

ANÁLISIS, DISEÑO Y DESARROLLO DE UN GENERADOR DE CÓDIGO FUENTE PARA GESTIÓN DE INFORMACIÓN DE MYSQL, SQL SERVER Y ACCESS PARA LOS LENGUAJES JAVA, PHP Y ASP

1 Eduardo Chávez Reina, 2 Ing. Edgar Hermosa, 3 Ing. César Villacís.

1 Escuela Politécnica del Ejército, Ecuador, chavezedu@gmail.com

2 Escuela Politécnica del Ejército, Ecuador, echermosa@espe.edu.ec

3 Escuela Politécnica del Ejército, Ecuador, cxvillacis@espe.edu.ec

RESUMEN

En la actualidad existen varios lenguajes de programación y bases de datos para desarrollar aplicaciones, cada uno con ventajas y características que los hacen diferentes al resto de su competencia, estas características inclinan a los programadores a trabajar casi exclusivamente con un solo lenguaje y motor de base de datos, por lo que, si al programador se le solicita desarrollar aplicaciones con otro lenguaje o base de datos, le resulta complicado entender la lógica de programación, llegando incluso a abandonar proyectos por falta de conocimientos sobre las plataformas de desarrollo solicitadas. En el mercado existen ciertos programas que ayudan al desarrollador a generar código fuente, con lo que se puede ahorrar tiempo de aprendizaje y programación, pero dichos programas no son 100% compatibles con todas las bases de datos, no pueden generar código fuente para cualquier lenguaje de programación o simplemente generan demasiado código "basura".

El presente proyecto de tesis tuvo como objetivo crear un programa que pueda generar código fuente para controlar las base de datos MySQL, SQL Server y Access, usando los lenguajes de programación A.S.P., J.S.P. y P.H.P. que se encuentren previamente desarrollados en plantillas de código fuente, las cuales puedan ser intercambiables entre sí, dando al programador la posibilidad de gestionar información en pantallas amigables sin necesidad de conocer los métodos de conexión de base de datos y lenguaje de programación.

Para comprobar las capacidades del sistema, se desarrollaron plantillas para:

- *Bases de datos: MySQL, SQL Server, Microsoft Access*
- *Lenguajes de programación: A.S.P., J.S.P, P.H.P.*

Palabras Clave:

- Código fuente
- Generación de código fuente
- Gestión de base de datos
- Programación extrema

ABSTRACT

There are currently several programming languages and databases to develop applications, each with advantages and features that make them different from the rest of the competition, these features tend to programmers to work almost exclusively with a single language and database engine, so that, if the developer is

asked to develop applications with other language or database, is difficult to understand the logic of programming, even to abandon projects due to lack of knowledge about development platforms requested. In the market there are programs that help the developer to generate source code, which can save time programming and learning, but such programs are not 100% compatible with all databases engines, and can't generate source code for any language programming or just generate "junk code".

This thesis project aimed to create a program that can generate source code to control database engines MySQL, SQL Server and Access, using programming languages A.S.P., J.S.P. and P.H.P. that are previously created in source code templates, which can be interchangeable, giving the programmer the ability to manage information screens without knowing methods of database connection and programming language.

To test the system's capabilities the following templates were developed:

- *Databases: MySQL, SQL Server, Microsoft Access*
- *Programming languages: ASP, JSP, PHP*

Words Key:

- *Source code*
- *Source code generator*
- *Database gestion*
- *Extreme programming*

1. INTRODUCCIÓN

Los lenguajes de programación actuales tienen, entre sus principales diferencias, la manera como el usuario debe escribir el código fuente, esto a veces hace que los programadores se inclinen a lenguajes de programación específicos y los estandaricen para desarrollar sus aplicaciones.

Al igual que sucede con los lenguajes de programación, los programadores trabajan según su criterio con bases de datos de su preferencia.

El problema nace cuando a un programador de un cierto tipo de lenguaje y base de datos se le solicita cambiar o utilizar otro lenguaje y base de datos. En este caso la programación del sistema se complica ya que, por lo general, un programador se adapta a un cierto lenguaje y base de datos, haciendo que dicho programador estudie acerca del lenguaje en el cual deberá trabajar, quitando así tiempo y calidad al sistema a ser creado.

Actualmente los desarrolladores de programas generan librerías con las cuales se realiza la conexión a la base de datos y lectura de la información existente en cada una de las tablas, gestionar los datos y, en ciertos casos, utilizarlas para realizar la lógica de negocio que interactuará con la información. Estas librerías por lo general son redundantes en sus funciones y eventos (add, edit, delete, etc.) por lo que resulta monótono realizarlas para cada una de las tablas existentes. Los programadores además crean, para la mayoría de las tablas de la base de datos, pantallas para gestionar la información (Buscar, ver, agregar, editar, eliminar) volviendo a la monotonía de crearlas manualmente, lo que implica tiempo y costos para cualquier proyecto.

