

Docente: Miguel Meza / Alejandro Padilla

Materia: Introducción a la Programación

Introducción a la Programación

Hay 3 niveles (en el diagnóstico):

1 Entrada, Salida y cálculo

2 Condicionales

3 Ciclos

Entrada, Salida y Cálculo

Lenguaje normal

En programación

2 números

x, y

Entrada

Ejemplo:

Imprimir "Usuario, dame un número"

Leer x

Imprimir "Otro número"

Leer z

respuesta = x + y

Imprimir respuesta

En estos casos: (en Pse Int)

Entrada

Leer

Salida

Imprimir

Cálculo

Operación escrita

- En Pse Int no usar "y" como variable -

Aprender la metodología
Eso es una serie de pasos o también llamado
"Algoritmo"

Metodología para resolver problemas computacionales
según el Dr. Meza

- 1 Entender el problema
- 2 Asegurar que entienda el problema
- 3 Analizar entrada, salida y procesos
- 4 Solucionar con pseudocódigo o diagrama de flujo
- 5 Codificar en lenguaje

* Tarea Leer el capítulo 2 del libro de
schoolology y ¿qué es una aplicación? y leer los 16 problemas
Para sacar temas de interés, dudas o no sé

En C++

Hay diferentes tipos de variables

2^{16} bit ~~int~~ int (Guarda números enteros)
1 bit bool (Guarda 0 ó 1 ó true o false)
 2^{16} bit float (Guarda decimales)
1 bit char (Guarda caracteres)

int y float pueden trabajar entre sí
bool y char no, sólo entre cada clase

Se guardan en la RAM

Si se necesitan más grandes hay otras variables

Practico

Practica

Comando scanf

Lee alguna variable:

scanf("%d");

int %d ó %i

bool %b

float %f

char %s ó %c

Para leer el tipo de variable:

scanf("%d", &x);

\n
Salto de línea

\t
tabulador

Problemas Básicos Secuenciales (Schooly)

9:

abc

x abc MOD 10 = 00c → f

abc MOD 100 = 0bc → g

abc trunc ÷ 100 = 00a → h

$$g - f = p$$

$$f \cdot 100 = g$$

$$g + p + h = cba$$

Problemas Básicos Secuenciales (Schooly): Tarea

Básico = Fácil, Intermedio = Puedo pero con ayuda, Avanzado = Complicado

1: ~~Básico~~ Básico pero confuso lo pedido 9: Intermedio

2: Básico

10: Básico

3: Básico

11: Avanzado porque no entendí el problema

4: Intermedio pero por complejidad

12: Intermedio por interpretación

5: Básico

13: Intermedio por interpretación

6: Básico

14: Intermedio

7: Básico

15: Básico

8: Básico

16: Intermedio por interpretación

Capítulo 2: Inicio de un Desarrollo

Metodologías de Desarrollo

Metodología de desarrollo → framework (marco de trabajo) que funciona para estructurar, planear y controlar el proceso de desarrollo → Organización a proyectos

Para conceptualizar una metodología, primero hay que definir los conceptos fundamentales que involucren a una metodología:

- Metodología: Conjunto de procedimientos y técnicas que utilizan los desarrolladores al hacer un software
- Tarea: Actividades elementales en las que se dividen los procesos
- Procedimiento: Forma que se define para ejecutar la tarea
- Técnica: Herramienta para aplicar un procedimiento
- Herramienta: Apoyos que automatizan la aplicación de un software
- Producto: Resultado de cada etapa

Tipos de Metodologías

- * Tradicionales / Pesadas (Cascada): Detallista en documentación de etapas
- Iterativas / Evolutivas (Prototipos)
- Ágiles (XP): Entregas más pequeñas, ciclos cortos y rápidos
- Tecnología Web
- Otras (Tiempo real)

Ciclo de Vida de un Desarrollo

Fases: Nacimiento (problema puntual) → Maduración → Muerte o reemplazo del producto

- Idea (Qué hacer y por qué)
- Análisis (Cómo confeccionarlo)
- Diseño (Qué aspecto elegir)
- Producción (Elaboración)

- Construirlo (Qué herramientas utilizaré y cómo)
- Mantenimiento (Upgrades y mejoras)

Funciones del Ciclo de Vida

- Determinar orden de fases del software
- Establecer transición entre fases
- Puntualizar entradas y salidas de fase
- Describir estados por los que pasa el producto
- Especificar actividades que transformen el producto
- Definir esquema → Metodología

Complejo → ámbito de desarrollo

Roles Profesionales

Poder jugar cualquier papel de los mens involucrados:

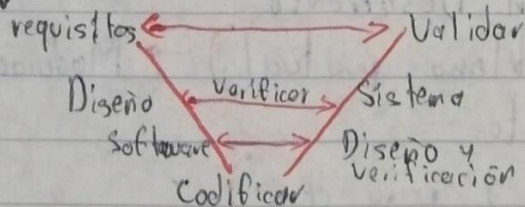
- Analista de sistema
- Líder de proyecto
- Analista funcional (Maquetación y diseño)
- Desarrollador (Codificador)
- Tester/Seguete
- Calidad (Documentación)

Modelos de Ciclo de Vida

- Modelo en cascada

Requisitos → Diseño → Implementación → Pruebas → Mantenimiento

- Modelo en V



- Modelo iterativo

Cascada → Versión 1 → Cascada → Versión 2 → ...

