



**UNIVERSIDAD ESTATAL DE MILAGRO  
FACULTAD CIENCIAS DE LA INGENIERIA**

**TRABAJO DE TITULACIÓN DE GRADO PREVIO A LA  
OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS  
COMPUTACIONALES  
PROPUESTA PRÁCTICA DEL EXAMEN DE GRADO O DE FIN  
DE CARRERA (DE CARÁCTER COMPLEXIVO)  
INVESTIGACIÓN DOCUMENTAL**

**TEMA:** ANÁLISIS DE LA IMPORTANCIA DE LAS HERRAMIENTAS CASE EN  
EL DESARROLLO DE SOFTWARE

**Autor:**

**RICHARD IVAN HUARACA BUÑAY**

**Tutor:**

**MSC. DANIEL ALEXANDER VERA PAREDES**

**MILAGRO, NOVIEMBRE DEL 2018**

**ECUADOR**

## DERECHOS DE AUTOR

### DERECHOS DE AUTOR

Ingeniero.

Fabrizio Guevara Viejó, PhD.

**RECTOR**

**Universidad Estatal de Milagro**

Presente.

Yo, **RICHARD IVÁN HUARACA BUÑAY** en calidad de autor y titular de los derechos morales y patrimoniales de la alternativa de Titulación – Examen Complexivo: Investigación Documental, modalidad presencial, mediante el presente documento, libre y voluntariamente procedo a hacer entrega de la Cesión de Derecho del Autor de la propuesta práctica realizado como requisito previo para la obtención de mi Título de Grado, como aporte a la Línea de Investigación **PROCESAMIENTO Y ANALISIS DE DATOS, DESARROLLO DE SOFTWARE** del grupo de investigación **TECNOLOGIAS DE INFORMACIÓN, CONTROL Y AUTOMATIZACIÓN DE PROCESOS** de conformidad con el Art. 114 del Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación, concedo a favor de la Universidad Estatal de Milagro una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos. Conservo a mi favor todos los derechos de autor sobre la obra, establecidos en la normativa citada.

Así mismo, autorizo a la Universidad Estatal de Milagro para que realice la digitalización y publicación de este Proyecto de Investigación en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

El autor declara que la obra objeto de la presente autorización es original en su forma de expresión y no infringe el derecho de autor de terceros, asumiendo la responsabilidad por cualquier reclamación que pudiera presentarse por esta causa y liberando a la Universidad de toda responsabilidad.

Milagro, a los 15 días del mes de enero de 2019



Richard Iván Huaraca Buñay

CI: 0605939222

# APROBACIÓN DEL TUTOR DE LA INVESTIGACION DOCUMENTAL

## APROBACIÓN DEL TUTOR DE LA INVESTIGACION DOCUMENTAL

Yo, **DANIEL ALEXANDER VERA PAREDES** en mi calidad de tutor del Proyecto de Investigación, elaborado por el estudiante **RICHARD IVÁN HUARACA BUÑAY** cuyo tema de trabajo de Titulación es **ANÁLISIS DE LA IMPORTANCIA DE LAS HERRAMIENTAS CASE EN EL DESARROLLO DE SOFTWARE** que aporta a la Línea de Investigación **PROCESAMIENTO Y ANALISIS DE DATOS, DESARROLLO DE SOFTWARE** previo a la obtención del Grado de **INGENIERO EN SISTEMAS COMPUTACIONALES**; trabajo de titulación que consiste en una propuesta innovadora que contiene, como mínimo, una investigación exploratoria y diagnóstica, base conceptual, conclusiones y fuentes de consulta, considero que el mismo reúne los requisitos y méritos necesarios para ser sometido a la evaluación por parte del tribunal calificador que se designe, por lo que lo **APRUEBO**, a fin de que el trabajo sea habilitado para continuar con el proceso de titulación de la alternativa de Emprendimiento de la Universidad Estatal de Milagro.

En la ciudad de Milagro, a los 15 días del mes de enero de 2019.



**DANIEL ALEXANDER VERA PAREDES**

Tutor

C.I.: 0914192182

## APROBACIÓN DEL TRIBUNAL CALIFICADOR

### APROBACIÓN DEL TRIBUNAL CALIFICADOR

El tribunal calificador constituido por:

**MSC. DANIEL ALEXANDER VERA PAREDES**

**ING. BERMEO PAUCAR JAVIER RICARDO**

**ING. LOPEZ BERMUDEZ RICAUTER MOISES**

Luego de realizar la revisión del Proyecto de Investigación, previo a la obtención del título (o grado académico) de INGENIERO EN SISTEMAS COMPUTACIONALES presentado por el señor RICHARD IVÁN HUARACA BUÑAY.

Con el tema de trabajo de Titulación: ANÁLISIS DE LA IMPORTANCIA DE LAS HERRAMIENTAS CASE EN EL DESARROLLO DE SOFTWARE




Otorga al presente Proyecto de Investigación, las siguientes calificaciones:

Estructura	[ 75 ]
Defensa oral	[ 18,33 ]
Total	[ 93,33 ]

Emite el siguiente veredicto: Aprobado.

Fecha: 15 de enero de 2019.

Para constancia de lo actuado firman:

	Nombres y Apellidos	Firma
Presidente	MSC. DANIEL ALEXANDER VERA PAREDES	
Secretario	ING. BERMEO PAUCAR JAVIER RICARDO	
Integrante	ING. LOPEZ BERMUDEZ RICAUTER MOISES	

## **DEDICATORIA**

Dedico este proyecto a Dios por ser el ser supremo que siempre nos ha bendecido con salud y vida para poder encaminarnos en nuestras metas, a mis padres que me han brindado su apoyo y su ejemplo de superación para seguir adelante y a nuestros queridos docentes que nos instruyeron para ser excelentes profesionales.

Richard Huaraca

## **AGRADECIMIENTO**

Agradezco infinitamente a Dios, por siempre estar guiando mi camino y por darme unos padres ejemplares dignos de admiración, a la Universidad Estatal de Milagro que me ha permitido conocer docentes que han sabido impartir sus conocimientos de una manera equitativa preparándome de una excelente manera para desenvolverme bien en el ámbito laboral y a mi tutor Msc. Daniel Vera que con paciencia y empeño supo brindarme sus conocimientos para concluir mi proyecto.

