

UNIDAD EDUCATIVA

“BAÑOS”

Baños de Agua Santa – Tungurahua



PROYECTO INTERMODULAR

“Temas de Tercero de Bachillerato del Currículo del Módulo Formativo N°3: Programación y Base de Datos”

Figura Profesional: Informática

Elaborado por: Anahi Sailema

Pamela Aponte

Elkyn Zambrano

Curso: Segundo “H”

Módulo formativo:

Módulo 3: Programación y Base de Datos

Docentes Técnicos: Mg. Diana Fuentes

Año Lectivo: 2024-202

Índice

Índice	2
Estructuras de control:.....	3
1. Clases y funciones	3
2. Librerías	3
3. Código fuente	3
4. Desarrollo de programas	3
5. Documentación de programas.....	4
Herramientas de desarrollo.....	5
1. Generadores de pantallas (Interfaces de usuario)	5
2. Generadores de informes.....	5
3. Generadores de consultas	6
4. Generador de aplicaciones	7
Generación y desarrollo de aplicaciones con herramientas CASE:	9
1. Características de las Herramientas CASE	9
2. Estructura de las Herramientas CASE	10
3. Actualización y Mantenimiento del Proyecto.....	10
4. Generación de Código.....	10
5. Documentación generada	11
6. Tutoriales y Reportes de Desarrollo	11
7. Ventajas de Usar Herramientas CASE	12

Estructuras de control:

Son instrucciones en un programa que permiten alterar el flujo de ejecución en función de ciertas condiciones o repeticiones. Ayudan a que un programa tome decisiones, repita ejecutar acciones o bloques de código de manera condicional. Por ejemplo, en Dev C++ se usan:

- **if, else, while, for, etc.,** se utilizan para controlar el flujo del programa.

1. Clases y funciones

Las clases son plantillas para crear objetos (instancias) en programación orientada a objetos (OOP), y las funciones son bloques de código reutilizables que realizan tareas específicas.

- C++ soporta la Programación Orientada a Objetos (OOP), permitiendo el uso de clases, constructores y métodos.

2. Librerías

Son colecciones de funciones, clases y otros recursos que permiten realizar tareas comunes sin necesidad de escribir el código desde cero.

- Se pueden usar librerías estándar, como **math**, **stdio.h**, **conio.h**, para realizar tareas matemáticas comunes

3. Código fuente

Es el conjunto de instrucciones escritas en un lenguaje de programación que define el comportamiento de un programa.

4. Desarrollo de programas

Es el proceso completo de crear un software, desde la planificación, escritura de código, pruebas y documentación.

5. Documentación de programas

Es la descripción detallada del funcionamiento de un programa, incluyendo la explicación del código, las funciones, la estructura de datos y cómo se usa el software.

- Los comentarios en C++ pueden explicar el **propósito del código**, lo que es **útil** para la **comprensión** y **mantenimiento** del programa.

(Prunello, 2024)

Herramientas de desarrollo

Las herramientas de desarrollo son software que ayudan en la creación, diseño, prueba y mantenimiento de aplicaciones. Estas herramientas simplifican la creación de interfaces de usuario, generación de informes y desarrollo de aplicaciones. A continuación, te explico cada una de estas herramientas con más detalle:

1. Generadores de pantallas (Interfaces de usuario)

Un generador de pantallas es una herramienta para crear interfaces gráficas de usuario (GUI) sin escribir mucho código. Permite arrastrar y soltar elementos para diseñar aplicaciones, facilitando el proceso a quienes tienen poca experiencia en diseño.

Ejemplos de generadores de pantalla:

Visual Studio: Ofrece herramientas como el Windows Forms Designer o WPF para crear interfaces gráficas en aplicaciones de escritorio en C# o VB.NET.

Delphi: Tiene un generador de pantallas visuales que permite crear aplicaciones de escritorio en Pascal rápidamente.

Qt Designer: Una herramienta para diseñar interfaces gráficas en C++ utilizando la librería Qt.

Ejemplo de uso:

En un generador de pantallas, puedes arrastrar un botón y un cuadro de texto sobre una "pantalla" en blanco. Luego, puedes asociar eventos, como el clic del botón, a una acción (por ejemplo, mostrar el texto escrito en el cuadro de texto). La herramienta genera el código necesario automáticamente.

2. Generadores de informes

Los generadores de informes son herramientas que crean informes automáticamente a partir de datos, sin programar la lógica desde cero. Permiten conectar la base de datos de una aplicación y diseñar informes visuales.

Ofrecen interfaces visuales para elegir los campos de datos y el formato de visualización.

Ejemplos de generadores de informes:

Crystal Reports: Una de las herramientas más populares para la creación de informes complejos a partir de bases de datos. Permite diseñar informes con diferentes tipos de visualización (tablas, gráficos, etc.).