- Modelo de desarrollo incremental

En qué son diferentes

- Modelo de Prototipos:

Escuchar cliente → Construir maqueta

↑ Cliente lo prueba ←

- Modelo en espiral

Determinar objetivos → Evaluar riesgos

↑ Planificar ← Desarrollar y probar

Generalidades sobre Metodologías

- Desarrollo Convencional (sin metodología)

- Resultados impredecibles
- Incontrolable
- Cambios en organización afectan negativamente

- Desarrollo Estructurado

- Estructurado (numa :v)
- Funcional
- Gráficas
- Particionadas
- Poco redundantes

- Desarrollo Orientado a Objetos

- Elimina fronteras entre fases
- Nueva forma de concebir lenguajes
- Alto grado de iteración → Dinámico
- Reutiliza componentes

winny {

Análisis de Sistema

Etapas → Encargar negocio

- Requisito: Pedido del cliente

- Relevamiento: Situación solucionar mejorar. → Planificar proyecto
- Diagnóstico: Procesos actuales involucrados
- Prototipado (Diseño de sistema): Dar "rostro" a aplicaciones
- Propuesta teórica: Mostrar a cliente cómo funcionaría
- Implementación y testing

Relevamiento

Identificar fuentes de información, preguntas apropiadas, analizar y confirmar

No siempre será fácil

Técnicas de recolección de información

- Entrevistas (más directo pero requieren tiempo y dinero)
- Observación
- Estudio de documentación
- Cuestionarios
- Brainstorming

Diseño de un Sistema

Diseño de prototipos y funcionamiento

Usar UML (lenguaje de modelado unificado)

Diagrama de Casos de Uso

Empleado en desarrollar sistemas informáticos

Esquema comportamental → situación

- Actores
- (Elipse)

Prototipos

Implementación del Desarrollo

Testing

Capacitación del Usuario

Capítulo 3: Ingreso al Mundo de la Programación

La Lógica de un Humano y de una Máquina

Alan Turing unió la lógica y la computación antes de que cualquier computadora fuera inventada

"El pensamiento lógico de las computadoras se basa en un lenguaje binario"

Pionero de la teoría de la computación

Pseudocódigo: El Lenguaje Humano

No podemos programar en lenguaje máquina \rightarrow Adaptar el lenguaje humano a formas lógicas \rightarrow Algoritmos

Normas

Pseudocódigo \rightarrow parecido \rightarrow lenguaje de programación en escritura

Contiene léxico

Sintaxis \rightarrow especificación de palabras clave \rightarrow oraciones

Semántica \rightarrow Significado

Pseudocódigo \leftrightarrow Diagrama Flujo

Tipos de Datos

Númericos ej. 230, 50

Alfanuméricos ej. texto, v@lor3s

Lógicos true, false

Variables

Almacena un dato

ej. $A = 100$, Ciudad = "Córdoba", $A = C + B$

- Simples

- Contador

- Acumulador

- De trabajo

Normas de Escritura

Operadores

Aritméticos

+ - * / ^ MOD

Lógicos

1 true, 0 false

and, or, not

Relacionales

< <= > >= = <>

Entrada y salida de información

Orden en la Programación

Estructuras algorítmicas \rightarrow Secuenciales, condicionales, cíclicas

Datos Estructurados

Vector

Conjunto ordenado de 0 o n homogéneo

También se llaman tablas, listas, arrays

Matrices

Datos homogéneos o mismo tipo

Fila y columna

Funciones y Procedimientos

Módulos pequeños \rightarrow Subalgoritmos

Ámbito de Variables

Locales (dentro de subprograma) y globales (algoritmo principal)

Funciones

Matemáticas (trigonometría)

Procedimientos

Parámetros \rightarrow Conjunto de instrucciones desde el programa principal

Programas Retroalimentación

De m/h

a kn/h

1 m/h

1.609

kn/h

Ciclos

de

0

a

100

de

20

en

20

Sin usar el "Repetir" \rightarrow "Hasta que"

Problemas

Basicos 6

Problema que me enchilo 7iv Prácticas de Pares

$$a^2 + b^2 = c^2$$

Dado un número c^2 encontrar los pares de $a^2 + b^2$ que formen a c^2

input

output

10

2

100

51 (2)

$$c^2 = a^2 + b^2$$

$b=1$

$a=1$

hasta

$a=c$

\rightarrow luego

$b=2$

$a=1$

hasta

$a=c$

\rightarrow luego

\vdots

$b=c$

$a=1$

hasta

$a=c$

Problemas

1- Pedir un número de 4 dígitos y hacer lo necesario para imprimir el número de forma inversa.

Ejemplo: 3485 \rightarrow 5843

2- Hacer un menú con las opciones de

1 3 dígitos

2 4 dígitos

3 5 dígitos

Según la opción seleccionada por el usuario pedir el número de los dígitos indicados para mostrar el inverso

3- Mostrar un menú:

1 3 dígitos

2 4 dígitos

3 5 dígitos

4 Salir

El programa se cierra y sólo sale con la opción 4 "Salir".

1- $abcd \quad abcd \bmod 10 = 000d \rightarrow e$
 $abcd \bmod 100 = 00cd \rightarrow f$
 $abcd \bmod 1000 = 0bcd \rightarrow g$

$$f - e = 00c0 = h$$

$$g - h = 0b00 = i$$

$$abcd \text{ trunc} \div 1000 = 000a = p$$

$$e \cdot 1000 = n$$

$$h \cdot 10 = m$$

$$i \div 10 = o$$

$$m + n + o = p$$

Introducción a la Programación - Juan Carlos Casale

Problemas Básicos Secuenciales (School day)

Problemas Básicos Secuenciales (School day): Tarea

Básico = Fácil, Intermedio = Puede que sea algo avanzado = (un poco)

1- Básico por enfoque lógico 1- Intermedio

2- Básico

3- Básico

4- Intermedio por complejidad 13- Intermedio por interacción

5- Básico

6- Básico

7- Básico

8- Básico