Richard Huaraca

## ÍNDICE GENERAL

DERECHOS DE AUTOR .....	i
APROBACIÓN DEL TUTOR DE LA INVESTIGACION DOCUMENTAL .....	i
APROBACIÓN DEL TRIBUNAL CALIFICADOR.....	iii
DEDICATORIA .....	iv
AGRADECIMIENTO .....	v
RESUMEN:.....	1
ABSTRACT .....	2
INTRODUCCION.....	3
CAPITULO I.....	4
PROBLEMA DE INVESTIGACION .....	4
OBJETIVO GENERAL .....	5
OBJETIVO ESPECIFICO.....	5
JUSTIFICACION.....	5
CAPITULO II.....	6
MARCO TEORICO CONCEPTUAL.....	6
DEFINICION DE HERRAMIENTAS CASE .....	6
EL PROCESO DE DESARROLLO DE SOFTWARE.....	6
CLASIFICACION DE LAS HERRAMIENTAS CASE .....	6
CASE EN EL CICLO DE VIDA DE UN SISTEMA .....	10
ESTRUCTURA DE LAS HERRAMIENTAS CASE .....	10
CAPITULO III.....	15
METODOLOGIA.....	15
CAPITULO IV .....	16
DESARROLLO DEL TEMA.....	16
DESARROLLO TRADICIONAL .....	18
DESARROLLO DE APLICACIONES.....	18

REQUISITOS PARA ADQUIRIR HERRAMIENTAS CASE .....	19
REQUISITOS FUNCIONALES .....	19
FACILIDADES DE USO DE UNA HERRAMIENTA CASE .....	20
CONCLUSION .....	22
REFERENCIAS BIBLIOGRÁFICAS .....	23

## **INDICE DE FIGURAS**

Figura 1 Clasificación general de las Herramientas CASE .....	8
Figura. 2 Clasificación de las Herramientas CASE .....	9
Figura. 3 Tipos de Herramientas CASE y en qué Parte del Ciclo de Vida Actúan.....	11
Figura. 4 Diagrama de un Repositorio.....	14
Figura. 5 Ciclo de Vida: Diseño – Prototipo .....	16
Figura.6 Metodología Incremental.....	21



## **Título la Investigación Documental: ANÁLISIS DE LA IMPORTANCIA DE LAS HERRAMIENTAS CASE EN EL DESARROLLO DE SOFTWARE.**

### **RESUMEN:**

La evolución de las Herramientas CASE en sus inicios ayuda en la documentación por computadora, diagramación, herramientas de análisis y diseño. A mediados de los 80's: Se presenta un diseño automático de análisis, pruebas y repositorios automáticos de información de sistemas. Posteriormente se logró la generación automática de código desde especificaciones de diseño.

Se puede definir a las Herramientas CASE como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del Ciclo de Vida de desarrollo de un Software. Como es sabido, los estados en el Ciclo de Vida de desarrollo de un Software son: Investigación Preliminar, Análisis, Diseño, Implementación e Instalación.

Ahora bien, con la aparición de las redes de ordenadores en empresas y universidades ha surgido en el mundo de la informática la tecnología cliente / servidor. Son muchas de las organizaciones que ya cuentan con un número considerable de aplicaciones cliente / servidor en operación: Servidores de Bases de Datos y Manejadores de Objetos Distribuidos. Cliente / servidor es una tecnología de bajo costo que proporciona recursos compartidos, escalabilidad, integridad, encapsulamiento de servicios, etc. Pero al igual que toda tecnología, el desarrollo de aplicaciones cliente / servidor requiere que la persona tenga conocimientos, experiencia y habilidades en procesamiento de transacciones, diseño de base de datos, redes de ordenadores y diseño gráfica de interface.

**PALABRAS CLAVES: HERRAMIENTA CASE-SISTEMA-SOFTWARE-DESARROLLO**

**Título la Investigación Documental: ANÁLISIS DE LA IMPORTANCIA DE LAS HERRAMIENTAS CASE EN EL DESARROLLO DE SOFTWARE.**

**ABSTRACT**

The evolution of CASE Tools in the beginning helps in computer documentation, diagramming, analysis and design tools. In the mid 80's: An automatic design of analysis, testing and automatic repositories of system information is presented. Subsequently, the automatic generation of code was achieved from design specifications.

The CASE Tools can be defined as a set of programs and aids that provide assistance to analysts, software engineers and developers, during all the steps of the Life Cycle of Software Development. As it is known, the states in the Life Cycle of Software development are: Preliminary Investigation, Analysis, Design, Implementation and Installation.

However, with the emergence of computer networks in companies and universities, the client / server technology has emerged in the computer world. There are many organizations that already have a considerable number of client / server applications in operation: Database Servers and Distributed Object Managers. Client / server is a low cost technology that provides shared resources, scalability, integrity, encapsulation of services, etc. But like all technology, the development of client / server applications requires that the person have knowledge, experience and skills in transaction processing, database design, computer networks and graphic interface design.

**KEY WORDS: TOOLS CASE-SYSTEM-SOFTWARE-DEVELOPMENT**

## **INTRODUCCION**

Las Herramientas CASE desde sus inicios en los años setenta empezaron con un simple procesador de palabras diseñado para generar y manipular documentación, continuando con diagramas de flujo y métodos gráficos que con el tiempo era necesario realizar sus respectivos cambios debido a que estos eran muy complejos.

A través del tiempo las herramientas CASE permitieron crear y modificar los diagramas, logrando incrementar la calidad al desarrollar el software, que pronto se desarrollaron herramientas graficas integradas con diccionarios de base de datos, finalmente se desarrollaron Herramientas Case para ayudar e cada etapa del ciclo de desarrollo de software.

El objetivo principal de las Herramientas CASE es el aumento en la productividad, debido a que esta permite generar de forma automática los códigos, logrando la mejora de calidad y rendimiento al desarrollar software. En la actualidad las Herramientas CASE permiten automatizar todo el ciclo de vida de desarrollo de software.

En las organizaciones se presenta con una evolución tecnológica y de gran potencial, se debe tener claro las herramientas CASE tomando en consideración que se debe tener en cuenta las facilidades que presentan las Herramientas CASE al adquirirlas.

## CAPITULO I

### PROBLEMA DE INVESTIGACION

En la actualidad el objetivo principal de los desarrolladores de software es elaborar aplicaciones de calidad, con costos bajos, tiempo mínimo y permitiendo que las aplicaciones desarrolladas tengan compatibilidad con los diferentes sistemas operativos, bases de datos y lenguajes de programación.

Para cumplirlo existen herramientas capaces de solucionar, consolidar, acoplar y unir los diversos requerimientos de los clientes. La principal es el uso de Herramientas CASE, la cual permite desarrollar aplicaciones en un entorno gráfico, manteniendo una metodología de programación ordenada y enfocada al desarrollo de sistemas basado en los estándares de la ingeniería de software, logrando desarrollar sistemas de alto nivel competitivo.

El desarrollo de soluciones informáticas debe ser adaptable y no enfocarse en una sola base de datos y un lenguaje de programación específico, debe ser capaz de ejecutarse en diferentes arquitecturas sin la necesidad de realizar cambios extremos en su sistema maestro.

