CAPITULO V

**5. IMPLEMENTACIÓN DEL SISTEMA**

* 1. **Prototipos del Sistema**

 **5.1.1 Utilidad**

El realizar un prototipo del sistema tiene por objetivo ejecutar los programas para encontrar las posibles falencias del mismo. Se desarrolla un prototipo del programa para probar que no existen errores en el desarrollo de un programa, es decir, hacer que una implementación no falle en su ejecución.

La confiabilidad es un aspecto del diseño, por lo que debe estar dentro del sistema, más adelante se indican las estrategias específicas que se utilizaron en el prototipo del sistema.

 **5.1.2. Principio básico de las pruebas**

La prueba del equipo lógico es el método más usado para determinar sistema funciona como debe. El proceso de pruebas es uno de los componentes de un conjunto de actividades que permiten asegurar la calidad del producto realizado.

Uno de los principios básicos en la realización de pruebas es que estas han de ser llevadas a cabo por personas distintas a los diseñadores de los programas, tanto para evitar una simple verificación de que el programa funcione correctamente, como para probar que ese programa ha sido concebido e interpretado correctamente.

* + 1. **Realización de pruebas**

Para cada sistema se realizarán diferentes clases de pruebas:

*Pruebas Unitarias*: Todos los componentes del sistema que se desarrollen individualmente para comprobar su correcto funcionamiento.

*Pruebas de integración*: Se prueba la integración entre los componentes del sistema para demostrar que se puede encajar correctamente.

 *Pruebas de sistemas*: Se prueba el sistema globalmente.

* + - 1. **Tipos de pruebas**

 Existen dos tipos de pruebas :

Pruebas del tipo CAJA BLANCA, que permite examinar la estructura interna del programa.

Pruebas del tipo CAJA NEGRA, donde los casos de prueba se diseñan considerando exclusivamente las entradas y salidas del sistema, sin preocuparse por la estructura interna del mismo.

 **5.1.3.1.1. Desarrollo incremental**

Al probar cada módulo individualmente es necesario crear módulos auxiliares que simulen las acciones de los módulos indicados por el módulo que se está probando.

 El tipo de prueba incremental consiste en agregar cada módulo o componente individual al conjunto de componentes existentes y el conjunto resultante de prueba. Esto reduce la necesidad de crear módulos conductores y permitiendo además, examinar en detalle las interfases. Cuando las pruebas unitarias y de integración se realizan separadamente, es difícil examinar los componentes individuales o módulos que causan resultados incorrectos, por el contrario, con el tipo de prueba incremental es probable que surjan al incorporar un nuevo componente a un grupo previamente probado, sean debidos precisamente a este último o a las interfases entre él y los otros componentes.

 **5.1.3.1.2 Estrategias de integración**

Es importante determinar la secuencia en que van a producir e integrar los componentes. Para una estructura de programas organizada jerárquicamente, se podrá enfocar el problema de la prueba utilizando varias estrategias diferentes:

 - Estrategias de arriba a abajo (Top-Down)

 - Estrategias de abajo a arriba (Bottom-Up)

 - Estrategias combinadas

* Comparación de estrategias.

 **5.1.3.1.3. Pruebas del sistema y de aceptación**

1. ***Pruebas Globales***. Una vez que se han probado los componentes individuales y se han integrado, se ha de probar el sistema global. En esta etapa pueden distinguirse los siguientes tipos de prueba, cada uno con un objetivo claramente definido:

 - Pruebas funcionales

 - Pruebas de comunicación

 - Pruebas de rendimiento

 - Pruebas de volumen

 - Pruebas de sobrecarga

 - Pruebas de disponibilidad de datos

 - Pruebas de facilidad de uso

 - Pruebas de operación

 - Pruebas de entorno

 - Pruebas de seguridad

***b) Pruebas de aceptación.*** Son aquellas pruebas que realiza el usuario con el objeto de comprobar si el sistema es aceptable. Estas pruebas son del mismo tipo que las mencionadas anteriormente, pero son determinadas por el usuario, en lugar de serlo por el equipo de desarrollo.

