mayor eficiencia. Así como; darles mantenimiento según la evolución de la tecnología y el cambio de las reglas del negocio según sea necesario.

De lo anterior, nace la necesidad de establecer un estándar de desarrollo para soluciones tecnológicas, en el Ministerio de Educación Pública, donde se establezcan las pautas a seguir para lograr la uniformidad de las soluciones y mitigar la dependencia del recurso humano para su mantenimiento.

La estandarización del desarrollo es una de las mejores prácticas que podemos implementar, en la Dirección de Informática de Gestión, con el fin de buscar el éxito en cada uno de nuestros proyectos.

De esta forma le invitamos a conocer este manual e implementarlo en su trabajo diario.

3 OBJETIVOS DEL MANUAL

3.1 Objetivo General

Establecer un estándar que dicte las pautas para el desarrollo de software en el Ministerio de Educación Pública para que sea utilizado tanto por los desarrolladores internos; así como, las contrataciones a terceros que se realicen.

3.2 Objetivos Específicos

- Establecer la arquitectura que deben de tener las soluciones tecnológicas que se desarrollen.
- Establecer un diseño de interfaz única; así como, nomenclaturas, procedimientos y estructuras para los elementos que conforman un desarrollo de software.
- Definir los lenguajes de programación a utilizar en el desarrollo de software.
- Fomentar la reutilización de código.
- Fomentar la calidad de los productos mediante la uniformidad del desarrollo.

4 ALCANCE

Este documento es de aplicación obligatoria para todos los desarrollos de software que se realicen para el Ministerio de Educación Pública realizados por la Dirección de Informática de Gestión; así como, por desarrollos externos.

5 ABREVIATURAS Y DEFINICIONES

5.1 Abreviaturas

MEP:	Ministerio de Educación Pública.
DIG:	Dirección de Informática de Gestión.
CSS:	Hojas de estilo en cascada.
BD:	Base de Datos.
TFS:	Team Foundation Server.
DMZ:	Demilitarized Zone (DMZ) (Zona desmilitarizada).
IIS:	Internet Information Services.

5.2 Definiciones

PasCal Case¹: Este convenio determina que la primera letra de cada palabra debe ser en mayúscula. Ejemplo: ExperienciaProfesional.

caMel Case²: Este convenio determina que la primera letra de la primera palabra debe ser en minúscula y las otras en mayúsculas. Ejemplo: experienciaProfesional.

¹ https://es.wikipedia.org/wiki/Camel_case

² https://es.wikipedia.org/wiki/Camel_case

Web Service³: Un servicio web (en inglés, web service o web services) es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.

DMZ: es una zona de publicación de servicios a internet protegida.

Indentación⁴: es un anglicismo (de la palabra inglesa indentation) de uso común en informática; no es un término reconocido por la Real Academia Española (consultado en la vigesimosegunda edición). Este término significa mover un bloque de texto hacia la derecha insertando espacios o tabuladores, para así separarlo del margen izquierdo y distinguirlo mejor del texto adyacente; en el ámbito de la imprenta, este concepto siempre se ha denominado sangrado o sangría.

6 AUTORES:

Funcionarios	Departamento	Fecha
MARI. Kattia Paniagua Alfaro.	Departamento de Innovación y Control Informático.	Setiembre, 2009.
Licda. Marcela Ramírez Rojas,		
Lic. John Mehlbaum Ucañan.	Departamento de Soporte Técnico.	
Licda. Caroline Gutiérrez.		

³ https://es.wikipedia.org/wiki/Servicio_web

⁴ <https://es.wikipedia.org/wiki/Indentaci%C3%B3n>

Licda. Jacqueline Sequeira Torres.	Departamento de Sistemas de Información.	
Ing. Fabio López Alfaro.		

7 ACTUALIZADO POR:

Funcionarios	Departamento	Fecha
MATl. Mayra Torres Aguilar.	Departamento de Sistemas de Información.	Setiembre, 2021.
Tec. Jendry Cascante Calvo.	Departamento de Gestión y Control Informático.	

8 DIRECTRICES

Este manual no debe ser de conocimiento público dado que podría hacer mención a nombres de servidores y ubicaciones de archivos en la red del MEP.

Todos los funcionarios de la DIG que realicen nuevos desarrollos o den mantenimiento a aplicaciones existentes para el MEP deben utilizarlo, a partir de la fecha de publicación.

Todos los desarrollos, **sin excepción**, deberán ser desarrollados bajo los estándares establecidos en este documento.

Durante el desarrollo de sistemas se debe hacer uso de los lineamientos definidos en este documento de referencia. Para mejorar la estructura y comprensión de las aplicaciones; así como, estandarizarlas con los otros sistemas de la plataforma tecnológica de la DIG.

9 RESPONSABILIDAD

Es responsabilidad de cada uno de los funcionarios de la DIG, que realicen nuevos desarrollos; así como, mantenimientos de sistemas en el MEP, cumplir con cada uno de los lineamientos de este documento. De la misma forma es responsabilidad de las jefaturas (tanto de secciones como a nivel de departamento) velar por el cumplimiento de lo indicado en este manual.