Para entenderlo de mejor manera presento el siguiente ejemplo: Un cliente A necesita un sistema de contabilidad desarrollado en Visual Basic, Sql Server y Windows. El cliente B necesita el mismo sistema pero en un sistema operativo diferente y un tercer cliente necesita el mismo sistema pero usando Java y en Linux.

Para satisfacer las demandas de los clientes tenemos dos opciones la primera contratar tres desarrolladores diferentes expertos en cada arquitectura requerida, la segunda opción es desarrollar un solo sistema con el uso de herramientas CASE que sea capaz de generar las aplicaciones cumpliendo los requerimientos mencionados.

La solución más óptima es la segunda, pues permite elaborar un solo sistema o prototipo para cada arquitectura requerida, el uso de las herramientas CASE permiten generar aplicaciones multilenguaje, multiplataforma y en base de datos diferente.

## **OBJETIVO GENERAL**

Analizar la importancia de las Herramientas CASE en el desarrollo de software.

## **OBJETIVO ESPECIFICO**

- Evaluar como las herramientas CASE aumenta la productividad en las áreas de desarrollo de los sistemas informáticos.
- Evaluar de qué manera inciden en la mejora de la calidad del software desarrollado.
- Determinar como el desarrollo mediante herramientas CASE permite reducir tiempo y costos.

## **JUSTIFICACION**

El desarrollo de sistemas avanza a gran escala por tal motivo en la actualidad existen muchos lenguajes de programación, bases de datos y arquitecturas diferentes. Para facilitar el desarrollo de soluciones informáticas las herramientas CASE permiten automatizar los procesos en el desarrollo de software, dando como resultados sistemas compatibles en el mercado actual.

Si se desea satisfacer la demanda actual de soluciones informáticas en tiempos cortos, reduciendo costos y con calidad, las herramientas CASE es una de las opciones más optimas porque permite realizar todo lo mencionado, es importante conocer los conceptos y formas de desarrollo convencional de software son de mucha importancia, los desarrolladores deben conocer sólidamente las bases al desarrollar, para tener una concepción adecuada sobre la automatización de los mismos por parte de las herramientas CASE.

## **CAPITULO II**

### **MARCO TEORICO CONCEPTUAL**

#### **DEFINICION DE HERRAMIENTAS CASE**

Una Herramienta CASE es un programa especializado en el control y desarrollo de aplicaciones informáticas, siguiendo alguna de las metodologías más extendidas como los diagramas de control de flujo de Yourdon o los diagramas Entidad / relación de P.P. Chen para la normalización de bases de datos. En grandes equipos de desarrollo, el sistema queda centralizado a través de un diccionario de datos o "repositorio" que coordina los desarrollos de todos los participantes (MONTES, 2018).

#### **EL PROCESO DE DESARROLLO DE SOFTWARE**

El proceso de desarrollo de software consiste en una serie de pasos bien definidos, que seguidos adecuadamente, conducen a un software mantenible y bien diseñado, aun así, muchas organizaciones olvidan las fases de análisis y diseño a favor de comenzar inmediatamente la implementación de código. (Rangel, 2010)

Es más positivo pensar en el desarrollo de software, no como un proceso lineal, sino como un ciclo, aunque el paso de una fase a otra se realiza en un sentido, también pueden existir vueltas atrás en determinados momentos, especialmente cuando aparecen requerimientos de usuario ocultos en las primeras fases, a continuación se muestra cuáles son las principales facetas para desarrollar software. (Caceres & Cruz, 2006)

Análisis de Requerimientos, Diseño de la Especificación (Prototipo), Implementación (Producción), Integración, Test y Documentación., Mantenimiento y Reingeniería.(Sommerville, 2011a)

#### **CLASIFICACION DE LAS HERRAMIENTAS CASE**

Las Herramientas CASE por su complejidad, no tienen una clasificación específica de sus tipos, varios autores las clasifican de diferente manera de acuerdo a la forma de ver e interpretar las cosas. Una de las más importantes clasificaciones de Herramientas CASE se observa en la figura 1.(Chavarría & Ocotilla, 2016)

## CLASIFICACION GENERAL DE LAS HERRAMIENTAS CASE

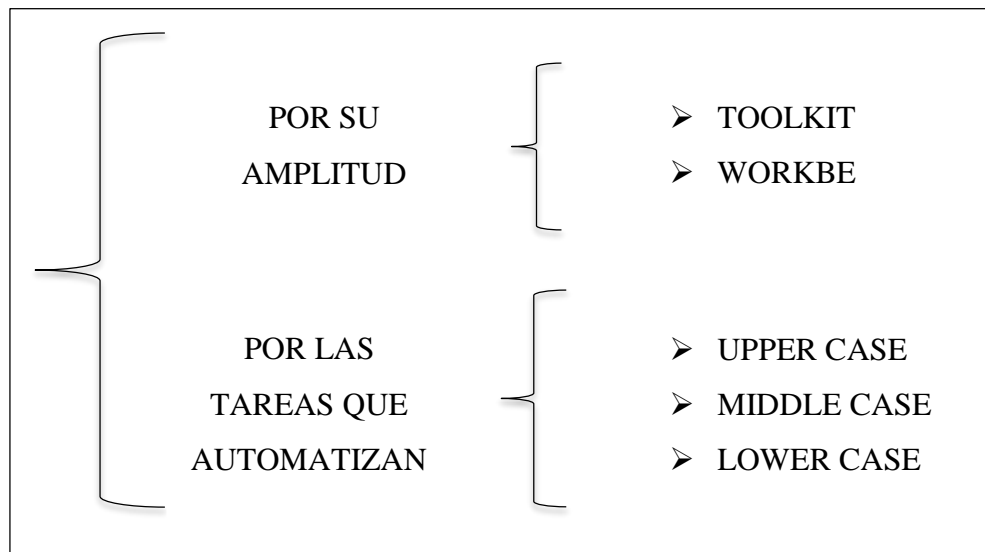


Figura. 1 Clasificación general de las Herramientas CASE

### POR SU AMPLITUD

\* **TOOLKIT**: se define a la colección de herramientas integradas en las que permiten mecanizar un conjunto de tareas en ciertas fases de un sistema informático.(Carrillo Sarabia, 2008)

\* **WORKBENCH**: estas son herramientas de soporte que permite la automatización del proceso completo en el desarrollo mediante una vía informática(Grado et al., 2015)

### POR LAS TAREAS QUE AUTOMATIZAN

- **UPPER CASE**: sirve como un planificador estratégico en las que efectúa desarrollo funcional de planes corporativos.(Grado et al., 2015)
- **MIDDLE CASE**: Análisis y diseño, como Designer de Oracle. (Gallardo, 2012)
- **LOWER CASE**: Generación de código, test e implantación, como Genexus de Artech.(Zapata & Chaverra, 2010)

## **OTROS TIPOS DE CASE**

Como se muestra en la figura 2 las Herramientas CASE también pueden clasificarse de la siguiente manera: (Universidad Pontificia de Valencia, 2010)

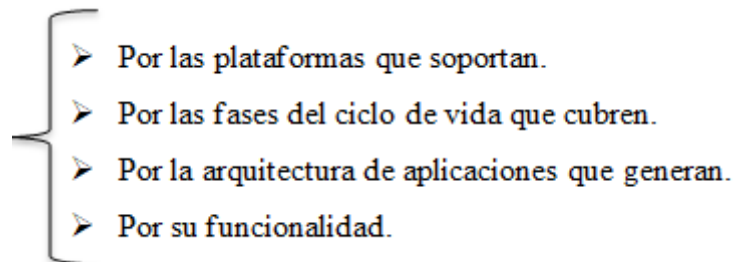
- 
- Por las plataformas que soportan.
  - Por las fases del ciclo de vida que cubren.
  - Por la arquitectura de aplicaciones que generan.
  - Por su funcionalidad.

