UNIVERSIDAD DE LAS FF. AA. "ESPE"

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

"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"

Previa a la obtención del Título de:

INGENIERO EN SISTEMAS E INFORMÁTICA

POR: EDUARDO RENÉ CHÁVEZ REINA

SANGOLQUÍ, 14 de agosto del 2012

DECLARACIÓN

Yo, Eduardo René Chávez Reina, declaro que el presente trabajo es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación personal; y, que he consultado las referencias bibliográficas que se incluyen en el documento.

La Universidad de las FF.AA. (ESPE) puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normativa institucional vigente.

Sangolquí, 14 de agosto del 2012

Eduardo René Chávez Reina

CERTIFICACIÓN

Certificamos que el presente trabajo fue realizado en su totalidad por el Sr. EDUARDO RENÉ CHÁVEZ REINA como requerimiento parcial a la obtención del título de INGENIERO EN SISTEMAS E INFORMÁTICA.

Sangolquí, 14 de Agosto del 2012

Ing. Edgar Hermosa	Ing. Cesar Villacís
DIRECTOR	CO-DIRECTOR

DEDICATORIA

El presente trabajo está dedicado a mis Padres, quienes me han sabido apoyar en todos los caminos de enseñanza y aprendizaje. Para ellos mis agradecimientos y mi eterna gratitud.

A mis hermanas, Patricia y Karina, quienes siempre han estado a mi lado y me han enseñado que toda meta en la vida debe ser cumplida.

Para mis tías queridas: Aidita y Olguita, que en la tierra siempre esperaron ver esta meta cumplida y ahora, desde el cielo, me mandaron las fuerzas necesarias para culminarla.

Eduardo Chávez Reina

DEDICATORIA

Para mi hijito Javier, quién aun siendo tan pequeño me ha sabido dar todas las fuerzas, ganas y motivos necesarios para lograr cumplir esta meta.

“Todo el trabajo y esfuerzo te lo dedico a ti, porque eres la razón de mi vida y el motivo de mi felicidad”.

Te quiero mucho!.

Tu Papito

AGRADECIMIENTOS

Doy gracias a DIOS por todas las bendiciones que me ha dado y por la fuerza para cumplir esta meta de vida.

A mi Madre, quién ha sabido esperar para verme cumplir este sueño, quién ha estado a mi lado en los momentos buenos y malos de mi vida y de quién siempre he recibido fuerza y amor para lograr cumplir mus sueños. *Gracias Maicita.*

A mi Padre, quién ha sabido darme la fuerza para lograr cumplir mis objetivos, quién me ha enseñado muchas de las cosas que ahora son parte de mi vida y siempre estará a mi lado para apoyarme. *Gracias Paicito*.

A mis directores de tesis, quiénes con su sabiduría y experiencia me guiaron a culminar el presente trabajo.

A mi amiga Patty Trujillo, con quién hicimos en conjunto los temas de tesis, gracias por todo el tiempo dedicado a cumplir nuestros sueños.

A mis amigos: Omar, Galo y Eddie, quiénes me apoyaron moralmente para lograr cumplir esta meta.

Eduardo Chávez Reina

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

ÍNDICE DE CÓDIGOS FUENTE

RESUMEN

El presente proyecto de tesis propone realizar un generador de código fuente para varios lenguajes de programación, el cual controlará y gestionará distintas bases de datos, con el fin de ayudar al programador a realizar programas de una manera ágil y eficiente.

Para realizar la tesis se ha optado por la metodología de desarrollo "Extremme Programming" (XP o Programación Extrema, en español) propuesta por Kent Beck, la cual es una alternativa ideal para desarrollo de software de pequeña y mediana complejidad, ya que omite diagramas y calendarios, pocas veces reales, que se deben realizar con otras metodologías de desarrollo de software.

La programación extrema se basa en pequeños prototipos funcionales, los cuales se van siguiendo a partir de historias de usuarios analizadas al inicio del proyecto y son entregados en períodos de tiempo relativamente cortos.

Esta metodología es ideal para el desarrollo del presente tema ya que ayuda al programador a dividir en partes cada uno de los módulos que compondrán el

sistema y las plantillas que se deban generar. Investigar uno a uno los lenguajes de programación y bases de datos propuestos y, para cada prueba, tener un producto totalmente terminado.

CAPÍTULO 1

GENERALIDADES

1.1. TÍTULO

“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”

1.2. 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.

Existen ciertos programas y lenguajes de programación que ayudan al programador a generar código fuente, con lo que se puede ahorrar tiempo de programación, pero desgraciadamente dichos programas no son 100% compatibles con todas las bases de datos o generación de código en un lenguaje específico, o simplemente generan demasiado código (Ej. Dreamweaver).

1.3. PLANTEAMIENTO DEL PROBLEMA

Una vez estudiadas y analizadas las distintas alternativas que tiene el programador como ayuda para generar código fuente, se ha llegado a la conclusión que:

- La mayoría de generadores de códigos de programación están diseñados para trabajar en un único lenguaje o lenguajes de su respectiva marca (Ej. .NET).

- Los generadores actuales crean demasiado código, siendo prácticamente imposible su comprensión y depuración. (Ej. Dreamweaver).
- No existe un generador de código para múltiples lenguajes de programación.
- No existe un generador de códigos redundantes, como es el caso de la codificación para gestión de información de bases de datos (Agregar, editar, modificar y eliminar).
- No existe un programa con el cual el usuario pueda generar sus propias plantillas[1] para ayudar a generar códigos.
- Los generadores existentes no crean completamente todo el código que el usuario necesita para la gestión de información de bases de datos, ni tampoco permiten al usuario manipular las plantillas de generación para adecuarlos a sus necesidades.
- No existen en el mercado generadores de bases de datos orientadas a objetos que permitan al usuario acceder de manera sencilla a su código para acoplarlo a sus necesidades.

1.4. JUSTIFICACIÓN

Un sistema que pueda generar código para distintas bases de datos y distintas plataformas ayudaría a los programadores a:

[1] Plantillas: (En software) Conjunto de instrucciones generalmente almacenadas en un archivo de texto que deben ser interpretados línea a línea en tiempo real para su ejecución.

- Generar todo tipo de código de gestión de información, sin importar la base de datos o el lenguaje de programación.
- Escribir sus propias plantillas o modificar las existentes a fin de adaptarlos a las necesidades del programador.
- Ahorrar tiempo de programación.
- Generar código entendible y de fácil depuración.
- Dada la naturaleza del sistema a ser realizado, puede servir además para generar plantillas que no necesariamente sean para aplicaciones de bases de datos o lenguajes de programación.

1.5. OBJETIVOS

1.5.1. OBJETIVO GENERAL

Analizar, diseñar y desarrollar un sistema para generar código fuente de gestión de información multibase y multiplataforma.

1.5.2. 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.

1.6. ALCANCE

La tesis tiene como objetivo principal el crear un sistema para generar código fuente de distintos tipos de lenguajes de programación y bases de datos, siendo sus principales características:

- Generar código fuente de gestión de información de base de datos entendible al usuario.
- Poder intercambiar fácilmente el lenguaje de programación o base de datos de un mismo proyecto.
- Permitir al usuario crear o manipular las plantillas preestablecidas a fin de modificarlos a su conveniencia.

El sistema contará con plantillas de demostración para:

Tabla 2.1. Tabla de plantillas a ser creadas

Bases de datos	**Lenguajes de Programación**
MySQL	PHP
SQL Server	Java (Javabeans)
Microsoft Access	A.S.P.

1.7. 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.

Programación Extrema determina una serie de características descritas a continuación:

- **Desarrollo iterativo e incremental**.- Se van creando pequeños programas y mejoras a cada una de las entregas del sistema en pequeños intervalos de tiempo.
- **Pruebas unitarias continuas**.- Cada extracto funcional de código es comprobado, usualmente con motores automatizados, para verificar su funcionalidad.
- **Programación en parejas**.- 2 personas realizan un mismo trabajo, con ello se obtiene mejores ideas al momento de desarrollar programas, además de tener la seguridad que siempre se tendrá a alguien que sepa cómo funciona el código desarrollado.

- **Integración directa con el cliente**.- El cliente colabora directamente con el desarrollo de la aplicación, de hecho, es parte del equipo de desarrollo del proyecto, así el cliente sabrá con exactitud el estado de su sistema y comprenderá cada una de las partes que se van desarrollando, pudiendo dar ideas y mejoras a fin de obtener un producto de su entera satisfacción.
- **Corrección de errores antes de entregas**.- Cada uno de los módulos a ser entregados deben tener la seguridad que son funcionales, ya que XP recicla mucho código fuente para el resto de módulos que se vayan desarrollando.
- **Refactorización de código**.- Si un módulo o extracto de código no es funcional, es mejor repetirlo hasta que cumpla con su objetivo, si el módulo ya es utilizado en otras partes del sistema se debe tener la certeza que el nuevo código será funcional.
- **Código compartido**.- Ninguna persona que forma parte del equipo de desarrollo es dueña de su código fuente, éste debe ser compartido a todos los miembros del proyecto, de hecho, se recomienda ir rotando al personal para que siempre esté al tanto de los cambios y mejoras que se vayan realizando al sistema.
- **Simplicidad de código**.- Es mejor ir realizando partes funcionales de código fuente que sean pequeñas, para después irlas mejorando o evolucionando, ya que si se hace código grande se corre el riesgo que nunca se use, retardando los tiempos y recursos del proyecto. Mientras

más simple es el código mejor será la comunicación entre todos los miembros del equipo de desarrollo.

La programación extrema cuenta con artefactos para ser realizada con éxito, dichos artefactos son tarjetas o “cards” que se irán llenando de acuerdo a la etapa en la que se encuentre el proyecto.

- **Historias de Usuario**.- Representan las necesidades del cliente las cuales deben ser escritas de manera natural (no técnica). Cada caraterísitica principal del sistema será una historia de usuario, las cuales se irán expandiendo con historias de usuario más pequeñas a lo largo del proyecto a fin de obtener todos los requisitos del sistema. Estás tarjetas ayudan a tener una estimación de tiempo para el desarrollo del sistema, evitan un gran documento de requisitos y omiten la creación de las pruebas de aceptación.
- **Tarjetas de aceptación**.- Una vez terminada una tarea, se debe llenar una tarjeta indicando la funcionalidad de dicha tarea.
- **Tarea de ingeniería (Task Cards)**.- Son las tareas que se asignan a cada grupo de trabajo del proyecto. Las tareas no deben extensas y deben realizar lo que se requiera en la tarea asignada.
- **Tarjetas CRC (Clase – Responsabilidad – Colaborador)**.- Tarjetas que indican la información de cada clase (objeto, formulario, evento, etc.) del sistema, las responsabilidades (métodos, eventos, funciones, etc.) y sus colaboradores (personal del grupo de trabajo del proyecto).

Al ser el presente tema de tesis un tema de investigación, cada una de las bases de datos y plantillas de código fuente serán una “historia de usuario”, cuando las plantillas se hayan realizado se procederá a la generación de código fuente y se probará su funcionalidad, con lo cual se irán cerrando cada una de las historias de usuario que XP propone. Tanto el código fuente, clases y módulos serán reutilizados para el resto de plantillas.

1.8. HERRAMIENTAS

Para el desarrollo del presente plan de tesis se ha utilizado las siguientes herramientas:

1.8.1. HARDWARE

- Intel Core 2 Duo 2.0 Ghz.
- 2 Gb Memoria RAM
- 500 Gb Disco duro
- Monitor 15.6” Widescreen

1.8.2. SOFTWARE

- Microsoft Windows XP/7, versión Professional
- Microsoft Visual Basic 2005/2010 Express Edition
- Microsoft C# 2005/2010 Express Edition

- Microsoft Office 2010
- MySQL 5.2
- MySQL-Front 3.1
- Power Designer 6 Data Architect
- Microsoft SQL Server Express
- Apache web Server 2.2.21
- PHP 5.3.10
- Apache Tomcat 6
- Macromedia Dreamweaver 8
- StarUML 5.0.2
- JCreator Pro 2.2
- ISTool / Inno Setup 4.2.7
- Jasc Saint Shop Pro 8
- Mozilla Firefox 14
- HTML Help Workshop 4.74

1.9. FACTIBILIDAD

1.9.1 TÉCNICA

Para el desarrollo del sistema se cuenta con fuentes de información seguras para la obtención de requerimientos, bibliografía acorde al tema y la experiencia del autor en desarrollo de sistemas de similares características.

1.9.2 OPERATIVA

Para el desarrollo del presente tema de tesis existe el apoyo suficiente por parte de personas o empresas que desean probar los beneficios que el generador de código fuente puede ofrecer.

El sistema propuesto ayudará a los programadores a desarrollar código fuente de alta calidad y adecuado a las necesidades de los proyectos.

CAPÍTULO 2

MARCO TEÓRICO

2.1. EXTREME PROGRAMMING (XP)

La Programación Extrema (Extreme Programming o “XP” por sus siglas en inglés) es una de las nuevas metodologías de desarrollo de software propuesta por Kent Beck, la cual forma parte de las “metodologías ágiles de desarrollo”.

2.1.1. BASES DE LA PROGRAMACIÓN EXTREMA

“En la programación extrema se da por supuesto que es imposible prever todo antes de empezar a codificar. Es imposible capturar todos los requisitos del

sistema, saber qué es todo lo que tiene que hacer ni hacer un diseño correcto al principio. Es bastante normal hacer un diseño, ponerse a codificar, ver que hay faltantes o errores en el diseño, empezar a codificar fuera del diseño y al final el código y el diseño, o no se parecen, o hemos echado un montón de tiempo en cambiar la documentación de diseño para que se parezca al código".[2]

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.

2.1.2. VALORES

Los valores de la Programación Extrema son:

2.1.2.1. 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

[2] "Programación extrema". Fuente: http://www.chuidiang.com/ood/metodologia/extrema.php

documentando el código fuente, utilizando nombres de clases y variables entendibles para todo el equipo de trabajo.

2.1.2.2. 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.

2.1.2.3. 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 mejorar 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.

2.1.2.4. 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.

2.1.2.5. 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.

2.1.3. FUNCIONAMIENTO

El funcionamiento de la Programación Extrema se basa en iteraciones pequeñas y entregas relativamente cortas. La reusabilidad de código y la continua

planificación hacen de esta metodología un método rápido de programación incremental.

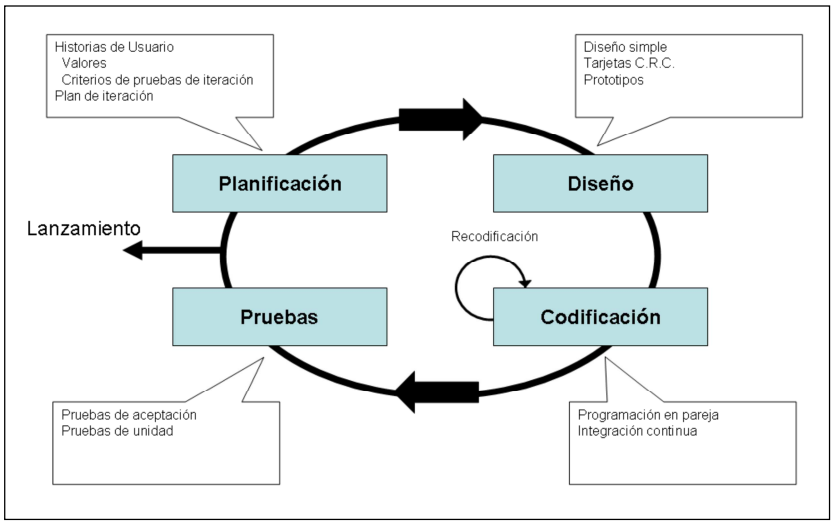


Figura 3.1. XP: Funcionamiento

2.1.4. CARACTERÍSITICAS

Programación Extrema determina una serie de características descritas a continuación:

- **Desarrollo iterativo e incremental**.- Se van creando pequeños programas y mejoras a cada una de las entregas del sistema en pequeños intervalos de tiempo.
- **Pruebas unitarias continuas**.- Cada extracto funcional de código es comprobado, usualmente con motores automatizados, para verificar su funcionalidad.
- **Programación en parejas**.- 2 personas realizan un mismo trabajo, con ello se obtiene mejores ideas al momento de desarrollar programas,

además de tener la seguridad que siempre se tendrá a alguien que sepa cómo funciona el código desarrollado.

- **Integración directa con el cliente**.- El cliente colabora directamente con el desarrollo de la aplicación, de hecho, es parte del equipo de desarrollo del proyecto, así el cliente sabrá con exactitud el estado de su sistema y comprenderá cada una de las partes que se van desarrollando, pudiendo dar ideas y mejoras a fin de obtener un producto de su entera satisfacción.
- **Corrección de errores antes de entregas**.- Cada uno de los módulos a ser entregados deben tener la seguridad que son funcionales, ya que XP recicla mucho código fuente para el resto de módulos que se vayan desarrollando.
- **Refactorización de código**.- Reescribir el código fuente a fin de que siempre sea de simple comprensión sin modificar su comportamiento. Todas las funciones que sea refactorizadas serán probadas a fin de comprobar su funcionalidad.
- **Código compartido**.- Ninguna persona que forma parte del equipo de desarrollo es dueña de su código fuente, éste debe ser compartido a todos los miembros del proyecto, de hecho, se recomienda ir rotando al personal para que siempre esté al tanto de los cambios y mejoras que se vayan realizando al sistema.
- **Simplicidad de código**.- Es mejor ir realizando partes funcionales de código fuente que sean pequeñas, para después irlas mejorando o evolucionando, ya que si se hace código grande se corre el riesgo que

nunca se use, retardando los tiempos y recursos del proyecto. Mientras más simple es el código mejor será la comunicación entre todos los miembros del equipo de desarrollo.

2.1.5. ARTEFACTOS

La programación extrema cuenta con artefactos para ser realizada con éxito, dichos artefactos son tarjetas o “cards” que se irán llenando de acuerdo a la etapa en la que se encuentre el proyecto.

2.1.5.1. HISTORIAS DE USUARIO

Representan las necesidades del cliente las cuales deben ser escritas de manera natural (no técnica). Cada caraterísitica principal del sistema será una historia de usuario, las cuales se irán expandiendo con historias de usuario más pequeñas a lo largo del proyecto a fin de obtener todos los requisitos del sistema. Estás tarjetas ayudan a tener una estimación de tiempo para el desarrollo del sistema, evitan un gran documento de requisitos y omiten la creación de las pruebas de aceptación.

Historia de Usuario	
Número:	**Usuario :**
Nombre :	
Prioridad :	**Riesgo en Desarrollo:**
Puntos Estimados :	**Iteración asignada :**
Programador responsable :	

Descripción:
Observaciones:

Figura 3.2. XP: Historia de Usuario

2.1.5.2. TARJETAS DE ACEPTACIÓN

Una vez terminada una tarea, se debe llenar una tarjeta indicando la funcionalidad de dicha tarea.

2.1.5.3. TAREA DE INGENIERÍA (TASK CARDS / TAREA)

Son las tareas que se asignan a cada grupo de trabajo del proyecto. Las tareas no deben extensas y deben realizar lo que se requiera en la tarea asignada.

	Tarea
Número de Tarea :	**Número de Historia :**
Nombre de Tarea :	
Tipo de Tarea:	**Puntos Estimados :**
Fecha de Inicio :	**Fecha Fin :**
Programador responsable :	
Descripción:	

Figura 3.3. XP: Tarea de Ingeniería

2.1.5.4. TARJETAS CRC (CLASE – RESPONSABILIDAD – COLABORADOR)

Tarjetas que indican la información de cada clase (objeto, formulario, evento, etc.) del sistema, las responsabilidades (métodos, eventos, funciones, etc.) y sus colaboradores (personal del grupo de trabajo del proyecto).

{Titulo}	
{Responsabilidades}	{Colaboradores}

Figura 3.4. XP: Tarjeta C.R.C.

2.1.6. COMPARACIÓN XP / RUP

Tabla 3.1. XP: Comparación XP / RUP[3]

XP	RUP
Características	
Desarrollo iterativo e incremental	Desarrollo interno en etapas interactivas
Pruebas unitarias continuas, frecuentemente repetidas y automatizadas	Esta integrado en todo el ciclo de vida
Programación por parejas	Programación por equipos
Interacción con el usuario final	Interacción con el usuario estratégico
Refactorización de código	
Propiedad de código	
Simplicidad de código	
Roles	
Programador	Analistas
Encargado de pruebas	Desarrolladores
Cliente	Gestores
Encargado de seguimiento	Especialistas

[3] "Metodología XP vs. RUP". Fuente: http://metodologiaxpvsmetodologiarup.blogspot.com/

Entrenador	Stakeholders
Consultor	Revisor
Gestor	Coordinador de revisiones
	Revisor técnico
Selección de metodología	
Los requisitos cambian (clientes indecisos)	Comunicación entre equipos
Proyectos con alto grado de riesgos	Complejidad de desarrollo de acuerdo al tamaño del proyecto
Grupos pequeños de programadores (Entre 2 y 12)	Configuración y control de cambios (Artefactos)
Ventajas	
Comunicación	Mayor documentación
Simplicidad de código	Verificar la calidad de software
Realimentación	Configuración y control de cambios
Coraje (Satisfacción de los programadores)	Es modelado, guiado por casos de uso
Disminuye traza de errores	Es centrado en arquitectura, guiado por riesgos
Alta calidad - Mínimo tiempo	
Desventajas	
Dificultad para determinar el costo del proyecto	Los cambios son en una fase
Se usa principalmente en proyectos pequeños	Proyectos grandes

2.2. UML

"UML es un lenguaje para especificar, construir, visualizar y documentar los artefactos de un sistema de software orientado a objetos (OO). Un artefacto es una información que es utilizada o producida mediante un proceso de desarrollo de software. UML se quiere convertir en un lenguaje estándar con el que sea posible modelar todos los componentes del proceso de desarrollo de aplicaciones. Sin embargo, hay que tener en cuenta un aspecto importante del modelo: no pretende definir un modelo estándar de desarrollo, sino únicamente un lenguaje de modelado. Otros métodos de modelaje como OMT (Object Modeling Technique) o Booch sí definen procesos concretos. En UML los procesos de desarrollo son diferentes según los distintos dominios de trabajo; no

puede ser el mismo el proceso para crear una aplicación en tiempo real, que el proceso de desarrollo de una aplicación orientada a gestión, por poner un ejemplo."[4]

Las diferencias son muy marcadas y afectan a todas las fases del proceso. El método del UML recomienda utilizar los procesos que otras metodologías tienen definidos.

2.2.1. ARTEFACTOS PARA EL DESARROLLO DE PROYECTOS

UML se especifica en forma de diagramas, que junto con la documentación, constituyen los artefactos de ésta metodología.

2.2.1.1. DIAGRAMAS DE CASOS DE USO

"Un caso de uso es una descripción de los pasos o las actividades que deberán realizarse para llevar a cabo algún proceso. Los personajes o entidades que participarán en un caso de uso se denominan actores. En el contexto de ingeniería del software, un caso de uso es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema."[5]

---

[4] "Lenguaje para modelamiento unificado". Fuente: *http://www.monografias.com/trabajos82/lenguaje-uml-importancia-modelar/lenguaje-uml-importancia-modelar2.shtml*

[5] "Caso de Uso". Fuente: http://es.wikipedia.org/wiki/Caso_de_uso

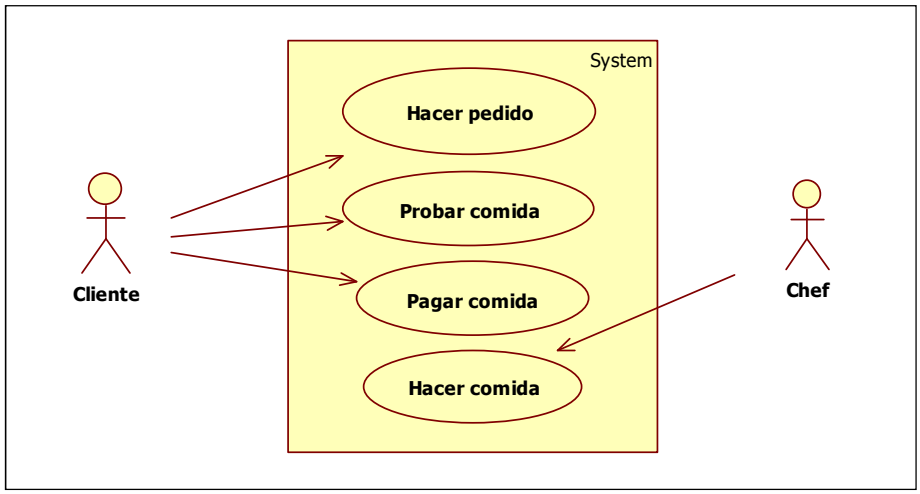


Figura 3.5. UML: Diagrama de “Caso de uso”

2.2.1.2. DIAGRAMAS DE SECUENCIA

“El diagrama de secuencia es un tipo de diagrama usado para modelar interacción entre objetos en un sistema según UML, el cual muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada caso de uso. Mientras que el diagrama de casos de uso permite el modelado de una vista business del escenario, el diagrama de secuencia contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario, y mensajes intercambiados entre los objetos.”[6]

[6] “Diagrama de Secuencia”. Fuente: http://es.wikipedia.org/wiki/Diagrama_de_secuencia

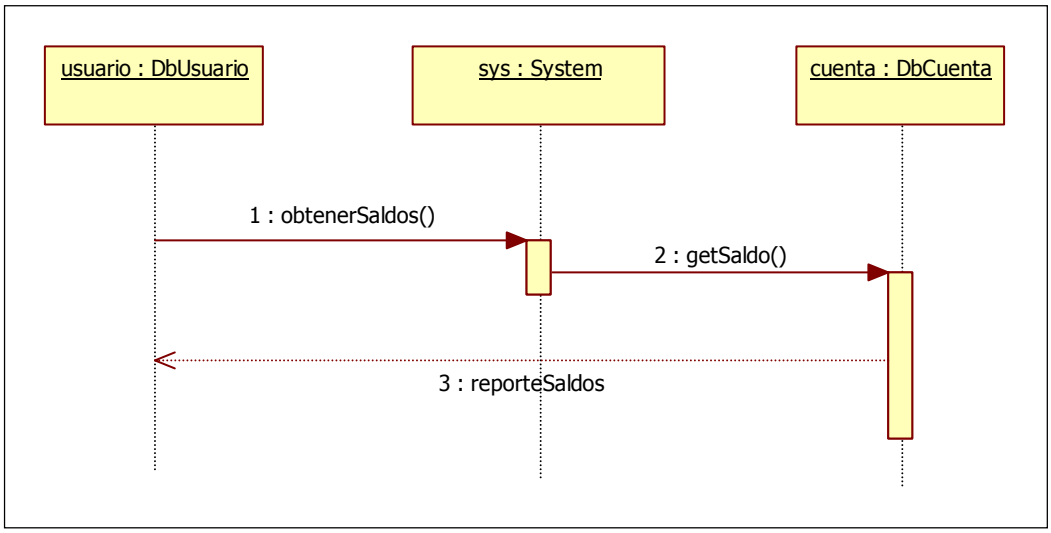


Figura 3.6. UML: Diagrama de “Secuencia”

2.3. INTERFAZ DE USUARIO

La interfaz gráfica, o también llamada GUI (Graphical User Interface, por sus siglas en inglés) es un conjunto de formas, textos, figuras y objetos que pretenden visualizar las distintas acciones, métodos e información de un sistema.

En los inicios de la computación, la interacción con el usuario se basaba en líneas de comandos de texto, las cuales enviaban y recibían información mediante el teclado. Hoy en día se pueden encontrar todavía equipos con esta característica como son: Routers, maquinaria y equipos industriales, centrales telefónicas, etc.

La interfaz de usuario está estrechamente ligada a una pantalla donde se muestra la información.

Hoy en día la gran mayoría de sistemas operativos cuentan con una interfaz gráfica para presentar al usuario las distintas características que lo componen. Incluso, algunos dispositivos o “gadgets” cuentan con interfaz gráfica para mejorar la interacción con el usuario, entre ellos: Teléfonos celulares, impresoras, reproductores MP3/MP4, etc.

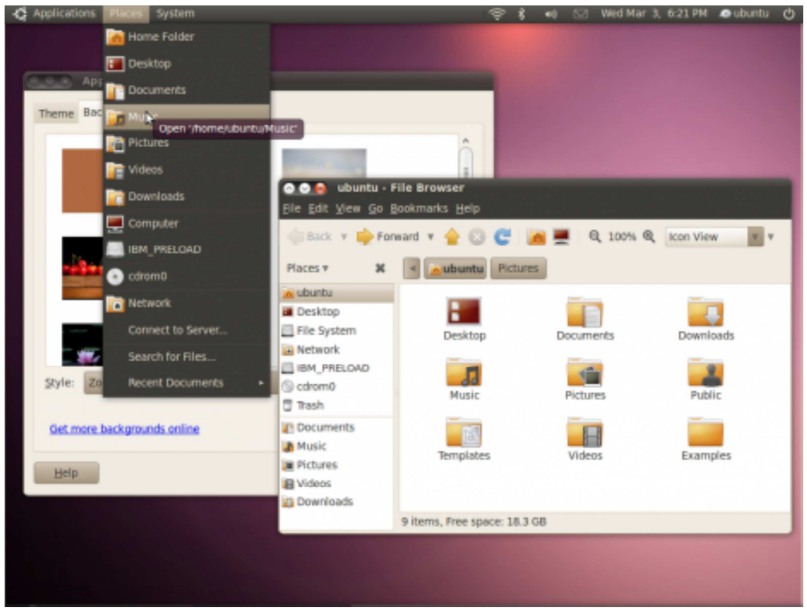


Figura 3.7. Interfaz gráfica del sistema operativo “Ubuntu”[7]

[7] Captura de pantalla del escritorio del sistema operativo “Ubuntu”.

2.4. MICROSOFT .NET

".NET es un framework de Microsoft que hace un énfasis en la transparencia de redes, con independencia de plataforma de hardware y que permita un rápido desarrollo de aplicaciones. Basado en ella, la empresa intenta desarrollar una estrategia horizontal que integre todos sus productos, desde el sistema operativo hasta las herramientas de mercado. .NET podría considerarse una respuesta de Microsoft al creciente mercado de los negocios en entornos Web, como competencia a la plataforma Java de Oracle Corporation y a los diversos framework de desarrollo web basados en PHP. Su propuesta es ofrecer una manera rápida y económica, a la vez que segura y robusta, de desarrollar aplicaciones –o como la misma plataforma las denomina, soluciones– permitiendo una integración más rápida y ágil entre empresas y un acceso más simple y universal a todo tipo de información desde cualquier tipo de dispositivo."[8]

2.4.1. COMMON LANGUAJE RUNTIME

"El Common Language Runtime o CLR ("entorno en tiempo de ejecución de lenguaje común") es un entorno de ejecución para los códigos de los programas que corren sobre la plataforma Microsoft .NET. El CLR es el encargado de compilar una forma de código intermedio llamada Common Intermediate

[8] Microsoft .NET. Fuente: http://es.wikipedia.org/wiki/Microsoft_.NET

Language (CIL, anteriormente conocido como MSIL, por Microsoft Intermediate Language), al código de maquina nativo, mediante un compilador en tiempo de ejecución."[9]

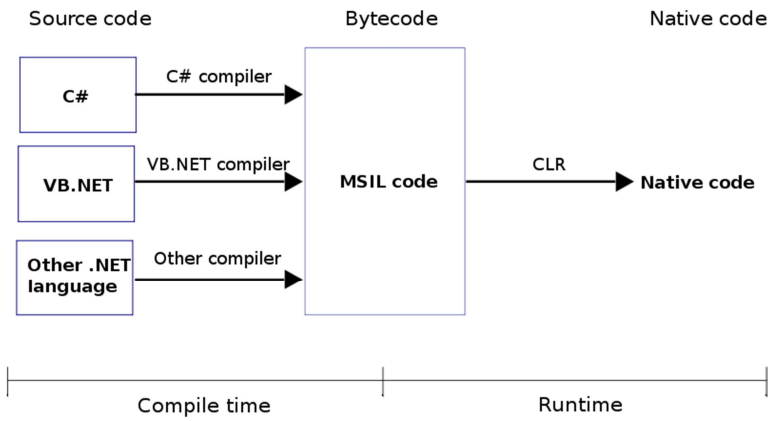


Figura 3.8. .NET Common Language Runtime[10]

2.4.2. MICROSOFT VISUAL STUDIO

"Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, y Visual Basic .NET, al igual que entornos de desarrollo web como ASP.NET, aunque actualmente se han desarrollado las extensiones necesarias para muchos otros. Visual Studio permite a los desarrolladores crear aplicaciones, sitios y aplicaciones web, así

[9] Common Lenguaje Runtime. Fuente: http://es.wikipedia.org/wiki/Common_Language_Runtime
[10] "C.L.R.". Fuente: http://commons.wikimedia.org/wiki/File:Common_Language_Runtime_diagram.svg

como servicios web en cualquier entorno que soporte la plataforma .NET (a partir de la versión .NET 2002). Así se pueden crear aplicaciones que se intercomuniquen entre estaciones de trabajo, páginas web y dispositivos móviles."[11]

2.4.3. VISUAL STUDIO EXPRESS

A partir de la versión de Visual Studio 2005 se decidió crear ediciones básicas de los entornos de programación de Microsoft las cuales se distribuyen de manera gratuita, dichas ediciones pretenden llegar al mercado educacional y a pequeños proyectos, aunque las posibilidades de éstas herramientas las hacen ideales para desarrollo de proyectos sin las complicaciones de licencias que tienen las herramientas de pago de Microsoft.