10 DESARROLLO DE SOFTWARE

10.1 Generalidades del Desarrollo de Software en el MEP

Para poder definir algunos estándares de desarrollo es importante definir el ambiente para el cual se va a desarrollar; por lo anterior, todas las aplicaciones que se desarrollen para el MEP deben ser ambiente **WEB**.

Para tener un punto de partida común, se creó un proyecto base que incluye todos los aspectos descritos en este manual. El Proyecto puede ser ubicado en \\dominiomep\\compartidas\\Documentos_Depto_Sistemas\\Estandar_de_Development_de_Sistemas\\ProyectoBase. En esta misma ubicación se pueden encontrar los iconos estándar, objetos de desarrollo (*.dll), entre otros. Además, se crea una carpeta en el

TFS (\$/Proyectos/Sección Arquitectura de Software/Proyecto Base) donde se ubica el proyecto base con la intención de que siempre se acceda a la última versión disponible.

Cada uno de los proyectos que sean desarrollados debe respaldarse en el Team Foundation Server⁵. El encargado del Proyecto debe solicitar, mediante correo electrónico, al administrador del TFS, la creación de una carpeta para el Proyecto. Este debe tener el nombre de la aplicación; además, se debe indicar el nombre de los colaboradores y el tipo de perfil para cada uno. Todo producto que genere el proyecto debe ser organizado de la siguiente forma:

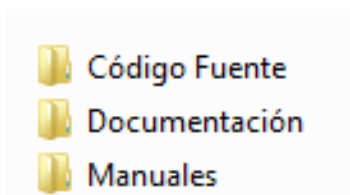


Ilustración 1. Organización de los productos

Cualquier cambio que se genere en un archivo debe ser respaldado mediante el TFS, con esta medida garantizamos el control de versiones y mitigar el riesgo ante cualquier eventualidad que produzca la pérdida de archivos.

Todo desarrollo de aplicaciones deberá ser gestionado según el Manual de Metodología de Gestión de Proyectos de la Dirección de Informática de Gestión, versión 2, código: DIG-SIG-OP-MAN-01⁶, en coordinación con la Oficina de Proyectos.

10.2 Lenguaje de Desarrollo

El lenguaje oficial de desarrollo de software es **Visual Basic. NET o C# 2012 o versión superior**, framework 4.5 o superior. Estas aplicaciones deberán ser publicadas en la

⁵ Instructivo Uso del Team Foundation DIG-DSI-INS-01.

⁶ \\dominiomep\compartidas\Manuales y Estándares\Vigente\Manuales.

LAN, o en la DMZ, según la necesidad de su utilización, por medio de una solicitud vía correo electrónico al administrador designado de los servidores bajo el dominio mep.go.cr.

Si la aplicación requiere de la confección de reportes, estos deben ser confeccionados con Reporting Services⁷ 2014 y publicados en el Server destinado para este servicio. La creación de la carpeta correspondiente deberá solicitarse mediante correo electrónico al administrador del Servidor de Reporting Services (Sección de Arquitectura de Software).

10.3 Conformación de los proyectos y sus capas

Se creará una aplicación en Visual Basic .Net 2012 o versión superior de tipo web de ASP.NET. Para nombrarlo se utilizará el siguiente formato: un nombre descriptivo del proyecto; el cual debe ser significativo al contenido de este, con una extensión no mayor a 3 palabras, las palabras no deben ir tildas y debe evitarse el uso de caracteres especiales; la primera letra de cada palabra debe ir en mayúscula, todo junto sin espacio y sin guion bajo (_), por ejemplo “**EstudiosCarreraAnuales**”.

Asimismo, cada proyecto utilizará una arquitectura de 3 capas como mínimo (Como se muestra en el gráfico 1): capa de acceso a datos, capa de lógica, capa de presentación.

⁷ <http://emb-rprt-01/Reports>

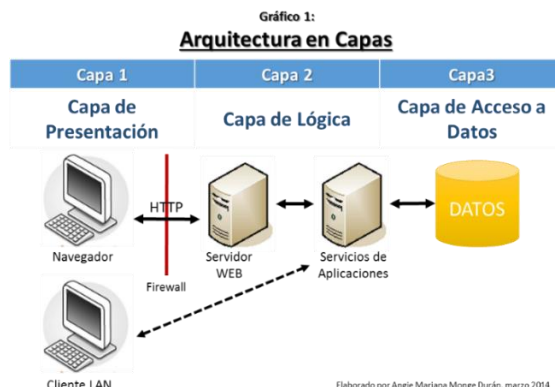


Ilustración 2. Arquitectura en Capas

Para nombrar cada una de estas capas debe ser nombrada utilizando el prefijo “cap” más el nombre correspondiente de la capa (Acceso a datos, Lógica, Presentación), el cual debe ir en mayúsculas todas las iniciales del nombre, sin preposiciones y todo junto sin espacios ni guion bajo () según se muestra en la imagen.