Figura. 2 Clasificación de las Herramientas CASE

### **POR LAS PLATAFORMAS QUE SOPORTAN:**

Case Uniplataforma y Case Multiplataforma.(Grado et al., 2015)

### **POR LAS FASES DEL CICLO DE VIDA QUE CUBREN**

- CASE integrado: estas permiten abarcar las distintas fases del ciclo de sistemas, también conocidas como workbench CASE (Salas, 2011)
- TOOLKIT. Herramientas que cubren alguna de las partes del desarrollo.(Grado et al., 2015)

### **POR LA ARQUITECTURA DE APLICACIONES QUE GENERAN**

Case Centralizadas, Case Cliente Servidor (Dos Capas) y Case Cliente Servidor (Multicapa).(Grado et al., 2015)



**VENTAJAS Y DESVENTAJAS DEPENDIENDO DEL TIPO DE HERRAMIENTA CASE UTILIZADA**

<i><b>TIPO</b></i>	<i><b>VENTAJAS</b></i>	<i><b>DESVENTAJAS</b></i>
<b>Upper CASE</b>	<p>Se utiliza en arquitecturas para PC y es aplicable en diferentes entornos</p> <p>Menor Costo</p>	<p>Mejora la calidad pero no la productividad.</p> <p>Permite la integración del ciclo de vida.</p>
<b>Lower CASE</b>	<p>Mejora la productividad a corto Plazo.</p> <p>Buen soporte al mantenimiento.</p>	<p>No garantiza la persistencia en niveles corporativos.</p> <p>No garantiza la eficiencia de análisis y diseño.</p> <p>No permite la integración del ciclo de vida.</p>
<b>I – CASE</b>	<p>Integra el ciclo de Vida.</p> <p>Mejora la productividad a mediano plazo.</p> <p>Buen soporte de mantenimiento.</p> <p>Mantiene la persistencia en niveles corporativos.</p>	<p>No es eficiente para niveles simples, sino para complejos.</p> <p>Depende del hardware y software.</p> <p>Costos elevados de implementación.</p>

(Instituto, 1999)

## CASE EN EL CICLO DE VIDA DE UN SISTEMA

En la figura 3 se puede observar qué partes del ciclo de vida pueden generar los diferentes tipos de Herramientas CASE. (ORTEGA & SANTA VILLA, 2012)

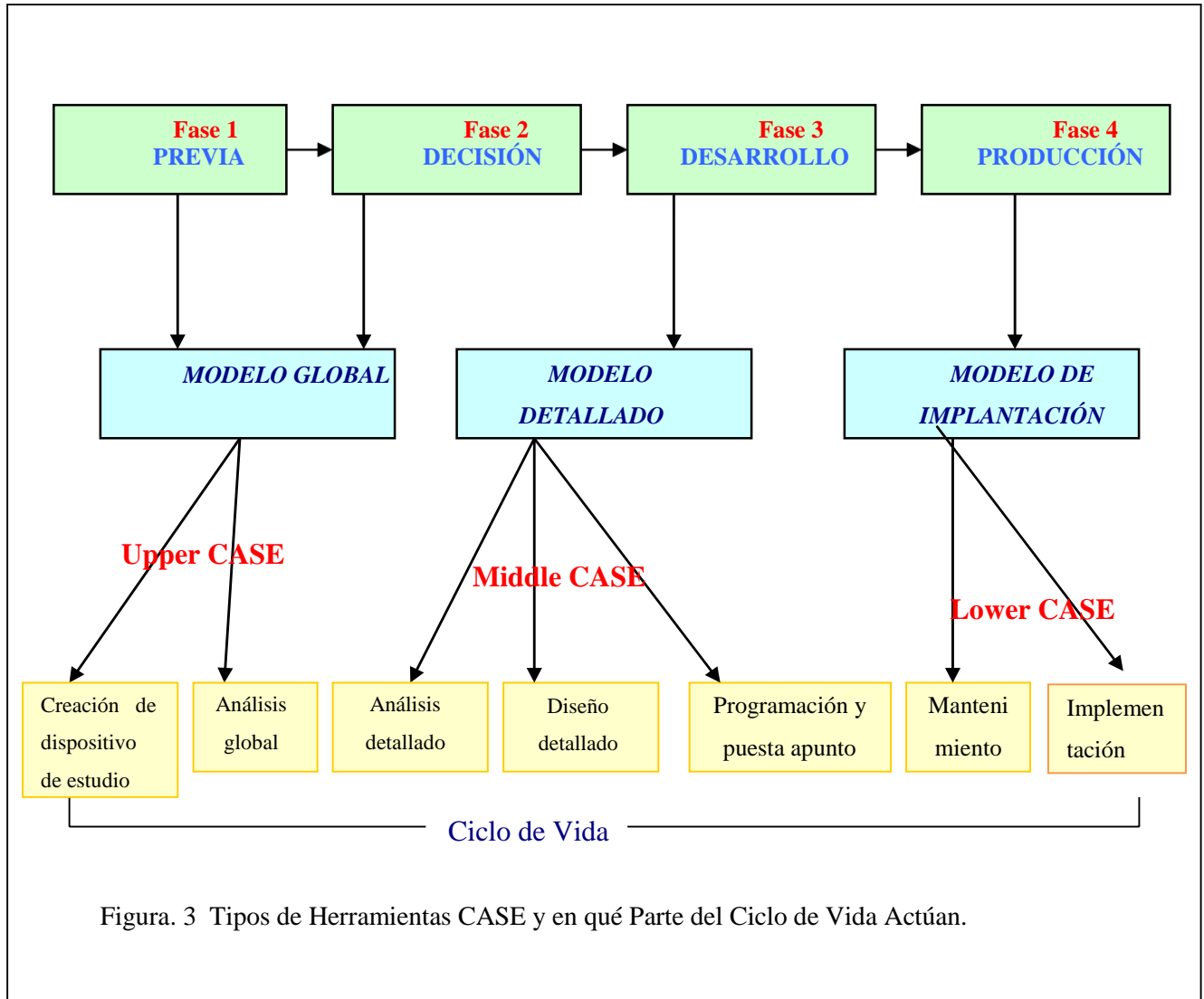


Figura. 3 Tipos de Herramientas CASE y en qué Parte del Ciclo de Vida Actúan.