En el mercado actual, algunas bases de datos tienen programas para generar código fuente que ayude al programador a facilitar el trabajo al momento de desarrollar un sistema, desgraciadamente dichos generadores son limitados a ciertos lenguajes de programación de tipo "propietario" o simplemente generan código "basura" el cual complica su depuración o uso, como por ejemplo StarUML y sus generadores JAVA o .NET. Adicionalmente existen herramientas "case" con las cuales se puede generar código fuente para varios lenguajes de programación, pero el código generado no es óptimo, por lo que los programadores se deciden a crear los programas desde cero sin ayuda de este tipo de herramientas.

Pensando en estos problemas, se decidió diseñar un programa con el cual el programador simplemente lea una base de datos, especifique los atributos de cada una de las tablas y campos y generar código fuente que ayude con la gestión de la información de dicha base. Además, el programa debería ser compatible para varias bases de datos y varios lenguajes de programación.

1.1 Objetivos

1.1.1 Objetivo General

Analizar, diseñar y desarrollar un sistema para generar código fuente de gestión de información de distintas bases de datos con distintos lenguajes de programación, utilizando la metodología XP (Programación Extrema).

1.1.2 Objetivos Específicos

- Analizar y buscar métodos para generación de plantillas entendibles para el usuario.
- Investigar los distintos lenguajes de programación y bases de datos.
- Desarrollar un sistema con el cual se pueda generar códigos para distintos lenguajes de programación y bases de datos.
- Aplicar la metodología XP (Programación Extrema) para la creación del sistema de generación de código fuente.

2. METODOLOGÍA

Para el desarrollo del presente plan de tesis se ha optado por trabajar por la metodología XP (Programación Extrema) ya que es la que más se adapta a las necesidades que se pretende realizar. La metodología de Programación Extrema divide en pequeños prototipos funcionales al sistema, los cuales se entregarán en espacios de tiempo relativamente cortos, además, evita diagramas y calendarios poca veces reales que se utilizan en otras metodologías de desarrollo de software. Los valores de la programación extrema son:

2.1 Simplicidad

Todas las etapas del proyecto deben ser simples. Mientras el código vaya creciendo se lo deberá ir refactorizando a fin de que siempre sea entendible para todo el equipo de trabajo, ya que con la evolución del proyecto más complejo resultará su comprensión y mejoramiento, es por ello que siempre se debe ir documentando el código fuente, utilizando nombres de clases y variables entendibles para todo el equipo de trabajo.

2.2 Comunicación

La comunicación se realiza de diferentes maneras, empezando por la simplicidad del código fuente, ya que mientras más simple es el código mejor será la comprensión y comunicación entre el equipo de trabajo. El código fuente siempre debe estar auto documentado y, para las funciones que no vayan a cambiar, se debe comentar la funcionalidad de las mismas.

La comunicación con el cliente siempre debe ser fluida, de hecho, el cliente forma parte del equipo de trabajo, esto garantiza que todos los requerimientos son los correctos y él decide las características más importantes que se deban crear y siempre debe estar presente para aclarar las dudas del equipo de trabajo.

2.3 Retroalimentación

El cliente forma parte del equipo de trabajo y siempre estará al tanto de la evolución del mismo. Se deben crear ciclos cortos de entrega de partes del sistema, ya que si algo no es de satisfacción del cliente se lo puede mejo-

rar o rehacer a tiempo. Ciclos de vida más largo pueden correr el riesgo de generar módulos que deban cambiar o rehacer debido a nuevos requerimientos del sistema o por malentendidos en el momento de recopilar la información del proyecto. Las pruebas unitarias de los módulos deben ser constantes a fin de garantizar la efectividad de los mismos.

2.4 Coraje o valentía

El programador debe tener valentía para borrar código fuente en el caso que sea obsoleto o no efectivo, sin importar el tiempo que haya tomado su realización. Es más rápido rehacer un módulo desde cero que tratar de comprender uno ya existente. Además, el programador debe ser persistente cuando se encuentre alguna traba en programación, dar el tiempo necesario para que todos los módulos y clases funcionen correctamente.

2.5 Respeto

Ningún programador es mejor o peor que otro. El respeto entre el grupo de trabajo debe prevalecer y siempre debe existir colaboración para resolver los problemas o retos que implique el proyecto. Si no existe respeto entre los miembros del equipo de trabajo se corre el riesgo que los tiempos o calidad del mismo se vean afectados.

