

## **“PORTAFOLIO DE EVIDENCIAS”**

# **APLICA LA METODOLOGIA DE DESARROLLO RAPIDO DE APLICACIONES CON PROGRAMACION ORIENTADA A EVENTOS**

**Profesor: L.I. Mario Lagunas Brito**

**NOMBRE: Luis Eduardo Bahena Castillo**

**GRADO Y GRUPO: 3BPM**

**Nº DE LISTA: #2**

**FECHA: VIERNES 20 DE SEPTIEMBRE DEL 2019**

**CORREO ELECTRONICO:**

**[luiseduardobahenacastillo007@gmail.com](mailto:luiseduardobahenacastillo007@gmail.com)**



# INDICE

**PORTADA.....**

**INDICE.....**

**PROTOCOLO DE ARRANQUE.....**

- Descripción General Del Módulo Profesional. Técnico En Programación
- Justificación De La Carrera
- Duración Del Módulo Profesional
- Submódulos Que Lo Integran
- Materiales Y Equipo
- Bibliografía Sugerida
- Criterios De Evaluación
- Reglas del Curso
- Competencias Profesionales a Desarrollar
- Contenido Temático

**EVALUACION DIAGNOSTICA.....**

**METODOLOGIAS DE PROGRAMACION.....**

- Otras definiciones
- Objetivos
- Elementos de una Metodología
- Actividades y Tareas
- Cronología de las Metodologías de la Programación
- Enfoques de Desarrollo de Software

**METODOLOGIA DE DESARROLLO RAD.....**

- Qué es?
- Autor
- Características
- El Entorno/Herramientas del RAD
- Modelo RAD de cuatro etapas
- El modelo RAD de cinco etapas
- Ventajas-Desventajas
- Conclusiones

**METODOLOGIA DE PROGRAMACION EN ESPIRAL.....**

**PROGRAMACIÓN ORIENTADA A OBJETOS**

**EJERCICIOS.....**

**CONCEPTOS DE PROGRAMACION.....**

**ANTECEDENTES DE VISUAL BASIC.....**

**EJERCICIOS.....**

**ABREVIATURAS PARA NOMBRES DE OBJETOS.....**

**COMENTARIOS PERSONALES.....**

**BIBLIOGRAFIA.....**

## Protocolo de Arranque

### Horario

Lunes (3)	07:00 – 09:30 (CA)
Miercoles (2)	08:40 – 10:40 (S20)
Viernes (3)	07:00 – 09:30 (CA)

## Descripción General Del Módulo Profesional Técnico En Programación

### Módulos Profesionales que comprende la carrera de Técnico en Programación

<b>Módulo I</b>	<b>Desarrolla software de aplicación con programación estructurada</b> Submódulo 1 - Construye algoritmos para la solución de problemas Submódulo 2 - Aplica estructuras de control con un lenguaje de programación Submódulo 3 - Aplica estructuras de datos con un lenguaje de programación
<b>Módulo II</b>	<b>Aplica metodologías de desarrollo de software con herramientas de programación visual</b> Submódulo 1 - Aplica la metodología espiral con programación orientada a objetos Submódulo 2 - Aplica la metodología de desarrollo rápido de aplicaciones con programación orientada a eventos
<b>Módulo III</b>	<b>Desarrolla aplicaciones Web</b> Submódulo 1 - Construye páginas Web Submódulo 2 - Desarrolla aplicaciones que se ejecutan en el cliente Submódulo 3 - Desarrolla aplicaciones que se ejecutan en el servidor
<b>Módulo IV</b>	<b>Desarrolla software de aplicación Web con almacenamiento persistente de datos</b> Submódulo 1 - Construye bases de datos para aplicaciones Web Submódulo 2 - Desarrolla aplicaciones Web con conexión a bases de datos
<b>Módulo V</b>	<b>Desarrolla aplicaciones para dispositivos móviles</b> Submódulo 1 - Desarrolla aplicaciones móviles para Android Submódulo 2 - Desarrolla aplicaciones móviles para IOS

## Justificación De La Carrera

En el contexto nacional la formación de Técnicos en: Programación es relevante porque: contribuye a la formación de personas capaces de integrarse a un mercado laboral dinámico y de alta demanda, que esta a la vanguardia en el uso de la tecnología y que contribuye a la transformación digital de los sectores productivos en el país.

La carrera de Técnico en Programación desarrolla en el estudiante las siguientes Competencias Profesionales:

- Desarrolla software de aplicación con programación estructurada
- Aplica metodologías de desarrollo de software utilizando herramientas de programación visual
- Desarrolla aplicaciones Web
- Desarrolla software de aplicación Web con almacenamiento persistente de datos
- Desarrolla aplicaciones para dispositivos móviles

Permite al egresado, a través de la articulación de saberes de diversos campos, realizar actividades dirigidas al desarrollo de software de aplicación en plataformas de escritorio, Web y móviles, utilizando los paradigmas de programación estructurada, programación orientada a objetos y programación orientada a eventos, aplicando metodologías de la Ingeniería del Software.

## Duración Del Módulo Profesional:

272 horas

Semestrales

Abarcando 14 semanas

del 26 de Agosto al 30 de Noviembre

## Submódulos Que Lo Integran:

No.	Submódulo	Horas (Semestre - Semana)
1.	Aplica Metodología Espiral Con Programación Orientada a Objetos	144-9
2.	Aplica Metodología de Desarrollo Rápido de Aplicaciones Con Programación Orientada a Eventos	128 - 8

## Materiales Y Equipo

- ✓ Equipo de cómputo y proyector
- ✓ Software de Programación Visual y Orientado a Eventos Microsoft Visual Basic
- ✓ Medios de almacenamiento (USB)
- ✓ Bibliografía
- ✓ Fuentes electrónicas

## Bibliografía Sugerida

- ✓ *Desarrollo de software ágil. (2ª. Edición).* México: IT Campus Academy. Lainez, J. 2015.
- ✓ *Análisis y Diseño de Sistemas.* 8ª. Edición. México: Pearson Prentice Hall. Kendall, K. E. 2011.
- ✓ *Curso de Programación Orientada a Objetos en C#.NET. Ejemplos con aplicaciones visuales y de consola.* México: Alfaomega. Bruno, T. 2016.
- ✓ *Programación Orientada a Objetos y Estructura de Datos a Fondo.* México: Alfaomega. Sznajdleder, P. 2017.
- ✓ *Programación en C# para principiantes.* México: Alfaomega. Troy, D. 2016.
- ✓ *Aprenda Microsoft Visual Basic Ya!.* México: Alfaomega. Gates, W. 2014.
- ✓ *Ingeniería de Software .* México: Trillas. Pressman, R. 2015.

## Criterios De Evaluación

- ✓ Saber declarativo (35%)
- ✓ Saber procedimental (55%)
- ✓ Saber actitudinal (10%)

## **Desglose**

- ✓ *Saber declarativo (35%)*
  - Examen Teórico
  - Firmas en cuaderno
  - Trabajo y tareas
  - Portafolio de evidencias
- ✓ *Saber procedimental (55%)*
  - Examen Práctico
  - Prácticas de laboratorio
- ✓ *Criterios de Evaluación*
  - Saber actitudinal (10%)
  - Asistencia (80% mínimo)
  - Cumplimiento de trabajos
  - Respeto a las reglas del curso

## **Reglas del Curso**

- Conducirnos con respeto
- Uso de celular solo para recibir llamadas urgentes
- Presentarse debidamente uniformado y con credencial del plantel
- Cumplimiento con las actividades en tiempo y forma
- Honestidad en el trabajo
- No música en clase

## **Competencias Profesionales a Desarrollar**

- Identifica los elementos del paradigma de la programación orientada a eventos.
- Aplica el modelo de desarrollo rápido de aplicaciones (RAD) para el desarrollo de software.

## **Contenido Temático**

### **UNIDAD 1. Metodologías de la programación**

- Introducción a las Metodologías de la Programación
- Metodología RAD

### **UNIDAD 2. Conceptos básicos de la programación basada en objetos y orientada a eventos**

- Controles
- Objetos

- Propiedades
- Métodos
- Eventos

**UNIDAD 3. Introducción a Microsoft Visual Basic**

- Instalación
- Entorno
- Antecedentes
- Reglas
- Palabras reservadas
- Operadores
- Tipos de datos
- Componentes pantalla principal
- Estructura de un programa en Visual Basic

**UNIDAD 4. Programación de Eventos en Visual Basic**

- Controles (propiedades, eventos y métodos)
- Entrada-proceso- salida
- Sentencias condicionales, selectivas y repetitivas

**UNIDAD 5. Programación de Eventos Avanzados en Visual Basic**

- Controles avanzados
  - Propiedades
  - Eventos
  - Métodos
- Conexión a bases de datos

## Evaluación Diagnóstica

1. *¿Qué es una variable?* Es un espacio almacenado donde se guarda y se recupera los datos que se utilizan en un programa
2. *¿Qué es una constante?* Es un valor que no puede ser modificado dentro de un programa
3. *¿Es lo mismo dato que información? ¿Cuál es la diferencia?* No, que los datos son representaciones simbólicas de una determinada variable y la información es el conjunto de datos almacenados y organizados que funcionan como instrucción
4. *¿Cuáles lenguajes de programación conoces?* Dos: algoritmo de Pseudocódigo y Lenguaje C
5. *¿Qué es una app, un programa y un sistema de información?*