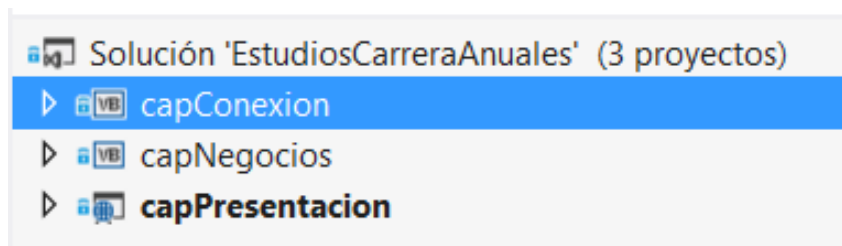


Ilustración 3. Nombramiento por prefijos de las Capas

10.3.1 Capa de acceso a Datos

En esta capa se gestiona el acceso a los datos de la aplicación por medio del gestor de base de datos autorizado por el Departamento de Bases de Datos.

El contenido de esta capa incluye todo el código específico del origen de datos. Incluye funciones y procedimientos de uso general para la ejecución de scripts, procedimientos

almacenados, funciones, vistas, entre otros y según las necesidades de extracción de datos que puede utilizar el sistema. Además del código de conexión a la base de datos.

Ejemplo:

```
Public Function EjecutaConsultaSQL(ByVal str_Sql As String, ByVal int_Conexion As Integer) As Boolean

    EstablecerCadenaConexion(int_Conexion)

    Dim SqlCmd As New SqlClient.SqlCommand

    Try
        ConexionSQLServer.Open()
        With SqlCmd
            .Connection = ConexionSQLServer
            .CommandType = CommandType.Text
            .CommandText = str_Sql
            .CommandTimeout = 100000
        End With

        SqlCmd.ExecuteNonQuery()
        Return True
    Catch ex As Exception
        Return False
    Finally
        SqlCmd.Dispose()
        ConexionSQLServer.Close()
        ConexionSQLServer.Dispose()
    End Try

End Function
```

Ilustración 4. Ejemplo de consulta a la base de datos

Es importante aclarar que todos los comandos Select, Insert, Update y Delete deben crearse en la base de datos mediante procedimientos almacenados, funciones, vistas, entre otros o mediante la utilización de Web Services que faciliten el mantenimiento de las aplicaciones y eviten la interrupción del servicio.

La conexión a la BD se puede definir:

- Por medio de un archivo .dll (clase), donde se importe la referencia al motor de base de datos que se va a utilizar:

```
Imports System.Data
Imports System.Data.SqlClient
Public Class Conexion

Public Conn As New SqlConnection

Public Const ConnString As String = "Server=[Nombre
Server];Database=[Nombre DB];User Id=[usuario];Password=[Password];"

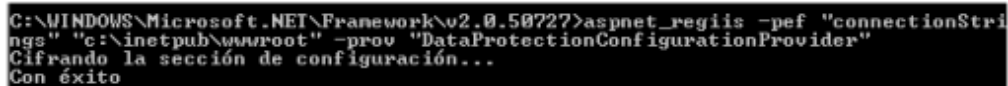
End Class
```

- En la capa de acceso de datos

```
Dim CnnSQLServer As String = "Data source='Nombre del Servidor';initial
catalog=nombre de la base de datos;User Id=usuario;Password=password;"
```

- En el web.config: por motivo de seguridad, si la preferencia del desarrollador es incluir el o los string de conexión en el archivo web.config es necesario que proceda a proteger la cadena de conexión⁸ en coordinación con el administrador del Servidor de Aplicaciones.

Para encriptar la sección correspondiente al string de conexión y cifrar su contenido es necesario ejecutar un comando propio de NET Framework. Es necesario ubicar el directorio de NET Framework correspondiente a la aplicación Web (según la versión del framework que se esté utilizando) y ejecutar el siguiente comando:



```
C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727>aspnet_regiis -pef "connectionStri
ngs" "c:\inetpub\wwwroot" -prov "DataProtectionConfigurationProvider"
Cifrando la sección de configuración...
Con éxito
```

Ilustración 5. Encriptación del string de conexión

Esta acción no requiere que en el código de la página ASP.NET en la que usamos la cadena de conexión realice ninguna acción especial, pues es el

⁸ [https://msdn.microsoft.com/es-es/library/ms254494\(v=vs.110\).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-4](https://msdn.microsoft.com/es-es/library/ms254494(v=vs.110).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-4)

propio IIS el que realiza la encriptación y des encriptación. Al leer la cadena desde la página ASPX, IIS ya realiza el descifrado. Para recuperar el archivo original, es necesario utilizar el mismo comando con otro parámetro:

```
C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727>aspnet_regiis -pdf "connectionStrings" "c:\inetpub\wwwroot"  
Descifrando la sección de configuración...  
Con éxito
```

Ilustración 6. Desencriptación del string de conexión

Es importante tener en cuenta:

1. El cifrado depende del servidor de IIS; por tanto, si tenemos dos entornos **NO** es posible copiar el archivo web.config con secciones cifradas de un servidor a otro pues en el nuevo no puede interpretar el cifrado. Debe ejecutarse el comando individualmente en ambos servidores, de ahí la necesidad de coordinar con el administrador del IIS.
2. Existen dos tipos de cifrado (que indicamos en el parámetro –prov) que se pueden usar: DataProtectionConfigurationProvider y RsaProtectionConfigurationProvider. Al desencriptar no es necesario indicar el proveedor pues ya se especifica en el archivo

Es importante recalcar que las conexiones a BD se hacen mediante un usuario único; es decir, el usuario y password utilizados en el string de conexión son proporcionados por el Departamento de Base de Datos, por lo que se debe hacer la solicitud de creación del usuario con la observación de que el password no debe tener vencimiento.