Un lugar especial de estas pruebas es el de la ejecución en paralelo con el viejo sistema, para comparar los resultados producidos por ambas ejecuciones.

* + - * 1. **Planificación de las pruebas.**

La fase de pruebas, por su envergadura e importancia necesita una organización seria y fiable. Ante una fase de pruebas, se debe tomar como axioma que se van a encontrar errores.

 Los componentes de una planificación ser:

* Objetivos: Definir los objetivos de cada fase de las pruebas.
* Criterios de terminación: Especificar cuando se deben acabar las pruebas.
* Cronología: Fijar los tiempos necesarios para cada fase diseño, escritura, ejecución).

- Responsabilidades: Especificar los responsables de cada fase, así como quién corregir  los errores detectados.

* Bibliotecas de caso de prueba y normas: Crear técnica de identificación, escritura y almacenamiento de casos de prueba.
* Herramientas: Establecer cuáles pueden ser  las herramientas de pruebas que se van a utilizar.
* Tiempo de máquina: Determinar el tiempo que se necesita en cada fase del proyecto de prueba.

- Configuración de equipo: Detallar la necesidad de hardware y software especiales de equipo o de un período concreto.

* Integración: Describir el plan de integración del sistema.
* Métodos de seguimiento: Especificar los métodos que se han de utilizar en las pruebas.
* Depuración: Definir un mecanismo para informar sobre los errores detectados, para seguir el proceso de las correcciones y para incorporar éstas al sistema.
* Terminación de las pruebas. Es difícil que el último error detectado, era el único que quedaba. Sin embargo, existen métodos para mostrar cuando esta próximo el final, los dos más comunes son:

- Terminar la prueba cuando el tiempo establecido para la misma ha expirado.

- Terminar la prueba cuando todos los casos de prueba se ejecutan sin detectar errores.

 Otros métodos mas complicados de aplicar pero más efectivos son:

- Estimación de número total de errores del programa.

- Estimación del porcentaje de estos errores que pueden encontrarse fácilmente.

* Estimación de qué fracción de errores se originan en procesos particulares de diseño.

 Para realizar cualquiera de estas estimaciones es necesario contar con una historia o experiencia previa que permita definir dichas estimaciones.

* + 1. **Análisis del prototipo**

A pesar de los mejores esfuerzos para la implementación del sistema, las necesidades de información no siempre se establecen correctamente. Esto puede ocurrir porque los usuarios pueden saber sólo que necesitan mejorar el sistema en ciertas  áreas, o que deben de modificar los procedimientos existentes, o simplemente no están muy seguros del tipo de información que necesitan.

Por lo tanto, se pueden considerar los prototipos para proporcionar la información necesaria y formular el diseño o desarrollar el sistema, debido a que un prototipo es un sistema de trabajo que se desarrolla con rapidez para probar las ideas y el entendimiento sobre el nuevo sistema, en otras palabras no sólo es un diseñó en papel, sino un software que produzca información impresa o en pantalla.

Los objetivos de esta técnica del prototipo son:

- Validar la funcionalidad del sistema.

- Probar los principios y verificar que el sistema funciona

- Incrementar el nivel de productividad del desarrollo de sistemas.

- Busca un acercamiento (interacción) entre el usuario y el sistema.

- Presentar en forma jerárquica las pantallas y el diálogo con el usuario.

 - Obtener una realimentación (evolución del prototipo).

Temas complejos de procesamiento de datos tienen la necesidad de prototipos, mediante éstos se busca obtener una retroalimentación del análisis y del diseño antes de realizar su construcción.

Para tal efecto, se trabaja con el usuario final creando diálogos (pantallas) para entrada de datos, consultas, menús, efectuar demostraciones de ellos y realiza generación de informes, manipulación de información por pantalla.

Al construir el prototipo se facilitó el desarrollo del sistema, éste se realizó de las tres formas siguientes; un prototipo en papel que describa la interacción hombre-máquina, de forma que facilite al usuario la comprensión de como se producir  tal interacción, un prototipo que funcione y por ultimo uno que implemente algunos subconjuntos de la función requerida del sistema requerido.