App: son programas que se instalan en el dispositivo móvil y que realizan funciones concretas de ámbito personal y profesional

## APLICA LA METODOLOGIA DE DESARROLLO RAPIDO DE APLICACIONES CON POE – PRIMER PARCIAL

Programa: es un conjunto de instrucciones u ordenes basadas en un lenguaje de programación para resolver un problema o una función especial

Sistema de Informacion: son conjunto de datos entre si con un fin común que ayudan a administran, recolectar, recolectar, procesar, almacenar, y distribuir la informacion

6. *¿Cuáles son las unidades de medida de la información digital y sus equivalencias?* El byte (8 bits), kilobyte (1024 bytes), megabyte (1024 kbytes), gigabyte (1024 mbytes), terabyte (1024 gbytes), petabyte (1024 tbytes), exabyte (1024 pbytes), zettabyte (1024 ebytes), yotta (1024 zbytes)
7. *¿Para qué sirve la extensión de un nombre de archivo, y cuáles conoces?* Sirve para distinguir el contenido de un archivo para el sistema operativo disponga el procedimiento para poderlo ejecutarlo

.pdf , .mp3, .pptx, .docx, .html

8. *¿Para qué sirven el disco duro y la memoria RAM de la computadora?*

Disco Duro: es el elemento en el que se almacenan todos los datos que introducimos en el ordenar tanto los programas como los archivos

Memoria RAM: es el lugar donde los ordenadores guardan los datos que se están utilizando en tiempo real y sólo se utiliza con el ordenador encendido

9. *¿Qué significan las siglas DVD, CD, USB, RAM, ROM, HD, VGA, CPU, WWW, HTTP, IT, MSDOS, PDF ?*

DVD: disco versátil digital                      RAM: memoria de acceso aleatorio      HD: alta definicion

CD: disco compacto regrabable              ROM: memoria de solo lectura              VGA: matriz de graficos de video

USB: se transfiere datos e informacion para almacenarlos      WWW: red informática mundial

HTTP: protocolo de transferencia de hipertexto      IT: Tecnologia Informativa

MSDOS: Sistema operativo de disco de Microsoft      PDF: Formato de Documento Portátil

10. *¿Qué son el hardware y el software?*

Hardware: Conjunto de elementos físicos o materiales que constituyen una computadora o un sistema informático.

Software: Conjunto de programas y rutinas que permiten a la computadora realizar determinadas tareas.

11. *¿Qué significan Entrada, Proceso y Salida en la programación?* Cada línea de proceso en “Velneo” se comporta como un proceso completo y dispone de un origen (entrada), comando de instrucción (proceso) y destino (salida).

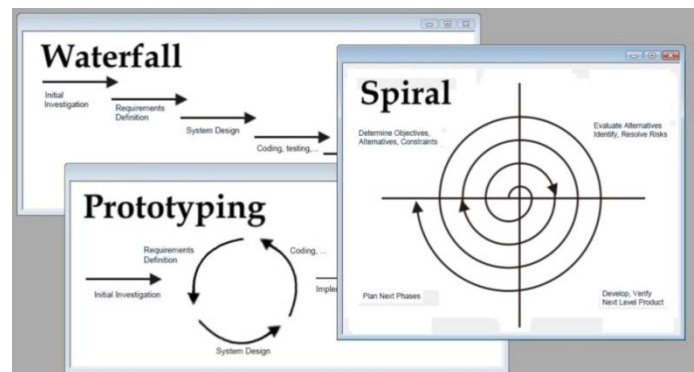
12. *¿ Cuáles tipos de decisión existen en la programación?* SI-ENTONCES, SI-NO , SEGUN

## APLICA LA METODOLOGIA DE DESARROLLO RAPIDO DE APLICACIONES CON POE – PRIMER PARCIAL

13. ¿Qué son los tipos de datos y cuáles son los del lenguaje C? En programación, un tipo de dato informático o simplemente tipo es un atributo de los datos que indica al ordenador (y/o al programador) sobre la clase de datos que se va a trabajar. Esto incluye imponer restricciones en los datos, como qué valores pueden tomar y qué operaciones se pueden realizar. **INT, FLOAT, PRINTF, DOUBLE, CHAR, BOOLEAN, LONG**
14. ¿Cuáles son tus expectativas para esta materia? Pues poner aprender a programar en un lenguaje avanzado y poderlo practicarlo en la vida cotidiana

## Metodologías de Programación

- Es un conjunto de procedimientos, técnicas, estrategias y ayudas para la documentación en el desarrollo de un producto de software.
- Hacer del desarrollo de software un proceso formal, con resultados, que permitan obtener un producto de alta calidad, que satisfaga las necesidades y expectativas del cliente.
- Debe omitirse el modo de trabajar artesanal, que casi siempre requiere invertir mayores esfuerzos para obtener el resultado final, y seguramente desfases de fechas y costos, lo que llevaría al desgaste personal del equipo de proyecto.
- Así, podremos resumir que utilizar las metodologías implica mejoras en los procesos de desarrollo, en el producto y en la satisfacción del cliente.



## Otras definiciones

- 1) Conjunto de pasos y procedimientos que deben seguirse para el desarrollo de software.
- 2) Conjunto de filosofías, fases, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para los desarrolladores de SI.
- 3) Conjunto de procedimientos, técnicas, herramientas y soporte documental que ayuda a los desarrolladores a realizar nuevo software.

Por tanto, una metodología representa el camino a seguir para desarrollar software de manera sistemática.

## Objetivos:

- ✓ Mejores Aplicaciones.
- ✓ Un mejor Proceso de Desarrollo que identifique salidas (o productos intermedios) de cada fase de forma que se pueda planificar y controlar los proyectos.
- ✓ Un Proceso Estándar en la organización.
- ✓ Metodologías de Programación
- ✓ Metodologías de Programación

## Elementos de una Metodología:



## Actividades y Tareas

El Proceso se descompone hasta el nivel de Actividades y Tareas (actividades elementales)

### Procedimientos

- ✓ Definen la forma de llevar a cabo las Tareas
- ✓ Vínculo de Comunicación entre Usuarios y Desarrolladores

### Productos

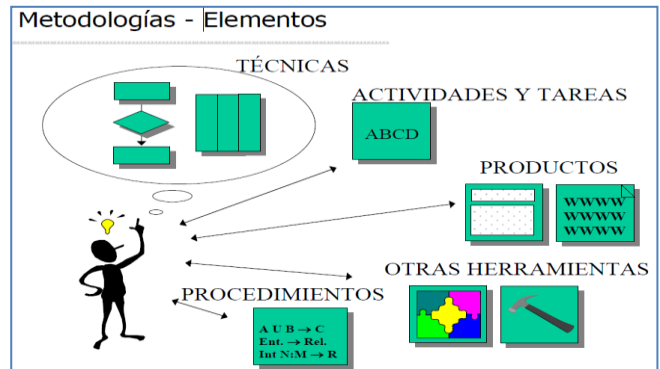
- ✓ Obtenidos como resultado de seguir un Procedimiento
- ✓ Pueden ser Intermedios o Finales

### Técnicas

- ✓ Se utilizan para aplicar un **Procedimiento**
- ✓ Pueden ser Gráficas y/o Textuales
- ✓ Determinan el formato de los Productos resultantes en cada Tarea

### Herramientas Software

- ✓ Proporcionan soporte a la aplicación de las Técnicas



## Cronología de las Metodologías de la Programación

### 1970

- ✓ Programación estructurada sol desde 1969
- ✓ Programación estructurada Jackson desde 1975

### 1980

- ✓ Structured Systems Analysis and Design
- ✓ Methodology (SSADM) desde 1980
- ✓ Structured Analysis and Design Technique (SADT) desde 1980
- ✓ Ingeniería de la información (IE/IEM) desde 1981
- ✓ Metodologías de Programación

### 1990

- ✓ *Rapid Application Development* (RAD) desde 1991.
- ✓ *Object Oriented Programming* (OOP) a lo largo de la década de los 90's
- ✓ *Virtual Finite State Machine* (VFSM) en los 90's
- ✓ *Dynamic Systems Development Method* desarrollado en Reino Unido desde 1995.
- ✓ *Scrum* (desarrollo), en la última parte de los 90's
- ✓ *Rational Unified Process* (RUP) desde 1999.
- ✓ *Extreme Programming* (XP) desde 1999

### Nuevo milenio

- ✓ *Enterprise Unified Process* (EUP) extensiones RUP desde 2002
- ✓ *Constructionist Design Methodology* (CDM) desde 2004
- ✓ *Agile Unified Process* (AUP) desde 2005 por Scott Ambler

### Timeline of Methodologies

1950s	Code & Fix
1960s	Design-Code-Test-Maintain
1970s	Waterfall Model
1980s	Spiral Model
1990s	Rapid Application Development, V Model
2000s	Agile Methods

## Enfoques de Desarrollo de Software

Cada metodología de desarrollo de software tiene más o menos su propio enfoque para el desarrollo de software.