Entre las herramientas que Microsoft entrega como versiones "Express" se encuentran:

- Visual Basic Express Edition
- Visual C# Express Edition
- Visual C++ Express Edition
- Visual J# Express Edition (Desapareció en Visual Studio 2008)
- Visual Web Developer Express Edition (para programar en ASP.NET)
- Visual F# (Apareció en Visual Studio 2010, es parecido al J#)*

[11] "Microsoft Visual Studio". Fuente: http://es.wikipedia.org/wiki/Microsoft_Visual_Studio

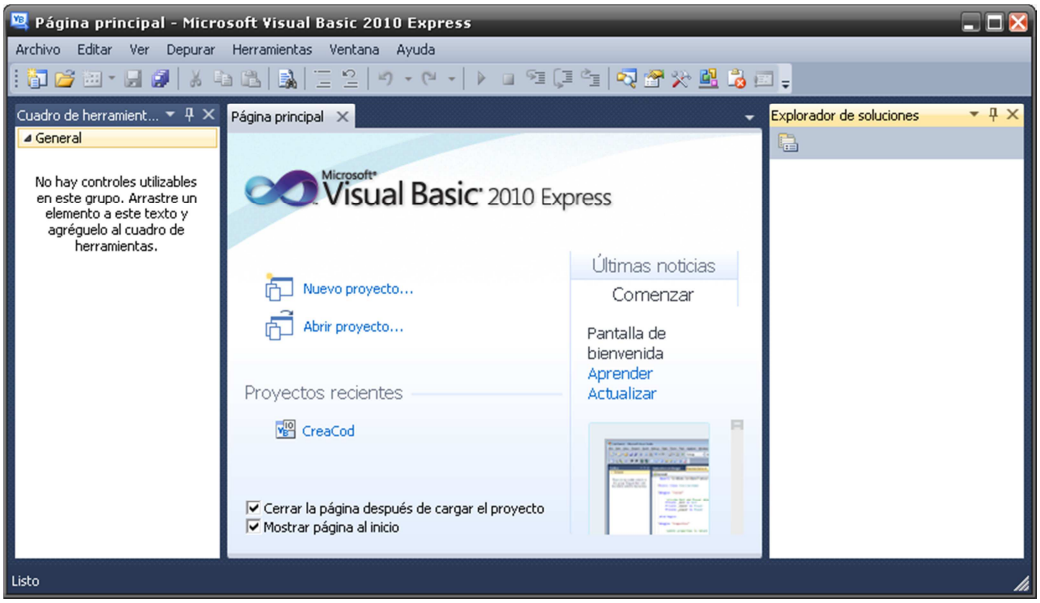


Figura 3.9. .NET: Interfaz de Visual Basic 2010 Express[12]

2.5. BASE DE DATOS

"Una base de datos o banco de datos (en ocasiones abreviada con la sigla BD o con la abreviatura "DB") es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta. Actualmente, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato

[12] Captura de pantalla del programa "Microsoft Visual Basic 2010 Express".

digital (electrónico), que ofrece un amplio rango de soluciones al problema de almacenar datos.”[13]

Existen programas denominados “sistemas gestores de bases de datos”, abreviado SGBD, que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Las propiedades de estos SGBD, así como su utilización y administración, se estudian dentro del ámbito de la informática.

2.5.1. BASES DE DATOS RELACIONALES

“El modelo relacional para la gestión de una base de datos es un modelo de datos basado en la lógica de predicados y en la teoría de conjuntos. Es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. Tras ser postuladas sus bases en 1970 por Edgar Frank Codd, de los laboratorios IBM en San José (California), no tardó en consolidarse como un nuevo paradigma en los modelos de base de datos. Su idea fundamental es el uso de «relaciones». Estas relaciones podrían considerarse en forma lógica como conjuntos de datos llamados «tuplas». Pese a que ésta es la teoría de las bases de datos relacionales creadas por Edgar Frank Codd, la mayoría de las veces se conceptualiza de una manera más fácil de imaginar, esto es, pensando en cada relación como si fuese una tabla que

[13] “Base de Datos”. Fuente: http://es.wikipedia.org/wiki/Base_de_datos

está compuesta por registros (cada fila de la tabla sería un registro o tupla), y columnas (también llamadas campos).”[14]

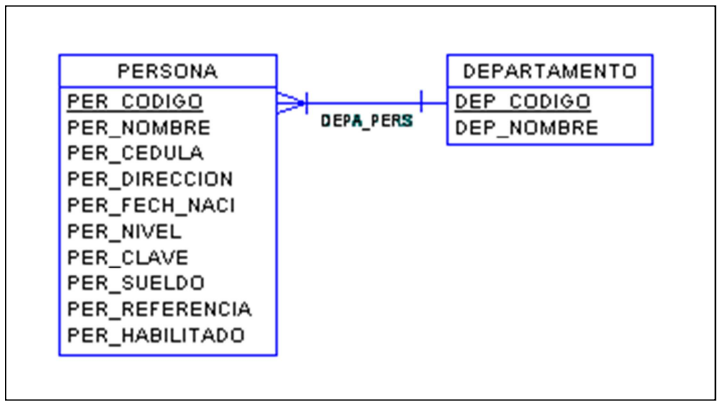


Figura 3.10. Ejemplo de base de datos relacional

2.5.2.CONCEPTOS DE BASES DE DATOS RELACIONALES

Una base de datos relacional maneja varios conceptos descritos a continuación:

2.5.2.1. ENTIDADES

Las entidades son las clases u objetos los cuales poseen características únicas que las diferencian del resto. Cada entidad debe ser considerada de acuerdo a las necesidades de la base de datos y debe poseer características suficientes para diferenciarla del resto de entidades. Por ejemplo, en una empresa se pueden encontrar personas, departamentos, clientes, etc.

[14] “Modelo relacional”. Fuente: http://es.wikipedia.org/wiki/Modelo_relacional

Las entidades son graficadas en rectángulos y se representan en forma de tablas.

Figura 3.11. Base de datos: Entidad

2.5.2.2. ATRIBUTOS

Los atributos o campos son el conjunto datos que describen de manera única y completa a cada entidad. Por ejemplo, una persona tiene cedula, nombre, dirección, teléfono, etc.

Cada atributo corresponde a una columna o campo de una tabla correspondiente a una entidad.

2.5.2.3. REGISTROS

Un registro es la representación de un objeto de una entidad. Ejemplo: Registros de una entidad *"persona"*, con atributos *"Nombre", "Cédula" y "Teléfono"* se tiene el registro *"Jorge","1718451213","2487-465"*.

En una base de datos, los registros corresponden a las filas de la tabla de entidad.

ALU_CODIGO	ALU_NOMBRE	ALU_CEDULA	ALU_FECHA	ALU_HABILITADO
1	Javier Chavez	1715035133	2006-08-14	1
2	Patty Trujillo	1123541245	2012-07-19	1
33	Paola Cando	1754213541	2012-07-24	1
36	Caro Yumbai	1718756834	2012-07-01	1
58	Carolina	0201911781	2012-07-12	1
59	Georgete Simona	1717496531	2000-01-01	1
60	Benjamin Ortiz	1700299751	2000-01-19	1

Figura 3.12. Base de datos: Ejemplo de registros

2.6. S.Q.L.

Structured Query Languaje (o SQL, por sus siglas en inglés) es un lenguaje declarativo para acceder y gestionar la información de una base de datos relacional. Una de las características de SQL es el manejo de algebra y cálculo relacional, con lo que se logra, de una manera sencilla, acceder y recuperar la información de la base de datos, además de realizar cambios sobre la misma.

Para la edición y modificación de datos se usa el “lenguaje de definición de datos” (o DDL, por sus siglas en inglés), el cual tiene 4 operaciones básicas:

- **CREATE**.- Crea un objeto en la base de datos.

Código fuente 3.1. SQL: Sentencia “Create”

```
create table DEPARTAMENTO
```

```
(
    DEP_CODIGO      bigint(20) NOT NULL auto_increment,
    DEP_NOMBRE      VARCHAR(100)                    ,
    primary key (DEP_CODIGO)
);
```

- **ALTER**.- Altera la estructura de un objeto existente.

Código fuente 3.2. SQL: Sentencia “Alter”

```
ALTER TABLE 'DEPARTAMENTO' ADD DEP_NUMERO INT UNSIGNED;
```

- **DROP**.- Elimina un objeto de la base de datos.

Código fuente 3.3. SQL: Sentencia “Drop”

```
DROP TABLE 'DEPARTAMENTO';
```

- **TRUNCATE**.- Vacía un objeto existente.

Código fuente 3.4. SQL: Sentencia “Truncate”

```
TRUNCATE TABLE 'DEPARTAMENTO';
```

El lenguaje de manipulación de datos (o DML, por sus siglas en inglés) permite recuperar información de la base de datos, además de manipularla y realizar consultas.

- **SELECT**.- Permite buscar y mostrar uno o varios registros de una o varias tablas.

Código fuente 3.5. SQL: Sentencia “Select”

```
SELECT * FROM 'DEPARTAMENTO' WHERE DEP_CODIGO=1;
```

- **INSERT**.- Permite agregar un nuevo registro a la tabla.

Código fuente 3.6. SQL: Sentencia “Insert”

```
INSERT INTO 'DEPARTAMENTO' VALUES(1, ‘Primer Piso);
```

- **UPDATE**.- Actualiza uno o varios registros existentes en la tabla.

Código fuente 3.7. SQL: Sentencia “Update”

```
UPDATE 'DEPARTAMENTO' SET DEP_NUMERO=1 WHERE DEP_CODIGO=1;
```

- **DELETE**.- Borra uno o más registros de la tabla.

Código fuente 3.8. SQL: Sentencia “Delete”

```
DELETE FROM 'DEPARTAMENTO' WHERE DEP_CODIGO=1;
```

2.7. ODBC

“Open DataBase Connectivity (ODBC, por sus siglas en inglés) es un estándar de acceso a bases de datos desarrollado por SQL Access Group en 1992. Su objetivo es enviar y recibir información necesaria para tener acceso, desde programas, a bases de datos que admitan el protocolo ODBC. Un origen de datos está formado por la procedencia de los datos y la información de conexión necesaria para tener acceso a los mismos. Ejemplos de orígenes de datos son Microsoft Access, Microsoft SQL Server, Oracle RDBMS, una hoja de cálculo y un archivo de texto. Ejemplos de información de conexión son la ubicación del servidor, el nombre de la base de datos, el Id. de inicio de sesión, la contraseña y diversas opciones de controlador ODBC que describen cómo conectarse al origen de datos.”[15]

[15] “Open Database Connectivity”. Fuente: http://es.wikipedia.org/wiki/Open_Database_Connectivity

Figura 3.13. ODBC: Orígenes de datos de Microsoft Windows

2.7.1. COMPONENTES

En la arquitectura ODBC, una aplicación se conecta con la interfaz ODBC, quién a su vez se conecta con el Gestor de Controlador. Éste a su vez utiliza un controlador o driver ODBC, quien se conecta a una base de datos.

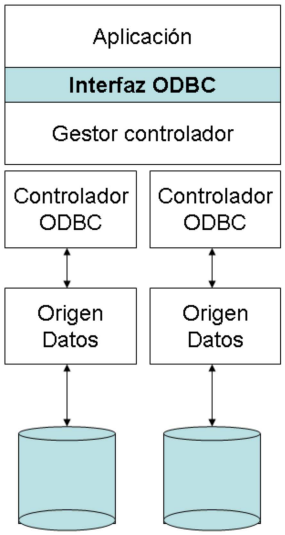


Figura 3.14. ODBC: Componentes

2.7.2. CADENAS DE CONEXIÓN

ODBC, para conectarse a una base de datos, utiliza una línea de comando estructurada de acuerdo a la base a la que vaya a conectarse. Estas líneas son conocidas como “Cadenas de Conexión” y tienen información relevante para conectar una base, como son: Servidor, base de datos, usuario, password, etc. Cada una de estas líneas se conecta a un controlador ODBC específico.

Ejemplos de cadenas de conexión:

Código fuente 3.9. ODBC: Cadena de conexión para “Access”

```
DBQ=D:\persona.mdb;Driver={Microsoft Access Driver
(*.mdb)};DriverId=281;FIL=MS
Access;MaxBufferSize=2048;MaxScanRows=8;PageTimeout=5;SafeTra
nsactions=0;Threads=3;UID=;UserCommitSync=Yes
```

Código fuente 3.10. ODBC: Cadena de conexión para “SQL Server”

```
DRIVER=SQL
Server;SERVER=.\SQLEXPRESS;UID=usuario;PWD=;APP=Visual
Basic;DATABASE=persona;Network=DBMSSOCN
```

Código fuente 3.11. ODBC: Cadena de conexión para “MySQL”

```
DRIVER=MySQL ODBC 3.51
Driver;DESC=;DATABASE=persona;SERVER=localhost;UID=usuario;PA
SSWORD=;PORT=3306;OPTION=;STMT=;
```

2.8. MYSQL

“MySQL es la base de datos “open source” más popular y, posiblemente, mejor del mundo. Su continuo desarrollo y su creciente popularidad están haciendo de MySQL un competidor cada vez más directo de gigantes en la materia de las bases de datos como Oracle, SQL Server, Sybase, etc. MySQL es un sistema de administración de bases de datos (Database Management System, DBMS) para bases de datos relacionales. Así, MySQL no es más que una aplicación que permite gestionar archivos llamados de bases de datos. MySQL fue escrito en C y C++ y destaca por su gran adaptación a diferentes entornos de desarrollo, permitiendo su ínter actuación con los lenguajes de programación más utilizados como PHP, Perl y Java y su integración en distintos sistemas operativos.”[16]

[16] “MySQL”. Fuente: http://www.esepestudio.com/articulo/desarrollo-web/bases-de-datos-mysql/Que-es-MySQL.htm

Figura 3.15. MySQL: Logo[17]

2.8.1. MYSQL – FRONT

MySQL-Front es una sencilla pero útil aplicación para trabajar bases de datos que trabajan con MySQL.

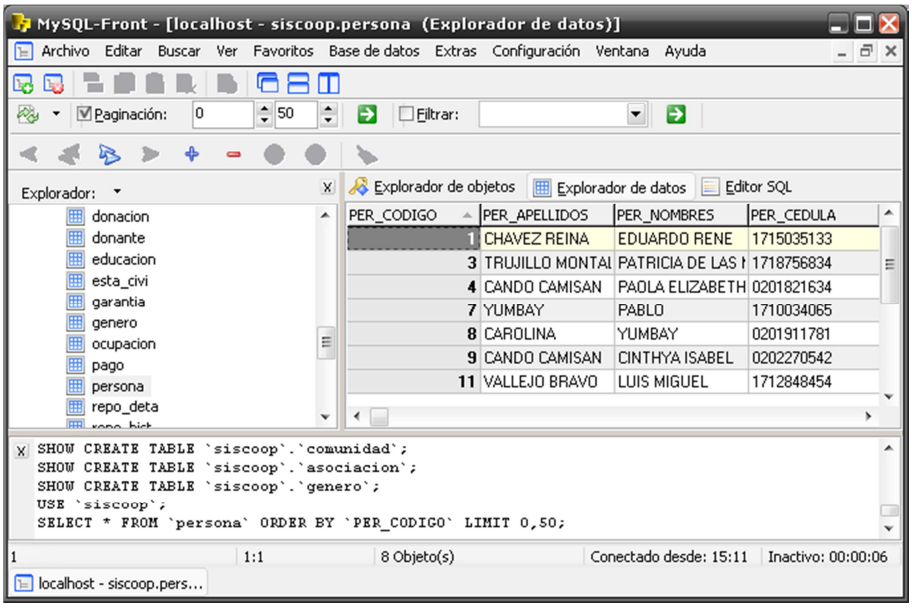


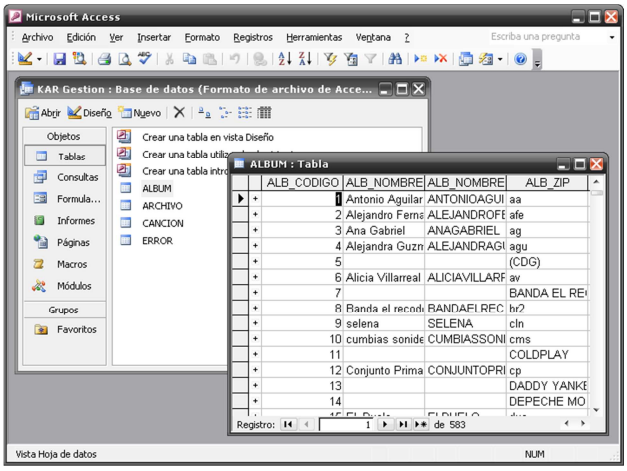
Figura 3.16. MySQL-Front[18]

[17] "MySQL Logo". Fuente: http://www.mysql.com/

Desde el primer momento en el que se empieza a utilizar este administrador se descubre su facilidad para obtener información sobre las bases de datos, tanto de sus tablas como de su estructura y contenido, todo ello desde un interfaz muy intuitivo que recuerda bastante a la estructura del Explorador de Windows.

2.9. MICROSOFT ACCESS

Microsoft Access es un sistema de administración de base de datos para Windows. Access es compatible con SQL y su característica principal es que sus archivos de base de datos son portables (archivo .mdb). Para la gestión de información, Access utiliza una interfaz propietaria que se distribuye en el paquete de ofimática “Microsoft Office”. Cuenta además con un método de programación tipo “Basic”, inclusive se pueden crear formularios y métodos desde el mismo administrador de base de datos.



[18] Captura de pantalla del programa “MySQL-Front”.

Figura 3.17. Microsoft Access 2003[19]

2.10. SQL SERVER

"Microsoft SQL Server es un sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional. Sus lenguajes para consultas son T-SQL y ANSI SQL. Microsoft SQL Server constituye la alternativa de Microsoft a otros potentes sistemas gestores de bases de datos como son Oracle, PostgreSQL o MySQL."[20]

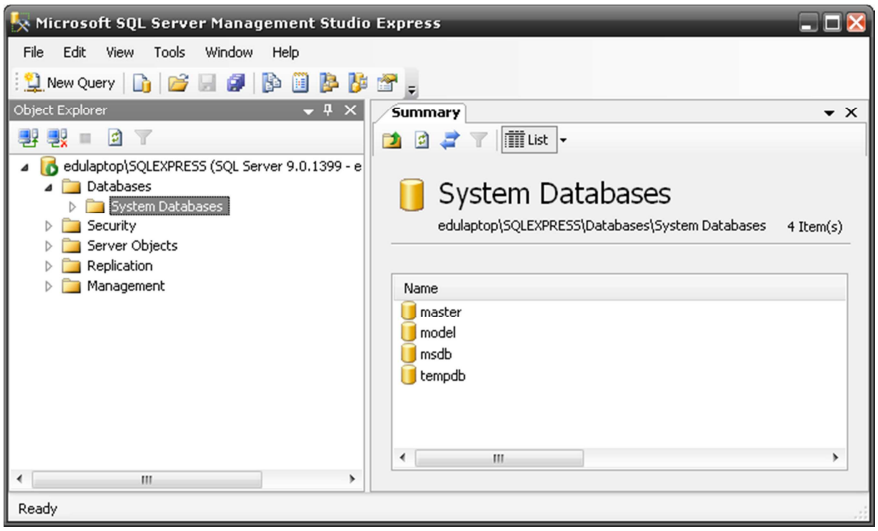


Figura 3.18. Microsoft SQL Server: Gestión de base de datos[21]

[19] Captura de pantalla del programa "Microsoft Access 2003".
[20] "Microsoft SQL Server". Link: http://es.wikipedia.org/wiki/Microsoft_SQL_Server
[21] Captura de pantalla del programa "Microsoft SQL Server Management Studio Express".

2.10.1. SQL SERVER EXPRESS

A partir de la versión de “Microsoft Visual Studio 2005” se creó una versión ligera de SQL Server, la cual se puede distribuir libremente (sin licencia) a los desarrolladores de software. A diferencia de la base de datos completa, las versiones Express de SQL Server sólo pueden tener un almacenamiento máximo de 4Gb.

2.11. HTML

“HTML, siglas de HyperText Markup Language («lenguaje de marcado de hipertexto»), hace referencia al lenguaje de marcado predominante para la elaboración de páginas web que se utiliza para describir y traducir la estructura y la información en forma de texto, así como para complementar el texto con objetos tales como imágenes. El HTML se escribe en forma de «etiquetas», rodeadas por corchetes angulares (<,>). HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script (por ejemplo JavaScript), el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML“.[22]

Código fuente 3.12. HTML: Ejemplo de código fuente

```
<HTML>
```

[22] “HTML”. Fuente: http://es.wikipedia.org/wiki/HTML

```
<HEAD>
<TITLE>ejemplo de HTML</TITLE>
</HEAD>
<BODY>
<P>Hola Mundo!</P>
</BODY>
</HTML>
```

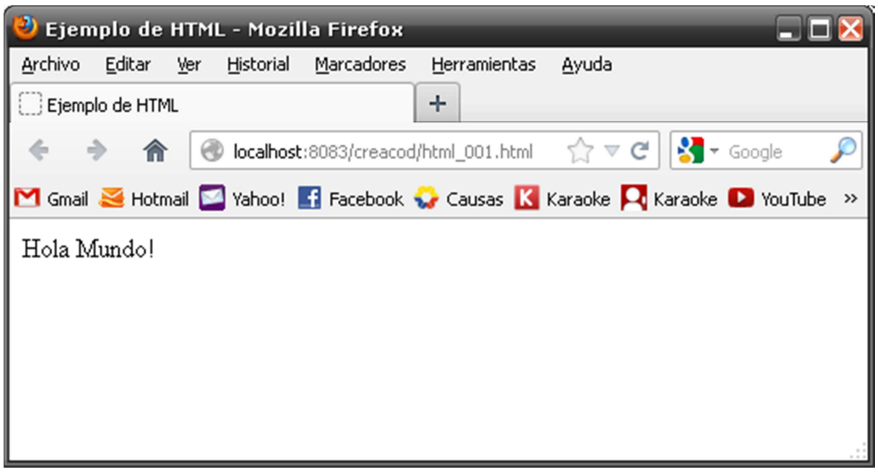


Figura 3.19. HTML: Resultado de ejemplo[23]

2.11.1. CREACIÓN DE PÁGINAS WEB CON LENGUAJE HTML

Para desarrollar archivos HTML basta con tener un editor de textos en la computadora, el método más simple es escribir el archivo desde el “Bloc de notas” de Windows, mientras que métodos más especializados pueden ser

[23] Captura de pantalla del programa “Mozilla Firefox 14” con ejemplo de HTML.

mediante el uso de herramientas diseñadas para desarrollo web como el caso de “Macromedia Dreamweaver” o “Microsoft FrontPage”.

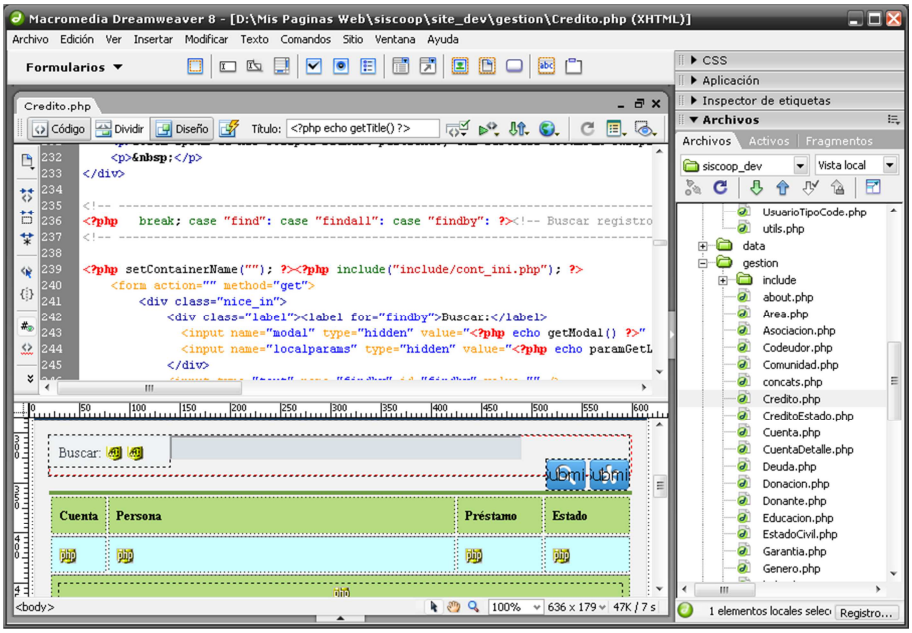


Figura 3.20. HTML: Ejemplo de edición de página con Dreamweaver[24]

2.12. PHP

“PHP es un acrónimo recursivo que significa PHP Hypertext Pre-processor. PHP es un lenguaje interpretado de propósito general ampliamente usado y que está diseñado especialmente para desarrollo web y puede ser embebido dentro de código HTML. Generalmente se ejecuta en un servidor web, tomando el código

[24] Captura de pantalla del programa “Macromedia Dreamweaver”

en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno".[25]

Código fuente 3.13. PHP: Ejemplo de código fuente

```
<html>
<head>
<title>Ejemplo de PHP</title>
</head>
<body>
</body>
<?php echo "Hola Mundo"; ?>

</html>
```

Figura 3.21. PHP: Resultado de ejemplo[26]

[25] "P.H.P.". Fuente: http://es.wikipedia.org/wiki/PHP
[26] Captura de pantalla del programa "Mozilla Firefox 14" con ejemplo de HTML.

2.13. ASP

"Active Server Pages (ASP) es una tecnología del lado servidor de Microsoft para páginas web generadas dinámicamente, que ha sido comercializada como un anexo a Internet Information Server (IIS)[27]. La tecnología ASP está estrechamente relacionada con el modelo tecnológico de su fabricante. Intenta ser solución para un modelo de programación rápida ya que programar en ASP es como programar en Visual Basic, por supuesto con muchas limitaciones ya que es una plataforma que no se ha desarrollado como lo esperaba Microsoft."[28]

ASP tiene básicamente 2 versiones, la primera es la clásica o el lenguaje A.S.P. inicial, mientras que las versiones ASP.NET cambian varios de los aspectos del lenguaje básico.

Código fuente 3.14. ASP: Ejemplo de código fuente

```
<html>
<head>
<title>Ejemplo de ASP</title>
</head>
<body>
<% Response.Write("Hola mundo") %>

</body>
</html>
```

[27] Internet Information Services o IIS es un servidor web y un conjunto de servicios para el sistema operativo Microsoft Windows. Los servicios que ofrece son: FTP, SMTP, NNTP y HTTP/HTTPS.
[28] "Active Server Pages". Fuente: http://es.wikipedia.org/wiki/Active_Server_Pages

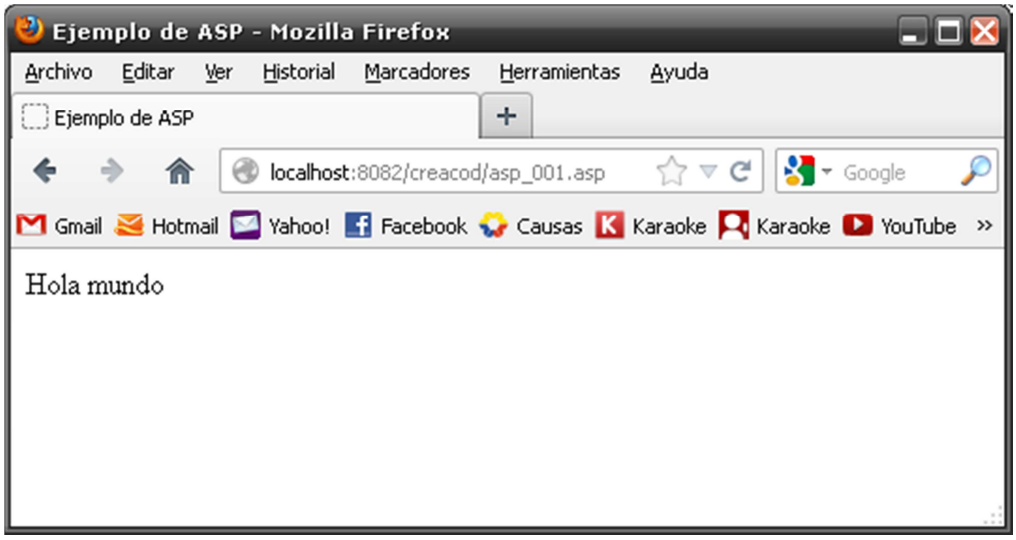


Figura 3.22. ASP: Resultado de ejemplo[29]

2.14. JSP

"JavaServer Pages (JSP) es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo. Esta tecnología es un desarrollo de la compañía Sun Microsystems. Las JSP's permiten la utilización de código Java mediante scripts. Además es posible utilizar algunas acciones JSP predefinidas mediante etiquetas. Estas etiquetas pueden ser enriquecidas mediante la utilización de Librerías de Etiquetas (TagLibs o Tag Libraries) externas e incluso personalizadas."[30]

[29] Captura de pantalla del programa "Mozilla Firefox 14" con ejemplo de HTML.
[30] "JavaServer Pages". Fuente: http://es.wikipedia.org/wiki/JavaServer_Pages

Código fuente 3.15. JSP: Ejemplo de código fuente

```
<HTML>
<HEAD>
<TITLE>Hola Mundo!</TITLE>
</HEAD>
<BODY>
<%

for ( int i = 0 ; i < 10 ; i ++ ){

%>

Hola Mundo!<br>

<%

}

%>

</BODY>
</HTML>
```

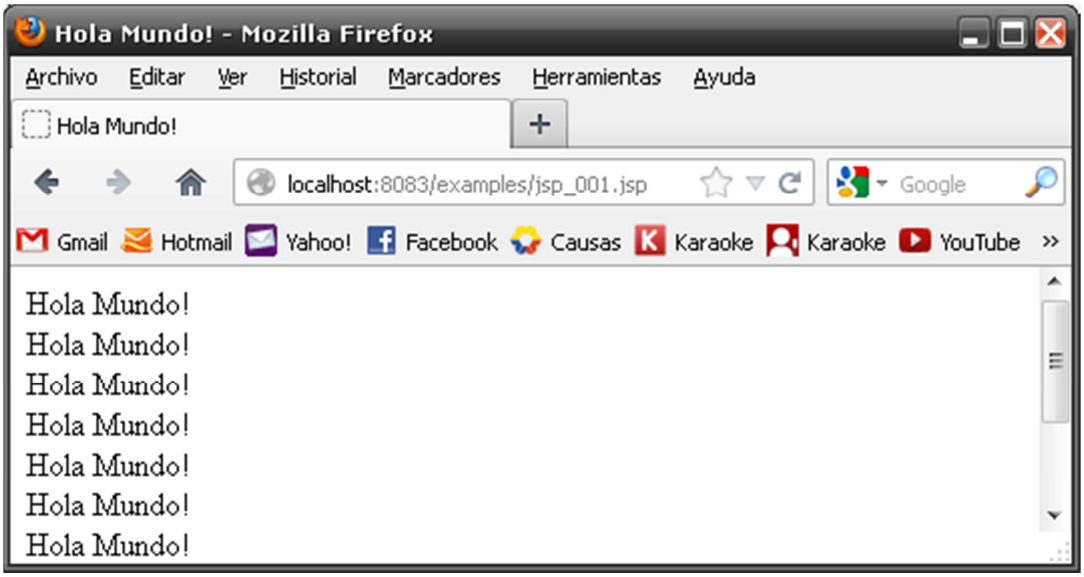


Figura 3.23. JSP: Resultado de ejemplo[31]

[31] Captura de pantalla del programa “Mozilla Firefox 14” con ejemplo de HTML.

2.15. COMPARACIÓN ENTRE PHP, ASP Y JSP

Tanto PHP, ASP como JSP con lenguajes que sirven para realizar páginas dinámicas (en cuanto a contenido) desde el servidor. Hay muchos aspectos que se han de tener en cuenta:

Tabla 3.2. Comparación entre PHP, JSP y ASP

ASPECTO	PHP	ASP	JSP
Licencias:	es gratuito	necesita de licencias de Microsoft	es gratuito
Código propietario	Necesita de un intérprete para ejecutarse. No siempre están en todas las plataformas.	solo funciona sobre Microsoft	Puede funcionar en cualquier plataforma.
Curva de aprendizaje	"sencillos" de aprender.	"sencillos" de aprender.	Es un poco más complejo ya que exige el conocimiento de las especificaciones de JAVA.
Diseño / Escalabilidad	Es un lenguaje interpretado en tiempo de ejecución. Permite diseños escalables y orientados a objetos.	Permiten diseños escalables y orientados a objetos.	Permiten diseños escalables y orientados a objetos.
Soportes de características avanzadas.	En PHP no tiene estas características.	tecnologías que soportan características como Transacciones, pool de conexiones, etc., de manera transparente al programador	tecnologías que soportan características como Transacciones, pool de conexiones, etc., de manera transparente al programador
Servidores	Hay diferentes intérpretes, algunos free y otros de pago	Solo podrá ser Microsoft.	Hay motores gratuitos (TOMCAT, JBOSS) y de pago.

2.16. XML (EXTENSIBLE MARKUP LANGUAJE)

"XML, siglas en inglés de eXtensible Markup Language ('lenguaje de marcas extensible'), es un lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C). Deriva del lenguaje SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML) para estructurar documentos grandes. A diferencia de otros lenguajes XML da soporte a bases de datos, siendo útil cuando varias aplicaciones se deben comunicar entre sí o integrar información. (Bases de datos Silberschatz). XML no ha nacido sólo para su aplicación en Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable."[32]

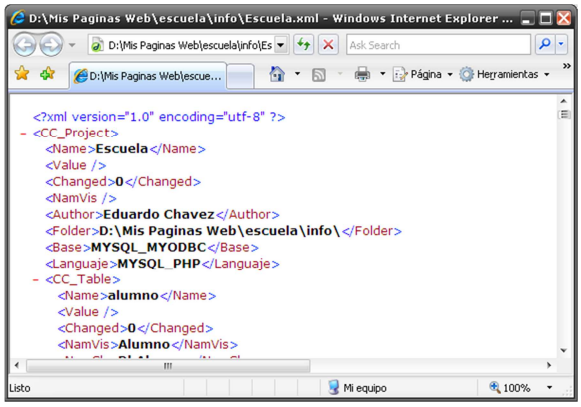


Figura 3.24. XML: Archivo de ejemplo[33]

[32] "Extensible Markup Language". Fuente: http://es.wikipedia.org/wiki/Extensible_Markup_Language
[33] Captura de pantalla del programa "Microsoft Internet Explorer 7" con ejemplo de XML

2.17. AJAX

“Ajax no es una tecnología en sí mismo. En realidad, se trata de varias tecnologías independientes que se unen de formas nuevas y sorprendentes. Las tecnologías que forman AJAX son:”[34]

- XHTML y CSS, para crear una presentación basada en estándares.
- DOM, para la interacción y manipulación dinámica de la presentación.
- XML, XSLT y JSON, para el intercambio y la manipulación de información.
- XMLHttpRequest, para el intercambio asíncrono de información.
- JavaScript, para unir todas las demás tecnologías.