(Grado et al., 2015)

## ESTRUCTURA DE LAS HERRAMIENTAS CASE

Es muy difícil especificar cuál es la estructura exacta de una Herramienta CASE, ya que no todas tienen las mismas características unas de otras, a continuación se presenta la estructura más general a manera de niveles o módulos: Módulo de Repositorio, Módulo de Diagramación y Modelamiento, Módulo de Prototipado, Módulo de Generación de Código y Módulo de Generación de Documentación. (Chavarría & Ocotilla, 2016)

## MODULO DE REPOSITORIO O ENCICLOPEDIA

En el contexto de las Herramientas CASE se entiende por enciclopedia a la base de datos que contiene toda la información relacionada con: las especificaciones, análisis y diseño de la aplicación, definiendo a cada objeto de la siguiente manera:(Universidad Pontificia de Valencia, 2010)

- **Datos:** Elementos, atributos (campos), asociaciones (relaciones), entidades (registros), almacenes de datos y estructuras.(Pare & Casillas, 2005)
- **Procesos:** Procesos, funciones y módulos.(Grado et al., 2015)
- **Gráficos:** Diagramas de flujo de datos, diagramas entidad relación, diagramas de descomposición funcional, diagramas de estructura de árbol y diagramas de clases.(Caceres & Cruz, 2006)
- **Reglas:** Reglas de gestión de métodos y comportamiento de datos y procesos.(Grado et al., 2015)

Las herramientas Case tiene como base de datos principal a un repositorio , debido a que este permite abarcar toda la información desde el inicio de la vida de dicho sistema. En algunas referencias se le denomina Diccionario de Recursos de Información y en otros casos se le llama Base de Conocimiento, si a una Herramienta CASE se la considera como un Sistema Experto, el repositorio sería la fuente de alimentación de la información.(Carrillo Sarabia, 2008)

Debido a que la mayoría de H.C. tienen un repositorio, es preferible trabajar con herramientas que tengan su propio repositorio ya que se adaptan a la tecnología propia de la herramienta y no tienen que heredar metodologías de otro fabricante.(Grado et al., 2015)

## CARACTERISTICAS DE UN REPOSITORIO

**Tipo de Información:** Nos permite formar: datos, gráficos, procesos, informes, modelos y reglas.(Grado et al., 2015)

**Tipo de Controles:** se encarga de incorporar modulos de gestión logrando su mantenimiento de versiones, información o acceso por claves.(Carrillo Sarabia, 2008)

**Tipo de Actualización:** los cambios que se efectúen en los diseños se verán reflejados en tiempo real en el repositorio. (Grado et al., 2015)

**Reutilización de Módulos para Otros Diseños:** la clave para poder localizar, identificar para extraer código es mediante el repositorio. (MONTES, 2018)

### **INTERFASES AUTOMÁTICAS CON OTROS REPOSITORIOS Y BASES DE DATOS EXTERNAS.**

Ayudan a la migración de aplicaciones, bases de datos, datos, y en si a la migración de toda la aplicación, ya sea como procesos de reingeniería o como procesos de cambios de versión de la herramienta, se toma como procesos de reingeniería a las aplicaciones y bases de datos realizadas con otras herramientas y se las acopla a la tecnología de una Herramienta CASE y los procesos de cambio de versión actúan cuando se quiere migrar aplicaciones de una versión más antigua a una versión más actual, sin que esto lleve a un cambio de fondo en la programación de la aplicación; en la figura 4. Se observa el módulo de repositorio. (Grado et al., 2015)

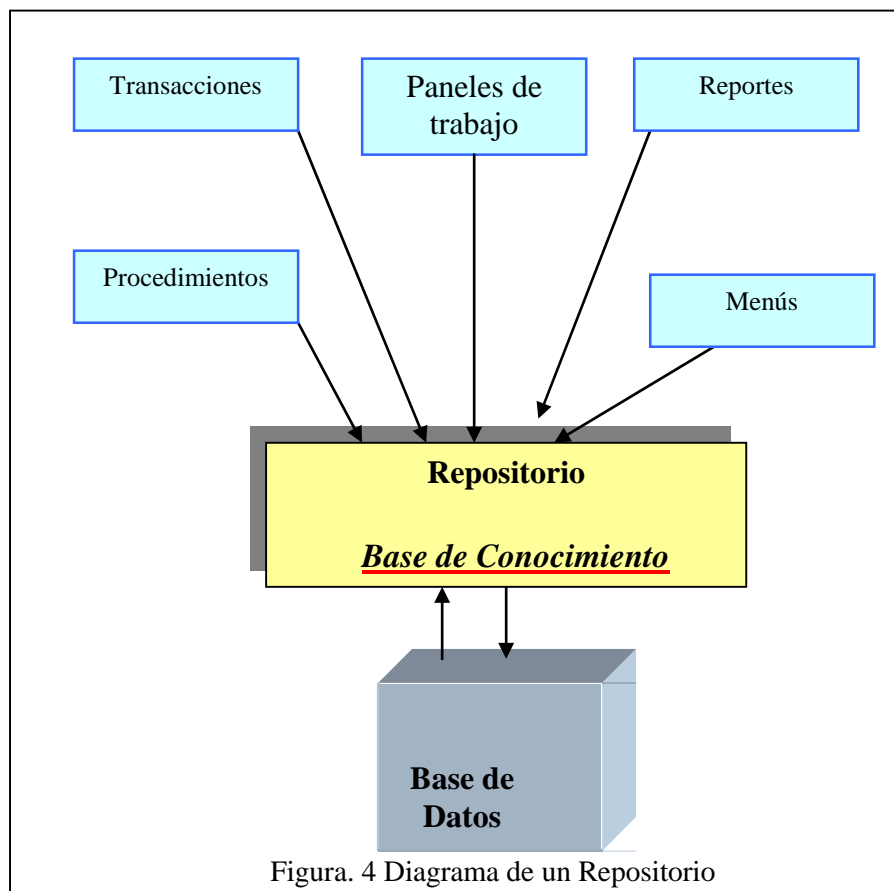


Figura. 4 Diagrama de un Repositorio

(Grado et al., 2015)