3. ANÁLISIS Y DISEÑO

3.1 Análisis

3.1.1 Situación actual

El tema de tesis desarrollado es un programa que genera código fuente para gestionar distintas bases de datos con distintos lenguajes de programación, utilizando para ello plantillas de código fuente totalmente personalizables, con una interfaz gráfica amigable para el usuario con las siguientes características:

- Los sistemas operativos en los que funciona el programa son: Windows XP/Vista/7.
- Su desarrollo fue realizado en Visual Basic.Net Express, debido a que es una herramienta de programación de libre distribución (gratis).
- La interfaz gráfica es amigable con el usuario e indica de la manera más óptima cada uno de los parámetros de las tablas y los campos que la componen.
- El sistema trabaja con las siguientes bases de datos: Microsoft Access, SQL Server y MySQL.
- El sistema genera código fuente para los lenguajes de programación: PHP, ASP y JSP.
- El sistema crea código fuente de librerías de gestión (ver, agregar, editar, eliminar) de las distintas tablas de la base de datos, usando para ello lenguaje SQL.
- El sistema genera pantallas de administración de información que se obtiene de las librerías de gestión de tablas.
- El programa es capaz de cambiar de base de datos y lenguaje de programación sin perder la configuración de atributos de tablas y campos previamente establecidos por el programador.

3.1.2 Situación actual

Dado que el presente programa es una aplicación práctica, basada en investigación y para el uso general de cualquier programador, el programa se lo denominó “CreaCod”, el cual significa “Creador de Código”. La aplicación se pretende distribuir bajo licencia GNU.

El Logo de CreaCod son dos letras “C”, la primera escrita de modo normal en color azul, mientras que la segunda tiene un efecto “espejo” de la primera y es de color rojo. Seguido de las 2 letras “C” se presenta el nombre del programa con fuente “Korataki” de color negro.



Figura 3.1. Logo de CreaCod

3.1.3 Identificación de Roles y Tareas

A) Roles

- Programador

Es el único usuario del sistema debido a que, al ser un programa sin roles de usuario, únicamente es necesario la persona quién indique al sistema los distintos atributos de base de datos, tablas, campos y lenguaje de programación del proyecto a ser creado.

B) Tareas

- Programador
 1. Indicar información general del proyecto.
 2. Seleccionar y configurar la base de datos a ser conectada.
 3. Seleccionar el lenguaje de programación en el que se creará el código fuente.
 4. Llenar los atributos de cada una de las tablas de la base de datos.
 5. Llenar los atributos de cada uno de los campos de las tablas.
 6. Seleccionar y llenar los atributos de la plantilla de administración GUI del proyecto.
 7. Generar el código fuente resultante.
 8. Cambiar idioma al IDE de CreaCod.

3.1.4 Especificación de Escenarios

Rol “Programador”

- **Iniciar proyecto:** Indicar al programa el nombre del proyecto a ser creado, el autor y la carpeta donde se almacenará la información de dicho proyecto.

- **Seleccionar base de datos y lenguaje de programación:** la base de datos de donde se obtendrá la información de tablas y campos y seleccionar el código fuente a generar
- **Conectar base de datos:** Indicar al programa “CreaCod” los parámetros de conexión a la base de datos.
- **Refrescar información de base:** Inicialmente el programa indica al usuario si desea refrescar la información de la base de datos. Al aceptar, se despliega la información de tablas y campos de la base.
- **Llenar parámetros de tablas:** Indicar al programa “CreaCod” los nombres de cada tabla, además del nombre de la clase que se creará para su gestión y el objeto que trabajará dicha clase. Indicar además las columnas visibles, de orden y de habilitación de registros.
- **Llenar parámetros de campos:** Indicar al programa “CreaCod” los nombres de cada campo que componen las tablas, indicando el nombre de función que gestionará su información, así como también el tipo de dato, la tabla relacional y el objeto de control (en ciertos casos) y el valor predeterminado del campo.
- **Seleccionar plantilla de gestión GUI:** Indicar al programa “CreaCod” si la tabla tendrá una plantilla de gestión GUI y, de seleccionarse, indicar para cada uno de los campos el tipo de control (Textbox, Combobox, etc.) con el que se presentará cada uno de los campos.
- **Generar proyecto:** Usando un botón, se generará el código fuente en la carpeta de proyecto, indicando los errores encontrados en caso de existir.
- **Refrescar base de datos (caso 2):** En el caso de que la base de datos haya cambiado su estructura, tener la posibilidad de refrescar la información de tablas y campos a fin de actualizar el proyecto a sus nuevas características.
- **Cambiar idioma al IDE de CreaCod:** El programa debe ser capaz de cambiar el idioma de la interfaz gráfica a fin de que programadores de otros países e idiomas lo puedan utilizar sin problemas.

4. CONSTRUCCIÓN Y PRUEBAS

4.1 Construcción

4.1.1 Historias de Usuario

- **Lectura de base de datos:** La estructura de las bases de datos debe ser leída por el sistema.
- **Almacenamiento de información de estructura de base:** Una vez leída la información de estructura de base de datos, debe almacenarse en memoria y en algún tipo de archivo.
- **Entorno GUI:** Se debe tener una manera de poder presentar la información al usuario, a fin que el sistema sea entendible al usuario.
- **Establecer valores de tablas y columns:** Se debe agregar información para Tablas y Columns, de acuerdo a la información que se necesite en las posibles plantillas que se va a generar.
- **Estructura de plantillas:** debe especificar los parámetros que deberán tener las plantillas a fin de poder ser leídas y cambiadas por el sistema.
- **Modelo de generación de código fuente:** Una vez establecidos todos los valores de plantillas y datos de tablas y columns, se deberá estructurar el método de generación del código fuente.