* + - 1. **Necesidad de utilizar un prototipo**

No se conocen los requerimientos

La naturaleza de esta aplicación es tal, que la información disponible en la base de datos creada en SQL Server no está actualizada con respecto a las características que debe tener el sistema para satisfacer los requerimientos de los usuarios.

Este problema se origina porque se dispone de los datos obtenidos en el IV Censo de Población y III de Vivienda y todavía no se puede tener acceso a la información actualizada en el 2001. Sin embargo el sistema posee un módulo de actualización de datos que sirve para solucionar esta falencia.

Nueva tecnología

El deseo de instalar nueva tecnología ya sea en los campos de la informática y comunicación de datos u otras áreas relacionadas, abre nuevas fronteras. Se evaluó la mejor posibilidad para la implementación de este sistema que inicialmente se realizaría en una aplicación en Visual Basic 6.0 con la utilización de la misma forma de la digitalización de imágenes en Macromedia Fireworks 4.0.

Después de la evaluación del sistema se determinó que su programación y diseño, debería implementarse vía Internet a través de una página web codificada en Active Server Pages (lenguaje de programación ASP) y demás herramientas descritas en la sección 5.1.

* + 1. **Determinación del Sistema a implementar**

El desarrollo de este prototipo y su evaluación final se llevó a cabo en una forma ordenada, sin importar las herramientas utilizadas. Los pasos seguidos fueron los siguientes:

Identificación de requerimientos conocidos

La determinación de los requerimientos de esta aplicación es tan importante para el método y ciclo del desarrollo del mismo. Por consiguiente se determinaron los requerimientos de hardware, software e interfaz de usuario y así establecer un estándar del sistema, además de conocer si se contaba con las herramientas necesarias.

 Desarrollo de un modelo de trabajo

Es útil comenzar el proceso de construcción del prototipo con el desarrollo de un plan general que permita a los usuarios conocer lo que se espera de ellos y del proceso de desarrollo. Un cronograma para el inicio y el fin de la implementación es de gran ayuda, al igual que la utilización de un bosquejo.

Para el desarrollo del prototipo se prepara los siguientes componentes:

* La interfaz del usuario. El lenguaje para el diálogo entre el usuario y el sistema, pantallas y formatos para la entrada de datos.
* Módulos esenciales de procesamiento. Modulo de digitalización de imágenes, módulo de consulta y búsqueda de información y módulo de actualización de datos.
* Salida del sistema. Diseño de la consulta y búsqueda de información.

Para construir este prototipo se tuvo que seguir estándares para que los datos que fueron empleados estuvieran ordenados y organizados (longitud de datos, características, tipo, entre otros).

 Revisión del prototipo

Durante la evaluación del sistema se capturó información sobre la satisfacción del usuario (lo que le agrada y desagrada). Los cambios al prototipo son planificados con los usuarios antes de llevar a cabo tales modificaciones.

Se realizaron revisiones periódicas de la implementación y diseño del sistema; por parte del Director del Proyecto. De esta manera se deberá asegurar que las especificaciones del diseño y arquitectura del sistema se ajusten a las especificaciones funcionales obtenidas en la fase de análisis de sistemas. Y por último, se obtendrá la aprobación formal del presente diseño.

* 1. **PROGRAMACION DEL SISTEMA**

Tanto en este como en otros sitios Web están escritos utilizando Visual Basic Script, normalmente referido como **VBScript**, orientado casi al 100% a su uso con el entorno ASP por lo que obviaremos su utilidad en la programación en el lado del cliente (como sustituto de Javascript).

