



UNIVERSIDAD CATÓLICA DEL TÁCHIRA
FACULTAD DE CIENCIAS ECONÓMICAS Y SOCIALES
ESCUELA DE ADMINISTRACIÓN Y CONTADURÍA PÚBLICA
COORDINACIÓN DE FORMACIÓN COMPLEMENTARIA

PROPUESTA PARA LA INCORPORACIÓN DE LA HERRAMIENTA CASE
“GENEXUS” EN LA FORMACIÓN DE LOS ALUMNOS DE
ADMINISTRACIÓN MENCIÓN INFORMÁTICA GERENCIAL DE LA
UNIVERSIDAD CATÓLICA DEL TÁCHIRA

San Cristóbal, Mayo de 2005



UNIVERSIDAD CATÓLICA DEL TÁCHIRA
FACULTAD DE CIENCIAS ECONÓMICAS Y SOCIALES
ESCUELA DE ADMINISTRACIÓN Y CONTADURÍA PÚBLICA
COORDINACIÓN DE FORMACIÓN COMPLEMENTARIA

PROPUESTA PARA LA INCORPORACIÓN DE LA HERRAMIENTA CASE
“GENEXUS” EN LA FORMACIÓN DE LOS ALUMNOS DE
ADMINISTRACIÓN MENCIÓN INFORMÁTICA GERENCIAL DE LA
UNIVERSIDAD CATÓLICA DEL TÁCHIRA

Trabajo presentado como requisito para optar al Título de
Licenciado en Administración mención Informática Gerencial

Autor: TZUPENG YAN

Tutor: Ing. Heberth Castillo

RECONOCIMIENTO

Quiero expresar mi sincero agradecimiento a Dios todo poderoso, quien me ha acompañado a lo largo de mi vida hasta ahora, especialmente en los momentos más necesarios, enviando siempre a alguien para guiarme y ayudarme.

A mi padre, mi madre, mis hermanas, mis familiares y amigos taiwaneses, quienes me dieron apoyo material y espiritual incondicional; y me han dado fuerza para seguir adelante.

A mis amigos venezolanos y mis compañeros de estudio, porque desde que llegue a Venezuela sin la ayuda de ustedes, creo que no hubiese podido terminar los cinco años de estudio en la UCAT. De verdad aprendí mucho de ustedes.

A mi tutor Ing. Heberth Castillo, quien me orientó con entusiasmo y paciencia para llevar a cabo el presente trabajo de investigación. A pesar de que tenía mucho trabajo siempre me dedicó tiempo para atenderme y aclarar mis dudas. Sin él no lo habría terminado. ¡Muchas gracias!

Gracias también a mis profesores Ing. María Cuberos, Lic. Jorge Galiano y Lic. Magali Salas, quienes me dieron las sugerencias y opiniones valiosas para el desarrollo de este trabajo de grado.

APROBACIÓN DEL TUTOR

En mi carácter de Tutor del Trabajo de Grado presentado por el ciudadano **TZUPENG YAN**, para optar al Título de Licenciado en Administración mención Informática Gerencial, cuyo título es: **Propuesta para la incorporación de la Herramienta CASE “GeneXus” en la formación de los alumnos de Administración mención Informática Gerencial de la Universidad Católica del Táchira**, considero que dicho trabajo reúne los requisitos y méritos suficientes para ser sometido a la evaluación correspondiente.

En San Cristóbal, a los veinte días del mes de mayo de 2005.

Ing. Heberth Omar Castillo Méndez

C.I.14.265.102

ÍNDICE GENERAL

RECONOCIMIENTO	iii
APROBACIÓN DEL TUTOR	iv
LISTA DE CUADROS	vii
LISTA DE GRÁFICOS	ix
RESUMEN	xi
INTRODUCCIÓN	1
CAPÍTULO I EL PROBLEMA	2
PLANTEAMIENTO DEL PROBLEMA	2
OBJETIVOS DE LA INVESTIGACIÓN	4
Objetivo General	4
Objetivos Específicos.....	5
JUSTIFICACIÓN E IMPORTANCIA	5
ALCANCES Y LIMITACIONES	6
CAPÍTULO II MARCO TEÓRICO	7
BASES TEÓRICAS.....	7
Sistema de información (SI)	7
Ciclo de vida de desarrollo de sistemas (CVDS).....	9
Metodologías del desarrollo de sistema de información.....	12
Reingeniería del software e ingeniería inversa	18
Herramientas CASE	20
Base de Datos.....	29
Introducción a GeneXus.....	37
Formación profesional y el aprendizaje	49
Universidad Católica del Táchira (UCAT).....	50
MARCO NORMATIVO	53
Los objetivos de la educación superior	53
Los valores, principios y filosofía educativa de la UCAT	53
La misión de la Universidad Católica del Táchira	54
La definición de los programas de computación.....	55
DEFINICIÓN DE TÉRMINOS.....	56
OPERACIONALIZACIÓN DE LAS VARIABLES	61
CAPÍTULO III MARCO METODOLÓGICO	63
NIVEL DE INVESTIGACIÓN	63
DISEÑO DE INVESTIGACIÓN	63
FASE I: DIAGNÓSTICO.....	64
La población del estudio	64

La muestra del estudio	65
Técnicas e instrumento de recolección de datos	67
Análisis e interpretación de resultados.....	68
FASE II: ESTUDIO DE FACTIBILIDAD.....	68
La factibilidad organizacional.....	69
La factibilidad económica.....	70
La factibilidad técnica.....	70
La factibilidad operacional.....	71
FASE III: DESARROLLO DE LA PROPUESTA	71
CAPÍTULO IV ANÁLISIS E INTERPRETACIÓN DE LOS RESULTADOS... 72	
FACTIBILIDAD ORGANIZACIONAL	72
FACTIBILIDAD ECONÓMICA	73
Entrevista al Presidente de la empresa Information System Solution C.A.....	73
Entrevista al Director del Centro de Computación de la UCAT.	77
Análisis de costos/beneficios	78
FACTIBILIDAD TÉCNICA	79
Entrevista al Director del Centro de Computación de la UCAT.	79
FACTIBILIDAD OPERACIONAL	80
Cuestionarios dirigidos a los profesores de la Escuela de Administración y Contaduría Pública en el área de Informática	80
Cuestionarios dirigidos a los alumnos de tercero y cuarto año de Informática Gerencial	89
CAPÍTULO V PROPUESTA..... 93	
DISEÑO DE APLICACIÓN CON GENEXUS	93
La creación de la base de conocimiento.....	95
La creación del modelo de diseño.....	96
La creación de prototipo funcional	108
DESARROLLO DE LA PROPUESTA	117
Objetivos de la propuesta.....	117
Justificación de la propuesta	117
Presentación de la propuesta.....	118
CONCLUSIONES Y RECOMENDACIONES 122	
REFERENCIAS BIBLIOGRÁFICAS 123	
ANEXOS 127	
ANEXO A INFORMACIÓN ACERCA DE GENEXUS 8.0 VERSIÓN DE EVALUACIÓN 2.4	128
ANEXO B INSTRUMENTO DE RECOLECCIÓN DE DATOS PARA LOS PROFESORES.....	132
ANEXO C INSTRUMENTO DE RECOLECCIÓN DE DATOS PARA LOS ALUMNOS.....	136
ANEXO D LA VALIDACIÓN DE LOS CUESTIONARIOS.....	138

LISTA DE CUADROS

<i>Cuadro 1</i>	La evolución de los roles que han tenido los sistemas de información.	8
<i>Cuadro 2</i>	Las organizaciones y los fabricantes de las herramientas CASE.....	28
<i>Cuadro 3</i>	Las principales características de la herramienta GeneXus.	38
<i>Cuadro 4</i>	Los requerimientos mínimos de la herramienta CASE “GeneXus”.	48
<i>Cuadro 5</i>	El programa de estudio de la carrera Informática Gerencial.	52
<i>Cuadro 6</i>	La operacionalización de las variables para los profesores en el área de Informática de la Escuela de Administración y Contaduría Pública de la UCAT.....	61
<i>Cuadro 7</i>	La operacionalización de las variables para los alumnos de tercero y cuarto año de Informática Gerencial.	62
<i>Cuadro 8</i>	La población de los alumnos de tercero y cuarto año de Informática Gerencial.	66
<i>Cuadro 9</i>	El tamaño de la muestra de los alumnos de tercero y cuarto año de Informática Gerencial.....	67
<i>Cuadro 10</i>	Estudio de factibilidad para la incorporación de la herramienta CASE “GeneXus” como software de aprendizaje.....	69
<i>Cuadro 11</i>	La descripción de los requerimientos de las computadoras en los laboratorios de computación de la UCAT.	80
<i>Cuadro 12</i>	Conocimiento de los profesores acerca de las herramientas CASE.....	81
<i>Cuadro 13</i>	Conocimiento de los profesores acerca de los productos de las herramientas CASE en el mercado.	82
<i>Cuadro 14</i>	Conocimiento de los profesores sobre las metodologías de desarrollo de sistemas de información.	83
<i>Cuadro 15</i>	Aceptación por partes de los profesores sobre la incorporación de la herramienta CASE dentro del contenido programático de la carrera Informática Gerencial.....	84
<i>Cuadro 16</i>	Opiniones de los profesores sobre el uso de las herramientas CASE en la formación de los alumnos de Informática Gerencial.	86
<i>Cuadro 17</i>	Opiniones de los profesores acerca de la capacitación en el manejo de herramienta CASE.	88
<i>Cuadro 18</i>	Conocimiento de los alumnos acerca de las herramientas CASE.....	89
<i>Cuadro 19</i>	Conocimiento de los alumnos acerca de los productos de las herramientas CASE en el mercado.	90

<i>Cuadro 20</i> Aceptación por parte de los alumnos para aprender a utilizar una herramienta CASE.	91
<i>Cuadro 21</i> Aceptación por parte de los alumnos para incorporar de la herramienta CASE dentro del contenido programático de la carrera Informática Gerencial.	92
<i>Cuadro 22</i> Los atributos de la estructura de la transacción Clientes	97
<i>Cuadro 23</i> Los atributos de la estructura de la transacción Proveedores.....	99
<i>Cuadro 24</i> Los atributos de la estructura de la transacción Productos.	100
<i>Cuadro 25</i> Los atributos de la estructura de la transacción Facturas (de 2 niveles). 101	
<i>Cuadro 26</i> Los atributos de la estructura de la transacción Pedidos (de 2 niveles)..	102
<i>Cuadro 27</i> Los nuevos atributos de la transacción Clientes.	113
<i>Cuadro 28</i> Los requerimientos de la computadora portátil utilizada durante la creación de varios prototipos funcionales que conforman el programa de aplicación.....	116

LISTA DE GRÁFICOS

<i>Gráfico 1</i> Sistema de Base de Datos.....	29
<i>Gráfico 2</i> Modelos de datos.....	30
<i>Gráfico 3</i> Esquema de una tabla del modelo relacional.....	33
<i>Gráfico 4</i> Ciclos de desarrollo de una aplicación basado en el enfoque evolutivo.....	40
<i>Gráfico 5</i> Pasos para crear aplicación a través de la metodología GeneXus.....	45
<i>Gráfico 6</i> Modelado de la realidad a partir de la visión de usuarios mediante la ingeniería inversa.....	46
<i>Gráfico 7</i> Resultado obtenido sobre el conocimiento de los profesores acerca de las herramientas CASE. Elaborado en SPSS 13.....	81
<i>Gráfico 8</i> Resultado obtenido sobre la aceptación por partes de los profesores sobre la incorporación de la herramienta CASE dentro del contenido programático de la carrera Informática Gerencial. Elaborado en SPSS 13.....	84
<i>Gráfico 9</i> Resultado obtenido sobre las opiniones de los profesores acerca de la capacitación en el manejo de herramienta CASE.....	88
<i>Gráfico 10</i> Resultado obtenido sobre el conocimiento de los alumnos acerca de la herramienta CASE.....	89
<i>Gráfico 11</i> Resultado obtenido sobre la aceptación por parte de los alumnos para aprender a utilizar una herramienta CASE.....	91
<i>Gráfico 12</i> Resultado obtenido sobre la aceptación por parte de los alumno para la incorporación de la herramienta CASE dentro del contenido programático de la carrera Informática Gerencial.....	92
<i>Gráfico 13</i> El diagrama de Entidad–Relación del Sistema de Facturación de Ventas y Registro de Pedidos. Elaborado en Microsoft Office Visio Professional 2003.....	94
<i>Gráfico 14</i> El ambiente de trabajo de la herramienta GeneXus 8.0 Versión de Evaluación 2.4.....	95
<i>Gráfico 15</i> La creación de base de conocimiento.....	95
<i>Gráfico 16</i> La base de conocimiento.....	96
<i>Gráfico 17</i> La creación de un objeto de transacción.....	96
<i>Gráfico 18</i> La estructura de la transacción Clientes.....	97
<i>Gráfico 19</i> La interfaz Form de la transacción Clientes.....	98

<i>Gráfico 20</i>	La interfaz Form personalizada de la transacción Clientes.	98
<i>Gráfico 21</i>	La estructura de la transacción Proveedores.....	99
<i>Gráfico 22</i>	La interfaz Form de la transacción Proveedores.....	100
<i>Gráfico 23</i>	La estructura de la transacción Productos.	100
<i>Gráfico 24</i>	La interfaz Form de la transacción Productos.	101
<i>Gráfico 25</i>	La estructura de la transacción Facturas.....	102
<i>Gráfico 26</i>	La interfaz Form de la transacción Facturas.....	102
<i>Gráfico 27</i>	La estructura de la transacción Pedidos.....	103
<i>Gráfico 28</i>	La interfaz Form de la transacción Pedidos.....	103
<i>Gráfico 29</i>	El modelo de diseño después de crear todas las transacciones.....	107
<i>Gráfico 30</i>	La creación de un modelo de prototipo.	108
<i>Gráfico 31</i>	Los pasos para crear el modelo de prototipo.	108
<i>Gráfico 32</i>	El reporte de análisis de creación de base de datos.	110
<i>Gráfico 33</i>	La ejecución de la reorganización.	110
<i>Gráfico 34</i>	El modelo de prototipo.	111
<i>Gráfico 35</i>	El reporte de la especificación.....	111
<i>Gráfico 36</i>	La generación del programa.	112
<i>Gráfico 37</i>	El modelo de prototipo después de la especificación.	112
<i>Gráfico 38</i>	El programa Developer Menu.	112
<i>Gráfico 39</i>	La ventana de diálogo de ejecución y de la compilación del programa.	113
<i>Gráfico 40</i>	La nueva estructura de la transacción Clientes.....	114
<i>Gráfico 41</i>	La nueva interfaz Form de la transacción Clientes.....	114

UNIVERSIDAD CATÓLICA DEL TÁCHIRA
FACULTAD DE CIENCIAS ECONÓMICAS Y SOCIALES
ESCUELA DE ADMINISTRACIÓN Y CONTADURÍA PÚBLICA
COORDINACIÓN DE FORMACIÓN COMPLEMENTARIA

**Propuesta para la incorporación de la herramienta CASE “GeneXus” en la
formación de los alumnos de Administración mención Informática Gerencial de
la Universidad Católica del Táchira**

Autor: TZUPENG YAN

Tutor: Ing. Heberth Castillo

Año: 2005

RESUMEN

El enfoque del presente investigación es proponer la utilización de las herramientas CASE como software de aprendizaje en la formación de los alumnos de Administración mención Informática Gerencial de la Universidad Católica del Táchira (UCAT). El tipo de investigación es un proyecto factible. La metodología utilizada es iniciarse como exploratorio y terminar como descriptiva. Se realizó un estudio de factibilidades para determinar la posibilidad de la incorporación de dicha herramientas mediante el uso de la encuesta como técnica para la recolección de datos y los instrumentos de recolección de datos fueron cuestionarios y entrevistas semi-estructuradas. La realización de este estudio permitió conocer que la herramienta CASE es el mejor soporte y tiene gran importancia para el proceso del desarrollo de sistemas o software de aplicación. Por lo tanto, su uso ha mejorado la calidad y la productividad del desarrollo de un sistema de información. Se determinó que existe carencia de conocimiento en el área de las herramientas de Ingeniería de Software Asistida por Computadora (CASE) por parte de los estudiantes de la carrera Informática Gerencial. El 95.5% de alumnos de tercero y cuarto año de Informática Gerencial y el 90 % de los profesores en el área de Informática de la Escuela de Administración y Contaduría Pública de la UCAT opinaron que están de acuerdo de que la UCAT incorpore el manejo de la herramienta CASE dentro del contenido programático de la carrera Informática Gerencial. Las computadoras existentes en los laboratorios de computación de la Universidad Católica del Táchira cumplen con los requerimientos mínimos para la instalación de la herramienta GeneXus.

Descriptores: Herramientas CASE, Modelo evolutivo (prototipo funcional), GeneXus.

INTRODUCCIÓN

Hoy en día, el surgimiento de las nuevas tecnologías de la información ha aportado una serie de herramientas para el desarrollo de aplicaciones. Una de ellas es la herramienta de Ingeniería de Software Asistida por Computadora (CASE) que proporciona un conjunto de programas y ayudas que dan asistencia a los analistas y desarrolladores de sistemas, durante todos los pasos del ciclo de vida de desarrollo de un software. Para muchas empresas la introducción de dicha herramienta ha provocado impactos, cambios y mejoras significativas.

En la presente investigación se plantea proponer la utilización de las herramientas CASE como software de aprendizaje en la formación de los alumnos de Administración mención Informática Gerencial.

La estructura del trabajo de grado está dividida en cinco capítulos:

El **Capítulo I** está compuesto por el origen de la investigación, la definición de los objetivos que sirven como guías para lograr la propuesta planteada; y la justificación e importancia que comprende el motivo de la investigación.

El **Capítulo II** se presenta el desarrollo del marco teórico que comprende aspectos teóricos y legales de los conceptos y términos relacionados con el problema en estudio.

El **Capítulo III** se describe la metodología, técnicas y procedimientos utilizados para la ejecución del trabajo de investigación.

El **Capítulo IV** contiene el análisis e interpretación de los resultados del estudio de factibilidad.

El **Capítulo V** está relacionado con la realización de una demostración del diseño de aplicación con la herramienta GeneXus y la presentación de la propuesta para la formación de los alumnos de Informática Gerencial.

CAPÍTULO I

EL PROBLEMA

Planteamiento del problema

La situación del mercado actual, la tecnología, la innovación y los cambios son constantes y vertiginosos día tras día en la era de la informática. El factor globalización ha ganado un lugar a nivel mundial, donde la competencia es fuerte y ha tenido gran impacto en las empresas del mundo. Las compañías que quieren ser competitivas e incrementar sus posiciones en el mercado deben mantenerse actualizadas y adaptarse a las nuevas tecnologías de información.

Por esta razón, las compañías necesitan el sistema de información computacional junto con la tecnología de punta, nuevas herramientas y recursos que brinden mayor capacidad y velocidad para hacer frente a la competencia del mercado regional, nacional e internacional; para solucionar los problemas internos y externos de la empresa. Por consiguiente, se logran alcanzar los objetivos de la empresa lo que acarrea una ventaja competitiva.

La utilización del sistema de información proporciona a las empresas el acceso, la distribución y el uso de información en forma rápida, facilitando de esta manera la toma de decisiones y la realización de cambios oportunos y efectivos. Pero el desarrollo de un sistema de información ha sido un desafío para las organizaciones, tanto la metodología utilizada en sí como los cambios tecnológicos. La minimización de costos resulta muy importante durante todo el proceso de desarrollo de dicho

sistema de información.

La metodología tradicional, conocida como el ciclo de vida de desarrollo de sistemas, en la cual la construcción de un sistema de información requiere cumplir una serie de pasos como la planificación, análisis, diseño, codificación, prueba e implantación lleva una inversión de dinero, tiempo y esfuerzos para alcanzarlo. Lo mismo sucede con el soporte y mantenimiento de un sistema de información. Según ARTech Consultores S.R.L. (2004:4), *GeneXus Visión General* “La mayoría de las empresas que trabajan de una manera convencional se admite que el 80% de los recursos que teóricamente están destinados al desarrollo, realmente se utilizan para hacer mantenimiento de las aplicaciones ya implementadas”.

De acuerdo con los autores Kendall, K.E. y Kendall, J.E. (1997:13) “En los últimos años los analistas han comenzado a beneficiarse de nuevas herramientas de productividad que han sido creadas implícitamente para mejorar su trabajo rutinario mediante un apoyo automatizado. A estas se les llama Herramientas CASE”.

Dicha herramientas han favorecido a muchas empresas en la automatización de los aspectos claves en el desarrollo de sistema de información, ofreciendo la portabilidad de las aplicaciones entre distintas plataformas; y la flexibilidad de realizar cambios sin tener que regresar a etapas anteriores. Lo más importante es la creación de sistemas de información o aplicaciones de manera eficiente y rápida sin necesidad de programar, evitándose de esta manera dedicarse a tareas rutinarias y pesadas.

La Universidad Católica del Táchira (UCAT), como institución de educación superior, toma la metodología tradicional como base para la formación de los estudiantes de la carrera de Licenciatura en Administración mención Informática Gerencial en las materias relacionadas con Sistema de Información. En los trabajos presentados como tesis de grado, se ha podido observar que los mismos, en su mayoría, alcanzan las etapas de análisis, diseño, dejando a un lado la etapa de desarrollos de

sistemas, que involucra la codificación y depuración de programas.

La circunstancia anterior es consecuencia de la carencia de conocimiento concerniente a las herramientas CASE que se aplica en la enseñanza de los alumnos de las carreras relacionadas con Informática; y poco empleo acerca de las metodologías alternativas para el desarrollo de sistemas de información o de aplicaciones. La Universidad Católica del Táchira debería adoptarse a estas herramientas como software didáctico e integrarse en el contenido programático de la carrera de Administración mención Informática Gerencial para que este al día a la demanda y exigencia de campo laboral. Por esta razón, el objeto de estudio está centrado en:

¿Cuáles son las características y las funcionalidades de las herramientas CASE?

¿Cuáles son los requerimientos que deben cumplir para la incorporación de la herramienta CASE “GeneXus” en la formación de los alumnos de Informática Gerencial de la UCAT?

¿Cómo crea un software de aplicación utilizando la herramienta CASE “GeneXus”?

Objetivos de la investigación

Objetivo General

Proponer la incorporación de la herramienta CASE “GeneXus” en la formación de los alumnos de Administración mención Informática Gerencial de la Universidad Católica del Táchira.

Objetivos Específicos

1. Indagar las características y las funciones de la herramienta CASE “GeneXus”.
2. Determinar la factibilidad de la incorporación de la herramienta CASE “GeneXus” como software de aprendizaje en la formación de los alumnos de Administración mención Informática Gerencial.
3. Desarrollar un modelo funcional de un programa de aplicación sencillo utilizando la herramienta CASE “GeneXus”.
4. Elaborar una propuesta para la incorporación del uso de la herramienta CASE “GeneXus” dentro del contenido programático de dicha carrera.

Justificación e Importancia

La creación de un sistema o una aplicación basada en la metodología tradicional requiere tiempo para desarrollarlo, especialmente en la etapa de codificación y prueba de un programa. El uso de las herramientas CASE que se basan en la automatización de todo el ciclo del desarrollo de sistemas de información han mejorado la calidad y productividad de dicho proceso de desarrollo, siendo esto posible, gracias a los avances tecnológicos. Es por ello, que la utilización de estas herramientas facilita el proceso de creación de sistema convirtiendo en una tarea sencilla y simple. Permite que el analista o desarrollador de sistemas enfoque su trabajo en la etapa de análisis y diseño.

Según Centro de Estudios y Diseño de Sistemas (CEDDS, 2005) se ha demostrado que la productividad de un analista que lleva a cabo su análisis con la ayuda del CASE es superior al 30%. Esta ganancia de productividad permite prácticamente por sí sola la recuperación de las inversiones llevadas a cabo en CASE en menos de 18 meses. A nivel de costos se estima una reducción de más del 30% en los costos totales de un proyecto realizado con una metodología de desarrollo soportada por herramientas CASE con generador automático de código. La repercusión en los costos del mantenimiento también es muy alta. El uso de las herramientas CASE permite

reducirlos en más del 50%, incluso en algunas empresas supere el 70%.

Por consiguiente, el uso de las herramientas CASE tiene gran importancia para el ciclo de desarrollo de sistemas de información. La incorporación de estas herramientas como una opción alternativa en la enseñanza de los alumnos de la carrera Informática Gerencial reforzaría los conocimientos teóricos y las capacidades en la práctica de construcción de sistemas de información, logrando así tanto la obtención de experiencias como un soporte para su preparación profesional.

Asimismo, la Universidad Católica del Táchira podría mejorar la calidad de educación como una fortaleza indiscutible comparando con otras instituciones de educación superior a nivel nacional, lo que beneficiaría en forma directa a los estudiantes y los profesores con la utilización de este tipo de herramientas. Lamentablemente, son pocas las personas que las conocen. Por esta razón, la investigación surge con el propósito de aportar el conocimiento y la importancia acerca de las herramientas CASE y la posibilidad de incorporarlas dentro de algún contenido programático de dicha carrera.

Alcances y Limitaciones

El alcance primordial es poder desarrollar una aplicación sencilla que sirva como un prototipo que muestra el uso de nueva tecnología en el diseño y desarrollo de sistemas de información, y que puede seguir como pauta para desarrollar sistema de información en la cátedra correspondiente o bien, la difusión de su uso en cátedras similares.

Durante la creación de varios prototipos funcionales que conforman el programa de aplicación sencilla, se utilizó la versión de evaluación 2.4 de la herramienta GeneXus 8.0, la cual está sujeta a ciertas limitaciones y restricciones (Ver el anexo A).

CAPÍTULO II

MARCO TEÓRICO

Bases Teóricas

Sistema de información (SI)

Como se mencionó en el planteamiento del problema, los cambios tecnológicos y la globalización han tenido un fuerte impacto en las organizaciones. Estos factores han obligado a las compañías a mantenerse actualizadas, y por ende, adaptarse a las nuevas tecnologías de información en aras de hacer frente a la competencia de los negocios.

El sistema de información surge como una respuesta para dicha competencia. Su propósito es satisfacer las necesidades de información de una organización, brindando facilidades en la toma de decisión por parte de gerentes de diferentes rangos jerárquicos.

¿Qué es un sistema de información? Es un conjunto de elementos o componentes que se interrelacionan entre si para recolectar o capturar, manipular y diseminar datos e información; además cuenta con un mecanismo de retroalimentación para el cumplimiento de un objetivo de la organización.

Según la definición de los autores Whitten, J. L., Bentley, L. D. y Barlow, V. M. (1996:49):

Un sistema de información es una disposición de personas, actividades, datos, redes y tecnología integrados entre sí con el propósito de apoyar y mejorar las operaciones cotidianas de una empresa, así como satisfacer las necesidades de información para la resolución de problema y la toma de decisiones por parte de los directivos de la empresa.

Si se observa la tendencia en los sistemas de información, se puede decir que su desempeño ha venido cambiando en el transcurso del tiempo. Según el autor O'Brien (2001:) las principales funciones que desempeñan los sistemas de información son:

1. Soporte para el proceso y las transacciones empresariales.
2. Soporte para la toma de decisiones.
3. Soporte estratégico para lograr ventaja competitiva.
4. Sistemas de información interconectados en red empresarial y global.

En la siguiente tabla se resume la evolución de los roles que han tenido los sistemas de información:

Cuadro 1 La evolución de los roles que han tenido los sistemas de información.

Tiempo	Los roles que desempeñan los sistemas de información	Sistemas de información que representan
1950-1960	Procesamiento de datos	Sistemas de procesamiento electrónico de datos.
1960-1970	Elaboración de informes gerenciales	Sistemas de información gerencial.
1970-1980	Soporte a la toma de decisiones	Sistemas de soporte a las decisiones.
1980-1990	Soporte estratégico y al usuario final	Sistemas de computación para el usuario final. Sistemas de información ejecutivo. Sistemas expertos. Sistemas de información estratégica.
1990-2000	SI interconectado en red empresarial y global.	Sistemas de información interconectados en red.

Ciclo de vida de desarrollo de sistemas (CVDS)

Al igual que los seres vivos, un sistema de información tiene proceso de nacimiento, crecimiento, maduración y muerte; este proceso representa su vida útil, llamándose de esta forma ciclo de vida de un sistema.

Definición del ciclo de vida de desarrollo de sistemas

El ciclo de vida de un sistema de información se entiende como las etapas que tienen lugar desde la identificación de problemas, oportunidades hasta la obsolescencia completa del sistema. Asimismo debido a los cambios tecnológicos y de las necesidades de los usuarios del sistema. El sistema de información debe realizar ajuste para adaptarse de acuerdo con dichos cambios y dichas necesidades.

Según los autores Whitten, J. L., Bentley, L. D. y Barlow, V. M. (1996:95) “Es un proceso por el cual los analistas de sistemas, los ingenieros de software, los programadores y los usuarios finales elaboran sistemas de información y aplicaciones informáticas. IBM y otras compañías refieren este término como ciclo de desarrollo de aplicaciones”.