Desarrollar aplicaciones AJAX requiere un conocimiento avanzado de todas y cada una de las tecnologías anteriores.

En la navegación web tradicional, el usuario realiza una acción en la página actual, la cual realiza una petición al servidor, quién envía una nueva página HTML, el navegador web entonces carga la nueva página.

Este método de navegación funciona bien, sin embargo cuando se necesita crear peticiones continuas al servidor resulta molesto tener que recargar la página varias veces. AJAX mejora esa característica de la navegación web, al actualizar

[34] “Extensible Markup Language”. Fuente: http://es.wikipedia.org/wiki/Extensible_Markup_Language

únicamente las partes del documento que se requieran. Su aplicación es tan difundida hoy en día que incluso ha llegado a superar a las aplicaciones FLASH para refrescamiento de información.

2.18. CSS

"CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas. Al crear una página web, se utiliza en primer lugar el lenguaje HTML/XHTML para marcar los contenidos, es decir, para designar la función de cada elemento dentro de la página: párrafo, titular, texto destacado, tabla, lista de elementos, etc. Una vez creados los contenidos, se utiliza el lenguaje CSS para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página, etc."[35]

Código fuente 3.16. CSS: Ejemplo de código fuente

```
<html>
<head>
<title>Ejemplo de HTML</title>
<style type="text/css">

.Estilo1 {

      font-family: Arial, Helvetica, sans-serif;

      font-size: 18px;

      color: #990000;}
```

[35] "Introducción a CSS". Fuente: http://www.librosweb.es/css/capitulo1.html

```
</style>

</head><body>
<h1 class="Estilo1">Hola mundo con CSS</h1>

</body></html>
```

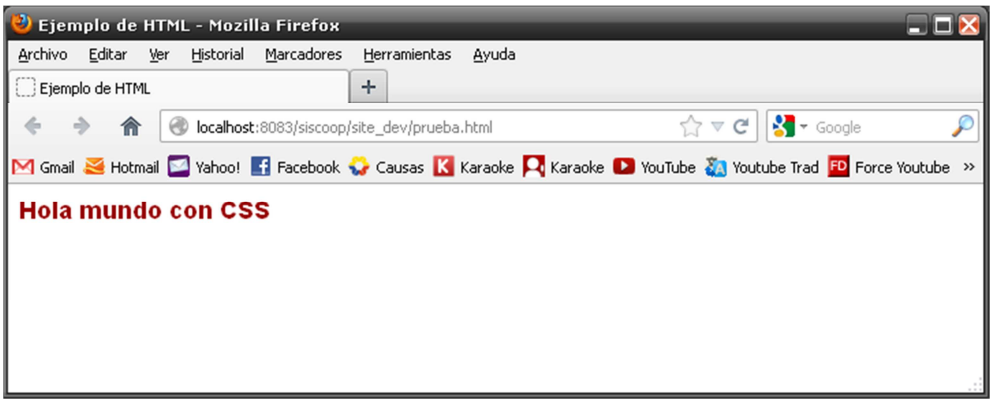


Figura 3.25. CSS: Resultado de ejemplo[36]

2.19. HERRAMIENTAS CASE

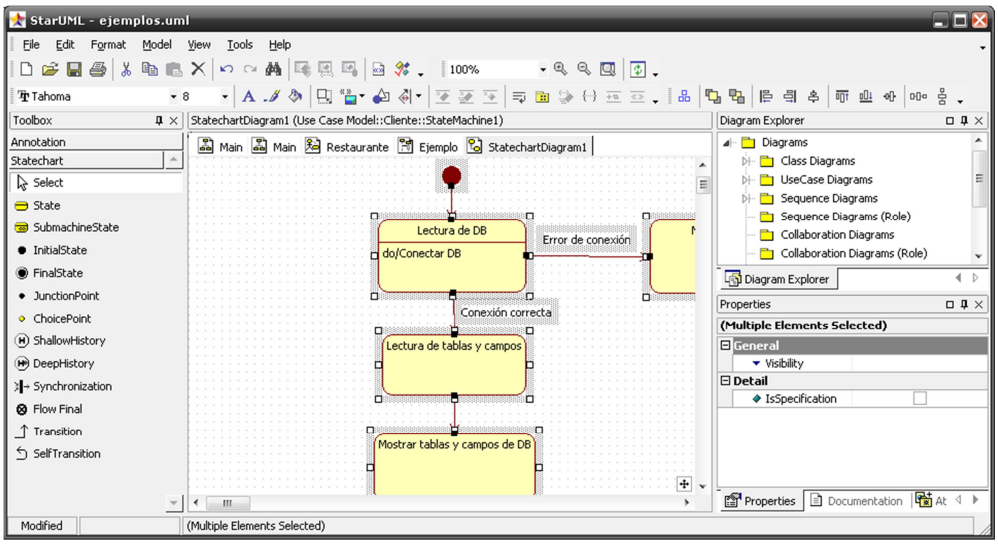
"Las herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Computadora) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y de dinero. Estas herramientas nos pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costos, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras,

[36] Captura de pantalla del programa "Mozilla Firefox 14" con resultado de ejemplo CSS.

que analizaba la relación existente entre los requisitos de un problema y las necesidades que éstos generaban, el lenguaje en cuestión se denominaba PSL (Problem Statement Language) y la aplicación que ayudaba a buscar las necesidades de los diseñadores PSA (Problem Statement Analyzer).”[37]

2.19.1. STARUML, HERRAMIENTA CASE

StarUML es una herramienta UML de código abierto, licenciado bajo una versión modificada de la licencia GNU GPL. El objetivo declarado del proyecto fue la de sustituir grandes y aplicaciones comerciales tales como Rational Rose y de Borland Together.



[37] “Herramienta CASE”. Fuente: http://es.wikipedia.org/wiki/Herramienta_CASE

Figura 3.26. StarUML[38]

StarUML soporta la mayoría de los tipos de diagramas especificados en UML 2.0. En la actualidad faltan los diagramas de objeto, paquete, el calendario y descripción de la interacción (aunque los dos primeros pueden ser adecuadamente el modelo a través del editor de diagramas de clase).

2.20. GENERADORES DE CÓDIGO FUENTE

"En programación, la generación de código es una de las fases mediante el cual un compilador convierte un programa sintácticamente correcto en una serie de instrucciones a ser interpretadas por una máquina. La entrada en esta fase viene representada, típicamente, por un Árbol Sintáctico, un Árbol de Sintaxis Abstracta, o una Representación Intermedia; la máquina destino puede ser un microprocesador o una máquina abstracta tal como una máquina virtual o un lenguaje intermedio, legible por un humano. Compiladores más sofisticados realizan múltiples traducciones en cadena (pipelining) con el fin de poder construir código para múltiples plataformas y evitar tener que construir todas las capas del compilador."[39]

En términos más generales, la generación de código: es usada para construir programas de una manera automática evitando que los programadores tengan

[38] Captura de pantalla del programa "StarUML 2.2" con ejemplo de diagrama UML.
[39] "Generación de código". Fuente: http://es.wikipedia.org/wiki/Generación_de_código

que escribir el código a mano. La generación de código puede realizarse en tiempo de ejecución, Tiempo de carga, o Tiempo de compilación.

2.21. SITIO WEB DE ÁERAS PROTEGIDAS DEL ECUADOR

Como aplicación práctica de la presente tesis se desarrollará el tema de tesis: "Análisis, diseño, desarrollo e implementación de una guía interactiva y sitio Web para las áreas protegidas del Ministerio del Ambiente"[40].

El tema de tesis permitirá al usuario tener una visión y conocimiento más amplio sobre el Sistema Nacional de Áreas Protegidas del Ecuador. Contendrá galerías fotográficas, sitios de interés turístico, tipo de vestuario, clima, geografía, así como flora, fauna, precipitación, ubicación y las provincias en las cuales se encuentran cada una de las Áreas Protegidas del Ecuador.

CAPÍTULO 3

ANÁLISIS Y DISEÑO

3.1. ANÁLISIS DE LA SITUACIÓN ACTUAL

[40] El 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).

Los programas o sitios web actuales, en su mayoría, se conectan a una base de datos para obtener información.

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.

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.

El tema de tesis a ser desarrollado es un proyecto que genera código fuente para gestionar distintas bases de datos en distintos lenguajes de programación, utilizando para ello plantillas totalmente personalizables y en una interfaz gráfica amigable para el usuario.

3.1.1 ESPECIFICACIONES DE HARDWARE Y SOFTWARE

El presente tema de tesis fue desarrollado con las siguientes especificaciones de Hardware y Software:

3.1.1.1 HARDWARE

- Computadora Intel Core2 Duo
- 2Gb de R.A.M.
- Disco duro de 500Gb

3.1.1.2 SOFTWARE

- Microsoft Windows XP/7, versión Professional
- Microsoft Visual Basic 2005/2008 Express Edition
- Microsoft C# 2005/2008 Express Edition
- Microsoft Office 2007
- MySQL 5.2
- MySQL-Front 3.1
- Power Designer 6 Data Architect
- Microsoft SQL Server 2005
- Apache web Server 2.2.21
- PHP 5.3.10
- Apache Tomcat 6
- Macromedia Dreamweaver 8
- StarUML 5.0.2
- JCreator Pro 2.2
- ISTool / Inno Setup 4.2.7
- Jasc Saint Shop Pro 8
- Mozilla Firefox 14
- HTML Help Workshop 4.74

3.2 FORMULACIÓN Y ANÁLISIS

Dada las limitaciones de los generadores de código actuales, se ha decidido que el nuevo programa a desarrollarse debe tener las siguientes características:

- El sistema operativo en el cuál funcionará el nuevo sistema será Windows XP/Vista/7
- Se programará en Visual Basic.Net Express, debido a que es una herramienta de programación de libre distribución (gratis).
- La interfaz gráfica debe ser amigable con el usuario, indicando de la manera más óptima cada uno de los parámetros de las tablas y los campos que la componen.
- El sistema se deberá conectar con distintas bases de datos (Access, SQL Server y MySQL) y debe ser capaz de leer los atributos de tablas y campos.
- El sistema generará código para los lenguajes de programación PHP, ASP y JSP.
- El sistema deberá crear 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 deberá generar las distintas pantallas para administrar la información que se obtenga de las librerías de gestión de tablas, dando la posibilidad al programador de seleccionar qué tipo de control (Combobox, Textbox, etc.) se usará para cada uno de los campos, además de indicar las funciones de validación para cada campo.

- El programa, en lo posible, debe ser 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.

3.3 ESPECIFICACIÓN DE REQUERIMIENTOS

La presente especificación de requerimientos pertenece al programa a ser desarrollado como tesis para la obtención del título de Ingeniero en Sistemas e Informática y está desarrollada siguiendo las directrices de la metodología XTREME PROGRAMMING junto con el diagrama de “Casos de Uso” del Lenguaje Unificado de Modelado (UML).

3.3.1 INTRODUCCIÓN

3.3.1.1 PROPÓSITO

El propósito del presente apartado es definir los requerimientos que debe tener el programa a ser desarrollado. Con la especificación de requerimientos se formalizará las funcionalidades de la aplicación.

3.3.1.2 METODOLOGÍA DE DESARROLLO

Para el desarrollo de la presente tesis se ha decidido utilizar la metodología "Extreme Programming" debido a que se adecúa de mejor manera a las necesidades del sistema. A continuación se presenta un cuadro comparativo entre otras 3 metodologías ágiles de desarrollo de software y la metodología XP.

Tabla 3.1. Comparación de metodologías rápidas Vs. XP.

	ASD[41]	Crystal	Scrum	XP
Sistema cambiante	5	4	5	5
Colaboración	5	5	5	5
Resultados	5	5	5	5
Simplicidad	4	4	5	5
Adaptabilidad	5	5	4	3
Excelencia técnica	3	3	3	4

3.3.1.3 NOMBRE Y LOGO DEL PROGRAMA

Dado que el presente tema de tesis es una aplicación práctica, basada en investigación y para el uso general de cualquier programador, se ha propuesto como nombre del programa a **"CreaCod"**, el cual significa "**Crea**dor de **Cod**igo"[42]. La aplicación se pretende distribuir bajo licencia GNU.

[41] ASD : Adaptive Software Development
[42] La palabra "Codigo" en el nombre "CreaCod" no tiene tilde con el fin de internacionalizar el nombre.

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.

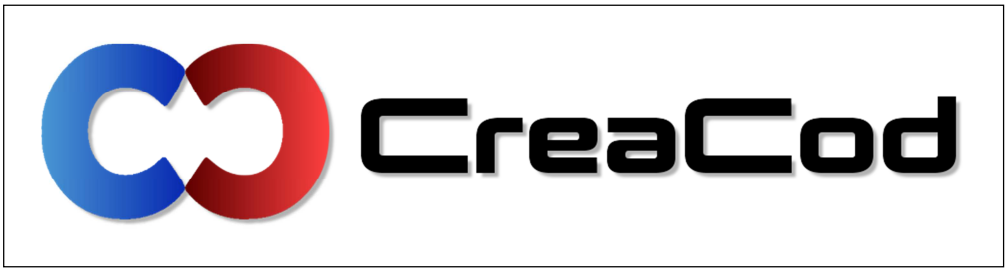


Figura 3.1. Logo de CreaCod

3.3.1.4 DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS

3.3.1.4.1 DEFINICIONES

- **Base de datos**.- Sistemas gestores de bases de datos (SGBD), que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada.
- **Tablas**.- Tipo de modelado de datos, donde se guardan los datos recogidos por un programa. Su estructura general se asemeja a la vista general de un programa de Hoja de cálculo.
- **Campos**.- Es cada una de las columnas que forman la tabla. Contienen datos de tipo diferente a los de otros campos.

- **Lenguaje de programación**.- Idioma artificial diseñado para expresar procesos que pueden ser llevadas a cabo por máquinas como las computadoras. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana.
- **Plantilla**.- Es un medio o un aparato que permite guiar, portar o construir un diseño o esquema predefinido. Una plantilla agiliza el trabajo de reproducción de muchas copias idénticas o casi idénticas (que no tiene que ser tan elaborado, sofisticado o personal). Si se quiere un trabajo más refinado, más creativo, la plantilla no es sino un punto de partida, un ejemplo, una idea aproximada de lo que se quiere hacer.

3.3.2 IDENTIFICACIÓN DE ROLES Y TAREAS

En el caso de CreaCod se ha establecido un único usuario: "*PROGRAMADOR*", 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.

3.3.2.1 TAREAS

3.3.2.1.1 PROGRAMADOR

- Indicar información general del proyecto.

- Seleccionar y configurar la base de datos a ser conectada.
- Seleccionar el lenguaje de programación en el que se creará el código fuente.
- Llenar los atributos de cada una de las tablas de la base de datos.
- Llenar los atributos de cada uno de los campos de las tablas.
- Seleccionar y llenar los atributos de la plantilla de administración GUI del proyecto.
- Generar el código fuente resultante.
- Cambiar idioma al IDE de CreaCod.

3.3.3 ESPECIFICACIÓN DE ESCENARIOS

3.3.3.1 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.

3.3.3.2 SELECCIONAR BASE DE DATOS Y LENGUAJE

la base de datos de donde se obtendrá la información de tablas y campos y seleccionar el código fuente a generar.

3.3.3.3 CONECTAR BASE DE DATOS

Indicar al programa “CreaCod” los parámetros de conexión a la base de datos.

3.3.3.4 REFRESCAR INFORMACIÓN DE BASE DE DATOS

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.

3.3.3.5 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.

3.3.3.6 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.

3.3.3.7 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.

3.3.3.8 GENERAR PROYECTO

Usando un botón, se generará el código fuente en la carpeta de proyecto, indicando los errores encontrados en caso de existir.

3.3.3.9 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.

3.3.3.10 CAMBIAR DE 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.

3.4. ANÁLISIS (EXPLORACIÓN)

3.4.1. PLANIFICACIÓN INICIAL

Planificación de requerimientos iniciales. Estas planificaciones son un modelo de cómo podría estar estructurado el proyecto.

Algunos de los requerimientos establecidos pudieron cambiar o haber sido eliminados.

3.4.2. HISTORIAS PREVISTAS

Las historias previstas son los posibles requerimientos iniciales del proyecto.

3.4.2.1. HISTORIA 1 (H1)

Historia de Usuario	
Número: 1	**Usuario :** Programador
Nombre : Lectura de base de datos	
Prioridad : ALTA	**Riesgo en Desarrollo:** MEDIO
Puntos Estimados : 10	**Iteración asignada :** 1
Programador responsable : Eduardo Chávez R.	
Descripción: La estructura de las bases de datos debe ser leída por el sistema.	
Observaciones: Cada base de datos tiene sus propios métodos para entregar la información de tablas y columnas. Las bases de datos a ser analizadas son: MySQL, SQL Server y Access	

3.4.2.2. HISTORIA 2 (H2)

Historia de Usuario	
Número: 2	**Usuario :** Programador
Nombre : Almacenamiento de información de estructura de base de datos	
Prioridad : ALTA	**Riesgo en Desarrollo:** ALTA
Puntos Estimados : 5	**Iteración asignada :** 1
Programador responsable : Eduardo Chávez R.	
Descripción: Una vez leída la información de estructura de base de datos, debe almacenarse en memoria y en algún tipo de archivo.	
Observaciones: La información deberá estar en memoria para poder trabajarla, para luego poder almacenarla en un archivo.	

3.4.2.3. HISTORIA 3 (H3)

Historia de Usuario	
Número: 3	**Usuario :** Programador
Nombre : Entorno GUI	
Prioridad : MEDIA	**Riesgo en Desarrollo:** MEDIA
Puntos Estimados : 4	**Iteración asignada :** 2
Programador responsable : Eduardo Chávez R.	
Descripción: Se debe tener una manera de poder presentar la información al usuario, a fin que el sistema sea entendible al usuario.	
Observaciones: Se deberá realizar un pre modelo de entorno GUI para acceder a la información.	

3.4.2.4. HISTORIA 4 (H4)

Historia de Usuario	
Número: 4	**Usuario :** Programador
Nombre : Establecer valores de Tablas y Columnas	
Prioridad : MEDIA	**Riesgo en Desarrollo:** MEDIA
Puntos Estimados : 7	**Iteración asignada :** 2
Programador responsable : Eduardo Chávez R.	
Descripción:	

Se debe agregar información para Tablas y Columnas, de acuerdo a la información que se necesite en las posibles plantillas que se va a generar.
Observaciones:

3.4.2.5. HISTORIA 5 (H5)

Historia de Usuario	
Número: 5	**Usuario :** Programador
Nombre : Estructura de plantillas	
Prioridad : ALTA	**Riesgo en Desarrollo:** ALTA
Puntos Estimados : 7	**Iteración asignada :** 3
Programador responsable : Eduardo Chávez R.	
Descripción: Se debe especificar los parámetros que deberán tener las plantillas a fin de poder ser leídas y cambiadas por el sistema	
Observaciones: Dependiendo de las necesidades que vayan apareciendo, se irán agregando más funciones o parámetros de plantillas.	

3.4.2.6. HISTORIA 6 (H6)

Historia de Usuario	
Número: 6	**Usuario :** Programador
Nombre : Módulo de Generación de código fuente	
Prioridad : ALTA	**Riesgo en Desarrollo:** ALTA
Puntos Estimados : 7	**Iteración asignada :** 4
Programador responsable : Eduardo Chávez R.	
Descripción: Una vez establecidos todos los valores de plantillas y datos de tablas y columnas, se deberá estructurar el método de generación del código fuente.	
Observaciones: Dependiendo de las necesidades que vayan apareciendo, se irán agregando más funciones o parámetros de plantillas.	

3.4.2.7. HISTORIA 7 (H7)

Historia de Usuario	
Número: 7	**Usuario :** Programador
Nombre : Plantillas para otros lenguajes de programación y bases	
Prioridad : ALTA	**Riesgo en Desarrollo:** ALTA
Puntos Estimados : 7	**Iteración asignada :** 5
Programador responsable : Eduardo Chávez R.	

Descripción: Se deberán crear las plantillas para el resto de bases de datos y lenguajes de programación.
Observaciones:

3.4.2.8. HISTORIA 8 (H8)

Historia de Usuario	
Número: 8	**Usuario :** Programador
Nombre : Módulos de Idioma	
Prioridad : ALTA	**Riesgo en Desarrollo:** ALTA
Puntos Estimados : 7	**Iteración asignada :** 5
Programador responsable : Eduardo Chávez R.	
Descripción: El sistema debe ser capaz de cambiar de idioma.	
Observaciones:	

3.4.2.9. HISTORIA 9 (H9)

Historia de Usuario	
Número: 9	**Usuario :** Programador
Nombre : Ayudas	
Prioridad : BAJA	**Riesgo en Desarrollo:** BAJA
Puntos Estimados : 7	**Iteración asignada :** 6
Programador responsable : Eduardo Chávez R.	
Descripción: Terminado el sistema, se deberá documentar todas las ayudas.	
Observaciones:	

3.4.3. PROTOTIPOS

3.4.3.1. PANTALLA PRINCIPAL

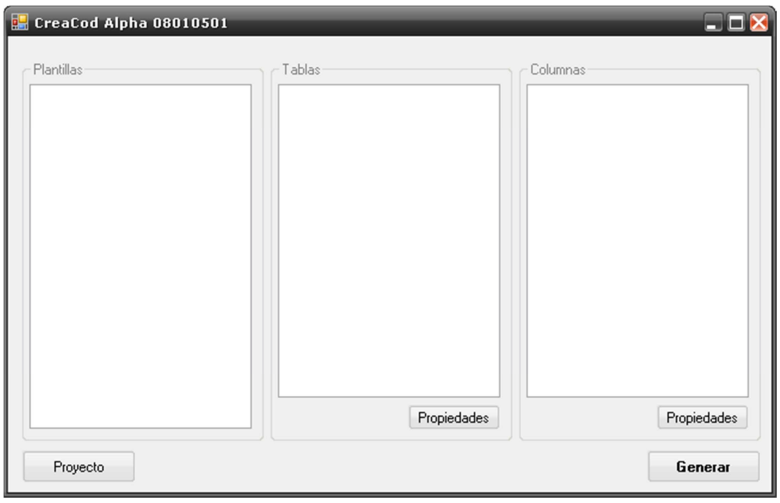


Figura 3.2. Prototipo: Pantalla principal

3.4.3.2. CONEXIÓN DE BASE Y PROPIEDADES DE PROYECTO

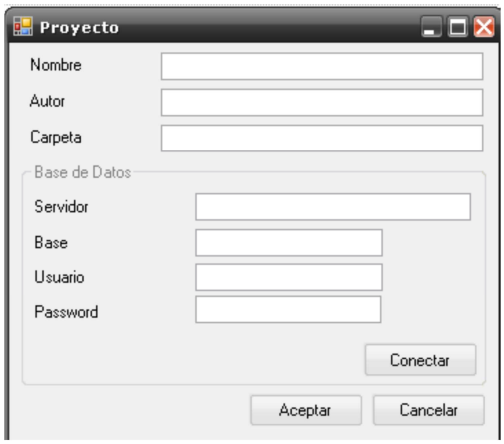


Figura 3.3. Prototipo: Pantalla de Conexión de base de datos

3.4.3.3. PANTALLA DE “TABLA”

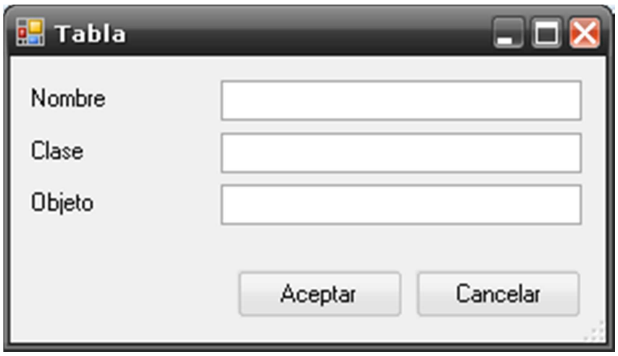


Figura 3.4. Prototipo: Propiedades de “Tabla”

3.4.3.4. PANTALLA DE “COLUMNAS”

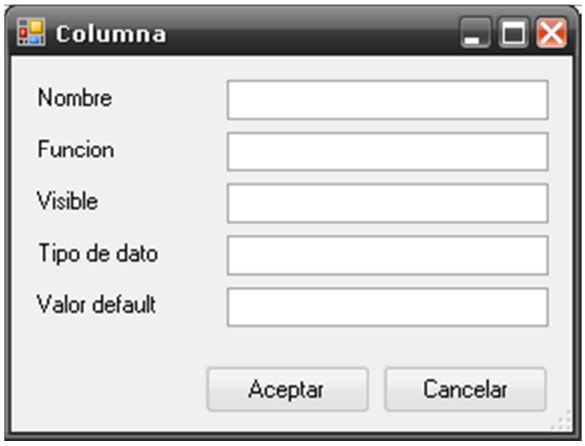


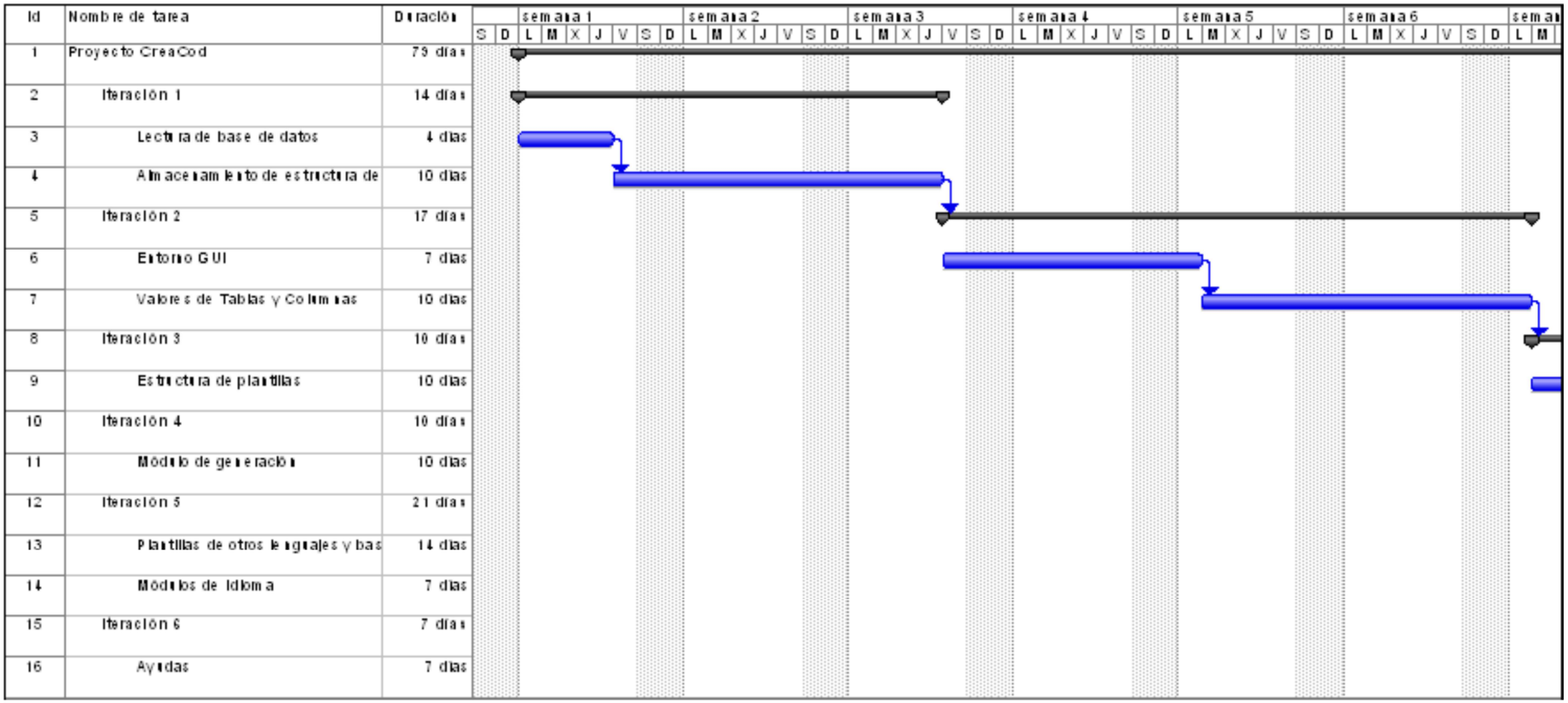
Figura 3.5. Prototipo: Propiedades de “Columna”

3.4.4. HISTORIAS DE USUARIO

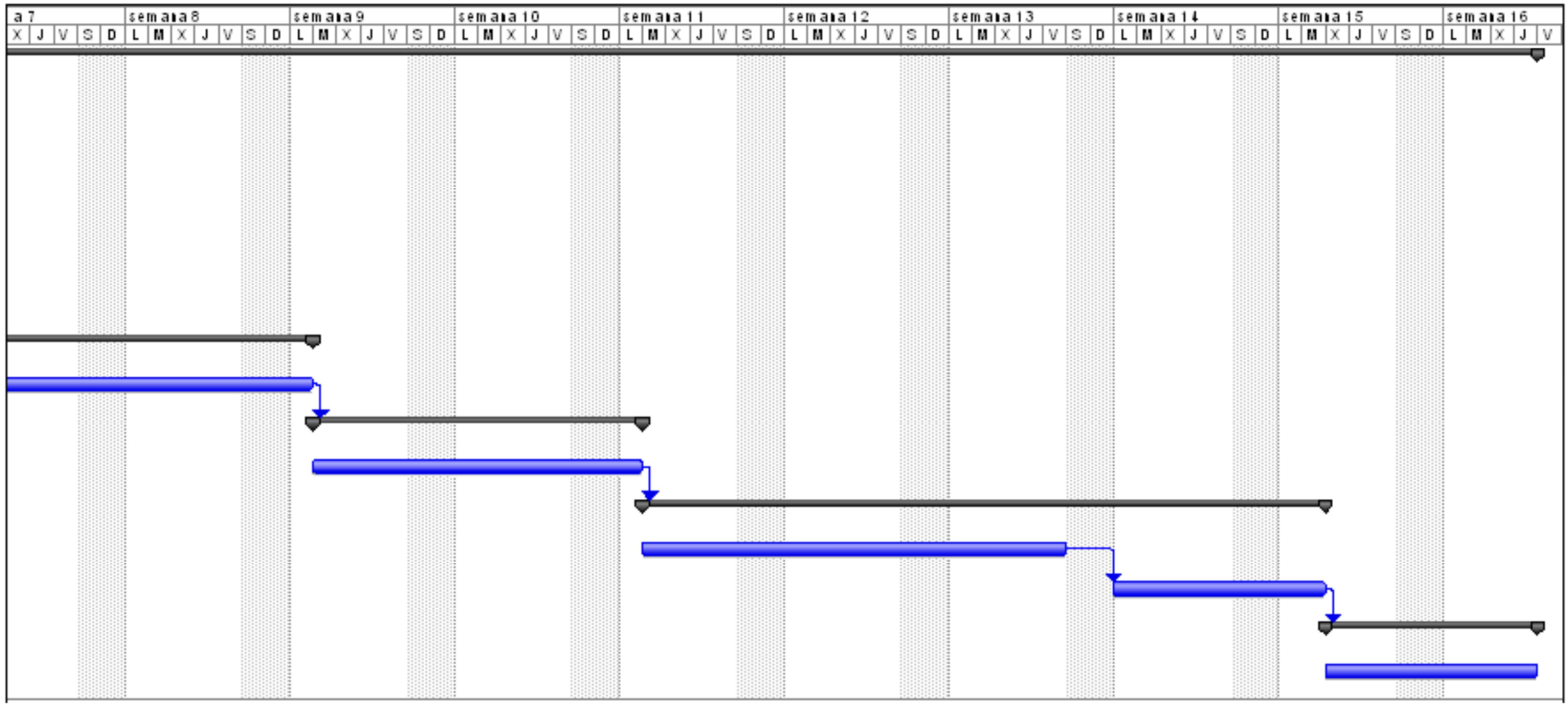
Tabla 3.2. Historias de usuario

Núm.	Nombre	Prioridad	Riesgo	Esfuerzo	Iteración
1	Lectura de base de datos	Alta	Alto	Alto	1
2	Almacenamiento de información de estructura de base de datos	Alta	Medio	Bajo	1
3	Entorno GUI	Media	Bajo	Moderado	2
4	Establecer valores de Tablas y Columnas	Alta	Alto	Alto	2
5	Estructura de plantillas	Media	Medio	Alto	3
6	Módulo de Generación de código fuente	Alta	Alto	Alto	4
7	Plantillas para otros lenguajes de programación y bases	Alta	Alto	Medio	5
8	Módulos de Idioma	Baja	Bajo	Bajo	5
9	Ayuda	Baja	Bajo	Moderado	6

3.4.5. PLAN DE ENTREGA



Id	Nombre de tarea	Duración
1	Proyecto CreaCod	79 días
2	Iteración 1	14 días
3	Lectura de base de datos	4 días
4	Almacenamiento de estructura de	10 días
5	Iteración 2	17 días
6	Entorno GUI	7 días
7	Valores de Tablas y Columnas	10 días
8	Iteración 3	10 días
9	Estructura de plantillas	10 días
10	Iteración 4	10 días
11	Módulo de generación	10 días
12	Iteración 5	21 días
13	Plantillas de otros lenguajes y bas	14 días
14	Módulos de Idioma	7 días
15	Iteración 6	7 días
16	Ayudas	7 días


a 7
semana 8
semana 9
semana 10
semana 11
semana 12
semana 13
semana 14
semana 15
semana 16
X J V S D L M X J V S D L M X J V S D L M X J V S D L M X J V S D L M X J V S D L M X J V S D L M X J V S D L M X J V

3.4.6. ITERACIONES

3.4.6.1. ITERACIÓN PRIMERA

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.

3.4.6.2. ITERACIÓN SEGUNDA

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.

3.4.6.3. ITERACIÓN TERCERA

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.

3.4.6.4. ITERACIÓN CUARTA

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.

3.4.6.5. ITERACIÓN QUINTA

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.