### Justificación del lenguaje VBScript

Los motivos que en mi opinión han motivado el uso extendido de VBScript para programar las páginas Active Server son principalmente:

* **Similitud con Visual Basic**. Es cierto que la gran mayoría de desarrolladores ASP, especialmente los primeros, no venían del entorno Web sino del de programación. Era más normal que un programador Visual Basic hiciese la programación de un sitio Web, utilizando por tanto VBScript por ser un subconjunto de lo que ya sabía.
* **Sencillez**, tanto en sintaxis como en funcionamiento. VBScript (y cualquiera de los lenguajes de la familia Visual Basic) son probablemente los lenguajes de programación más cercanos al lenguaje natural (inglés, claro) que existen. Esto hace más sencillo su aprendizaje: si se quiere hacer algo **mientras** ocurra tal cosa, se utilizará la instrucción while, por ejemplo. Todas las instrucciones terminan con alguna de cierre, como if..end if, lo que hace que el código sea mucho más sencillo de leer que utilizando, por ejemplo, la sintaxis de corchetes de C o Javascript.
* Es imposible no pensar en una treta de Microsoft para extender su lenguaje: puesto que ASP es una tecnología de Microsoft, parece lógico que la propia empresa favoreciese este lenguaje en detrimento de Javascript, probablemente utilizándolo desde el principio en los primeros ejemplos, tutoriales y libros sobre ASP.

No es en lo absoluto obligatorio utilizar VBScript para las páginas ASP. De hecho, si se está familiarizado con Javascript o se es programador de C y no de Visual Basic, es mucho mejor utilizar JavaScript. Tiene prácticamente las mismas posibilidades y su sintaxis es muy cercana a la de estos dos lenguajes.

### Utilización de VBScript

VBScript es un lenguaje de script. Estos lenguajes, que en castellano se denominan en teoría lenguajes de *secuencias de comandos* o también de *automatización*, se caracterizan por depender de una aplicación **anfitriona**: VBScript no puede crear un programa que envíe texto al usuario; depende de la aplicación anfitriona. En el caso de utilizarlo para escribir páginas ASP, dependerá de los objetos intrínsecos de ASP para recibir y enviar datos. En el caso de utilizarlo en el cliente, Internet Explorer se encargará de "entenderle" y suministrarle los medios que necesita.

#### ActiveX Scripting

La aplicación anfitriona (vamos a hablar a partir de ahora del servidor Web) utiliza **ActiveX Scripting** para ejecutar código VBScript. ActiveX Scripting (o Active Scripting) es un motor de script reutilizable que se puede usar en cualquier aplicación, en este caso en ASP (o Internet Explorer). Por defecto, viene con dos lenguajes integrados: VBScript y JScript.

ASP funciona extrayendo todo el código VBScript de la página, instanciando ActiveX Scripting y pidiéndole que ejecute el código. Además, ASP le pasa al motor de script el contexto, que no son más que referencias a los objetos ASP (Response, Request...) para que el código pueda ejecutarse: cuando pone Response.Write, tiene que escribir el texto para que sea enviado como HTML, por ejemplo. Eso no es problema de VBScript, sino de ASP.

Por defecto, el lenguaje de script de las páginas ASP es VBScript, así que si no definimos nada, ActiveX Scripting pensará que el código es VBScript e intentará ejecutarlo como tal. Si queremos que ejecute código JScript, tendremos que indicarlo en la página ASP así:

 <%@LANGUAGE = "JScript"%>

O establecerlo por defecto en la configuración del servidor Web.

####  5.2.2.2 Declarar variables

No es obligatorio declarar las variables que utilicemos en VBScript. Además, como es un lenguaje sin tipo, declararla se limita a "avisar" de que vamos a utilizar tal variable con tal nombre:

 Dim sql

Si no se declara la variable, se podrá utilizar de todas formas en el primer momento en que se haga referencia a ella. Esto, sin embargo, no es muy aconsejable, puesto que se podría escribir incorrectamente y obtener errores difíciles de encontrar.

De hecho, es muy recomendable incluir la instrucción Option Explicit al principio del script, que **obliga** a tener todas las variables que utilicemos declaradas antes de su uso, devolviendo un error en caso contrario. Así se sabrá si en lugar de **nombre** se ha puesto por error **nimbre** y por esto no funcionaba el script.

#### Constantes

Las constantes son como las variables en cuanto que se referencian mediante el nombre pero, a diferencia de éstas, su valor **nunca cambia**. VBScript además define sus constantes **intrínsecas** o constantes de VBScript.