10.3.2 Capa lógica

La capa lógica de negocios ocupa un lugar relevante en la construcción de una infraestructura de software, al permitir el crecimiento y la extensión de servicios para todas las aplicaciones existentes y futuras.

En esta capa se reciben todas las peticiones del usuario y se envían las respuestas después de procesadas. También se le puede llamar “capa lógica de negocios”, porque es aquí donde se establecen todas las reglas, procesos y restricciones del entorno de negocio, los cuales deben cumplirse para que pueda procesarse cada solicitud. Está se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos el almacenamiento y la recuperación de datos.

Se debe incluir: las tareas que forman parte de los procesos, las reglas y restricciones que aplican. No toda la lógica de negocios es la misma, algunas no requieren un frecuente acceso a los datos, pero una interface de usuario robusta necesitara de la lógica de negocios para la validación en la entrada de campo, cálculos en tiempo real u otras interacciones de usuarios.

En esta capa se deben crear clases que contengan las reglas de negocio, divididas según características que sean definidas por los desarrolladores y que permitan realizar el mantenimiento de las aplicaciones de forma eficiente.

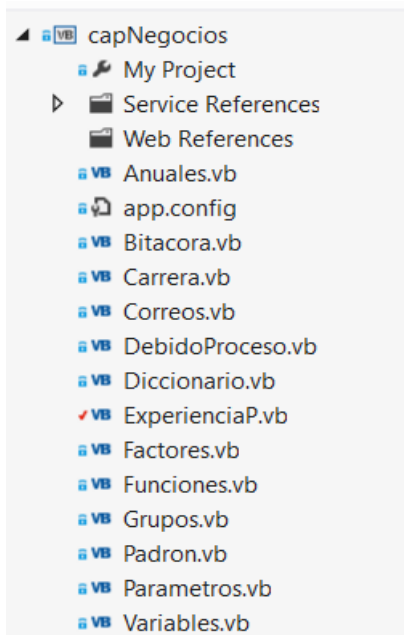


Ilustración 7. Capa de Negocios

En esta capa también se deben declarar las referencias de Web Services. Actualmente existen Web Services de consulta a las siguientes bases de datos:

- INTEGRA2 : <http://webapps.intranet.mep.go.cr/webseviceintegra/>
- Sigrh_real: <http://webapps.intranet.mep.go.cr/webseviceSIGRH/>
- Sumas Giradas de más: <http://webapps.intranet.mep.go.cr/webseviceSGM/>
- DIG: <http://webapps.intranet.mep.go.cr/webseviceDIG/>
- Tribunal Supremo de Elecciones: <https://apps.mep.go.cr/WSTSE/>
- SABER: <http://webapps.intranet.mep.go.cr/webseviceSABER/>
- Pagos: <http://webapps.intranet.mep.go.cr/websevicePagosIntegra/>

Toda nueva necesidad de web services a una base de datos de consulta o webmethod debe ser solicitado mediante correo electrónico a los funcionarios de la Sección de Arquitectura de Software.

Todas las consultas a catálogos se deben hacer por medio de web services, para evitar así las islas de información.

10.3.3 Capa de presentación

El objetivo primordial de esta capa es la separación de la lógica de negocios de la lógica de diseño. La ventaja principal de este estilo es que el desarrollo se puede llevar a cabo en varios niveles y, en caso de que sobrevenga algún cambio, sólo es necesario actuar sobre el nivel requerido sin que sea necesario realizar modificaciones en el código de los restantes niveles.

La capa de presentación es la que ve el usuario, presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo proceso. Esta capa se comunica únicamente con la capa de negocio. Por lo que no trabaja directamente con datos; en su lugar, invoca clases y métodos en la capa de acceso a datos para todas las solicitudes de datos.

También es conocida como interfaz gráfica y debe tener la característica de ser "amigable" (comprensible y fácil de usar) para el usuario. Típicamente incluyen el manejo y aspecto de las ventanas, el formato de los reportes, menús, gráficos y elementos multimedia en general.

Es importante crear una estructura de directorios que permita la clasificación y rápida e identificación de los formularios.