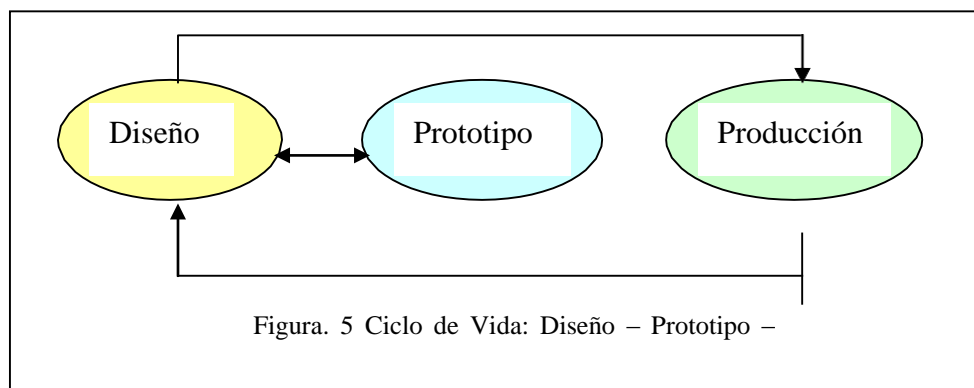
## MODULO DE DIAGRAMACION Y MODELAMIENTO

Es la parte del sistema, donde el programador diseña la realidad, dicho de otra manera diseña lo que desea ver propiamente en su programa, luego de esto la Herramienta CASE automáticamente genera los diferentes diagramas, como: Diagramas de flujo de datos, Diagramas entidad relación, Historia de vida de las entidades, Diagramas de estructura de datos, Diagrama de estructura de cuadros, Diagramas de estructura de árboles y Técnicas matriciales.(Martínez, 2013)

## MODULO DE PROTOTIPADO

Esta herramienta permite revelar al usuario desde el inicio del diseño, permitiendo que el software cuente con la mayoría de las necesidades que pueda tener el usuario, antes de que la aplicación entre a producción.(Sommerville, 2011b)

El programador diseñara el software conjuntamente con el usuario, haciéndole ver como se van a presentar en pantalla los criterios requeridos por él. En la figura 5 se observa el ciclo de vida de diseño, prototipo y producción. (Chavarría & Ocotilla, 2016)



(Grado et al., 2015)

La herramienta será más efectiva cuando consiga la implicación del usuario final en el diseño de aplicación la construcción del prototipo.(Grado et al., 2015)

Los prototipos fueron diseñados con la finalidad de utilizarse en el desarrollo de sistemas tradicionales tomando en cuenta que estos proporcionan una realimentación al instante ayudando de esta manera al sistema. Las HC están efectuadas para la creación de prototipos con seguridad y con rapidez. En el ciclo de prototipo, el diseñador recorrerá repetidamente el bucle Diseño – Prototipo durante la fase de diseño, construyendo y probando sucesivos prototipos del modelo. (Valle, 2017)

En el ciclo de producción por el contrario, pasará con menos frecuencia el bucle Diseño Producción, ya que la generación del sistema se realiza solamente cuando el prototipo ha sido totalmente aprobado, o luego de haber instrumentado y probado algún cambio. (Lopez, Gonzalez, & Gallud, 2010)

### **MODULO DE GENERADORES DE CODIGO**

Estos son utilizados normalmente en ordenadores de estaciones de trabajo o personales. Herramienta CASE genera código automáticamente en diferentes lenguajes. (Grado et al., 2015)

### **MODULO GENERADOR DE DOCUMENTACION**

Este módulo se suman la ayuda en línea, los help o las ayudas del programa. (Instituto, 1999)

## **CAPITULO III**

### **METODOLOGIA**

El marco metodológico utilizado en la presente investigación permitió analizar y profundizar el tema propuesto, haciendo uso de la investigación descriptiva e investigación documental.

- Investigación descriptiva: Permite describir completamente la investigación realizada por medio de la información obtenida de las diferentes fuentes bibliográficas.
  
- Investigación Documental: Este tipo de investigación nos permitió recopilar datos e información a través del uso de técnicas e instrumentos de recolección con la finalidad de analizarlos e interpretarlos.

Las herramientas de investigación utilizadas principalmente el análisis de documentos nos permite obtener información amplia de nuestro problema de investigación, planteando lo más relevante del tema a investigar, para su posterior análisis y verificación.

## CAPITULO IV

### DESARROLLO DEL TEMA

En la actualidad existen una gran variedad de Herramientas CASE una de las más completas y la cual propongo para el desarrollo de software es GENEXUS porque cubre todo el ciclo de vida en el desarrollo de software, la cual está dentro de las herramientas tipo WORKBENCH. Desarrollado por un equipo de investigadores Uruguayos, y está patentado por la empresa Uruguaya ARTECH, actualmente distribuido en Ecuador por GMS (Grupo Microsistemas).

El objetivo de Genexus, es ayudar a los desarrolladores a elaborar sus aplicaciones en el menor tiempo y con la mejor calidad posible, omitiendo ciertos pasos que son automatizables, y concentrándose en las partes verdaderamente complejas como son el análisis y diseño. Desde el punto de vista teórico, Genexus es más una metodología para desarrollar aplicaciones que una herramienta de software. Está relacionado con las metodologías tradicionales, pero aporta grandes y novedosos cambios a las mismas.

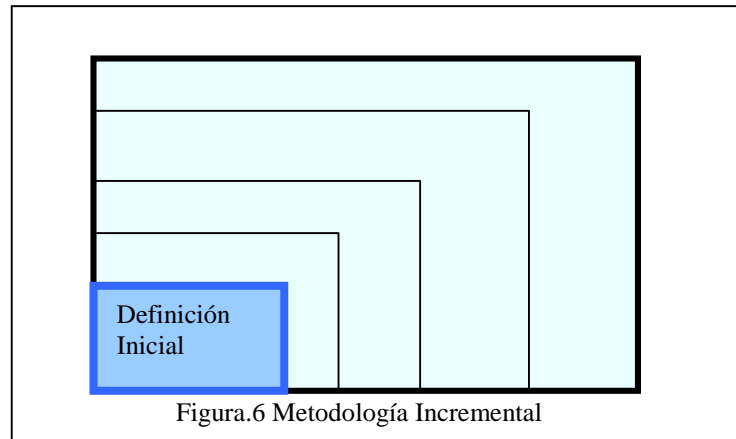
Al comenzar las aplicaciones por el desarrollo de la interfase de usuario, revoluciona el mundo de la metodología tradicional, ya que el programador no tiene que preocuparse por la implementación física del sistema como: creación y normalización de las bases de datos, generación de programas, generación de diagramas, generación de documentación, etc.

El punto de partida de Genexus, es “DESCRIBIR LAS VISIONES DE LOS USUARIOS O DISEÑO DE LA REALIDAD”; a partir de este modelo mediante una “síntesis de visiones” se construye el soporte computacional que es el eje de la aplicación. En la mayoría de metodologías tradicionales como: análisis estructurado, ingeniería de la información, análisis esencial, etc. Se separan el análisis de datos, de los procesos y programas; Genexus hace un análisis conceptual unificado que se lo conoce como “metodología incremental”.