###

### Comentarios

Los **comentarios** son fragmentos de texto que podemos incluir en el script y que no se ejecuta, de forma que sólo son visibles cuando leemos el código del script. Son muy útiles y recomendables, puesto que pueden ayudar a clarificar ciertas partes de código que no es muy clara.

Para escribir comentarios utilizamos la apóstrofe (') o comilla simple al comienzo de la línea. El resto de la línea es un comentario:

 Dim sql

 'Esto es un comentario

 **5.2.3 Lenguaje de programación Active Server Pages**

El lenguaje de programación utilizado como ya hemos dicho en muchas ocasiones es ASP (Active Server Pages) con aplicaciones VBScript y JavaScript, apoyados en lenguaje HTML. A continuación se mostrará las características generales del sistema y la programación usada a lo largo del sistema.

Los símbolos <% función %> indican que el comando que se va a ejecutar esta implementado en Lenguaje ASP el cual se muestra esta figura si observamos la pagina web sin activar.

Dentro de este comando ASP se especifica en la aplicación a utilizar, en este caso es VBScript.

 <%@LANGUAGE="VBSCRIPT"%>

# Conexión a la base de datos

Los controladores ODBC suministrados con Microsoft Access le permiten conectarse a bases de datos SQL e importar, exportar o adjuntar los datos almacenados en dichas bases de datos.

Puede utilizar la opción ODBC del Panel de control (o el Administrador de ODBC) para agregar, modificar y eliminar controladores y orígenes de datos ODBC en cualquier momento posterior a la instalación de los archivos ODBC mediante el programa Instalar.

Para iniciar la opción ODBC del Panel de control:

1. En el grupo Principal del Administrador de programas de Microsoft Windows, haga doble clic en el icono "Panel de control".
2. En la ventana Panel de control, haga doble clic en el icono "ODBC".

Cuando inicie la opción ODBC del Panel de control (o el Administrador de ODBC), puede hacer clic en el botón "Ayuda" que aparece en cada pantalla para ver Ayuda sobre la forma de introducir la información en dicha pantalla. La Ayuda para la opción ODBC del Panel de control muestra información detallada sobre la forma de utilizar este programa.

Habiendo especificado con anterioridad el Data Source Name en el Administrador ODBC, se establece la conexión del sistema a la base de datos SQL Server de la siguiente manera.

<!--#include file="Connections/Guayas.asp" -->

# Abriendo la base de datos

Se creará un objeto *recordset*, este no es mas que una forma de obtener registros de la fuente de datos, para poder identificar el origen de la información, especificando que tabla es la que vamos a utilizar. Aquí se abre la base de datos y se define la consulta que se va a realizar.

 set rs = Server.CreateObject("ADODB.Recordset")

 rs.ActiveConnection = MM\_Prueba\_STRING

 rs.Source = strsql

 rs.CursorType = 0

 rs.CursorLocation = 2

 rs.LockType = 3

 rs.Open()

 rs\_numRows = 0

 %>

# Tablas

Para implementar una estructura tabla donde se mostrarán los datos en forma alfabética se crea una tabla con varias filas, las cuales se incrementan según la cantidad de datos que se van a mostrar por cantón o parroquia respectivamente. El código para crear la tabla es el siguiente:

<table width="753" border="1" cellspacing="1" cellpadding="1" height="25">

Como se mencionó anteriormente las filas de las tablas se repiten según el numero de datos a mostrar, entonces se ejecuta un comando que implementa esta función.

 <%

 Dim Repeat1\_\_numRows

 Repeat1\_\_numRows = -1

 Dim Repeat1\_\_index

 Repeat1\_\_index = 0

 rs\_numRows = rs\_numRows + Repeat1\_\_numRows

 %>

Después de que el recordset recorre todo el registro, termina la repetición de las filas.