Las etapas del ciclo de vida de desarrollo de sistemas

Durante el ciclo de vida de un sistema de información, se establece un conjunto de etapas o fases para orientar al personal encargado en el desarrollo de un sistema de información. Existen diferencias entre varios autores sobre las etapas que cubre el ciclo de vida de desarrollo de un sistema de información, pero todos siguen un enfoque organizado.

Por lo general, las etapas del ciclo de vida de desarrollo de un sistema de información son: la planificación de sistemas, análisis de sistemas, diseño de sistemas,

implantación de sistemas y el soporte y mantenimiento de sistemas.

Planificación de sistemas. En esta fase de planificación, se cubren las actividades de la identificación de problemas, oportunidades, objetivos y las actividades correspondientes al establecimiento de recursos humanos, financieros, tiempo, tecnología requerida y cronograma de actividades para implantar un sistema de información. De esta manera, permite controlar el desarrollo de un sistema estableciendo pautas y normas para que se pueda construir un sistema en un tiempo establecido con los recursos disponibles. Es de vital importancia destacar que el éxito o fracaso para el resto de las fases, dependerá del desarrollo de esta fase.

Análisis de sistemas. En la fase de análisis de sistemas está involucrado el análisis de las necesidades del sistema. También está relacionada con la determinación de los requerimientos de información. Se pretende obtener la mayor información acerca del contexto organizacional en el cual funcionará el sistema de información. Es decir, se realiza con el fin de comprender qué información necesitan los usuarios para realizar su trabajo.

El uso de diagrama de flujo de datos puede ayudar al personal que hace el análisis de sistema cumplir dicha fase de desarrollo ya que de esta manera se estructura en forma gráfica la entrada, proceso y salida de las funciones del negocio.

Diseño de sistemas. Se refiere a una serie de pasos detallados para construir el sistema basado en la información recolectada anteriormente. Se realiza el diseño lógico de la interfaz de usuario, el archivo, la salida de información que sea por la pantalla o la impresora y el procedimiento de salvaguarda y respaldo de información. Asimismo se crea la base de datos que soportará al sistema de información.

En esta fase, se hace el uso de lenguajes de programación de tercera (3GL) o de cuarta generación (4GL). Se concentra su trabajo en diseñar, codificar, corregir los

errores de sintaxis de codificación y realizar prueba de programa diseñado, dejando listo para la implantación de sistemas. También se desarrolla la documentación del uso de software o el manual de procedimientos.

Implantación de sistemas. Se pone en funcionamiento el sistema de información nuevo o mejorado. Si se va a implantar un sistema de información mejorado, se realiza la conversión del sistema antiguo al sistema nuevo. El sistema es instalado para ser utilizado por parte de usuarios en su trabajo diario. A su vez, se hacen las actividades de entrenamiento al usuario para que pueda manejar el sistema de manera eficiente. Dependiendo del tipo de metodología empleada, especialmente la basada en prototipos, no se requiere entonces mucho tiempo para realizar dicho entrenamiento.

Soporte y mantenimiento de sistemas. Después de ser instalado el sistema, pueda que existen posibilidades de encontrar errores y por ende se deba realizar la modificación y la actualización del sistema. Como también la intención de mejorar la capacidad del software por razón de las necesidades de la empresa o por los cambios tecnológicos. Según los autores Kendall, K.E. y Kendall, J.E. (1997:12) “La estimación de tiempo gastado por los departamentos en mantenimiento ha ido de 48 al 60 por ciento del tiempo total empleado en el desarrollo de sistemas”. Esta fase es un proceso continuo a lo largo del ciclo de desarrollo de un sistema de información.

Metodologías del desarrollo de sistema de información

Existen diversos enfoques que se toman como metodología para desarrollar un sistema de información. De acuerdo con el autor Barrios (2001:41), “Un enfoque es un punto de vista, un modo de ver. Podría traducirse como el modo de estructurar y ejecutar las actividades que llevan al logro de un objetivo”.

En varias bibliografías consultadas, la ingeniería de la programación contempla cinco enfoques básicos para el desarrollo de sistemas de información que son el tradicional, el de análisis estructurado, el de prototipos, el evolutivo y el en espiral.

Enfoque tradicional o en cascada

Según el autor Barrios (2001:41) “Consiste en la realización, secuencial, de cada una de las fases del ciclo de desarrollo de sistemas, se caracteriza por la forma exhaustiva y detallada en que se completa cada una de las fases”. Es decir antes de avanzar a la siguiente fase, cada fase culminada debe ser revisada y corregida. Este enfoque es poco flexible para realizar revisiones de las fases anteriores ya que sólo se realiza cambios en los cuales no pueden ser profundos o radicales.

Enfoque de análisis estructurado

Según el autor Senn (1992:38) El enfoque de análisis estructurado surge por la finalidad de superar la dificultad de comprender de manera completa sistemas grandes y complejos. El método utilizado en dicho enfoque es dividir el sistema en componentes y construir un modelo del sistema mediante análisis y diseño estructurado.

El análisis estructurado: se concentra en especificar lo que se requiere que haga el sistema o la aplicación. No se establece cómo se cumplirán los requerimiento o la

forma en que implantará la aplicación. Más bien permite que las personas observen los elementos lógicos separados de los componentes físicos. Después de esto se puede desarrollar un diseño físico eficiente para la saturación donde será utilizado.

Los elementos del análisis estructurado son símbolos gráficos, diagramas de flujo de datos y el diccionario de datos. En vez de las palabras el análisis estructurado se utiliza símbolos o íconos para crear un modelo gráfico del sistema.

El diagrama de flujo de datos: muestra las fuentes y destinos de los datos, identifica y da nombre a los grupos de datos que relacionan una función con otra y señala los almacenes de datos a los que se tiene acceso.

El diccionario de datos: todas las definiciones de los elementos en el sistema como e flujo de datos, procesos y almacenes de datos están descritos en forma detallada en el diccionario de datos.

El diseño estructurado: es una técnica específica para el diseño de programas que se enfoca en la especificación funcional del software empleando la descripción gráfica. La meta del diseño es crear programas formados por módulos independientes unos de otros antes del comienzo de la fase de escritura de código. Es importante señalar que esta etapa de diseño estructurado no sólo conduce hacia mejores programas sino también facilita el mantenimiento de los mismos.

El análisis estructurado se combina con frecuencia con el enfoque tradicional, pero muchas organizaciones optan por no utilizar este enfoque debido a que el desarrollo de diagramas y esquemas se consume mucho tiempo, especialmente cuando el sistema es grande y complejo. Otros autores señalan que faltan los elementos como las personas y los procedimientos de control, son parte del sistema mismo que no se puede omitirse en la descripción de éste.

Enfoque de prototipos

Este enfoque está basado en el concepto de prototipo o modelo a escala del sistema de información. Según la Biblioteca de Consulta Encarta (2004)

Un prototipo es un modelo o versión inicial de un producto, previsto para probar y desarrollar el diseño. Antes de invertir en el equipo necesario para fabricar en serie un producto, el fabricante debe estar convencido de que el diseño es seguro y fiable.

Se muestra las funciones del sistema antes de que éste sea construido, de esta manera permite a los usuarios observar la interfaz. El personal encargado de desarrollar el sistema tiene muchos contactos con los usuarios, logrando así que el usuario aporte sugerencias para mejorar la interacción del sistema y corroborar las especificaciones de requerimiento.

En la construcción del prototipo no solo se toma en cuenta los requerimientos del usuario en lo correspondiente a modos de interacción, sino también en la parte relacionada con las funciones que el sistema debe ejecutar. A tales efectos, se ahorra tiempo en el entrenamiento del personal que va a usar el sistema, puesto que el usuario conoce de antemano el funcionamiento del sistema.

El área que corresponde a la creación de prototipos dentro del ciclo de vida de desarrollo de sistemas de información es parte de la fase de definición de requerimientos y parte de la fase de diseño del sistema de información.

En este sentido, el autor Barrios (2001:43) plantea que el prototipo puede ser incluido en el ciclo de desarrollo de sistemas de información, de dos modos distintos, según sean las necesidades:

1. Como modelo desechable que sólo muestra la posible interfaz del sistema que va a construir, el cual una vez cumplido su objetivo de ajuste y

refinamiento de requerimientos se descarta y se comienza la construcción del verdadero sistema.

2. Como parte utilizable en el sistema real, la cual una vez cumplido su objetivo como prototipo, se refina y se reajusta y pasa a formar parte del SI que se está construyendo.

Al final, una vez construido el prototipo, el usuario podrá colaborar con el desarrollador del sistema sobre posibles inconvenientes en el uso de sistema, señalar dificultades de trabajo y corregir fallas en la definición y en el establecimiento de requerimientos o por malos entendidos entre el usuario y el desarrollador.

Este mismo autor señala que los tipos de herramientas disponibles para la construcción del prototipo pueden ser:

1. En papel: se dibujan las pantallas con todos los detalles de diseño pertinentes y se acompañan de un diagrama de flujo de pantallas que muestran la relación y el camino existente entre ellas.
2. Automatizado de pantallas: por medio de un generador de pantallas, se construye un prototipo que permite moverse de una pantalla a otra simulando un sistema funcional, pero sin ejecutar verdaderamente ninguna actividad sobre los datos.
3. Automatizado funcional: es cuando al prototipo de pantallas se le agregan facilidades funcionales, dotándolo de procesos sobre los datos. Algunos autores lo llaman prototipo versión cero o versión inicial, en muchos casos es el inicio de un conjunto de versiones del sistema de información. Este proceso de refinamiento culminará con el sistema de información deseado por el usuario que cumple con todos los requerimientos establecidos. Este tipo de prototipos constituyen la base del enfoque evolutivo.

Enfoque evolutivo

En el enfoque evolutivo, el modelo de desarrollo evolutivo asume que los requerimientos no son completamente conocidos al inicio del proyecto debido al cambio constante de los requerimientos de los usuarios.

Según el autor Barrios (2001:47) “Este enfoque está basado en el uso de prototipos funcionales. El prototipo inicial o versión cero se va refinando y ajustando según los requerimientos del usuario, dando origen a una nueva versión del sistema”. Por tanto, el prototipo va adquiriendo poco a poco la estructura y capacidades que satisfacen a los requerimientos que propiciaron de desarrollo de un sistema de información.

Según el mismo autor existen dos modos de aplicarlo al desarrollo de proyectos de sistemas de información:

1. Cada versión es estrictamente una mejora de la anterior. Se construye el prototipo inicial después de un análisis y definición de requerimientos superficiales; se evalúa ante el usuario y según sus requerimientos se le hacen modificaciones, surgiendo una segunda versión a la cual se le hace el mismo proceso que a la anterior; se continúa modificando cada versión hasta que el sistema sea aceptado por el usuario como definitivo.
2. Se diseñan todas las versiones del sistema y luego se van construyendo una por una. Cada versión es complemento de la anterior; esto quiere decir que se construye una versión, se evalúa ante el usuario, se aceptan las modificaciones y se construye la siguiente versión tomando en cuenta, no sólo las especificaciones iniciales para la versión, sino las sugerencias del usuario a la versión previa. Al inicio del proyecto, se determinan cuántas versiones se construirán y cuáles son las capacidades funcionales y de interacción de cada una de ellas. Generalmente

se utiliza cuando el sistema requerido es muy grande y complejo; al dividirlo en subsistema que dan origen a cada versión del sistema, se va desarrollando en forma evolutiva. Cada versión agrega más funciones al sistema previo hasta completarlo.

Enfoque en espiral

El enfoque en espiral es desarrollado por Bohem en 1988 basado en el desarrollo evolutivo que acompaña a la naturaleza interactiva de construcción de prototipos con los aspectos controlados y sistemáticos del enfoque en cascada, y al mismo tiempo añadiendo un nuevo elemento: el análisis de riesgo. Es importante señalar que dicho enfoque no es una alternativa del enfoque tradicional, ellos son completamente compatibles.

Para crear un software de aplicación, el enfoque está representado mediante la espiral que se desarrolla en una serie de versiones incrementales. Cada iteración es una vuelta que se divide en un número de actividades estructurales llamadas regiones de tareas que puede ser de tres a seis.

1. Definición de objetivos, alternativas y restricciones.
2. Evaluación de las diferentes alternativas.
3. Análisis de riesgo: análisis de alternativas e identificación/resolución de riesgos.
4. Desarrollo, verificación y producto.
5. Planificaciones próximas.

Cada región tiene tareas que se adaptan a las características del proyecto.

El primer circuito de la espiral produce una especificación de productos; los pasos siguientes en la espiral se podrían usar para desarrollar un prototipo y progresivamente versiones más sofisticadas del software. Cada paso por la región de planificación produce ajustes en el plan del proyecto. El costo y la planificación se

ajustan según la reacción ante la evaluación del cliente. A diferencia del enfoque tradicional que termina cuando se entrega el software, el enfoque en espiral se puede adaptar y aplicarse a lo largo de toda el ciclo de vida del software.

Reingeniería del software e ingeniería inversa

La reingeniería de software y la ingeniería inversa son métodos para extender la vida de programas antiguos. Se usan software de Reingeniería Asistida por Computador (CARE) para reestructurar el código de computadora existente.

Definición de la reingeniería del software

La reingeniería del software es la tecnología que surge mediante la aplicación de las técnicas de inteligencia artificial y matemática sofisticada para el análisis automatizado y la modificación del código fuente de programas, con el fin de abreviarlo y hacerlo más eficiente.

Según Isabel y Galipienso (2004:9) “Es el proceso de reorganizar y modificar sistemas software existentes para hacerlos más mantenibles. Se re-estructuran o re-escriben partes o la totalidad de un sistema heredado (*legacy system*) sin cambiar su funcionalidad”.

Legacy system según Wikipedia (2005), Traducción al español: “Un sistema heredado es un sistema de la computadora anticuado o programa de la aplicación que continúa siendo usado porque el usuario (típicamente un organización) no quiera reemplazar o rediseñarlo”.¹

¹ Original en inglés: “A legacy system is an antiquated computer system or application program which continues to be used because the user (typically an organization) does not want to replace or redesign it”.

Es aplicable cuando requiere un mantenimiento frecuente en alguna de las partes de un sistema y también cuando se dispone de herramientas automáticas para reestructurar el sistema. La reingeniería implica un esfuerzo añadido para conseguir que el sistema sea más fácil de mantener. El sistema puede ser re-estructurado y re-documentado.

La importancia de la reingeniería

La importancia de la reingeniería según (“Mantenimiento de sistema”, 2003:8) son las siguientes:

1. Puede reducir los riesgos evolutivos de una organización.
2. Puede ayudar a las organizaciones a recuperar sus inversiones en software.
3. Puede hacer el software más fácilmente modificable.
4. Amplía las capacidades de las herramientas CASE.
5. Es un catalizador para la automatización del mantenimiento del software.
6. Puede actuar como catalizador para la aplicación de técnicas de inteligencia artificial (IA) para resolver problemas de reingeniería.

Definición de la ingeniería inversa

Este método es denominado ingeniería inversa porque avanza en dirección opuesta a las tareas habituales de ingeniería (directa) que corresponde al desarrollo de software tradicional. Los productos más comunes que son sometidos a la ingeniería inversa son los programas de computadoras y los componentes electrónicos. La definición de ella se encuentra:

Según (“Mantenimiento del Software”, 2003:11) “La ingeniería inversa es el proceso de análisis de un sistema para identificar sus componentes e interrelaciones y crear representaciones del sistema en otra forma o a un nivel más alto de abstracción”.

Asimismo, según los autores Whitten, J. L., Bentley, L. D. y Barlow, V. M. (1997:188) “La ingeniería inversa es el proceso de análisis del código de los programas de aplicación y las bases de datos existentes para crear representaciones de alto nivel de dicho código. A veces recibe también el nombre de recuperación de diseño”.

Herramientas CASE

Se puede definir a las herramientas CASE como un conjunto de programas que ayuda a los analistas, ingenieros de software y desarrolladores de sistemas, durante todos los pasos del ciclo de vida de desarrollo de sistemas de información.

La definición de herramientas CASE según la Biblioteca de Consulta Encarta (2004):

CASE, en informática, acrónimo de Computer Aided Software Engineering, que define el uso de computadoras para el desarrollo de software en todo su ciclo de vida. Es un entorno de trabajo asistido por computadora que ayuda tanto a jefes de proyecto como a analistas y programadores en el diseño e implementación de todo tipo de software.

Historia de las herramientas CASE

Las primeras herramientas CASE fueron creadas al inicio de la década de los ochenta, fueron un simple procesador de palabras que ayudaba a crear y manipular la documentación; y una herramienta de diagramación asistida por computadora que ayudaba a graficar las diagramas de flujo de datos, permitiendo que los diagramas puedan ser creados y modificados fácilmente.

A inicios de los ochenta, se lanzó la primera herramienta comercial, aproximadamente, en el año 1982, pero fue hasta 1985 que las herramientas CASE se volvieron realmente importantes para el desarrollo de software. Al mediano de los años 80, surge el concepto de repositorio como núcleo de un entorno CASE, así como

generador automática de código que automatizan gran parte de las últimas fases del ciclo de vida.

A mediados de los noventa esta tecnología entró en su fase de madurez, siendo capaz de resolver problemas asociados al desarrollo de sistemas de información y contribuyendo de esta manera a aumentar la productividad y mejorar la calidad en el desarrollo de tales sistemas. Hoy en día, hay diferentes instituciones que se dedica a realizar investigación sobre dicha herramienta. La industria de Herramientas CASE está creciendo y esta tomando cada vez mayor importancia.

Clasificación de las herramientas CASE

No existe una sola clasificación de herramientas CASE y es difícil incluirlas en una clase determinada. Según Universidad Nacional de Colombia (UNC, 2004), Para clasificarse se toma en cuenta a:

1. Las plataformas que soportan.
2. Las fases del ciclo de vida del desarrollo de sistemas que cubren.
3. La arquitectura de las aplicaciones que producen.
4. Su funcionalidad.

A. Las herramientas CASE, en función de *las fases del ciclo de vida abarcadas*, se pueden agrupar de la forma siguiente:

1. **Herramientas integradas, I-CASE (Integrated CASE, CASE integrado):** son aquellas herramientas que abarcan todas las fases del ciclo de vida del desarrollo de sistemas. Generalmente las herramientas integradas se basan en una metodología. Tienen un repositorio, es decir una base de datos y aportan técnicas para todas las fases del ciclo de vida.

2. **Herramientas de alto nivel, U-CASE (Upper CASE - CASE superior):** son aquellas herramientas que automatizan y apoyan las primeras fases del ciclo de vida del desarrollo de sistemas como el análisis y diseño.
3. **Herramientas de bajo nivel, L-CASE (Lower CASE - CASE inferior):** son aquellas herramientas dirigidas a las últimas fases del desarrollo de sistemas como la fase de desarrollo y de implantación. Ej. La generación del código fuente de computadora.
4. **Juegos de herramientas o CASE cruzado del ciclo de vida:** son el tipo más simple de herramientas CASE que automatizan una fase dentro del ciclo de vida. En ellos se encontrarían las herramientas de reingeniería, orientadas a la fase de mantenimiento.

B. La clasificación de las herramientas CASE, según *la funcionalidad* como criterio principal, es la siguiente:

1. **Herramientas de planificación de sistemas de gestión:** son las que proporciona una ayuda importante para modelizar los requisitos de información estratégica de una organización. Su objetivo principal es ayudar a comprender mejor cómo se mueve la información entre las distintas unidades organizativas.
2. **Herramientas de Análisis y Diseño:** son las herramientas que permiten al desarrollador crear un modelo del sistema que proporcionan un grado de confianza en la representación del análisis y ayudan a eliminar errores con anticipación. Entre ellas se puede encontrar:
 - a. Herramientas de análisis y diseño.
 - b. Herramientas de creación de prototipos y de simulación.
 - c. Herramientas para el diseño y desarrollo de interfaces.

3. **Herramientas de programación:** son los compiladores, los editores y los depuradores de los lenguajes de programación convencionales. Los ejemplos de estas herramientas son:
 - a. Herramientas de codificación convencionales.
 - b. Herramientas de codificación de cuarta generación.
 - c. Herramientas de programación orientadas a objetos.

4. **Herramientas de integración y prueba:** son las herramientas que ayudan a la adquisición, medición, simulación y prueba de los equipos lógicos desarrollados. Entre las más utilizadas están:
 - a. Herramientas de análisis estático.
 - b. Herramientas de generación de casos de prueba.

5. **Herramientas de gestión de prototipos:** son ampliamente utilizados en el desarrollo de aplicaciones tanto para la evaluación de especificaciones de un sistema de información como para un mejor entendimiento de cómo los requisitos de un sistema de información. De esta manera se ajustan a los objetivos perseguidos.

6. **Herramientas de mantenimiento:** la categoría de herramientas de mantenimiento se puede subdividir en:
 - a. Herramientas de ingeniería inversa.
 - b. Herramientas de reestructuración y análisis de código.
 - c. Herramientas de reingeniería.

7. **Herramientas de gestión de proyectos:** la mayoría de las herramientas CASE de gestión de proyectos, se centran en un elemento específico de la gestión del proyecto, en lugar de proporcionar un soporte global para la actividad de gestión. Utilizando un conjunto seleccionado de las mismas se puede: realizar estimaciones de esfuerzo, coste y duración, hacer un seguimiento continuo del

proyecto, estimar la productividad y la calidad, etc. Existen también herramientas que permiten al comprador del desarrollo de un sistema, hacer un seguimiento que va desde los requisitos del pliego de condiciones técnicas inicial, hasta el trabajo de desarrollo que convierte estos requisitos en un producto final. Se incluyen dentro de las herramientas de control de proyectos las siguientes:

- a. Herramientas de planificación de proyectos.
- b. Herramientas de seguimiento de requisitos.
- c. Herramientas de gestión y medida.

8. **Herramientas de soporte:** se engloban las herramientas que recogen las actividades aplicables en todo el proceso de desarrollo, como las que se relacionan a continuación:

- a. Herramientas de documentación.
- b. Herramientas para software de sistemas.
- c. Herramientas de control de calidad.
- d. Herramientas de bases de datos.

Componentes y funcionalidades de una herramienta CASE

Según Instituto Nacional de Estadística e Informática (INEI, 1999:25) A continuación se describen los principales componentes de una herramienta CASE y sus funcionalidades.

1. **Repositorio:** Base de datos central de una herramienta CASE. El repositorio amplía el concepto de diccionario de datos para incluir toda la información que se va generando a lo largo del ciclo de vida del sistema.
2. **Módulos de diagramación y modelización:** Algunos de los diagramas y modelos utilizados con mayor frecuencia son:

- a. Diagrama de flujo de datos.
 - b. Modelo entidad - interrelación.
 - c. Historia de la vida de las entidades.
 - d. Diagrama Estructura de datos.
 - e. Diagrama Estructura de cuadros.
 - f. Técnicas matriciales.
3. **Herramienta de prototipazo:** El objetivo principal de esta herramienta es poder mostrar al usuario, desde los momentos iniciales del diseño, el aspecto que tendrá la aplicación una vez desarrollada. Ello facilitará la aplicación de los cambios que se consideren necesarios, todavía en la fase de diseño.
4. **Generador de código:** Normalmente se suele utilizar sobre computadoras personales o estaciones de trabajo, por lo que el paso posterior del código al host puede traer problemas, al tener que compilar en ambos entornos.
5. **Módulo generador de documentación:** El módulo generador de la documentación se alimenta del repositorio para transcribir las especificaciones allí contenidas.

Beneficios y debilidades del uso de las herramientas CASE

Los beneficios que ofrecen las herramientas CASE según INEI (1999:10) Estas herramientas pueden proveer muchos beneficios en todas las etapas del proceso de desarrollo de software, algunas de ellas son:

1. Verificar el uso de todos los elementos en el sistema diseñado.
2. Automatizar el dibujo de diagramas.
3. Ayudar en la documentación del sistema.
4. Ayudar en la creación de relaciones en la Base de Datos.

5. Generar estructuras de código.

De acuerdo con los autores Kendall, K.E. y Kendall, J.E. (1997:14) “Los analistas se apoyan en las herramientas CASE para aumentar la productividad, comunicarse más efectivamente con los usuarios e integrar el trabajo que realizan en el sistema, desde el principio hasta el fin del ciclo de vida”.

En cuanto a las debilidades de herramientas CASE, los costos asociados a la adquisición de dichas herramientas y el entrenamiento de personal algunas veces puede ser bastante elevado, dependiendo de tipo de herramientas que escoge la empresa.

Según el autor Barrios (2001:56) “El problema esencial de las herramientas CASE es el hecho de estar amarrado a un modo de trabajo o metodología, restringiendo la libertad de programador de seleccionar sus propios métodos y técnicas de desarrollo”. Pero, son ideales para estandarizar los métodos y técnicas de desarrollo de sistema dentro de una organización, logrando así que los elementos involucrados y la documentación durante el desarrollo sean uniformes para todos los sistemas computarizados de dicha organización.

Evolución y línea futura de las herramientas CASE

Según UNC (2004), las principales líneas de evolución hacia las que parecen encaminarse las herramientas CASE son las siguientes:

1. **CASE para sistemas bajo arquitectura cliente/servidor.** Para facilitar una mayor liberalización del programador de todos los aspectos que no sean propios de la aplicación (protocolos de red, seguridad, etc.).
2. **CASE multiplataforma.** Estas herramientas soportan las combinaciones dominantes de diferentes plataformas físicas, sistemas operativos, interfases

gráficas de usuario, sistemas de gestión de bases de datos, lenguajes de programación y protocolos de red. En este sentido el futuro podrá ser de apertura creciente a nuevas plataformas y portabilidad más generalizada.

3. **CASE para ingeniería inversa y directa.** Ya existen algunas herramientas de este tipo. Su evolución marcará notables mejoras en la obtención de los diseños a partir del código ya existente (ingeniería inversa) y la regeneración del mismo, una vez optimizado el diseño (ingeniería directa).
4. **CASE para trabajo en grupo (groupware).** Estas herramientas se centran en el proceso de desarrollo más que en el producto a desarrollar, facilitando la integración de diferentes grupos humanos, pertenecientes incluso a empresas diferentes, trabajando conjuntamente en un gran proyecto. Deberían incorporar las facilidades clásicas como correo electrónico, calendarios en línea, planificación de actividades, preparación de documentos, actas de reuniones y entre otras.
5. **CASE para desarrollo de sistemas orientados a objetos.** En la actualidad existen algunas herramientas que cubren alguna de las fases del ciclo de vida de desarrollo de aplicaciones orientadas a objetos. El objetivo futuro podría ser cubrir el ciclo de vida completo. Aunque hoy en día, la mayor efectividad se consigue con las herramientas CASE para métodos estructurados, en un futuro no muy lejano esta situación se invertirá a favor de las que soportan objetos.

Organizaciones y fabricantes de las herramientas CASE

Según CEDS (2005) *Herramientas CASE*, en la siguiente la tabla se presenta algunas organizaciones y fabricantes más populares que se dedican a proponer o desarrollar las herramientas CASE:

Cuadro 2 Las organizaciones y los fabricantes de las herramientas CASE.

Organizaciones de las herramientas CASE	
International Council on Systems Engineering (INCOSE)	INCOSE es una de las más importantes organizaciones mundiales en Ingeniería de Sistemas en cuyo ámbito están integradas las herramientas CASE. http://www.incose.org/
Evalu8IT.com	Estudios y evaluación sobre las herramientas CASE. http://www.evalu8it.com/swcase.html
Fabricantes de las herramientas CASE	
ARTech Consultores S.R.L.	GeneXus 8.0 Ver introducción a GeneXus. http://www.genexus.com/
Excel Software	Win A&D 4.0 Herramientas CASE para análisis y diseño, incluye técnicas estructuradas y orientadas a objetos. http://www.excelsoftware.com/
Grandite	SILVERRUN Conjunto integrado de de herramientas CASE para el modelado de negocios. http://www.silverrun.com/
Microsoft Corporation	Microsoft Office Visio 2003 Es el programa de diagramas técnico y comercial que le ayuda a mostrar ideas, procesos, sistemas y datos numéricos con diagramas y a la reingeniería de procesos. http://office.microsoft.com/es-hn/FX010857983082.aspx
Oracle Corporation	Oracle Designer Es una herramienta completa para modelar, generar y capturar los requerimientos y el diseño de aplicaciones de la empresa. http://www.oracle.com/tools/tools_editions.html
POPKIN Software	SYSTEM ARCHITECT 10 Herramientas CASE para análisis y diseño, incluye técnicas estructuradas y orientadas a objetos. http://www.popkin.com/
Sybase Inc.,	Sybase PowerDesigner ® 11 Es una herramienta de diseño del modelado de datos para empresas que necesitan construir o reestructurar las aplicaciones rápidamente, rentablemente y de forma consistente. http://www.sybase.com/products/developmentintegration/powerdesigner_
Structured Software Systems Ltd (3SL)	Cradle® 5 Es un conjunto de herramientas CASE integradas que dan soporte a la planificación estratégica, Análisis y Diseño. http://www.threesl.com/

Base de Datos

Definición de Base de Datos

Una base de datos es un conjunto de datos organizados para su almacenamiento en la memoria de una computadora. Desde su aparición en la década de 1950, está diseñado para facilitar su mantenimiento y acceso de una forma estándar. Hoy en día, se han hecho imprescindibles para las sociedades industriales y el mundo de negocios.