Cuando se comenzaron a utilizar verdaderos modelos corporativos que normalmente poseen cientos de tablas, se confirmó que no debía perderse más tiempo buscando algo que no existe: las bases de datos estables. Luego de varias investigaciones, se desarrolló una teoría la cual fue crear una herramienta que pudiera posibilita el desarrollo



incremental y permitir la convivencia con las bases de datos reales, la figura.6 explica el crecimiento del modelo incremental.



La construcción automática de la base de datos y programas ejecutables a partir de una única especificación, permitirá a Genexus utilizar la metodología incremental; este proceso se realiza mediante “*aproximaciones sucesivas*”.

En cada momento se desarrolla la aplicación con el conocimiento que se tiene, y luego cuando se pasa a tener: más, menos o nuevo conocimiento; Genexus se ocupará de hacer automáticamente todas las nuevas adaptaciones en las bases de datos y programas.

Utilizando Genexus la tarea básica del analista es la “*descripción de la realidad*”. Solo el hombre podría realizar esta tarea, ya que solo él puede entender los problemas del usuario. Desde luego que esto modifica la actividad del analista e incluso su perfil supuestamente óptimo ya que lo transforma en un *Business Analyst* o analista del negocio. (Genexus, 2018)

Ahora trabaja en alto nivel, discutiendo problemas con el usuario y probando con él las especificaciones a nivel de prototipo, en vez de desarrollar su actividad en tareas de bajo nivel como: diseñar archivos, normalizar, diseñar programas, programar, buscar y eliminar errores de programación, etc.

## DESARROLLO TRADICIONAL

Para empezar a analizar la estructura de las Herramientas CASE generadores de aplicaciones distribuidas, se tomará como patrón a JAVA, ya que es el lenguaje de programación pionero y más versátil para la realización de aplicaciones distribuidas.

Nov 2018	Nov 2017	Change	Programming Language	Ratings	Change
1	1		Java	16.746%	+3.51%
2	2		C	14.396%	+5.10%
3	3		C++	8.282%	+2.94%
4	4		Python	7.683%	+3.20%
5	7	▲	Visual Basic .NET	6.490%	+3.58%

(TIOBE, 2018)

JAVA es el estándar de la mayoría de plataformas de software, sin menospreciar a otras herramientas como la tecnología .NET de Microsoft y otras herramientas existentes en el mercado. Este estudio se va a realizar tomando como referencia a JAVA interactuando con la Herramienta CASE GENEXUS.

## DESARROLLO DE APLICACIONES

Al desarrollar en Genexus se debe tener claro su estructura posee objetos básicos generados por Genexus que sirven en todo el desarrollo del software:

**TRANSACCIONES:** Las transacciones permiten definir los objetos de la realidad. La mayor parte de las transacciones pueden ser identificadas prestando atención a las entidades que el usuario menciona, como: clientes, proveedores, productos, etc. A partir de las transacciones, Genexus infiere el diseño de la base de datos.

**PROCEDIMIENTOS:** Son los procesos no interactivos de actualización de la base de datos; son conocidos como procesos batch.

**REPORTES:** Son aquellos que recuperan información a partir de los datos almacenados y no los alteran. A los reportes generalmente se los conoce como listados.

**PANELES DE TRABAJO Y WEB PANELS:** Permiten definir consultas interactivas a las bases de datos, ya sea en modo windows o en modo web.

**MENUES:** Son objetos organizadores del resto de objetos Genexus, los cuales sirven para realizar listas desplegables de selección de objetos.

### **REQUISITOS PARA ADQUIRIR HERRAMIENTAS CASE**

La primera fase que debe abordarse de manera metódico dentro de la evolución de adquisición, es el estudio de las micción existentes, que deberán ser satisfechas a través de la creación del instrumento que se va a obtener. El comprador debe coincidir lo siguiente: (gobierno digital, 2018)

- Los principales requisitos funcionales que debe cumplir la herramienta. (gobierno digital, 2018)
- El tipo de facilidades de uso que debe prestar. (gobierno digital, 2018)
- Las limitaciones y restricciones que se derivan del entorno de operación previsto. (gobierno digital, 2018)

### **REQUISITOS FUNCIONALES**

En función de los requisitos funcionales se podrá deducir qué tipo de herramienta es la más adecuada. Algunos factores a tener en cuenta y que son comunes en todas estas herramientas son: (gobierno digital, 2018)

- Tipo(s) de plataforma(s) sobre las que deberá funcionar la herramienta, tanto desde el punto de vista del equipamiento lógico como del equipamiento físico. (gobierno digital, 2018)
- Requisitos físicos (espacio en disco, memoria RAM, UCP, etc). (gobierno digital, 2018)
- Necesidad de integración con otras herramientas de ayuda al desarrollo ya existentes. (gobierno digital, 2018)
- Necesidad de acceso simultáneo para diferentes usuarios. Esto puede enfocar la elección hacia una herramienta que permita accesos compartidos a los datos y que cuente con una definición de perfiles de usuario para la protección de información. (gobierno digital, 2018)

- Necesidad de compartir datos con aplicaciones externas. Se valorará más a aquella aplicación que permita exportar sus datos o que almacene la información en un formato de fácil acceso para otra aplicación. (gobierno digital, 2018)

## **FACILIDADES DE USO DE UNA HERRAMIENTA CASE**

**Funcionalidad Requerida:** Es importante definir con el mayor grado de aproximación, cuáles son las funciones que se le van a pedir a la herramienta. Para ello, es necesario analizar si las necesidades son cubiertas con una herramienta integrada o con una orientada a alguna de las fases del ciclo de vida del desarrollo. (repositorio.utn, 2018)

**Metodología Soportada:** Si en la organización ya existe una metodología y técnicas, la herramienta deberá soportar dicha metodología, así como las técnicas empleadas en cada fase. Si la Herramienta CASE va a servir precisamente para introducir un nuevo método de trabajo, habrá que asegurarse de que dicho método es el adecuado. En ocasiones, para adaptarse a una metodología, es preciso realizar desarrollos adicionales en la herramienta. (repositorio.utn, 2018)

**Generación Automática de Código:** En algunos casos la necesidad predominante del usuario puede consistir en la generación automática de código fuente (programas), a partir de productos del diseño fuertemente formalizados (scripts, formatos, etc.). En tal caso, deberán conocerse los pormenores de tal necesidad, como lenguajes de programación admisibles como salida, generación en tiempo real o en un proceso por lotes, etc. (repositorio.utn, 2018)

