



Especialidad: Técnico en Programación

Módulo II. Submódulo2

Aplica la metodología de desarrollo rápido de aplicaciones con programación orientada a eventos.

Prof: Hilda Lucía Rodríguez Gómez

METODOLOGÍA RAD

El desarrollo rápido de aplicaciones o RAD (acrónimo en inglés de rapid application development) es un proceso de desarrollo de software, desarrollado inicialmente por James Martin en 1980. Se centra más en proyectos de software en curso y comentarios de los usuarios y menos en seguir un plan estricto. Por tanto, prioriza la creación rápida de prototipos sobre la planificación costosa.

FASES DEL RAD

Modelado de gestión: el flujo de información entre las funciones de gestión se modela de forma que responda a las siguientes preguntas: ¿Qué información conduce el proceso de gestión? ¿Qué información se genera? ¿Quién la genera? ¿A dónde va la información? ¿Quién la procesa?

Modelado de datos: el flujo de información definido como parte de la fase de modelado de gestión se refina como un conjunto de objetos de datos necesarios para apoyar la empresa. Se definen las características (llamadas atributos) de cada uno de los objetos y las relaciones entre estos objetos.

Modelado de proceso: los objetos de datos definidos en la fase de modelado de datos quedan transformados para lograr el flujo de información necesario para implementar una función de gestión. Las descripciones del proceso se crean para añadir, modificar, suprimir, o recuperar un objeto de datos. Es la comunicación entre los objetos.

Generación de aplicaciones: El RAD asume la utilización de técnicas de cuarta generación. En lugar de crear software con lenguajes de programación de tercera generación, el proceso RAD trabaja para volver a utilizar componentes de programas ya existentes (cuando es posible) o a crear componentes reutilizables (cuando sea necesario). En todos los casos se utilizan herramientas automáticas para facilitar la construcción del software.

Pruebas de entrega: Como el proceso RAD enfatiza la reutilización, ya se han comprobado muchos de los componentes de los programas. Esto reduce tiempo de pruebas. Sin embargo, se deben probar todos los componentes nuevos y se deben ejercitar todas las interfaces a fondo.





Ventajas y Desventajas RAD

Ventajas:

- Velocidad: Con RAD, es más probable que los proyectos terminen a tiempo y para satisfacción del cliente en el momento de la entrega.
- Costo: Con el desarrollo rápido de aplicaciones, los desarrolladores crean los sistemas exactos que requiere el cliente, y nada más.
- En general:
 - o El desarrollo se realiza a un nivel de abstracción mayor.
 - o Visibilidad temprana.
 - o Menor codificación manual.
 - o Posiblemente menos fallas.
 - o Posiblemente menor costo.
 - o Ciclos de desarrollo más pequeños.
 - o Tiene mayor probabilidad de dejar "a gusto" al cliente ya que se pueden generar con mayor facilidad y rapidez aplicaciones prototipo con una GUI vistosa y lista para su uso prematuro, con una diferencia de tiempo de entrega abismal a software desarrollado con otros modelos de desarrollo enfocados en la robustez del sistema.

Desventajas:

- Escala: Las prácticas de RAD se complican cuando se expanden más allá de un solo equipo o requieren comunicación entre equipos.
- Compromiso: El ciclo frecuente de prototipos RAD requiere que los desarrolladores y clientes se comprometan a reuniones frecuentes que, al principio, pueden parecer consumir ciclos innecesarios para ambas partes.
- Enfoque de interfaz: La metodología RAD motiva a los desarrolladores a encontrar la solución perfecta para el cliente. El cliente juzga la calidad de la solución en su interacción y, a menudo, todo con lo que interactúa es una fachada.
- Como consecuencia, algunos desarrolladores se enfocan menos en las prácticas back-end para acelerar el desarrollo del prototipo enfocado en el front-end.
- Y en general:
 - o Progreso más difícil de medir.
 - o Más fallas (por síndrome de "codificar a lo bestia").

¿QUÉ ES LA PROGRAMACIÓN ORIENTADA A EVENTOS?

La programación dirigida por eventos es un paradigma de programación en el que el flujo del programa está determinado por eventos o mensajes desde otros programas o hilos de ejecución. En este tipo de programación, se deben comprender los siguientes términos: Eventos, Propiedades y Métodos.