 <%

 While ((Repeat1\_\_numRows <> 0) AND (NOT rs.EOF))

 %>

#  5.2.3.4 Mostrando la información

Para utilizar las funciones de agregado (SQL) desde ASP a través de un objeto Recordset, tendremos que utilizar el índice para acceder al campo así: rs.Fields(0), o bien dar un **alias** a la expresión:

SELECT SUM(Total) AS STotal FROM Población

Se muestra la información en la tabla señalando el tamaño y la fuente de la letra, con el comando rs.Fields.Item.***DATOS***.value muestra el campo *DATOS* de la base de datos con las especificaciones de las consultas. Y escribir luego: rs.Fiels("STotal").

 Ejemplo:

<p align="left"><font size="1" face="Verdana, Arial, Helvetica, sansserif"> <%=(***rs.Fields.***("STotal").Value)%> </font></p>

# Consulta de información

La consulta de la base de datos se realiza mediante instrucciones SQL, este lenguaje consiste en sentencias muy próximas al inglés diseñadas para seleccionar registros de una o varias tablas de acuerdo con el criterio utilizado. Las sentencias de solicitud SQL pueden ser usadas en el momento del diseño en la propiedad recordsource de un control de datos esto permitiría crear mediante programación ASP asociados a controles de datos los criterios SQL utilizan la palabra clave SELECT, seguida de una de estas palabras claves WHERE, FROM HAVING, GROUP-BY, ORDER BY.

EL criterio SELECT selecciona los campos de la tabla, GROUP BY se utiliza para agrupar datos, ORDER BY el campo de referencia que ordena la información que se muestre en las tablas de forma ascendente y FROM es el nombre de la tabla

- SELECT **\*** FROM crecimiento ORDER BY CENSAL ASC

Devuelve todos los campos de la tabla crecimiento

- SELECT **Canton, Tothom, Totmuj** FROM nombre\_tabla

Devuelve sólo los campos canton, tothom, totmuj de la tabla..

### Cláusula WHERE

*La cláusula WHERE nos permite especificar un criterio para recuperar los datos.*

- SELECT \* FROM Población **WHERE** Canton = 'Guayaquil'

Devuelve todos los campos de todas las filas que tengan Guayaquil en el campo CANTON

- SELECT \* FROM Población WHERE Canton = 'Guayaquil' **AND** Area = Urbana

Devuelve todos los campos de la tabla de las filas en las que Canton valga Guayaquil **y** Area sea Urbana.

- SELECT Canton FROM Vivienda WHERE Nelec **<=** 100

Devuelve todas los cantones que tienen menos de 100 viviendas con servicios eléctricos.

### Funciones de agregación

- SELECT **AVG**(Total) FROM Población)

Devuelve la media (average) del campo total, es decir el promedio de la población

- SELECT Canton, Area, Sum(TOTAL) AS STOTAL, Sum(totHom) AS TOTHom, Sum(TOTMUJ) AS STOTMUJ FROM Poblacion GROUP BY Canton, Area

Devuelve la suma total de la población de la provincia del Guayas agrupados por cantón y área.

* + 1. **Java Script**

Para poder utilizar los mapas de forma que se pueda interactuar con ellos de manera dinámica procedemos colocar la imagen en el diseñador y manejador de imágenes Macromedia Fireworks. Para esta implementación se dividió en slides la imagen, en este caso el mapa político de la provincia del Guayas y de sus cantones; a cada división se la almacenará en dichos slides, con el cual es que se sectoriza los mapas.

Luego se importa la imagen desde Macromedia Dreamweaver para empezar las conexiones del mismo a las consultas y gráficos del sistema.

El script utilizado para efectuar esta operación es JavaSript y se lo codifica de la siguiente manera:

 <script language="JavaScript">

 <!—

 function fwLoadMenus() {

 if (window.fw\_menu\_0) return;

window.fw\_menu\_0 = new Menu("root",148,21,"Verdana, Arial, Helvetica, sans-serif",14,"#000000","#cccc66","#cccccc","#0033cc");

 fw\_menu\_0.hideOnMouseOut=true;

window.fw\_menu\_1 = new Menu("root",148,21,"Verdana, Arial, Helvetica,sansserif",14,"#000000","#cccc66","#cccccc","#0033cc");

 fw\_menu\_1.hideOnMouseOut=true;

window.fw\_menu\_2 = new Menu("root",148,21,"Verdana, Arial, Helvetica, sansserif",14,"#000000","#cccc66","#cccccc","#0033cc");

 fw\_menu\_2.hideOnMouseOut=true;

window.fw\_menu\_3 = new Menu("root",148,21,"Verdana, Arial, Helvetica, sans-serif",14,"#000000","#cccc66","#cccccc","#0033cc");

 fw\_menu\_3.hideOnMouseOut=true;

 fw\_menu\_3.writeMenus();