Según el autor Ezquiel Rozic (2004:20) “Una base de datos es el lugar donde se guardan los datos en reposo y al cual acceden las diferentes aplicaciones (sistemas o programas) de una organización dada”.

Sistema de administración de base de datos (DBMS)

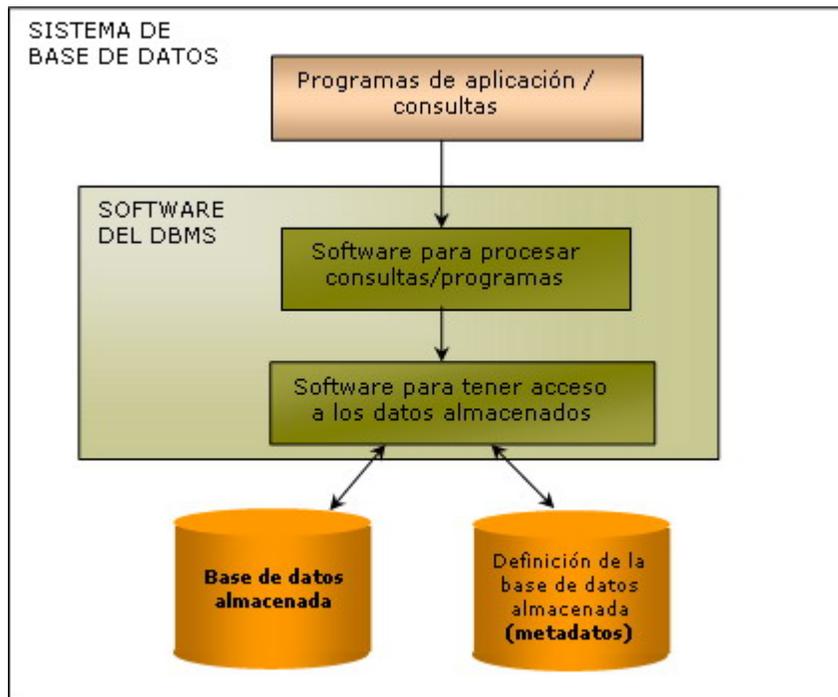


Gráfico 1 Sistema de Base de Datos.
Tomado de “Curso GeneXus Versión 8.0” por ARTech Consultores S.R.L., 2004, Uruguay.

El sistema de administración de base de datos (DBMS, Database Management System) es un conjunto de programas que permite a los usuarios crear y mantener una base de datos permitiendo facilitar el proceso de definición, construcción y manipulación de bases de datos para diversas aplicaciones.

Modelos de datos

Un modelo de datos es un conjunto de conceptos que pueden servir para describir la estructura de una base de datos con cierto nivel de abstracción, ocultando así detalles de almacenamiento que la mayoría de los usuarios no necesitan conocer.

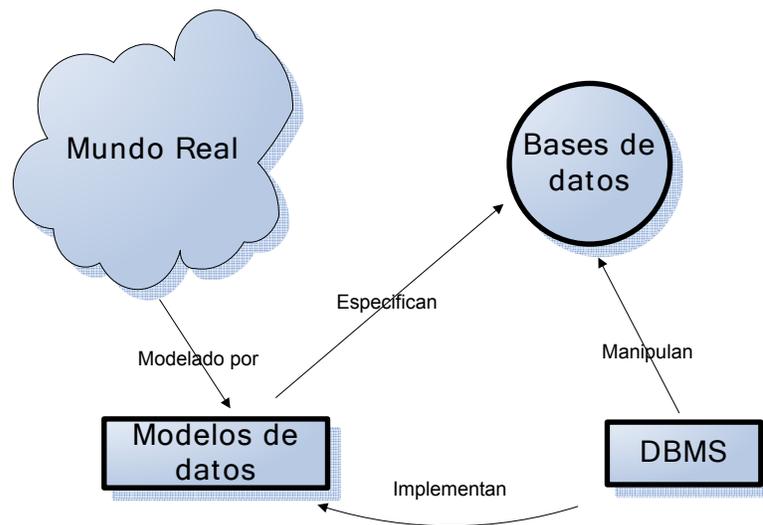


Gráfico 2 Modelos de datos.
Elaborado en Microsoft Office Visio Professional 2003.

La estructura o el diseño de una base de datos se refieren a los tipos de datos, las relaciones y las restricciones que deben cumplirse para esos datos.

Categorías de los modelos de datos

Según ARTech Consultores S.R.L. (2004) *Curso GeneXus Versión 8.0*, “Una clasificación posible de los modelos de datos se realiza de acuerdo al nivel de

abstracción que ofrecen para describir la estructura de la base de datos. Existen en este sentido tres categorías: conceptuales, lógicos y físicos”.

1. **Conceptuales:**

Es una descripción concisa de los requerimientos de información de los usuarios, y contiene descripciones detalladas de los tipos de datos, las relaciones y las restricciones. Ej. El modelo Entidad-Relación.

2. **Lógicos:**

Consiste en implementar una descripción de la estructura de la base de datos que puede procesar el software de DBMS, esto es conocido como esquema lógico. Ej. El modelo relacional.

3. **Físicos:**

Se proporcionan conceptos que describen los detalles de cómo se almacenan los datos en la computadora.

Modelo Entidad-Relación

El modelo Entidad-Relación fue desarrollado por Peter Chen en el año 1976. Este es un modelo conceptual de datos de alto nivel muy utilizado que sirve para la representación de estructuras de información, no conteniendo un lenguaje para representación de manipulaciones de datos. Se centra en buscar las entidades básicas del modelo y luego la relación que existe entre ellas.

Las definiciones básicas del modelo Entidad-Relación se encuentran:

1. **Entidad:** Elementos que se pueden identificar claramente.
2. **Propiedad:** Una propiedad es una característica propia que permite definir una entidad y distinguir entre una entidad en particular de otra.

3. **Relaciones:** Una relación es un vínculo entre dos entidades.
 - a. **Orden:** El número mínimo de presencias de una entidad con respecto a otra.
 - b. **Cardinalidad:** El número máximo de presencias de una entidad para una única presencia de la entidad relacionada.

4. **Tipos de relaciones:** Según los autores Korth, H.F. y Siberschatz A. (1988:25)
 - a. **Uno a uno.** Una entidad en A está asociada únicamente con una entidad en B, y una entidad en B está asociada sólo con una entidad A.
 - b. **Uno a muchas.** Una entidad en A está relacionada con cualquier número de entidad en B, pero una entidad B puede asociarse únicamente con una entidad en A.
 - c. **Muchas a una.** Una entidad en A está vinculada únicamente con una entidad en B, pero una entidad en B está relacionada con cualquier número de entidades en A.
 - d. **Muchas a muchas.** Una entidad en A está asociada con cualquier número de entidades en B, y una entidad en B está vinculada con cualquier número de entidades en A.

5. **El diagrama Entidad-Relación:** La estructura de una base de datos se puede expresarse en forma gráfica a través de un diagrama que se denomina el diagrama Entidad-Relación.

El modelo relacional

Según el autor Ezquiel Rozic (2004:38) “Una base de datos relacional consiste en un conjunto de relaciones, donde cada una de ellas varía o puede variar con el transcurso del tiempo y se identifica de manera única por medio de un nombre”.

La tabla es una *relación*, una cabecera de columna es un *atributo* y una fila se denomina *tupla*. El tipo de datos que describe los tipos de valores que pueden aparecer

en cada columna se llama *dominio*.

En la siguiente tabla se muestra un esquema o estructura de relación de una tabla del modelo relacional.

CLIENTES	CódCliente	Nombre	Dirección
	1	Adriana Martínez	18 de Julio 1333
	2	Marcos López	Av.Rivera 1515

Gráfico 3 Esquema de una tabla del modelo relacional.

Tomado de “Curso GeneXus Versión 8.0” por ARTech Consultores S.R.L., 2004, Uruguay.

Las definiciones básicas sobre base de datos relacional son las siguientes:

1. **Relación:** Una relación es un conjunto de tuplas bien definida con una cantidad de atributos fijos donde cada valor dado para cada atributo de cada tupla pertenece al dominio de valores para dicho atributo.
2. **Propiedad de la relación:** Por la definición y propia estructura de las relaciones, estas cumplen y satisfacen ciertas propiedades:
 - a. Las tuplas no poseen un orden definido.
 - b. No existen tuplas repetidas.
 - c. Los atributos dentro de una tupla no poseen un orden definido.
 - d. Todos los atributos que componen las tuplas son atómicos, o sea, es indivisible.
3. **Índices:** Los índices son estructuras o vías de acceso eficientes, que sirven para agilizar la obtención de registros en respuesta a ciertas condiciones de búsqueda.
4. **Claves:** El concepto de superclave es el de poseer uno o un conjunto de atributo que permitan identificar a cada tupla de forma única sin ninguna ambigüedad.

5. **Clave primaria(PK, Primary Key):** Un atributo es la clave primaria, si el resto de los atributos de la relación que no son clave, dependen funcional y únicamente de dicho atributo y no dependen funcionalmente de ningún otro.
6. **Clave foránea (FK, Foreign Key):** Un conjunto de atributos de una tabla se califica como “clave foránea” cuando establece una referencia a otra tabla, es decir, cuando establece un nexo con otra tabla, la que deberá tener como clave primaria a un conjunto de atributos que corresponderán al mismo concepto.
7. **Clave Candidata:** Por lo general, una tabla puede tener más de una clave. En tal caso, cada una de ellas se denomina clave candidata.
8. **Integridad referencial:** Se refiere a que debe haber reglas de consistencia entre los datos de las distintas tablas de una base de datos relacional.
9. **Dependencia funcional:** Una dependencia funcional, denotada por $X \rightarrow Y$, entre dos conjuntos de atributos X e Y especifica una restricción sobre los posibles registros que podrán aparecer en la base de datos asociada. La restricción dice que si para dos registros coinciden los valores de X, entonces también deben coincidir los valores de Y. También se conoce como dependencia de igualdad.
10. **Dependencia transitiva:** Una dependencia funcional $X \rightarrow Y$ en una tabla es una dependencia transitiva si existe un conjunto de atributos Z que no sea un subconjunto de ninguna clave de la tabla y se cumplen tanto $X \rightarrow Z$ como $Z \rightarrow Y$.
11. **Dependencia multivaluada:** Según los autores Korth, H.F. y Siberschatz A. (1988:194) Si $A \rightarrow B$, entonces no puede tenerse dos tuplas con el mismo valor en A pero diferente valor en B. Una dependencia multivaluada no prohíbe la

existencia de ciertas tuplas, sino que requiere que estén presentes en la relación otras tuplas de una forma determinada. Esta dependencia es conocida también como dependencia “generadora de tupla”. Sea R un esquema de relaciones y sea V pertenece a R y W pertenece a R. La dependencia multivaluada es $V \twoheadrightarrow W$.

Normalización

La normalización es una técnica utilizada para prevenir las anomalías de las tablas en una base de datos. Dicha metodología fue enunciada por Codd que consistía en definir un conjunto de normas a las que llamó formas normales (FN). Una tabla puede estar en la 1era forma normal, segunda forma normal y entre otras.

Según Ezquiel Rozic (2004:48) “La normalización es un término que deriva de la metodología que se utiliza para evitar la redundancia de datos y fácil de acceso y actualización de estos”.

Lo más importante de esta metodología es que para que una relación esté en 2ºFN sí y solo si está en 1ºFN, para estar en 3ºFN debe estar antes en 2ºFN y por transitividad, debe estar en 1ºFN. Por tanto, para que una relación esté en una determinada forma normal antes debe cumplir los requisitos en todas las formas normales que la preceden.

1º Forma Normal. La 1era forma normal debe existir una clave primaria y todos los dominios simples contienen únicamente valores atómicos, es decir que no es divisible.

2º Forma Normal. Para que una relación se encuentre en 2ºFN debe estar en 1ºFN y todos los atributos no clave dependen funcionalmente de la clave primaria.

3º Forma Normal. Para que una relación se encuentre en 3ºFN debe estar en 2ºFN. Si todos los atributos no claves dependen de manera no transitiva de la clave

primaria.

Forma Normal de Boyce-Codd. Muchas veces la 3^oFN no satisface totalmente los criterios de normalización. Para solucionarlo se definió la forma normal de Boyce-Codd que es una relación que se encuentra sí y sólo sí un determinante es una clave candidata.

4^o Forma Normal. La 4^oFN se aplica para dependencias multivaluada. Una relación se encuentra 4^oFN, debe estar la forma normal de Boyce Codd y todas las dependencias multivaluadas de dicha relación son por defecto dependencias funcionales.

5^o Forma Normal. Una relación se encuentra en 5^oFN si se satisface que toda la dependencia de reunión es consecuencia de las claves candidatas de la relación.

Introducción a GeneXus

La herramienta GeneXus fue desarrollado por la empresa ARTech Consultores S.R.L. Es una empresa líder en herramientas de desarrollo de software basado en requerimientos y en soluciones de gestión del conocimiento. Desde 1984, ARTech ha realizado importantes actividades de investigación en el área de las bases de datos relacionales, desarrollo de aplicaciones, herramientas CASE e inteligencia artificial. Hoy en día, cuentan con más de 4.500 clientes en el mundo que utilizan la herramienta GeneXus para crear e integrar aplicaciones que se adaptan fácilmente a los cambios constantes del mundo de negocio.

Según ARTech Consultores S.R.L. (2004) *Curso GeneXus Versión 8.0*, “GeneXus es una herramienta para el desarrollo de aplicaciones sobre bases de datos. Su objetivo es permitir la implantación de aplicaciones en el menor tiempo y con la mejor calidad posible”.

Visión general

Según ARTech Consultores S.R.L. (2004:3), *GeneXus Visión General*, “La idea básica de GeneXus es automatizar todo aquello que es automatizable: normalización de los datos y diseño, generación y mantenimiento de la base de datos y de los programas de aplicación”.

Por consiguiente, se evita que el analista deba dedicarse a tareas rutinarias y tediosas, se concentra su esfuerzo en poner su atención en entender los problemas del usuario.

Principales características

Según ARTech Consultores S.R.L. (2003:2) *GeneXus Filosofía*, “GeneXus es, esencialmente, un sistema que permite una buena administración automática del conocimiento de los sistemas de negocios”.

En la siguiente tabla se muestra las principales características de dicha herramienta:

Cuadro 3 Las principales características de la herramienta GeneXus.

Diseño basado en el Conocimiento	Diseño de aplicaciones a partir de las visiones de los usuarios. Diseño del modelo de datos totalmente automático. Diseño y prototipo independiente de la plataforma.
Desarrollo Automático	Generación automática de la base de datos y aplicaciones. Generación y mantenimiento automático de la documentación de aplicación.
Mantenimiento Automático	Mantenimiento automático de la base de datos y el código. Migración automática de los datos a la nueva estructura.
Desarrollo e implementación multiplataforma	Microsoft .NET JAVA Cliente/Servidor iSeries (AS/400)

GeneXus es una poderosa herramienta multiplataforma para el desarrollo de aplicaciones que permite realizar un desarrollo incremental de aplicaciones de negocios, independientemente de la plataforma de producción. A su vez genera el 100% de la aplicación, y mantiene automáticamente el modelo de datos, la información y las aplicaciones.

Según ARTech Consultores S.R.L. (2005) *Principales características*. “GeneXus es una herramienta CASE que cubre todo el ciclo de desarrollo de una aplicación y genera y mantiene el 100% de su aplicación”.

Asimismo, GeneXus aprovecha la infraestructura tecnológica actual de la empresa, proporcionando máximo resultados en el desarrollo de sistemas, tales como obtener las funcionalidades que necesita a partir de sistemas actuales o heredados, integrar perfectamente nuevas aplicaciones a los sistemas heredados (legacy system), adaptar a nuevas tecnologías con un costo muy bajo, sin la necesidad de rehacer los sistemas e implementar poderosas soluciones de Business Intelligence, Reporting

Service, Workflow y soporte para Web, aprovechando los beneficios de los Web Services y funcionalidades XML.

Tecnologías soportadas

1. Plataformas:

- a. Plataformas de ejecución: Java, Microsoft .NET, Pocket PC.
- b. Sistemas Operativos: IBM OS/400, LINUX, UNIX, Windows NT/2000/2003 Servers y Windows NT/2000/XP/CE.
- c. Internet: Java, ASP.NET, Visual Basic (ASP), C/SQL, HTML y Web Service.
- d. Sistemas manejadores de bases de datos: IBM DB2 UDB, Informix, Microsoft SQL Server, Oracle, PostgreSQL y MySQL.
- e. Lenguajes de programación: Java, C[#], C/SQL, COBOL, RPG, Visual Basic y Visual FoxPro.

2. **Servidores Web:** Microsoft Internet Information Server (IIS), Apache, WebSphere, etc.

3. **Múltiples Arquitecturas:** Arquitecturas de múltiples capas, basadas en Web, Cliente/Servidor y centralizadas (iSeries).

4. **Herramientas de Business Intelligence y Workflow:** Solución de Reporting, Data Warehousing, y Workflow para todos los servidores soportados.

Componentes de GeneXus

La herramienta GeneXus consta de los siguientes componentes:

1. Ambientes de Desarrollo

Es un ambiente integrado de desarrollo para diseñar y prototipar sus aplicaciones de forma independiente de la plataforma de producción.

2. Generadores

GeneXus genera código nativo para las plataformas líderes. Los generadores disponibles son: Microsoft .NET, JAVA, Visual Basic, Pocket PC, C/SQL, Visual FoxPro, COBOL for iSeries y RPG for iSeries.

3. GeneXus Query

Es una herramienta de análisis flexible para usuarios finales que necesitan crear sus propios reportes dinámicos.

- a. Acceso simple y seguro a toda la información de su organización.
- b. Permite a los usuarios finales a construir sus propios reportes.

Ciclos de desarrollo de una aplicación

El ciclo de desarrollo de una aplicación de GeneXus está basado en el enfoque evolutivo. En este proceso de la creación de prototipos funcionales, se pasa con mucha frecuencia del modelo de Diseño al de Prototipo con el que este probando la aplicación, y de éste nuevamente al de Diseño. Se pasará a Producción cuando se termine de probar la aplicación.

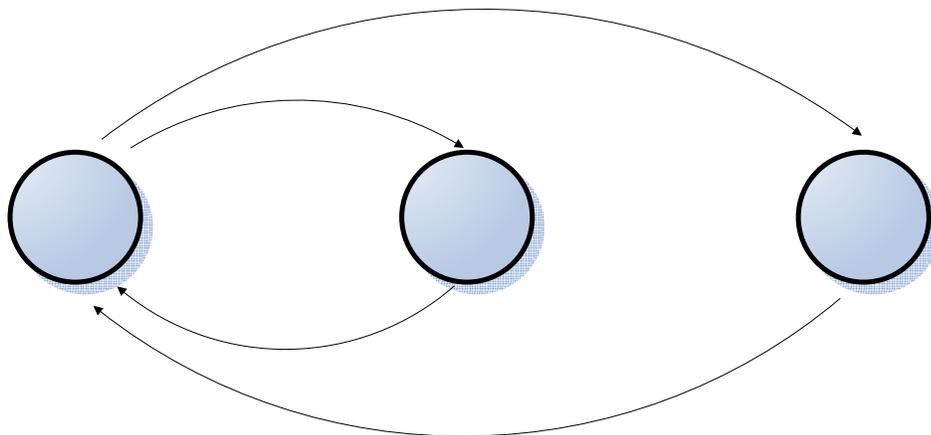


Gráfico 4 Ciclos de desarrollo de una aplicación basado en el enfoque evolutivo.
Elaborado en Microsoft Office Visio Professional 2003.

1. Ciclo de prototipación

El ciclo de Diseño-Prototipo se recorre repetidamente, construyendo y probando sucesivos prototipos funcionales, lo que asegura el total cumplimiento de los requerimientos de sus clientes o usuarios.

La aplicación de producción es funcionalmente equivalente a un prototipo GeneXus. Esto se logra mediante la generación automática de aplicaciones de la herramienta GeneXus.

2. Ciclo de producción

El ciclo de Diseño-Producción se utiliza con menos frecuencia, ya que este ciclo corresponde a la puesta en producción del sistema y esto se realiza solamente cuando el prototipo ha sido aprobado o luego de haber instrumentado y probado algún cambio.

Existe diferencia entre prototipación y producción, esto consiste en que la primera se hace en un ambiente de microcomputador, aunque puede realizarse sobre cualquier plataforma de las soportadas por GeneXus, mientras que la de producción se realiza en el ambiente objeto del usuario.

Base de conocimiento

Cuando se comienza a desarrollar una aplicación con GeneXus la primera tarea es crear la base de conocimiento que representará e implementará al sistema. GeneXus proporciona una manera fácil de diseñar aplicaciones usando el conocimiento exclusivo de su organización. Este conocimiento se almacena en una carpeta Windows.

Esta base de conocimiento es como una especie de repositorio, como se menciona en los componentes de una herramientas CASE, que esta compuesta por objetos comunes de GeneXus tales como Transacciones, Work Panels y Reportes y de otras

informaciones introducidas por el desarrollador de sistemas; y de inferencias que realizará GeneXus a partir de toda esa información.

Dentro de cada base de conocimiento coexisten varios modelos: diseño, prototipo y producción. Cada modelo tiene un tipo que lo caracteriza.

1. Modelo de **Diseño**: el cual corresponde a la representación lógica del sistema. Se crea automáticamente al crear una nueva base de conocimiento. En este entorno no existe lenguaje de programación asociado ni base de datos, por lo tanto, no es posible generar programa alguno.
2. Modelo de **Prototipo**: Se utiliza para prototipar la aplicación en la etapa de desarrollo, probando lo modelado, haciendo modificaciones, volviendo a probar.
3. Modelo de **Producción**: Se utiliza en la etapa de puesta en producción, cuando el prototipo fue aprobado y la aplicación o los cambios efectuados ya pueden ser llevados al cliente.

Cuando una aplicación implementada en GeneXus ha sido puesta en producción, necesariamente deberá haber al menos una de cada uno de los 3 modelos mencionados definidos en la base de conocimiento. Sólo existirá un único modelo de diseño y puede haber varios de prototipo y producción.

Objetos de GeneXus

Una vez creada una base de conocimiento para contener todo el conocimiento de la aplicación que se va a desarrollarse, el siguiente paso consiste en comenzar a describir los objetos de la realidad mediante objetos GeneXus.

Los objetos de GeneXus son creado y definido dentro de un modelo en la base de conocimiento y los objetos creados en cualquier modelo de Prototipo o Producción son automáticamente copiados por GeneXus al modelo de Diseño de la base de conocimiento.

A continuación se describen los objetos GeneXus más importantes según ARTech Consultores S.R.L. (2004) *Curso GeneXus Versión 8.0*:

1. **Transacciones**

Permiten definir objetos de la realidad que el usuario manipula (Ej.: clientes, productos, proveedores, facturas, etc.). Son los primeros objetos que se definen, ya que a través de las transacciones GeneXus infiere el diseño de la base de datos. Además de tener por objetivo la definición de la realidad y la consecuente creación de la base de datos normalizada, cada transacción tiene asociada una pantalla para ambiente Windows y otra para ambiente Web, para permitir al usuario dar altas, bajas y modificaciones en forma interactiva a la base de datos. El analista GeneXus decidirá si trabajar en ambiente Windows, Web, o ambos, y GeneXus generará los programas para ello.

2. **Reportes**

Permiten recuperar información de la base de datos, y desplegarla ya sea en la pantalla, en un archivo o impresa en papel. Son los típicos listados o informes. No permiten actualizar la información de la base de datos.

3. **Procedimientos**

Tienen las mismas características que los reportes, pero a diferencia de éstos, permiten además la actualización de la información de la base de datos.

4. **Work Panels**

Permiten al usuario realizar interactivamente consultas a la base de datos, a

través de una pantalla. Ejemplo: un Work Panel permite al usuario ingresar un rango de caracteres, y muestra a continuación todos los clientes cuyos nombres se encuentran dentro del rango. Son objetos muy flexibles que se prestan para múltiples usos. No permiten la actualización de la base de datos, sólo consultar.

5. **Web Panels**

Son similares a los Work Panels, salvo que son usados a través de navegadores en ambiente Internet/Intranet/Extranet.

6. **Data Views**

Permiten manejar archivos externos como si fueran archivos pertenecientes a la base de conocimiento.

Metodología de desarrollo GeneXus

Es importante señalar que el énfasis de las metodologías tradicionales para el desarrollo de una aplicación es de análisis y diseño de la aplicación sobre la implementación. Aunque el uso de GeneXus también se mantiene este mismo énfasis, pero GeneXus emplea una metodología que tiene un enfoque muy diferente al de las metodologías tradicionales.

Debido a que el desarrollador GeneXus dedicará la mayor parte de su tiempo realizando tareas de análisis, GeneXus se encarga de las tareas de diseño e implementación como por ejemplo, normalización de la base de datos, creación de la base de datos, generación de programas... y entre otros. Por lo tanto, para aprender a utilizar GeneXus adecuadamente va más allá de conocer un nuevo lenguaje: **lo más importante es aprender su metodología.**

De modo que la parte más importante de cualquier herramienta CASE es su metodología de desarrollo. Si los desarrolladores no están siguiendo estrictamente una metodología, las herramientas automatizadas no ayudarán mucho. Pero, una vez que los desarrolladores adoptan una metodología, el uso de la herramienta CASE con dicha metodología ayuda enormemente.

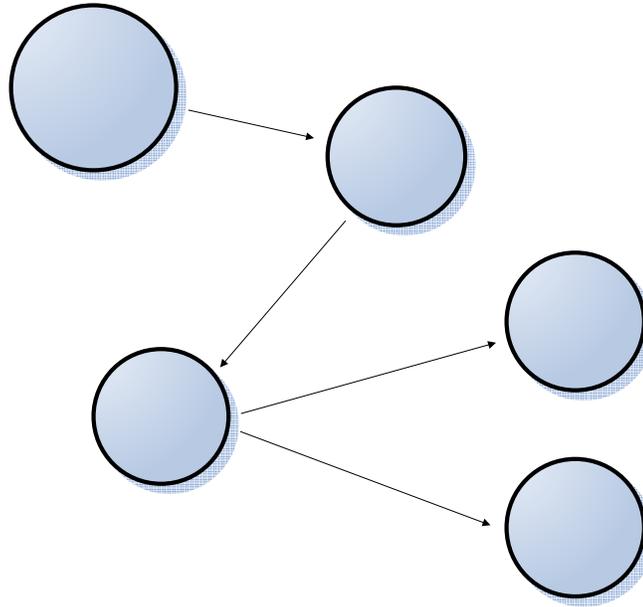


Gráfico 5 Pasos para crear aplicación a través de la metodología GeneXus.
Elaborado en Microsoft Office Visio Professional 2003.

Realidad

El desarrollador del sistema trabaja en alto nivel, en vez de realizar tareas de bajo nivel como diseñar archivos, normalizar, diseñar programas, programar, buscar y eliminar los errores de los programas.

Para comenzar el desarrollo de una aplicación con GeneXus, el primer paso consiste en crear un nuevo proyecto o base de conocimiento. Una vez creada una nueva base de conocimiento, el siguiente paso es describir las visiones de los usuarios.

La metodología GeneXus parte de la premisa de que no es posible construir un modelo de datos estable de la empresa debido a que nadie dentro de la empresa conoce

Base de
Conocimiento

los requerimientos y el alcance de la aplicación a desarrollar como un todo. Para lograr obtener la descripción de la realidad de una forma lo suficientemente objetiva y detallada al mismo tiempo, que permita construir un modelo de datos corporativo, es necesario describir las visiones de los usuarios mediante el uso de la ingeniería inversa.

Para ello, se deben identificar los objetos de la realidad, prestando atención a los sustantivos que los usuarios mencionan en sus descripciones, como por ejemplo: clientes, productos, facturas, y pasar a definirlos mediante objetos GeneXus.

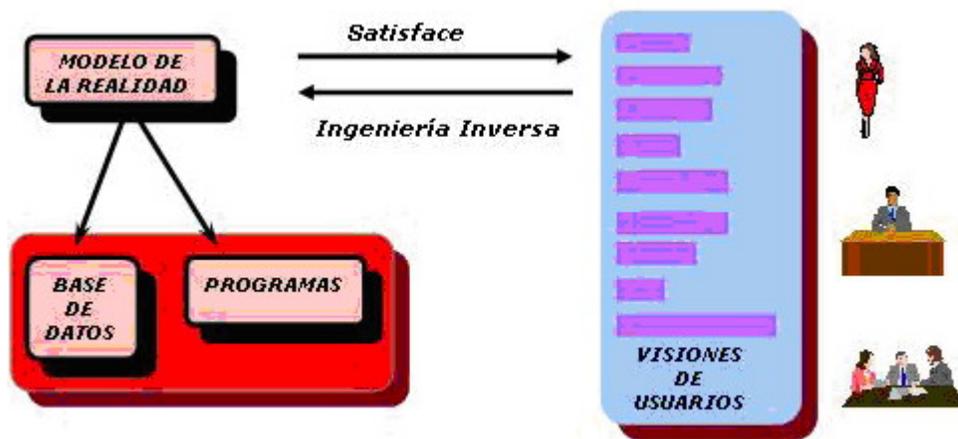


Gráfico 6 Modelado de la realidad a partir de la visión de usuarios mediante la ingeniería inversa. Tomado de “Curso GeneXus Versión 8.0” por ARTech Consultores S.R.L., 2004, Uruguay.