- **Plantillas para otros lenguajes de programación y bases:** Se deberán crear las plantillas para el resto de bases de datos y lenguajes de programación.
- **Módulos de idioma:** El sistema debe ser capaz de cambiar de idioma.
- **Ayudas:** Terminado el sistema, se deberá documentar todas las ayudas.

4.1.2 Iteraciones

➤ **Iteración 1:**

Se investigará la manera cómo se puede acceder a la información de tablas y columnas de las bases de datos, ya que esta información es necesaria a fin de obtener la estructura de la misma para luego generar el código fuente.

Se investigará además el método para mantener en memoria la información recopilada de la base de datos y guardar dicha información en algún tipo de archivo, esta parte irá cambiando a medida que se vayan agregando más parámetros de tablas y columnas.

➤ **Iteración 2:**

Se irá diseñando el entorno de trabajo a medida que se vaya especificando los distintos parámetros que se pueda considerar para tablas, columnas y plantillas.

➤ **Iteración 3:**

Se creará el modelo para crear las plantillas de:

- Bases de datos
- Clases de control de bases de datos
- Gestión de información

Estas plantillas serán leídas por el sistema a fin de obtener los distintos valores que un lenguaje de programación o base de datos pueda necesitar.

➤ **Iteración 4:**

Se creará todas las funciones necesarias a fin de que el sistema pueda procesar la información contenida en las plantillas y cambiarla con la información de base de datos, tablas y columnas.

En esta iteración el sistema deberá ser capaz de crear el código fuente esperado.

➤ **Iteración 5:**

Se crearán el resto de plantillas para otras bases de datos y lenguajes de programación.

El sistema deberá ser capaz de cambiar el idioma para que sea accesible a la mayoría de programadores.

➤ **Iteración 6:**

Una vez finalizado el sistema, se procederá a crear las ayudas necesarias.

4.2 Pruebas

4.2.1 Base de datos “persona”

Para las pruebas de CreaCod se ha creado una base de datos con todos los campos, tipos de datos y controles necesarios para demostrar la funcionalidad del sistema. La base de datos tiene una sola relación, los campos primarios son numéricos autoincrementables como lo especifica los requerimientos de CreaCod.

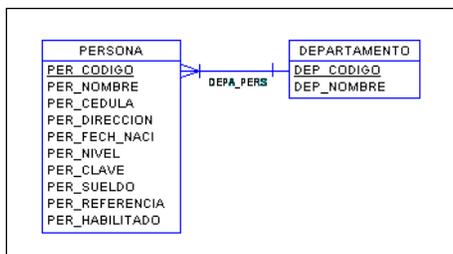


Figura 4.1. Modelo lógico de base de datos “persona”

4.2.2 Prerrequisitos de pruebas

Para las pruebas del programa “CreaCod” fue necesario levantar servidores para las distintas bases de datos que se utilizarán con el sistema, al igual que servidores para que la computadora “entienda” los lenguajes de programación utilizados. En el siguiente cuadro se especifica los servidores levantados en la computadora de pruebas.

Tabla 4.1. Servidores

Tipo	Nombre	Servidor
Base de datos	Microsoft Access	Librerías de control Access
Base de datos	SQL Server	SQL Server Express 2010
Base de datos	MySQL	MySQL Server 5.5.27 / WAMP 2.2
Servidor Web	I.I.S.	I.I.S. 6.0 (Windows XP)
Servidor Web	Apache	Apache web Server 2.2.1 / WAMP 2.2
Servidor Web	Apache Tomcat	Apache Tomcat 6
Lenguaje de programación	A.S.P.	Incluido en I.I.S. 6.0
Lenguaje de programación	J.S.P.	Incluido en Apache Tomcat 6
Lenguaje de programación	P.H.P.	P.H.P. 5.3.10 / WAMP 2.2

4.2.3 Pruebas de bases de datos vs. Lenguajes de programación

Las pruebas de CreaCod se basaron en las plantillas generadas para las siguientes bases de datos:

- Microsoft Access 2003.
- Microsoft SQL Server Express.
- MySQL 5.5.27.

Las pruebas de CreaCod se basan en las plantillas generadas para los siguientes lenguajes de programación:

- A.S.P.
- J.S.P.
- P.H.P.

Cada una de las bases de datos fue probada con cada uno de los lenguajes de programación antes mencionados.