Ejemplo:

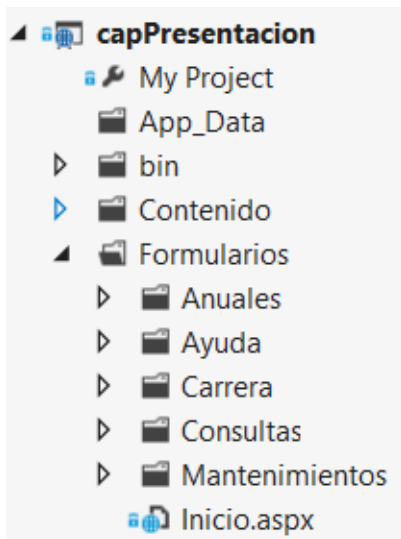


Ilustración 8. Capa de Presentación

10.4 Diseño de las Pantallas

Toda aplicación debe llevar el nombre de la sección en el encabezado de la ventana, por medio de la etiqueta <title>. Por lo que debe llevar un nombre descriptivo que represente la aplicación. Así mismo, se debe seguir el aspecto de las pantallas que contiene el “Proyecto Base”.

La página debe de estar estructurada en dos secciones: encabezado y contenido. El tamaño de la plantilla se debe definir de manera proporcional al tamaño de la pantalla, de modo que la mayor parte del contenido sea visible a primera instancia, para que el usuario no se vea obligado a descender en forma vertical por más de 3 pantallas, la definición de estos tamaños depende mucho del contenido y el diseño de la plantilla, por lo que es recomendable que durante la fase de pruebas del proyecto se ejecute la aplicación en varios monitores de diferentes tamaños y en varias resoluciones para poder detectar algún problema y hacer las correcciones del caso.

Actualmente existe una hoja de estilos (CSS), incluida en el “Proyecto Base”, para ser utilizada por los desarrolladores de software, para que la apariencia de todas las pantallas del sistema sea homogénea.

Por otra parte, se recomienda separar los archivos de presentación, es decir todos los archivos de estilo generados deben ir en una sola carpeta con el nombre de “Content” o “Contenido” en la capa de presentación. Asimismo, en otra carpeta llamada “Scripts” se deben almacenar los archivos .js, script de JavaScript; los cuales son muy importantes para el funcionamiento de los controles que ayudan a mejorar la experiencia del usuario.

Cabe indicar, que todos los enlaces de la aplicación deben tener íconos o textos que representen el contenido al que apuntan, por lo que deben ser claros e intuitivos. Todas las entradas de datos a la aplicación (Ejemplo: TextBox) deben estar debidamente validadas con rutinas o mascarar que aseguren que el tipo de dato y/o longitud que ingresa el usuario es el correcto. Además, se deben incluir filtros para ordenar las listas de información que se muestran a los usuarios.

Se debe contemplar la exportación de información a archivos en diferentes formatos (Excel, Pdf, entre otros). Incluso, considerar todas las opciones posibles de búsqueda de información (Ej. Fecha, nombre, apellido, clase de puesto, institución, entre otros).

10.5 Reglas para Codificar:

- Mantener las clases cortas, con no más de 2.000 líneas de código y que estén divididas en estructuras.

- Crear un archivo para cada clase, el nombre del archivo debe ser igual a la clase.
- Utilizar PasCal case para los nombres de las clases e interfaces.
- Las interfaces deben utilizar el prefijo I, manteniendo la primera letra del nombre en mayúscula.
- Los nombres de las clases deben utilizar sustantivos o adjetivos que describan el comportamiento.
- No añadir prefijos al nombre de la clase.
- No utilizar puntos en el nombre de directorios o clases.
- Utilizar Tab para la indentación del código fuente. No se deben utilizar espacios.
- Si una expresión no se puede mantener en sola línea de código, dividirla de acuerdo con:
 - ✓ Empezar la nueva línea después de una coma.

Ejemplo:

```
If SisrecaService.InsertarFuncionarioAdministrador(HttpUtility.HtmlDecode(CorreoTextBox.Text.Trim),  
HttpUtility.HtmlDecode(NombreTextBox.Text.Trim), Context.User.Identity.Name) Then
```

Ilustración 9. Nueva Línea después de una coma

- ✓ Empezar una nueva línea después de un operador aritmético. Es importante buscar operadores con la prioridad más alta. Ejemplo:

```
If Puntos + CInt(PuntosFactorAcumulados \ CInt(dr_Parametros(0).Item("int_horas_punto"))) >  
dr_Parametros(0).Item("int_punto_maximo_total") Then  
Mensajes("Sumatoria de puntos excede el maximo permitido")  
VerificaSuperaMaximoPuntosCI = True  
End If
```

Ilustración 10. Nueva Línea después de un operador

- Alinear, usando Tab y complementar con espacios (solo de esta forma se permiten los espacios, para efectos de orden), la nueva línea siempre con el inicio de una sección en la que se encuentra el código.
- El código se debe organizar en regiones, para facilitar la comprensión y organización del mismo, de modo que cualquier otro programador interno o externo pueda asumir el proyecto sin problemas, además se logrará realizar un mantenimiento más eficiente cuando sea necesario.
- Los comentarios de línea se utilizan para explicar cada línea de código fuente, deben tener el mismo nivel de indentación que el código al que hacen referencia y la longitud no debe exceder la línea de código que explica.
- Para descripción de clases y sus respectivos miembros, evite los comentarios de bloque.
- Comentarios de código, son utilizados para describir el funcionamiento del código, todos los métodos deben utilizar el formato XML. Existen herramientas tanto en Visual Studio, como de terceros para generar documentación automática a partir de Tags XML. Las etiquetas pueden ser de dos tipos:
 - ✓ Elementos de documentación: como <summary>, <param> o <exception>. Estas etiquetas se relacionan con los elementos de la API del programa que se documenta, suelen tener atributos que son revisados por el compilador; siendo así, deben tener valores válidos.
 - ✓ Formato / referencia: Etiquetas como <code>, <list> o <para> que permiten definir el formato que la documentación debe tener.
- Antes de la implementación, quite todos los comentarios temporales o innecesarios.