Con la definición de estos objetos GeneXus, GeneXus puede extraer el conocimiento y diseñar la base de datos y los programas de la aplicación en forma totalmente automática.

Si un objeto de la realidad cambia, si se identifican nuevas o diferentes características acerca del mismo, o si se encuentran objetos aún no modelados, el analista GeneXus debe reflejar dichos cambios en los objetos GeneXus que correspondan, y la herramienta se encargará automáticamente de realizar las modificaciones que sean necesarias tanto en la base de datos como en los programas asociados.

La metodología GeneXus está basada en el enfoque evolutivo (prototipo funcional), pues parte de la base de que la construcción de un sistema se realiza mediante aproximaciones sucesivas.

En cada momento el desarrollador del sistema define el conocimiento que tiene; y si luego pasa a tener más conocimiento, lo refleja en la base de conocimiento GeneXus se ocupará de hacer automáticamente todas las adaptaciones en la base de datos y programas.

Requerimiento de preparación

Existen dos requisitos que son necesarios cumplir antes de comenzar a utilizar esta herramienta. Ellos son tener conocimientos de programación y de base de datos relacional.

El primero es el uso de herramienta GeneXus está dirigido a personas con conocimientos de programación que hayan programado alguna vez en algún lenguaje de estas características. Es posible que tenga la dificultad para alcanzar un nivel de comprensión aceptable si no se cuenta con estos conocimientos.

El segundo es tener los conocimientos básicos sobre base de datos relacional para poder desarrollar correctamente aplicaciones con GeneXus. Estas son los puntos fundamentales que debe tener claro:

- a. ¿Qué es un DBMS?
- b. El concepto de tabla, clave primaria, clave candidata, clave foránea.
- c. ¿Qué son las dependencias funcionales?
- d. ¿Qué es la normalización de base de datos?
- e. ¿Para qué se normaliza una base de datos?
- f. ¿Qué son los índices y para qué se utilizan?

Requerimiento computacional

Los elementos que son necesarios para poder usar la herramienta GeneXus dependen del ambiente de trabajo y el lenguaje de programación escogido. Por cada generador de código escogido es posible que tenga instalar algunos elementos relacionado con dicho lenguaje.

A continuación se detallan los requerimientos mínimos del hardware y software necesario para ejecutar GeneXus:

Cuadro 4 Los requerimientos mínimos de la herramienta CASE “GeneXus”.

Requerimiento de Hardware	<ul style="list-style-type: none">■ Microprocesador: Intel Pentium 500 megahertz (MHz) o posterior.■ Memoria RAM: 64megabytes (MB) (se recomiendan 256MB).■ Disco duro: 50 MB mínimo de espacio en disco para la instalación del Ambiente de Desarrollo, más un promedio de 10 MB para cada generador que vaya a instalar. Para crear aplicaciones GeneXus necesitará espacio adicional o una unidad de disco compartida para crear las bases de conocimiento.■ Video: resolución 800x600 o mayor, con 256 colores.
Requerimiento de Software	<ul style="list-style-type: none">■ Sistema operativo de Microsoft Windows 98 o superior. Si usted está utilizando Windows NT, debe instalar el Service Pack 6.0 o superior.■ Paquete redistribuible de Microsoft® .NET Framework versión 1.1.■ Paquete redistribuible de Microsoft Visual J# .NET versión 1.1.■ Microsoft SQL Server Desktop Engine.■ ODBC Driver para SQL Server (es opcional).■ Microsoft Internet Explorer 6.0 Service Pack 1 o superior.

Formación profesional y el aprendizaje

Formación profesional

Según la Biblioteca de Consulta Encarta (2004) “Formación profesional, enseñanza de habilidades directamente relacionadas con la preparación para una profesión o actividad laboral concreta, incorporando en el proceso educativo aspectos teóricos y prácticos”.

A lo largo de los años, la naturaleza de la formación profesional ha variado de forma considerable en razón del cambio social y de las nuevas tecnologías. Por tanto, para enfrentar a estos rápidos cambios los gobiernos y las instituciones educativas realizan estudios de las formas de educación, de formación para atender a las empresas y la identificación de las nuevas estructuras socioeconómicas para que este al día a la demanda y exigencia de campo laboral. La finalidad y el desarrollo de la formación profesional son algunas veces tema de intensos debates.

Aprendizaje

Según la Biblioteca de Consulta Encarta (2004) “Aprendizaje, adquisición de una nueva conducta en un individuo a consecuencia de su interacción con el medio externo”.

Las teorías del aprendizaje ayudan a comprender, predecir, y controlar el comportamiento humano y tratan de explicar como los sujetos acceden al conocimiento. Su objeto se centra en la adquisición de destrezas y habilidades, en el razonamiento y en la adquisición de nuevos conceptos.

Universidad Católica del Táchira (UCAT)

Orígenes de la Universidad Católica del Táchira

Según la Universidad Católica del Táchira (UCAT, 2004) *Orígenes de la Universidad Católica del Táchira*, La Universidad Católica del Táchira es una institución de Educación Superior de la Diócesis de San Cristóbal y encuentra su origen en la creación de la Extensión Táchira de la Universidad Católica Andrés Bello de Caracas, mediante resolución del Consejo Nacional de Universidades, Oficio No. 226, del día 10 de julio de 1962.

Su autonomía fue autorizada por el Decreto Presidencial No. 1567 de fecha 26 de julio de 1982, publicado en la Gaceta Oficial de la República de Venezuela número extraordinario 32.524. Obteniendo con este instrumento la independencia tras 20 años de sujeción fecunda con la capital de la República. Con la autonomía, la UCAT (siglas que identifican a la Universidad Católica del Táchira en el argot universitario venezolano) entró al mundo de la competitividad para consolidar así un proyecto propio de universidad regional.

La toma de conciencia de la nueva identidad que debía asumir la recién fundada universidad fue adquiriendo un ritmo más apropiado de progreso para responder a la realidad de un país que iniciaba un camino de crisis económica ante el agotamiento del modelo de desarrollo petrolero. La nueva conciencia universitaria se resumiría en el incremento radical de la capacidad humana productiva y organizativa de la sociedad venezolana, orientada por un sentido profundo y operante de justicia social que tienda al desarrollo integral de nuestra población.

Licenciatura en Administración mención: Informática Gerencial

Fuente: Escuela de Administración y Contaduría Pública de la UCAT.

Perfil del egresado

Visión

Ser profesionales de alta calidad humana, capaces de manejar las ciencias administrativas para la toma de decisiones empresariales, soportadas con la tecnología de la información como recurso estratégico para la competitividad.

Misión

Formar profesionales con conocimiento para la toma de decisiones soportadas por la tecnología de la información que les permitan optimizar los procesos organizacionales y administrativos existentes al nivel de media gerencia.

Habilidades del estudiante que ingresa

- Capacidad lógica
- Capacidad analítica
- Capacidad de raciocinio
- Participativo
- Objetivo
- Disposición para trabajar en equipo
- Dinámico
- Perseverante
- Creativo e Innovador
- Con valores y principios

Perfil del egresado

Profesionales capaces de planificar, dirigir y controlar el relevamiento, diseño, y ejecución de proyectos de tecnología de información para optimizar los procesos organizacionales que le otorgan competitividad a la organización.

Programa de estudios de la carrera Informática Gerencial

En la siguiente tabla se muestra el contenido del programa de estudio de la carrera Informática Gerencial según la UCAT (2004) *Licenciatura en Administración mención Informática Gerencial*:

Cuadro 5 El programa de estudio de la carrera Informática Gerencial.

<p>Primero Año:</p> <p>Contabilidad General Matemática I Introducción a la Administración Lenguaje y Metodología de Estudio Lógica Matemática Informática I Fundamentos de programación Ambiente y desarrollo</p>	<p>Cuarto Año:</p> <p>Base de Datos Investigación de Operaciones Tributación Pensamiento Social Ética Empresarial Matemática Financiera Sistemas de información I Derecho II</p>
<p>Segundo Año:</p> <p>Sistemas de Costos Industriales Matemática II Economía I Administración Gerencial Estadística I Informática II Derecho I Programación I</p>	<p>Quinto Año:</p> <p>Finanzas Planificación Presupuestaria Auditoria de sistemas Desarrollo Organizacional Decisiones Gerenciales Mercadeo Sistemas de información II Tendencias de la informática Prácticas empresariales Trabajo de Grado</p>
<p>Tercero Año:</p> <p>Evaluación de Estados Financieros Estadística II Economía II Inglés Sistemas operativos Programación II Estructura de datos Informática III</p>	

Marco Normativo

Los objetivos de la educación superior

Según la Ley Orgánica de Universidad de la República Bolivariana de Venezuela (1980), en su Art. 27 establece que, La educación superior tendrá los siguientes objetivos:

1. Continuar el proceso de formación integral del hombre, formar profesionales y especialistas y promover su actualización y mejoramiento conforme a las necesidades del desarrollo nacional y del progreso científico.
2. Fomentar la investigación de nuevos conocimientos e impulsar el progreso de la ciencia, la tecnología, las letras, las artes y demás manifestaciones creadoras del espíritu en beneficio del bienestar del ser humano, de la sociedad y del desarrollo independiente de la nación.
3. Difundir los conocimientos para elevar el nivel cultural y ponerlos al servicio de la sociedad y del desarrollo integral del hombre.

Los valores, principios y filosofía educativa de la UCAT

El Estatuto Orgánico de la Universidad Católica del Táchira, en su Art. 5 establece que, La Universidad Católica del Táchira proclama como suyos los siguientes valores, principios y filosofía educativa:

1. La Universidad Católica del Táchira es fundamentalmente una comunidad de intereses espirituales que reúne a autoridades, profesores y estudiantes en la tarea de buscar la verdad y afianzar los valores trascendentales del hombre.

2. La Universidad es una Institución al servicio de la Nación y le corresponde colaborar con la orientación de la vida del país mediante su contribución doctrinaria en el esclarecimiento de los problemas nacionales.
3. La Universidad debe realizar una función rectora en la educación, la cultura y la ciencia. Para cumplir esta misión sus actividades se dirigirán a crear, asimilar y difundir el saber mediante la investigación y la enseñanza; a completar la formación iniciada en los ciclos educacionales anteriores y, a formar los equipos profesionales y técnicos que necesita la Nación para su desarrollo y progreso.
4. La enseñanza universitaria se inspirará en un definido espíritu de democracia, de justicia social y de solidaridad humana, y estará abierta a todas las corrientes del pensamiento universal, las cuales se expondrán y analizarán de manera rigurosamente científica.

La misión de la Universidad Católica del Táchira

El Estatuto Orgánico de la Universidad Católica del Táchira, en su Art. 6 establece que, la misión de la Universidad Católica del Táchira es:

1. Contribuir a la formación integral de la juventud universitaria, en su aspecto personal y comunitario, dentro de la concepción de los valores cristianos de la vida.
2. Esforzarse por colaborar con el proceso de desarrollo nacional, creando conciencia de su problemática y promoviendo la voluntad de desarrollo. Por lo mismo, concederá especial importancia a la promoción de los recursos humanos y particularmente de la juventud, a fin de lograr la promoción de

todo el hombre y de todos los hombres.

3. Trabajar por la integración de los pueblos de América y por salvaguardar y enriquecer su común patrimonio histórico-cultural y en especial, por la mutua comprensión y acercamiento de los pueblos de nuestra comunidad iberoamericana. Orientado por la implantación de la justicia social, con una superación de los prejuicios y contrastes que dividen y separan a las naciones y en la búsqueda del establecimiento de la paz fundada en un profundo humanismo ecuménico.
4. Irradiar su acción, especialmente a los sectores más marginados de la comunidad nacional.
5. Promover el diálogo de las ciencias entre sí y de éstas con la Filosofía y la Teología, a fin de lograr un saber superior, universal y comprensivo, que llene de sentido el quehacer universitario.

La definición de los programas de computación

Según la Ley sobre Derecho de Autor de la República Bolivariana de Venezuela (1993), en su Art. 17 establece que, los Programas de Computación:

Se entiende por programa de computación a la expresión en cualquier modo, lenguaje, notación o código, de un conjunto de instrucciones cuyo propósito es que un computador lleve a cabo una tarea o una función determinada, cualquiera que sea su forma de expresarse o el soporte material en que se haya realizado la fijación.

Definición de Términos

3GL (Third Generation Language): Los lenguajes de tercera generación son aquellos lenguajes de programación utilizados por los especialistas para construir aplicaciones que incluyen el procedimiento. Es decir, el programador especifica en su programa qué tiene que hacer el ordenador y cómo debe hacerlo. COBOL, C, Pascal o Fortran son lenguajes de tercera generación.

4GL (Fourth Generation Language): Los lenguajes de cuarta generación suponen una evolución de los de tercera generación. En estos lenguajes de programación avanzados, el programador no incorpora el procedimiento a seguir, ya que el propio lenguaje es capaz de indicar al ordenador cómo debe ejecutar el programa.

AS/400 (Application System/400): Familia de computadoras de gama media desarrollada por IBM dirigida a los entornos empresariales.

ASP (Active Server Pages): es una tecnología del lado servidor de Microsoft para páginas Web generadas dinámicamente, que ha sido comercializado como un anexo a Internet Information Server.

Business Intelligence: Se refiere a los procesos, herramientas, metodologías y tecnologías requeridas para convertir datos en información, información en conocimiento y planes que ayuden dirigir efectivamente la actividad comercial del negocio.

Cliente/Servidor: Binomio que describe el principio de las relaciones en una red: un ordenador, a disposición de otros ordenadores, los clientes, unidos a él mediante una red o una línea telefónica. Estos servicios pueden consistir, en el acceso a, y la manipulación de, base de datos centralizados, o en el envío y recepción de correo

electrónico.

Data Warehouse: Un lugar donde estén todos los datos de mi empresa, a los que pueda acceder de forma fácil, eficiente y sin esfuerzo. Una tecnología y una disciplina orientada a la construcción de una colección de datos que permita y facilite el acceso a la información según lo requieran los procesos de toma de decisiones.

Globalización: Concepto que pretende describir la realidad inmediata como una sociedad planetaria, más allá de fronteras, barreras arancelarias diferencias étnicas, credos religiosos, ideologías políticas y condiciones socio-económicas o culturales.

Hardware: Conjunto de componentes materiales de un sistema informático. Cada una de las partes físicas que forman un ordenador, incluidos sus periféricos.

Host: Computador central que permite a uno o varios usuarios comunicarse en red con otro computador.

Ingeniería del software: Es una ingeniería que cubre todos los aspectos relativos a la producción de software.

Lenguaje de programación: Conjunto de normas «lingüísticas» que permiten escribir un programa y que éste sea entendido por el ordenador y pueda ser trasladado a ordenadores similares para su funcionamiento en otros sistemas.

Microsoft .NET .NET es una "plataforma de software". Es un ambiente de lenguaje neutral para escribir programas capaces de operar entre sí de manera fácil y segura. En lugar de ejecutarse sobre una combinación particular de hardware/SO, los programas se ejecutarán en .NET. Los componentes que forman parte de la plataforma .NET se llaman colectivamente .NET Framework.

Multiplataforma: es un término utilizado frecuentemente en informática para indicar la capacidad o características de poder funcionar o mantener una interoperabilidad de forma similar en diferentes sistemas operativos o plataformas.

ODBC (Open Database Connectivity): Es un estándar de acceso a Bases de Datos desarrollado por Microsoft Corporation. El objetivo de ODBC es hacer posible el acceder a cualquier dato de cualquier aplicación, sin importar qué DBMS almacene los datos.

Orientación a Objeto: En la programación tradicional, se distingue entre los datos y los procedimientos. En la técnica de programación orientada a objeto no es así, puesto que no existen los procedimientos como tales. Los elementos de los programas se denominan objetos y son considerados como entidades independientes que se relacionan e interactúan entre sí.

Programa: 1. Redacción de un algoritmo en un lenguaje de programación. 2. Conjunto de instrucciones ordenadas correctamente que permiten realizar una tarea o trabajo específico.

Programación: Programar es automatizar y definir una serie de procesos para resolver un problema y obtener un resultado final.

Reporting Services: Es una plataforma de reportes basada en servidores, la misma que puede ser empleada para crear y administrar reportes tabulares, de matrices, gráficos y de libre formato, la información de estos reportes pueden provenir de diferentes orígenes de datos.

RPG (Report Program Generator): Es un lenguaje de programación para la gestión y producción de informes, formularios, etcétera.

Servidor: Ordenador central de una red y su correspondiente software que pone sus servicios a disposición de los computadores integrantes de la red por medio de un software cliente/servidor.

Sistema: En informática, este término utilizado sin otra palabra que lo adjetive designa un conjunto de hardware y software específico.

Sistema Operativo: Conjunto de programas fundamentales sin los cuales no sería posible hacer funcionar el ordenador con los programas de aplicación que se desee utilizar.

Software: El término inglés original define el concepto por oposición a hardware: blando-duro, en referencia a la intangibilidad de los programas y corporeidad de la máquina.

SQL (Structured Query Language): Es un estándar en el lenguaje de acceso a bases de datos.

Web Services: Un Web Service es una aplicación que puede ser descrita, publicada, localizada e invocada a través de una red, generalmente Internet. Combinan los mejores aspectos del desarrollo basado en componentes y la Web. La interfaz de los Web Services esta definida en términos de los mensajes que el mismo acepta y retorna, por lo cual los consumidores de los Web Services pueden ser implementados en cualquier plataforma y en cualquier lenguaje de programación, solo tiene que poder crear y consumir los mensajes definidos por la interfaz de los Web Services.

Workflow (Flujo de trabajo): Es el estudio de los aspectos operacionales de una actividad de trabajo: cómo se estructuran las tareas, cómo se realizan, cuál es su orden correlativo, cómo se sincronizan, cómo fluye la información que soporta las tareas y cómo se le hace seguimiento al cumplimiento de las tareas.

XML (eXtensible Markup Language): Lenguaje de marcado ampliable o extensible desarrollado por el World Wide Web Consortium (W3C). No es un lenguaje de programación propiamente dicho, sino un lenguaje de marcas para describir información, cuyo objetivo principal es conseguir una página Web más semántica.

Operacionalización de las variables

Cuestionario para los profesores en el área de Informática

Cuadro 6 La operacionalización de las variables para los profesores en el área de Informática de la Escuela de Administración y Contaduría Pública de la UCAT.

Objetivo específico	Variables	Definición Operacional	Indicadores	Dimensiones	Ítems
Determinar la factibilidad de la incorporación de la herramienta CASE “GeneXus” en la formación de los alumnos de Administración mención Informática Gerencial.	Factibilidad Operacional.	La evaluación de la factibilidad operacional sobre la disposición de los profesores en el área de Informática para utilizar herramienta CASE “GeneXus” como software de aprendizaje.	<ol style="list-style-type: none"> 1. Conocimiento acerca de las herramientas CASE y de las metodologías de desarrollo de sistema de información. 2. Aceptación de la utilización de la herramienta CASE “GeneXus” por parte de los profesores como software de aprendizaje. 3. Incorporación el uso de las herramientas CASE en el diseño curricular de programa del estudio para la formación de los alumnos de Administración mención Informática Gerencial. 4. Disposición de actualización pedagógica o de capacitación sobre el manejo de herramienta CASE “GeneXus” por parte de los docentes en el área de informática de la Escuela de Administración y Contaduría Pública de la UCAT. 	Los profesores de la Escuela de Administración y Contaduría Pública de la UCAT en el área de Informática.	<p>1, 2 y 3.</p> <p>4.</p> <p>5 y 6.</p> <p>7.</p>

Cuestionario para los alumnos de tercero y cuarto año de Informática Gerencial

Cuadro 7 La operacionalización de las variables para los alumnos de tercero y cuarto año de Informática Gerencial.

Objetivo específico	Variables	Definición Operacional	Indicadores	Dimensiones	Ítems
Determinar la factibilidad de la incorporación de la herramienta CASE “GeneXus” en la formación de los alumnos de Administración mención Informática Gerencial.	Factibilidad Operacional.	La evaluación de la factibilidad operacional sobre la disposición de los alumnos de Administración mención Informática Gerencial para utilizar herramienta CASE “GeneXus” como software de aprendizaje.	<ol style="list-style-type: none"> 1. Conocimiento acerca de las herramientas CASE. 2. Disposición por parte de los alumnos para aprender la herramienta CASE “GeneXus”. 3. Incorporación el uso de las herramientas CASE en el diseño curricular de programa del estudio para la formación de los alumnos de Administración mención Informática Gerencial. 	Los alumnos de tercero y cuarto año de la carrera Informática Gerencial.	<ol style="list-style-type: none"> 1 y 2. 3. 4.

CAPÍTULO III

MARCO METODOLÓGICO

Nivel de investigación

El presente trabajo de investigación es un proyecto factible. Según la Universidad Pedagógica Experimental Libertador (UPEL, 1998:7) “El proyecto factible consiste en la investigación, elaboración y desarrollo de una propuesta de un modelo operativo viable para solucionar problemas, requerimientos o necesidades de organizaciones o grupos sociales; puede referirse a la formulación de políticas, programas, tecnologías, métodos o procesos”.

También se considera como exploratorio porque no tiene antecedente y según los autores Hernández, Fernández y Baptista (2003:115) “Los estudios exploratorios se efectúa, normalmente, cuando el objetivo es examinar un tema o problema de investigación poco estudiado, del cual se tienen muchas dudas o no se ha abordado antes”.

Diseño de Investigación

Desde punto de vista del tipo de investigación, el presente trabajo se considera como una investigación no experimental según los autores Hernández, Fernández y Baptista (2003:267) “Podría definirse como la investigación que se realiza sin manipular deliberadamente variables”. De acuerdo con los mismos autores (2003:267) “Lo que hace en la investigación no experimental es observar fenómenos tal y como se dan en su contexto natural, para después analizarlos”.

También se considera que el diseño de investigación es de campo porque la recolección de datos se obtiene por contacto con la realidad donde ocurren los hechos. Según el autor Sabino C. A. (2000:97) “el diseño de campo, se basan en informaciones o datos primarios, obtenido directamente de la realidad”. Y de acuerdo con la UPEL (1998:5):

Se entiende por Investigación de Campo, el análisis sistemático de problema en la realidad, con el propósito bien sea de describirlos, interpretarlos, entender su naturaleza y factores constituyentes, explicar sus causas y efectos o predecir su ocurrencia, haciendo uso de método característico de cualquiera de los paradigmas o enfoques de investigación conocidos o en desarrollo.

Desde punto de vista de la trascendencia en el tiempo se considera como una investigación transeccional porque se ubica en un período de tiempo corto. Y de acuerdo con los autores Hernández, Fernández y Baptista (2003:270) “Los diseños de investigación transeccional o transversal recolectan datos en un solo momento, en un tiempo único. Su propósito es describir variables y analizar su incidencia e interrelación en un momento dado”.

Fase I: Diagnóstico

En esta etapa se dedicó a determinar los requerimientos de la información, la cual se definió con la población y la muestra del estudio, las técnicas e instrumento de recolección de datos y la herramienta utilizada para el análisis e interpretación de resultados.

La población del estudio

Según Webster (2000:8) “La población es la recolección completa de todas las observaciones de interés para el investigador”. En el presente trabajo la población está conformada por:

- a. El Presidente de la empresa Information System Solution (INSYSO) C.A. El distribuidor de venta de la herramienta GeneXus en Venezuela.
- b. El Director del Centro de Computación de la UCAT.
- c. Los trece (13) profesores de la Escuela de Administración y Contaduría Pública de la UCAT en el área de Informática que dan clases a los alumnos de Administración mención Informática Gerencial.
- d. Los ciento veintitrés (123) alumnos de tercero y cuarto año de la carrera Informática Gerencial de la UCAT.

La muestra del estudio

1. ***Muestra no probabilística.*** Según los autores Hernández, Fernández y Baptista (2003:305) “La elección de los elementos no depende de la probabilidad, sino de causas relacionadas con las características de la investigación o de quien hace la muestra”.

Muestreo intencional u opinático: la selección de los elementos con base en criterios o juicios del investigador. En el presente trabajo se tiene las siguientes muestras no probabilística:

- a. El Presidente de la empresa INSYSO C.A.
- b. El Director del Centro de Computación de la UCAT.
- c. Los trece (13) profesores de la Escuela de Administración y Contaduría Pública de la UCAT en el área de Informática que dan clases a los alumnos de Administración mención Informática Gerencial. Debido a que la población es pequeña. Por consiguiente, se toma en su totalidad.

2. **Muestra probabilística.** La muestra probabilística según Webster (2000:9) “La muestra es una parte representativa de la población que se selecciona para ser estudiada ya que la población es demasiado grande como para analizarla en su totalidad”. Y según los autores Hernández, Fernández y Baptista (2003:305) “En la muestra probabilística todos los elementos de la población tienen la misma posibilidad de ser escogidos”.

En el presente trabajo la muestra probabilística se calculó en base a la población de alumnos de tercero y cuarto año de la carrera de Administración mención Informática Gerencial. Se excluyeron a los alumnos de primero y segundo año porque se consideró que ellos aún no tienen criterios formales en el estudio de dicha carrera, y a los de quinto porque se van a graduar pronto, por lo tanto ya no tienen la posibilidad de aprender a manejar las herramientas CASE.

Cuadro 8 La población de los alumnos de tercero y cuarto año de Informática Gerencial.

Los alumnos de Informática Gerencial	Número de alumnos
Tercero año	70
Cuarto año	53
Total de los alumnos	123

La fórmula para calcular el tamaño de la muestra es la siguiente:

$$n = \frac{\frac{Z^2 \times P \times Q}{d^2}}{1 + \frac{1}{N} \left(\frac{Z^2 \times P \times Q}{d^2} - 1 \right)}$$

Donde

Z : es el nivel de confianza. 90% = 1.65

d : Error esperado (E = 0.10)

n : es el tamaño de la muestra.

P : Proporción de éxito = 0.5

N : es el tamaño de la población. 123

Q : Proporción de fracaso = 0.5

$$n = \frac{\frac{1,65^2 \times 0,5 \times 0,5}{0,10^2}}{1 + \frac{1}{123} \left(\frac{1,65^2 \times 0,5 \times 0,5}{0,10^2} - 1 \right)}$$

$$n = 44.04702401 \approx 44$$

Según la muestra calculada anteriormente se escogió cuarenta cuatro (44) alumnos de tercero y cuarto año de la carrera Informática Gerencial que iban a contestar el cuestionario mediante el muestreo aleatorio simple.

Se utilizó la función ALEATORIO.ENTRE(1,123) de Microsoft Excel XP para generar los números aleatorios entre uno (1) y ciento veintitrés (123). De esta manera las personas seleccionadas desde una lista de alumnos para ser encuestadas se distribuyeron en la siguiente tabla según el año que cursan:

Cuadro 9 El tamaño de la muestra de los alumnos de tercero y cuarto año de Informática Gerencial.

Los alumnos de Informática Gerencial	Número de alumnos a encuestar
Tercero año	23
Cuarto año	21
Total de los alumnos	44

Técnicas e instrumento de recolección de datos

La técnica de recolección de datos para el presidente de la empresa INSYSO C.A. y el Director del Centro de Computación de la UCAT consistió en la encuesta, aplicando como instrumento la entrevista.

Igualmente, la técnica de recolección de datos para los profesores en el área de informática de la UCAT y los alumnos de tercero y cuarto año de Informática Gerencial se basó en la encuesta, utilizando como instrumento el cuestionario.

Antes de aplicar los cuestionarios a los profesores y a los alumnos se seleccionó a dos especialistas en el área de sistema de información para la validación de los cuestionarios, anexando la lista de objetivos y los cuadros de la operacionalización de las variables.

Según la Biblioteca de Consulta Encarta (2004) “Encuesta, instrumento cuantitativo de investigación social mediante la consulta a un grupo de personas elegidas de forma estadística, realizada con ayuda de un cuestionario”. Y de acuerdo con los autores Hernández, Fernández y Baptista (1992:276) “Un cuestionario consiste en un conjunto de preguntas respecto a una o más variables a medir”.

Análisis e interpretación de resultados

Una vez recolectada la información, se realizó el procesamiento de dicha información utilizando la herramienta SPSS 13.0 para realizar un análisis estadístico e interpretación de los resultados obtenidos.