3.4.6.6. ITERACIÓN SEXTA

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

3.4.7. INCIDENCIAS

- Cada una de las iteraciones fueron especificadas para realizar el sistema por módulos, cada uno de los módulos depende directa o indirectamente de los módulos predecesores.
- Se cuenta con las herramientas de programación adecuadas, ya que el sistema requiere una interfaz GUI entendible al usuario.
- Se dedicará el tiempo apropiado para la investigación de las distintas bases de datos y lenguajes de programación. Este tiempo puede variar dependiendo de la historia de usuario que se esté diseñando.
- El cambio de idioma se lo ha dejado para la última iteración, contemplando desde el inicio del sistema este requisito.

3.5. PLANEAMIENTO

3.5.1. ITERACIÓN 1

3.5.1.1. TAREAS

3.5.1.1.1. TAREA 1

	Tarea
Número de Tarea : 1	**Número de Historia :** 1
Nombre de Tarea : Lectura de Información MySQL	
Tipo de Tarea: Investigación	**Puntos Estimados :**
Fecha de Inicio :	**Fecha Fin :**
Programador responsable : Eduardo Chávez R.	

Descripción: Revisar la forma que se puede obtener los datos de MySQL y describir los posibles parámetros que sean necesarios para almacenar la información dicha lectura.

3.5.1.1.2. TAREA 2

	Tarea
Número de Tarea : 2	**Número de Historia :** 1
Nombre de Tarea : Funciones de lectura	
Tipo de Tarea: Desarrollo	**Puntos Estimados :**
Fecha de Inicio :	**Fecha Fin :**
Programador responsable : Eduardo Chávez R.	
Descripción: Implementar la(s) funciones(s) con las cuales se automatice el proceso de lectura de base de datos.	

3.5.1.1.3. TAREA 3

	Tarea
Número de Tarea : 3	**Número de Historia :** 1
Nombre de Tarea : Pantalla de lectura de base de datos	
Tipo de Tarea: Desarrollo	**Puntos Estimados :**
Fecha de Inicio :	**Fecha Fin :**
Programador responsable : Eduardo Chávez R.	
Descripción: Implementar la pantalla con la cual se conectará con la base de datos.	

3.5.1.1.4. TAREA 4

	Tarea
Número de Tarea : 4	**Número de Historia :** 2
Nombre de Tarea : Almacenamiento de información	
Tipo de Tarea: Desarrollo	**Puntos Estimados :**
Fecha de Inicio :	**Fecha Fin :**
Programador responsable : Eduardo Chávez R.	
Descripción:	

Implementar las funciones para almacenar en un archivo la información de conexión a la base de datos y la información del proyecto.

3.5.1.1.5. Tarea 5

	Tarea
Número de Tarea : 5	**Número de Historia :** 2
Nombre de Tarea : Pantalla de proyecto	
Tipo de Tarea: Desarrollo	**Puntos Estimados :**
Fecha de Inicio :	**Fecha Fin :**
Programador responsable : Eduardo Chávez R.	
Descripción: Implementar la pantalla de propiedades de proyecto, a fin de separarla de la información de lectura de base de datos.	

3.5.1.2. TARJETAS C.R.C.

3.5.1.2.1. TARJETA “MiniCreator”

MiniCreator	
• Archivo Excel para ayudar al programador a crear el código fuente de las clases de información de proyecto. • En la plantilla de código fuente se debe establecer el poder guardar y recuperar la información del proyecto (conexión, tablas y columnas), el método para almacenar y leer la información debe estar en formato XML.	Programador

3.5.1.2.2. TARJETA “FrmProjectDB”

FrmProjectDB	
• Formulario de conexión de base de datos. • Establece la conexión con una base de datos.	Programador

• Obtiene todos los parámetros necesarios para realizar una conexión.
• Los parámetros son:
  o Archivo
  o Conexión DSN
  o Servidor
  o Base
  o Usuario
  o Password
  o Puerto de comunicación
  o Driver de conexión
• Retorna la cadena de conexión de la base de datos.	

3.5.1.2.3. TARJETA “InfoParam”

InfoParam	
• Clase para guardar y recuperar en un archivo XML parámetros que se puedan necesitar • Esta clase contendrá todos los parámetros necesarios para la conexión de la base de datos. • Esta clase deberá ser creada con el archivo “MiniCreator”.	Programador

TARJETA “FrmProject”

FrmProject	

• Formulario de propiedades de proyecto.
• Obtiene los datos de:
  o Nombre de Proyecto
  o Autor de proyecto
  o Carpeta de proyecto
  o Base de datos
  o Lenguaje de programación
• En esta pantalla se llama al formulario “FrmProjectDB” una vez seleccionada la base de datos.
	Programador

3.5.1.2.4. TARJETA “InfoTemplates”

InfoTemplates	
• Clase para almacenar y recuperar la información de las distintas plantillas de bases de datos y lenguajes de programación.	Programador

3.5.1.2.5. Tarjeta “FindTemplates”

FindTemplates	
• Función para buscar los archivos “.cct” y recuperar su información en las clases “InfoTemplates”.	Programador

3.5.1.3. PROTOTIPOS

Figura 3.6. Pantalla “FrmProjectDb.vb”

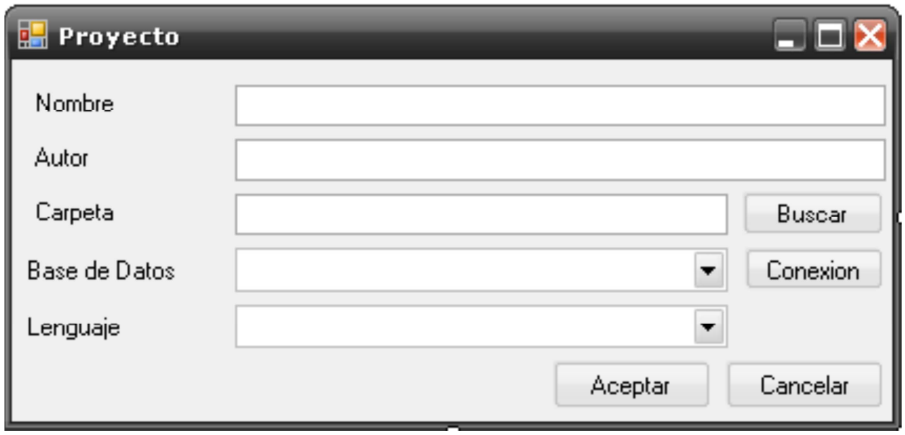


Figura 3.7. Pantalla “FrmProject.vb”

3.5.1.4. INCIDENCIAS

3.5.1.4.1. TAREA 1

- Se ha optado trabajar inicialmente con MySQL, ya que el programador utiliza unas librerías realizadas anteriormente para comunicarse con dicha base de datos.
- Adicionalmente se revisó la manera de conexión de las otras bases de datos a ser aplicadas en el proyecto (Access y SQL Server), pero la conexión con estas bases de datos se ha dejado para una futura iteración.
- Se ha revisado los métodos de conexión que existen en las bases de datos, el más óptimo es a través de "*cadenas de conexión"*, ya que este método abarca todo tipo de bases de datos.
- En las cadenas de conexión existen parámetros de:
    - Nombre de servidor
    - Nombre de base de datos o archivo de base de datos (Access)
    - Nombre de usuario
    - Contraseña
    - Driver de conexión
    - Puerto de comunicación con la base de datos

- Adicionalmente, existe la comunicación de base de datos a través de ODBC y conexiones D.S.N.[43] las cuales también pueden ser llamadas con cadenas de conexión.
- Se advierte que las bases de datos deben, necesariamente, poseer un código primario auto numérico como clave principal, además que la base de datos debe tener integridad referencial a fin de poder concatenar los valores de tablas entre sí.

3.5.1.4.2. TAREA 2

- Se ha optado por utilizar la librería creada por el programador para acceder a la información de la base de datos MySQL, ya que, utilizado lenguaje SQL, se puede acceder a la información de tablas y columnas de la base de datos.
- Las funciones para leer la base de datos ya se encuentran implementadas, sin embargo, todavía no han sido probadas para las otras bases de datos del proyecto.

3.5.1.4.3. TAREA 3

[43] **Conexión D.S.N.:** Conexión automática de Windows para tener acceso a la información de una base de datos. Cuando se establece conexión con un origen de datos desde una aplicación mediante un controlador ODBC, el controlador efectúa la conexión en lugar del usuario, ya sea localmente o a través de una red.

- Se ha creado la pantalla de “Conexión de base de datos”, denominada “FrmProjectDb.vb”.
- Se ha añadido los parámetros adicionales que no fueron considerados en un inicio.
- Se ha probado la conexión, obteniendo resultados satisfactorios.
- Existen parámetros que aún no se usan, ya que se ha planificado usarlos cuando se establezcan las conexiones con las otras bases de datos del proyecto. Los parámetros que no se utilizan son:
  - Archivo
  - Conexión DSN

3.5.1.4.4. TAREA 4

- Se ha optado utilizar el lenguaje XML para crear el archivo en el cual se guardará la información de la base de datos, ya que Visual Basic 2005 contiene funciones (`XmlReader y XmlWriter`) con las cuales se puede leer y almacenar información de este tipo de archivo.
- La información, para ser utilizada en el sistema, se almacena en una clase, para hacer más fácil su manipulación, en la clase están contenidas las funciones para leer y guardar automáticamente la información.
- Se ha creado un archivo denominado “MiniCreator.xls”, con el cual se automatiza la generación del código fuente para la creación de las clases contenedoras de información de proyecto. Este archivo contiene una plantilla de código fuente, similar a las plantillas que serán usadas para

la creación de código del sistema. A partir de este momento a este archivo se lo conocerá como “Archivo creador”.

- Luego de pruebas para obtener la información de bases de datos y lenguajes que el sistema puede utilizar, se ha decidido crear unos archivos de información previa, los cuales tendrán como extensión “.cct”.
- Los archivos “.cct” no tendrán como formato a XML, ya que dichos archivos pueden ser editados o creados por el usuario final. Para facilidad se ha decidido trabajarlos como archivos de texto delimitados.
- La información de bases de datos se ha colocado en la carpeta “{PROJECT}\Templates\Bases”. Los archivos de cabecera tendrán como formato “{NOMBRE_BASE}.cct”. A partir de este momento a estos archivos se los conocerá como “Archivos Bases”.
- La información de lenguajes de programación se ha colocado en la carpeta “{PROJECT}\Templates\Classes\{LANGUAJE}”. Dado que existirán archivos de plantillas por cada lenguaje, se ha decidido separarlos en carpetas, los archivos de cabecera tendrán en nombre “info.cct”. A partir de este momento a estos archivos se los conocerá como “Archivos Classes”.
- La clase para los archivos “Classes” y “Templates” se crearán con la hoja “TEMPLATES” del “Archivo creador”, ya que la información que contendrán es muy similar, pudiendo ser almacenada con una misma clase.

3.5.1.4.5. TAREA 5

- Se ha creado la pantalla de “Propiedades de proyecto”, denominada “FrmProject.vb”.
- Se ha analizado la posible estructuración de la creación de código fuente y se ha decidido seleccionar primeramente la base de datos, para luego seleccionar el lenguaje de programación que se va a utilizar.
- Se ha creado la variable “Carpeta de Proyecto”, en la cual se almacenarán los archivos que se generen, a la vez que guardará el archivo de proyecto, donde se almacenará toda la información de la base de datos (Conexiones, tablas, columnas, etc.)
- Se ha creado el botón “Conexión” con el cual, una vez seleccionada la base de datos, se procederá a mostrar la pantalla de conexión.
- Se ha creado la función “FindTemplates” la cual hará un barrido buscando los archivos “.cct” de “Bases” y “Classes”.

3.5.2. ITERACIÓN 2

3.5.2.1. HISTORIAS ADICIONALES

3.5.2.1.1. HISTORIA 10 (H10)

Historia de Programador	
Número: 10	**Usuario :** Programador
Nombre : Funciones de Columnas	
Prioridad : BAJA	**Riesgo en Desarrollo:** MEDIA
Puntos Estimados : 7	**Iteración asignada :** 2
Programador responsable : Eduardo Chávez R.	

Descripción: Las columnas deberán tener funciones de entrada, salida y validación. Las columnas pueden tener varios tipos de datos, en el caso de los lenguajes de programación, estos valores deben ser ingresados o extraídos con funciones específicas de cada lenguaje.
Observaciones: Por ejemplo, en JAVA, para leer un registro de tipo entero se usa la función “Integer.parseInt”, mientras que para agregar el dato de registro se usa la función “String.valueOf”. Al igual que en el ejemplo, también se deberá preestablecer ciertas funciones para validar los datos antes que sean ingresados a la base de datos.

3.4.1.1.1. Historia 11 (H11)

Historia de Programador	
Número: 11	**Usuario :** Programador
Nombre : Entidades y Relaciones	
Prioridad : MEDIA	**Riesgo en Desarrollo:** MEDIA
Puntos Estimados : 7	**Iteración asignada :** 2
Programador responsable : Eduardo Chávez R.	
Descripción: Una columna de una tabla puede ser una relación hacia otra Tabla, se deberá tener en cuenta este detalle para las propiedades de tablas y columnas.	
Observaciones: Dado que el sistema creará clases para la comunicación con la base de datos, si en el caso exista una entidad y una relación, se deberá crear automáticamente un objeto de clase de la entidad.	

3.5.2.2 TAREAS

3.5.2.2.1. TAREA 6

	Tarea
Número de Tarea : 6	**Número de Historia :** 3
Nombre de Tarea : Entorno GUI	

Tipo de Tarea: Investigación	**Puntos Estimados :**
Fecha de Inicio :	**Fecha Fin :**
Programador responsable : Eduardo Chávez R.	
Descripción: Revisar la posible manera de mejorar el entorno GUI de la pantalla principal.	

3.5.2.2.2. TAREA 7

	Tarea
Número de Tarea : 7	**Número de Historia :** 3
Nombre de Tarea : Pantalla principal	
Tipo de Tarea: Desarrollo	**Puntos Estimados :**
Fecha de Inicio :	**Fecha Fin :**
Programador responsable : Eduardo Chávez R.	
Descripción: Crear la pantalla principal del sistema, colocar todos los botones que vayan a ser necesarios, aun cuando todavía no vayan a ser utilizados. Dividir en paneles las partes de tablas, columnas y plantillas.	

3.5.2.2.3. TAREA 8

	Tarea
Número de Tarea : 8	**Número de Historia :** 3
Nombre de Tarea : Datos de Tabla	
Tipo de Tarea: Investigación	**Puntos Estimados :**
Fecha de Inicio :	**Fecha Fin :**
Programador responsable : Eduardo Chávez R.	
Descripción: Investigar los posibles parámetros que se necesiten para los valores de tablas. Crear la clase de control para la información de tablas.	

3.5.2.2.4. TAREA 9

	Tarea
Número de Tarea : 9	**Número de Historia :** 4
Nombre de Tarea : Pantalla de Propiedades de Tabla	

Tipo de Tarea: Desarrollo	**Puntos Estimados :**
Fecha de Inicio :	**Fecha Fin :**
Programador responsable : Eduardo Chávez R.	
Descripción: Desarrollar la pantalla de propiedades de Tabla.	

3.5.2.2.5. TAREA 10

	Tarea
Número de Tarea : 10	**Número de Historia :** 4/10/11
Nombre de Tarea : Datos de columna	
Tipo de Tarea: Investigación	**Puntos Estimados :**
Fecha de Inicio :	**Fecha Fin :**
Programador responsable : Eduardo Chávez R.	
Descripción: Investigar los posibles parámetros que se necesiten para los valores de columnas. Crear la clase de control para la información de columnas.	

3.5.2.2.6. TAREA 11

	Tarea
Número de Tarea : 11	**Número de Historia :** 4/10/11
Nombre de Tarea : Pantalla de Propiedades de Columna	
Tipo de Tarea: Investigación	**Puntos Estimados :**
Fecha de Inicio :	**Fecha Fin :**
Programador responsable : Eduardo Chávez R.	
Descripción: Desarrollar la pantalla de propiedades de Columna.	

3.5.2.3 TARJETAS C.R.C.

3.5.2.3.1 TARJETA “FrmMain”

FrmMain	

• Formulario principal del programa.
• Muestra la información correspondiente al proyecto en 4 lugares:
  o Barra de menú
  o Barra de íconos
  o Propiedades de tablas
  o Plantillas
  o Propiedades de columnas
• Los botones y partes del formulario fueron colocados, sin embargo el sistema por lo pronto sólo puede leer la información de tablas y columnas.
• La información de tablas se ubica en la parte izquierda, mientras que la información de columnas se muestra en la parte izquierda, siempre y cuando se haya seleccionado una tabla.
• La información de las plantillas se ubica en la parte izquierda inferior de la pantalla.
• Se analiza y coloca los posibles menús e íconos que puede contener la pantalla principal.
	Programador

3.5.2.3.2 TARJETA “FrmTables”

FrmTables	
• Formulario para editar las propiedades de cada tabla. • Este formulario es accedido usando el icono  de la pantalla principal o presionando doble-clic sobre la tabla a ser editada • Se analizó los posibles parámetros de tabla y se colocaron los siguientes valores: o Nombre de clase	Programador

o Nombre de objeto o Nombre visible o Columna/Campo principal o Columna/Campo visible o Columna/Campo de orden o Columna/Campo habilitado	

3.5.2.3.3 TARJETA “InfoProject”

InfoProject	
• Clase donde se almacena y edita la información general del proyecto, la información de las plantillas usadas y demás configuraciones directamente ligadas con la información de la base de datos. • Tiene objetos de tipo: o InfoTable.- Información de cada una de las tablas. o InfoParam.- Parámetros generales del proyecto. o InfoParam.- Datos y propiedades de las plantillas seleccionadas. • Esta clase es llamada por la librería de almacenamiento de archivo.	Programador

3.5.2.3.4 TARJETA “InfoTable”

InfoTable	
• Clase donde se almacena y edita la información de cada una de las tablas, la información de las plantilla de código fuente y demás configuraciones directamente ligadas con la información de la tabla. • Tiene objetos de tipo: o InfoColumn.- Información de cada una de los campos. o InfoParam.- Datos y propiedades de la plantilla seleccionada. • Esta clase es llamada por la librería InfoProject para entregar o recibir la información de tablas.	Programador

3.5.2.3.5 TARJETA “InfoColumn”

InfoColumn	
• Clase donde se almacena y edita la información de cada uno de los registros, la información de las plantilla de código fuente y demás configuraciones directamente ligadas con la información de columna. • Tiene objetos de tipo: o InfoColumnTem.- Datos y propiedades de las plantillas seleccionadas. • Esta clase es llamada por la librería InfoTable para entregar o recibir la información de columnas.	Programador

3.5.2.3.6 TARJETA “InfoColumnTem”

InfoColumnTem	
• Clase donde se almacena y edita la información de las plantillas de columna. Tiene objetos de tipo: o InfoParams.- Datos y propiedades de parámetros. • Esta clase es llamada por la librería InfoColumn para entregar o recibir la información de columnas. •	Programador

3.5.2.3.7 TARJETA “InfoParam”

InfoParam	
• Clase donde se almacena y edita los distintos parámetros de tabla, celda o plantilla: • Esta clase es llamada por la gran mayoría de de librerías con el fin de almacenar cada uno de los parámetros que se seleccionen para Proyecto, Tabla, Campo o plantilla.	Programador

3.5.2.4 PANTALLAS

A partir de esta historia de usuario las pantallas y clases dejan de ser "Prototipos" y pasan a ser las pantallas y clases finales del programa.

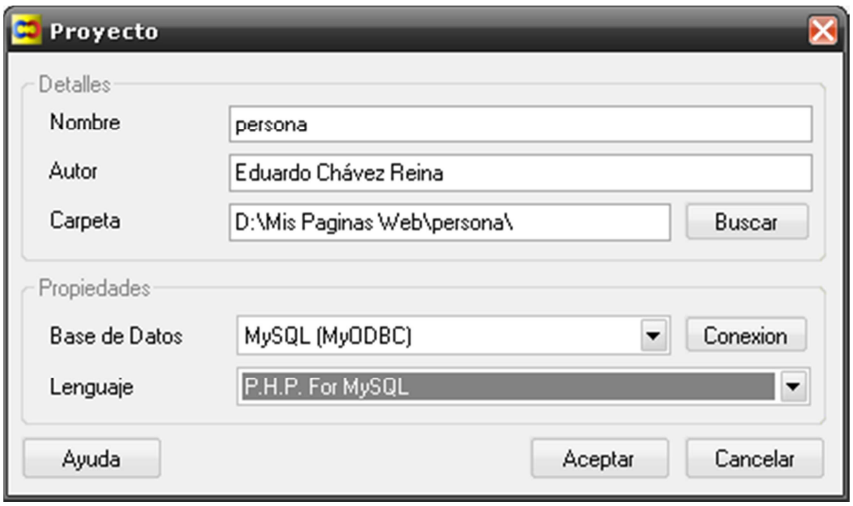


Figura 3.8. Pantalla de propiedades de Proyecto

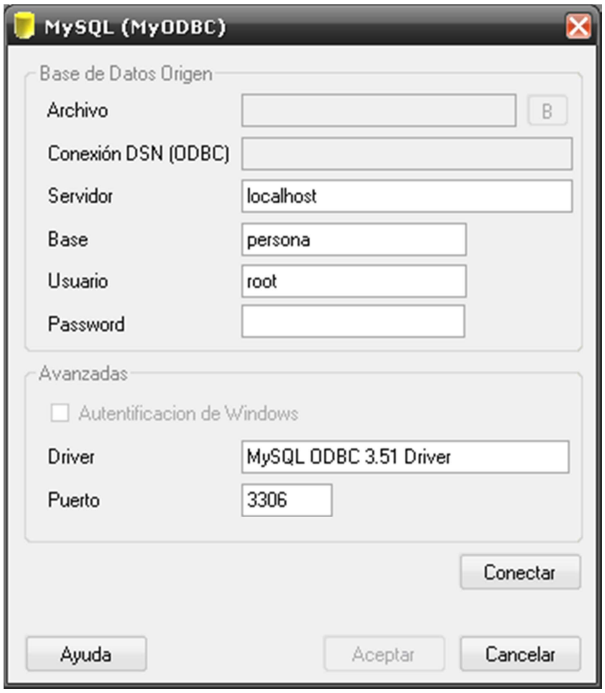


Figura 3.9. Pantalla de conexión de base de datos

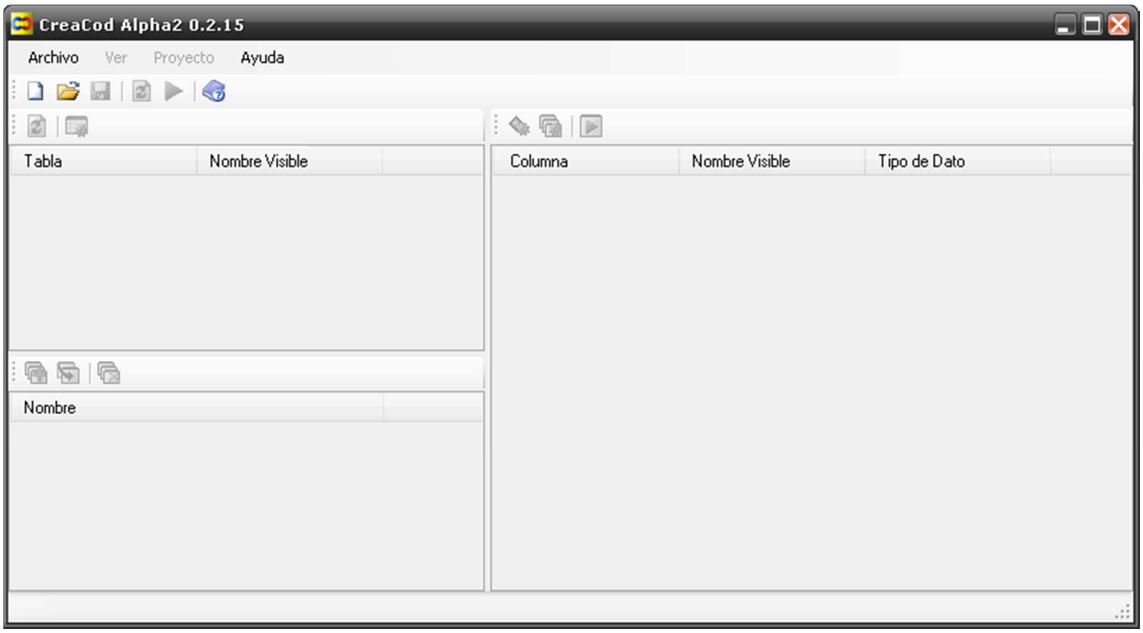


Figura 3.10. Pantalla principal sin datos

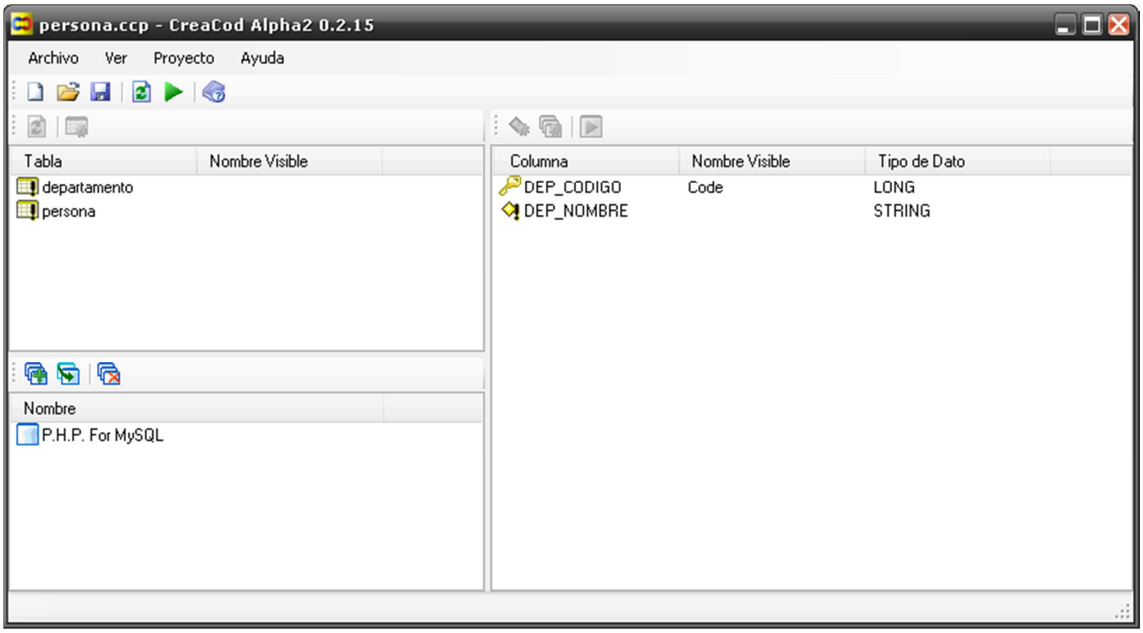


Figura 3.11. Prototipo: Pantalla principal con datos reales e íconos identificativos

Figura 3.12. Edición de parámetros de “Tabla” sin datos

Figura 3.13. Edición de parámetros de “Tabla” con datos

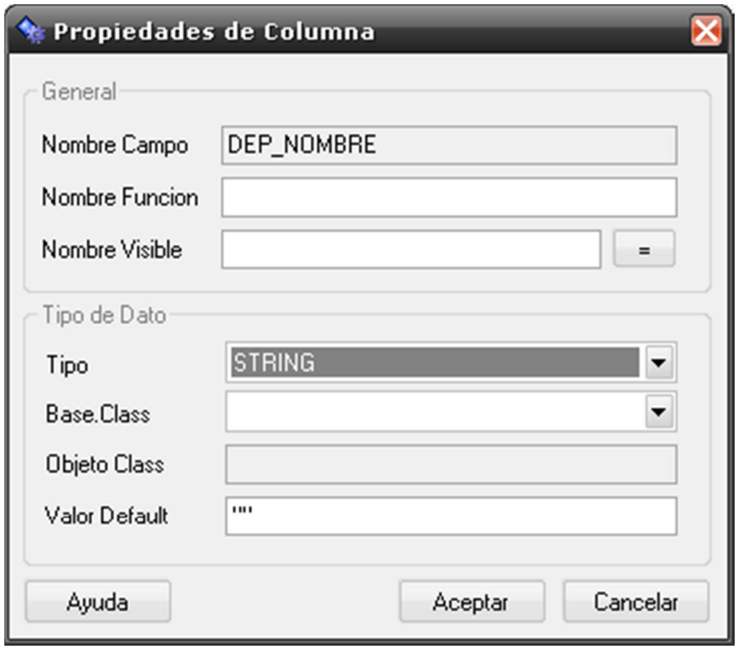


Figura 3.14. Edición de parámetros de “Columnas” sin datos

Figura 3.15. Edición de parámetros de “Columnas” con datos

Figura 3.16. Edición de parámetros de “Columnas” con Base.Class

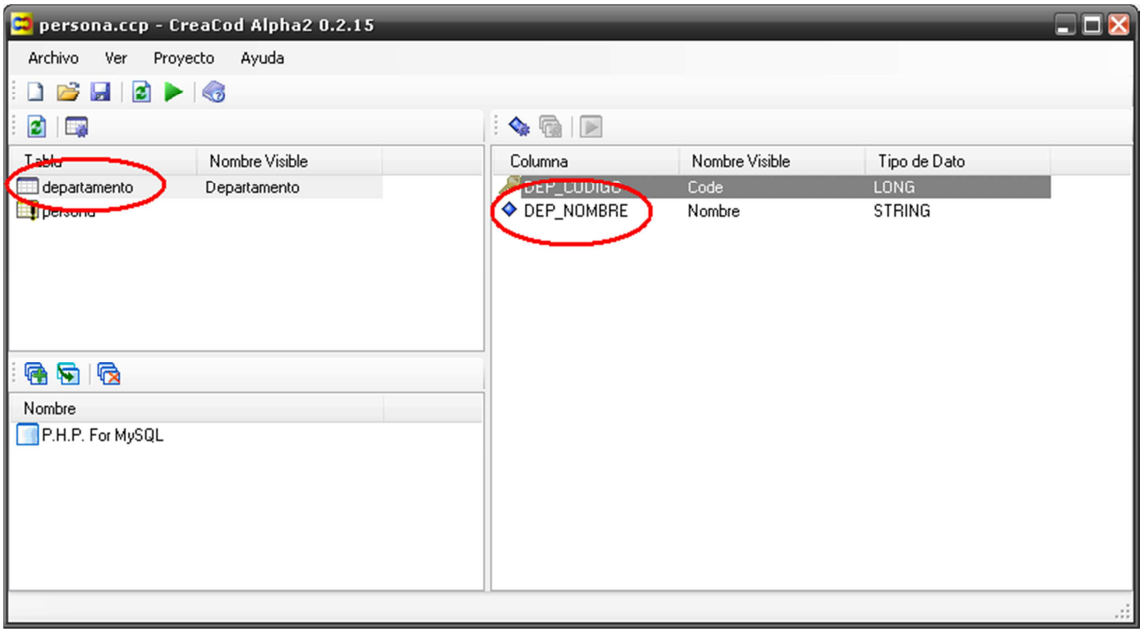


Figura 3.17. Cambio de íconos informativos de tabla y columna

3.5.2.5 INCIDENCIAS

3.5.2.5.1 TAREA 6

- Se creó una nueva pantalla principal para CreaCod.
- A los formularios existentes se mejoró la presentación.
- Los formularios poseen menú y/o botón de “Ayuda”, sin embargo aún no son funcionales
- Se creó el menú de cambio de idioma, sin embargo aún no es funcional.
- Se creó el botón de configuración de plantilla GUI, sin embargo aún no es funcional. Este requerimiento se ha movido a la iteración 3 debido a

que aún no se ha creado una plantilla para la generación de dicho código.

3.5.2.5.2 TAREA 7

- Se colocaron todos los posibles menús, botones y contenedores que pueda utilizar el programa, los controles son funcionales en un 70%.
- La información de la pantalla principal se dividió en paneles y se agregó barras de íconos para los procesos frecuentes del programa, además, las propiedades de tablas, columnas y plantillas tienen una mini-barra de íconos en la parte superior.
- El sistema posee la clase “InfoProject” que almacena la información del proyecto, “InfoTable” para almacenar la información de tablas e “InfoColumn” e “InfoParam” para almacenar la información de columnas y parámetros. Todas estas librerías se comunican entre sí para entregar y recibir de la clase de almacenamiento la información de los parámetros.
- Se agregaron íconos identificativos al lado izquierdo del nombre de las tablas para saber si están configuradas o no, esta funcionalidad la controla directamente la clase “InfoTable” retornando “true” si la información de tabla está completa.
- Se agregaron íconos identificativos al lado izquierdo del nombre de las columnas para saber si están configuradas o no, esta funcionalidad la

controla directamente la clase “InfoColumn” retornando “true” si la información de columna está completa.

3.5.2.5.3 TAREA 8

- Se creó la clase de control necesaria para almacenar y gestionar la información de tablas, esta clase además se conecta directamente con la clase de almacenamiento de información: reenvía su información cuando almacena y lee el archivo .cpp al abrirlo. Se encarga además de almacenar la información de plantilla de la clase de control de base de datos.
- Se agregó el formulario de propiedades de tabla con todos los posibles parámetros necesarios para su correcto funcionamiento.

3.5.2.5.4 TAREA 9

- Se agregó el formulario de propiedades de tabla con todos los posibles parámetros necesarios para su correcto funcionamiento.
- El formulario es funcional al 100%.
- En el caso de requerir nuevos parámetros, se debe cambiar la clase de control de tablas y agregar el nuevo control al formulario. Las clases directamente almacenarán y leerán los parámetros.

3.5.2.5.5 TAREA 10

- Se creó la clase de control necesaria para almacenar y gestionar la información de columnas, esta clase se utiliza en la clase de control de tablas, envía y recibe la información para almacenar o recibir la información de columnas del archivo .ccp. Se encarga además de almacenar la información de las distintas plantillas que se usen en las tablas y campos.
- Se agregó el formulario de propiedades de columna con todos los posibles parámetros necesarios para su correcto funcionamiento.
- El parámetro Base.Class se conecta directamente con la clase “InfoTable” para obtener los nombres de las tablas existentes en la base de datos. Se debe tomar en cuenta que la plantilla de generación de código fuente debe crear el objeto que asocie este dato con la clase seleccionada.
- El valor de columna “Code” es el código primario auto numérico que se usa como clave primaria en la base de datos y no es editable. Este dato deberá ponerse como requisito en el manual.

3.5.2.5.6 TAREA 11

- Se agregó el formulario de propiedades de columna con todos los posibles parámetros necesarios para su correcto funcionamiento.
- El formulario es funcional en un 80%.

- El botón de configuración GUI no es funcional debido a que aún no se especifica los valores para las plantillas de gestión GUI de las tablas.
- En el caso de requerir nuevos parámetros, se debe cambiar la clase de control de columnas y agregar el nuevo control al formulario. Las clases directamente almacenarán y leerán los parámetros.

3.5.3. ITERACIÓN 3

3.5.3.1. HISTORIAS ADICIONALES

3.5.3.1.1. HISTORIA 12 (H12)

Historia de Programador	
Número: 12	**Usuario :** Programador
Nombre : Plantilla de Administración GUI	
Prioridad : ALTO	**Riesgo en Desarrollo:** ALTO
Puntos Estimados : 9	**Iteración asignada :** 3
Programador responsable : Eduardo Chávez R.	
Descripción: Se debe crear la plantilla para generar código de administración GUI para PHP.	
Observaciones: Esta historia no se pudo realizar en la iteración 2 debido a que aún no se tenía un modelo de plantilla para gestión de administración GUI. Tomar en cuenta que este requisito puede afectar el calendario de entregas del programa.	

3.5.3.1.2. HISTORIA 13 (H13)

Historia de Programador	
Número: 13	**Usuario :** Programador
Nombre : plantilla para base de datos MySQL	
Prioridad : ALTO	**Riesgo en Desarrollo:** MEDIO

Puntos Estimados : 5	**Iteración asignada :** 3
Programador responsable : Eduardo Chávez R.	
Descripción: Se debe depurar y dejar a punto la plantilla para obtener los campos y columnas de una base de datos MySQL.	
Observaciones: Esta plantilla debe estar 100% funcional debido a que será una especie de "matriz" para el resto de plantillas de conexión de bases de datos.	

3.5.3.1.3. HISTORIA 14 (H14)

Historia de Programador	
Número: 14	**Usuario :** Programador
Nombre : plantilla para base de datos Access	
Prioridad : ALTO	**Riesgo en Desarrollo:** MEDIO
Puntos Estimados : 5	**Iteración asignada :** 3
Programador responsable : Eduardo Chávez R.	
Descripción: Se debe crear la plantilla para obtener los campos y columnas de una base de datos Access.	
Observaciones: Según se pudo investigar, la base de datos Access no puede enviar valores de tipo de datos mediante sentencias SQL. De ser posible investigar un método para obtener dichos valores.	

3.5.3.1.4. HISTORIA 15 (H15)

Historia de Programador	
Número: 15	**Usuario :** Programador
Nombre : plantilla para base de datos SQL Server	
Prioridad : ALTO	**Riesgo en Desarrollo:** MEDIO
Puntos Estimados : 5	**Iteración asignada :** 3
Programador responsable : Eduardo Chávez R.	
Descripción: Se debe crear la plantilla para obtener los campos y columnas de una base de datos SQL Server.	
Observaciones:	

Según se pudo investigar, la base de datos SQL Server necesita de ciertos plugins para conectarse con PHP y aún no se ha podido encontrar un método para conexión con JSP.

3.5.3.1.5. HISTORIA 16 (H16)

Historia de Programador	
Número: 16	**Usuario :** Programador
Nombre : plantilla para lenguaje de programación PHP	
Prioridad : ALTO	**Riesgo en Desarrollo:** MEDIO
Puntos Estimados : 5	**Iteración asignada :** 3
Programador responsable : Eduardo Chávez R.	
Descripción: Se debe depurar la plantilla para la gestión de base de datos usando PHP. Establecer las funciones y métodos a ser usados por la clase para enviar y recibir información de la base de datos. Establecer las funciones y métodos a ser usados por la clase para enviar y recibir la información de los registros de la tabla.	
Observaciones: Esta plantilla debe estar 100% funcional debido a que será una especie de “matriz” para el resto de plantillas de conexión de bases de datos. Se debe investigar cómo realizar la conexión de PHP para las bases de datos MySQL, Access y SQL Server.	

3.5.3.1.6. HISTORIA 17 (H17)

Historia de Programador	
Número: 17	**Usuario :** Programador
Nombre : plantilla para lenguaje de programación ASP	
Prioridad : ALTO	**Riesgo en Desarrollo:** MEDIO
Puntos Estimados : 5	**Iteración asignada :** 3
Programador responsable : Eduardo Chávez R.	
Descripción: Se debe depurar la plantilla para la gestión de base de datos usando ASP.	
Observaciones: Se debe investigar cómo realizar la conexión de ASP para las bases de datos MySQL, Access y SQL Server.	

3.5.3.1.7. HISTORIA 18 (H18)

Historia de Programador	
Número: 18	**Usuario :** Programador
Nombre : plantilla para lenguaje de programación JSP	
Prioridad : ALTO	**Riesgo en Desarrollo:** MEDIO
Puntos Estimados : 5	**Iteración asignada :** 3
Programador responsable : Eduardo Chávez R.	
Descripción: Se debe depurar la plantilla para la gestión de base de datos usando JSP.	
Observaciones: Se debe investigar cómo realizar la conexión de JSP para las bases de datos MySQL, Access y SQL Server.	

3.5.3.2. TAREAS

3.5.3.2.1. TAREA 12

	Tarea
Número de Tarea : 12	**Número de Historia :** 5/12
Nombre de Tarea : Plantilla de administración GUI para PHP	
Tipo de Tarea: Investigación/Programación	**Puntos Estimados :**
Fecha de Inicio :	**Fecha Fin :**
Programador responsable : Eduardo Chávez R.	
Descripción: Diseñar una plantilla en lenguaje PHP que pueda realizar la gestión de tablas, usando para ello las clases y objetos generados por las clases de control de bases de datos.	

3.5.3.2.2. TAREA 13

	Tarea
Número de Tarea : 13	**Número de Historia :** 5/13
Nombre de Tarea : Plantilla de conexión de base de datos MySQL	
Tipo de Tarea: Investigación/Programación	**Puntos Estimados :**
Fecha de Inicio :	**Fecha Fin :**

Programador responsable : Eduardo Chávez R.
Descripción: Optimizar la plantilla de conexión de MySQL, indicando los parámetros necesarios para una correcta configuración. Indicar en la plantilla los tipos de datos, métodos y funciones necesarios para realizar la conexión y manipulación de tablas y campos. Indicar en la plantilla los archivos y/o carpetas necesarios para el correcto funcionamiento de la plantilla.

3.5.3.2.3. TAREA 14

	Tarea
Número de Tarea : 14	**Número de Historia :** 5/14
Nombre de Tarea : Plantilla de clase de control para base de datos Access	
Tipo de Tarea: Investigación/Programación	**Puntos Estimados :**
Fecha de Inicio :	**Fecha Fin :**
Programador responsable : Eduardo Chávez R.	
Descripción: Crear la plantilla de conexión de Access, indicando los parámetros necesarios para una correcta configuración. Indicar en la plantilla los tipos de datos, métodos y funciones necesarios para realizar la conexión y manipulación de tablas y campos. Indicar en la plantilla los archivos y/o carpetas necesarios para el correcto funcionamiento de la plantilla.	

3.5.3.2.4. TAREA 15

	Tarea
Número de Tarea : 15	**Número de Historia :** 5/15
Nombre de Tarea : Plantilla de conexión de base de datos SQL Server	
Tipo de Tarea: Investigación/Programación	**Puntos Estimados :**
Fecha de Inicio :	**Fecha Fin :**
Programador responsable : Eduardo Chávez R.	
Descripción: Crear la plantilla de conexión de SQL Server 2005, indicando los parámetros necesarios para una correcta configuración.	

Indicar en la plantilla los tipos de datos, métodos y funciones necesarios para realizar la conexión y manipulación de tablas y campos. Indicar en la plantilla los archivos y/o carpetas necesarios para el correcto funcionamiento de la plantilla.

3.5.3.2.5. TAREA 16

	Tarea
Número de Tarea : 16	**Número de Historia :** 5/16
Nombre de Tarea : Plantilla de clase de control para PHP	
Tipo de Tarea: Investigación/Programación	**Puntos Estimados :**
Fecha de Inicio :	**Fecha Fin :**
Programador responsable : Eduardo Chávez R.	
Descripción: Optimizar la plantilla de clase de control de PHP. Indicar en la plantilla los tipos de datos, métodos y funciones necesarios para realizar la conexión y manipulación de tablas y campos. Indicar en la plantilla los archivos y/o carpetas necesarios para el correcto funcionamiento de la plantilla.	

3.5.3.2.6. TAREA 17

	Tarea
Número de Tarea : 17	**Número de Historia :** 5/17
Nombre de Tarea : Plantilla de clase de control para ASP	
Tipo de Tarea: Investigación/Programación	**Puntos Estimados :**
Fecha de Inicio :	**Fecha Fin :**
Programador responsable : Eduardo Chávez R.	
Descripción: Crear la plantilla de clase de control de ASP. Indicar en la plantilla los tipos de datos, métodos y funciones necesarios para realizar la conexión y manipulación de tablas y campos. Indicar en la plantilla los archivos y/o carpetas necesarios para el correcto funcionamiento de la plantilla.	

3.5.3.2.7. TAREA 18

Tarea	
Número de Tarea : 18	**Número de Historia :** 5/18
Nombre de Tarea : Plantilla de clase de control para JSP	
Tipo de Tarea: Investigación/Programación	**Puntos Estimados :**
Fecha de Inicio :	**Fecha Fin :**
Programador responsable : Eduardo Chávez R.	
Descripción: Crear la plantilla de clase de control de JSP. Indicar en la plantilla los tipos de datos, métodos y funciones necesarios para realizar la conexión y manipulación de tablas y campos. Indicar en la plantilla los archivos y/o carpetas necesarios para el correcto funcionamiento de la plantilla.	

3.5.3.3 TARJETAS C.R.C.

3.5.3.3.1 TARJETA “Access 2003 (ODBC).cct”

Access 2003 (ODBC).cct	

• Se crea el archivo de configuración para Access 2003, usando conexión ODBC.
• Código de compatibilidad de base de datos: MSACCESSO
• Se establecieron las cadenas de conexión y SQL para:
  o Conexión de Base de datos
  o Obtención de tablas
  o Obtención de columnas
	Programador

3.5.3.3.2 TARJETA “MySQL 5.x.cct”

MySQL 5.x.cct	
• Se crea el archivo de configuración para MySQL versiones 5.x. • Código de compatibilidad de base de	Programador

datos: MYSQLX • Se establecieron las cadenas de conexión y SQL para: o Conexión de Base de datos o Obtención de tablas o Obtención de columnas o Tipos de datos de columnas	

3.5.3.3.3 TARJETA "SQL Server (2005 Express).cct"

SQL Server (2005 Express).cct	
• Se crea el archivo de configuración para SQL Server Express 2005. • Código de compatibilidad de base de datos: MSSQL. • Este Script se probó además en bases de datos SQL Server 2000, dando compatibilidad en un 100% • Se establecieron las cadenas de conexión y SQL para: o Conexión de Base de datos o Obtención de tablas o Obtención de columnas o Tipos de datos de columnas	Programador

3.5.3.3.4 TARJETA "ACCESS 2003 (ODBC) - ASP"

ACCESS 2003 (ODBC) - ASP	
• Se crea el archivo de configuración para ASP con base de datos Access 2003 (ODBC). • Código de compatibilidad de base de datos: MSACCESSO. • Código de compatibilidad de clases de gestión: MSACCESSO_ASP • Archivos de plantilla: o CLASS_ODBC_120807.asp o DBINFO.asp • Se establecieron los valores para tipos de datos y valores predefinidos para cada columna encontrada.	Programador

3.5.3.3.5 TARJETA “ACCESS 2003 (ODBC) - JSP”

ACCESS 2003 (ODBC) - JSP	
• Se crea el archivo de configuración para JSP con base de datos Access 2003 (ODBC). • Código de compatibilidad de base de datos: MSACCESSO. • Código de compatibilidad de clases de gestión: MSACCESSO_JSP • Archivos de plantilla: o CLASS_ODBC_120807.java o DBINFO.JSP o COMPILE.BA • Se establecieron los valores para tipos de datos y valores predefinidos para cada columna encontrada.	Programador

3.5.3.3.6 TARJETA “ACCESS 2003 (ODBC) - PHP”

ACCESS 2003 (ODBC) - PHP	
• Se crea el archivo de configuración para PHP con base de datos Access 2003 (ODBC). • Código de compatibilidad de base de datos: MSACCESSO. • Código de compatibilidad de clases de gestión: MSACCESSO_PHP • Archivos de plantilla: o CLASS_ODBC_120806.php o DBINFO.php • Se establecieron los valores para tipos de datos y valores predefinidos para cada columna encontrada.	Programador

3.5.3.3.7 TARJETA “MySQL - ASP”

MySQL - ASP	
• Se crea el archivo de configuración para ASP con base de datos MySQL 5.x. • Código de compatibilidad de base de datos: MYSQLX. • Código de compatibilidad de clases de gestión: MYSQLX_ASP • Archivos de plantilla: o CLASS_ODBC_120807.asp o DBINFO.asp • Se establecieron los valores para tipos de datos y valores predefinidos para cada columna encontrada.	Programador

3.5.3.3.8 TARJETA “MySQL - JSP”

MySQL - JSP	
• Se crea el archivo de configuración para JSP con base de datos MySQL 5.x. • Código de compatibilidad de base de datos: MYSQLX. • Código de compatibilidad de clases de gestión: MYSQLX_JSP • Archivos de plantilla: o CLASS_MYSQL_120807.java o DBINFO.JSP o COMPILE.BA • Se establecieron los valores para tipos de datos y valores predefinidos para cada columna encontrada.	Programador

3.5.3.3.9 TARJETA “MySQL - PHP”

MySQL – PHP	
• Se crea el archivo de configuración para PHP con base de datos MySQL 5.x. • Código de compatibilidad de base de datos: MYSQLX. • Código de compatibilidad de clases de	Programador

gestión: MYSQLX_PHP • Archivos de plantilla: o CLASS_MYSQL_120807.php o DBINFO.php • Se establecieron los valores para tipos de datos y valores predefinidos para cada columna encontrada.	

3.5.3.3.10 TARJETA "SQL SERVER - ASP"

SQL SERVER - ASP	
• Se crea el archivo de configuración para ASP con base de datos SQL Server 2005 • Código de compatibilidad de base de datos: MSSQL. • Código de compatibilidad de clases de gestión: SQLSERVER_ASP • Archivos de plantilla: o CLASS_ODBC_120807.asp o DBINFO.asp • Se establecieron los valores para tipos de datos y valores predefinidos para cada columna encontrada.	Programador

3.5.3.3.11 TARJETA "SQL SERVER - JSP"

SQL SERVER - JSP	
• Se crea el archivo de configuración para JSP con base de datos SQL Server 2005 • Código de compatibilidad de base de datos: MSSQL. • Código de compatibilidad de clases de gestión: SQLSERVER_JSP • Archivos de plantilla: o CLASS_MSSQL_120807.java o DBINFO.jsp o COMPILE.BA • Se establecieron los valores para tipos de datos y valores predefinidos para cada columna encontrada.	Programador

3.5.3.3.12 TARJETA “SQL SERVER - PHP”

SQL SERVER - PHP	
• Se crea el archivo de configuración para PHP con base de datos SQL Server 2005 • Código de compatibislidad de base de datos: MSSQL. • Código de compatibilidad de clases de gestión: SQLSERVER_PHP • Archivos de plantilla: o CLASS_ODBC_120806.php o DBINFO.php • Se establecieron los valores para tipos de datos y valores predefinidos para cada columna encontrada.	Programador

3.5.3.3.13 TARJETA “ASP Gestion”

ASP Gestion	
• Se crea el archivo de configuración y plantillas para gestión de información usando lenguaje ASP. • Código de compatibilidad clases de gestión: ASP_CLASS. • Código de compatibilidad de plantillas de gestión: PAN_BAS_GES_ASP_001 • Archivos de plantilla: o ASP_PAGE_080516.asp o INDEX.asp • Se establecieron los valores para: o Funciones de entrada o Funciones de salida o Controles o Funciones de validación por tipo de dato • La plantilla es compatible para todas las clases de control de tipo ASP, indistintamente de la base de datos que manejen.	Programador

3.5.3.3.14 TARJETA “JSP Gestion”

JSP Gestion	
• Se crea el archivo de configuración y plantillas para gestión de información usando lenguaje JSP. • Código de compatibilidad clases de gestión: JSP_CLASS. • Código de compatibilidad de plantillas de gestión: PAN_BAS_GES_JSP_001 • Archivos de plantilla: o INDEX.html o JSP_PAGE_12080601.jsp • Se establecieron los valores para: o Funciones de entrada o Funciones de salida o Controles o Funciones de validación por tipo de dato • La plantilla es compatible para todas las clases de control de tipo JSP, indistintamente de la base de datos que manejen.	Programador

3.5.3.3.15 TARJETA “PHP Gestion”

PHP Gestion	
• Se crea el archivo de configuración y plantillas para gestión de información usando lenguaje PHP. • Código de compatibilidad clases de gestión: PHP_CLASS. • Código de compatibilidad de plantillas de gestión: PAN_BAS_GES_PHP_001 • Archivos de plantilla: o PHP_PAGE_080512.php o INDEX.php • Se establecieron los valores para: o Funciones de entrada o Funciones de salida o Controles o Validación por tipo de dato • La plantilla es compatible para todas las clases de control de tipo PHP,	Programador

indistintamente de la base de datos que manejen.	

3.5.3.3.16 TARJETA “PHPc Gestion”

PHPc Gestion	

• Caso especial de plantilla con valores agregados como:
  o HTML5
  o JavaScript
  o AJAX
  o CSS
  o Reporting PDF
  o Controles interactivos jQuery
  o Menús despegables
  o Pantalla de Login de usuario
  o Control de Logs
• Esta plantilla se utilizará en el proyecto “rSNAP” el cual es la comprobación de funcionalidad de “CreaCod”.
• Se crea el archivo de configuración y plantillas para gestión de información usando lenguaje PHP.
• Código de compatibilidad clases de gestión: PHP_CLASS.
• Código de compatibilidad de plantillas de gestión: PAN_BAS_GES_PHP_001
• Archivos de plantilla:
  o PHP_PAGE_12071801.php
  o toolbar.php
  o INDEX.php
• El proyecto requiere de algunos archivos de configuración adicionales, los cuales se copian directamente y se encuentran ubicados en las carpetas:
  o data
  o gestion
  o images
  o system
• Se establecieron los valores para:
  o Funciones de entrada
  o Funciones de salida
  o Controles
  o Funciones de validación por tipo de dato usando AJAX
• La plantilla es compatible para todas las clases de control de tipo PHP, sin embargo no se realizan pruebas ya que el proyecto “rSNAP” funciona
	Programador

exclusivamente con la base de datos MySQL para la cual se hicieron todas las pruebas necesarias.	

3.5.3.4 INFORMACIÓN

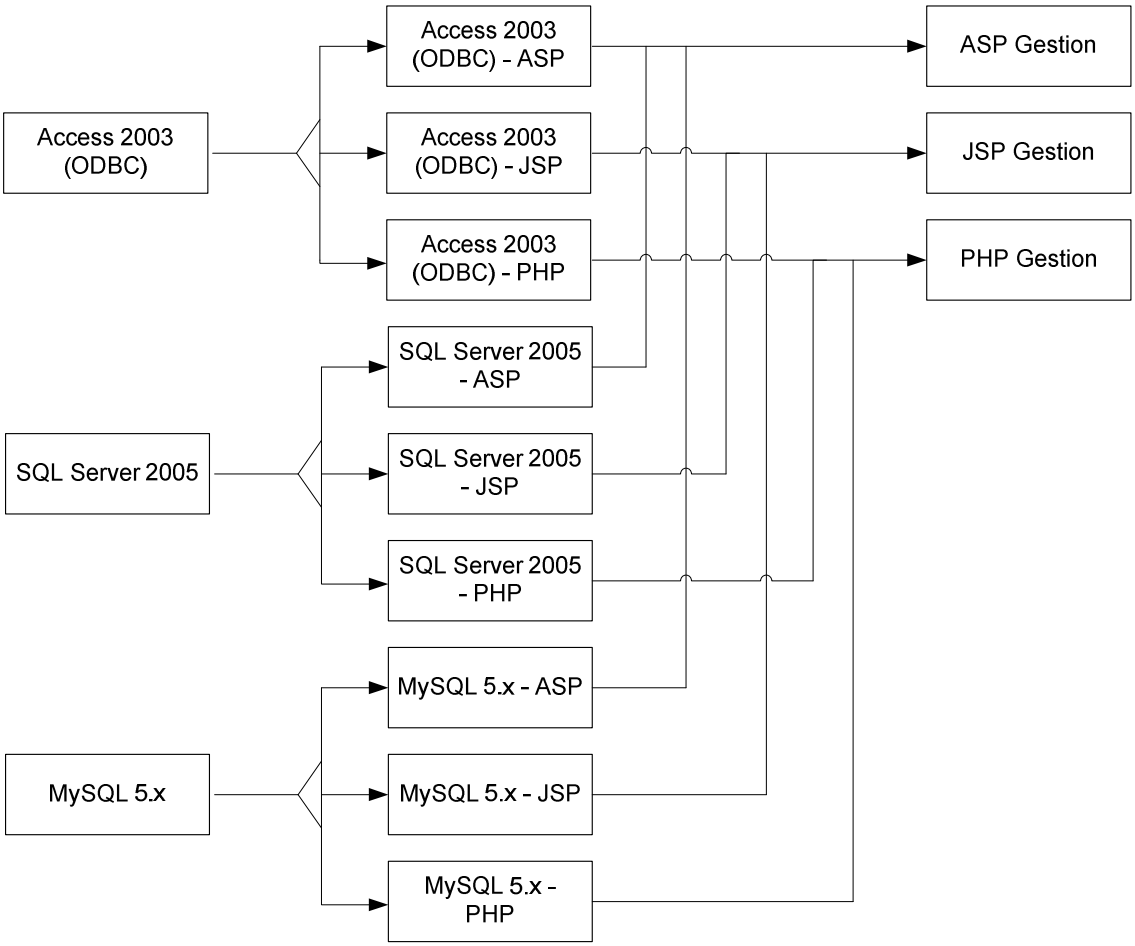


Figura 3.18. Diagrama de plantillas de Base de datos, clases de control y gestión GUI

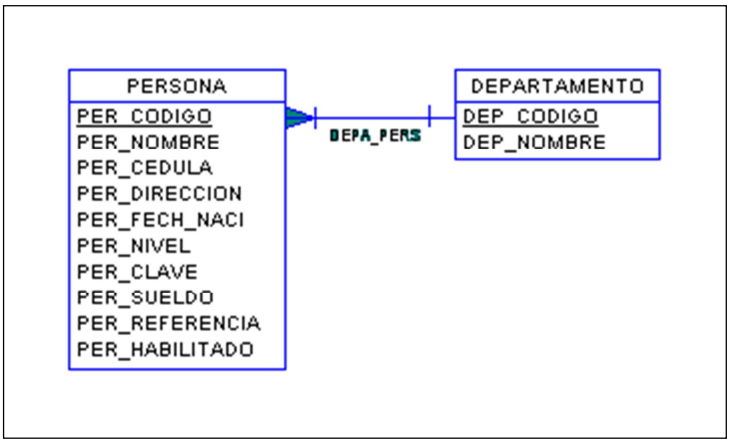


Figura 3.19. Base de datos de pruebas: “persona”

3.5.3.5 PANTALLAS

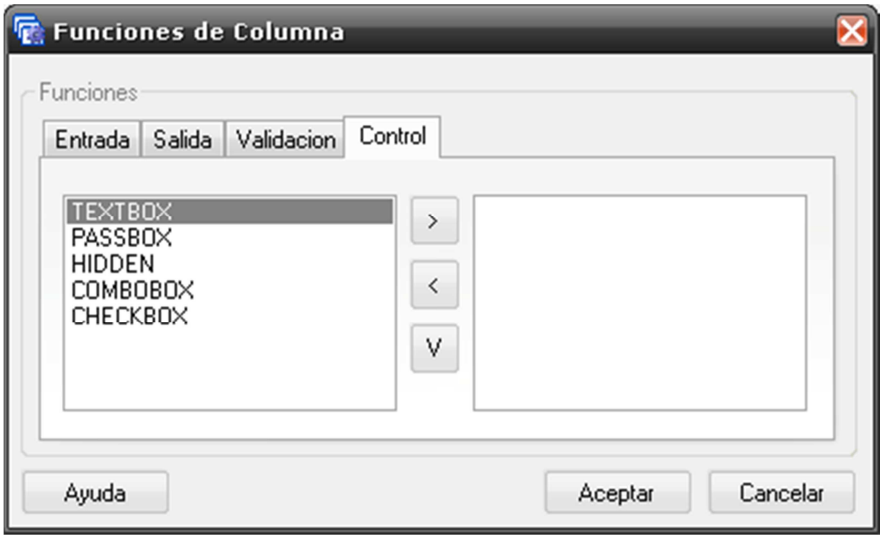


Figura 3.20. Configuración de plantilla de gestión GUI.

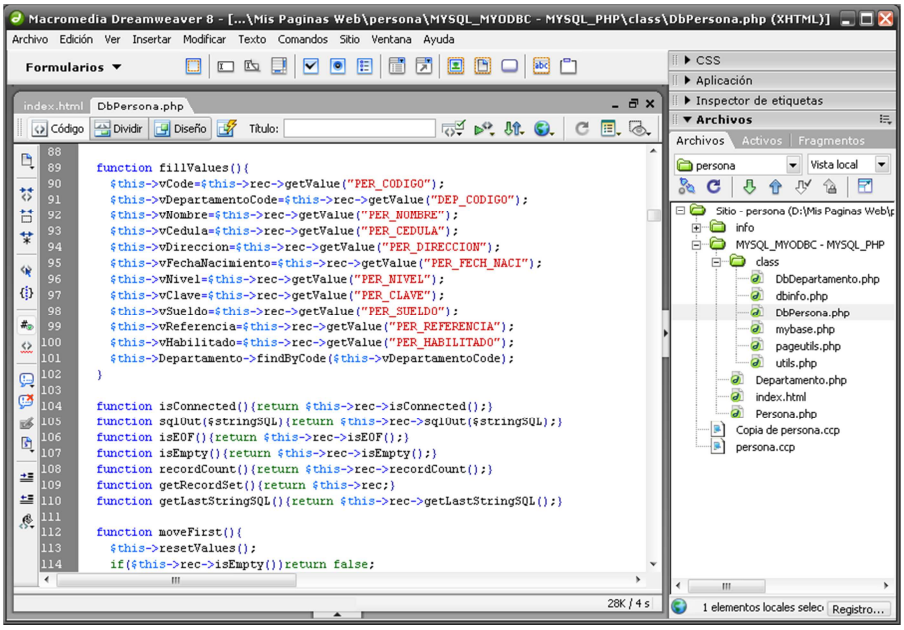


Figura 3.21. Captura de Dreamweaver abriendo archivo de clase "DbPersona.php"

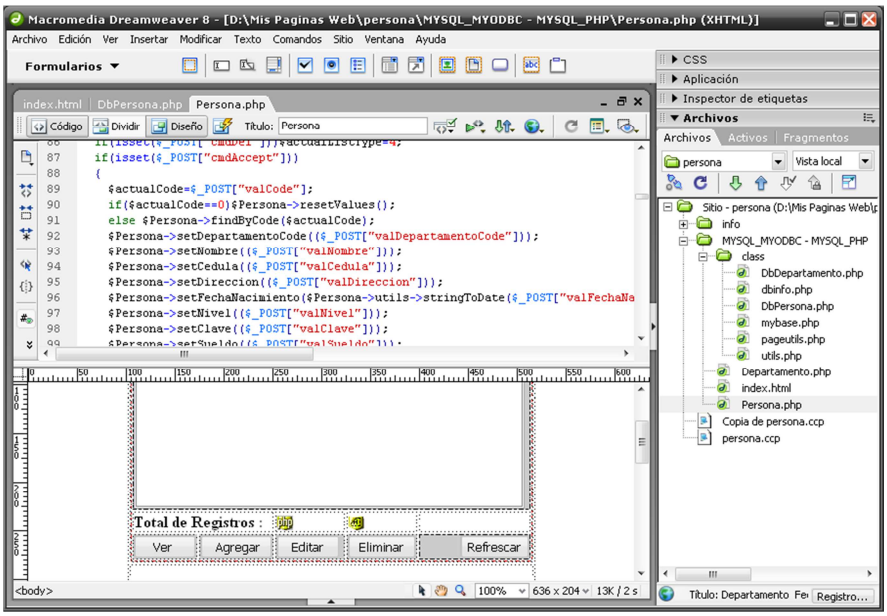


Figura 3.22. Captura de Dreamweaver abriendo archivo de gestión GUI "persona.php"

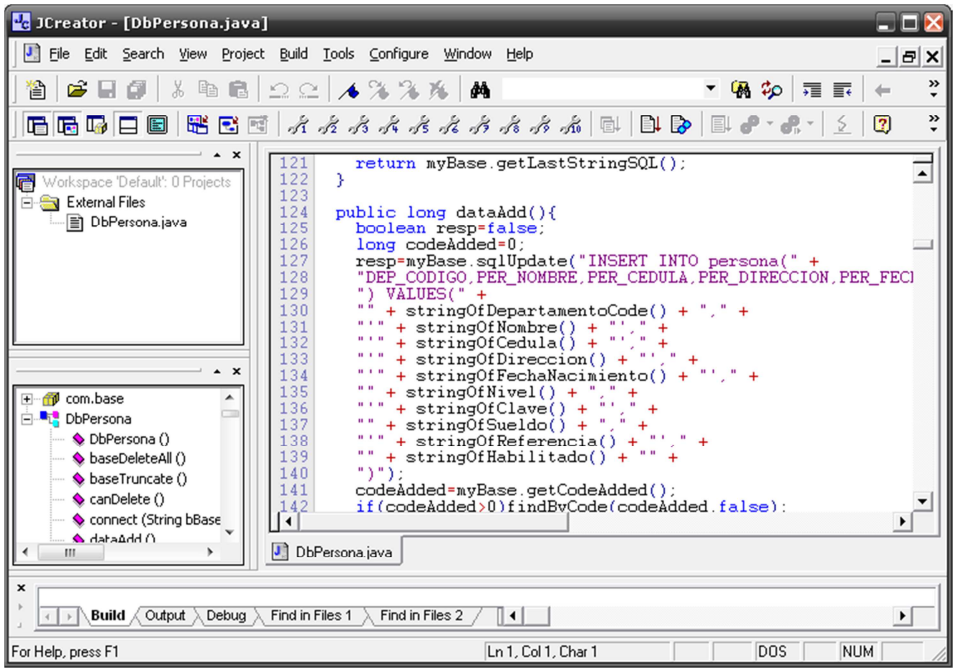


Figura 3.23. Captura de Jcreator abriendo archivo de clase “DbPersona.java”

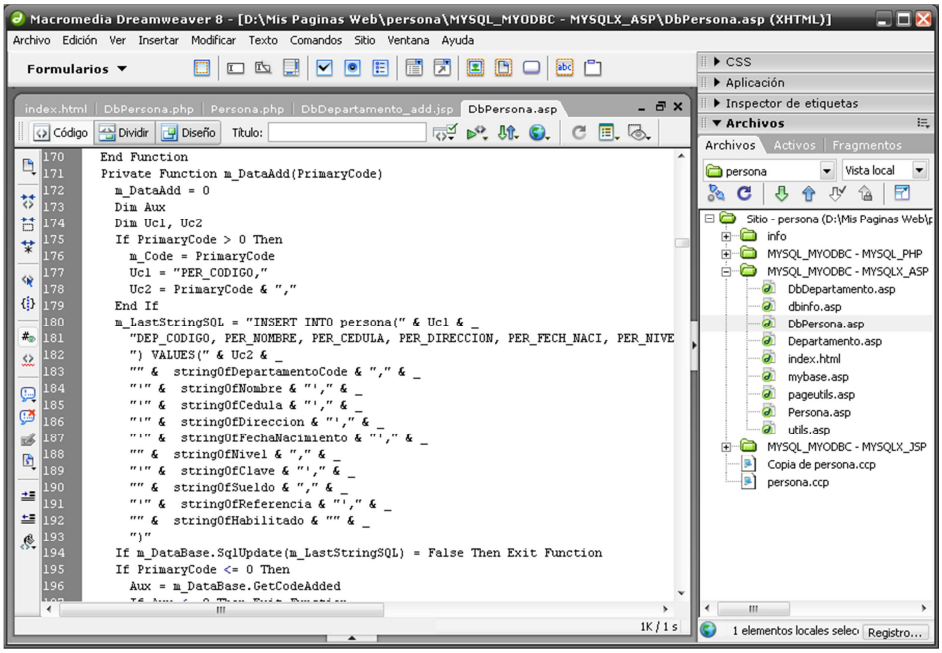


Figura 3.24. Captura de Dreamweaver abriendo archivo de clase “DbPersona.asp”

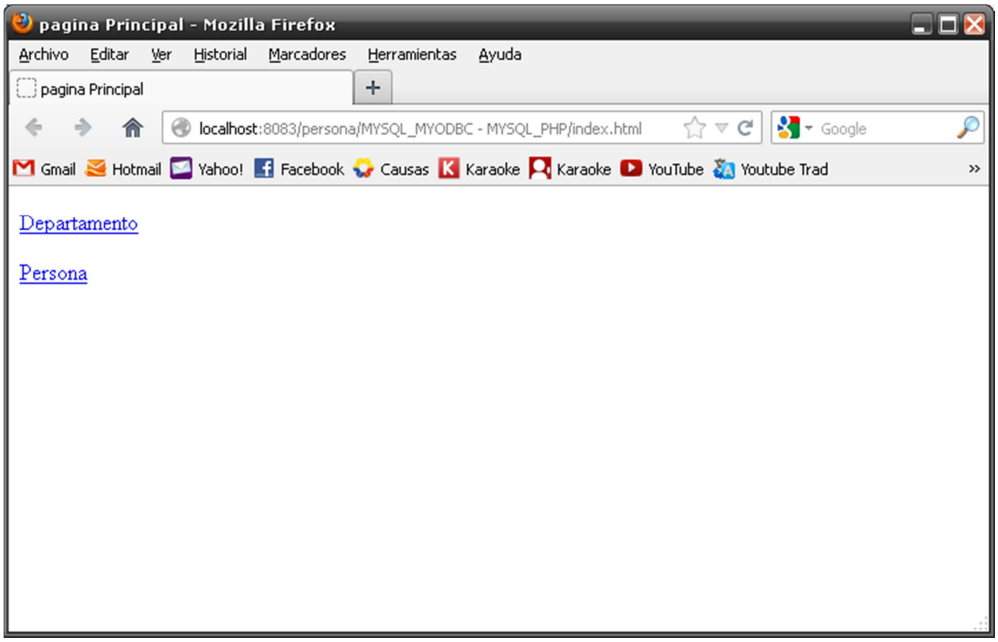


Figura 3.25. Resultado de proyecto: Página index.html

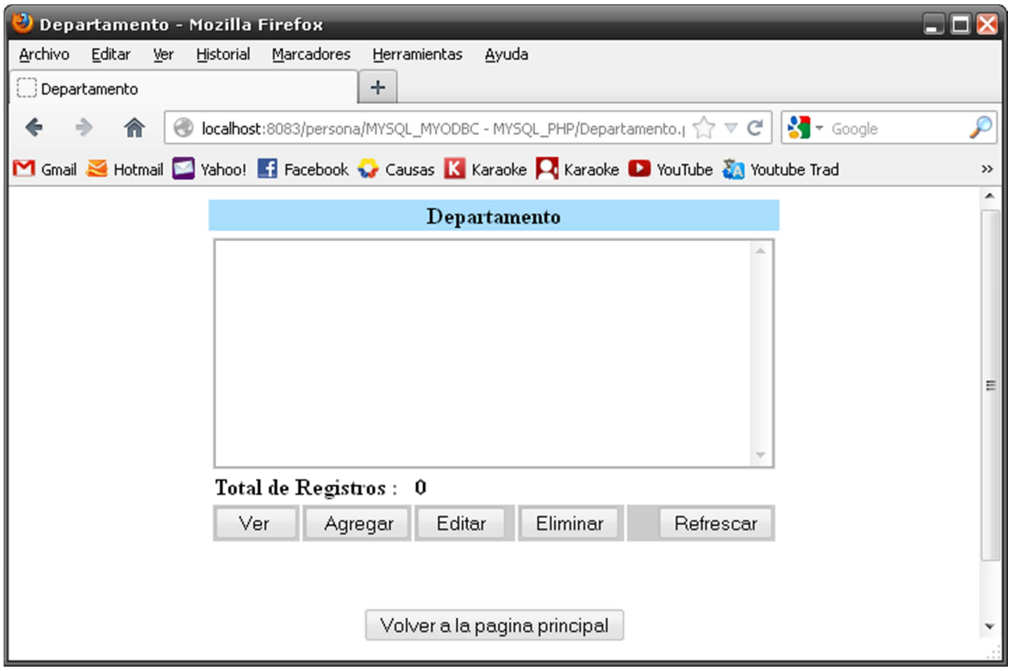


Figura 3.26. Resultado de proyecto: Página departamento.php, ver datos

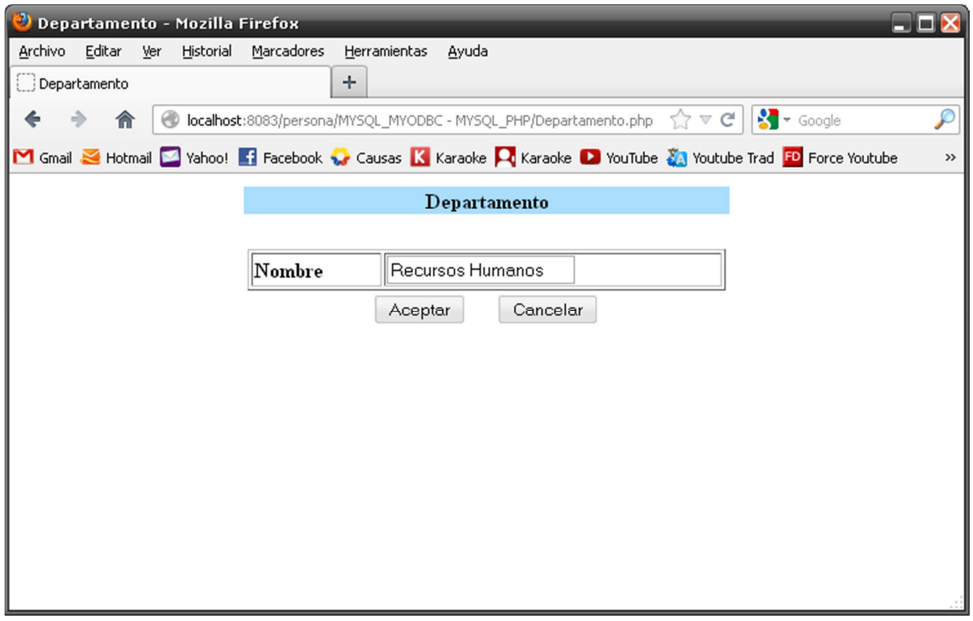


Figura 3.27. Resultado de proyecto: departamento.php, agregar registro

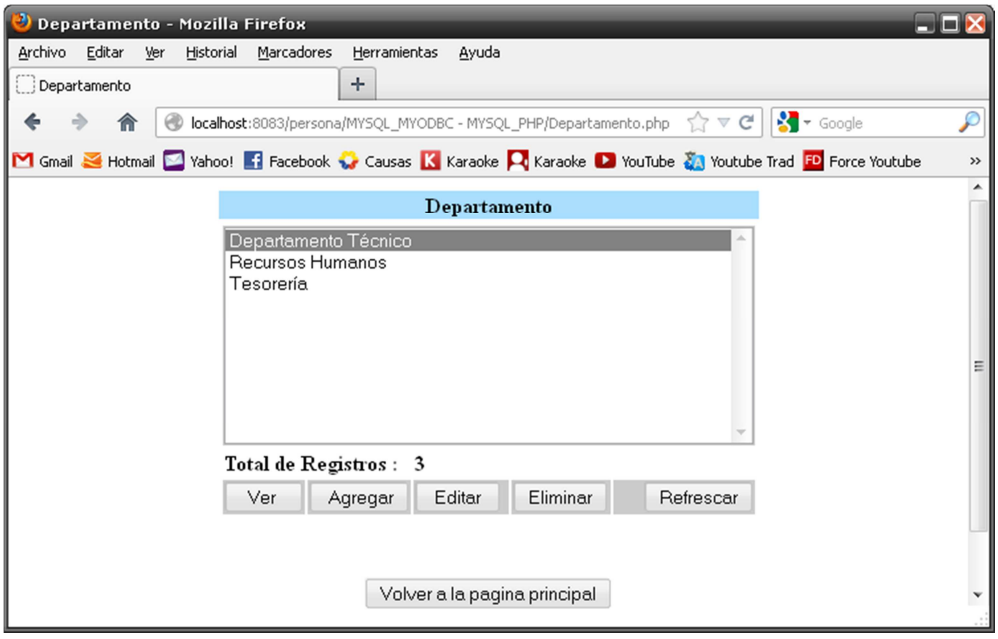


Figura 3.28. Resultado de proyecto: departamento.php con nuevos registros

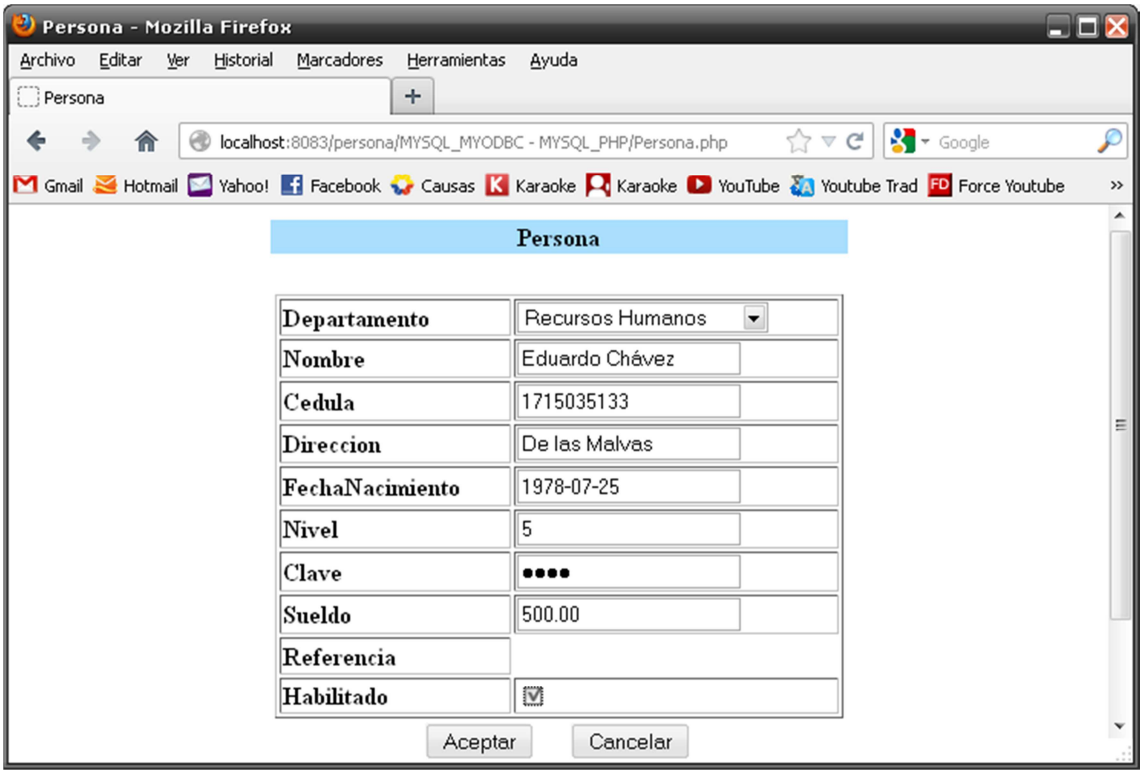


Figura 3.29. Resultado de proyecto: Agregando registro a persona.php

3.5.3.6. INCIDENCIAS

3.5.3.6.1. TAREA 12

- Se creó la plantilla de gestión GUI para PHP, la cual fue desarrollada únicamente con lenguaje HTML y, para validaciones, con JavaScript.
- Se decidió desarrollar una página básica debido a que se pretende demostrar la efectividad del programa “CreaCod” para generar código fuente.

- Esta plantilla se utilizó como molde para desarrollar el resto de plantillas (ASP y JSP) por lo que la estructura de las 3 plantillas es similar.
- Se creó además una plantilla especial denominada “PHPc Gestion” la cual se usará en el sistema “rSNAP” y dicha plantilla cuenta con:
  - HTML5
  - JavaScript
  - AJAX
  - CSS
  - Reporting PDF
  - Controles interactivos jQuery
  - Menús despegables
  - Pantalla de ingreso de usuario
  - Control de Logs
- Esta plantilla servirá para demostrar todo el potencial de “CreaCod” para realizar código fuente de alto desempeño y sólo se la ha probado con la base de datos MySQL ya que dicha base es el requisito en el sistema “rSNAP”.

3.5.3.6.2. TAREA 13

- La plantilla para conectar la base de datos MySQL se realizó sin contratiempos y será usada como molde para el resto de plantillas.

3.5.3.6.3. TAREA 14

- La plantilla de datos Access tiene el inconveniente de no poder recibir, a través de SQL, los tipos de datos de las columnas. Este incidente fue investigado y no se encontró una solución efectiva para el incidente.

3.5.3.6.4. TAREA 15

- La plantilla para conectar la base de datos SQL Server se realizó sin contratiempos y será usada como molde para el resto de plantillas.
- Cabe aclarar que se deben configurar ciertos parámetros de SQL Server para que se pueda leer la información a través de HTML y cualquier lenguaje de programación (PHP, JSP y ASP).

3.5.3.6.5. TAREA 16

- Se depuró la plantilla de clase para PHP.
- Se agregaron algunas funciones necesarias para la correcta comunicación con la plantilla de gestión GUI.
- A partir de esta plantilla se generaron las demás plantillas para los otros lenguajes de programación (JSP y ASP) por lo que las funciones y métodos son similares entre los 3 lenguajes.

3.5.3.6.6. TAREA 17

- Esta plantilla fue realizada en base a la plantilla de clase PHP.
- Se debe considerar que los archivos ASP únicamente funcionan a través del servidor Web IIS.
- La plantilla funciona sin inconvenientes.

3.5.3.6.7. TAREA 18

- Esta plantilla fue realizada en base a la plantilla de clase PHP.
- Se debe considerar que los archivos ASP únicamente funcionan a través del servidor Web IIS.
- La plantilla funciona sin inconvenientes.

3.5.4. ITERACIÓN 4

3.5.4.1. HISTORIAS ADICIONALES

3.5.4.1.1. HISTORIA 19 (H19)

Historia de Programador	
Número: 19	**Usuario :** Programador
Nombre : Datos de Tablas y Columnas	
Prioridad : ALTO	**Riesgo en Desarrollo:** ALTO
Puntos Estimados : 9	**Iteración asignada :** 4
Programador responsable : Eduardo Chávez R.	
Descripción: Las plantillas deben ser capaces de repetir código para cada tabla y columna	

Observaciones: Buscar la manera que la plantilla pueda repetir código fuente.

3.5.4.2. TAREAS

3.5.4.2.1. TAREA 19

	Tarea
Número de Tarea : 19	**Número de Historia :** 6/19
Nombre de Tarea : Datos redundantes de Tablas y Columnas	
Tipo de Tarea: Investigación/Programación	**Puntos Estimados :**
Fecha de Inicio :	**Fecha Fin :**
Programador responsable : Eduardo Chávez R.	
Descripción: Buscar una alternativa para que el código fuente en las plantillas se pueda repetir para cada tabla y columna.	

3.5.4.3. TARJETAS C.R.C.

3.5.4.3.1. TARJETA “ModMakeCode”

ModMakeCode	
• Se analiza la posibilidad que las plantillas repitan la información de código fuente, sin embargo esta nueva característica cambia el planteamiento total del modelo de desarrollo de plantillas y código fuente. • Se decide crear un módulo externo a InfoProject, InfoTable, InfoColumn e InfoParam para realizar el proceso de generación de código fuente repetitivo. • Este módulo ahora es capaz de recibir los comandos: o {BEGIN} y {END} para bucles de datos	Programador

- o {OL_BEGIN} y {OL_END} para bucles de datos en una misma línea - o {COUNTER} para contadores - En el caso de InfoProject, los bucles repiten la información para cada objeto de InfoTable. - En el caso de InfoTable, los bucles repiten la información para cada objeto de InfoColumn/InfoParam. - Este módulo va cambiando, nivel por nivel,los valores de: - o BASE.- Ámbito de información de DB. - o CLASS.- Ámbito de clases - o TEMPLATE.- Ámbito de gestión GUI. - El módulo además cambia los valores de: - o {COUNTER} - o {DATE} - o {TABLE_COLVISFUN} - o {FUNSEP} - o {FUNCTION_INPUT} - o {FUNCTION_OUTPUT} - o {FUNCTION_VALIDATION} - o {FUNCTION_VARS} - o {CONTROL_[CONTROL]} - o {NOCONTROL_[CONTROL]} - o {DATATYPE_[TYPE]} - o {NODATATYPE_[TYPE]} - o {COMMA} - El resto de cambios los siguen haciendo las clases asociadas.	

3.5.4.4. INFORMACIÓN

Código fuente 3.1. Ejemplo de código {BEGIN} / {END}

```
class {TABLE_NAMCLS}
{
  var $tableClassName;
  var $tableClassObject;
  var $tableClassTitle;
  var $utils;
```

```
  var $rec;
{BEGIN}
  var ${COLUMN_CLSOBJ};
{END}
……
```

Código fuente 3.2. Resultado de {BEGIN} / {END}

```
class DbDepartamento
{
  var $tableClassName;
  var $tableClassObject;
  var $tableClassTitle;
  var $utils;

  var $rec;
  var $vCode;
  var $colOrder;
  var $colView;
  var $vNombre;
…
```

3.5.4.5. INCIDENCIAS

3.5.4.5.1. TAREA 19

- El módulo de generación de código fuente ya fue creado en la historia 5, debido a que se necesitaba ver los resultados de las plantillas.
- El código fuente no era óptimo debido a que no había un método de repetición de código fuente de plantillas, por lo que se decidió crear un nuevo módulo que controle bucles. Por esta razón se rediseñaron todas las plantillas para adecuarlas al nuevo modelo de código, se obtuvo resultados satisfactorios y el código de las plantillas es mucho más limpio.

- Se notó que, además, se necesitaba un “contador” de numeración. En casos como los controles de Visual Basic, estos contadores son óptimos para saber qué posición de la pantalla (x,y) será ubicado dicho control.
- Se depuró código fuente de las plantillas de gestión GUI para que sean más amigables al usuario.

3.5.5. ITERACIÓN 5

3.5.5.1. HISTORIAS ADICIONALES

3.5.5.1.1. HISTORIA 20

Historia de Programador	
Número: 20	**Usuario :** Programador
Nombre : Idioma	
Prioridad : BAJO	**Riesgo en Desarrollo:** BAJO
Puntos Estimados : 3	**Iteración asignada :** 5
Programador responsable : Eduardo Chávez R.	
Descripción: El sistema debe cambiar de idioma a español, inglés y otro idioma de gusto del programador	
Observaciones:	

3.5.5.2. TAREAS

3.5.5.2.1. TAREA 20

	Tarea
Número de Tarea : 20	**Número de Historia :** 7
Nombre de Tarea : Plantillas faltantes	
Tipo de Tarea: Investigación/Programación	**Puntos Estimados :**
Fecha de Inicio :	**Fecha Fin :**
Programador responsable : Eduardo Chávez R.	
Descripción: Investigar y desarrollar las plantillas para Access, SQL Server y MySQL. Investigar y desarrollar las plantillas para ASP, JSP y PHP	

3.5.5.2.2. TAREA 21

	Tarea
Número de Tarea : 21	**Número de Historia :** 20
Nombre de Tarea : Idioma CreaCod	
Tipo de Tarea: Investigación/Programación	**Puntos Estimados :**
Fecha de Inicio :	**Fecha Fin :**
Programador responsable : Eduardo Chávez R.	
Descripción: Investigar y desarrollar e método para cambiar de idioma a CreaCod.	

3.5.5.3. TARJETAS C.R.C.

3.5.5.3.1. TARJETA “SysLanguaje”

SysLanguaje	
• Se crea la clase SysLanguaje, con la cual se controla todos los controles del sistema para cambiarlos de idioma. • La clase, al arrancar el programa, revisa la carpeta “LANGUAJE” para localizar los idiomas aceptados por CreaCod. • La extensión para los archivos de idioma es “ccl” (CreaCod Languaje). • Todos los formularios y mensajes se comunican con SysLanguaje antes de ser	Programador

presentados a pantalla. • Cada uno de los valores de SysLanguaje son de tipo “texto”. • Los archivos de idioma son fáciles de interpretar y pueden ser traducidos, incluso, con herramientas comunes de traducción. • Se crearon idiomas para: o Español (nativo de CreaCod) o Inglés (archivo ccl) o Francés (archivo ccl)	

3.5.5.4. PANTALLAS

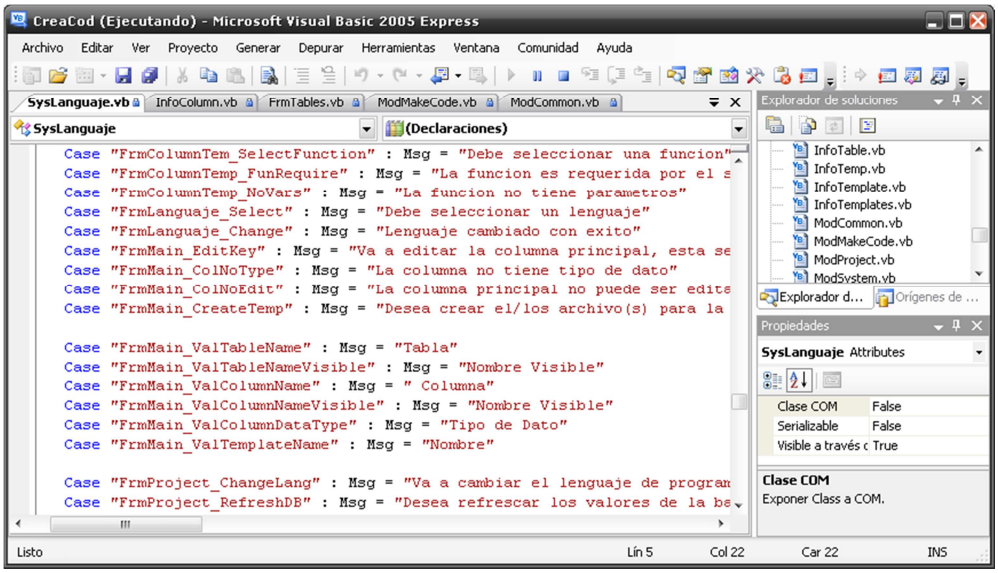


Figura 3.30. Clase “SysLanguaje” con “Español” como idioma nativo

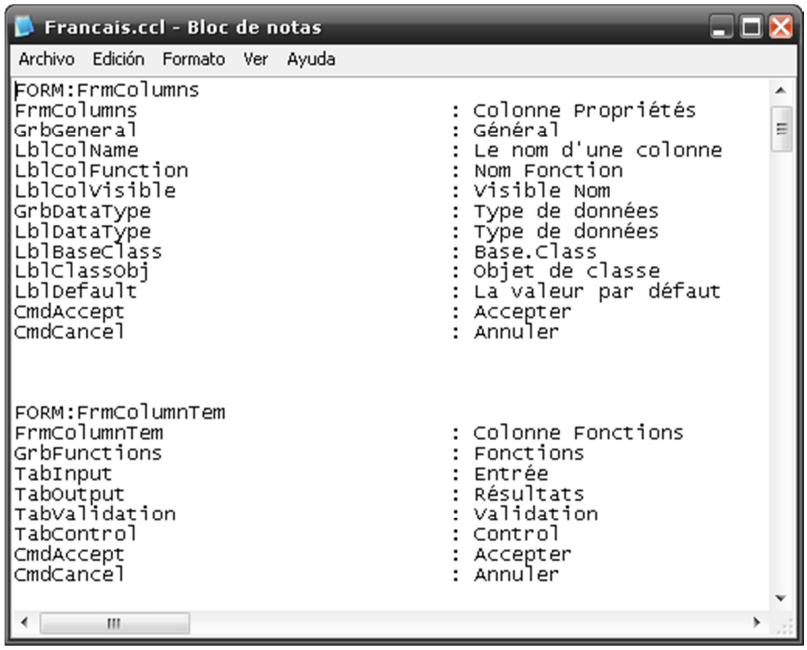


Figura 3.31. Ejemplo de archivo de idioma “Francais.ccl”

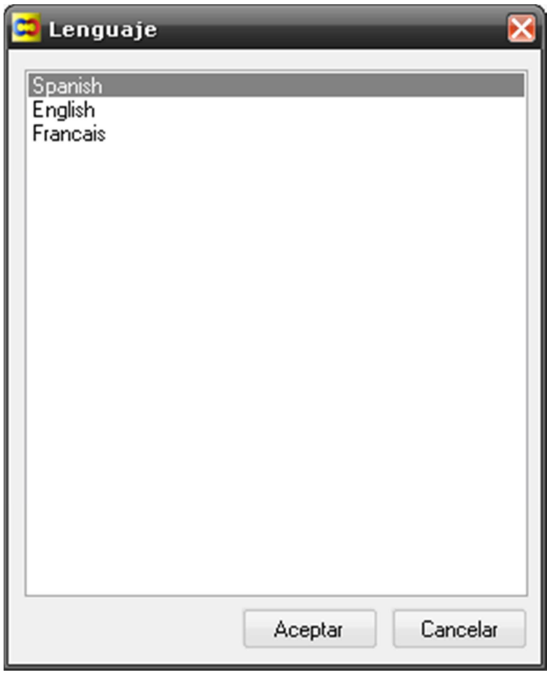


Figura 3.32. Pantalla de selección de idioma

Figura 3.33. CreaCod en idioma francés

3.5.5.5. INCIDENCIAS

3.5.5.5.1. TAREA 20

- Las plantillas para base de datos, clases de control y gestión GUI ya fueron desarrolladas en tareas anteriores, por lo que la historia de usuario esta lista.

3.5.5.5.2. TAREA 21

- Se creó una nueva clase, la cual controla directamente el idioma de cada uno de los formularios y mensajes que se presenten en CreaCod.

- Se decidió al idioma francés como lenguaje alternativo a español e inglés.
- El módulo está terminado y es funcional al 100%.
- Se crearon archivos de idiomas, los cuales se encuentran en la carpeta “LANGUAJE”.

3.5.6. ITERACIÓN 6

3.5.6.1. HISTORIAS ADICIONALES

3.5.6.1.1. HISTORIA 21

Historia de Programador	
Número: 21	**Usuario :** Programador
Nombre : Manual técnico de funciones	
Prioridad : BAJO	**Riesgo en Desarrollo:** BAJO
Puntos Estimados : 5	**Iteración asignada :** 6
Programador responsable : Eduardo Chávez R.	
Descripción: El sistema debe tener un manual técnico de la estructura de los archivos “cct” y las plantillas de código fuente.	
Observaciones:	

3.5.6.1.2. HISTORIA 22

Historia de Programador	
Número: 22	**Usuario :** Programador
Nombre : Botones de ayuda	
Prioridad : BAJO	**Riesgo en Desarrollo:** BAJO
Puntos Estimados : 3	**Iteración asignada :** 6
Programador responsable : Eduardo Chávez R.	
Descripción: El sistema debe tener el botón de ayuda en cada una de sus pantallas, los	

cuales redireccionarán al tema en el archivo de ayuda.
Observaciones:

3.5.6.2. TAREAS

3.5.6.2.1. TAREA 23

	Tarea
Número de Tarea : 23	**Número de Historia :** 9
Nombre de Tarea : Manual de Usuario	
Tipo de Tarea: Documentación	**Puntos Estimados :**
Fecha de Inicio :	**Fecha Fin :**
Programador responsable : Eduardo Chávez R.	
Descripción: Crear el archivo de ayuda de CreaCod, en formato “chm” y, de ser posible, formato web.	

3.5.6.2.2. TAREA 24

	Tarea
Número de Tarea : 24	**Número de Historia :** 21
Nombre de Tarea : Manual de Usuario	
Tipo de Tarea: Documentación	**Puntos Estimados :**
Fecha de Inicio :	**Fecha Fin :**
Programador responsable : Eduardo Chávez R.	
Descripción: Crear el manual técnico de la estructura de los archivos “cct” y las plantillas de código fuente.	

3.5.6.2.3. TAREA 25

	Tarea
Número de Tarea : 25	**Número de Historia :** 22
Nombre de Tarea : Botones de Ayuda	
Tipo de Tarea: Documentación	**Puntos Estimados :**
Fecha de Inicio :	**Fecha Fin :**
Programador responsable : Eduardo Chávez R.	
Descripción: Botones de ayuda en cada una de las pantallas de CreaCod.	

3.5.6.3. TARJETAS C.R.C.

3.5.6.3.1. TARJETA “creacod.doc/chm”

creacod.doc/chm	
• Se documenta toda la información del programa CreaCod: o Breve introducción de CreaCod, sus ventajas y beneficios. o Capturas de pantalla. o Descripción de cada uno de los controles. o Especificación del uso de cada uno de los comandos del programa. o Ejemplos para cada uno de los casos. o Índices de ayuda para los botones que formen parte de los formularios de CreaCod.	Programador

3.5.6.3.2. TARJETA “creacod.doc/chm técnico”

creacod.doc/chm técnico	
• Se documenta toda la información del método de creación de archivos “cct” y de los comandos de plantillas de CreaCod: o Descripción de los parámetros de proyecto. o Descripción de los parámetros de tablas. o Descripción de los parámetros de columnas. o Descripción de los parámetros de plantillas de clases. o Descripción de los parámetros de plantillas de gestión GUI. o Descripción de partes de los archivos “cct”. o Descripción de cada uno de los comandos de CreaCod. o Especificación de tipos de datos o Especificación de sintaxis o Detalles de los comandos o Ejemplo de código de plantilla o Ejemplo de código generado o Datos de “Vea también”.	Programador

3.5.6.3.3. TARJETA “Botones de ayuda”

Botones de ayuda	
• Se compila el archivo de ayuda “chm” y se investiga el método de lectura de este tipo de archivos en Visual Basic. • Se crean índices en el archivo de ayuda “chm” para que cada botón de CreaCod apunte a la ayuda requerida. • Se habilitan los botones de ayuda y se especifican los índices del archivo “chm”.	Programador

3.5.6.4. PANTALLAS

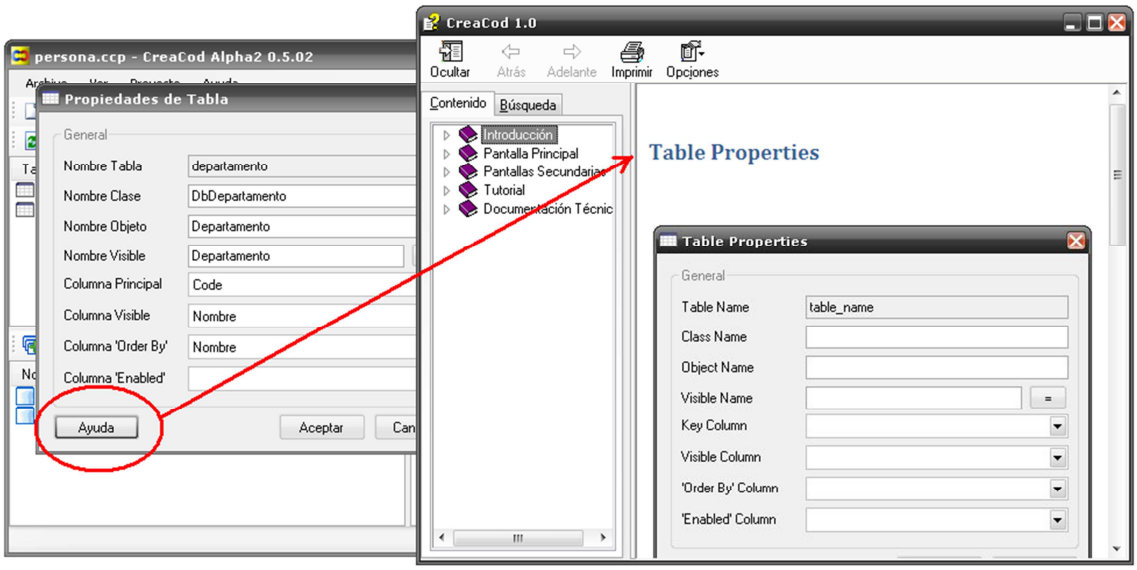


Figura 3.34. Llamada al archivo de ayuda desde los botones de CreaCod

3.5.6.5. INCIDENCIAS

3.5.6.5.1. TAREA 23

- Se creó el archivo de ayuda de CreaCod.
- El archivo de ayuda se creó a partir de la última versión estable de CreaCod.

3.5.6.5.2. TAREA 24

- Se creó el archivo de “Manual Técnico”.

- El manual técnico se lo adjuntó al archivo de ayuda para integrar la documentación en un mismo archivo.
- El archivo de “Manual Técnico” se lo desarrollo a partir de la última versión estable de las clases de control y plantillas de gestión GUI.

3.5.6.5.3. TAREA 25

- Se activaron todos los botones de ayuda del sistema CreaCod.
- Se utilizó la función “Help” nativa de Visual Basic para presentar la información de ayuda de CreaCod, la cual se conecta directamente con el archivo “creacod.chm”.

3.5.7. ESPECIFICACIÓN DE CASOS DE USO

Los presentes casos de uso están enfocados al usuario final del sistema, que desde este momento se lo denominará “Programador”.

- CU-CREACOD-01: Datos iniciales de proyecto.
- CU-CREACOD-02: Seleccionar base de datos.
- CU-CREACOD-02.1: Conectar base de datos.
- CU-CREACOD-03: Seleccionar lenguaje de programación.
- CU-CREACOD-04: Refrescar información de base de datos.
- CU-CREACOD-05: Parámetros de tablas.
- CU-CREACOD-06: Parámetros de campos.

- CU-CREACOD-07: Parámetros de plantilla de gestión GUI.
- CU-CREACOD-08: Generar código fuente.
- CU-CREACOD-09: Cambiar idioma del IDE de CreaCod.

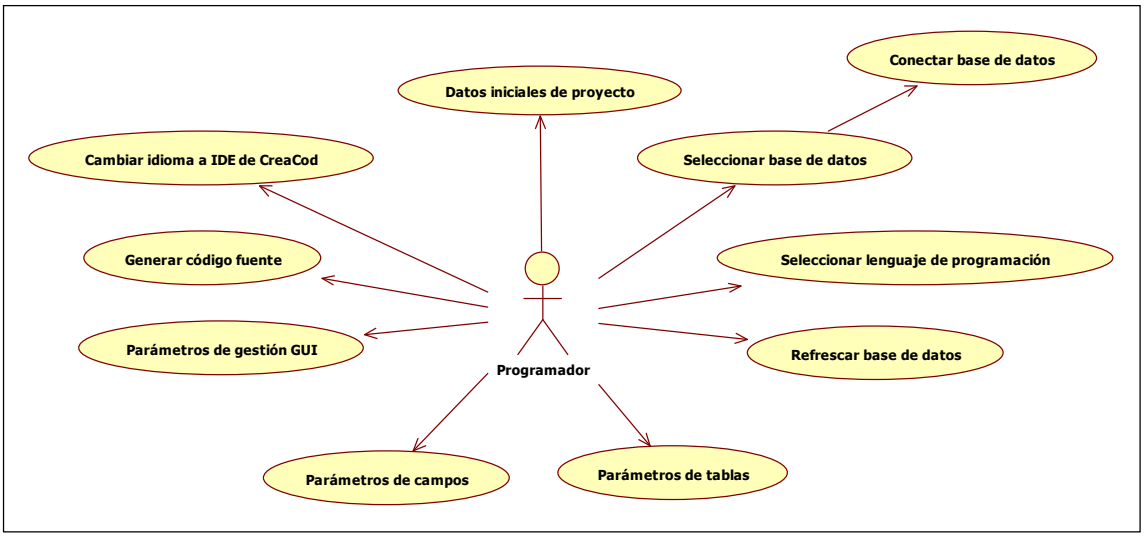


Figura 3.35. Casos de uso de "Programador"

3.5.8. CASOS DE USO DE USUARIO "PROGRAMADOR"

3.5.8.1. DATOS INICIALES DE PROYECTO

Tabla 3.3. Caso de uso: Datos iniciales de proyecto

CU-CREACOD-01: Datos iniciales de proyecto	
Resumen:	Indicar al programa los parámetros iniciales del proyecto
Prioridad:	Esencial
Actores directos:	Programador

Escenarios

Tipo de escenario	**Descripción**
Principal	Se deben establecer los valores de: o Nombre del proyecto

	o Autor del proyecto o Carpeta contenedora del proyecto	
Secundario	Crear	El sistema muestra los controles necesarios para los parámetros iniciales del proyecto: o El usuario ingresa el Nombre del proyecto. o El usuario ingresa el Autor del proyecto o El usuario ingresa la carpeta destino del proyecto, usando para ello un botón de búsquedas de carpetas.
Secundario	Editar	El sistema muestra los controles necesarios para la edición parámetros iniciales del proyecto: o El usuario cambia el Nombre del proyecto. o El usuario cambia el Autor del proyecto o El usuario cambia la carpeta destino del proyecto, usando para ello un botón de búsquedas de carpetas.

Pre-condiciones
Notas: El usuario debió ingresar al sistema CreaCod e iniciar un nuevo proyecto.

3.5.8.2. SELECCIONAR BASE DE DATOS

Tabla 3.4. Caso de uso: Seleccionar base de datos

CU-CREACOD-02: Seleccionar base de datos	
Resumen:	Indicar al programa la base de datos que se conectará para el proyecto.
Prioridad:	Esencial
Actores directos:	Programador

Escenarios

Tipo de escenario	**Descripción**	
Principal	Se deben establecer los valores de: o Base de datos	
Secundario	Crear	El sistema muestra los controles necesarios para seleccionar la base de datos entre: o MySQL o Access o SQL Server
Secundario	Modificar	El sistema muestra los controles necesarios para modificar la selección de la base de datos entre:

		o MySQL o Access o SQL Server

Pre-condiciones
Notas: El sistema deberá encontrar las distintas plantillas de bases de datos soportadas.

Validaciones
El sistema alertará en el caso que algún parámetro esté vacío o sea incorrecto

3.5.8.3. CONECTAR BASE DE DATOS

Tabla 3.5. Caso de uso: Conectar base de datos

CU-CREACOD-02.1: Conectar base de datos	
Resumen:	Indicar al programa los parámetros necesarios para la conexión de la base de datos.
Prioridad:	Esencial
Actores directos:	Programador

Escenarios

Tipo de escenario	**Descripción**	
Principal	Se deben establecer los valores de: o Servidor o Base o Login-Usuario o Login-Password En determinados casos, se deberá también establecer los valores de: o Driver o Puerto o Archivo de base de datos o Autentificación de Windows	
Secundario	Crear	El sistema muestra los controles necesarios para la conexión de base de datos en un formulario modal: o Servidor o Base o Login-Usuario o Login-Password En determinados casos, se deberá mostrar los controles para los valores de: o Driver

		o Puerto o Archivo de base de datos o Autentificación de Windows
Secundario	Modificar	El sistema muestra los controles necesarios para editar la conexión de base de datos en un formulario modal: o Servidor o Base o Login-Usuario o Login-Password En determinados casos, se deberá mostrar los controles para los valores de: o Driver o Puerto o Archivo de base de datos o Autentificación de Windows

Pre-condiciones
El usuario debe asegurarse que los valores escritos sean los correctos.

Validaciones
El sistema verifica la conexión a la base de datos, presentando un mensaje de error o confirmación de conexión.

3.5.8.4. SELECCIONAR LENGUAJE DE PROGRAMACIÓN

Tabla 3.6. Caso de uso: Seleccionar lenguaje de programación

CU-CREACOD-03: Seleccionar lenguaje de programación	
Resumen:	Indicar al programa el lenguaje de programación que generará el programa.
Prioridad:	Esencial
Actores directos:	Programador

Escenarios

Tipo de escenario	**Descripción**	
Principal	Se deben establecer los valores de: o Lenguaje	
Secundario	Crear	El sistema muestra los controles necesarios para seleccionar el lenguaje de programación entre: o PHP o ASP o JSP

		Adicionalmente, para ciertos casos, se mostrará los siguientes lenguajes: o Visual Basic 6 o Visual Basic 2005 o Visual C#
Secundario	Modificar	El sistema muestra los controles necesarios para modificar la selección de lenguaje de programación entre: o PHP o ASP o JSP Adicionalmente, para ciertos casos, se mostrará los siguientes lenguajes: o Visual Basic 6 o Visual Basic 2005 o Visual C#

Pre-condiciones
Notas: El sistema deberá encontrar las distintas plantillas de lenguajes de programación soportados.

Validaciones
El sistema alertará en el caso de que el parámetro esté vacío o sea incorrecto

3.5.8.5. REFRESCAR INFORMACIÓN DE BASE DE DATOS

Tabla 3.7. Caso de uso: Refrescar información de base de datos

CU-CREACOD-04: Refrescar información de base de datos		
Resumen:	Indicar al programa si se desea refrescar la información de la base de datos.	
Prioridad:	Esencial al inicio. Opcional si se modifica la base de datos.	
Actores directos:	Programador	
Escenarios		
Tipo de escenario	**Descripción**	
Principal	Se indica al usuario: o En nuevo proyecto: Mensaje opcional si se desea refrescar la base o En proyecto existente: Botón y menú para refrescar base de datos	
Secundario	Crear	El sistema muestra en todo momento las

		distintas opciones de refrescamiento de base de datos: o Botón  en la barra de íconos o Menú: Proyecto – Refrescar Base

3.5.8.6. PARÁMETROS DE TABLAS

Tabla 3.8. Caso de uso: Parámetros de tablas

CU-CREACOD-01: Parámetros de tablas		
Resumen:	Indicar al programa los distintos parámetros para cada una de las tablas de la base de datos	
Prioridad:	Esencial	
Actores directos:	Programador	

Escenarios

Tipo de escenario	**Descripción**	
Principal	Se deben establecer, para cada uno de las tablas, los valores de: o Nombre de clase o Nombre de objeto o Nombre visible o Columna/Campo principal o Columna/Campo visible o Columna/Campo de orden o Columna/Campo habilitado	
Secundario	Crear	El sistema muestra una pantalla modal con los controles necesarios para los parámetros de tabla: o Nombre de clase o Nombre de objeto o Nombre visible o Columna/Campo principal o Columna/Campo visible o Columna/Campo de orden o Columna/Campo habilitado
Secundario	Modificar	El sistema muestra una pantalla modal con los controles necesarios para modificar los parámetros de tabla: o Nombre de clase o Nombre de objeto o Nombre visible o Columna/Campo principal o Columna/Campo visible

		o Columna/Campo de orden o Columna/Campo habilitado

Validaciones
El sistema alertará en el caso de que algún parámetro esté vacío o sea incorrecto

3.5.8.7. PARÁMETROS DE CAMPOS

Tabla 3.9. Caso de uso: Parámetros de campos

CU-CREACOD-06: Parámetros de tablas		
Resumen:	Indicar al programa los distintos parámetros para cada una de los campos de las tablas de la base de datos	
Prioridad:	Esencial	
Actores directos:	Programador	

Escenarios

Tipo de escenario	**Descripción**	
Principal	Se deben establecer, para cada uno de las tablas, los valores de: o Nombre de función o Nombre visible o Tipo de dato o Valor predefinido En el caso que el campo sea relacional, se deberá indicar los siguientes parámetros: o Clase/Tabla relacional o Nombre de objeto	
Secundario	Crear	El sistema muestra una pantalla modal con los controles necesarios para los parámetros de campo: o Nombre de función o Nombre visible o Tipo de dato o Valor predefinido o Clase/Tabla relacional o Nombre de objeto
Secundario	Modificar	El sistema muestra una pantalla modal con los controles necesarios para la edición de los parámetros de campo: o Nombre de función o Nombre visible o Tipo de dato o Valor predefinido o Clase/Tabla relacional

		o Nombre de objeto

Validaciones
El sistema alertará en el caso de que algún parámetro esté vacío o sea incorrecto

3.5.8.8. PARÁMETROS DE PLANTILLA DE GESTIÓN GUI

Tabla 3.10. Caso de uso: Parámetros de plantilla de gestión GUI

CU-CREACOD-07: Datos iniciales de proyecto	
Resumen:	Indicar al programa los parámetros de los campos para su presentación en entorno GUI.
Prioridad:	Opcional, en el caso que se seleccione una plantilla de gestión GUI.
Actores directos:	Programador

Escenarios

Tipo de escenario	**Descripción**	
Principal	Se deben establecer los valores de: o Funciones de entrada (Predefinidas por el sistema) o Funciones de salida (Predefinidas por el sistema) o Funciones de validaciones o Tipo de control para el campo	
Secundario	Crear	El sistema muestra una pantalla modal con los controles necesarios para la edición de parámetros de la plantilla de gestión GUI: o Funciones de entrada (Predefinidas por el sistema) o Funciones de salida (Predefinidas por el sistema) o Funciones de validaciones o Tipo de control para el campo
Secundario	Editar	El sistema muestra una pantalla modal con los controles necesarios para la edición de parámetros de la plantilla de gestión GUI: o Funciones de entrada (Predefinidas por el sistema) o Funciones de salida (Predefinidas por el sistema) o Funciones de validaciones o Tipo de control para el campo

Condiciones
Notas: En el caso de seleccionar una plantilla de gestión GUI, es necesario que todos los campos de la tabla cuenten con los parámetros correctamente establecidos.

Validaciones
o El sistema alertará en el caso de que algún parámetro esté vacío o sea incorrecto. o El sistema alertará si algún campo de la tabla no se encuentra con los parámetros de plantilla GUI correctamente establecidos.

3.5.8.9. GENERAR CÓDIGO FUENTE

Tabla 3.11. Caso de uso: Generar código fuente

CU-CREACOD-08: Generar código fuente		
Resumen:	Generar código fuente de las clases de gestión de base de datos a partir de los parámetros de la base de datos. Adicionalmente crear archivos de gestión GUI en el caso de haber seleccionado plantillas de gestión.	
Prioridad:	Opcional	
Actores directos:	Programador	

Escenarios

Tipo de escenario	Descripción	
Principal	Se indica al usuario: o En proyecto existente: Botón y menú para refrescar base de datos	
Secundario	Crear	El sistema muestra en todo momento las distintas opciones de creación de código o Botón  en la barra de íconos o Menú: Proyecto – Generar

Validaciones
o El sistema alertará en el caso de que algún parámetro esté vacío o sea incorrecto. o El sistema alertará si algún campo de la tabla no se encuentra con los parámetros de plantilla GUI correctamente establecidos.

3.5.8.10. CAMBIAR IDIOMA DEL IDE DE CREACOD

Tabla 3.12. Caso de uso: Cambiar idioma de IDE de CreaCod

CU-CREACOD-01: Datos iniciales de proyecto	
Resumen:	Indicar al programa el idioma del IDE de CreaCod
Prioridad:	Opcional
Actores directos:	Programador

Escenarios

Tipo de escenario	**Descripción**	
Principal	Se deben establecer los valores de: o Idioma de IDE	
Secundario	Seleccionar	El sistema muestra una lista con los distintos idiomas, entre ellos: o Español o Ingles o Francés

Condiciones
Notas: El sistema cambia automáticamente todos los valores al idioma seleccionado y no necesita ser reiniciado.

CAPÍTULO 4

PRUEBAS DE CREACOD

4.1. DEFINICIÓN

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

- Microsoft Access 2003
- SQL Server 2005 Express
- MySQL 5.5.27

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

- ASP
- JPS
- PHP

Cada una de las bases de datos será probada para cada uno de los lenguajes de programación antes mencionados.

4.2. CARACTERÍSTICAS

4.2.1. CREACOD 1.0.1

El sistema CreaCod fue desarrollado en Visual Basic 2005 Express Edition, por lo que su ambiente de trabajo es únicamente Windows XP/Vista/7.

4.2.1.1. PASOS DE INSTALACIÓN

Ejecutar el archivo “setup.exe”, presionar Next.

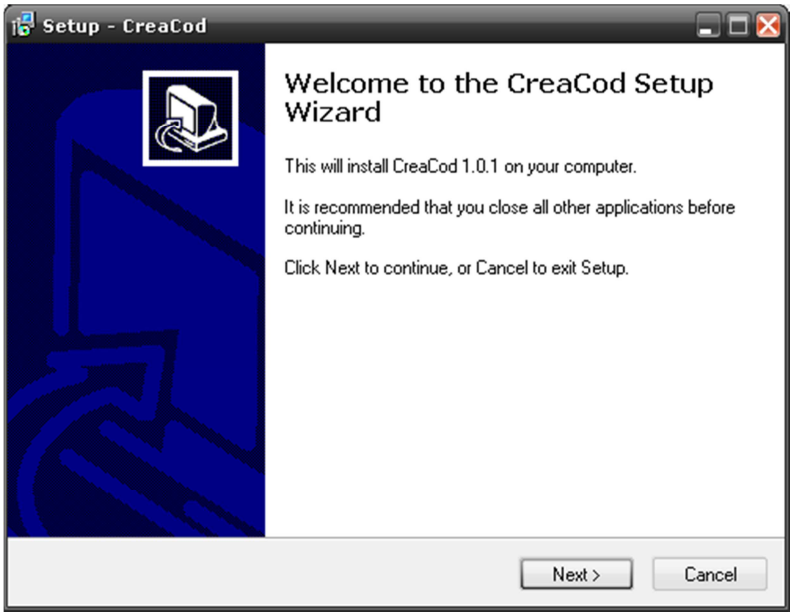


Figura 4.1. Instalación CreaCod: Archivo setup.exe

Seleccionar la carpeta donde se desea instalar el programa. Presionar Next.

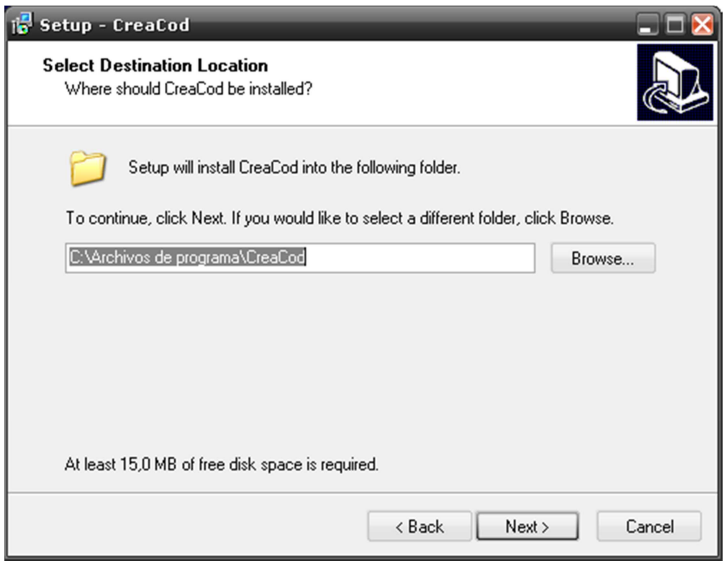


Figura 4.2. Instalación CreaCod: Selección de carpeta destino

Escribir el nombre del acceso directo de CreaCod, presionar Next.

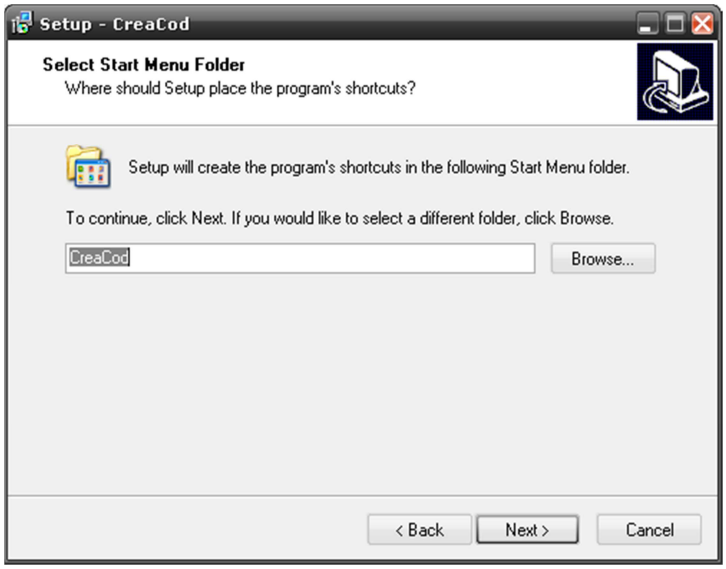


Figura 4.3. Instalación CreaCod: Acceso directo

Indicar al instalador si se desea crear un acceso directo en el escritorio, presionar el botón Next.

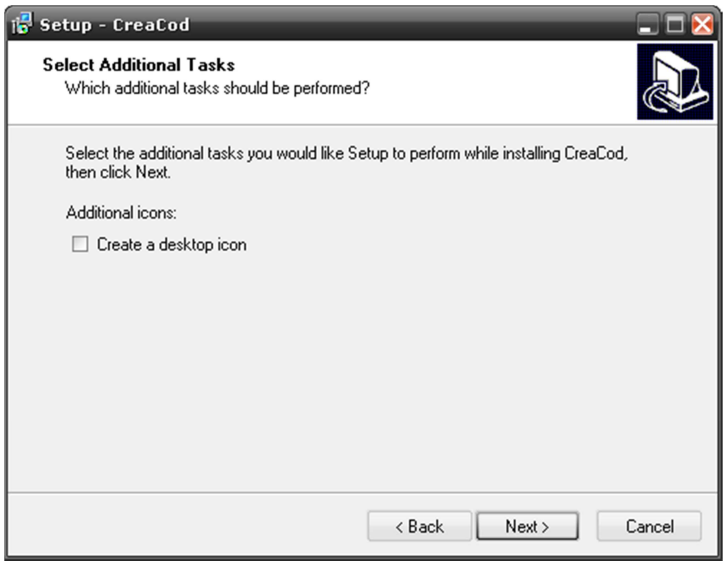


Figura 4.4. Instalación CreaCod: Acceso directo

Una vez establecidos los parámetros de instalación, el programa se instalará en la ubicación seleccionada.

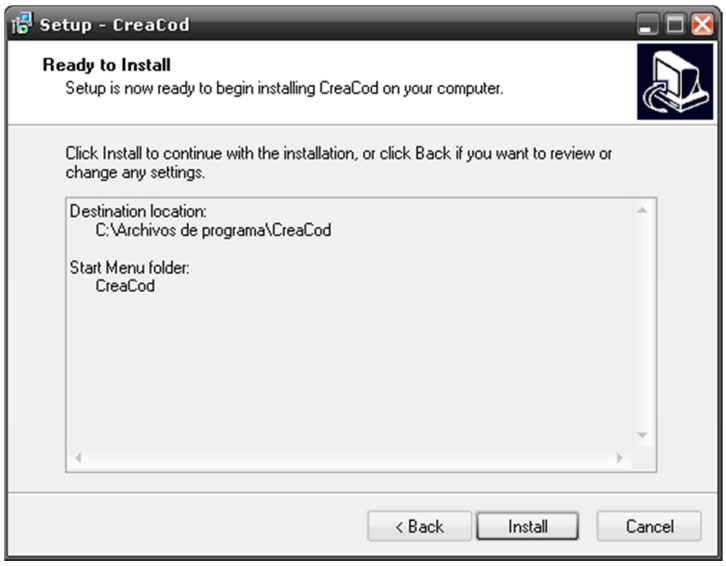


Figura 4.5. Instalación CreaCod: Advertencia de valores de instalación

Una vez instalado el sistema se creará el acceso directo en la barra de inicio de Windows.

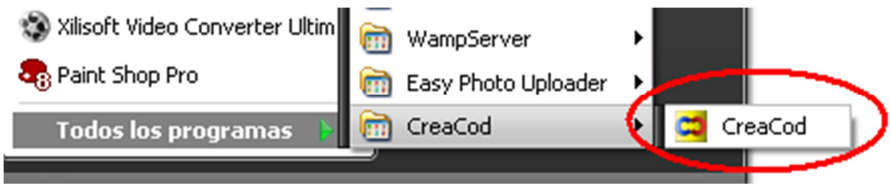


Figura 4.6. Instalación CreaCod: Acceso directo en la barra Inicio

4.2.2. BASE DE DATOS DE PRUEBA “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.

4.2.2.1. MODELO LÓGICO

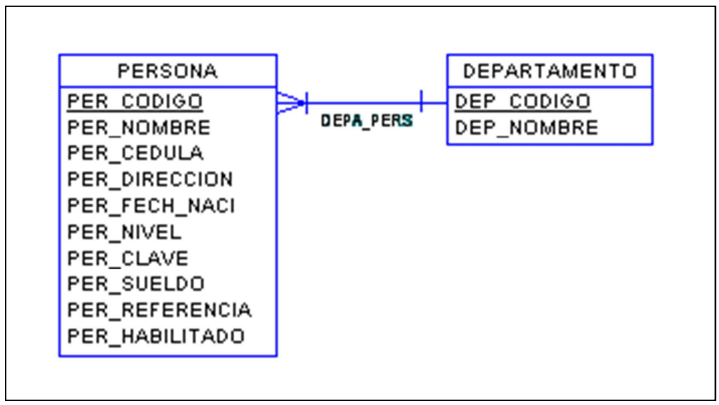


Figura 4.7. Modelo lógico de la base de datos “persona”

4.2.2.2. MODELO FÍSICO

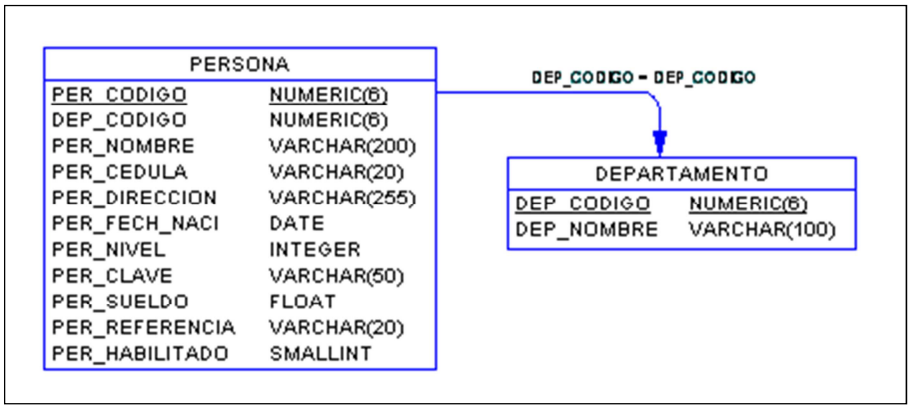


Figura 4.8. Modelo físico de la base de datos “persona”

4.2.3. BASE DE DATOS MYSQL Y SERVIDOR APACHE/PHP

Existe en el mercado un paquete que contiene el Servidor web Apache, el lenguaje de programación PHP y la base de datos MySQL denominado “WAMP”. Para las pruebas de CreaCod se ha decidido instalar dicho paquete, ya que ahorra tiempo de configuración y es muy estable.

Figura 4.9. Barra de configuración de WAMP Server 2.2

4.2.3.1. CREACIÓN DE BASE DE DATOS “PERSONA”

1. En la computadora de pruebas se ha instalado el programa “MySQL-Front” para la gestión de la base de datos MySQL. Abrir el programa MySQL-Front y dar clic derecho sobre el servidor, seleccionar “Nueva – Base de datos”.

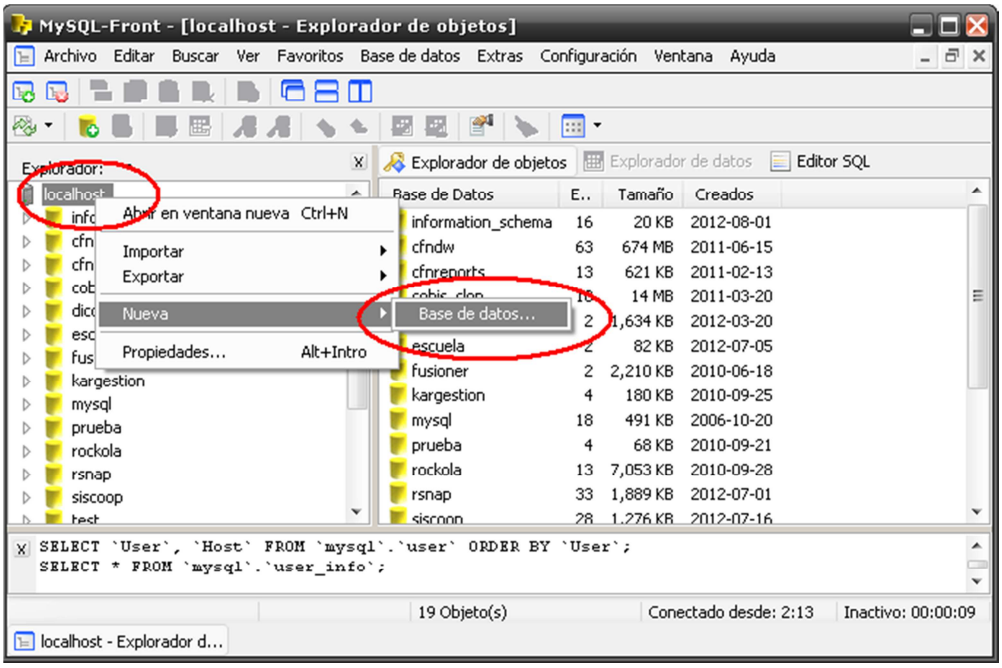


Figura 4.10. MySQL: Creación de nueva base de datos

2. Se escribe el nombre de la nueva base de datos y aceptar.

Figura 4.11. MySQL: Nombre de nueva base de datos

3. La nueva base de datos aparecerá en la lista de base de datos.

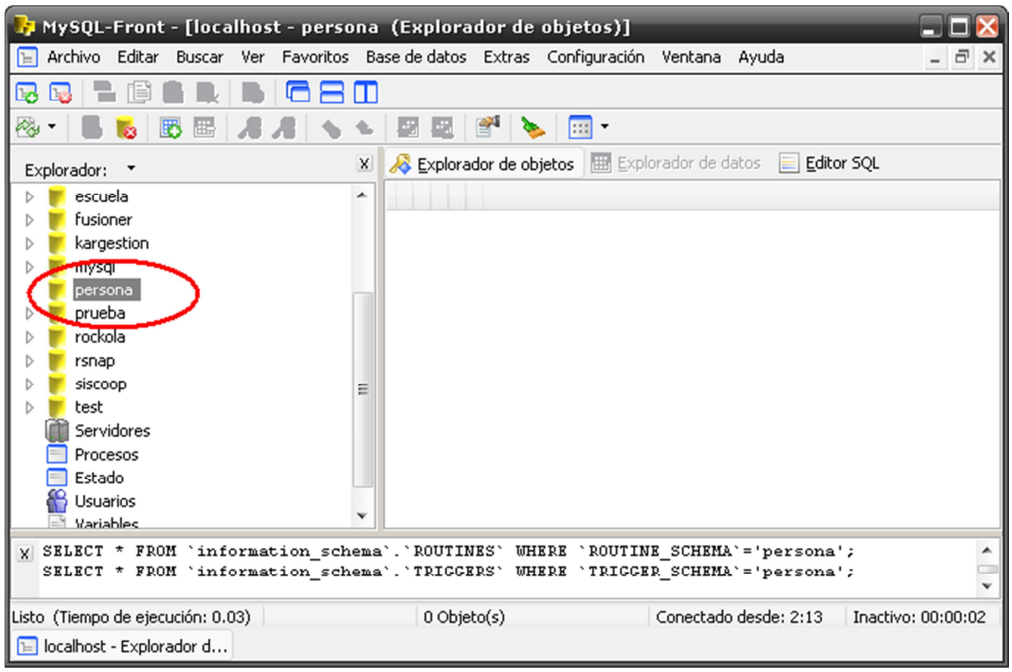


Figura 4.12. MySQL: Base de datos “persona”

4. MySQL puede ejecutar Scripts de creación de tablas y columnas. El script se obtiene del programa DataArchitect.

Código fuente 4.1. Script para la creación de base de datos “persona” en MySQL

```
create table DEPARTAMENTO
(
    DEP_CODIGO      bigint(20) NOT NULL auto_increment,
    DEP_NOMBRE      VARCHAR(100)                    ,
    primary key (DEP_CODIGO)
);
create unique index DEPARTAMENTO_PK on DEPARTAMENTO (DEP_CODIGO asc);

create table PERSONA
(
    PER_CODIGO      bigint(20) NOT NULL auto_increment,
    DEP_CODIGO      bigint(20)             not null,
    PER_NOMBRE      VARCHAR(200)                    ,
    PER_CEDULA      VARCHAR(20)                     ,
    PER_DIRECCION   VARCHAR(255)                    ,
```

```
    PER_FECH_NACI   date                            ,
    PER_NIVEL       INTEGER                         ,
    PER_CLAVE       VARCHAR(50)                     ,
    PER_SUELDO      double                           ,
    PER_REFERENCIA  VARCHAR(20)                     ,
    PER_HABILITADO  SMALLINT                        ,
    primary key (PER_CODIGO),
    foreign key  (DEP_CODIGO)
       references DEPARTAMENTO (DEP_CODIGO)
);

create unique index PERSONA_PK on PERSONA (PER_CODIGO asc);

create index DEPA_PERS_FK on PERSONA (DEP_CODIGO asc);
```

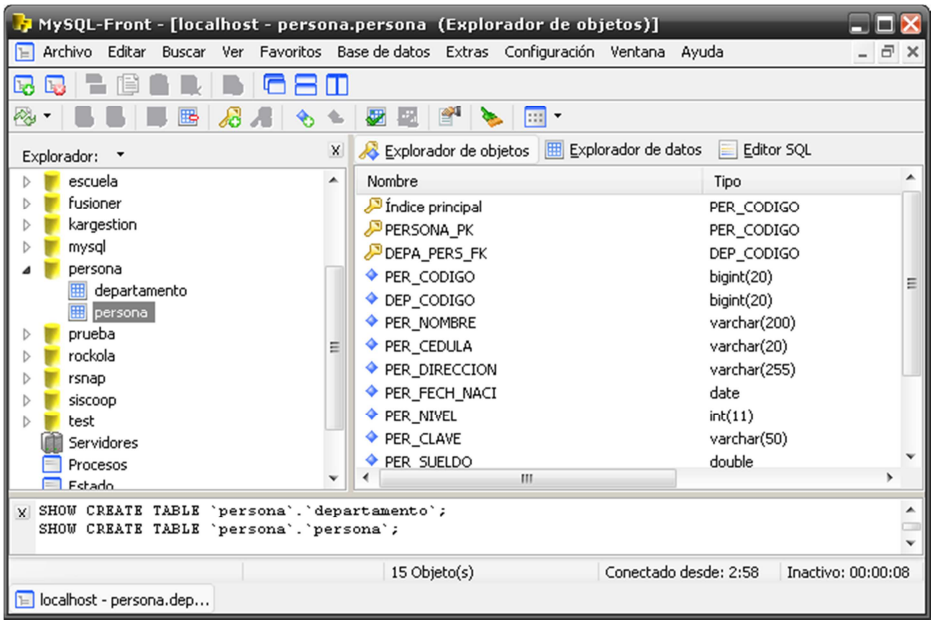


Figura 4.13. MySQL: Base de datos “persona” con tablas y campos

4.2.3.2. PREREQUISITOS PARA PRUEBAS

Se debe crear un usuario distinto de “root” para poder acceder a MySQL desde los servidores web. La creación de nuevos usuarios es la siguiente:

1. Abrir el programa “MySQL Administrator”. Es posible que la distribución de MySQL no venga con dicho programa, por lo que se lo debe descargar desde la página oficial de MySQL.
2. Iniciar sesión con las credenciales de “root”.

Figura 4.14. “MySQL Administrator”: Inicio de sesión

3. Seleccionar “User Administration” y a continuación presionar el botón “New User”.

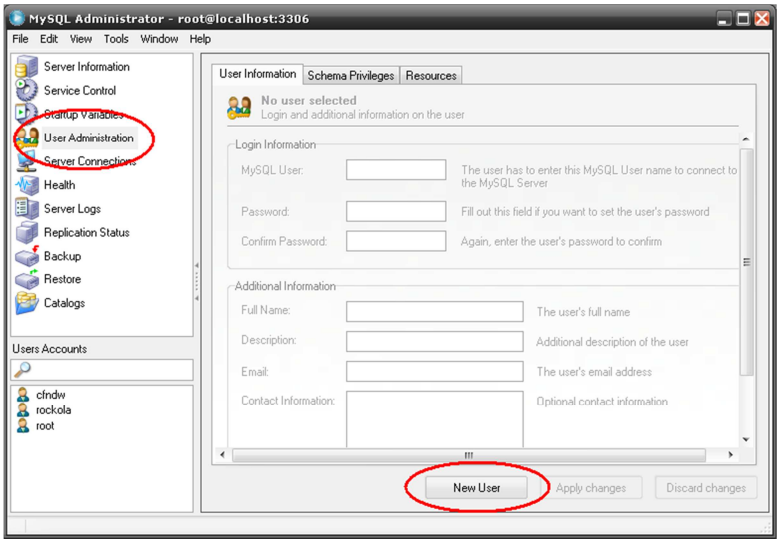


Figura 4.15. MySQL Administrador: Selección para crear nuevo usuario

4. Llenar los datos con los valores del nuevo usuario.

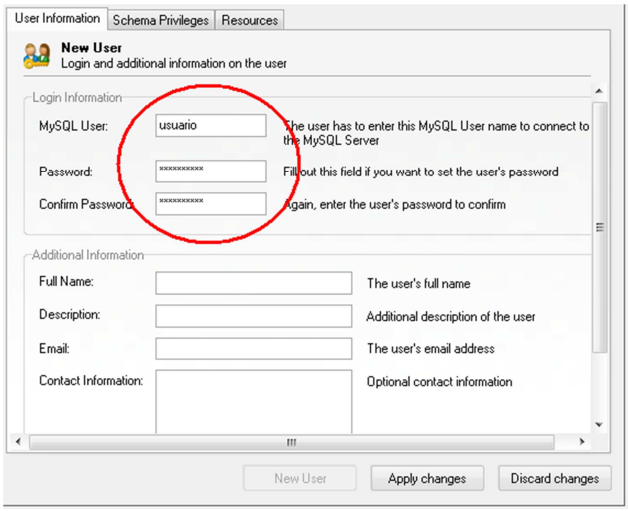


Figura 4.16. MySQL Administrator: Datos de nuevo usuario

5. Seleccionar la pestaña “Schema privileges”, seleccionar la base de datos “persona” y a continuación dar todos los permisos al usuario creado presionando el botón “<<”.

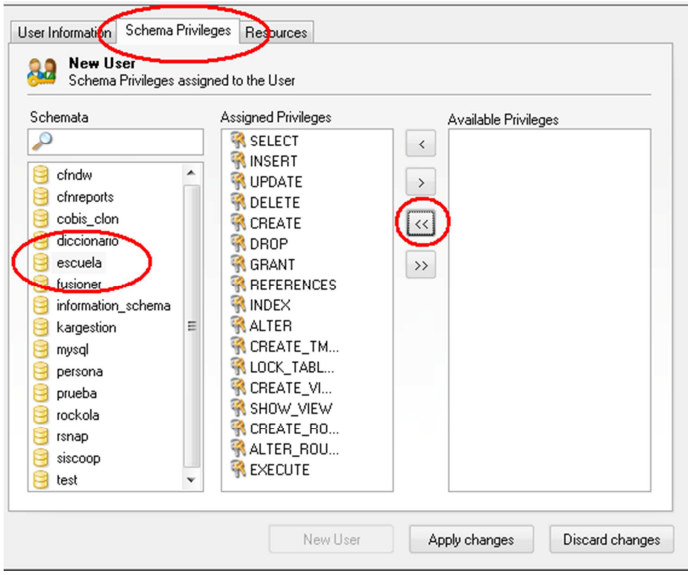


Figura 4.17. MySQL Administrator: Asignando permisos a nuevo usuario

6. Presionar el botón “Apply changes”. El nuevo usuario se mostrará en la lista de usuarios de MySQL.

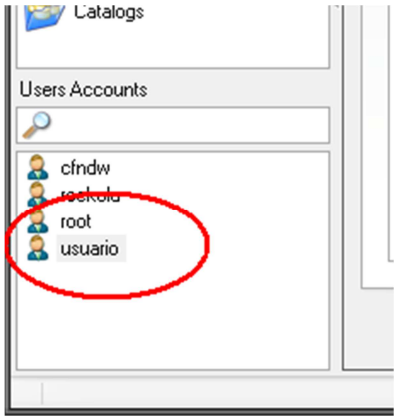


Figura 4.18. MySQL Administrator: Nuevo usuario creado

4.2.4. BASE DE DATOS ACCESS

Los archivos “mdb” de Microsoft Access necesitan runtimes ya adjuntos en todas las versiones de Windows, por lo que no se requiere de su instalación. Sin embargo se aconseja instalar el paquete “Microsoft Office Professional” el cual contiene el gestor de base de datos Access.

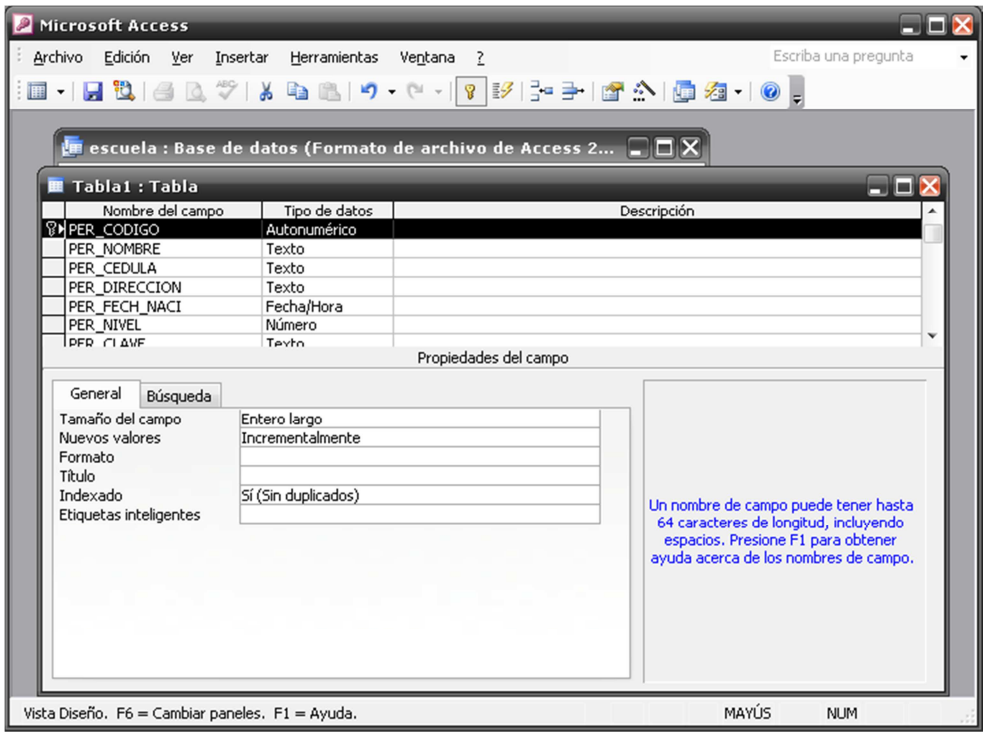


Figura 4.19. Interfaz de usuario de Microsoft Access

4.2.4.1. CREACIÓN DE BASE DE DATOS “PERSONA”

Microsoft Access no permite la creación de campos y tablas a través de sentencias SQL, por lo que se te tiene que crear la base de datos de forma manual, además de las relaciones entre tablas.

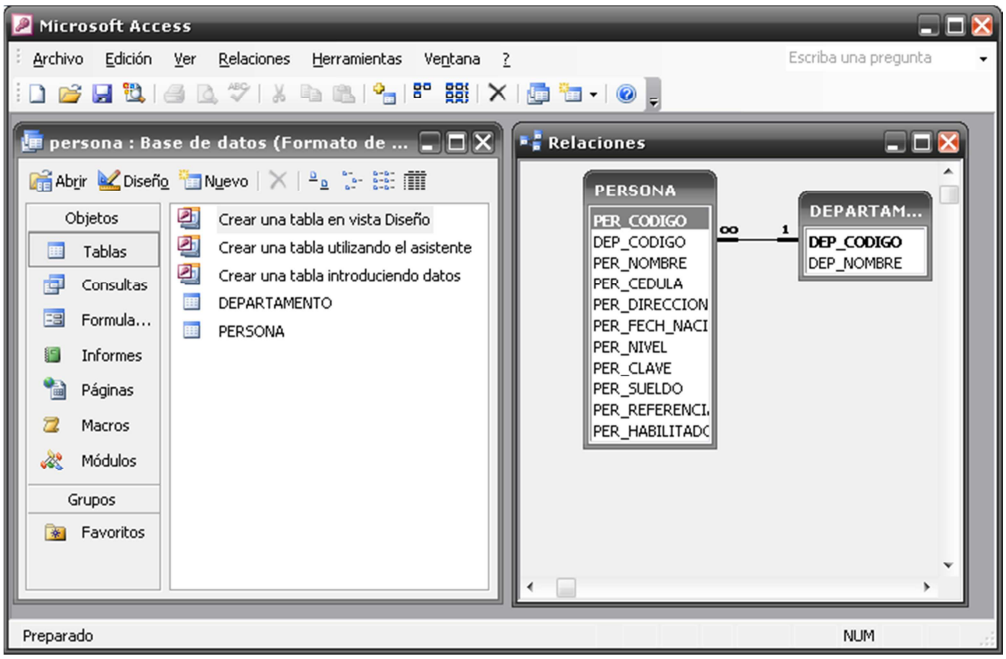


Figura 4.20. Microsoft Access: Base de datos “persona”

4.2.4.2. PREREQUISITOS PARA PRUEBAS

Dado que los servidores web no pueden conectarse directamente con archivos Access es necesario crear una conexión ODBC. Los pasos para la conexión son los siguientes:

1. Ir a Inicio – Panel de control – Herramientas Administrativas – Orígenes de datos (ODBC)

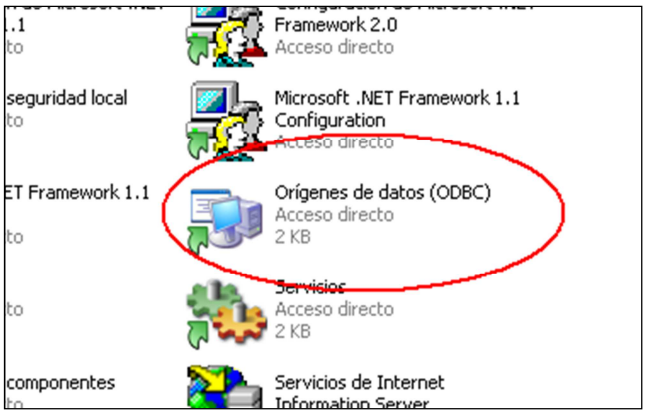


Figura 4.21. Access ODBC: Orígenes de datos de Panel de Control

2. Agregar una nueva conexión.

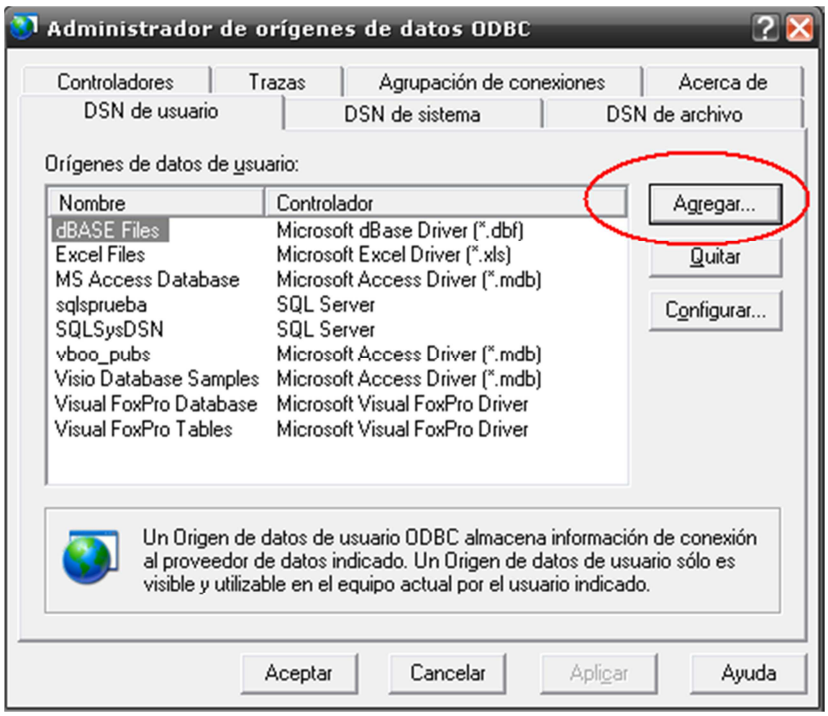


Figura 4.22. Access ODBC: Nuevo DSN

3. Seleccionar el Driver para Access.

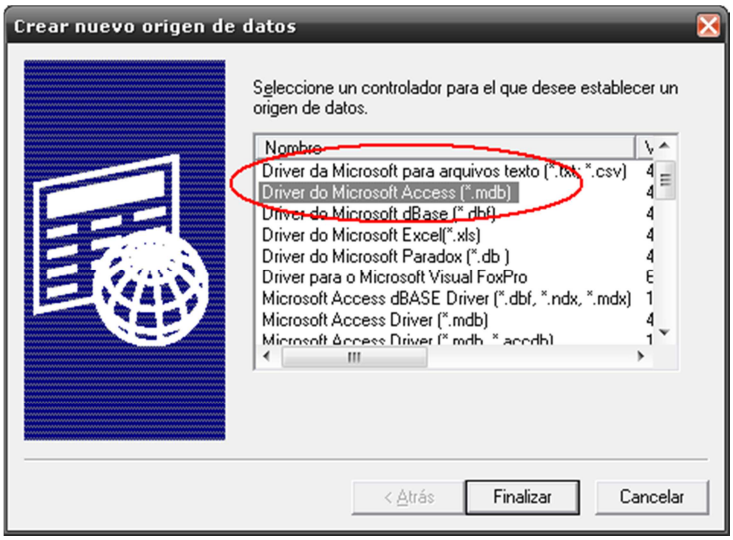


Figura 4.23. Access ODBC: Driver de conexión ODBC

4. Seleccionar el archivo Access y escribir el nombre de la nueva conexión. Y aceptar.

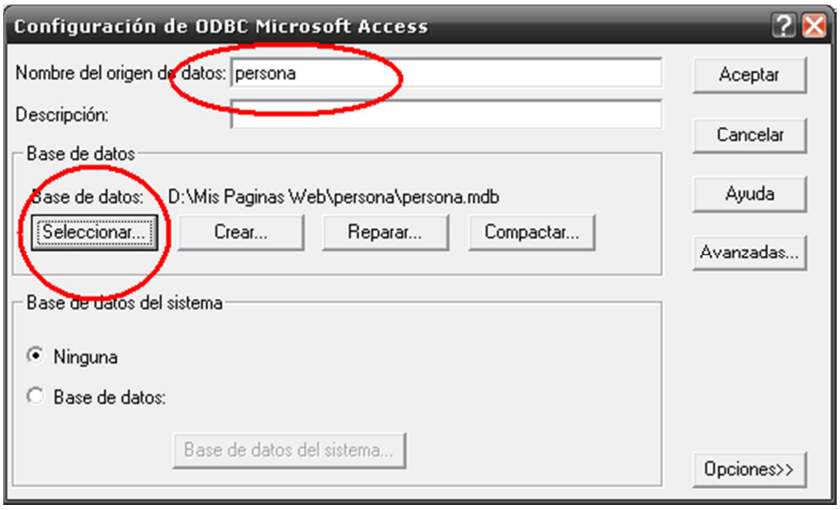


Figura 4.24. Access ODBC: Creación de ODBC para “persona”

5. La nueva conexión se presentará en la lista de conexiones.

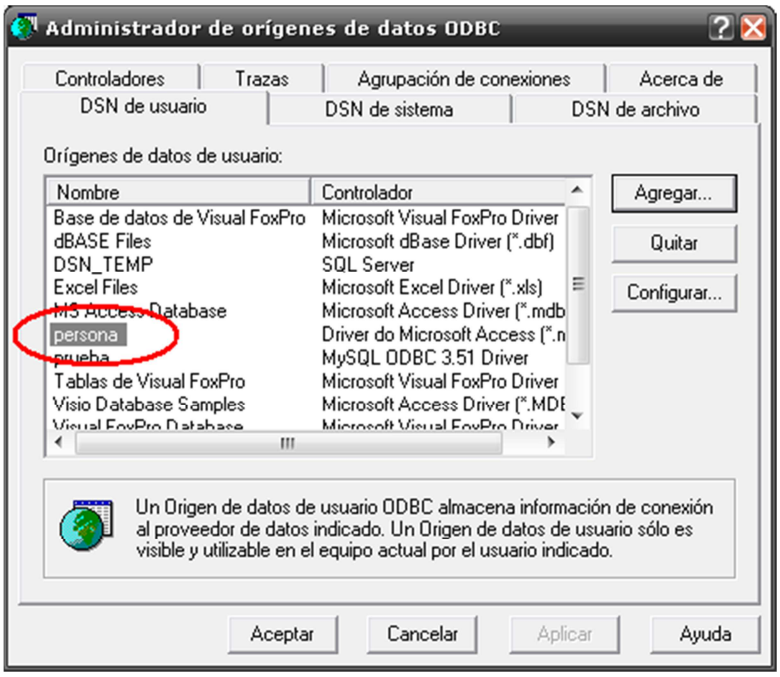


Figura 4.25. Access ODBC: DSN “persona”

4.2.4.3. ARCHIVOS ACCESS COMO FUENTES DE DATOS PARA WEB

Por seguridad, los archivos Access no pueden ser leídos directamente desde los servidores web, razón por la cual se deben realizar ciertos cambios de seguridad para que los archivos Access puedan ser abiertos. Los pasos son los siguientes:

1. Abrir el explorador de archivos de Windows, seleccionar el menú “Herramientas” y seleccionar “Opciones de Carpeta”

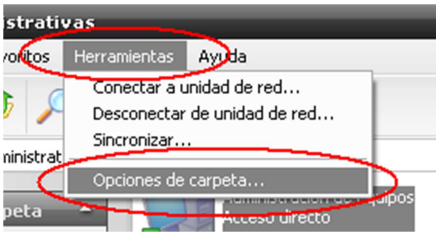


Figura 4.26. Archivos Access: Opciones de carpeta

2. Ir a la pestaña “Ver” y quitar la selección a la opción “Utilizar uso compartido simple de archivos”.

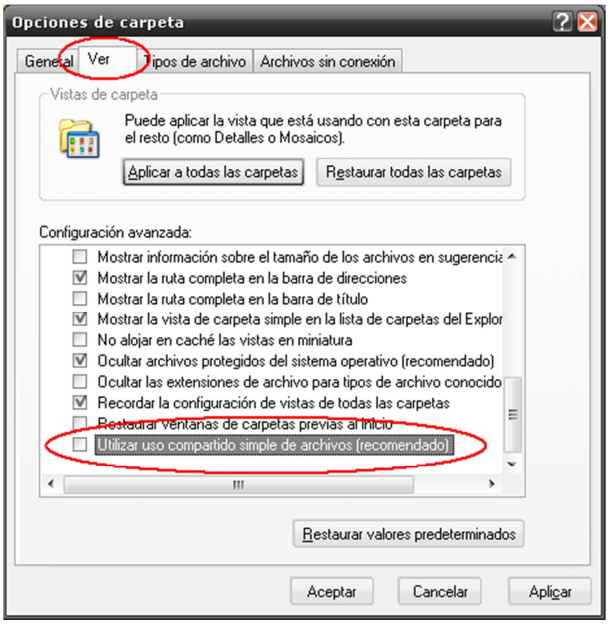


Figura 4.27. Archivos Access: Uso avanzado de compartición de archivos

3. Seleccionar la carpeta donde se encuentra el archivo Access, presionar clic derecho y seleccionar “Propiedades”

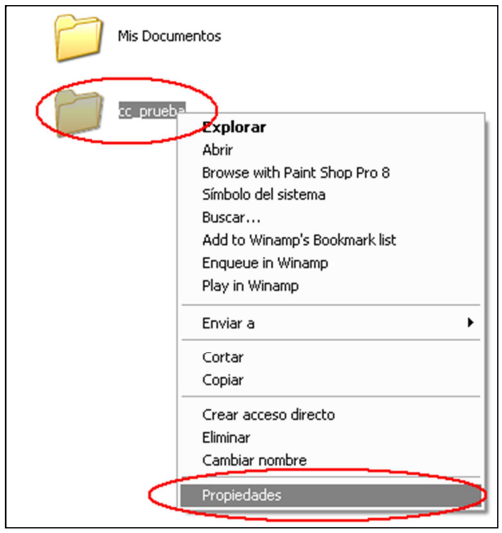


Figura 4.28. Archivos Access: Propiedades de carpeta

4. En la pestaña “Seguridad” seleccionar el botón “Agregar”

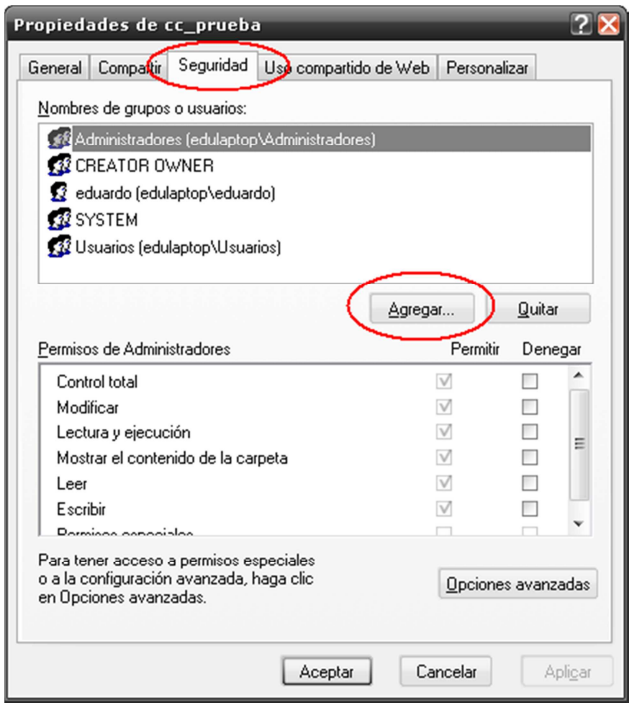


Figura 4.29. Archivos Access: Nuevo rol de seguridad

5. Escribir el usuario “IUSR_{NOMBRE_EQUIPO}” y aceptar.

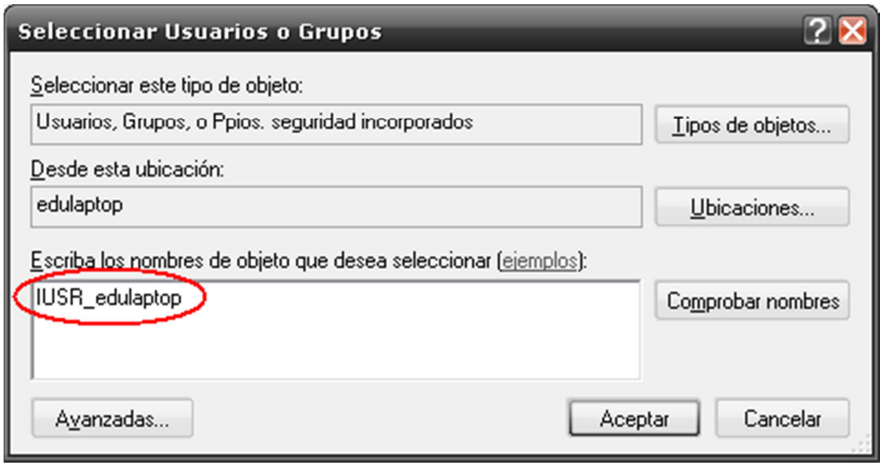


Figura 4.30. Archivos Access: Nuevo usuario

6. Darle todos los permisos al usuario seleccionado

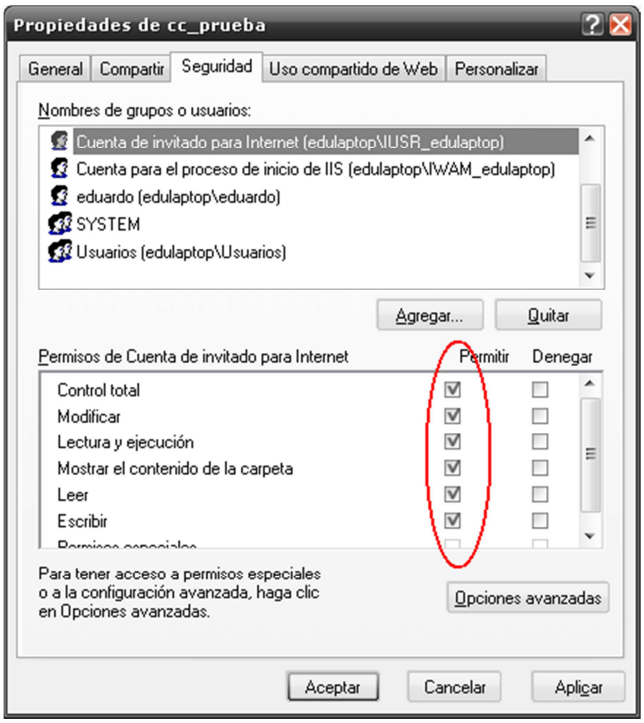


Figura 4.31. Archivos Access: Permisos de nuevo usuario

7. Realizar el mismo proceso anterior, pero para el usuario “IWAM_{NOMBRE_EQUIPO}”

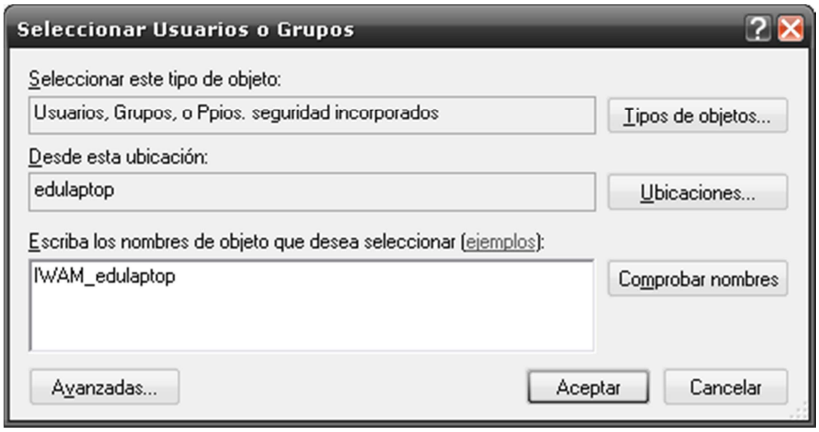


Figura 4.32. Archivos Access: Usuario 2

8. Ahora se debe compartir la carpeta para que sea accedida desde el Internet, para ello se debe dar clic derecho sobre la carpeta, seleccionar propiedades y en la pestaña “Uso compartido de Web” seleccionar “Compartir esta carpeta”, inmediatamente aparecerá la pantalla de “Modificar Alias”, dar permiso de lectura y escritura y aceptar los cambios.

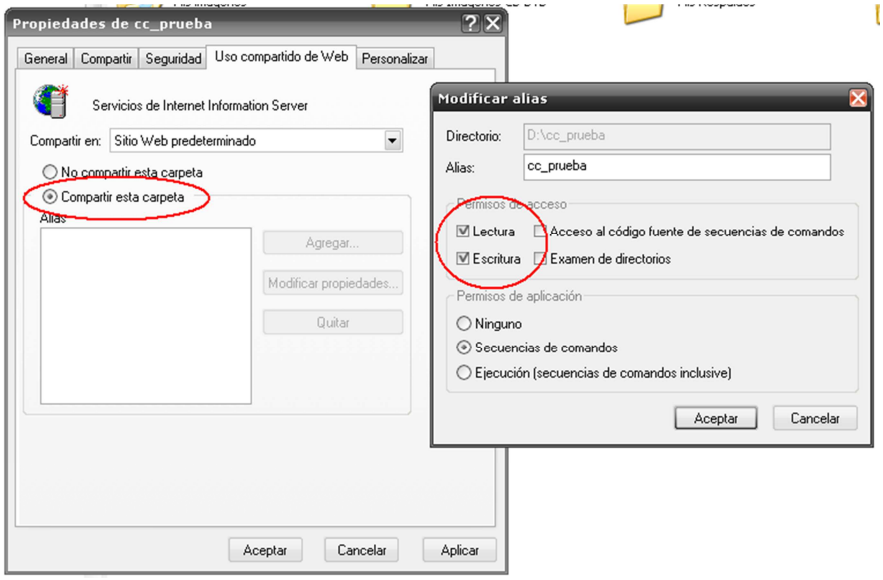


Figura 4.33. Archivos Access: Compartición de carpeta

9. Aceptar los cambios. Ahora se puede utilizar los archivos de Access contenidos en la carpeta seleccionada como base de datos de páginas dinámicas.

4.2.5. SQL SERVER 2005 EXPRESS

La base de datos SQL Server 2005 Express se puede descargar del sitio oficial de Microsoft y se decidió usarla debido a que es de libre distribución.

4.2.5.1. CREACIÓN DE BASE DE DATOS “PERSONA”

1. Abrir el programa “Microsoft SQL Server Management Studio Express” (MSSMSE) e iniciar como usuario Windows. Es posible que la distribución de

SQL Express no tenga este programa, por lo que deberá descargarse del sitio oficial de Microsoft.

Figura 4.34. MSSMSE: Inicio de sesión de Windows

2. Clic derecho en “Databases” y seleccionar “New database”.

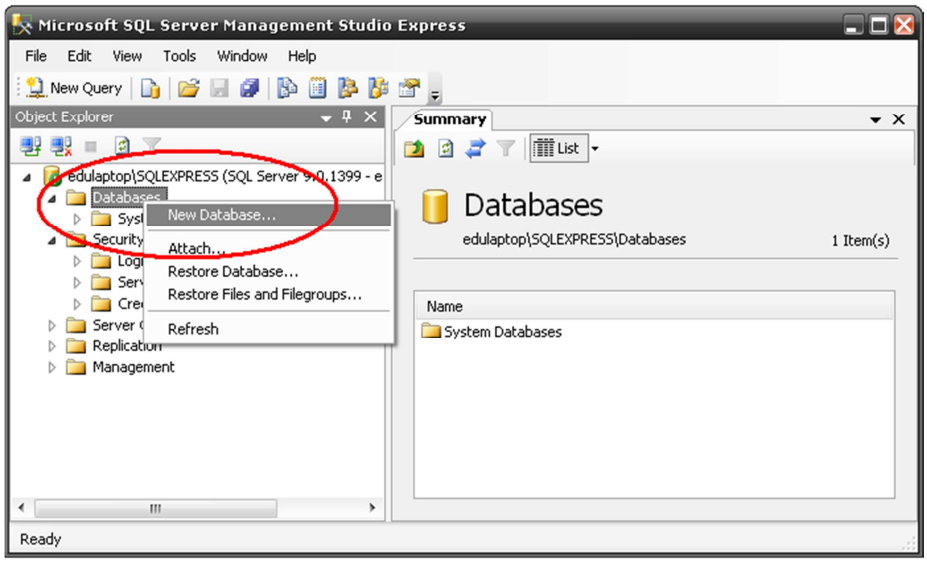


Figura 4.35. MSSMSE: Nueva base de datos

3. Escribir el nuevo nombre de la base de datos.

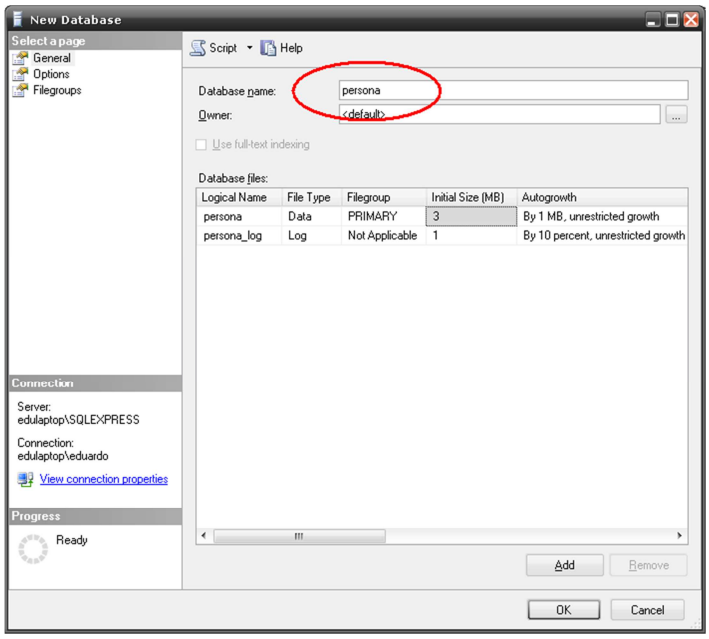


Figura 4.36. MSSMSE: Creando la base de datos “persona”

4. Presionar Ok. La base de datos está creada.

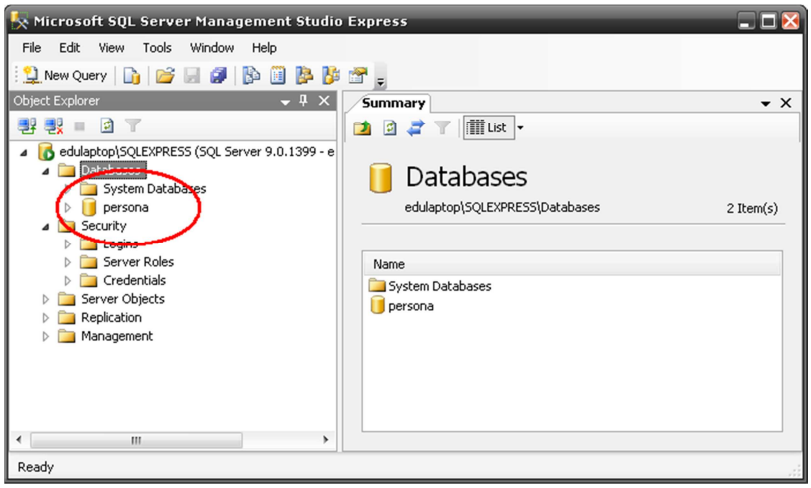


Figura 4.37. MSSMSE: Base de datos “persona”

5. SQL Server 2005 Express puede ejecutar Scripts de creación de tablas y columnas. El script se obtiene del programa DataArchitect.

Código fuente 4.2. Script para la creación de base de datos “persona” en SQL Server

```
create table DEPARTAMENTO
(
    DEP_CODIGO      numeric               identity,
    DEP_NOMBRE      varchar(100)          null    ,
    constraint PK_DEPARTAMENTO primary key (DEP_CODIGO)
)
go
create table PERSONA
(
    PER_CODIGO      numeric               identity,
    DEP_CODIGO      numeric               not null,
    PER_NOMBRE      varchar(200)          null    ,
    PER_CEDULA      varchar(20)           null    ,
    PER_DIRECCION   varchar(255)          null    ,
    PER_FECH_NACI   datetime              null    ,
    PER_NIVEL       int                   null    ,
    PER_CLAVE       varchar(50)           null    ,
    PER_SUELDO      float                 null    ,
    PER_REFERENCIA  varchar(20)           null    ,
    PER_HABILITADO  tinyint               null    ,
    constraint PK_PERSONA primary key (PER_CODIGO)
)
go
alter table PERSONA
    add constraint FK_PERSONA_DEPA_PERS_DEPARTAM foreign key  (DEP_CODIGO)
       references DEPARTAMENTO (DEP_CODIGO)
go
```

6. Seleccionar la base de datos “persona” y dar clic derecho, seleccionar la opción “New Query”.