¿Qué son los eventos? son las acciones sobre el programa, como, por ejemplo:

- Clic sobre un botón
- Doble clic sobre el nombre de un fichero para abrirlo
- Arrastrar un icono
- Pulsar una tecla o una combinación de teclas
- Elegir una opción de un menú
- Escribir en una caja de texto
- Simplemente mover el mouse

En este paradigma de programación, las aplicaciones reaccionan a eventos, esto permite al usuario realizar una serie de acciones. Cuando se produce un evento (por ejemplo, hacer clic en un botón, arrastrar el dedo o tocar la pantalla) la aplicación reacciona llamando a una secuencia de instrucciones programadas por el programador para ese evento concreto, como establecer el color de fondo de un botón a azul o cambiar el texto de una etiqueta.



Los eventos pueden ser divididos en 2 tipos diferentes:

- Automáticos como al abrir una ventana.
- Iniciados por el usuario como hacer clic en un botón, tocar o arrastrar en la pantalla, inclinar el teléfono, etc.

¿Qué es una propiedad? son características que definen un objeto, por ejemplo, el tamaño de la letra de un control, el tipo de letra, la alineación, etc. Se puede asignar en tiempo de diseño usando la ventana Propiedades y/o en tiempo de ejecución en el momento de programar el control de objetos.

¿Qué es un método?

Un método es una función o procedimiento, que se llama desde el programa, son pequeños módulos de código que se diseñan con propósitos específicos, los métodos realizan tareas particulares y específicas. Los métodos solo pueden ser **ejecutados** en tiempo de ejecución no en tiempo de diseño. Algunos ejemplos de métodos son: En una ventana, el método MOVE, que mueve un formulario en un espacio de dos dimensiones en la pantalla, SetFocus, LostFocus, etc...

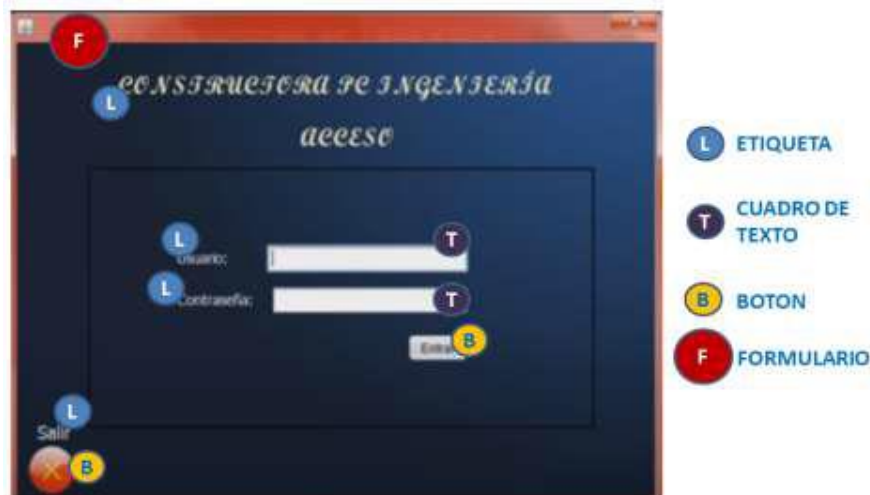
SOBRE LA INTERFAZ:

El formulario es el primer objeto o control que se visualiza y constituye la pantalla o ventana sobre la que se colocan otros objetos o controles como etiquetas, controles de texto, botones, etc. y por supuesto el código necesario de nuestros programas, por lo tanto, constituirán la Interfaz de usuario.

Un botón es un componente en el que el usuario selecciona para desencadenar cierta acción. Una aplicación de Java puede utilizar varios tipos de botones, incluyendo botones de comando, casillas de verificación, botones interruptores y botones de opción.

Las etiquetas (Label) proporcionan una forma de colocar texto estático en una ventana gráfica, para mostrar información que no varía, normalmente se limita a presentar textos en pantalla.

Los cuadros de texto son elementos gráficos en donde el usuario puede escribir información en la aplicación. Escribe datos sensibles que permitirán alimentar datos para ejecutar o resolver, o controlar determinados propósitos del programa. El componente de texto más usado es TextField el cual puede mostrar varias líneas, ajustar el texto al tamaño del control y agregar formato básico.



En el mundo del desarrollo de las aplicaciones orientadas a eventos. La aplicación, primero debe de funcionar y siguiente elemento a valorar es cuidar la estética, es decir que tenga armonía en el diseño de la interfaz.