 } // fwLoadMenus()

 //-->

La función fwLoadMenus es la que controla la acción a realizar de cada slides o conjuntos de slides que pertenecen a un cantón y cada fw\_menu es un slide almacenado.

* + 1. **Applet Java**

Se utilizó un Applet Java para la creación de las pirámides poblacionales, gráficos de barras y de burbujas. Esta instrucción se la puede llamar desde ASP, especificando los campos que se desea graficar (rs(“Población”)) El proceso es el siguiente:

 ‘Se abre el Applet

<applet codebase="." code="barchart.class" archive = "BarChart.jar" width=800 height=800 align="middle">

 <% while not rs.eof

 i=i+1

 if (i mod 2 = 0) then

response.write "<param name=C" & i & " value=" & Int(1000\*(rs("TotalMujeres")/TotalMujeres)) & ">" & vbCtrl

response.write "<param name=C" & i & "\_color value=red>" & vbCtrl

response.write "<param name=C" & i & "\_style value=solid>" & vbCtrl

 else

response.write "<param name=C" & i & " value=" & Int(1000\*(rs("TotalHombres")/TotalHombres)) & ">" & vbCtrl

response.write "<param name=C" & i & "\_color value=red>" & vbCtrl

response.write "<param name=C" & i & "\_style value=solid>" & vbCtrl

 end if

 rs.movenext

 wend

 %>

 <param name=columns value=28>

 <param name=orientation value="piramide">

 <param name=scale value="3">

 </applet> ‘Se cierra el applet

* 1. **MANEJO DE LA INFORMACIÓN**

La tecnología de nuestros días pone al alcance de la humanidad capacidades hasta ahora impensadas. En los últimos años el avance en la capacidad de almacenamiento y la velocidad de acceso a la misma se vio afectada por factores de crecimiento en escala pasando de pocos Mega a cientos de mega y aún miles como el caso de los discos ópticos con cientos de Gigabytes (1024 Megabytes) y más de capacidad.

Toneladas de papel pueden ser reemplazadas en forma simple y segura por su correlato en disco, con las consiguientes ventajas en disponibilidad inmediata, tiempo de búsqueda y acceso a la información y aún seguridad física, tanto en el acceso como ante factores externos.

El proceso general del SIDEG consiste en capturar, generar, procesar, empaquetar y divulgar información sobre las características de población y vivienda de la provincia del Guayas.

Los datos se obtienen como hemos mencionado en muchas ocasiones del IV Censo de Población y III de Vivienda realizado en 1990 y de futuras actualizaciones con el censo de 2001.

Estos datos son convertidos en información mediante nuestro proceso modular, el cual requiere también del uso del conocimiento en áreas tales como geografía, demografía e informática. La información producida se brinda a los usuarios y al público en general y a la vez, realimenta el proceso de generación de más información.

SISDE es un sistema que mantiene una base de datos relacional que incluye varios registros, cada uno de los cuales corresponde a un cantón. Mediante un código único, la información asociada a cualquier cantón puede ser accedida y administrada en forma eficiente y sistemática.

SISDE incluye módulos para generación, consulta, edición y análisis de información demográfica. Es un sistema que incluye información geográfica básica e interfaces para intercambiar información con herramientas informáticas estándar tales como el WWW.

Las tecnologías de multimedia e Internet son ideales para divulgar de manera eficiente, económica y amena la información que SISDE genera en todos sus procesos.

En cada uno de ellos se presenta mediante mapas digitales interactivos, información sobre la población de cada cantón y sus características demográficas relaciones.