JasperReports: Una herramienta basada en Java que permite crear informes en diferentes formatos (PDF, HTML, Excel).

SQL Server Reporting Services (SSRS): una plataforma de Microsoft para crear, administrar y entregar informes interactivos y basados en datos a los usuarios.

Ejemplo de uso:

Si tienes una base de datos con una tabla de ventas, puedes usar un generador de informes como Crystal Reports para crear un informe que muestre las ventas totales por cada vendedor, de forma visual y estructurada.

3. Generadores de consultas

Un generador de consultas ayuda a crear consultas SQL sin escribir manualmente. Permite seleccionar visualmente tablas, campos y condiciones, generando automáticamente el código SQL. Es útil para quienes no tienen experiencia en SQL o necesitan generar consultas complejas rápidamente.

Ejemplos de generadores de consultas:

SQL Server Management Studio (SSMS) : La herramienta de Microsoft para trabajar con SQL Server incluye un generador de consultas visual que permite crear consultas sin escribir código manualmente.

Toad para MySQL/Oracle: Herramientas que permiten generar consultas SQL de manera visual.

MySQL Workbench: Ofrece un generador de consultas SQL visual para bases de datos MySQL.

Ejemplo de uso:

Si tienes una base de datos de empleados y deseas obtener todos los empleados que trabajan en el departamento de "Ventas", un generador de consultas puede permitirte seleccionar la tabla "empleados", elegir los campos "nombre" y "departamento", y aplicar un filtro con "departamento = 'Ventas'" para generar la consulta SQL automáticamente.

4. Generador de aplicaciones

Un generador de aplicaciones es una herramienta que permite crear aplicaciones completas sin escribir todo el código desde cero. Incluye módulos predefinidos para tareas comunes, facilitando el trabajo a desarrolladores y usuarios sin experiencia técnica gracias a interfaces de "arrastrar y soltar".

Ejemplos de generadores de aplicaciones:

AppGyver: Plataforma para crear aplicaciones móviles sin código (no-code), con una interfaz visual.

OutSystems: Un entorno de desarrollo rápido de aplicaciones que permite crear aplicaciones empresariales a gran escala sin necesidad de escribir mucho código.

FileMaker: Herramienta de creación de aplicaciones empresariales que facilita la creación de soluciones personalizadas, como bases de datos, con poco o ningún código.

Ejemplo de uso:

Usando una herramienta como OutSystems, puedes crear una aplicación de gestión de empleados arrastrando módulos predefinidos, como formularios para agregar empleados, listas para ver todos los empleados, y pantallas para editar la información de los empleados, todo sin necesidad de escribir mucho código.

Beneficios:

- ✓ **Ahorro de tiempo:** Permiten desarrollar aplicaciones y sistemas mucho más rápidos, ya que automatizan tareas complejas o repetitivas.
- ✓ **Facilidad de uso:** Muchas de estas herramientas están diseñadas para que personas con pocos conocimientos de programación puedan usarlas, facilitando la creación de aplicaciones e informes.
- ✓ **Reducción de errores:** Al generar automáticamente el código necesario, se reducen los errores humanos que podrían ocurrir al escribirlo manualmente.
- ✓ **Flexibilidad:** Aunque estas herramientas facilitan el trabajo, aún permiten que los desarrolladores agreguen código personalizado cuando sea necesario para adaptar las soluciones a necesidades específicas.

(AWS, 2024) - (Euroinnova, 2025)

Generación y desarrollo de aplicaciones con herramientas CASE:

Las herramientas CASE (Computer-Aided Software Engineering) son software que ayudan en el desarrollo de aplicaciones de software, mejorando el ciclo de vida desde la planificación hasta el mantenimiento. Se explicará la generación y desarrollo de aplicaciones con estas herramientas, destacando características, estructura, generación de código, documentación y más.

1. Características de las Herramientas CASE

Las herramientas CASE tienen características diseñadas para optimizar y hacer más eficiente el proceso de desarrollo de software. Algunas de sus características más comunes incluyen:

Automatización de tareas repetitivas: Ayudan a automatizar muchas de las tareas involucradas en el desarrollo, como la creación de diagramas, la generación de código o la creación de documentación.

Soporte para Modelado: Ofrecen herramientas para diseñar diagramas de flujo, diagramas de casos de uso, diagramas de clases, diagramas de actividad y otros modelos que representan el sistema a desarrollar.

Generación de código: Pueden generar automáticamente código a partir de los modelos desarrollados, reduciendo el tiempo y los errores humanos al escribir código manualmente.

Manejo de la documentación: Ayudan a generar documentación del software y proporcionar informes sobre el desarrollo y el estado del proyecto.