- Declaraciones:
 - ✓ En la codificación de las clases y sus miembros no se deben incluir espacios entre el nombre de un método o utilizar el guion bajo (_).
 - ✓ Se debe utilizar PasCal Case.
- Sentencias:
 - ✓ Se debe incluir una sentencia por línea.
 - ✓ No utilizar paréntesis para encerrar el valor de retorno. Ejemplo: No utilizar `Return (True)`. Utilizar `Return True`.
 - ✓ Utilizar sentencia try catch que especifiquen que hacer si una excepción es generada en el bloque try. Utilizar la sentencia finally al finalizar el bloque Try para asegurarse de realizar un dispose de los objetos cargados en memoria.
 - ✓ Las sentencias “case”; así como todas las estructuras de control (if, while, for, etc.) deben estar indentadas.
- Espaciado:
 - ✓ Es importante utilizar líneas en blanco que separen los bloques de código que están relacionados lógicamente.
 - ✓ Estas deben de estar indentadas como si contuvieran una sentencia para facilitar la inserción de código en caso de ser necesario.
 - ✓ Debe haber un espacio luego de una coma o punto y coma; así como, alrededor de los operadores.
- Asignación de nombres:

- ✓ Se debe nombrar con la Notación Húngara los elementos de la GUI⁹ añadiendo como sufijo el tipo de objeto sin abreviar. Esta será la única forma de utilización de esta notación dado que simplifica el uso del IntelliSense para la digitación de código. Ejemplo: emailTextbox, emailLabel, etc.
 - ✓ Los parámetros y nombres de variables deben ser camel case. No utilizar la notación húngara
 - ✓ Utilice nombres de variables de una sola letra, como i o j sólo para índices (ciclos).
 - ✓ Evite el uso de caracteres especiales para nombrar variables, procesos y funciones como \$, ', %, ñ o Ñ, ü, etc.
 - ✓ Los nombres de las enumeraciones; así como sus valores debe utilizar el Pascal Case. No se debe utilizar la Notación Húngara.
 - ✓ Deben utilizar nombres en singular para enumeraciones que permitan la escogencia de un solo valor y en plural para aquellos que permitan una selección múltiple.
-
- Es de suma importancia que en cualquier desarrollo que se vaya a hacer se tome en cuenta la reutilización de código de otras aplicaciones; para funcionalidades similares o iguales, pues puede ayudar a simplificar la programación de alguna funcionalidad del sistema y de igual forma ayuda a reducir el tiempo de desarrollo del proyecto, aprovechando así mejor los recursos asignados al proyecto.

⁹ Interfaz Gráfica de Usuario

10.6 Sistema de Control de Accesos y Permisos.

En cuanto a la seguridad de la aplicación se debe utilizar el Sistema de Control de Accesos y Permisos; al cual, todos los sistemas que utilicen el usuario de Active Directory, deben conectarse para verificar los permisos (perfil) de cada usuario para el uso de una aplicación, de esta manera se elimina el desarrollo de un módulo de administración de seguridad en cada sistema.

El Sistema de Control de Accesos y Permisos es una aplicación web (<http://aplicaciones.intranet.mep.go.cr/Seguridad>) que permite centralizar los permisos de acceso a los diferentes sistemas usados en el MEP. De una forma centralizada otorga acceso a los usuarios de los sistemas; es decir, desde una sola aplicación se le puede conceder la autorización de ingreso al usuario a uno o varios sistemas a la vez, en lugar de tener que ir sistema por sistema para darle acceso.

Su uso puede ser consultado en el Manual de usuario “SISTEMA DE CONTROL DE ACCESOS Y PERMISOS DIG-DSI-SAS-MAN-01”.