Estos son los enfoques más generales, que se desarrollan en varias metodologías específicas:

- ✓ **Cascada:** Framework lineal.
- ✓ **Prototipado:** Framework iterativo.
- ✓ **Incremental:** Combinación de framework lineal e iterativo.
- ✓ **Espiral:** Combinación de framework lineal e iterativo.
- ✓ **RAD:** *Rapid Application Development*, framework iterativo.

## METODOLOGIA DE DESARROLLO RAD

### Rapid Application Development



### Qué es?

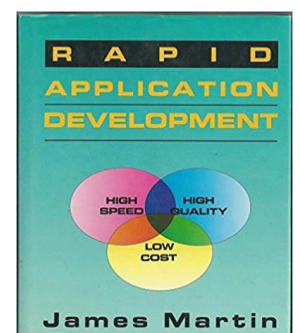
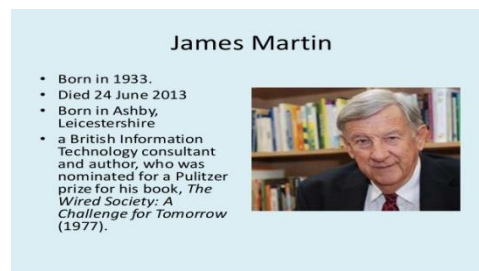
La metodología de desarrollo conocida como Diseño Rápido de Aplicaciones RAD (por sus siglas en inglés) ha tomado gran auge debido a la necesidad que tienen las instituciones de crear aplicaciones funcionales en un plazo de tiempo corto.

RAD es un ciclo de desarrollo diseñado para crear aplicaciones de computadoras de alta calidad de las que acontecen en corporaciones grandes.



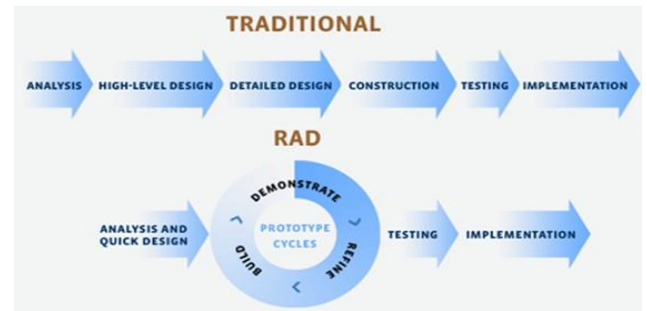
### Autor

Desarrollado inicialmente por James Martin en 1980.

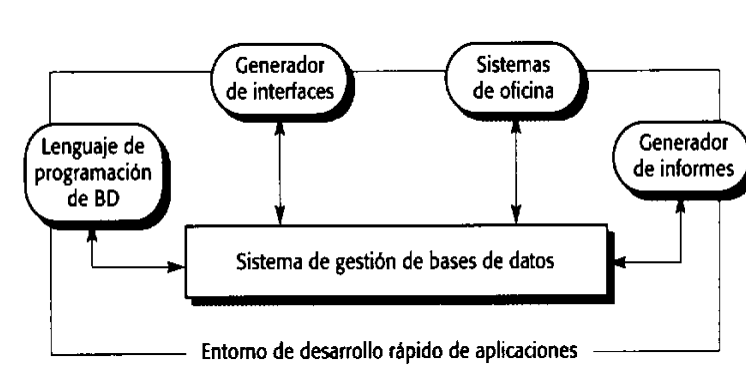


## Características

- El método comprende el desarrollo iterativo, la construcción de prototipos y el uso de utilidades CASE (Computer Aided Software Engineering).
- Permite crear sistemas funcionales (utilizables) en poco tiempo, entre 60 y 90 días.
- Tradicionalmente, el desarrollo rápido de aplicaciones tiende a englobar también la usabilidad, utilidad y la rapidez de ejecución.
- El usuario puede ver prontamente un prototipo del software.
- RAD da prioridad a la fecha de entrega, si el proyecto empieza a aplazarse, se ajustan los requisitos para no aplazar la fecha de entrega.
- Descompone el sistema en componentes, para que así cada uno sea abordado por un equipo distinto, de esta manera mas tarde se integran todos los componentes en la solución final.



## El Entorno/Herramientas del RAD



### • **Lenguaje de programación de bases de datos.**

Contiene la estructura de la base de datos e incluye las operaciones básicas de manipulación de bases de datos, utilizando el lenguaje estándar de programación de base de datos: SQL (*Structured Query Language – Lenguaje Estructurado de Consultas*).



Los comandos SQL se pueden introducir directamente o generar de forma automática a partir de formularios rellenos por el usuario.

### • **Generador de interfaces.**

Se utiliza para crear formularios de introducción y visualización de datos. Las pantallas o interfaces deben tener las siguientes características:

#### **a) Definición de formularios interactivos:**

## APLICA LA METODOLOGIA DE DESARROLLO RAPIDO DE APLICACIONES CON POE – PRIMER PARCIAL

Que permitan al desarrollador definir los campos a visualizar y la manera en que éstos deben organizarse.

### b) *Vinculación de los formularios:*

Que permitan al desarrollador especificar que ciertas entradas provocan la visualización de formularios adicionales.

### c) *Verificación de campos.*

Que permitan al desarrollador definir los rangos permitidos para los valores de entrada en los campos de los formularios.

- **Enlaces a aplicaciones de oficina.**

Por ejemplo, enlaces a una hoja de cálculo, para el análisis y manipulación de información numérica; o un procesador de textos para la creación de plantillas de informes.

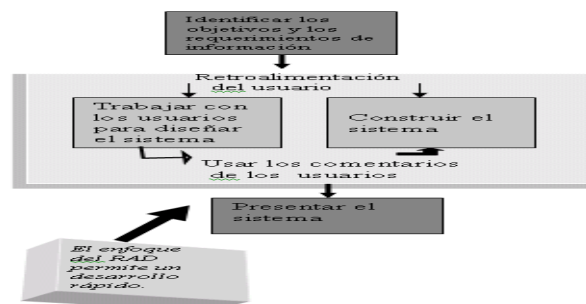
- **Un generador de informes.**

Se utiliza para definir y crear informes a partir de la la base de datos.



## Modelo RAD de cuatro etapas

- Planificación de los requisitos
- Diseño
- Construcción
- Implementación



- **Planificación de los requisitos.**

Esta etapa requiere que usuarios con un vasto conocimiento de los procesos de la compañía determinen cuales serán las funciones del sistema. Debe darse una discusión estructurada sobre los problemas de la compañía que necesitan solución.



- **Diseño.**

Consiste de un análisis detallado de las actividades de la compañía relación al sistema propuesto. Los usuarios participan activamente talleres bajo la tutela de profesionales de la informática. En ellos descomponen funciones y definen entidades asociadas con el sistema. Una vez se completa el análisis se crean los diagramas definen las alteraciones entre los procesos y los datos.



en  
en  
que

- **Construcción.**



En construcción

El equipo de desarrolladores trabajando de cerca con los usuarios finalizan el diseño y la construcción del sistema. La construcción de la aplicación consiste de una serie de pasos donde los usuarios tienen la oportunidad de afirmar los requisitos y repasar los resultados.

- **Implementación.**



## APLICA LA METODOLOGIA DE DESARROLLO RAPIDO DE APLICACIONES CON POE – PRIMER PARCIAL

Esta etapa envuelve la implementación del nuevo producto y el manejo del cambio del viejo al nuevo sistema. Se hacen pruebas al sistema y se adiestran los usuarios.

### El modelo RAD de cinco etapas

El Modelo RAD de cinco etapas comprende las siguientes:

- Modelado de gestión.
- Modelado de datos.
- Modelado del proceso.
- Generación de aplicaciones.
- Pruebas y entrega.

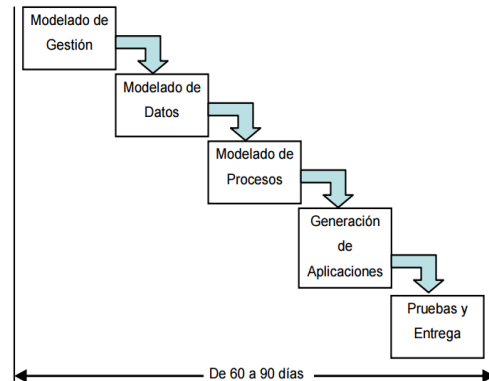


Figura 3 Modelo RAD

- **Modelado de gestión.**

Se modela el flujo de información entre las funciones de gestión. Este modelo se basa en dar respuesta a las siguientes preguntas:

- ¿Qué información conduce el proceso de gestión?
- ¿Qué información genera?
- ¿A dónde va la información?
- ¿Quién la procesa?

- **Modelado de datos.**

En este modelo se definen los almacenes de datos y cómo se relacionan los almacenes entre sí.

- **Modelado del proceso.**

Se utiliza para añadir, modificar, suprimir o recuperar un objeto de datos.

- **Generación de aplicaciones.**

Para esto se utiliza una herramienta de cuarta generación que permite crear el software y facilitar la construcción del programa.

- **Pruebas y entrega.**

El proceso de desarrollo finaliza realizando pruebas de calidad del software diseñado con la herramienta RAD, posteriormente se realiza la implementación de la aplicación

### Ventajas

#### Bajos costos

RAD, por lo general resulta en costos más bajos. Esto se debe a que se forman pequeños equipos de profesionales quienes utilizan herramientas de alta capacidad para generar los sistemas.