Control de versiones y seguimiento de cambios: Muchas herramientas CASE permiten gestionar versiones del proyecto, lo que facilita la actualización y el mantenimiento de los sistemas.

2. Estructura de las Herramientas CASE

Las herramientas CASE están divididas en varios módulos, cada uno para una fase del ciclo de vida del software. Incluyen:

- Módulo de análisis y diseño: crea modelos del sistema.
- Módulo de generación de código: genera código fuente automáticamente.
- Módulo de pruebas: gestiona pruebas automatizadas.
- Módulo de documentación: genera documentación técnica.
- Módulo de gestión de proyectos: planifica y sigue actividades del proyecto.

3. Actualización y Mantenimiento del Proyecto

Las herramientas CASE permiten actualizar y mantener el software a lo largo de su ciclo de vida. Algunas de las funcionalidades relacionadas con la actualización incluyen:

Refactorización automática: Permite actualizar y mejorar el código de manera automática sin alterar su funcionalidad.

Control de versiones: Las herramientas CASE suelen incluir un sistema para gestionar las versiones del software y realizar un seguimiento de los cambios a lo largo del tiempo.

Mantenimiento de modelos: Si se realizan cambios en los requisitos o el diseño, las herramientas CASE permiten actualizar los diagramas y modelos, lo que actualiza automáticamente la documentación y el código correspondiente.

4. Generación de Código

La generación de código es una de las funcionalidades clave de las herramientas CASE. Estas herramientas permiten generar código de forma automática a partir de los modelos creados, lo que ayuda a reducir la cantidad de código repetitivo que debe escribirse manualmente. Esto también ayuda a garantizar que el código sea consistente y esté alineado con los requisitos del sistema.

Generación desde diagramas: Por ejemplo, si se crea un diagrama de clases, la herramienta CASE puede generar las clases correspondientes en el lenguaje de programación deseado (por ejemplo, C++, Java, Python).

Generación de código de bases de datos: También pueden generar código para la creación de bases de datos y la manipulación de datos (por ejemplo, sentencias SQL).

5. Documentación generada

Las herramientas CASE también son esenciales para la creación y mantenimiento de la documentación de software. Pueden generar documentación técnica y de usuario de forma automática y actualizada en función del estado del proyecto.

Documentación técnica: A partir de los diagramas y el código generado, las herramientas CASE pueden crear documentación sobre el diseño, las clases, los métodos, las relaciones entre componentes y las decisiones arquitectónicas.

Manuales de usuario: Las herramientas CASE también pueden generar manuales de usuario para explicar cómo interactuar con el software, con instrucciones sobre cómo usar las funcionalidades.

Manuales de programador: Además de los manuales de usuario, pueden generar documentación dirigida a los programadores, que describe la estructura del código, las clases y métodos, y cómo se puede modificar o extender el software.

6. Tutoriales y Reportes de Desarrollo

Las herramientas CASE suelen incluir tutoriales integrados e informes de desarrollo que ayudan a los desarrolladores a entender cómo usar la herramienta ya seguir el progreso del proyecto. Algunos ejemplos hijo:

Tutoriales interactivos: Proporcionan guías paso a paso para aprender a usar la herramienta, lo que puede ser útil para los nuevos usuarios.

Reportes de desarrollo: Incluyen información sobre el estado del proyecto, los componentes desarrollados, las tareas completadas, las pruebas realizadas, etc. Estos informes ayudan en la gestión del proyecto y permiten a los desarrolladores y líderes de equipo mantenerse al tanto de los avances.

7. Ventajas de Usar Herramientas CASE

Aumento de la productividad: La automatización de tareas repetitivas como la generación de código y la documentación reducen significativamente el tiempo de desarrollo.

Mejora de la calidad: Al generar código automáticamente desde modelos bien definidos, las herramientas CASE reducen la probabilidad de errores humanos en la codificación.

Facilita la colaboración: Proporcionan una representación visual de la arquitectura y los componentes del sistema, lo que facilita la colaboración entre los miembros del equipo de desarrollo.

Mejora la gestión del proyecto: Al integrar la planificación, el diseño y la gestión de versiones en una sola herramienta, los equipos pueden llevar un control más efectivo del proyecto.

Ejemplos de herramientas CASE:

Algunas de las herramientas CASE más utilizadas en el desarrollo de software incluyen:

IBM Rational Rose: una herramienta de modelado que permite crear diagramas UML y generar código.

Enterprise Architect: Utilizada para el modelado y diseño de sistemas, incluye funciones de generación de código, pruebas y documentación.

Visual Paradigm: Ofrece soporte para el modelado UML, la generación de código y la creación de diagramas de bases de datos.

PowerDesigner: Enfocada en el diseño de bases de datos y la ingeniería de software.

(Chicaiza, 2014)

(Antonioj20, 2016)