Para cada desarrollo, el encargado del proyecto debe enviar un correo a la Sección de Arquitectura de Software solicitando el registro del sistema en el Sistema de seguridad y control de acceso, de esta forma le será devuelto un “ID” que deberá cambiar en el proyecto base y con el cual el Sistema de Control de Accesos y Permisos identificará cada sistema.

```
Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
    Try
        lblVersion.Text = "Versión del Sistema: " + System.Reflection.Assembly.GetExecutingAssembly.GetName.Version.ToString
        Dim dsInfoUsuario As DataSet
        Dim intAplicacion As Integer
        'Consecutivo que genera el Sistema de Control de Accesos y Seguridad
        intAplicacion = 2 'Codigo del ambiente de Producción.
        intAplicacion = 71 'Codigo del ambiente de pruebas.

        dsInfoUsuario = CType(LoginMEP1.VerificarLogeo(intAplicacion, Nothing), DataSet)
        If dsInfoUsuario IsNot Nothing Then
            If dsInfoUsuario.Tables("tblUsuario").Rows.Count > 0 Then
                Session("NombreUsuario") = dsInfoUsuario.Tables("tblUsuario").Rows(0).Item("Nombre")
                Session("CedulaUsuario") = dsInfoUsuario.Tables("tblUsuario").Rows(0).Item("Cedula")
                Session("Usuario") = dsInfoUsuario.Tables("tblUsuario").Rows(0).Item("Usuario")
            If dsInfoUsuario.Tables("tblPermisos").Rows.Count > 0 Then
                Session("Perfil") = dsInfoUsuario.Tables("tblPermisos").Rows(0).Item("vchIdPerfil")

                If dsInfoUsuario.Tables("tblPermisos").Rows(0).Item("dti_Ult_logeo") Is DBNull.Value Then
                    LoginMEP1.IncluyeLogeo(CStr(Session("Usuario")), intAplicacion)
                    Session("ModulosPermisos") = dsInfoUsuario.Tables("tblPermisos")

                    Response.Redirect(ResolveUrl("Inicio"))
                Else
                    If DateDiff(DateInterval.Day, CDate(dsInfoUsuario.Tables("tblPermisos").Rows(0).Item("dti_Ult_logeo")), CDate(Now.T
                        Mensajes("Lleva más de un Mes de Inactividad, comuníquese con el Administrador del Sistema para que le habilite
                    Else
                        LoginMEP1.ActualizaLogeo(CStr(Session("Usuario")), intAplicacion)
                        Session("ModulosPermisos") = dsInfoUsuario.Tables("tblPermisos")
                        Response.Redirect(ResolveUrl("Inicio"))
                    End If
                End If
            End If
        End Try
    End Sub
```

Ilustración 11. ID – Sistema de Control de Accesos y Permisos

Para estandarizar el acceso a los diferentes sistemas, se utiliza el usuario y password establecido en el Active Directory, de esta forma el usuario y password de un funcionario siempre será el mismo, independientemente de la aplicación.

Para el logeo de los sistemas que se utilizan por medio de la LAN o la oficina virtual se utiliza un objeto pre desarrollado llamado Logincontrol.dll, también se encuentra incluido en el proyecto base, es el responsable de la comunicación entre el sistema en desarrollo y el Sistema de Control de Accesos y Permisos.

El control de logeo está desarrollado para que sea implementado como un Control Personalizado (dll) que se carga en la barra de herramientas del Visual Studio. Se recomienda que se cree un espacio solo para cargar controles personalizados.

Para utilizarlo debe:

- Sobre la barra de herramientas, clic derecho,
- “Añadir Pestaña”, después dentro del espacio de la nueva Pestaña,

- clic derecho, “Escoger Elementos” y se busca la ubicación física del archivo “LoginControl.dll”.

Es importante recordar que estos elementos ya están incorporados en el Proyecto Base por lo que solo se deberán incluir los datos variables para cada aplicación.

Para utilizar el control, solo debe arrastrarlo al formulario. Una vez agregado al formulario, en el evento Page_Load del formulario Login.aspx, se invoca el evento VerificarLogeo del control de logeo.

Al dispararse este evento del control, devuelve al formulario un dataset (dsInfoUsuario) que contiene dos tablas con la siguiente información:

Tabla 1. Formulario un dataset

Tablar	Campos
tblUsuario	Cédula Nombre Usuario
tblPermiso	intIdAplicacion: Consecutivo de la aplicación vchIdPerfil: Perfil intIdModulo: Id Modulo vchModulo: Nombre del Modulo dti_Ult_logeo: Fecha del último logeo

Si la aplicación tiene más de un módulo, la tabla permisos tendrá más de un registro por lo que el procedimiento del evento Page_Load deberá ser modificado según la información que sea devuelta por el control.

Además, en el evento Page_Load actualiza la fecha del último logeo realizado y valida si han pasado 30 días desde su último acceso a la aplicación con el fin de deshabilitar el acceso.

10.7 Implementación de Aplicaciones en la DMZ

Todo desarrollo que deba ser implementado en la DMZ, mantendrá su base de datos en la LAN. La comunicación entre ambos debe de realizarse por medio de web services que utilicen canales de comunicación que están debidamente configurados en la red y que permiten este tipo de comunicación.

10.8 Reportes

Los reportes deben ser diseñados e implementados por medio de Reporting Services. Deben estar divididos de la siguiente forma:

Encabezado: se debe ubicar el logo del Ministerio en la esquina superior izquierda, además en la parte central se coloca el nombre de la aplicación y debajo de este el nombre del reporte. Finalmente, debajo de todo se pone una línea para dividir la sección.