Fase II: Estudio de factibilidad

En esta fase se realizó un estudio de factibilidad para determinar la viabilidad del desarrollo del proyecto planteado.

El estudio de factibilidad consiste en realizar un estudio preliminar para obtener las informaciones requeridas, determinar los costos, los beneficios, los requerimientos de recursos; y evaluar la factibilidad del proyecto propuesto. Según el criterio del autor O'Brien (2001:93) El estudio de factibilidad se puede evaluar en términos de cuatro enfoques importantes que son la factibilidad organizacional, económica, técnica y operacional.

Cuadro 10 Estudio de factibilidad para la incorporación de la herramienta CASE “GeneXus” como software de aprendizaje.

Factibilidad organizacional	Factibilidad económica
Determinar qué tan bien respalda la herramienta propuesta como software de aprendizaje con los objetivos de educación superior, los valores, principios y filosofía educativa de la UCAT; y el perfil del egresado de la carrera Informática Gerencial de la UCAT.	<ol style="list-style-type: none"> 1. Determinar los costos asociados en la adquisición de la licencia de software. 2. Determinar la disposición económica de la UCAT en la adquisición de la licencia de la herramienta propuesta. 3. Análisis de costos/beneficios.
Factibilidad técnica	Factibilidad operacional
Determinar la capacidad y disponibilidad de los recursos de software, el hardware y redes que cuenta la UCAT, comparando con los requisitos de la herramienta propuesta.	Determinar la aceptación del uso de la herramienta CASE como software de aprendizaje por parte de los profesores y de los alumnos de tercero y cuarto año de Informática Gerencial.

La factibilidad organizacional

El enfoque de la factibilidad organizacional se centra en qué tan bien respalda la herramienta propuesta los objetivos de la educación superior, los valores, principios y filosofía educativa de la UCAT; y el perfil del egresado de la carrera Informática Gerencial de la UCAT.

Se compararon los objetivos de la educación superior, los valores, principios filosóficos de la Universidad Católica del Táchira y el perfil del egresado de la carrera Informática Gerencial con la propuesta de la incorporación de la herramienta CASE en la formación de los alumnos de Informática Gerencial para saber si dicha propuesta

está bien respaldada por los que están establecidos en la Ley Orgánica de Educación, el Estatuto Orgánico de la UCAT y el perfil del egresado de la carrera Informática Gerencial.

La factibilidad económica

La factibilidad económica tiene que ver con el hecho de si los beneficios esperados excederán a los costos de adquirir y capacitar la utilización de la herramienta propuesta.

Se realizaron entrevistas al Presidente de la empresa INSYSO C.A. para averiguar los costos asociados en la adquisición de la licencia y los cursos de capacitación de la herramienta propuesta; y al Director del Centro de Computación para investigar cuáles son los procedimientos que deben seguir para que la UCAT realice la adquisición de la licencia de la herramienta propuesta. Después de realizar las mismas se llevó a cabo un análisis de costos/beneficios.

La factibilidad técnica

La factibilidad técnica puede demostrarse si la empresa cuenta con la capacidad y disponibilidad de los recursos de software, el hardware y redes que satisfacen las necesidades de la herramienta propuesta.

En el presente estudio se realizó una entrevista con el Director del Centro de Computación de la UCAT para determinar si las computadoras de la UCAT cumplen con los requerimientos mínimos de hardware de la herramienta propuesta; y para saber si la UCAT cuenta con el personal especializado en la instalación y mantenimiento de software de computación.

La factibilidad operacional

La factibilidad operacional es la disposición de los profesores en el área de Informática y los alumnos de tercero y cuarto año para utilizar y respaldar la herramienta propuesta.

Se aplicaron cuestionarios a los profesores en el área de Informática de la Escuela de Administración y Contaduría Pública; y los alumnos de tercero y cuarto año de la carrera Informática Gerencial para determinar la aceptación de ellos para el uso de la herramienta CASE como software de aprendizaje.

Fase III: Desarrollo de la propuesta

Antes del desarrollo de la propuesta, se realizó una demostración del uso de la herramienta GeneXus en el desarrollo de un software de aplicación sencillo basada en una situación simulada. Se omitió los pasos previos al análisis de sistema porque se supone, ya fueron realizados. La metodología de desarrollo de sistema de información se basó en el enfoque evolutivo (prototipo funcional).

En cuanto al desarrollo de la propuesta, se mencionó las asignaturas que podrían utilizar esta herramienta dentro del contenido programático de la carrera Informática Gerencial y se definió los lineamientos a seguir para la incorporación de la misma dentro del proceso de enseñanza/aprendizaje de los alumnos de dicha carrera.

CAPÍTULO IV

ANÁLISIS E INTERPRETACIÓN DE LOS RESULTADOS

Factibilidad organizacional

Conforme al ordinal 1º del Art. 7 de la Ley Orgánica de Educación establece que “Continuar el proceso de formación integral del hombre, formar profesionales y especialistas y promover su actualización y mejoramiento conforme a las necesidades del desarrollo nacional y del progreso científico”.

La herramienta propuesta está respalda por este artículo ya que se propone el mejoramiento a la formación de los alumnos de Administración mención Informática Gerencial.

De acuerdo al ordinal 3º del Art. 5 del Estatuto Orgánico de la Universidad Católica del Táchira, establece que:

La Universidad debe realizar una función rectora en la educación, la cultura y la ciencia. Para cumplir esta misión sus actividades se dirigirán a crear, asimilar y difundir el saber mediante la investigación y la enseñanza; a completar la formación iniciada en los ciclos educacionales anteriores y, a formar los equipos profesionales y técnicos que necesita la Nación para su desarrollo y progreso.

El uso de la herramienta CASE como software de aprendizaje se puede contribuir a la UCAT formar profesionales preparados para el desarrollo y el progreso de la nación, ya que su objetivo es proponer la utilización de nueva tecnología en el

diseño y desarrollo de sistemas de información. Por tanto está respaldado por los valores, principios y filosofía educativa de la UCAT.

Según el Perfil del egresado de la Licenciatura en Administración mención Informática Gerencial “Profesionales capaces de planificar, dirigir y controlar el relevamiento, diseño, y ejecución de proyectos de tecnología de información para optimizar los procesos organizacionales que le otorgan competitividad a la organización”.

La utilización de herramienta propuesta también está respaldada por el perfil de egresados debido a que el uso de la misma está relacionado a la ejecución de proyectos de tecnología de información.

Por consiguiente, se puede afirmar que el uso de la herramienta propuesta como software de aprendizaje está respaldado por los objetivos de educación superior, los valores, principios y filosofía educativa de la UCAT; y el perfil del egresado de la carrera Informática Gerencial de la UCAT.

Factibilidad económica

Entrevista al Presidente de la empresa Information System Solution C.A.

1. ¿Cuáles son los costos asociados en la adquisición de la herramienta GeneXus?

Respuesta: El precio de la herramienta GeneXus es barato en comparación con otras herramientas CASE. Los costos asociados en la adquisición de la herramienta GeneXus son los costos de la licencia y del mantenimiento.

- a. Costo de licencia: Básicamente GeneXus tiene para desarrollar 2 líneas de producto: GeneXus Developer y GeneXus Generador.

- **GeneXus Developer:** Este producto es en el que un analista GeneXus describe una realidad de negocios y este se encarga de armar de forma lógica toda la aplicación. Los costos de las licencias básicas GeneXus Developer son:

1ª licencia	\$ 4800
2ª a 4ª licencias	\$ 4200 cada una
5ª a 9ª licencias	\$ 3600 cada una
10ª en adelante	\$ 3000 cada una

- **GeneXus Generator:** En este el programador que tiene que escribir todos los programas necesarios para formar la aplicación en el lenguaje específico que los clientes determinen. Por lo tanto, antes de comprar su empresa debe decidir el lenguaje a operar como puede ser JAVA, Microsoft .NET o otros.

Los precios de las licencias de Generadores de Visual Basic, JAVA y Microsoft .NET son:

1ª licencia	\$ 2000
2ª a 9ª licencias	\$ 1000 cada una
10ª en adelante	\$ 500 cada una

Los precios de las licencias de Generador PDA (Pocket PC) son:

1ª licencia	\$ 750
2ª en adelante	\$ 500 cada una

Los precios de las licencias de Generadores para los equipos IBM (AS/400, i-Seres o i-5) Para pedir precios de estos equipos lo hace según otra lista ya que según sea la máquina a utilizar es el valor del producto. Por lo tanto, lo hace caso a caso.

En resumen: los valores de las licencias serán menores en la medida en que se adquieran más licencias.

- b. Costo de mantenimiento: En todos los casos los productos ofrecidos cuentan con una garantía de un año a partir de su instalación. Esto involucra errores de producto, soporte a usuarios por Internet y nuevas versiones de los productos que se liberen dentro del período de garantía.

Una vez finalizado este período se realizará un contrato de Mantenimiento y Soporte por un año (y así sucesivamente año a año) con un valor del 15% de todos los productos adquiridos en su precio de lista.

- 2. ¿Ofrece usted algunos cursos de capacitación del manejo de la herramienta GeneXus?

Respuesta: Sí, la casa matriz de la herramienta GeneXus ARTech Consultores S.R.L. ofrece curso de capacitación a distancia y también se puede realizar dicho curso en Venezuela, pero se requiere coordinar entre las partes. Los cursos que deben darse son:

- a. Curso de Desarrollador GeneXus (3 semanas en promedio ajustable a las necesidades del cliente).
- b. Curso de Desarrollador GeneXus para Internet (1 semana).

Para capacitar en GeneXus 8.0 al personal de su empresa que estará involucrado en el desarrollo de aplicaciones GeneXus, la capacitación se llevara a cabo en instalaciones provistas por ustedes y estará dirigida únicamente para miembro de su empresa.

a. Capacitación en GeneXus 8.0

Objetivo: Capacitar a los desarrolladores para conseguir el cambio de mentalidad que se requiere y dar elementos básicos para el diseño, desarrollo y

prototipación de aplicación GeneXus.

Orientado: A las personas involucradas en el desarrollo de aplicaciones utilizando GeneXus.

Condiciones previas: Conocimientos de Base de Datos (preferentemente haber aprobado dicha materia en las carreras de Ingeniero o Analista en Computación) y Conocimiento básico de programación.

Autoestudio: El alumno deberá leer el manual “Diseño de Aplicaciones con GeneXus” en forma previa al comienzo de la capacitación. La lectura del mismo prepara a los participantes para un mejor aprovechamiento de la capacitación. Descargue el archivo desde la siguiente dirección:

<http://www.gxtechnical.com/main/hdcenter.aspx?2,5,36,659>

Fechas: A coordinar entre las partes.

Carga horaria: 120 horas distribuidas en 3 semanas (a confirmar)

Horarios: A coordinar.

Instructor: Instructor GeneXus certificado.

Facturación: El precio para brindar esta capacitación en las condiciones antes mencionadas, asciende a la suma de US\$ 5,000 (cinco mil dólares americanos, precio ajustable de acuerdo a las condiciones finales acordadas). Todos los gastos serán asumidos por el cliente, estos son: pasaje aéreo, gasto de hotel y estadía del instructor.

b. Capacitación en Desarrollo de Aplicaciones para Internet con GeneXus.

Objetivo:

- Dar los elementos necesarios para poder generar aplicaciones a ser utilizadas en Internet utilizando los objetos Web GeneXus.
- Integrar las operaciones de Internet con la Base de Datos Corporativa.

Orientado: Gerente de Proyecto y Técnico de la empresa.

Condiciones previas: Tener aprobado el Curso GeneXus.

Fechas: A coordinar entre las partes.

Carga horaria: 40 horas en una semana (full time).

Horario: A coordinar.

Instructor: Instructor GeneXus certificado.

Facturación: El precio de ir a dar este curso es de US\$ 3,500 (tres mil quinientos dólares americanos) más gastos de pasaje y estadía del docente... etc.).

La factura será emitida por la empresa Langecor y los pagos se realizarán a través del Banco American Express.

American Express Bank Internacional

ABA # 0660-1220-7

Para el crédito en la cuenta N° 267083424 a nombre de Langecor S.A.

Referencia: CIF 102635

El original de la factura será entregada por el instructor al comienzo de las clases, siendo las condiciones de pago *Contado* una vez finalizado el dictado de clases.

Entrevista al Director del Centro de Computación de la UCAT.

¿Cuáles son los procedimientos que deben seguir para que la UCAT realice la adquisición de un software de aplicación?

En el caso de tratarse de software dirigido a la enseñanza de los alumnos, las facultades hacen la solicitud de este software al Vice-Rectorado Administrativo, ente que se encarga de coordinar y supervisar de acuerdo con el Rector y el Consejo de

Administración las actividades económicas, financieras y administrativas de la Universidad, para la adquisición o compra de la licencia de software.

Por lo que explicó el Director, se ha encontrado dificultad de determinar si actualmente la UCAT está o no preparada económicamente para financiar la adquisición de la herramienta CASE “GeneXus” como software de aprendizaje.

A pesar de que no ha podido determinar la disposición económica actual de la UCAT en la adquisición de licencia de un software de aplicación, según el Director la UCAT ha hecho inversiones en la adquisición de licencias de software tanto para el uso interno como para la formación de los alumnos que estudia en dicha universidad.

Dentro de éstas se encuentran Borland C++Builder 5.0, Borland Delphi 4, Borland Java Builder, Corel Draw, Macromedia Dreamweaver MX 2004, Macromedia Flash MX, Microsoft Office 2003, Microsoft Project 98, Microsoft Visual Studio .NET, Microsoft Visual Studio 6.0, MIXNET, SPSS 10.0, Symantec Norton Anti-Virus y entre otras aplicaciones más.

Por lo tanto, se puede afirmar que la UCAT estaría dispuesta a realizar la compra del software de aplicaciones siempre y cuando exista una solicitud por parte de la Facultad o Escuela ante la necesidad de los profesores en utilizar la herramienta propuesta para la enseñanza de sus alumnos.

Análisis de costos/beneficios

Según el autor O’Brien (2001:94) “Por lo general, los estudios de factibilidad abarcan un análisis de costos/beneficios”.

Los beneficios que puede traer con esta propuesta es mejorar la calidad de educación a la formación de los alumnos de Administración mención Informática

Gerencial. Dichos beneficios son intangibles porque no se pueden cuantificar en término monetario y es difícil de estimar.

En cuanto al costo, no habrá la recuperación o retorno de inversión ya que la Universidad Católica del Táchira es una institución sin fines de lucro.

Factibilidad técnica

Entrevista al Director del Centro de Computación de la UCAT.

1. ¿Conoce usted las herramientas de Ingeniería de Software Asistida por Computadora (CASE)?

Respuesta: Sí.

2. ¿De las siguientes herramientas CASE cuáles conoce o ha utilizado alguna vez? GeneXus, Oracle Designer, Sybase PowerDesigner, System Architect, Silverrun, Win A&D, Microsoft Office Visio, ninguno u otro.

Respuesta: Conozco GeneXus, Oracle Designer y Microsoft Office Visio.

3. ¿Existe personal capacitado para realizar tarea de instalación y mantenimiento de software de computación?

Respuesta: Sí, la Universidad Católica del Táchira cuenta con personal especializado en la instalación y mantenimiento de software de computación. El Centro de Informática y Sistema (CIS) es la unidad encargada de realizar la instalación de software correspondiente al área académica de la UCAT.

4. ¿Cuáles son las características de las computadoras existentes en los laboratorios de computación?

Respuesta: Los requerimientos de las computadoras en los laboratorios de computación de la UCAT son las siguientes:

Cuadro 11 La descripción de los requerimientos de las computadoras en los laboratorios de computación de la UCAT.

Requerimiento de Hardware	Marca	IBM.
	Modelo	NetVista 6648-TAS Computador Personal
	Microprocesador	Intel Pentium III 1 GHz con Caché de 256 KB.
	Tarjeta madre	Tarjeta madre Intel 815e
	Memoria RAM	128 Mega Bytes.
	Tarjeta de Red	Intel PRO/100 VE Ethernet.
	Disco Duro	IDE 20 Giga Bytes.
	Unidad de CD-ROM	24X.
	Unidad Floppy	1.44MB.
	Monitor CRT	Monitor de 15 pulgadas de IBM.
Requerimiento de Software	Sistema Operativo	Microsoft Windows 2000 Professional.
	Navegador Internet	Microsoft Internet Explorer 6.0 Service Pack 1.

Factibilidad operacional

Cuestionarios dirigidos a los profesores de la Escuela de Administración y Contaduría Pública en el área de Informática

Inicialmente los cuestionarios estaban dirigidos a los trece (13) profesores en el área de Informática de la Escuela de Administración y Contaduría Pública. Fue imposible hacerles la entrega de los cuestionarios a tres (3) profesores. Por lo tanto, solo se recolectaron las informaciones y las opiniones de diez (10) profesores.

1. ¿Conoce usted las herramientas de Ingeniería de Software Asistida por Computadora (CASE)? Sí _____ No _____

Cuadro 12 Conocimiento de los profesores acerca de las herramientas CASE.

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos Sí	9	90,0	90,0	90,0
No	1	10,0	10,0	100,0
Total	10	100,0	100,0	

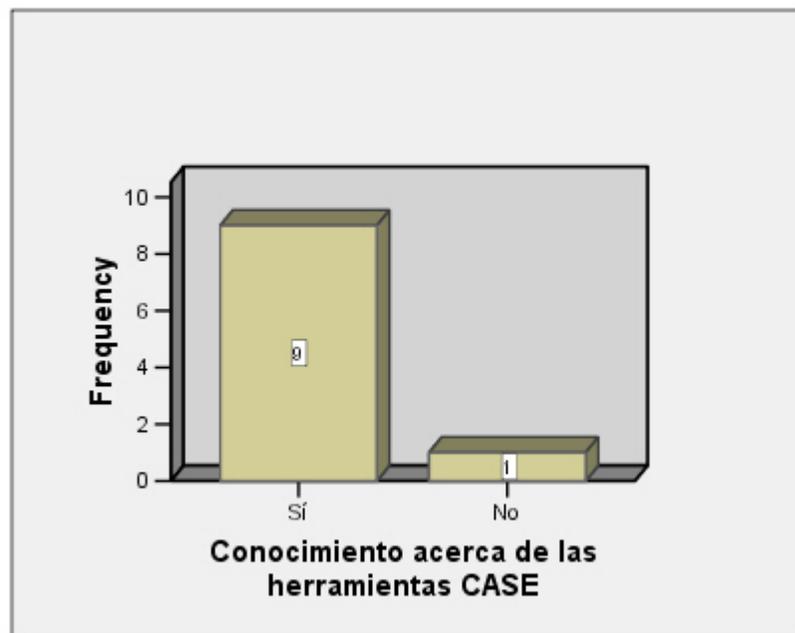


Gráfico 7 Resultado obtenido sobre el conocimiento de los profesores acerca de las herramientas CASE. Elaborado en SPSS 13.

A través del gráfico 7 se puede observar que nueve (9) de los diez (10) profesores respondieron que sí conocen las herramientas de Ingeniería de Software Asistida por Computadora, los cuales representan el 90% del total de los profesores y sólo hay un (1) profesor que no las conoce.

3. ¿Cuáles de las metodologías de desarrollo de sistema de información conoce o domina usted?

_____ Modelo tradicional o en cascada. (Planificación, análisis, diseño, implantación y soporte de sistemas)

_____ Modelo de análisis estructurado.

_____ Modelo Prototipos.

_____ Modelo evolutivo. (Prototipo funcional)

_____ Modelo en espiral.

_____ Ninguno

Otros _____

Cuadro 14 Conocimiento de los profesores sobre las metodologías de desarrollo de sistemas de información.

		Responses		Percent of Cases
		N	Percent	
Conocimiento - metodología de desarrollo	Modelo tradicional	8	22,9%	80,0%
	Modelo de análisis estructurado	6	17,1%	60,0%
	Modelo Prototipos	6	17,1%	60,0%
	Modelo evolutivo	6	17,1%	60,0%
	Modelo en espiral	6	17,1%	60,0%
	Ninguno	1	2,9%	10,0%
	Otro	2	5,7%	20,0%
Total		35	100,0%	350,0%

Elaborado en SPSS 13.

De acuerdo con el resultado obtenido se puede observar que ocho (8) de los diez (10) profesores conocen o dominan el modelo tradicional, seis (6) profesores el modelo de análisis estructurado, seis (6) profesores el modelo de prototipos, seis (6) profesores el modelo evolutivo, seis (6) profesores el modelo en espiral y hay un (1) profesor que no conoce o domina ninguna metodología de desarrollo de sistemas; dos (2) profesores indicaron que conocen otra metodología que es el modelo orientado a objetos.

4. ¿Esta usted de acuerdo de que la Universidad Católica del Táchira incorpore la herramientas CASE dentro del contenido programático de la carrera Informática Gerencial?

_____ De acuerdo

_____ Ni de acuerdo ni en desacuerdo

_____ En desacuerdo

Cuadro 15 Aceptación por partes de los profesores sobre la incorporación de la herramienta CASE dentro del contenido programático de la carrera Informática Gerencial.

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos De acuerdo	9	90,0	90,0	90,0
Ni de acuerdo ni en desacuerdo	1	10,0	10,0	100,0
Total	10	100,0	100,0	

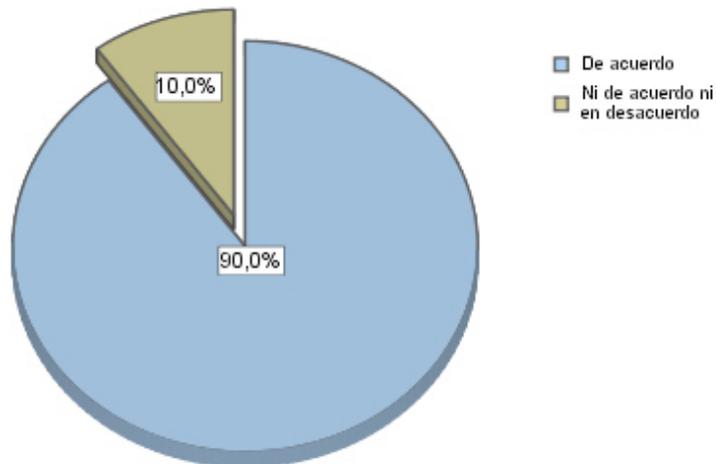


Gráfico 8 Resultado obtenido sobre la aceptación por partes de los profesores sobre la incorporación de la herramienta CASE dentro del contenido programático de la carrera Informática Gerencial. Elaborado en SPSS 13.

Al analizar el gráfico se observa que el 90 % de los profesores opinaron que están de acuerdo que la UCAT incorpore la herramientas CASE dentro del contenido programático de la carrera Informática Gerencial.

En caso que esté de acuerdo con la pregunta 4, conteste las siguientes preguntas.

5. Las herramientas CASE proveen muchos beneficios en todas las etapas de desarrollo de sistema de información, indique usted ¿En qué actividad(es) le gustaría hacer más énfasis con la utilización de esta herramienta CASE en la formación de los alumnos de Informática Gerencial?

- Realizar práctica que abarque todo el ciclo de vida del desarrollo de sistemas de información.
 - Automatizar el dibujo de diagramas como el diagrama de flujo de dato (DFD) y entre otros.
 - Ayudar en la creación de relaciones de la base de datos y la generación de scripts.
 - Ayudar en la documentación del sistema.
 - Generar código fuente automáticamente sin la necesidad de programar.
 - Todas las anteriores.
 - Ninguno.
- Otros _____
- _____
- _____

Según la pregunta anterior, como hay un (1) profesor que opinó que no está de acuerdo ni en desacuerdo con la incorporación de la herramientas CASE como software de aprendizaje, entonces a partir de la quinta pregunta quedan nueve (9) profesores que iban a contestar las preguntas restantes.

Resultado de la pregunta No. 5:

Cuadro 16 Opiniones de los profesores sobre el uso de las herramientas CASE en la formación de los alumnos de Informática Gerencial.

		Responses		Percent of Cases
		N	Percent	
Actividades para usar herramientas CASE	Realizar práctica que abarca todo el ciclo ...	3	23,1%	33,3%
	Automatizar el dibujo de diagramas	1	7,7%	11,1%
	Ayudar en la creación de relaciones de la ...	2	15,4%	22,2%
	Ayudar en la documentación del sistema	1	7,7%	11,1%
	Generar código fuente automáticamente	1	7,7%	11,1%
	Todas las anteriores	5	38,5%	55,6%
Total		13	100,0%	144,4%

Elaborado en SPSS 13.

Se puede observar en el resultado obtenido que tres (3) de los nueve (9) profesores respondieron que les gustaría hacer más énfasis en realizar prácticas que abarque todo el ciclo de vida del desarrollo de sistemas de información. Un (1) profesor le gustaría hacer más énfasis en automatizar el dibujo de diagramas como el de DFD. Dos (2) profesores les gustaría hacer más énfasis en ayudar en la creación de relaciones de la base de datos y la generación de scripts. Un (1) profesor le gustaría hacer más énfasis en ayudar en la documentación del sistema. Un (1) profesor le gustaría hacer más énfasis en generar código fuente automáticamente sin la necesidad de programar; y hay cinco (5) profesores opinaron que se realicen todas las actividades anteriores en la formación de los alumnos de Informática Gerencial.

6. ¿En qué asignatura(s) considera usted que podría(n) utilizar la herramienta CASE como software de aprendizaje?

De acuerdo con las informaciones recogidas, los profesores opinaron que se podría utilizar la herramienta CASE en las siguientes asignaturas:

- a. Análisis y Diseño de Sistema de Información o Sistema de información.
- b. Auditoria de Sistemas.
- c. Las asignaturas relacionadas con la Base de datos como Administración de Base de Datos, Estructura de Datos y Base de Datos.
- d. Programación.
- e. Sistema operativo.
- f. Ingeniería de Software.

7. ¿Considera usted que la Universidad Católica del Táchira debe capacitar al personal docente en el manejo o utilización de herramientas CASE?

Sí _____ No _____

Cuadro 17 Opiniones de los profesores acerca de la capacitación en el manejo de herramienta CASE.

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos Sí	9	100,0	100,0	100,0

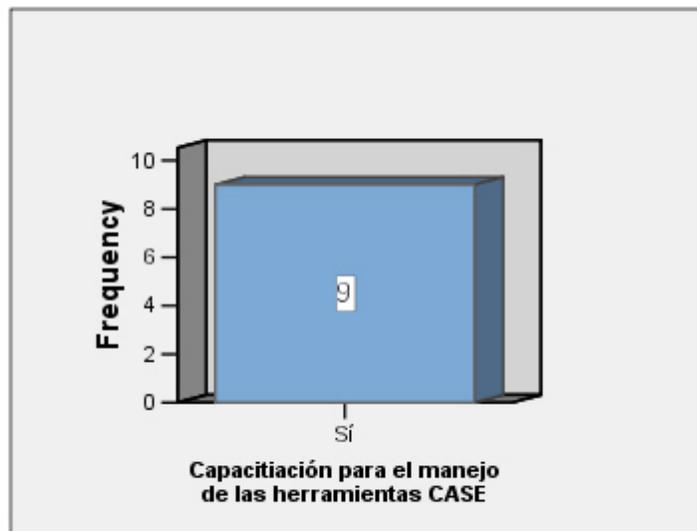


Gráfico 9 Resultado obtenido sobre las opiniones de los profesores acerca de la capacitación en el manejo de herramienta CASE.
Elaborado en SPSS 13.

Según el resultado de esta pregunta, todos los profesores consideraron que la Universidad Católica del Táchira debe capacitar a ellos con respecto al manejo y la utilización de las herramienta CASE.

**Cuestionarios dirigidos a los alumnos de tercero y cuarto año de
Informática Gerencial**

1. ¿Conoce usted las herramientas de Ingeniería de Software Asistida por Computadora (CASE)? Sí _____ No _____

Cuadro 18 Conocimiento de los alumnos acerca de las herramientas CASE.

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	Sí	19	43,2	43,2	43,2
	No	25	56,8	56,8	100,0
	Total	44	100,0	100,0	

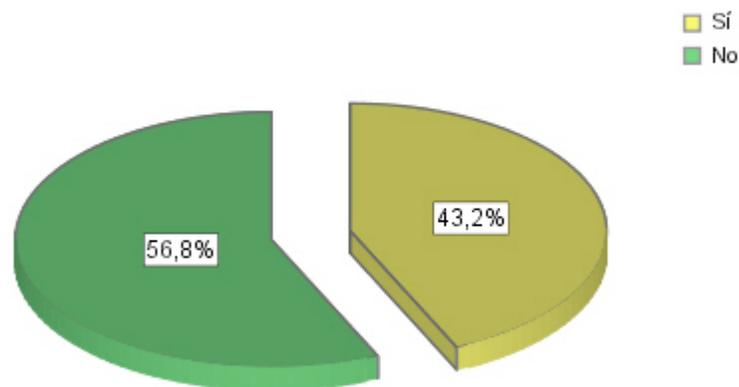


Gráfico 10 Resultado obtenido sobre el conocimiento de los alumnos acerca de la herramienta CASE.
Elaborado en SPSS 13.