Estas herramientas conocidas como ""CASE"" (*Computer-Aided Software Engineering*) permiten que se aligere el proceso, lo cual ayuda a que los costos aún sean más bajos.

#### Calidad

El uso de herramientas "CASE" tiene el propósito de integrar diagramas para representar la información y crear modelos del sistema. Se crean diseños y estructuras bien detalladas. Los diagramas ayudan a visualizar los conceptos.

- Actualmente, muchos sistemas RAD incluyen herramientas de programación visual que permiten desarrollar sistemas de forma interactiva.
- Los sistemas de desarrollo visual (Visual Basic), permiten este enfoque, basado en la reutilización para el desarrollo de aplicaciones.
- Velocidad de desarrollo .
- Visibilidad temprana
- Producto final cumple con las necesidades del cliente
- Mayor involucramiento de los usuarios
- Interfaz gráfica estándar

### Desventajas

- Para proyectos grandes aunque por escalas, el RAD requiere recursos humanos suficientes como para crear el numero correcto de equipos RAD.
- RAD requiere clientes y desarrolladores comprometidos en las rápidas actividades necesarias para completar un sistema en un marco de tiempo abreviado. Si no hay compromiso, por ninguna de las partes constituyentes, los proyectos RAD fracasaran.
- No todos los tipos de aplicaciones son apropiados para RAD. Si un sistema no se puede modularizar adecuadamente, la construcción de los componentes necesarios para RAD será problemático.
- Si los usuarios no saben cómo utilizar las aplicaciones, el aprendizaje puede ser difícil.
- Puede haber problemas de rendimiento en la aplicación por tener que cambiar de una aplicación del sistema a otra.
- Este esfuerzo adicional para realizar el cambio entre aplicaciones depende de la ayuda que brinde el sistema operativo.
- Progreso mas difícil de medir.
- A mayor velocidad, menor eficiencia.
- Riesgo de revertirse a las practicas sin control de antaño.
- Mas fallas (por síndrome de “codificar a lo bestia”)

### Lo que debe incluir un modelado RAD

- Diagramas de flujo de datos
- Modelo de actividades UML
- Diagramas de casos de uso
- Diagramas de interacción de secuencias

### Conclusiones.

Hoy día el uso de la metodología de diseño rápido de aplicaciones ha adquirido mucha popularidad en el campo de la informática.

Es posible asegurar un resultado exitoso si los proyectos se desarrollan para cumplir con un itinerario estricto y sacrificando algún tipo de funcionalidad.

## APLICA LA METODOLOGIA DE DESARROLLO RAPIDO DE APLICACIONES CON POE – PRIMER PARCIAL

RAD resulta un modelo de desarrollo de software muy útil cuando se requiera poner en marcha un sistema en un periodo de tiempo corto.

**Desde el punto de vista del cliente:**

El modelo RAD de desarrollo es idóneo para toda empresa que quiera ver desde antes resultados parciales y funcionales, así como un producto final en un corto tiempo.

**Desde el punto de vista del desarrollador:**

Tiene mayor probabilidad de dejar “a gusto” al cliente ya que se pueden generar con mayor facilidad y rapidez aplicaciones prototipo con una GUI (*Graphical User Interface*) vistosa y listo 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.

# METODOLOGIA DE PROGRAMACION EN ESPIRAL

**SUBMODULO PROFESIONAL: Aplica Metodología de Desarrollo Rápido de Aplicaciones Con Programación Orientada a Eventos**

**PROFESOR: L.I. Mario Lagunas Brito**

INTEGRANTES:

- \* **LUIS EDUARDO BAHENA CASTILLO**
- \* **JUAN ODILON DOMINGUEZ**
- \* **ISRAEL DAVID GARCIA RAMIREZ**
- \* **CARLOS PERALTA TADEO**

# 3BPM

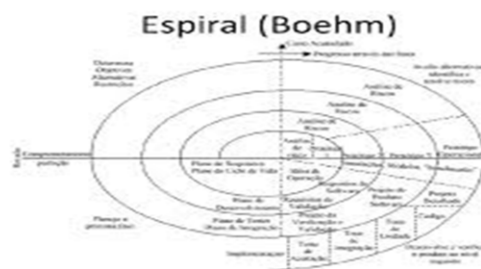




# ¿QUÉ ES?

**El desarrollo en espiral es un modelo de ciclo de vida del software definido por primera vez por Barry Boehm en 1986,<sup>1</sup> utilizado generalmente en la ingeniería de software.**

**Las actividades de este modelo se conforman en una espiral, en la que cada bucle o iteración representa un conjunto de actividades. Las actividades no están fijadas a ninguna prioridad, sino que las siguientes se eligen en función del análisis de riesgo, comenzando por el bucle interior.**



## AUTOR DE LA METODOLOGIA EN ESPIRAL

**Barry Boehm, autor de diversos artículos de ingeniería del software; modelos de estimación de esfuerzo y tiempo que se consume en hacer productos software; y Modelos de Ciclo de Vida; ideó y promulgó un modelo desde un enfoque distinto al tradicional en Cascada: El Modelo Evolutivo Espiral.**





## CARACTERÍSTICAS DEL MODELO EN ESPIRAL PARA EL DESARROLLO DE SOFTWARE

Es considerado como un modelo evolutivo ya que combina el modelo clásico con el diseño de prototipos. \*Contiene una nueva etapa que es el análisis de riesgos, no incluida anteriormente

\*Este modelo es el indicado para desarrollar software con diferentes versiones actualizadas como se hace con los programas modernos de PC's

\*La ingeniería puede desarrollarse a través del ciclo de vida clásico o el de construcción de prototipos

\*Este es el enfoque más realista actualmente

## ETAPAS O FASES DEL MODELO EN ESPIRAL

En cuanto a su ejecución, el modelo en espiral consiste en seguir ciclos crecientes de cuatro fases cada uno, que se van realizando siguiendo una forma de espiral. En cada ciclo se pasa por dichas fases bien definidas, como en el modelo de cascada, pero con capacidad de evolucionar su complejidad con cada ciclo.

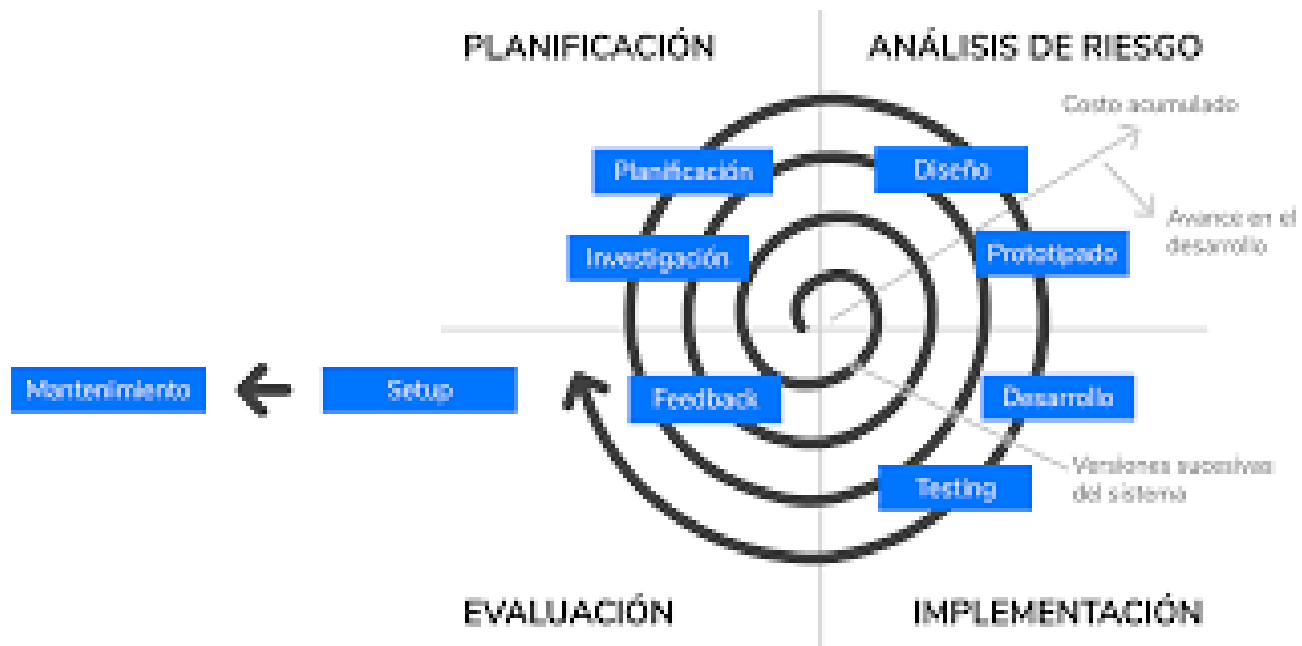
Las fases por las que pasa cada ciclo de la espiral son:

**Planificación.** Se determinan los objetivos y el alcance del ciclo que comienza, tras un necesario ejercicio de investigación. Con cada iteración, se irá incrementando el tamaño de software entregado y la funcionalidad cubierta.