4.2.4 Resultados de pruebas

El programa CreaCod fue desarrollado en Visual Basic Express Edition, por lo que su ambiente de trabajo es únicamente Windows XP/Vista/7.

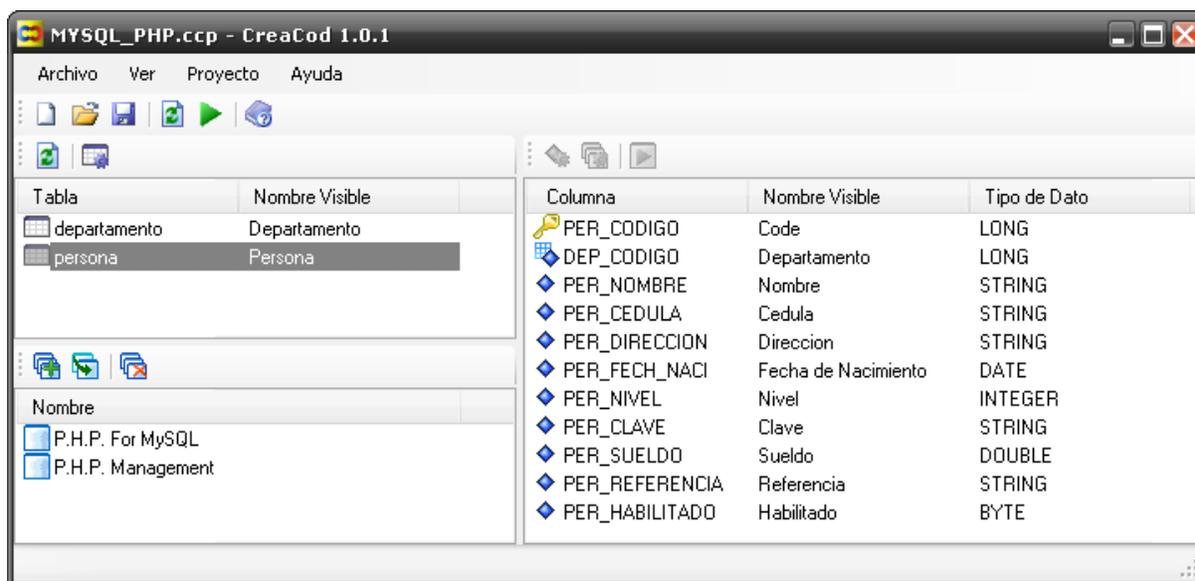


Figura 4.2. CreaCod 1.0.1

4.2.4.1 Ejemplo de resultados

Para el siguiente ejemplo de resultados se ha tomado la base de datos MySQL junto al lenguaje de programación PHP.