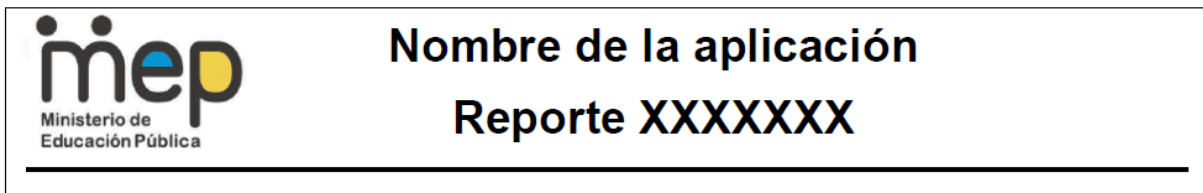


Ilustración 12. Encabezado de Reportes

Contenido: en esta sección se colocaran todos los campos incluidos en el reporte, dependiendo de la consulta ejecutada, algunos de ellos pueden ser el nombre, número de cedula, departamento, fecha, detalles, etc. Todo depende de los requerimientos del usuario final. Es necesario que los nombres de los campos sean claros y ubiquen al usuario en la información que muestra el reporte rápidamente.

Pie de página: se debe trazar una línea divisoria para indicar el inicio de la sección en la parte inferior al lado izquierdo se debe colocar el nombre del usuario que generó el reporte, en el centro la cantidad de páginas del reporte; indicando el número de página actual y el número total de páginas, finalmente al lado derecho se debe colocar la fecha y hora en que se está generando el reporte.

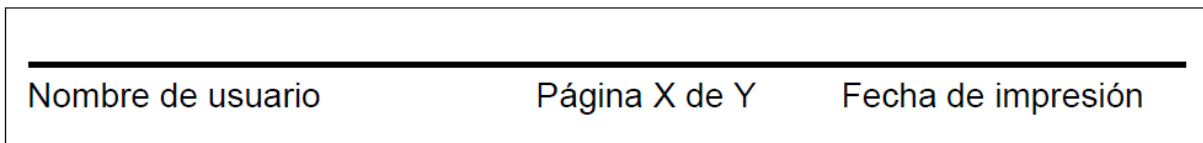


Ilustración 13. Pie de Página Reportes

10.9 Manuales e Instructivos

Los manuales e instructivos que se generen en cada proyecto de desarrollo se deben realizar siguiendo lo establecido en el Manual de Estándar para Documentación vigente.

10.9.1 Manual Técnico

El contenido del manual propiamente debe ser:

- Descripción de herramientas utilizadas (Requerimientos de Hardware y Software, motor de base de datos, reportes, lenguaje de desarrollo, entre otros).
- Descripción de tablas de la base de datos¹⁰, incluye el diagrama de entidad relación, el diccionario de datos.
- Descripción de procedimientos almacenados, triggers, funciones y scripts utilizados en el sistema, se debe crear una tabla para cada uno de estos que indique la funcionalidad, parámetros de entrada, parámetros de salida, entre otros.
- Pantallas del sistema, para cada pantalla se debe indicar la funcionalidad, tablas que consulta o modifica, variables de Session() o cualquier característica especial.
- Reportes del sistema, indicar para cada reporte la información que se lista.
- Interrelación, indicar con claridad todos aquellos sistemas, bases de datos, webservices, entre otros, que utilice el sistema.
- El manual Técnico no debe incluir el código fuente de la aplicación o queries (SQL), toda esta información y/o documentación deberá estar respaldada en el TFS.

¹⁰ Se debe guardar en el TFS los Scripts de creación de objetos en la BD: para la creación de tablas, llaves primarias y foráneas, índices, constraints(restricciones), roles, usuarios y cualquier otro componente de la base de datos. En el manual se debe incluir únicamente la referencia

10.9.2 Manual de Usuario

- Debe detallar cada módulo y componentes que conforman la interfaz, descripción de cada una de las pantallas que conforman el sistema, detallando su uso con ilustración de estas y su respectiva utilidad. Se debe indicar claramente la funcionabilidad de cada uno de los botones.
- Descripción detallada de los reportes.

11 DOCUMENTOS DE REFERENCIA

- Manual de Usuario Sistema de Control de Accesos y Permisos, versión 1.1, código DIG-DSI-SAS-MAN-01.
- Instructivo Uso del Team Foundation, versión 1, código DIG-DSI-INS-01.
- Manual de Metodología de Gestión de Proyectos de la Dirección de Informática de Gestión, versión 2, código: DIG-SIG-OP-MAN-01.
- Manual de Estándar para Documentación, versión 7, Código DVM-A-DIG MAN-01

12 ANEXOS

No hay anexos.

13 HOJA DE FIRMAS

MINISTERIO DE EDUCACIÓN PÚBLICA
DIRECCIÓN DE INFORMÁTICA DE GESTIÓN
HOJA DE REVISIÓN Y ACEPTACIÓN
MANUAL ESTANDAR DE DESARROLLO y
MANTENIMIENTO DE SOFTWARE
CÓDIGO: DIG-DSI-MAN-01

ACTUALIZADO POR:

MATI. Mayra Torres Aguilar

NOMBRE

FIRMA

Tec. Jendry Cascante Calvo

NOMBRE

FIRMA

REVISADO POR:

Lic. Gabriel Dennis Dennis

NOMBRE

FIRMA

MARI. Kattia Paniagua Alfaro

NOMBRE

FIRMA

APROBADO POR:

Lic. José Sandí Zúñiga

NOMBRE

FIRMA

23/09/2021