**Análisis de Riesgo.** Se evalúa todo aquello que pueda afectar al proyecto según el estado en que se encuentre y su grado de avance. Para ello, se diseñarán los prototipos que deberán ser validados en el ciclo.

**Implementación.** Se desarrolla y valida el software según el alcance acordado, el cual está íntimamente relacionado y condicionado con el análisis de riesgos anterior.

**Evaluación.** Antes de proceder a realizar otra vuelta en la espiral, se debe prestar atención a lo que sucedió en la vuelta anterior. Se debe analizar en detalle si los riesgos detectados anteriormente ya tuvieron solución. Básicamente, esta fase servirá para determinar el avance del proyecto y dar pistas de hacia dónde debe enfocarse la próxima iteración.



## VENTAJAS Y DESVENTAJAS DEL MODELO ESPIRAL

### VENTAJAS

- \*No requiere una definición completa de los requerimientos del software a desarrollar para comenzar su funcionalidad
- \*En la terminación de un producto desde el final de la primera iteración es muy factible aprobar los requisitos
- \*Sufrir retrasos corre un riesgo menor, por que se comprueban los conflictos presentados tempranamente y existe la forma de poder corregirlos a tiempo

### DESVENTAJAS

- \*Existe complicación cuando se evalúa los riesgos
- \*Se requiere la participación continua por parte del cliente
- \*Se pierde tiempo al volver producir inicialmente una especificación completa de los requerimientos cuando se modifica o mejora el software

# CONCLUSIONES

El modelo en espiral no se ha utilizado tanto como el modelo lineal o secuencial (Cascada) y dio construcción de prototipos, es de gran complejidad y solo es utilizado en sistemas grandes, que si son desarrollados con éxito logran doblar la productividad, utiliza un enfoque evolutivo para la ingeniería de software, permitiendo al desarrollador y al cliente entender y reaccionar a los riesgos en cada nivel evolutivo, utiliza la creación de prototipos como un mecanismo de reducción de riesgo, pero, lo más importante permite a quien lo desarrolla aplicar el enfoque de creación de prototipos en cualquier etapa de la evolución de prototipos, mejora el Modelo de Cascada enfatizando la naturaleza iterativa del proceso de diseño. Eso introduce un ciclo de prototipo iterativo, cada iteración, las nuevas expresiones que son obtenidas transformando otras dadas son examinadas para ver si representan progresos hacia el objetivo.

## PROGRAMACIÓN ORIENTADA A OBJETOS

### Conceptos básicos.



### Conceptualización en el mundo real

## Qué es un objeto?

Pensar en términos de objetos es muy parecido a cómo lo haríamos en la vida real.

Mira a tu alrededor y encontrarás muchos ejemplos de objetos *físicos* del mundo real: mesa, sillas, plumón, cuaderno, laptop, personas, árboles, teléfonos, etc.

También existen objetos intangibles: materias, cursos, control escolar, horarios, etc.

Los objetos comparten dos características: *atributos* y *comportamientos* (métodos).

## Qué es un atributo o propiedad?

Son las características que definen cómo es un objeto. Podríamos decir que son *adjetivos*.

- Una laptop tiene atributos o propiedades: color, marca, procesador, modelo.
- Una persona tiene atributos o propiedades: nombre, edad, peso, talla.
- Un curso tiene atributos o propiedades: nombre, horario, lugar, fecha.

## Qué es un comportamiento o método?

Son las acciones que puede hacer o que podemos hacer con un objeto. Podríamos decir que son *verbos*.

- Una laptop tiene comportamientos: encender, apagar, emitir sonidos, formatear.
- Una persona tiene comportamientos: correr, estudiar, comer.
- Un curso tiene comportamientos: iniciar, formar, plantear, desarrollar.

Identificando atributos y comportamientos de objetos del mundo real es un buen inicio para pensar en términos de la POO.

Ahora identificaremos las clases

## Qué es una clase?

En el mundo real, podemos encontrar frecuentemente muchos objetos individuales que son todos del mismo tipo.

Podemos encontrar muchas laptops que fueron hechas bajo el mismo tipo o modelo.

Cada una de ellas fue construida bajo las mismas especificaciones y contienen los mismos componentes.

En términos de POO, podemos decir que nuestra laptop es una instancia de la clase de objetos conocida como computadoras.

Una *clase* es el conjunto de especificaciones con las cuales los objetos individuales son creados.

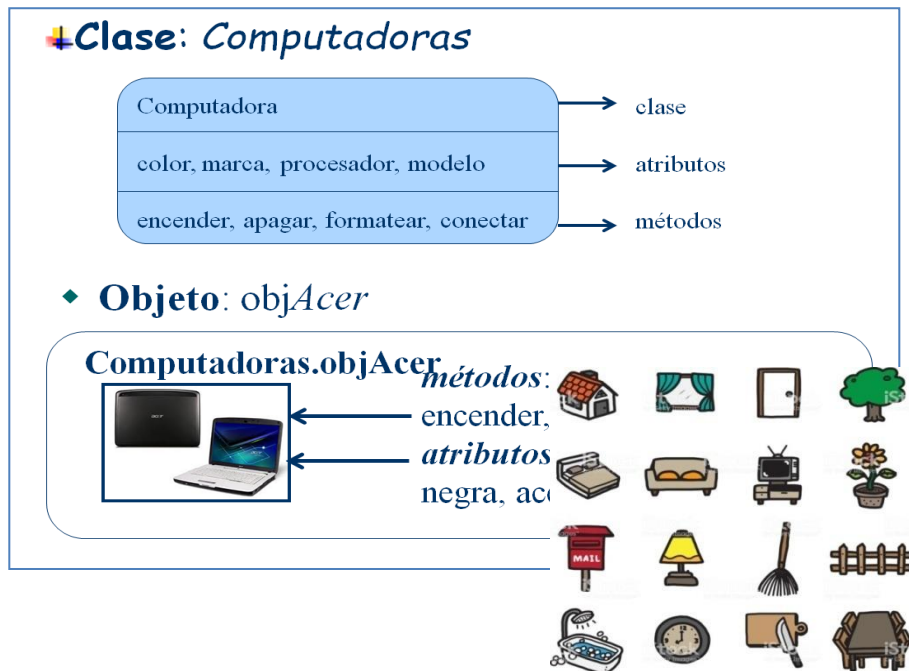
Dichas especificaciones quedan definidas en POO como propiedades o atributos y comportamientos o métodos.

## APLICA LA METODOLOGIA DE DESARROLLO RAPIDO DE APLICACIONES CON POE – PRIMER PARCIAL

Una clase es como un tipo de dato creado por el usuario, que posee dos categorías de miembros:

- ✚ **Atributos** (Datos)
  - ⇒ Estado
- ✚ **Métodos** (Algoritmos)
  - ⇒ Comportamiento

Ejemplo: ⇒



### Ejercicio:

Encuentra 5 clases del mundo real (3 tangibles y 2 intangibles), y define para cada una 5 atributos y 4 métodos. Instanciar 4 objetos. Ejemplo:

#### Clase: Computadoras

##### Atributos:

Marca	Color	RAM Gb	DD Tb	Vel MHz
Dell	Rojo	4	2	2
Hacer	Oro	8	4	3
HP	Plata	4	2	4
Lenovo	Azul	6	3	5

} Objetos tangibles

##### Métodos:

Encender, apagar, formatear, imprimir

#### Clase: AUTOMOVILES

##### Atributos:

Marca	Color	Nombre	velocidad	hp
Kia Motors	Rojo	Rio	180	500
Wolkswagen	Negro	Vento	180	400
FORD	Verde	Lobo	180	475
Chevrolet	Azul	Aveo	180	425

} Objetos tangibles

##### Métodos:

Encender, apagar, conducir, cargar

#### Clase: VIDEOJUEGOS

##### Atributos:

Nombre	Color	RAM Gb	Generación	bits
PS2	Negro	4	Sexta	128
Xbox	Negro	8	Sexta	128

} Objetos tangibles

Nintendo	Blanco	4	Sexta	128
Wii	Blanco	6	Séptima	?

**Métodos:**

Encender, apagar, Jugar, Restablecer

**Clase:** SISTEMAS OPERATIVOS

**Atributos:**

Nombre	Forma	Versión reciente	Dispositivos compatibles	Codigo
Android	Pie	9.0	Celular	Java, C++
iOS	manzana	13.0	Celular	C, C++
Microsoft Windows	ventana	8.1	Celular/PC	C,C++
Balckberry OS	?	7.1.0	Celular	Java.C++

} **Objetos intangibles**

**Métodos:**

Configurar, Restablecer, almacenar,

**Clase:** FRANQUICIAS

**Atributos:**

Nombre	Pais	ambito	Tipo	sucursales MEX
McDonalds	Estados Unidos	Comida	Hamburguesa	436
PEMEX	Mexico	Recursos naturales	Gasolina	168,000
Burguer King	Estados Unidos	Comida	Hamburguesa	513
Farmacias del Ahorro	Mexico	Medicina	medicamentos	1,400

} **Objetos intangibles**

**Métodos:**

Vender, comprar, adquirir, negociar

## 1. Qué es la programación orientada a objetos (POO)?

La programación orientada a objetos es un enfoque de programación que combina datos y acciones asociadas (métodos) en estructuras lógicas (objetos). Este enfoque aumenta la capacidad para administrar la complejidad del software, lo cual resulta especialmente importante cuando se desarrollan y mantienen aplicaciones y estructuras de datos de gran tamaño. (MATLAB, 2018).