**Capacidad de Integración en la Arquitectura Existente:** Habrá que tener en cuenta la plataforma o plataformas diferentes, ordenadores que deberán soportar la Herramienta CASE, su tipología (fabricante, modelo y sistema operativo cuando menos) y las características de la red de interconexión cuando exista. Ello tendrá importancia a la hora de garantizar la compatibilidad del equipamiento existente con los nuevos productos que se van a adquirir. Lo mismo debe hacerse en relación con las herramientas lógicas previamente existentes en esas plataformas, siempre que deban integrarse en mayor o menor medida con los nuevos productos. Se deberá considerar cuáles son los recursos disponibles en el equipamiento existente para la implantación de la Herramienta CASE en cuestión. Deberán conocerse con el mayor detalle, posibles cuestiones como memoria

RAM y espacio en disco necesario, grado de utilización de la(s) UCP(s) en condiciones normales de operación y de picos de demanda de la nueva herramienta. Este punto es importante de cara a un posible redimensionamiento del equipamiento disponible. Estas mismas consideraciones también deben ser tomadas en cuenta no ya para la propia Herramienta CASE, sino para las aplicaciones desarrolladas con ayuda de dicha herramienta. (repositorio.utn, 2018)

**Modo de Funcionamiento:** Será bueno conocer el modo de funcionamiento: monousuario o multiusuario, así como el grado deseable de centralización de los recursos y funciones asociadas con la administración y operación de la Herramienta CASE que se va a implantar. (repositorio.utn, 2018)

**Personalización del Entorno:** Finalmente, deberán considerarse las necesidades o conveniencias de la personalización del sistema, en función de los diferentes perfiles de usuario de la herramienta. (repositorio.utn, 2018)

## **CAPITULO V**

### **CONCLUSION**

La utilización de Herramientas CASE maximiza la calidad en todo el ciclo de vida al desarrollar aplicaciones informáticas mejorando la productividad y automatizando los procesos de creación, normalización y generación de la base de datos, generación de documentación, generación de diagramas, generación de código, generación de pantallas y generación de paneles de trabajo. Permite al programador trabajar bajo una metodología de diseño de software, dejando a un lado la programación de la aplicación, centrándose en el análisis y el diseño, la cual solo puede ser realizada por un humano.

Las Herramientas CASE crea una cultura ordenada y sistemática de programación, esto permite reducir costos y tiempo al desarrollar; facilita la generación de software porque permite que cada aplicación generada no se tenga que programar nuevamente todo el software, sino solo realizar pequeñas modificaciones en la base de conocimiento, logrando de esta manera la reutilización del código ya generado, también permite migrar a diferentes lenguajes de programación, bases de datos y plataformas existentes en el mercado.

Genexus posee un entorno grafico amigable e interactivo el cual facilita a la generación de software, cuenta con todas las ventajas de una Herramienta CASE WORKBENCH, la cual aporta con la generación de prototipos desde el principio de las etapas de desarrollo, permitiendo corregir errores desde el inicio del desarrollo de software.

## REFERENCIAS BIBLIOGRÁFICAS

- Alarcon, A., & Sandoval, E. (2011). Herramienta CASE para ingeniería de requisitos. *Herramienta CASE Para Ingeniería de Requisitos*, (January), 1–14. <https://doi.org/10.1109/ICCRD.2010.71>
- Caceres, A. M., & Cruz, R. E. (2006). Facultad de Ingeniería Escuela de Computación Análisis y Diseño de Sistemas. Retrieved from <http://rcruz0423.galeon.com/docs/case.pdf>
- Carrillo Sarabia, P. de las M. (2008). ANÁLISIS DE HERRAMIENTAS CASE APLICADO A UN SISTEMA DE PROVEEDURÍA USANDO LA METODOLOGÍA ADOOSI. *Evolution*, 1–14.
- Chavarría & Ocotilla. (2016). Sobre el uso de herramientas CASE para la enseñanza de bases de datos, 51–56.
- Gallardo, Z. (2012). *Academia*. Obtenido de [https://www.academia.edu/5012832/Herramienta\\_CASE](https://www.academia.edu/5012832/Herramienta_CASE)
- Genexus. (2018). *Genexus Software That Makes Software*. Obtenido de <https://www.genexus.com/es/global/productos/genexus-2012>
- Grado, T. De, Francisco, J., & Bastidas, G. (2015). INTRODUCCIÓN A LAS HERRAMIENTAS CASE.
- Instituto, N. de E. e I. (1999). Herramientas Case, 54.
- Lopez, P., Gonzalez, A., & Gallud, J. (2010). *Herramientas CASE. ¿Como incorporarlas con éxito en nuestra organizacion?* Castilla.
- Luis, J., & Valero, P. (2010). Ingeniería de Software Asistida por Computadora (CASE) Orígenes del CASE, 1–12. Retrieved from <https://www.virtuniversidad.com/greenstone/collect/informatica/archives/HASH01ea.dir/doc.pdf>
- Martínez, B. E. V. (2013). PANORAMA GENERAL DE LAS “HERRAMIENTAS

CASE,” 2013.

MONTES, M. (2018). *ACADEMIA*. Obtenido de [https://www.academia.edu/4513393/Libro\\_HERRAMIENTAS\\_CASE](https://www.academia.edu/4513393/Libro_HERRAMIENTAS_CASE)

ORTEGA, D. L., & SANTA VILLA, J. A. (2012). ESTUDIO COMPARATIVO DE LAS HERRAMIENTAS CASE: STARUML, POSEIDON FOR UML Y ENTERPRISE ARCHITECT, PARA EL MODELAMIENTO DE DIAGRAMAS UML.

Pare, R., & Casillas, L. (2005). *Bases de datos. Nursing times* (Vol. 64). <https://doi.org/B-15.562-200>

Rangel, M. M. (2010). *Ingenieriaderequisitosdesoftware*. Obtenido de <https://ingenieriaderequisitosdesoftware.wordpress.com/2010/04/03/el-proceso-de-la-ingenieria-de-requisitos/>

Salas, M. Z. (2011). *Fundamentos de Ingeniería de Software*. Obtenido de <https://sites.google.com/site/fundamentosdeingendofware/u1-fundamentos-ingenieria-de-software/1-6-clasificacion-de-las-herramientas-case>

Sommerville, I. (2011a). *INGENIERÍA DE SOFTWARE*. <https://doi.org/10.1111/j.1365-2362.2005.01463.x>

Sommerville, I. (2011b). Procesos de Software. *Ingeniería de Software*, 36–38.

TIOBE. (2018). *TIOBE the software quality company*. Obtenido de <https://www.tiobe.com/tiobe-index/>

Universidad Pontificia de Valencia. (2010). Introducción a Herramientas CASE y System Architect, 1–13.

Valle, E. C. (2017). *EJECUTIVOS de INFORMATICA*.

Zapata, C. M., & Chaverra, J. J. (2010). UNA MIRADA CONCEPTUAL A LA GENERACIÓN AUTOMÁTICA DE CÓDIGO, 143–154.