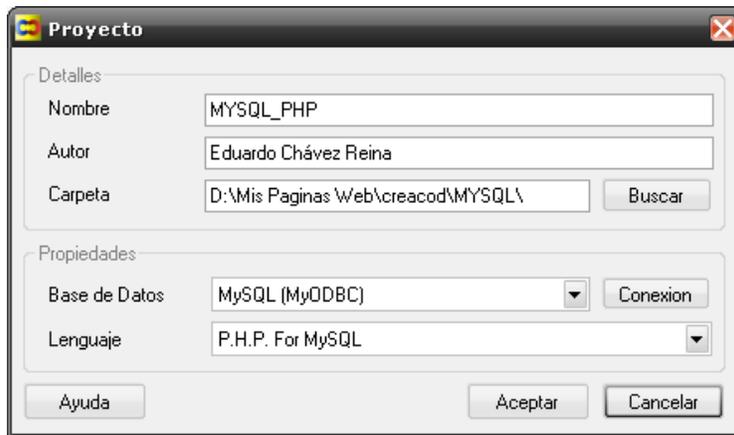


Figura 4.3. Propiedades de proyecto MySQL / PHP

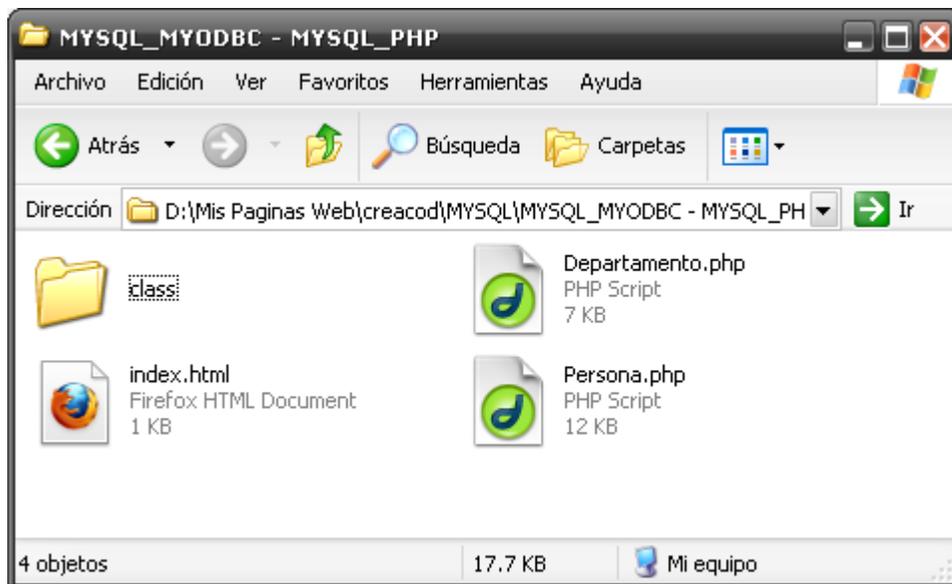


Figura 4.4. Archivos generados MySQL / PHP

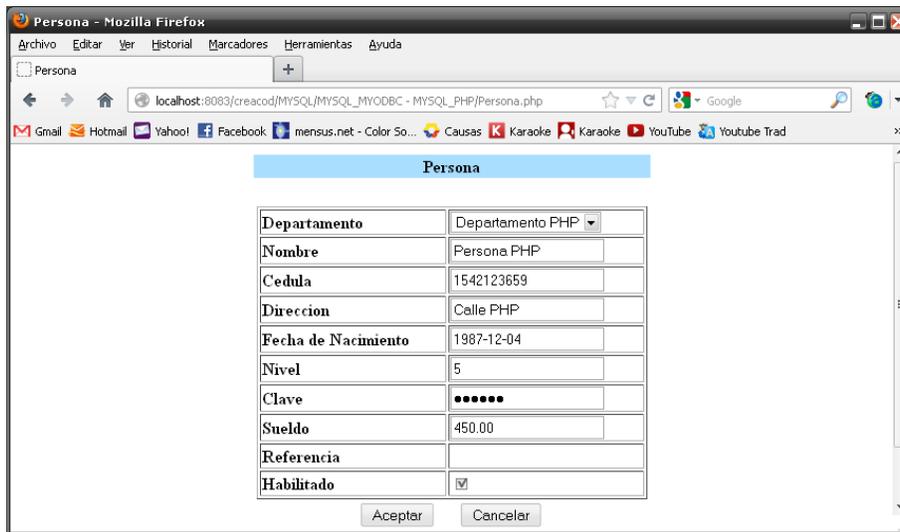


Figura 4.5. Agregar registro MySQL / PHP

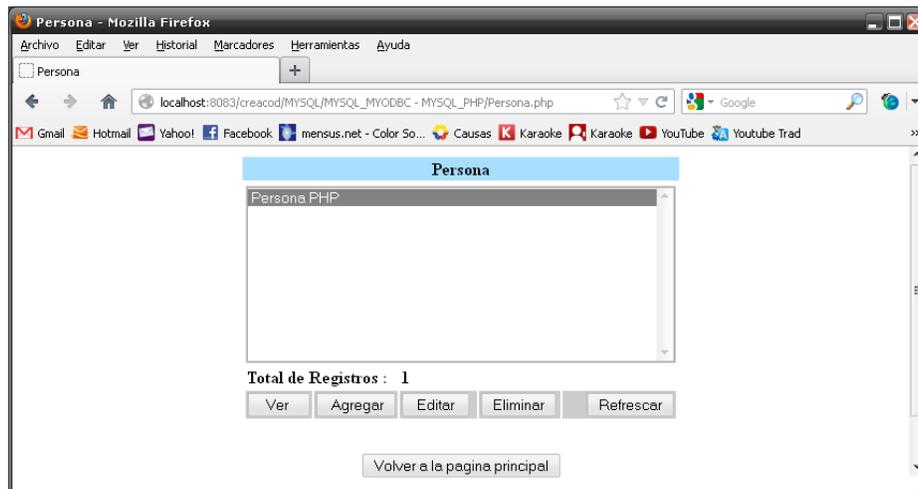


Figura 4.6. Ver registros MySQL – PHP

4.2.5 Casos de uso reales de CreaCod

CreaCod ha sido utilizado para realizar proyectos de alto nivel, donde la comunicación de base de datos es primordial y los tiempos de ejecución deben ser óptimos.

A continuación se citan algunos proyectos en donde las librerías y entornos GUI generados por CreaCod fueron utilizados.

- RockOla, sistema multimedia.