**5.4 REQUERIMIENTOS DE HARDWARE Y SOFTWARE**

 **5.4.1 Hardware y Software**

 Como se mencionó anteriormente el diseño es una actividad en la cual se toman decisiones importantes, frecuentemente de naturaleza estructural. Comparte con la programación un interés por la abstracción de la representación de la información y secuencias de procesamiento, pero el nivel de detalle es muy diferente en ambos casos. El diseño constituye representaciones coherentes y bien planificadas de los programas concentrándose en las interrelaciones de los componentes al mayor nivel y en la manipulación de los objetos de datos implicados en los niveles inferiores. A continuación se presenta el documento que conforma el núcleo del sistema propuesto: el documento de especificación de diseño.

La plataforma sobre la que se desarrolla este sistema es Windows XP Professional que solo es compatible con Windows 2000 Professional. Los lenguajes de programación utilizados para el desarrollo de esta tesis, se han escogido de manera que brinden la facilidad de mantenimiento y la disponibilidad de herramientas de desarrollo.

Específicamente el lenguaje de programación utilizado en la implementación de los módulos del sistema es ASP, que incluye los lenguajes de scripts: JavaScript y VBScript. Se prefirió este programa por la necesidad de la generación de páginas de contenido dinámico que tengo un código que se pueda ejecutar del lado del cliente y del lado del servidor. Por otro lado, debido a la naturaleza del sistema se utilizará también un Java Applet para la generación de los gráficos, instalado en una página HTML.

El modulo de consulta y búsqueda de información también esta implementado utilizando ASP, ya que este lenguaje permite invocar a herramientas especializadas de acceso de datos.; de esta manera poder realizar las consultas con sentencias de SQL Server, donde además esta diseñada la base de datos.

El módulo de digitalización de imágenes esta implementado en el editor y animación de imágenes Fireworks; donde se segmentan cada una de las provincias creando un acceso a la consulta de datos.

La especificación de los requisitos del software implica la culminación de la tarea del análisis de sistemas. Dicha especificación se logra estableciendo una completa descripción de las clases que colaboran, su función y el comportamiento del sistema. Este documento y el modelado que contiene deben lograr tres objetivos en mente:

* Describir lo que requiere el usuario.
* Establecer una base para la creación de un diseño de software.
* Definir un conjunto de requisitos que se puedan validar una vez que se ha construido el software.

De esta manera, se logran establecer la base para un buen diseño de sistemas, documentando una descripción del problema que el software va a resolver al definir las clases principales que componen al sistema, así como los atributos y métodos que las componen, además de las relaciones que existen entre ellas.

Sin embargo, los requerimientos de hardware recomendables pueden resumirse en los siguientes puntos:

* Pentium o equivalente
* 64 MB de memoria RAM.
* Tarjeta de video de 16-bits de color.
* Tarjeta de Video de 2 MB de memoria.
* 200 MB de Disco Duro disponible.
* Módem de 56 Kbps de velocidad, pero es recomendable una conexión T1 (cable coaxial) a 256 Kbps.
* Servicio de Internet requerido.

Para un desarrollo satisfactorio del proyecto, serán requeridos los siguientes componentes de software:

* Macromedia Fireworks 4.0.- Desarrollo de la digitalización de mapas usados a lo largo de la navegación en el sistema
* Macromedia Flash 5.0.- Creación y manejo de animaciones para el sistema.
* Macromedia Dreamweaver Ultradev 4.0.- Diseñador de paginas web con acceso a base de datos.
* Active Server Pages. Lenguaje de programación utilizado para la implementación de la aplicación.
* SQL Server DBMS. Implementación de la base de datos demográfica.

Por otro lado, para la ejecución del sistema se requiere de un navegador de Internet (Netscape Navigator/Communicator o Internet Explorer de Microsoft) con la capacidad de ejecutar Applets. Esto implica la instalación del ambiente de ejecución de Java.

 **5.4.2. Recursos especiales**

Para el desarrollo del proyecto han sido necesitados manuales y libros especializados en programación en Java, así como libros y páginas de Internet que funcionen como fundamentos teóricos referentes al problema que se desea resolver (Capítulos 2, 3 y 4 de este trabajo).