La programación Orientada a objetos (POO, u OOP según sus siglas en inglés) es un paradigma de programación que usa objetos en sus interacciones, para diseñar aplicaciones y programas informáticos.

Está basada en varias técnicas, incluyendo herencia, cohesión, abstracción, polimorfismo, acoplamiento y encapsulamiento. Su uso se popularizó a principios de la década de 1990. En la actualidad, existe una gran variedad de lenguajes de programación que soportan la orientación a objetos. (Ángel Roldán, 2017).

## 2. Qué es la programación orientada a eventos (POE)?

La programación dirigida por eventos (creado en 2013), es un paradigma de programación en el que tanto la estructura como la ejecución de los programas van determinados por los sucesos que ocurran en el sistema, definidos por el usuario o que ellos mismos provoquen.

Para entender la programación dirigida por eventos, podemos oponerla a lo que no es: mientras en la programación secuencial (o estructurada) es el programador el que define cuál va a ser el flujo del programa, en la programación dirigida por eventos será el propio usuario —o lo que sea que esté accionando el programa— el que dirija el flujo del programa. Aunque en la programación secuencial puede haber intervención de un agente externo al programa, estas intervenciones ocurrirán cuando el programador lo haya determinado, y no en cualquier momento como puede ser en el caso de la programación dirigida por eventos.

### 3. Características de la programación orientada a objetos

**Abstracción:** denota las características esenciales de un objeto, donde se capturan sus comportamientos. Cada objeto en el sistema sirve como modelo de un "agente" abstracto que puede realizar trabajo, informar y cambiar su estado, y "comunicarse" con otros objetos en el sistema sin revelar cómo se implementan estas características. Los procesos, las funciones o los métodos pueden también ser abstraídos, y, cuando lo están, una variedad de técnicas son requeridas para ampliar una abstracción. El proceso de abstracción permite seleccionar las características relevantes dentro de un conjunto e identificar comportamientos comunes para definir nuevos tipos de entidades en el mundo real. La abstracción es clave en el proceso de análisis y diseño orientado a objetos, ya que mediante ella podemos llegar a armar un conjunto de clases que permitan modelar la realidad o el problema que se quiere atacar.

**Encapsulamiento:** significa reunir todos los elementos que pueden considerarse pertenecientes a una misma entidad, al mismo nivel de abstracción. Esto permite aumentar la cohesión de los componentes del sistema. Algunos autores confunden este concepto con el principio de ocultación, principalmente porque se suelen emplear conjuntamente.

**Herencia:** las clases no están aisladas, sino que se relacionan entre sí, formando una jerarquía de clasificación. Los objetos heredan las propiedades y el comportamiento de todas las clases a las que pertenecen. La herencia organiza y facilita el polimorfismo y el encapsulamiento, permitiendo a los objetos ser definidos y creados como tipos especializados de objetos preexistentes. Estos pueden compartir (y extender) su comportamiento sin tener que volver a implementarlo. Esto suele hacerse habitualmente agrupando los objetos en clases y estas en árboles o enrejados que reflejan un comportamiento común. Cuando un objeto hereda de más de una clase se dice que hay herencia múltiple.

**Polimorfismo:** comportamientos diferentes, asociados a objetos distintos, pueden compartir el mismo nombre; al llamarlos por ese nombre se utilizará el comportamiento correspondiente al objeto que se esté usando. O, dicho de otro modo, las referencias y las colecciones de objetos pueden contener objetos de diferentes tipos, y la invocación de un comportamiento en una referencia producirá el comportamiento correcto para el tipo real del objeto referenciado. Cuando esto ocurre en "tiempo de ejecución", esta última característica se llama asignación tardía o asignación dinámica. Algunos lenguajes proporcionan medios más estáticos (en "tiempo de compilación") de polimorfismo, tales como las plantillas y la sobrecarga de operadores de C++.

### 4. Principales lenguajes de programación orientados a eventos.

**JavaScript:** (abreviado comúnmente JS) es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos,<sup>3</sup> basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente en su forma del lado del cliente (*client-side*), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas<sup>4</sup> aunque existe una forma de JavaScript del lado del servidor (*Server-side JavaScript* o *SSJS*).



## APLICA LA METODOLOGIA DE DESARROLLO RAPIDO DE APLICACIONES CON POE – PRIMER PARCIAL

Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo.

**Visual Basic (VB):** es un lenguaje de programación dirigido por eventos, desarrollado por Alan Cooper para Microsoft. Este lenguaje de programación es un dialecto de BASIC, con importantes agregados. Su primera versión fue presentada en 1991, con la intención de simplificar la programación utilizando un ambiente de desarrollo. La última versión fue la 6, liberada en 1998, para la que Microsoft extendió el soporte hasta marzo de 2008.

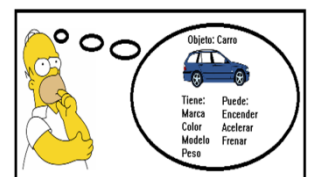
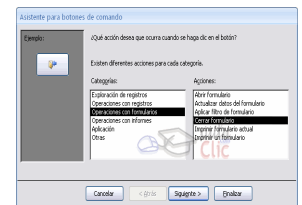
**Visual C++:** (también conocido como MSVC++, Microsoft Visual C++) es un entorno de desarrollo integrado (IDE) para lenguajes de programación C, C++ y C++/CLI. Visual C++ engloba el desarrollo de aplicaciones hechas en C, C++ y C++/CLI en el entorno Windows. Visual C++ incluye además las bibliotecas de Windows (WinApi), las bibliotecas MFC y el entorno de desarrollo para .NET Framework. Visual C++ cuenta con su propio compilador (de igual nombre) y otras herramientas como IntelliSense, TeamFoundation Server, Debug,... Además provee de bibliotecas propias de cada versión del sistema operativo y sockets. Como otros compiladores, se le pueden añadir nuevas bibliotecas como DirectX, wxWidgets o SDL.

## Objetos en mundo real

En el mundo real un objeto es cualquier cosa material y determinada que tiene alguna función de utilidad y que posee características o propiedades iguales o distintas a la de otros objetos, por ejemplo, un carro, una computadora, un televisor, un celular, etc.

## Objetos en programación

En el mundo de las computadoras un *objeto* es todo lo que se ve en una aplicación típica de Windows, Linux, Mac, etc. Los *objetos* son, por ejemplo, un *botón de comando*, una *caja de texto*, una *imagen*, un *botón de opción*, una *lista desplegable*, una *barra de desplazamiento*, en general todo objeto visible que usted puede observar en la pantalla.



- ✚ No podemos definir un *objeto* como algo que simplemente se ve en la pantalla, un objeto es algo más abstracto y de inmensa profusión de datos.
- ✚ Un *objeto* posee *características* o *propiedades* que definen su aspecto exterior, por ejemplo, el color, el tamaño, el estilo, etc.
- ✚ Un *objeto* también posee *métodos* y *eventos* que son elementos significativos del objeto.



## Programas orientados a eventos

Todas las aplicaciones creadas en un *lenguaje de programación orientado a objetos* serán por defecto un *programa orientado a eventos*.

Es un programa orientado a eventos, porque cuando éste es cargado o ejecutado solo espera a que el usuario realice alguna acción sobre uno de los *objetos* que posee, por ejemplo, la calculadora de Windows espera a que el usuario haga **clik (Evento Click)** con el Mouse sobre uno de los botones de comando que contienen los números para luego ponerlo en la *caja de texto*, o bien, espera a que el usuario pulse un número (**Evento KeyPress**) desde el teclado para ponerlo en la *caja de texto*.

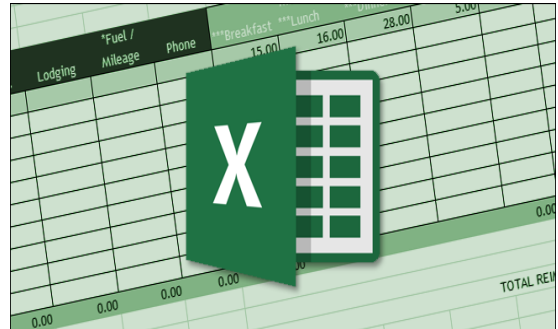
Los programas o paquetes de programas que utilizamos con frecuencia en Windows son *programas orientados a eventos*. Solo cuando el usuario realiza alguna acción sobre los objetos de un programa ocurre un determinado tipo de evento.