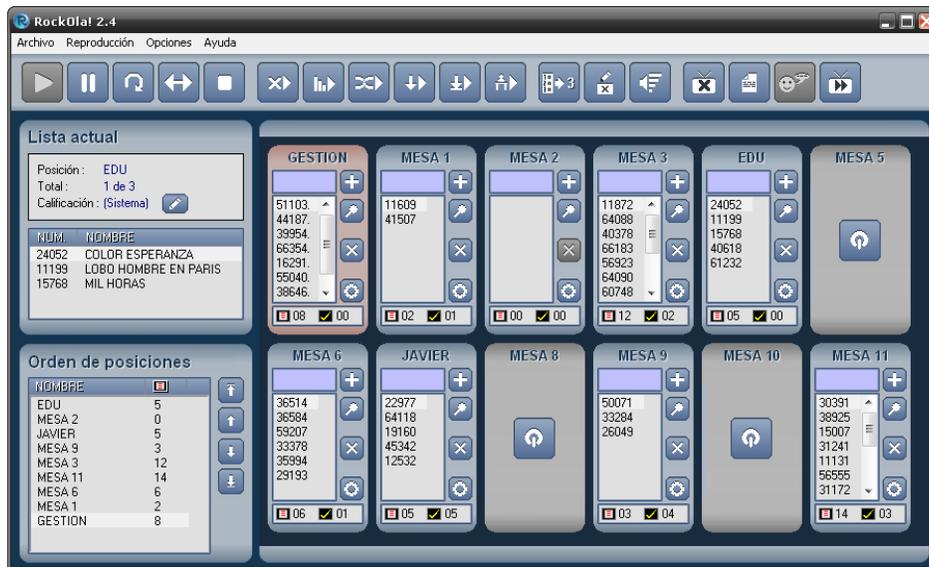


Figura 4.7. RockOla 2.4

- rSNAP, sitio Web de áreas protegidas del Ecuador¹.

Reserva Ecológica "Antisana"

Antisana es un estratovolcán del arco de los Andes. Se sitúa en Ecuador, entre las provincias de Napo y Pichincha. Situado entre los volcanes de Cotopaxi, Chacana, Cayambe, Reventador y Sumaco. Perdió al menos 36% de su masa original en los últimos 50 años. Por su importancia estratégica, es la más estudiado de estas cumbres andinas. Se mide su longitud anualmente y su masa cada mes. Es una de las fuentes de agua potable de Quito, capital de Ecuador.

Flora

Características	
Tipo	Reserva Ecológica
Creación	Resolución No.18 de Julio 21 de 1993
Extensión	120 000 ha
Ubicación	Se encuentra localizada en la vertiente oriental de la Cordillera de los Andes, a 50 Km al sureste de Quito, en la provincia de Napo.
Altitud	5758 metros
Precipitación	500 - 1500 m.m
Provincias	Napo

Figura 4.8. Ver rSNAP

- SisCoop, gestor de cuentas y préstamos.

¹ rSNAP, tema de tesis será desarrollado por Patricia Trujillo Montalvo, previo a la obtención del título de Ingeniera De Sistemas e Informática de la Universidad de las FF. AA. (ESPE)

ADMINISTRADOR, SUPER ADMINISTRADOR

Inicio Movimientos Gestión Administración Reportes Ayuda

Cuenta

Buscar:

Número	Persona	Saldo	Act
01042	CANDO CAMISAN CINTHYA ISABEL (0202270542)	0.00	SI
01038	CANDO CAMISAN PAOLA ELIZABETH (0201821634)	90.00	SI
01039	CHAVEZ REINA EDUARDO RENE (1715035133)	45.00	SI
01041	SANCHEZ CAMISAN DAYANNA VALERIA (1104684806)	0.00	SI
01040	TRUJILLO MONTALVO PATRICIA DE LAS MERCEDES (1718756834)	0.00	SI
01043	VALLEJO BRAVO LUIS MIGUEL (1712848454)	0.00	SI
Total de registros : 6			

Imprimir Nuevo

Tareas comunes

- Mostrar como formulario externo
- Imprimir resultados
- Nuevo registro
- Gestión de Cuenta
- Ir a "Inicio"
- Cerrar sesión

Figura 4.9. SisCoop

5. CONCLUSIONES Y RECOMENDACIONES

➤ **Conclusión**

El programa CreaCod evita a los programadores varias horas laborables tratando de conseguir información o capacitándose en un lenguaje de programación o una base de datos específica.

➤ **Recomendación**

Es muy útil que un software cree las funciones de conexión y gestión de una base de datos para que el programador se dedique directamente a desarrollar la lógica de negocio sin necesidad de pasar el tiempo en investigación y aprendizaje.

➤ **Conclusión**

La programación de clases de control de bases de datos es similar para todas las tablas que la componen, por lo que muchas veces el programador "copia y pega" varias funciones y métodos repetitivos a fin de ahorrar tiempo y esfuerzo. Desgraciadamente muchas veces resulta más complicado adecuar código fuente entre una clase y otra.

➤ **Recomendación**

Es recomendado que el programador tenga ayuda con la generación de código fuente al momento de desarrollar un sistema computacional, ya que se puede ahorrar tiempo de desarrollo además que puede aprender nuevos e innovadores métodos de programación.

➤ **Conclusión**

Programar para un lenguaje resulta monótono debido a que hay demasiado código fuente que es similar entre clases, librerías o formularios.