A través del gráfico se puede observar que diecinueve (19) alumnos de tercero y cuarto año de Informática Gerencial respondieron que sí conocen lo que son las herramientas CASE, los cuales representan el 43.2% de total de los alumnos; mientras que veinticinco (25) alumnos indicaron que no las conocen. Es decir que el 56,8% de los alumnos de tercero y cuarto año de Informática Gerencial no conocen las herramientas CASE.

3. ¿Le gustaría aprender a utilizar una herramienta de desarrollo de sistema de información mediante el diseño de la estructura de la base de datos sin necesidad de programar o codificar? Sí _____ No _____

Cuadro 20 Aceptación por parte de los alumnos para aprender a utilizar una herramienta CASE.

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	Sí	40	90,9	90,9	90,9
	No	4	9,1	9,1	100,0
	Total	44	100,0	100,0	

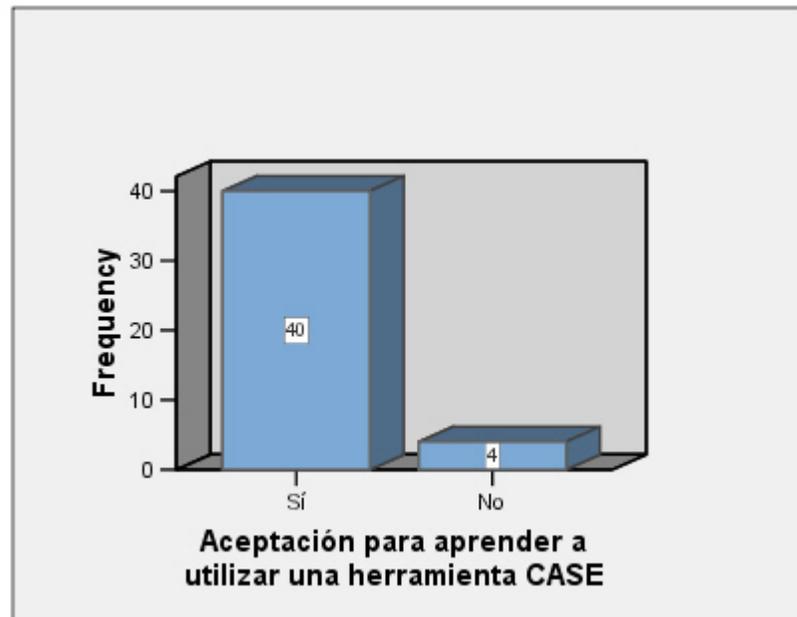


Gráfico 11 Resultado obtenido sobre la aceptación por parte de los alumnos para aprender a utilizar una herramienta CASE.
Elaborado en SPSS 13.

El gráfico expuesto representa que cuarenta (40) de los cuarenta cuatro (42) alumnos de Informática Gerencial demostraron su interés en aprender a manejar una herramienta CASE sin dedicarse a programar, los cuales representan el 90,9 % del total de los alumnos. Los cuatro (4) alumnos restantes opinaron que no les gustaría aprender a manejarlas.

4. ¿Le gustaría que la Universidad Católica del Táchira incorpore el manejo de la herramienta CASE dentro del contenido programático de la carrera Informática Gerencial? Sí _____ No _____

Cuadro 21 Aceptación por parte de los alumnos para incorporar de la herramienta CASE dentro del contenido programático de la carrera Informática Gerencial.

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	Sí	42	95,5	95,5	95,5
	No	2	4,5	4,5	100,0
	Total	44	100,0	100,0	

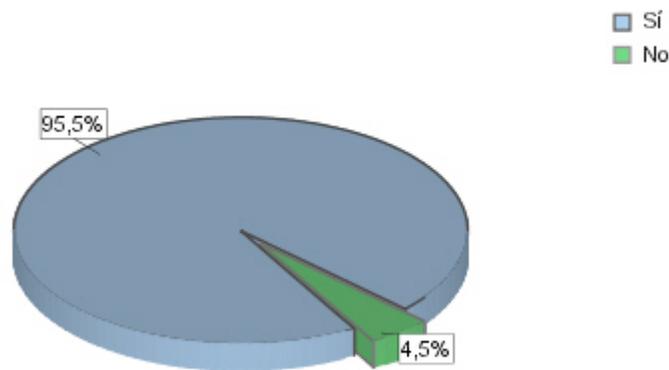


Gráfico 12 Resultado obtenido sobre la aceptación por parte de los alumno para la incorporación de la herramienta CASE dentro del contenido programático de la carrera Informática Gerencial. Elaborado en SPSS 13.

Al analizar la gráfica se observa que cuarenta dos (42) alumnos opinaron que sí les gustaría que la UCAT incorpore el manejo de la herramienta CASE dentro del contenido programático de la carrera Informática Gerencial, los cuales representan el 95.5% del total de los alumnos.

CAPÍTULO V

PROPUESTA

Antes de presentar la propuesta se realizó una demostración de la herramienta GeneXus basándose en una situación simulada para crear varios prototipos funcionales que conforman el programa de aplicación. Se omitió los pasos previos al análisis de sistema porque se supone, ya fueron realizados. Se dedicó sólo a la creación de prototipos funcionales utilizando la versión de evaluación 2.4 de GeneXus 8.0. Ver el anexo A para conocer más sobre esta versión de prueba tales como la instalación y sus limitaciones.

Diseño de aplicación con GeneXus

La situación simulada es la siguiente: la empresa “Computadora Táchira C.A.” es una empresa comercial que se dedica a la compra y venta del hardware de computación. Se desea implementar un Sistema de Facturación de Ventas y Registro de Pedidos que permita al operador administrar las informaciones de sus clientes, productos y proveedores; realizar la facturación de ventas del hardware de computación y registrar los pedidos de hardware manteniendo actualizado el stock del producto y el precio de ventas.

En el momento de hacer la factura de ventas es necesario hacer varias consultas sobre las informaciones de clientes y de los productos para la venta; se quiere además que el sistema calcule el monto total de pago sumando el impuesto de valor agregado y mientras tanto se resta al cada uno de los stocks de los productos vendidos. Para el

registro de los pedidos se debe calcular el monto total gastado en la compra de hardware, sumando a cada stock de los productos comprados y actualizar el precio de venta, aplicando una política de ganancia del 50% sobre el precio del pedido para cada producto vendido.

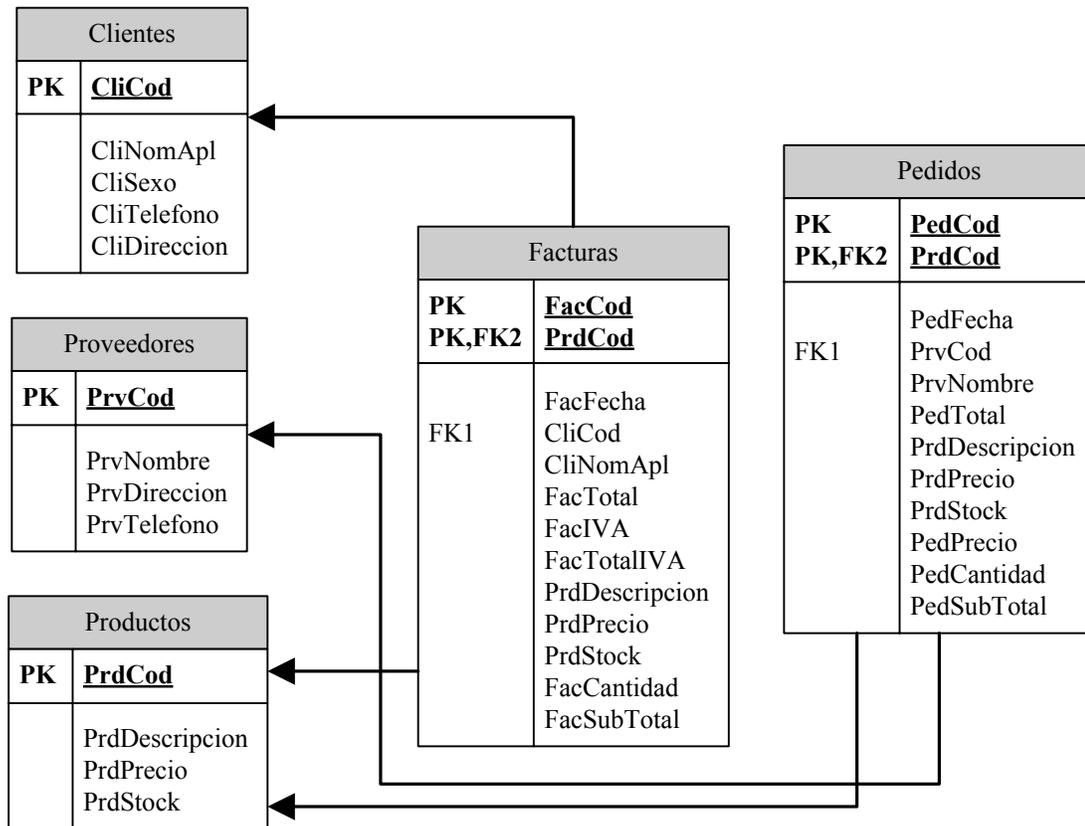


Gráfico 13 El diagrama de Entidad-Relación del Sistema de Facturación de Ventas y Registro de Pedidos. Elaborado en Microsoft Office Visio Professional 2003.

En el sistema propuesto se requieren las siguientes transacciones: Clientes, Proveedores, Productos, Pedidos y Facturas. En el gráfico 13 se muestra los atributos de cada entidad y sus relaciones entre distintas entidades. La punta de flecha indica una cardinalidad de 1 y la línea continúa donde termina la flecha indica muchos, quiere decir que la relación que existe entre ambas es de “uno a muchos”. **PK** significa la clave primaria y **FK** la clave foránea.

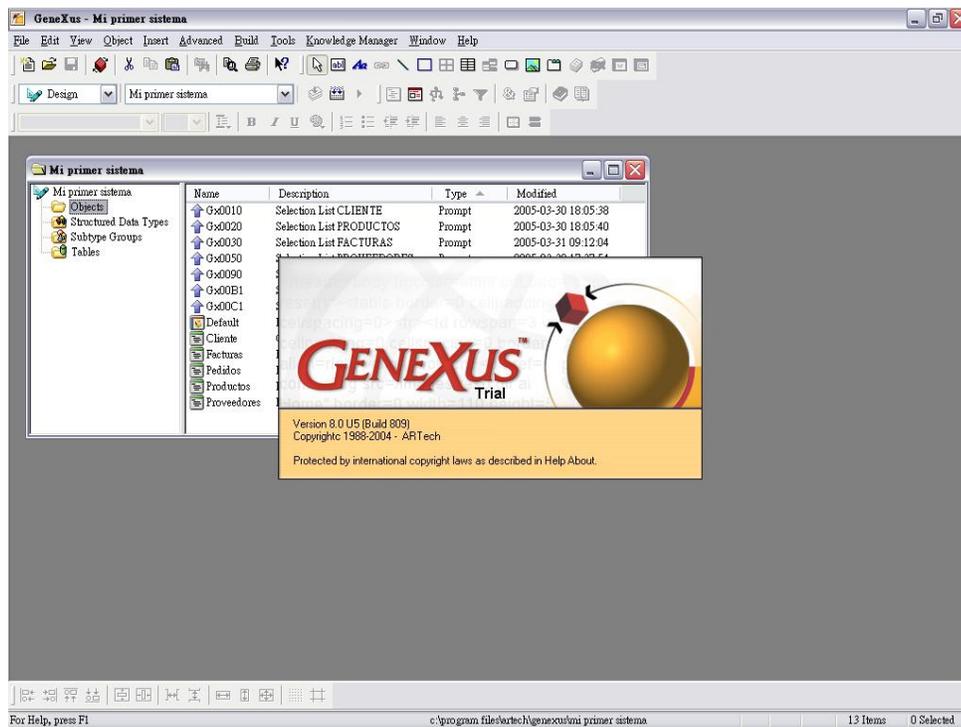


Gráfico 14 El ambiente de trabajo de la herramienta GeneXus 8.0 Versión de Evaluación 2.4.

La creación de la base de conocimiento

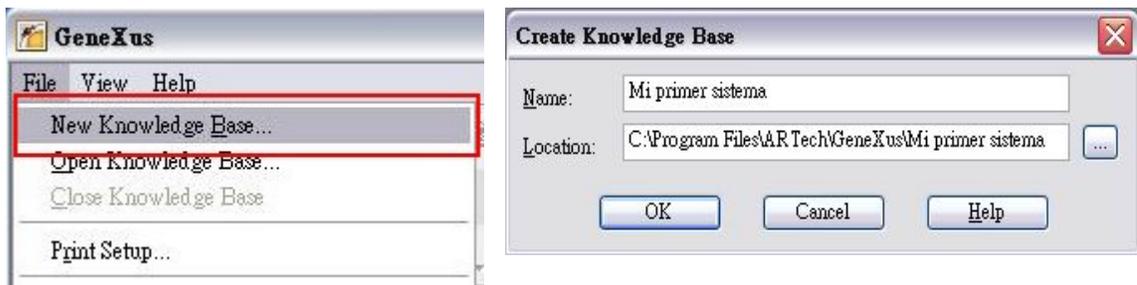


Gráfico 15 La creación de base de conocimiento.

El primer paso en el desarrollo de una aplicación con GeneXus es seleccionar el ítem *New Knowledge Base* del menú *File* para crear la base de conocimiento. Después de crearla, se abre una pantalla que aparece dividida en dos regiones: A la izquierda es un árbol de carpetas como *Objects* (Objetos), *Structured Data Types* (Tipos de dato estructurado) y entre otros; y a la derecha: aparece una lista con columnas *Name* (Nombre), *Description* (Descripción), *Type* (Tipo) y *Modified* (Modificado).

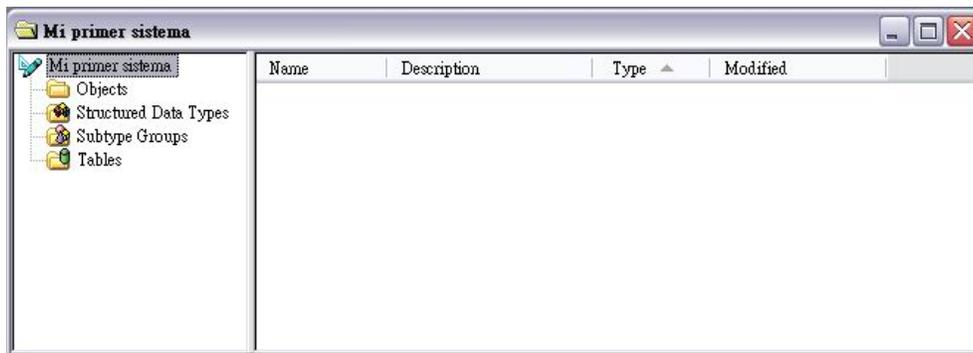


Gráfico 16 La base de conocimiento.

La creación del modelo de diseño

El modelo de Diseño es una representación lógica del sistema, se debe indicar cuáles son los requerimientos asociados con el sistema al cual se quiere crear. Cabe destacar que este modelo no está asociado a un lenguaje de programación ni base de datos. Se crea automáticamente al crear la base de conocimiento.

Crear los objetos GeneXus

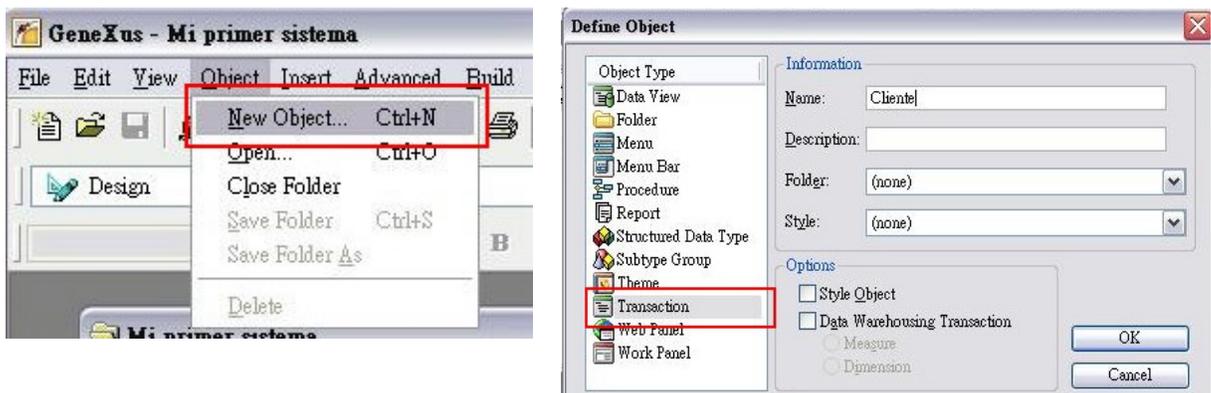


Gráfico 17 La creación de un objeto de transacción.

Se define primero los objetos de transacciones mediante los cuales GeneXus extrae el conocimiento necesario para diseñar el modelo de datos en la tercera forma normal. Luego se van definiendo los demás objetos que correspondan.

La transacción “Clientes”

Para crear el objeto de transacción se debe abrir el menú *Object / New Object* y seguir los pasos que indican GeneXus.

En la siguiente tabla, se presenta los atributos que conforman la estructura de la transacción Clientes:

Cuadro 22 Los atributos de la estructura de la transacción Clientes

Atributos	Descripción de atributos	Tipos de datos
<i>CliCod*</i>	Código de cliente.	Numérico(6.0)
<i>CliNomApl</i>	Nombre y apellido de cliente.	Caracteres(50)
<i>CliSexo</i>	Sexo de cliente.	Caracteres(1)
<i>CliTelefono</i>	Teléfono de cliente.	Caracteres(15)
<i>CliDireccion</i>	Dirección de cliente.	Caracteres(60)

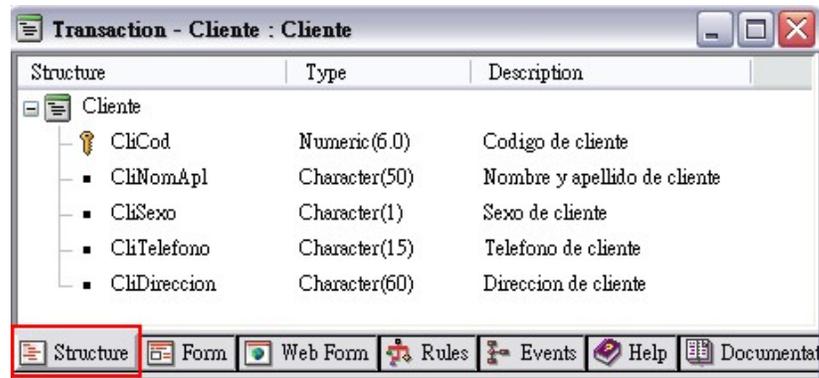


Gráfico 18 La estructura de la transacción Clientes.

Se define la estructura, los tipos de datos, las descripciones de transacción Clientes para luego a través de GeneXus generar un prototipo funcional o aplicación. En cuanto a los tipos de datos se pueden clasificar en: *Numeric* (datos numéricos), *Character* (Cualquier tipo de textos), *VarChar* (textos de larga variable), *Long VarChar* (largos textos o comentarios), *Date* (Fecha), *DateTime* (Fecha y tiempo) y *Blob* (Cualquier tipo de información: texto, imágenes y video).

Después de crear los atributos, estableciendo los tipos de datos y las descripciones de la transacción Clientes, sólo tiene que guardarla, GeneXus generará por defecto la interfaz Form y Web Form de la aplicación. Se podrán realizar cambios en la interfaz mediante la propiedad de control Atributos/Variables en la cual se establece los tipos de controles como *Combo Box*, *Radio Button*, *Check Box* y entre otras opciones más.

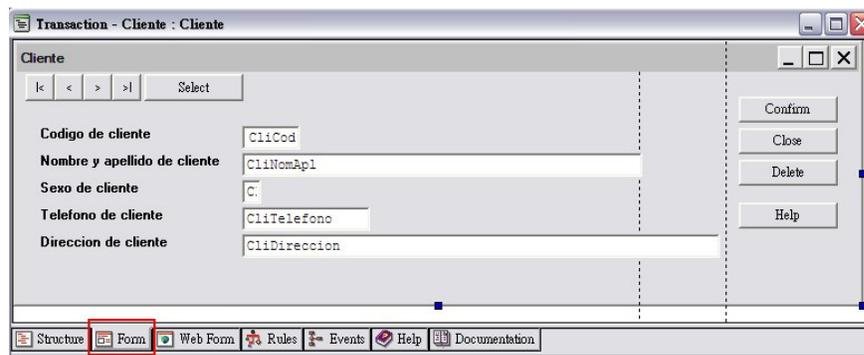


Gráfico 19 La interfaz Form de la transacción Clientes.



Gráfico 20 La interfaz Form personalizada de la transacción Clientes.

Pero después de personalizar la interfaz Form, cada vez que se cambie los atributos de una estructura de transacción, GeneXus ya no actualizará la interfaz automáticamente, por lo que se deberá actualizarla manualmente. Para que se actualice automáticamente se debe ir al menú *Edit* y seleccionar la opción *Default Data Area*.

GeneXus genera la interfaz de dos tipos de diálogo para las transacciones: campo a campo y pantalla completa (full-screen). En el diálogo campo a campo se controla su validez cada vez que introduce un valor en un campo. Se utiliza este tipo para el interfaz Form. El diálogo a pantalla completa (full-screen) se realizan todas las validaciones después de introducir todos los campos de la pantalla. Este último se utiliza más que todo en el ambiente Web.

Se aplican los pasos anteriores para crear las transacciones de Proveedores, Productos, Pedidos y Facturas.

La transacción “Proveedores”

En la siguiente tabla, se presenta los atributos que conforman la estructura de la transacción Proveedores:

Cuadro 23 Los atributos de la estructura de la transacción Proveedores

Atributos	Descripción de atributos	Tipos de datos
<i>PrvCod*</i>	Código de proveedor.	Numérico(6.0)
<i>PrvNombre</i>	Nombre de proveedor.	Caracteres(30)
<i>PrvTelefono</i>	Teléfono de proveedor.	Caracteres(15)
<i>PrvDireccion</i>	Dirección de proveedor.	Caracteres(30)



Gráfico 21 La estructura de la transacción Proveedores.

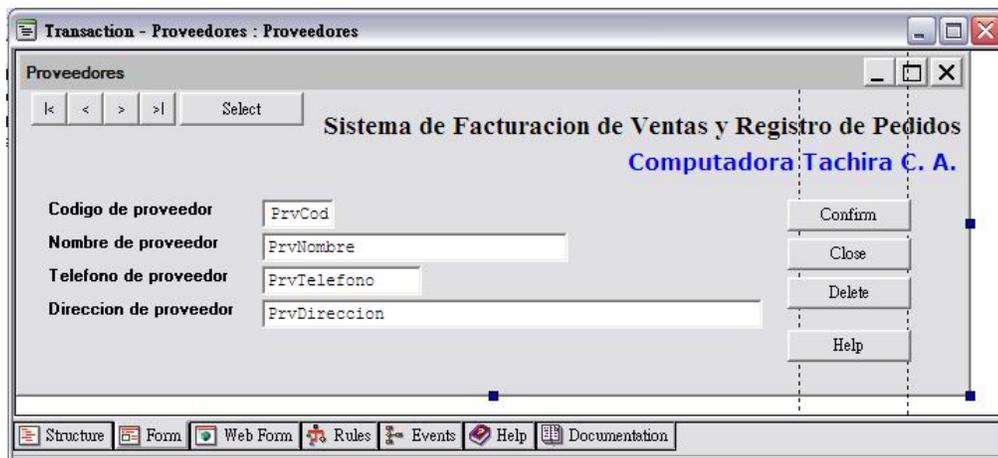


Gráfico 22 La interfaz Form de la transacción Proveedores.

La transacción “Productos”

En la siguiente tabla, se presenta los atributos que conforman la estructura la estructura de la transacción Productos:

Cuadro 24 Los atributos de la estructura de la transacción Productos.

Atributos	Descripción de atributos	Tipos de datos
<i>PrdCod*</i>	Código de producto.	Numérico(6.0)
<i>PrdDescripcion</i>	Descripción de producto.	Caracteres(40)
<i>PrdPrecio</i>	Precio de producto.	Numérico(10.2)
<i>PrdStock</i>	Stock de producto.	Numérico(4.0)

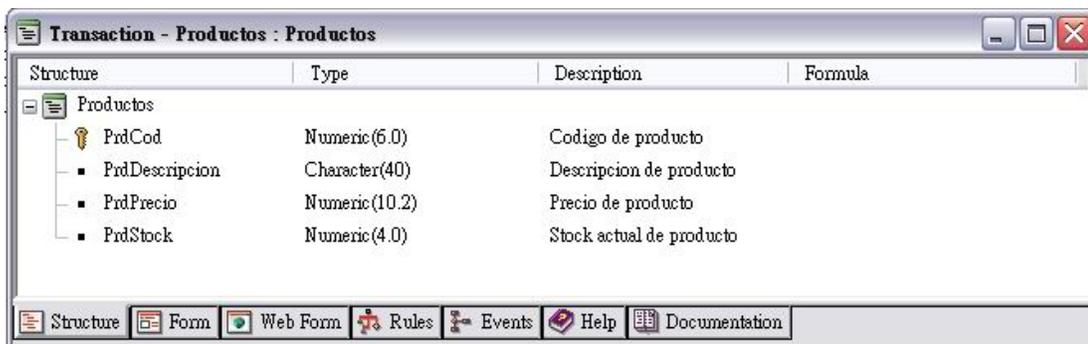


Gráfico 23 La estructura de la transacción Productos.

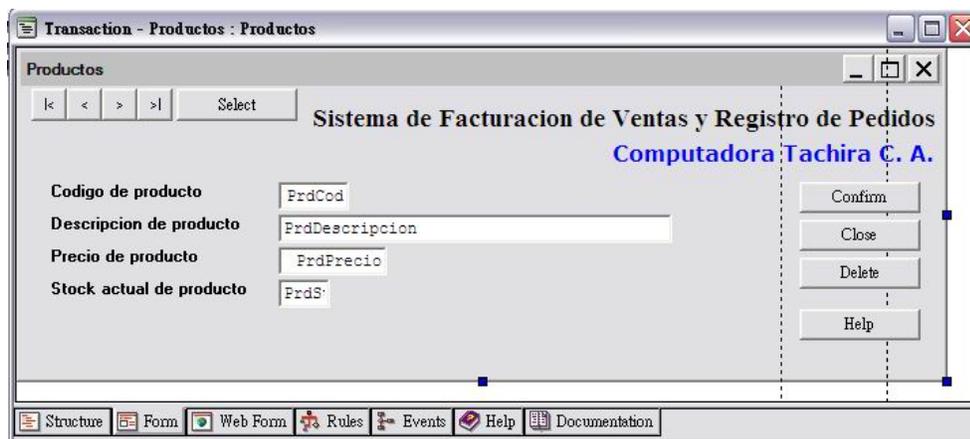


Gráfico 24 La interfaz Form de la transacción Productos.

La transacción “Facturas”

En la siguiente tabla, se presenta los atributos que conforman la estructura de la transacción Facturas:

Cuadro 25 Los atributos de la estructura de la transacción Facturas (de 2 niveles).

Atributos	Descripción de atributos	Tipos de datos	Fórmulas
FacNro*	Número de factura.	Numérico(6.0)	
FacFecha	Fecha de factura .	Date	
CliCod	Código de cliente.	Numérico(6.0)	
CliNomApl	Nombre de cliente.	Caracteres(50)	
FacTotal	Importe total de la factura.	Numérico(10.2)	SUM(FacSubTotal)
FacIVA	Impuesto de Valor Agregado.	Numérico(10.2)	FacTotal * 0.15
FacTotalIVA	Total al pagar + I.V.A.	Numérico(10.2)	FacTotal + FacIVA
(PrdCod*	Código de producto.	Numérico(6.0)	
PrdDescripcion	Descripción de producto.	Caracteres(40)	
PrdPrecio	Precio de producto.	Numérico(10.2)	
PrdStock	Stock de producto.	Numérico(4.0)	
FacCantidad	Cantidad llevada de producto.	Numérico(4.0)	
FacSubTotal)	Importe por producto de la línea.	Numérico(10.2)	PrdPrecio * FacCantidad

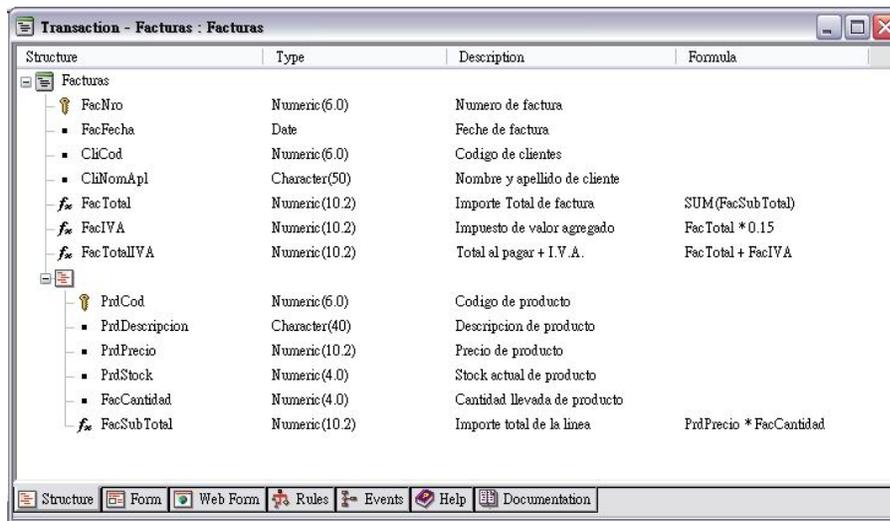


Gráfico 25 La estructura de la transacción Facturas.