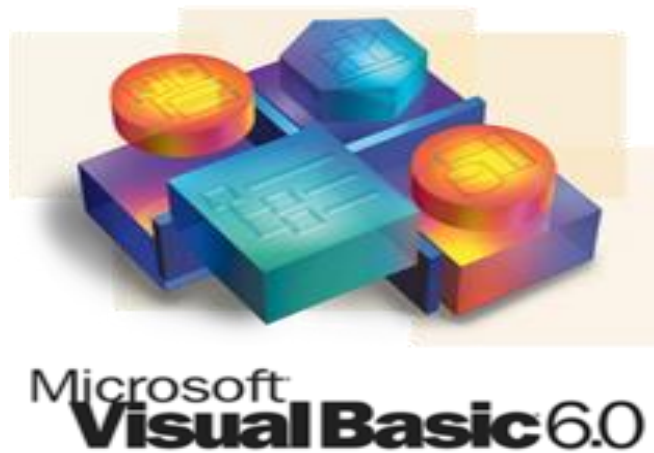
## APLICA LA METODOLOGIA DE DESARROLLO RAPIDO DE APLICACIONES CON POE – PRIMER PARCIAL

Mediante ese evento el programa realiza una operación y luego notifica al usuario sobre los resultados de dicha operación. Puede probar esto cargando cualquier aplicación de Windows ya sea *Microsoft Word* o *Microsoft Excel*, cuando este es cargado se queda a la espera de que usted haga algo, por ejemplo, *Word espera a que usted escriba* alguna carta u otro tipo de documento, así como, *Excel espera a que el usuario escriba*, por ejemplo, los valores de alguna nomina de pago.

Un programa orientado a eventos es capaz de manejar un sin numero de determinados *tipos de eventos* y que también son capaces de responder a cada uno de ellos.



### Antecedentes de Visual Basic



### Qué es Visual Basic?

Es un lenguaje de programación dirigido por eventos, desarrollado por Alan Cooper para Microsoft. Alan Cooper nació el 3 de junio de 1952, es un diseñador de software y programador estadounidense, ampliamente reconocido como el "Padre de Visual Basic".



La palabra "Visual" hace referencia al método que se utiliza para crear la interfaz gráfica de usuario (GUI). En lugar de escribir numerosas líneas de código para describir la apariencia y la ubicación de los elementos de la interfaz, simplemente puede agregar objetos prefabricados en su lugar dentro de la pantalla.

La palabra "Basic" hace referencia al lenguaje BASIC (Beginners All-Purpose Symbolic Instruction Code).

### Antecedentes Visual Basic

Su primera versión fue presentada en 1991, con la intención de simplificar la programación utilizando un ambiente de desarrollo.

Visual Basic ha evolucionado a partir del lenguaje BASIC original y ahora contiene centenares de instrucciones, funciones y palabras clave, muchas de las cuales están directamente relacionadas con la interfaz gráfica de Windows.

La última versión fue la 6, liberada en 1998, para la que Microsoft extendió el soporte hasta marzo de 2008.

### Características Visual Basic

*Visual Basic 6.0* es un lenguaje de programación visual, también llamado lenguaje de 4ª generación. Esto quiere decir que un gran número de tareas se realizan sin escribir código, simplemente con operaciones gráficas realizadas con el ratón sobre la pantalla.

*Microsoft Visual Basic*, presenta una manera más rápida y sencilla de crear aplicaciones para *Microsoft Windows®*; proporciona un juego completo de herramientas que facilitan el desarrollo rápido de aplicaciones.

Aunque Visual Basic es de propósito general, también provee facilidades para el desarrollo de aplicaciones de bases de datos usando *Data Access Objects*, *Remote Data Objects* o *ActiveX Data Objects*.

Visual Basic contiene un entorno de desarrollo integrado o IDE que integra editor de textos para edición del código fuente, un depurador, un compilador (y enlazador) y un editor de interfaces gráficas o GUI.

Visual Basic 6.0 es también un programa basado en objetos, aunque no orientado a objetos como C++ o Java. La diferencia está en que Visual Basic 6.0 utiliza objetos con propiedades y métodos, pero carece de los mecanismos de herencia y polimorfismo propios de los verdaderos lenguajes orientados a objetos como Java y C++.

### Tipos de programas

Un programa **secuencial** es un programa que se arranca, lee los datos que necesita, realiza los cálculos e imprime o guarda en el disco los resultados. De ordinario, mientras un programa secuencial está ejecutándose no necesita ninguna intervención del usuario.

Los programas **interactivos** exigen la intervención del usuario en tiempo de ejecución, bien para suministrar datos, bien para indicar al programa lo que debe hacer por medio de menús. Los programas interactivos limitan y orientan la acción del usuario.

Por su parte los programas **orientados a eventos** son los programas típicos de Windows, tales como Word, Excel y PowerPoint. Cuando uno de estos programas ha arrancado, lo único que hace es quedarse a la espera de las acciones del usuario, que en este caso son llamadas **eventos**.

### Reglas de nomenclatura de Visual Basic

Respete las reglas siguientes cuando asigne nombres a procedimientos, constantes, variables y argumentos en un módulo de Visual Basic:

- El primer carácter tiene que ser una letra.
  - No puede usar un espacio, un punto (.), un signo de número #, de exclamación, @, &, en general ningún de los caracteres especiales.
  - La longitud del nombre no puede exceder los 255 caracteres.
- ✓ Por lo general, no debe usar ningún nombre que sea igual a los nombres de función, instrucción, método y constante intrínseca utilizados en Visual Basic o por la aplicación host.
- ✓ No se pueden repetir nombres dentro del mismo nivel de ámbito. Por ejemplo, no puede declarar dos variables denominadas **age** dentro del mismo procedimiento. Sin embargo, puede declarar una variable privada llamada **age** y una variable de nivel de procedimiento **age** denominada dentro del mismo módulo.

## APLICA LA METODOLOGIA DE DESARROLLO RAPIDO DE APLICACIONES CON POE – PRIMER PARCIAL

- ✓ Visual Basic no distingue entre mayúsculas y minúsculas, pero conserva las mayúsculas en la instrucción en la que se declara el nombre.

### Tipos De Datos

Tipo	Descripción	Carácter de declaración	Rango
Boolean	Binario		True o False
Byte	Entero corto		0 a 255
Integer	Entero (2 bytes)	%	-32768 a 32767
Long	Entero largo (4 bytes)	&	-2147483648 a 2147483647
Single	Real simple precisión (4 bytes )	!	-3.40E+38 a 3.40E+38
Double	Real doble precisión ( 8 bytes)	#	-1.79D+308 a 1.79D+308
Currency	Número con punto decimal fijo (8 bytes)	@	-9.22E+14 a 9.22E+14
String	Cadena de caracteres (4 bytes + 1 byte/car hasta 64 K)	\$	0 a 65500 caracteres.
Date	Fecha (8 bytes)		1 de enero de 100 a 31 de diciembre de 9999. Indica también la hora, desde 0:00:00 a 23:59:59.
Variant	Fecha/hora; números enteros, reales, o caracteres (16 bytes + 1 byte/car. en cadenas de caracteres)	ninguno	F/h: como Date números: mismo rango que el tipo de valor almacenado
User-defined	Cualquier tipo de dato o estructura de datos. Se crean utilizando la sentencia Type (Ver Apartado 3.10)	ninguno	

### Ejercicio.

Definir 7 variables con su tipo de dato para los siguientes casos:

- Estudiantes
- Productos de un supermercado
- Cuentas bancarias
- Historial de calificaciones

Se agrega un ejemplo.

#### Estudiantes

No.	Dato	Variable	Tipo de Dato	Valor de Ejemplo
1	Nombre de pila	Nombre	String	Juan Perez
2	Fecha de inscripción	FechaInscripcion	Date	11/09/2019
3	Cuota a pagar por Inscripcion	CuotaInscripcion	Currency	\$890.00
4	Año de nacimiento	AnioNacimiento	Integer	1995
5	Número de control	NumeroControl	String	18317060440212
6	Teléfono	Nelefono	String	7771230948
7	Edad	Edad	byte	17

### Productos de un Supermercado

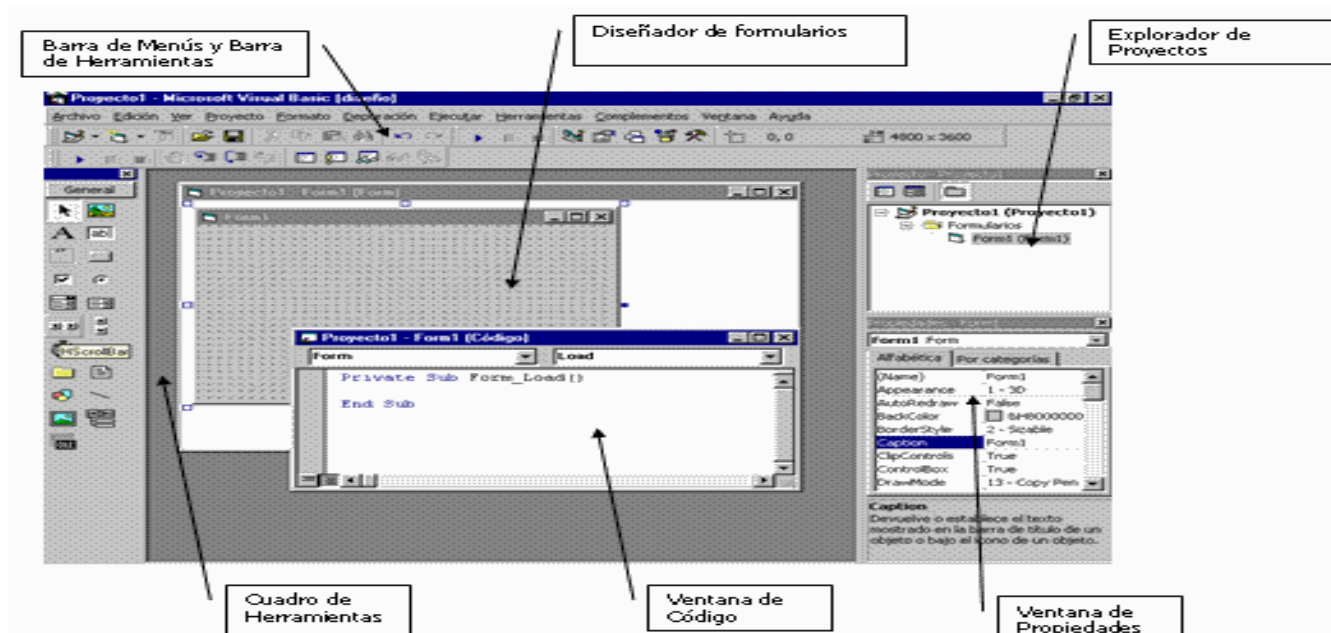
No.	Dato	Variable	Tipo de Dato	Valor de Ejemplo
1	Nombre del producto	Nombre	String	refresco
2	Fecha de caducidad	FechaCad	Date	11/09/2019
3	Precio 1	P1	Currency	\$15.00
4	Precio 2	P2	Currency	1995
5	Sku	codigo	integer	06375
6	Marca del producto	marca	String	Coca Cola
7	piezas	pz	byte	87

### Cuentas bancarias

No.	Dato	Variable	Tipo de Dato	Valor de Ejemplo
1	Nombre de pila	Nombre	String	Camila Gomez
2	Monto de la cuenta	monto	currency	\$4500
3	Banco	banco	String	BBVA
4	Empresa crediticia	empresacre	String	Visa
5	Número de cuenta	NumeroControl	String	4915669338677840
6	Teléfono	Nelefono	String	7771234567
7	Fecha de vencimiento	fechaven	Date	6/05/2022

### Historial de calificaciones

No.	Dato	Variable	Tipo de Dato	Valor de Ejemplo
1	Nombre de pila	Nombre	String	Ulises Sanchez
2	Grupo	grupo	string	E
3	Promedio	CuotaInscripcion	double	9.5
4	Fecha de nacimiento	fechaNacimiento	Integer	1/06/2003
5	Número de control	NumeroControl	String	18317060440212
6	Semestre	semestre	byte	3
7	Edad	Edad	byte	17



## Componentes Pantalla Principal

### 1. La **barra de títulos**, la **barra de menús** y la **barra de herramientas**

Permite el acceso a las opciones para manejo de archivos, compilación y ejecución.

### 2. **Caja de herramientas (toolbox)**

Contiene los controles disponibles (clases) con los cuales podemos crear objetos y colocarlos en el diseño de nuestro formulario.

### 3. **Ventana de proyecto**

Muestra los formularios y otros módulos de programas que forman parte de la aplicación.

### 4. **Diseñador de Formulario (form)**

Aquí se pueden ir insertando los objetos e ir dando diseño, forma a nuestras aplicaciones. Está dotado de una rejilla (**grid**) para facilitar la alineación de los controles.

### 5. **Ventana de Código.**

Es la parte que contiene el editor de texto donde podemos escribir nuestro código fuente que irá asociado los objetos creados en el formulario. Aquí podemos asignar valor a las propiedades, mandar llamar los métodos y programar los eventos asociados a los objetos.

### 6. **Ventana de Propiedades.**

Permite ver las propiedades y sus valores del objeto seleccionado o del propio formulario, también aquí podemos modificar dichos valores. Si esta ventana no aparece, se puede hacer visible con la tecla <F4>.

### 7. **Ventana de Posición en el formulario (FormLayout).**

Permite determinar la forma en que se abrirá la aplicación cuando comience a ejecutarse (abajo a la derecha).

### 8. **Explorador de Proyectos.**

Permite ver la estructura de archivos del proyecto abierto, ya sean formularios o archivos de módulo. También permite el acceso rápido las modalidades de diseño y código.

## Entorno

### **Modo de Diseño y Modo de Ejecución**

Permite trabajar de dos modos distintos: en modo de diseño y en modo de ejecución.

#### **Modo de diseño**

El usuario construye interactivamente la aplicación, colocando objetos en el formulario, definiendo sus propiedades y desarrollando código para gestionar los eventos.

#### **Modo Ejecución**

Una vez terminado el diseño y el código, la aplicación se pone a prueba en modo de ejecución. En ese caso el usuario actúa sobre el programa (introduce eventos) y prueba cómo responde el programa.

## Nombres De Objetos

## APLICA LA METODOLOGIA DE DESARROLLO RAPIDO DE APLICACIONES CON POE – PRIMER PARCIAL

En principio cada objeto de Visual Basic 6.0 debe tener un nombre, por medio del cual se hace referencia a dicho objeto. El nombre puede ser el que el usuario desee, e incluso Visual Basic 6.0 proporciona nombres por defecto para los diversos controles.

Estos nombres por defecto hacen referencia al tipo de control y van seguidos de un número que se incrementa a medida que se van introduciendo más controles de ese tipo en el formulario (por ejemplo Command1, para un botón de comando, Command2, para un segundo botón y así sucesivamente).




















Los nombres por defecto **no son adecuados** porque hacen referencia al tipo de control, pero no al uso que de dicho control está haciendo el programador.

Existe una **convención** ampliamente aceptada que es la siguiente:

- Se utilizan siempre tres letras minúsculas que indican el tipo de control.
- Los siguientes caracteres son otras letras (la primera mayúscula, a modo de separación) libremente escogidas por el usuario, que tienen que hacer referencia al uso que se le da al objeto.
- La palabra (s) que hace referencia al uso, se usa su primer letra pero las subsecuentes se omiten las vocales a excepción de la vocal de la última sílaba.
- Ejemplo:  
Una caja de texto que recibirá el apellido paterno de una persona, su nombre correcto sería:

**TxtApIldoPtrno**

## ABREVIATURAS PARA NOMBRES DE OBJETOS

ABREVIATURA	CONTROL	ICONO	ABREVIATURA	CONTROL	ICONO
chk	CheckBox		lbl	Label	
cmb	ComboBox		lin	Line	
cmd	CommandButton		lst	ListBox	
dir	DirListBox		mnu	Menu	
drv	DriveListBox		opt	OptionButton	
fil	FileListBox		pct	PictureBox	
frm	Form		shp	Shape	
Fra	Frame		txt	TextBox	
hsb	HorizontalScrollBar		tmr	Timer	
img	Image		vsb	VerticalScrollBar	

## COMENTARIOS PERSONALES

PUES QUIERO DECIR QUE LO QUE APRENDI DE ESTE PARCIAL DE ESTA MATERIA FUE QUE PUDE VER TODO LO QUE DESCONOCIA, AL VER NUEVOS CONCEPTOS DE PROGRAMACION, EL DESARROLLO DE UN NUEVO LENGUAJE QUE ESTOY ADQUIRIENDO EN EL CURSO, ME GUSTO EN UNA PARTE DONDE VIMOS LOS CONCEPTOS DE VISUAL BASIC, QUE ES, COMO FUNCIONA Y COMO SE PROGRAMA, LO MALO ES QUE HAY PEQUEÑAS DEFINICIONES QUE NO SE ME QUEDARON EN LA MEMORIA, AUNQUE PUEDO DECIR QUE SI PUEDO MEJORAR UN POQUITO EN LOS CPNCEPTOS, ME GUSTARIA QUE FUERA MAS DINAMICA LA CLASE, NO DIGO QUE ES ABURRIDA, SINO QUE PODAMOS TODOS ACOPLARNOS A LOS CONCEPTOS Y PUES FUE MUY FACIL ESTE PARCIAL

## BIBLIOGRAFIA

- \*[http://docs.google.com/viewer?a=v&q=cache:E1lDxNn\\_NAJ:curiosisios.files.wordpress.com/2009/12/modelo-de-desarrollo-rapido-de-aplicaciones.pdf+desventajas+del+modelo+rad](http://docs.google.com/viewer?a=v&q=cache:E1lDxNn_NAJ:curiosisios.files.wordpress.com/2009/12/modelo-de-desarrollo-rapido-de-aplicaciones.pdf+desventajas+del+modelo+rad)
- \*[http://es.wikipedia.org/wiki/Herramienta\\_CASE](http://es.wikipedia.org/wiki/Herramienta_CASE)
- \*<http://members.fortunecity.com>
- \*<https://postparaprogramadores.com/desarrollo-rapido-de-aplicaciones/>
- \*<https://slideplayer.es/slide/4620488/>
- \*<https://www.slideshare.net/DamianGordon1/rapid-application-development-model-80626463>
- \*Rapid Application Development. James Martin. 1980
- \*Ingenieria del Software I. Tema 2.
- \*Universidad de Cantabria. Facultad de Ciencias
- \*[https://es.wikipedia.org/wiki/Metodolog%C3%ADa\\_de\\_desarrollo\\_de\\_software](https://es.wikipedia.org/wiki/Metodolog%C3%ADa_de_desarrollo_de_software)