➤ **Recomendación**

Auto-generar código puede ahorrar a un proyecto informático varias horas de trabajo, recurso humano y costos económicos.

➤ **Conclusión**

Cuando se desarrolla código fuente, el programador se acostumbra a un determinado tipo de lenguaje. Si al programador se le solicita un software en otro lenguaje le resulta complicado investigar o re-aprender sobre dicho lenguaje.

➤ **Recomendación**

Ayudar, mediante la generación de código, a que el programador recuerde o aprenda nuevas técnicas de programación es beneficioso para todo desarrollo de software y para la capacitación del profesional.

➤ **Conclusión**

El uso de herramientas CASE para generar código fuente no siempre se adecua a las necesidades reales del sistema y crean demasiado código “basura”.

➤ **Recomendación**

Poder realizar plantillas de código fuente personalizadas es beneficioso para el programador debido a que únicamente se genera código fuente entendible para el propósito de dicha plantilla y a la medida de las necesidades del sistema.

➤ **Conclusión**

CreaCod tiene varias ventajas respecto a otros generadores de código, sin embargo las plantillas y/o el programa pueden quedar obsoletos debido al continuo cambio en los lenguajes de programación.

➤ **Recomendación**

Es necesario evolucionar constantemente a CreaCod y sus plantillas, a fin de que se mantengan a la par con las nuevas tecnologías y métodos de programación.

➤ **Conclusión**

La gestión de una base de datos es más eficaz si se crean clases y objetos que la controlen directamente, con esto se obtiene un código fuente más limpio y no se mezcla código fuente con “SQL”.

➤ **Recomendación**

Cuando se desarrolle código fuente, es recomendado agrupar cada una de las partes de la lógica de negocio y bases de datos en librerías, clases y funciones específicas para cada módulo, ya que se logra mejores resultados al momento de depurar o modificar el código fuente.

6. REFERENCIAS

Web:

Extreme Programming (XP) – An Overview

http://www.c-sharpcorner.com/UploadFile/sakthivel_a/ExtremeProgramming11282005054653AM/ExtremeProgramming.aspx

Metodología XP Vs. Metodología Rup

<http://metodologiapvsmetodologiarup.blogspot.com/>

UML - Diagramas de estados

<http://jms32.eresmas.net/tacticos/UML/UML08/UML0801.html>

Ejemplos de diagramas UML

<http://www.monografias.com/trabajos67/diagramas-uml/diagramas-uml2.shtml>

UML: Caso de uso

http://es.wikipedia.org/wiki/Caso_de_uso

UML: diagrama de caso de uso

http://es.wikipedia.org/wiki/Diagrama_de_casos_de_uso

.NET

http://es.wikipedia.org/wiki/Microsoft_.NET

Microsoft Visual Basic Express Edition

http://es.wikipedia.org/wiki/Visual_Basic_.NET#Visual_Basic_Express_Edition

SQL

<http://es.wikipedia.org/wiki/SQL>

MySQL

<http://es.wikipedia.org/wiki/MySQL>

Open Database Connectivity

http://es.wikipedia.org/wiki/Open_Database_Connectivity

Manual de Microsoft Access 2003

<http://www.duiops.net/manuales/access/access16.htm>

Microsoft Access

http://es.wikipedia.org/wiki/Microsoft_Access

Características de Microsoft Access

<http://estudiantodufuturo.blogspot.es/1251222395/>

Microsoft SQL Server

http://es.wikipedia.org/wiki/Microsoft_SQL_Server

SQL Server (Microsoft)

<http://www.microsoft.com/latam/technet/productos/servers/sql/2005/overview.msp>

PHP

<http://es.wikipedia.org/wiki/PHP>

Características del PHP

<http://foros.monografias.com/showthread.php/60249-Caracteristicas-del-PHP..>

Manual de PHP

<http://www.php.net/manual/es/index.php>

Active Server Pages

http://es.wikipedia.org/wiki/Active_Server_Pages

Introducción a ASP

http://gpsl.dlsi.ua.es/~slujan/asp/Introduccion_a_ASP.htm

Historia de ASP

<http://www.maestrosdelweb.com/editorial/historiaasp/>

Internet Information Services

<http://es.wikipedia.org/wiki/IIS>

JAVA Server pages

http://es.wikipedia.org/wiki/JavaServer_Pages

Características de JSP

<http://es.scribd.com/doc/27519905/35/Caracteristicas-de-JSP>

Configuration / Integrating Tomcat an Apache Web server

<http://www.slideshare.net/topsealrockers/integrating-apache-tomcat-using-modjk-for-windows-xp>

Instalación y Configuración de Apache Tomcat 6.0 en Windows XP

<http://jcalderon.wordpress.com/2008/01/04/instalacion-y-configuracion-de-apache-tomcat-60-en-windows-xp/>