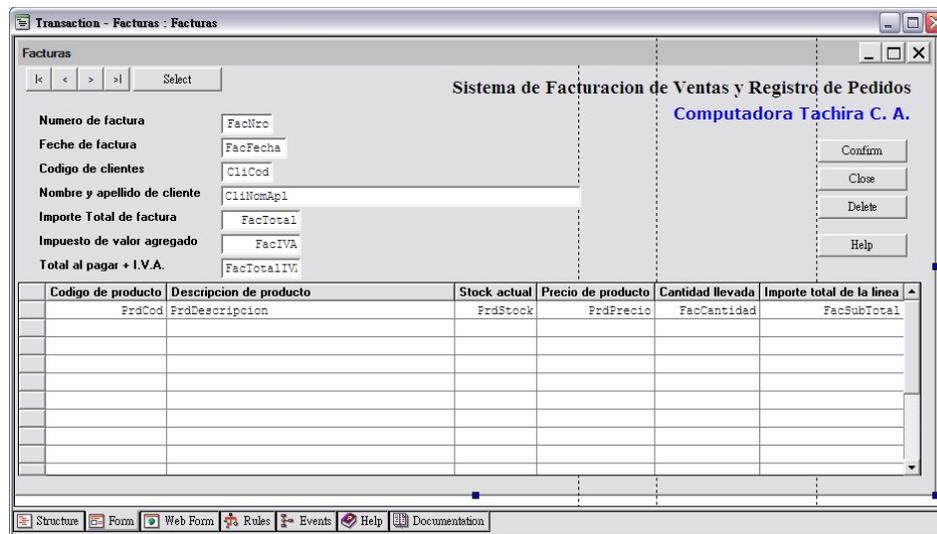


Gráfico 26 La interfaz Form de la transacción Facturas.

La transacción “Pedidos”

En la siguiente tabla, se presenta los atributos que conforman la estructura de la transacción Pedidos:

Cuadro 26 Los atributos de la estructura de la transacción Pedidos (de 2 niveles).

Atributos	Descripción de atributos	Tipos de datos	Fórmulas
PedCod*	Código de pedido.	Numérico(6.0)	
PedFecha	Fecha de pedido .	Date	

PrvCod	Código de proveedor.	Numérico(6.0)	
PrvNombre	Nombre de proveedor.	Caracteres(30)	
PedTotal	Total pedido al pagar.	Numérico(10.2)	SUM(PedSubTotal)
(PrdCod*	Código de producto.	Numérico(6.0)	
PrdDescripcion	Descripción de producto.	Caracteres(40)	
PrdPrecio	Precio de producto.	Numérico(10.2)	
PrdStock	Stock de producto.	Numérico(4.0)	
PedPrecio	Precio de pedido.	Numérico(10.2)	
PedCantidad	Cantidad pedida de producto.	Numérico(4.0)	
PedSubTotal)	Importe por producto.	Numérico(10.2)	PedPrecio * PedCantidad

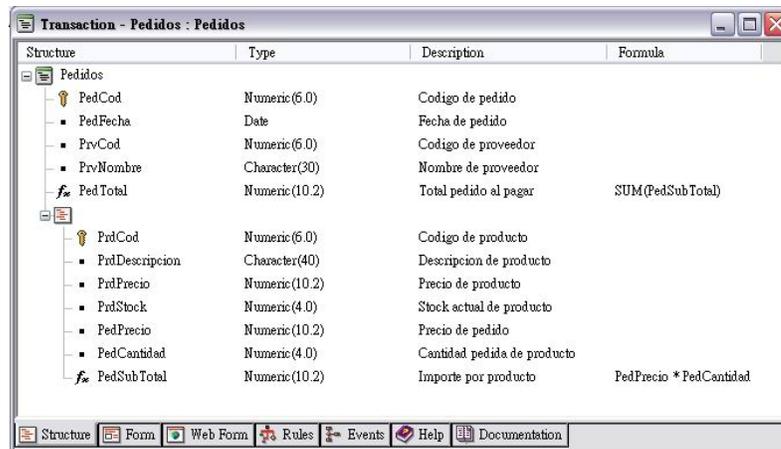


Gráfico 27 La estructura de la transacción Pedidos.

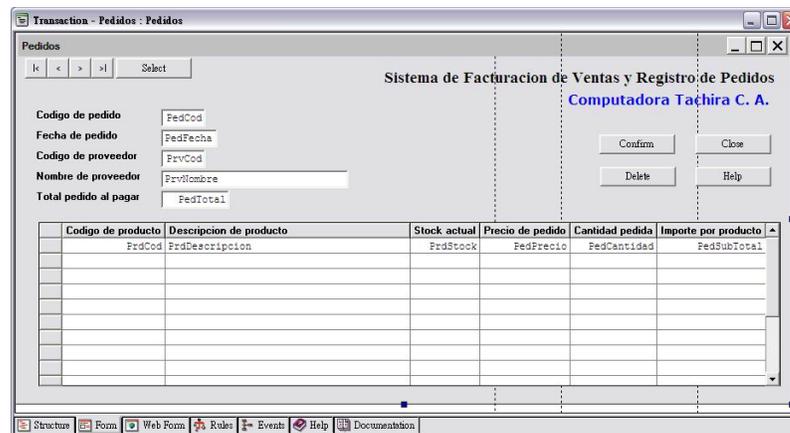


Gráfico 28 La interfaz Form de la transacción Pedidos.

Definir el dominio

Se define el dominio *Códigos* con tipo de datos **Numérico (6.0)** a los atributos *CliCod*, *PrvCod*, *PrdCod* y *FacNro* y *PedCod*. La función de dominio es compartir definiciones similares entre distintos atributos que no tengan ninguna relación. Esto permite realizar definiciones genéricas para futuros cambios de una vez sola a los atributos asociados. Para establecerlo abre el menú *Advanced* y seleccione *Domain* o a través del botón *New Domain* de la propiedad de los atributos.

También se define un nuevo dominio *Importes*, con tipo de datos **Numérico (10.2)** y este lo asocia a los atributos *PrdPrecio*, *FacTotal*, *FacIVA*, *FacTotalIVA*, *FacSubTotal*, *PedPrecio*, *PedSubTotal*, *PedTotal*, *PedIVA* y *PedTotalIVA*.

Insertar las fórmulas

A diferencia de las transacciones Clientes, Proveedores y Productos, se agregan las fórmulas para realizar cálculos, tales como importe total de la factura, impuesto de valor agregado y otros cálculos. Existen cuatro tipos de fórmulas: horizontales, verticales, Aggregate / Select y compuesta.

Fórmulas horizontales: Permite definir una o varias expresiones condicionales. Los atributos involucrados deberán pertenecer a la misma tabla extendida.

Fórmulas verticales: Sólo hay dos fórmulas verticales que son *sum* y *count*. La fórmula *sum* permite sumar y la de *count* permite contar todas las ocurrencias del atributo referenciado en la tabla en la que se encuentre.

Fórmulas Aggregate / Select: Permite realizar las funciones de buscar, sumar, contar atributos que cumplan condiciones establecidas en cualquier tabla del modelo.

Fórmula compuesta: Está fórmula se compone de varias fórmulas horizontales y Aggregate/Select condicionales.

En la transacción **Facturas**, se tiene las siguientes fórmulas:

1. Fórmula vertical:

$FacTotal = SUM(FacSubTotal)$ Importe total de la factura = la suma de los importes por producto de la línea

2. Formula horizontal:

$FacIVA = FacTotal * 0.15$ Impuesto de valor agregado = Importe total * 15 %

$FacTotalIVA = FacTotal + FacIVA$ Total al pagar + I.V.A.= Importe total + I.V.A.

$FacSubTotal = PrdPrecio * FacCantidad$ Importe por producto de la línea = Precio * Cantidad llevada

En la transacción **Pedidos**, se tiene las siguientes fórmulas:

1. Fórmula vertical:

$PedTotal = SUM(PedSubTotal)$ Total pedido al pagar = la suma de los importes por producto

2. Formula horizontal:

$PedSubTotal = PedPrecio * PedCantidad$ Importe por producto = Precio de pedido * Cantidad pedida

Insertar las reglas para los objetos GeneXus

La definición de reglas permite programar el comportamiento de un objeto de transacción. Se debe escribir de forma *declarativa* y se puede involucrar a los atributos definidos dentro de la misma estructura de la transacción. Para utilizar esta

función abre primer el objeto de transacción que desea insertar las reglas, luego haz el clic sobre el menú *Object* y seleccione la opción *Rules*.

1. Reglas para la transacción Clientes.
 - a. **Error**('No se permite cliente sin nombre.') **if null**(CliNomApl);
Si el campo del nombre de cliente es vacío aparecerá el mensaje de error “No se permite cliente sin nombre”.

2. Reglas para la transacción Facturas.
 - a. **Default**(FacFecha,&Today);
Si no asigna valor en el campo fecha de factura, se tomará automáticamente el valor de la fecha del día para el registro insertado.
 - b. **Noaccept**(FacFecha) **if Update**;
Si se está modificando una factura en el modo actualización (Update), no se permite que se modifique su fecha.
 - c. **Error**('No se permite eliminar la factura.') **if Delete**;
Si se intenta eliminar la factura, o sea, cuando haz el clic el botón *Delete* aparecerá el mensaje de error que indica que “No se permite eliminar la factura”.
 - d. **Error**('La cantidad llevada debe ser mayor que 0.') **if FacCantidad =0**;
Si la cantidad llevada es menor igual al cero aparecerá el mensaje de error que indica que “La cantidad llevada debe ser mayor que 0”.
 - e. **Error**('Se sobrepasa el stock máximo del producto.') **if FacCantidad > PrdStock**;
Si la cantidad llevada es mayor que la cantidad existente en el stock aparecerá el mensaje de error que indica que “Se sobrepasa el stock máximo del producto”.
 - f. **Subtract**(FacCantidad, PrdStock);
Si se está insertando una línea de facturación de venta, el stock del producto va restando a la cantidad llevada por el cliente.

3. Reglas para la transacción Pedidos.
 - a. **Default**(PedFecha,&Today);
Si no asigna valor en el campo fecha de pedido, se tomará automáticamente el valor de la fecha del día para el registro insertado.
 - b. **Noaccept**(PedFecha) **if** Update;
Si se está modificando un pedido en el modo actualización (Update), no se permite que se modifique su fecha.
 - c. **Add**(PedCantidad, PrdStock);
Si se está insertando una línea del pedido de producto, la cantidad pedida se va sumando al stock del producto.
 - d. $\text{PrdPrecio} = \text{PedPrecio} * 1.5$ **if** insert;
Si inserta el monto del precio de pedido, el precio de venta del producto va a ser precio de pedido por 1.5, o sea, se establece 50% como ingreso por cada venta del producto.
 - e. **Error**('La cantidad pedida debe ser mayor que 0.') **if** PedCantidad ≤ 0 ;
Si la cantidad pedida es cero, aparecerá un mensaje de error que indica que “La cantidad pedida debe ser mayor que 0”.

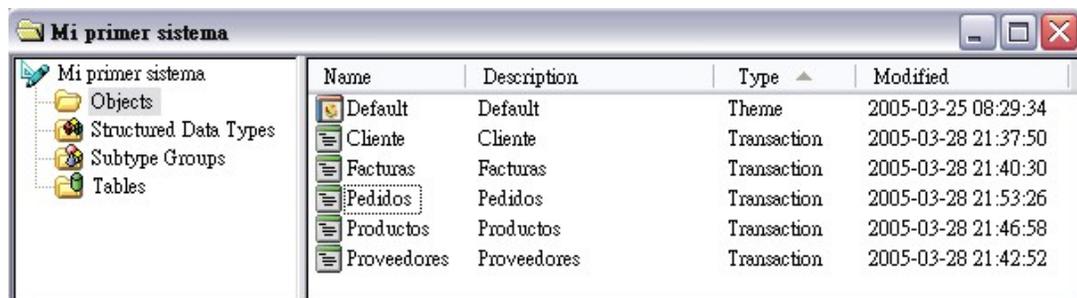


Gráfico 29 El modelo de diseño después de crear todas las transacciones.

Se puede insertar también las reglas a los objetos GeneXus en el modelo de Prototipo, GeneXus encargará de actualizar a los objetos que están en el modelo de Diseño mediante la especificación de objetos.

La creación de prototipo funcional

El propósito de crear un modelo de Prototipo consiste en probar el sistema definido en el modelo de Diseño y asociar una plataforma o ambiente de trabajo. Para crearlo se debe abrir el menú *File* y seleccione la opción *New Model* o seleccione *Prototype* en el combo box con los tipos de modelos directamente por la barra de herramientas; y seguir los pasos indicados.



Gráfico 30 La creación de un modelo de prototipo.

Seleccionar el ambiente de trabajo

Para crear un prototipo, se debe asociarle una plataforma o ambiente de trabajo. En el presente estudio la arquitectura está basada en *cliente/servidor*.



Gráfico 31 Los pasos para crear el modelo de prototipo.

- El servidor: se utiliza una versión de libre distribución del DBMS local de *Microsoft SQL Server Desktop Engine (MSDE)*. Cuando se instala SQL Server, se crean cuatro bases de datos, las cuales se llaman Bases de datos del sistema que están conformado por Master, Model, Tempdb y Msdb. La que va a utilizar por el presente estudio es la de *master*. Ella es la encargada de guardar la

información que se utiliza con las operaciones básicas de SQL Server.

- b. El cliente: el método de conexión a la base de datos es *ODBC* y el lenguaje de programación elegido es *.NET* con interfaz de usuario Windows.

Se activa la propiedad *Client Side Validation*. Es una propiedad específica de los Generadores *.NET* y Java para aplicaciones con interfaz Windows que provee más interacción con el usuario frente al diálogo a pantalla completa (full screen). Si la propiedad es activada este valor, en la medida que el usuario final vaya ingresando información a los campos, se irán validando los datos. Todas estas operaciones se ejecutarán en forma interactiva para que el usuario final pueda ir viendo resultados. Esta función es desactivada por defecto que indica que el diálogo a pantalla completa será puro sin agregar las validaciones en forma interactiva.

Cuando pasa desde el modelo de Diseño al de Prototipo, GeneXus compara las definiciones realizadas en el modelo de Diseño con respecto a las definiciones del modelo al cual se está ingresando. Esto se produce un análisis de impactos (*Impact Analysis*) o de creación de base de datos (*Database Creation Analysis*) si es por primera vez.

Realizar el análisis de impactos y la especificación

GeneXus analiza el impacto causado por la definición en el modelo de Diseño sobre el modelo de Prototipo. Se creará un reporte como resultado de dicho análisis que informa al desarrollador del sistema qué cambios físicos o estructurales habría que realizar en la base de datos asociada al modelo de Prototipo en cuestión. Es necesario leer todo el informe antes de decidir si proseguirá o no el cambio sobre la base de datos.

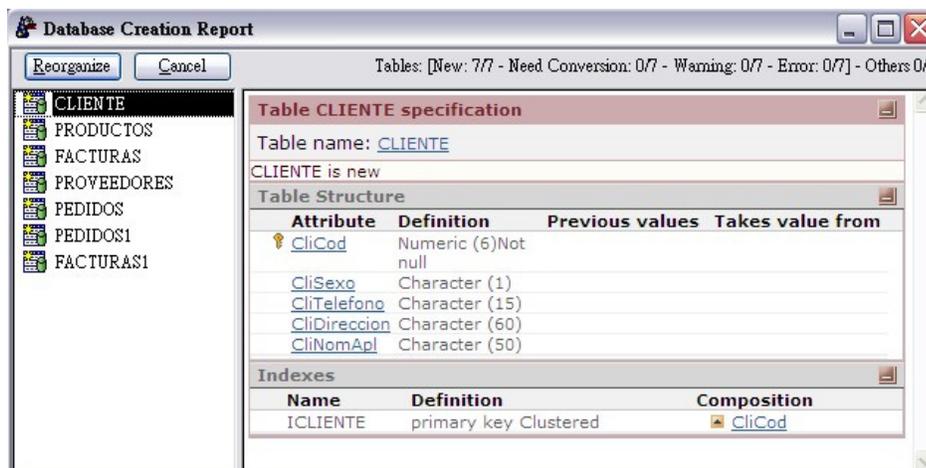


Gráfico 32 El reporte de análisis de creación de base de datos.

Si el desarrollador del sistema está de acuerdo con los cambios estructurales informados en el reporte de análisis de impacto, proseguirá a reorganizar la base de datos donde se efectuará los cambios físicos en la base de datos. De ahí se crean los objetos definidos en el modelo de Diseño,

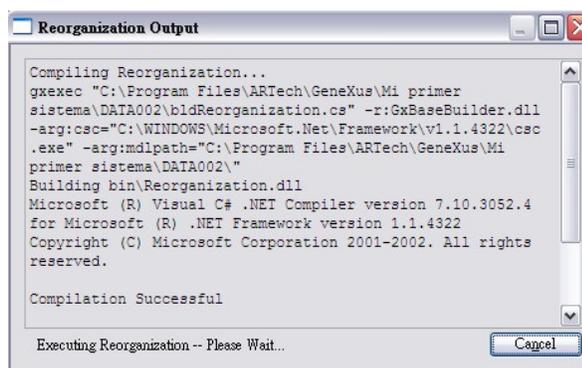


Gráfico 33 La ejecución de la reorganización.

las tablas de las transacciones que correspondan y la base de datos asociada al modelo de Prototipo. En el presente caso de estudio, la base de datos es creada al momento de crear el modelo de Prototipo; debido a que la plataforma es cliente/servidor. Si el ambiente de trabajo es centralizado, no se debe tener una base de datos creada.

Si no está de acuerdo con lo que está informado en el reporte de análisis de impacto, no efectuará cambios sobre la base de datos y volver al modelo de Diseño para realizar las redefiniciones que se consideren convenientes.

Después de efectuar una reorganización, se copiarán automáticamente las nuevas definiciones del modelo de Diseño, al modelo de Prototipo en el cual se está

trabajando. Así se creó el modelo de Prototipo.

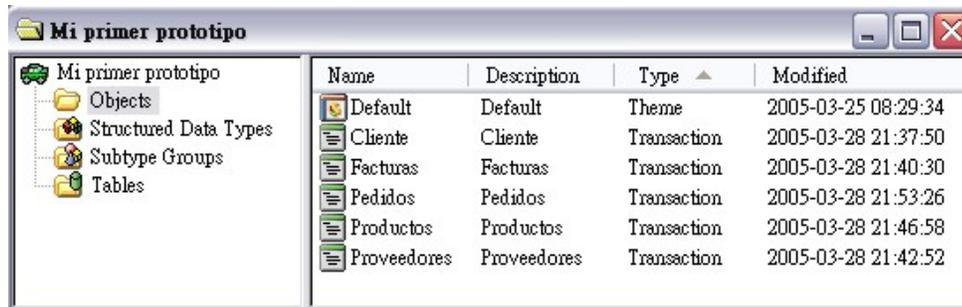


Gráfico 34 El modelo de prototipo.

El desarrollador del sistema debe indicar al GeneXus especificar y generar los programas de aplicación asociados a los objetos GeneXus en el lenguaje elegido en la plataforma del modelo. De esta manera, GeneXus analizará todo lo definido en cada uno de los elementos que lo componen, la sintaxis de las definiciones y la validez de lo definido.



Gráfico 35 El reporte de la especificación.

Para especificar se debe seleccionar la opción *Specify* en el menú *Build* y seguir los pasos indicados. Mediante dicho proceso GeneXus toma en cuenta el control de integridad referencial y la normalización de una base de datos; y mostrará un reporte de la especificación llamado listado de navegación, en el cual le informará la lógica que ha interpretado y los errores si los hay.

Luego de verificar el listado de navegación, el desarrollador del sistema podrá indicar a GeneXus que prosiga con la generación de los programas de aplicación en el lenguaje elegido. GeneXus genera nuevos objetos especificados automáticamente durante la especificación. Hasta aquí los programas .NET generados ya están listos para ejecutarlos.

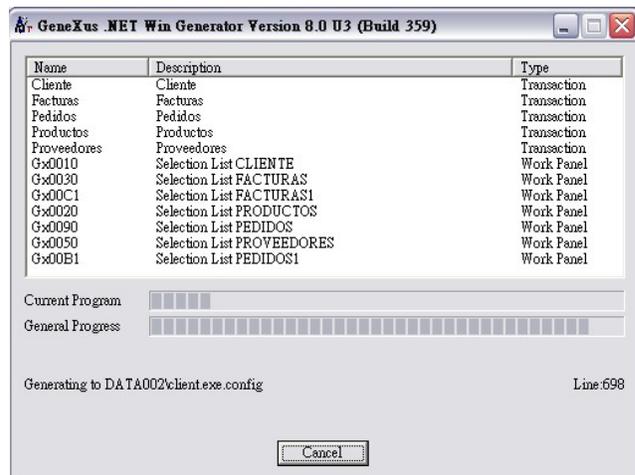


Gráfico 36 La generación del programa.

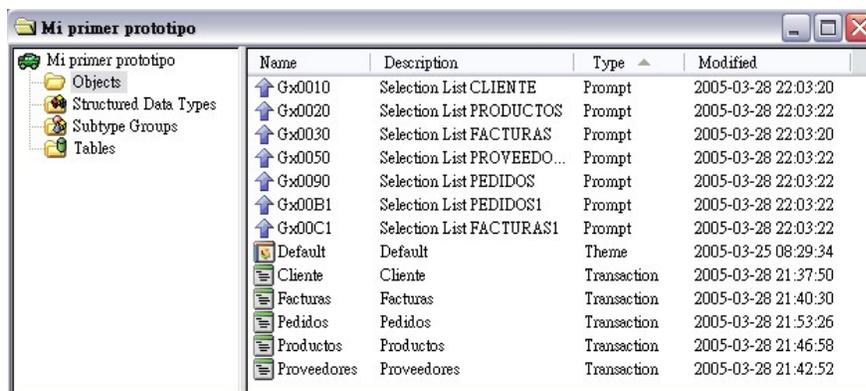


Gráfico 37 El modelo de prototipo después de la especificación.

Ejecutar el prototipo funcional

Se selecciona la opción *Run* del menú *Build* para ejecutar el prototipo funcional. Se abrirá un diálogo de ejecución y se ofrecen las funciones de compilar y de ejecutar el programa *Developer Menu*. De esta manera, el desarrollador del sistema podrá insertar, modificar y eliminar registros en las tablas Clientes, Productos, Proveedores, Facturas

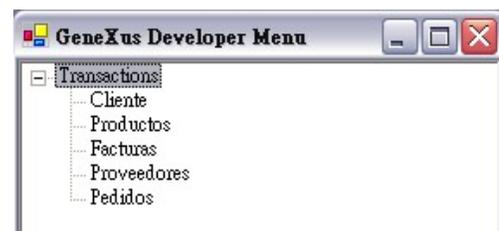


Gráfico 38 El programa Developer Menu.

y Pedidos, ejecutando las transacciones “Clientes”, “Productos” y “Facturas” de su aplicación respectivamente para probar el comportamiento de los programas .NET generados.

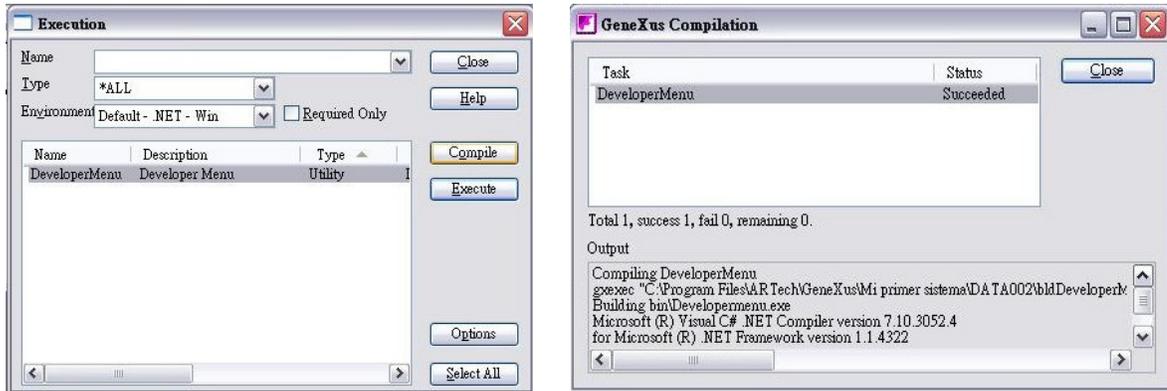


Gráfico 39 La ventana de diálogo de ejecución y de la compilación del programa.

Se supone que después de que el operador realizó prueba del sistema creado, la empresa ha presentado el informe para cambiar algunos de los requerimientos del sistema actual. Se plantean las siguientes modificaciones: en la estructura de la transacción les faltaron dos elementos que son la cédula de identidad y la nacionalidad; los campos con los tipos de datos Numérico(10,2), diez espacios y dos para números decimales, asociados con los precios debido a que por el factor de inflación, ya no son suficientes y se deben aumentar en quince espacios, es decir Numérico(15,2), para adaptar a la situación actual; y por está misma causa, las ventas de hardware han bajado mucho, la empresa ha decidido cambiar la política de ganancia por venta de 50% a la de 35%, con el fin de incentivar las ventas.

Para satisfacer estos requerimientos de la empresa se realizan los siguientes cambios:

1. Se pasa al modelo de Diseño, se agrega los siguientes atributos en la transacción Clientes:

Cuadro 27 Los nuevos atributos de la transacción Clientes.

Atributos	Tipo de datos	Descripciones
CliCI	Caracteres(10)	Cédula de identidad
CliNac	Caracteres (20)	Nacionalidad

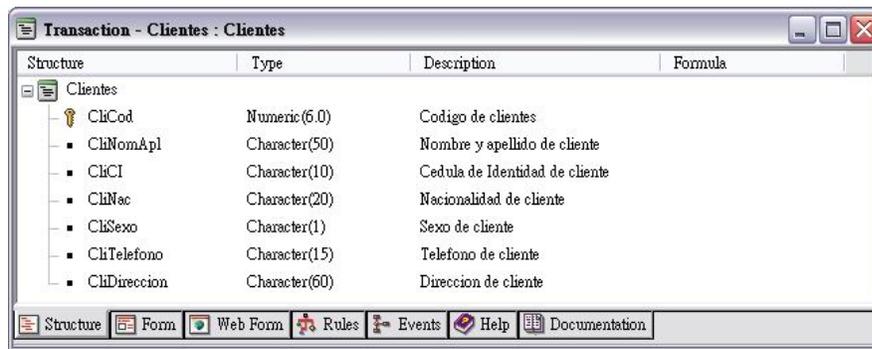


Gráfico 40 La nueva estructura de la transacción Clientes.



Gráfico 41 La nueva interfaz Form de la transacción Clientes.

- Para ajustar la longitud de los campos relacionados con el precio (estos campos ya estaban agrupados en un dominio llamado *Importes* cuando se diseñó el sistema) lo que se tiene que hacer es abrir el menú *Advanced* y seleccionar la opción *Domain*. De ahí aparece una ventana *Domain Selector*, es aquí en donde se guardan todo los dominios definidos por el desarrollador del sistema, se selecciona el dominio *Importes*, se presiona el botón *Edit* y se sustituye el valor de longitud 10 a 15.
- Se abre la transacción *Pedidos*, selecciona el ítem *Rules* del menú *Object* y cambia 1.5 de la regla “*PrdPrc = PedPrc * 1.5 if insert;*” a 1.35.
- Se pasa de nuevo al modelo de Prototipo, inmediatamente GeneXus hace una pregunta al desarrollador del sistema ¿desea realizar análisis de impacto? Y se repite el mismo proceso para crear el prototipo funcional. GeneXus presenta el

reporte de análisis de impacto para luego ejecutarse en la reorganización. Así se crea una nueva versión de la base de datos. Por último, se especifica y genera a los objetos existentes que han sido modificados y los objetos nuevos que han sido definidos o creados, obteniendo así los nuevos programas de aplicación listos para que el operador realice la prueba de nuevo. Este proceso de mantenimiento de sistema lo hace automáticamente por la herramienta GeneXus.

Este ciclo de Diseño-Prototipo se recorre repetidamente, construyendo y probando sucesivos prototipos funcionales de acuerdo con la necesidad de hacer modificaciones en la base de conocimiento por razón del cambio de las visiones de los usuarios o de las correcciones de eventual problema ocurrido. Se termina cuando cumple los requerimientos de clientes. Esta es la metodología GeneXus, se base en la construcción de un sistema mediante aproximaciones sucesivas.

Se pone en producción de una aplicación para una plataforma elegida por el cliente que se realiza solamente cuando el prototipo ha sido aprobado y tener una imagen fiel de la aplicación que se lleva al cliente en la plataforma de producción. Este ciclo es llamado el ciclo de Diseño-Producción. Se aplica el mismo procedimiento que el de Diseño-Prototipación.

Según ARTech Consultores S.R.L. (2004:4) *Getting Started with GeneXus 8.0*, El ambiente de desarrollo de GeneXus 8.0 Versión de Evaluación 2.4. Traducción al español: “Es un Ambiente de Desarrollo Integrado parar el diseño y la prototipación de sus aplicaciones, sin tener en cuenta la plataforma de la producción”.² En razón de esta limitación, no se puede realizar una demostración sobre este ciclo de Diseño-Producción.

² Original en inglés: “It is an Integrated Development Environment (IDE) for designing and prototyping your applications, regardless of the production platform”.

En resumen, después de haber realizar la demostración de está herramienta se ha mostrado que:

1. Es fácil de manejar la herramienta CASE “GeneXus”.
2. La utilización de la herramienta CASE permite convertir el proceso de creación de un software de aplicación en una tarea simple.
3. El uso de la herramienta CASE en el desarrollo de un sistema de información o de aplicaciones, cambió el paradigma y el comportamiento de la gente implicada.
4. El uso del prototipo funcional hace que la prueba y corregir error de aplicación pueda realizar de manera inmediata.
5. La productividad y la generación de código fuente en tiempo récord son los beneficios claves que conllevan a la disminución del tiempo durante el desarrollo y el mantenimiento de un programa de aplicación.

Durante la creación de varios prototipos funcionales que conforman el programa de aplicación utilizando la herramienta GeneXus, se trabajó en una computadora portátil bajo los siguientes requerimientos:

Cuadro 28 Los requerimientos de la computadora portátil utilizada durante la creación de varios prototipos funcionales que conforman el programa de aplicación.

Requerimiento de Hardware	Marca	ASUSTeK Computer Inc.,
	Modelo	ASUS T9400 Series Notebook PC.
	Microprocesador	Mobile Intel Pentium III 900 MHz
	Tarjeta madre	ASUSTeK T9000 82815EP/EP/E
	Memoria RAM	128 MB.
	Tarjeta de video	Intel 82815 Graphics Controller, 36MB.
	Monitor LCD	Resolución 1024x768, 16M Truecolours (24bit)
	Disco Duro	IBM Travelstar™ IDE 20 GB.
	Unidad de CD	TOSHIBA DVD-ROM. SD-C2502
Requerimiento de Software	Sistema Operativo	Microsoft Windows XP Professional Service Pack 1
	Navegador Internet	Microsoft Internet Explorer 6.0 Service Pack 2

Desarrollo de la propuesta

Objetivos de la propuesta

Objetivo General

Definir los requerimientos a seguir para la utilización de la herramienta CASE “GeneXus” como software de enseñanza/aprendizaje dentro de la carrera Informática Gerencial.

Objetivos Específicos

1. Mencionar las asignaturas que pudieran adaptarse con la utilización de esta herramienta CASE “GeneXus” dentro del contenido programático de la carrera Informática Gerencial.
2. Definir los lineamientos a seguir en la adquisición y utilización de la herramienta CASE “GeneXus” dentro del proceso de enseñanza/aprendizaje de los alumnos de Informática Gerencial.

Justificación de la propuesta

La propuesta de usar la herramienta CASE “GeneXus” surge con el propósito de mejorar la formación de los alumnos de Administración mención Informática Gerencial en el área relacionado con el proyecto de desarrollo de sistemas de información. El propósito no es sólo adquirir los conocimientos teóricos en dicho área sino también tener experiencias en la práctica de ejecución del proyecto de tecnología de información. Es por ello se recomienda incorporarla dentro del contenido programático de dicha carrera.

Presentación de la propuesta

La incorporación de la herramienta GeneXus dentro del contenido programático

Antes de incorporar la herramienta CASE “GeneXus” se deben cumplir 2 requisitos previos en la enseñanza a los alumnos de Informática Gerencial: tener los conocimientos de programación y de base de datos relacional.

Al cumplir con los requisitos antes mencionados, se propone el uso de la herramienta CASE “GeneXus” como aplicación de prácticas de laboratorio en las asignaturas que pudiera hacer uso de esta herramienta. Esto dependerá siempre del criterio del profesor, ya que el mismo es autónomo en su materia. Las asignaturas que pudieran utilizar tal herramienta son:

- a. Análisis y Diseño de Sistema de Información o Sistema de Información.
- b. Base de Datos.
- c. Programación.
- d. Tendencias de la informática.

También puede servir como herramienta de desarrollo a los alumnos de Informática Gerencial para realizar trabajos de investigación.

En cuanto a un programa de estudios para el manejo de la herramienta CASE “GeneXus”, se propone el siguiente contenido:

Unidad I Aspecto general sobre las herramientas CASE

Tema 1. Repaso de los conceptos básicos de Base de Datos.

Definición de Base de Datos y DBMS. Modelos de datos: conceptual, lógico y físico. Conceptos de tabla, índice, clave primaria, clave candidata y clave foránea. Dependencia funcional. Normalización.

Se propone realizar un breve repaso sobre los conceptos básicos de la base de datos con la finalidad de que los alumnos tengan una buena base al comienzo del estudio de la herramienta CASE “GeneXus”.

Tema 2. Conceptos básicos de herramientas CASE.

Definición de las herramientas CASE. Historia y su evolución. Clasificación de las herramientas CASE. Beneficios y debilidades de herramientas CASE. Ejemplos de las herramientas CASE en el mercado.

Se trata de una introducción de conocimientos relacionados con las herramientas CASE como la definición, la historia... entre otros conceptos básicos.

Tema 3. Introducción a GeneXus.

Introducción. Visión general. Filosofía. Historia. Principales características. Tecnologías soportadas. Componentes de la herramienta GeneXus. Base de Conocimiento. Modelo de Diseño, Prototipo y Producción. Objetos de GeneXus.

Se pretende dar una breve presentación a la herramienta CASE “GeneXus” antes de usarla.

Tema 4. Metodología de desarrollo de GeneXus

Modelo evolutivo. Concepto de Prototipo y Prototipo funcional. Proceso del desarrollo de una aplicación con GeneXus. Ciclo de Diseño-Prototipación. Ciclo de Diseño-Producción. Comparación de la metodología tradicional con la metodología

GeneXus.

En este tema, se estudian las teorías de la metodología de desarrollo GeneXus, su ciclo de desarrollo... y otros conceptos importantes que van a servir como base el aprendizaje de los temas restantes.

Unidad II Diseño de aplicaciones con GeneXus

Tema 5. La creación del modelo de diseño.

Modelado de la realidad a partir de las visiones de los usuarios. Ingeniería Inversa. Base de Conocimiento. Tipos de modelos. Creación la estructura de un objeto de transacción. Tipo del diálogo para la transacción. La interfaz Form y Web Form. Relaciones entre la estructura y Forms. El uso de Domino. Atributos Fórmula: Tipo de fórmula. Función de las reglas de transacciones. Índice: Tipo de índices. Comunicación entre los objetos. Casos prácticos.

A partir de este tema, se comienza a aprender a crear un sistema de información utilizando la herramienta CASE “GeneXus”. Desde la creación de Base de conocimiento hasta la inserción de las reglas.

Tema 6. La creación del modelo de prototipo y de producción.

La selección de la plataforma o el ambiente del trabajo. Análisis de Impacto (o de creación de base de datos). La reorganización: cambio físico de la base de datos. La especificación y generación de un programa de aplicación. La ejecución y prueba de un programa de aplicación. Generación de aplicaciones. Casos prácticos.

En esta etapa, se enseña a los alumnos a ejecutar y probar el sistema definido en el modelo de Diseño para luego ponerlo en producción de código fuente de un lenguaje de programación seleccionado como C[#] .NET, Java, Visual Basic, Visual FoxPro... entre otros.

La definición de los pasos a seguir en la adquisición y utilización de la herramienta CASE “GeneXus” en la UCAT.

Para definir los lineamientos a seguir en la adquisición y utilización de la herramienta CASE “GeneXus” en la UCAT. Se propone realizar las siguientes actividades:

1. Preparar un plan previamente estudiado por parte de los profesores en el área de Informática sobre la propuesta de incorporación de herramienta CASE “GeneXus” ante la Escuela de Administración y Contaduría Pública.
2. Hacer la solicitud por parte de la Escuela Administración y Contaduría Pública ante el Vice-Rectorado Administrativo para la adquisición de dicha herramienta.
3. Adquirir la licencia de herramienta CASE “GeneXus” por parte de Vice-Rectorado Administrativo de la UCAT.
4. Realizar cursos de capacitación a los profesores en el área de Informática que dictan materias relacionadas con el análisis y diseño de sistema de información.
5. Capacitar al personal especializado en la instalación y mantenimiento de la herramienta CASE “GeneXus”.

CONCLUSIONES Y RECOMENDACIONES

1. La herramienta CASE es el mejor soporte y tiene gran importancia para el proceso del desarrollo de sistemas o software de aplicación. Por lo tanto, su uso ha mejorado la calidad y la productividad del desarrollo de un sistema de información.
2. Se determinó que existe carencia de conocimiento en el área de las herramientas de Ingeniería de Software Asistida por Computadora (CASE) por parte de los estudiantes de la carrera Informática Gerencial.
3. El 95.5% de alumnos de tercero y cuarto año de Informática Gerencial y el 90 % de los profesores en el área de Informática de la Escuela de Administración y Contaduría Pública de la UCAT opinaron que están de acuerdo de que la UCAT incorpore el manejo de la herramienta CASE dentro del contenido programático de la carrera Informática Gerencial.
4. Las computadoras existentes en los laboratorios de computación de la Universidad Católica del Táchira cumplen con los requerimientos mínimos para la instalación de la herramienta GeneXus.
5. Igualmente, se recomienda que el uso de las herramientas CASE como software de aprendizaje no sea exclusivo para los alumnos de Informática Gerencial sino también que el mismo sea aplicable en la formación de los alumnos de Educación mención Informática de la Universidad Católica del Táchira.

REFERENCIAS BIBLIOGRÁFICAS

Libros:

- Barrios Albornoz, Judith (2001), *Estudio de estructuras, componentes, interrelaciones, metodologías y tecnologías asociadas a los sistemas de información*. Mérida: Universidad de Los Andes (ULA)
- Ezquiel Rozic, Sergio (2004), *Bases de Datos y su aplicación con SQL*. Buenos Aires: MP Ediciones S.A.
- Hernández Sampieri, Roberto, Collado Fernández, Carlos y Lucio Baptista, Pilar (1992), *Metodología de la Investigación*. (2^a. ed.) México: McGraw-Hill/Interamericana Editores, S.A. DE C.V.
- Hernández Sampieri, Roberto, Collado Fernández, Carlos y Lucio Baptista, Pilar (2003), *Metodología de la Investigación*. (3^a. ed.) México: McGraw-Hill/Interamericana Editores, S.A. DE C.V.
- Kendall, Kenneth E. y Kendall Julie E. (1997), *Análisis y Diseño de Sistemas*. (3^a. ed.) México: Prentice Hall Hispanoamericana, S.A.
- Korth, Henry F. y Siferschatz, Abraham. (1988), *Fundamentos de Bases de Datos* (1^a ed.) México: McGRAW-HILL/INTERAMERICANA DE MÉXICO S.A. DE C.V.
- O'Brien, James A. (2001), *Sistema de Información Gerencial* (4^a. ed.) Bogotá: McGRAW-HILL INTERAMERICANA, S.A.
- Sabino, Carlos A. (2000), *El Proceso de Investigación*. Caracas: Editorial Panapo.
- Senn, James A. (1992), *Análisis y Diseño de Sistema de Información* (2^a. ed.) México: McGRAW-HILL/INTERAMERICANA DE MÉXICO S.A. DE C.V.
- Universidad Pedagógica Experimental Libertador (UPEL) (1998, Julio), *Manual de Trabajos de Grado de Especialización y Maestría y Tesis Doctorales*. Caracas: Autor.
- Webster, Allen L. (2000), *Estadística aplicada a los negocios y la economía*. (3^a. ed.) Santa Fe de Bogotá: McGRAW-HILL INTERAMERICANA S.A.

Witten, Jeffrey L., Bentley, Lonnie D. y Barlow, Victor M. (1996), *Análisis y Diseño de Sistemas de Información*. (3ª. ed.) Colombia: McGraw-Hill/IRWIN.

Fuentes electrónicas en línea:

ARTech Consultores S.R.L (2002-2003), *GeneXus: Filosofía*. [Libro en línea]. Disponible: <http://www.gxtechnical.com/main/hdcenter.aspx?2,5,36,+++++++877> [Consulta: 04/02/2005 11:49am]

ARTech Consultores S.R.L. (2004, Agosto), *GeneXus Visión General*. [Libro en línea]. Disponible: <http://www.gxtechnical.com/main/hdcenter.aspx?2,5,36,688> [Consulta: 04/02/2005 11:55am]

ARTech Consultores S.R.L. (2004, Junio), *Folleto GeneXus* [Libro en línea]. Disponible: <http://www.gxtechnical.com/main/hdcenter.aspx?2,5,36,++++++1060> [Consulta: 23/03/2005 05:13pm]

ARTech Consultores S.R.L. (2004, Octubre), *Getting Started with GeneXus 8.0* [Libro en línea]. Disponible: <http://www.gxtechnical.com/main/hdcenter.aspx?2,5,36,1051> [Consulta: 04/02/2005 11:38am]

ARTech Consultores S.R.L. (2004, Diciembre), *Curso GeneXus® Versión 8.0* [Libro en línea]. Disponible: <http://www.gxtechnical.com/main/hdcenter.aspx?2,5,36,1506> [Consulta: 04/02/2005 11:35am]

ARTech Consultores S.R.L. (2005), *Principales características*. [Página Web en línea]. Disponible: <http://www.genexus.com/portal/hgxpp001.aspx?2,3,50,O,S,0,2MNU;E;9;4;MNU,> [Consulta: 10/03/2005 10:39pm]

Centro de Estudios y Diseño de Sistemas (CEDDS) (2005), *Objetivos y Recuperación de inversión en las Herramientas CASE*. [Página Web en línea]. Disponible: <http://cedds.nauta.es/informes/case03.htm> [Consulta: 25/04/2005 04:52pm]

Centro de Estudios y Diseño de Sistemas (CEDDS) (2005), *Herramientas CASE*. [Página Web en línea]. Disponible: <http://cedds.nauta.es/informes/case04.htm> [Consulta: 25/04/2005 04:58pm]

Instituto Nacional de Estadística e Informática (INEI) (1999, Noviembre), *Herramientas CASE*. [Libro en línea]. Colección Cultura Informática. Disponible: http://www.pcm.gob.pe/portal_ongei/publicaciones/cultura/lib5103/Libro.pdf [Consulta: 08/12/2004 03:25pm]

Mantenimiento del Software. [Documento en línea]. (2003), Disponible: <http://alarcos.inf-cr.uclm.es/doc/ISOFTWAREI/Tema16.pdf> [Consulta: 04/03/2005 4:48pm]

María Isabel y Alfonso Galipienso (2004), *Mantenimiento y Reingeniería* [Documento en línea] Disponible: <http://www.dccia.ua.es/dccia/inf/asignaturas/IS2/plan93/tema7.pdf> [Consulta: 06/03/2005 3:16pm]

Universidad Católica del Táchira (2004), *Orígenes de la Universidad Católica del Táchira* [Página Web en línea]. Disponible: <http://www.ucat.edu.ve/ucat/historia.htm> [Consulta: 24/12/2004 3:48pm]

Universidad Católica del Táchira (2004), *Licenciatura en Administración mención Informática Gerencial* [Página Web en línea]. Disponible: http://www.ucat.edu.ve/carreras/informatica_gere.htm [Consulta: 24/12/2004 2:50pm]

Universidad Nacional de Colombia (2004), *Herramientas CASE* [Página Web en línea] Consulta: <http://www.virtual.unal.edu.co/cursos/sedes/manizales/4100010/Lecciones/Cap5/HtasCASE.htm> [Consulta: 07/03/2005 3:53pm]

Wikipedia (2005), *The Free Encyclopedia - Legacy System* [Página Web en línea]. Disponible: http://en.wikipedia.org/wiki/Legacy_system [Consulta: 07/03/2005 3:45pm]

Fuentes de tipo Legal:

Ley Orgánica de Educación. (1980), Gaceta Oficial Extraordinario de la República Bolivariana de Venezuela, No. 2635. Julio 9, 1980. [Página Web en línea] Disponible: http://www.analitica.com/bitblioteca/congreso_venezuela/ley_educacion.asp Consulta: [04/05/2005 10:35am]

Ley sobre Derecho de Autor. (1993), Gaceta Oficial Extraordinario de la República Bolivariana de Venezuela, No. 4638. Agosto 14, 1993. [Página Web en línea] Disponible: http://www.analitica.com/bitblioteca/congreso_venezuela/derecho_de_autor.asp [Consulta: 04/05/2005 10:25am]

Estatuto Orgánico de la Universidad Católica del Táchira

Revistas especializadas:

Michael Floyd (2001, Octubre), Dentro de Visual Studio .NET. *PC Magazine en Español*, 12(10), 20-23.

Ricardo Mínguez (2002, junio), Microsoft .NET: la nueva generación. *Programadores en acción*, 2(8), 100-106.

Programa de Computación y Multimedia:

Biblioteca de Consulta Microsoft® Encarta® 2005 [Multimedia en Disco Compacto]. (2004). Disponible: Microsoft Corporation.

GeneXus® 8.0 Versión de Evaluación 2.4 [Programa en Disco Duro]. (2004) Disponible: ARTech Consultores S.R.L.

Microsoft® Excel XP Service Pack 1 (SP1) [Programa en Disco Duro]. (2001). Disponible: Microsoft Corporation.

Microsoft® Office Visio® Professional 2003 SP1 [Programa en Disco Duro]. (2003). Disponible: Microsoft Corporation.

SPSS (Statistical Program for Social Sciences) 13.0 for Windows [Programa en Disco Duro]. (2004) Disponible: SPSS Inc.,

Páginas Web en línea:

Diccionario informático: <http://www.glosarium.com/list/14/>

Microsoft Corporation: <http://www.microsoft.com/>

Wikipedia La Enciclopedia Libre: <http://es.wikipedia.org>

ANEXOS

Anexo A

Información acerca de GeneXus 8.0 Versión de Evaluación 2.4

GeneXus 8.0 Versión de Evaluación 2.4

Estas son los productos que integran en GeneXus 8.0 Versión de Evaluación 2.4:

1. El ambiente de desarrollo GeneXus.
2. Tres (3) Generadores de Código.
 - a. Generador .NET,
 - b. Generador JAVA
 - c. Generador Visual Basic.

La instalación de GeneXus 8.0 Versión de Evaluación 2.4

Los elementos que son necesarios para poder usar la herramienta GeneXus dependen del ambiente de trabajo y el lenguaje de programación escogido. Por cada generador de código escogido es posible que tenga instalar algunos elementos relacionados con dicho lenguaje. En el presente trabajo de grado, se escoge la arquitectura Cliente/Servidor como ambiente de trabajo y el lenguaje de programación es Microsoft .NET.

Paso 1: Preparación antes del inicio de instalación

- a. Actualizar el navegador Microsoft Internet Explorer a la versión 6.0 con Service Pack 1.
- b. Descargar el paquete redistribuible de Microsoft® .NET Framework versión 1.1 para poder utilizar Generador .NET accediendo a <http://www.microsoft.com/downloads/details.aspx?FamilyID=262d25e3-f589-4842-8157-034d1e7cf3a3&DisplayLang=es>

- c. Descargar el paquete redistribuible de Microsoft Visual J# .NET versión 1.1
<http://www.microsoft.com/downloads/details.aspx?familyid=E3CF70A9-84CA-4FEA-9E7D-7D674D2C7CA1&displaylang=es>

- d. Descargar Microsoft SQL Server 2000 Desktop Engine
<http://www.microsoft.com/downloads/details.aspx?FamilyID=413744d1-a0bc-479f-bafa-e4b278eb9147&DisplayLang=es>

- e. Acceder a la página Web de Versión de evaluación de GeneXus
<http://www.genexus.com/portal/hgxpp001.aspx?2,3,55,O,E,0,MNU;E;41;2;MNU;> para poder registrar a la comunidad GeneXus y descargar la herramienta GeneXus.

Paso 2: Instalación de las aplicaciones

- a. Instalar el paquete redistribuible de Microsoft .NET Framework 1.1, y de Microsoft Visual J# 1.1; y Microsoft SQL Server 2000.

- b. Instalar GeneXus 8.0 versión de evaluación.

Paso 3: Solicitud de la clave de autorización

Después de la instalación de GeneXus, ejecute GeneXus y copie el Código del sitio (*Site Code*) y luego se debe dirigirse a esta dirección: <http://www.genexus.com/trial/authorize> para obtener la autorización de la llave del sitio (*Site Key*) de GeneXus 8.0 Versión de Evaluación.

Limitación de GeneXus 8.0 Versión de Evaluación

La licencia de la herramienta GeneXus Versión de Evaluación es ilimitada (sin fecha de expiración) y todos los generadores instalados permanecen autorizados mediante una única llave de sitio (*Site Key*).

Se aplican algunas restricciones respecto al número máximo de objetos GeneXus que se pueden crear para una Base de Conocimiento dada:

- Transacciones: 30
- Work Panels (incluyendo GeneXus selection prompts): 50
- Web Panels (incluyendo GeneXus selection prompts): 50
- Procedimientos: 20
- Reportes: 20

Las bases de conocimiento generadas con la GeneXus Versión de Evaluación no pueden abrirse con una versión estándar de GeneXus y viceversa. La instalación de la GeneXus Versión de Evaluación es local y para un solo usuario, por lo tanto no está habilitada la instalación en red.

No incluye la opción de la "Distribución" en el menú *Knowledge Manager*. Los archivos con extensión xpz creados con la GeneXus Versiones de Evaluación pueden consolidarse (siempre que no excedan los límites mencionados). Es decir que se pueden consolidar proyectos pero no distribuirlos. El apoyo para GeneXus Versión de Evaluación es limitada.

Anexo B

Instrumento de recolección de datos para los profesores

UNIVERSIDAD CATÓLICA DEL TÁCHIRA
FACULTAD DE CIENCIAS ECONÓMICAS Y SOCIALES
ESCUELA DE ADMINISTRACIÓN Y CONTADURÍA PÚBLICA
COORDINACIÓN DE FORMACIÓN COMPLEMENTARIA

Estimado:

Profesor.-

El presente cuestionario tiene por finalidad recoger información para desarrollar la investigación titulada **“Propuesta para la incorporación de la herramienta CASE “GeneXus” en la formación de los alumnos de Administración mención Informática Gerencial de la Universidad Católica del Táchira”**.; por lo cual se pide que la colaboración en responder con objetividad y honestidad ya que esto depende el éxito de la investigación.

Instrucciones:

Responda clara y brevemente cada una de las preguntas que se le presenten :

- No deje preguntas sin responder.
- Marque con una “X” la casilla de “SI” cuando considere que su respuesta es afirmativa.
- Marque con una “X” la casilla de “NO” cuando considere que su respuesta es negativa.
- Marque con una “X” la casilla de su selección en las preguntas de selección múltiple.

Gracias por su colaboración.

En caso que esté de acuerdo con la pregunta 4, conteste las siguientes preguntas.

5. Las herramientas CASE proveen muchos beneficios en todas las etapas de desarrollo de sistema de información, indique usted ¿En qué actividad(es) le gustaría hacer más énfasis con la utilización de esta herramienta CASE en la formación de los alumnos de Informática Gerencial?

_____ Realizar práctica que abarque todo el ciclo de vida del desarrollo de sistemas de información.

_____ Automatizar el dibujo de diagramas como el diagrama de flujo de dato (DFD) y entre otros.

_____ Ayudar en la creación de relaciones de la base de datos y la generación de scripts.

_____ Ayudar en la documentación del sistema.

_____ Generar código fuente automáticamente sin la necesidad de programar.

_____ Todas las anteriores.

_____ Ninguno.

Otros _____

6. ¿En qué asignatura(s) considera usted que podría(n) utilizar la herramienta CASE como software de aprendizaje?

7. ¿Considera usted que la Universidad Católica del Táchira debe capacitar al personal docente en el manejo o utilización de herramientas CASE?

Sí _____ No _____

Anexo C

Instrumento de recolección de datos para los alumnos

Instrucciones:

Responda clara y brevemente cada una de las preguntas que se le presenten:

- No deje preguntas sin responder.
- Marque con una “X” la casilla de “SI” cuando considere que su respuesta es afirmativa.
- Marque con una “X” la casilla de “NO” cuando considere que su respuesta es negativa.
- Marque con una “X” la casilla de su selección en las preguntas de selección múltiple.

1. ¿Conoce usted las herramientas de Ingeniería de Software Asistida por Computadora (CASE)? Sí _____ No _____
2. ¿Cuáles son las siguientes herramientas CASE conoce o ha utilizado algunas vez?

_____ GeneXus	_____ Oracle Designer
_____ Sybase PowerDesigner	_____ System Architect
_____ SILVERRUN	_____ Win A&D
_____ Microsoft Office Visio	_____ Ninguno
Otro _____	

3. ¿Le gustaría aprender a utilizar una herramienta de desarrollo de sistema de información mediante el diseño de la estructura de la base de datos sin necesidad de programar o codificar? Sí _____ No _____
4. ¿Le gustaría que la Universidad Católica del Táchira incorpore el manejo de la herramienta CASE dentro del contenido programático de la carrera Informática Gerencial? Sí _____ No _____

Anexo D

La validación de los cuestionarios

HOJA DE REGISTRO PARA LA VALIDACIÓN DEL INSTRUMENTO

ÍTEM	CRITERIOS					OBSERVACIONES
	P	A	C	R	E	
1. ¿Conoce usted las herramientas de Ingeniería de Software Asistida por Computadora (CASE)? (Sí/No)						
2. ¿De las siguientes herramientas CASE cuáles conoce o ha utilizado alguna vez? (GenXus, Oracle Designer, Sybase PowerDesigner, System Architect, Silverrun, Win A&D, Microsoft Office Visio, Ninguno y Otro.)						
3. ¿Cuáles de las metodologías de desarrollo de sistema de información conoce o domina usted? (Modelo tradicional, de análisis estructurado, de prototipos, evolutivo, en espiral y otras.)						
4. ¿Esta usted de acuerdo de que la Universidad Católica del Táchira incorpore la herramientas CASE dentro del contenido programático de la carrera Informática Gerencial? (De acuerdo, Ni de acuerdo ni en desacuerdo y En desacuerdo.)						
5. Las herramientas CASE proveen muchos beneficios en todas las etapas de desarrollo de sistema de información, indique usted ¿En qué actividad(es) le gustaría hacer más énfasis con la utilización de esta herramienta CASE en la formación de los alumnos de Informática Gerencial?						
6. ¿En qué asignatura(s) considera usted que podría(n) utilizar la herramienta CASE como software de aprendizaje?						

P = Pertinente, C = Claridad, E = Eliminar, A = Ambiguo, R = Reformular

7. ¿Considera usted que la Universidad Católica del Táchira debe capacitar al personal docente en el manejo o utilización de herramientas CASE?						
---	--	--	--	--	--	--

P = Pertinente, C = Claridad, E = Eliminar, A = Ambiguo, R = Reformular

Sugerencia: _____

Nombre del Validador:	Especialidad:	Firma:	Fecha:

HOJA DE REGISTRO PARA LA VALIDACIÓN DEL INSTRUMENTO

ÍTEM	CRITERIOS					OBSERVACIONES
	P	A	C	R	E	
1. ¿Conoce usted las herramientas de Ingeniería de Software Asistida por Computadora (CASE)? (Sí/No)						
2. ¿De las siguientes herramientas CASE cuáles conoce o ha utilizado alguna vez? (GenXus, Oracle Designer, Sybase PowerDesigner, System Architect, Silverrun, Win A&D, Microsoft Office Visio, Ninguno y Otro.)						
3. ¿Le gustaría aprender a utilizar una herramienta de desarrollo de sistema de información mediante el diseño de la estructura de la base de datos sin necesidad de programar o codificar?						
4. ¿Le gustaría que la Universidad Católica del Táchira incorpore el manejo de la herramienta CASE dentro del contenido programático de la carrera Informática Gerencial?						

P = Pertinente, C = Claridad, E = Eliminar, A = Ambiguo, R = Reformular

Sugerencia: _____

Nombre del Validador:	Especialidad:	Firma:	Fecha: