

Trabajo Practico N° 1

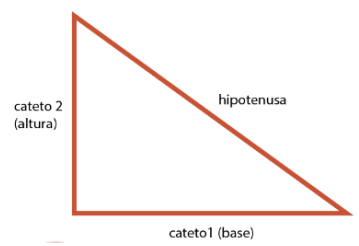
(Ejercicios del 12 al 18)

Olarte Tatiana Magdalena

TUV000467

**Sección Análisis – Diseño y Codificación de algoritmos – Aplicación de estructuras de control**

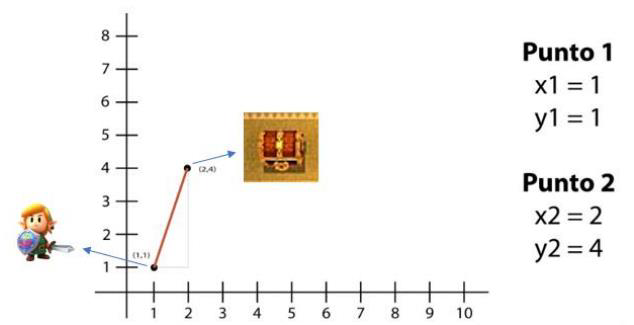
* Ejercicio 12: Un problema sencillo. Deberá pedir por teclado al usuario un nombre y posteriormente realizará la presentación en pantalla de un saludo con el nombre indicado.
* Ejercicio 13: Será común resolver problemas utilizando variables. Calcule el perímetro y área de un rectángulo dada su base y su altura.
* Ejercicio 14: Una ayuda importante al momento de resolver problemas con algoritmos es asumir que su gran amigo son las matemáticas. Obtenga la hipotenusa de un triángulo rectángulo conociendo sus catetos.



* Ejercicio 15: Si viste algo de los apuntes y vídeos, esto debería ser muy fácil de resolver. Dados dos números permita calcular la suma, resta, multiplicación y división de estos. Considere que cada una de estas operaciones es un algoritmo cuando realice el diseño. Obviamente muestre los resultados.
* Ejercicio 16: Necesitamos convertir una temperatura Fahrenheit en grados Celsius. Si no conoce la forma en la que se realiza esta conversión, debería investigarlo; para eso sirve la etapa de análisis. Pero como somos buenos, daremos una ayuda.



* Ejercicio 17: Si queremos representar personajes o power ups (premios) en la pantalla debemos primero ubicarlos en alguna posición dentro de la pantalla. Imagine que está en un juego donde un power up desaparece porque el personaje se acerca a una distancia de x unidades, sin importar por donde se acerque. Por tanto, para que desaparezca, en primer lugar, hay que determinar esa distancia. La forma de representar la posición de un objeto en la pantalla es a través de las coordenadas de un punto. Suponga que la posición de Link está representada por la coordenada (𝑥1, 𝑦1), mientras que las de la caja de tesoro se halla en la posición (𝑥2, 𝑦2). Si observa con detenimiento se observa la conformación de un triángulo rectángulo, por lo que es posible aplicar Pitágoras para obtener la distancia.



Para esto debe calcular el tamaño de los catetos y luego aplicar el teorema. Halle la distancia entre ambos objetos. Cuando programe, represente a lLnk con un Circulo, y al tesoro con un cuadrado. Además, mueva a Link mediante el mouse.

* Ejercicio 18: Desarrolle el análisis y diseño de un algoritmo que permita obtener las raíces de una ecuación de segundo grado. Además, utilice la estructura según para el análisis de la discriminante de la ecuación cuadrática. Obviamente codifique en Processing.

**DESARROLLO:**

* **Ejercicio 12:**

**Definición del Problema**: Se debe ingresar el nombre de un usuario y realizar una presentación de un saludo con el nombre.

**Análisis:**

* Datos de Entrada: El nombre del usuario.
* Datos de Salida: El saludo (“Bienvenido…”) y el nombre a lado.
* Proceso:

¿Quién debe realizar el proceso?: Un usuario

¿Cuál es el proceso que realiza…? Se debe unir el saludo con el nombre (En conjunto).

**Diseño:**

|  |
| --- |
| **ENTIDAD QUE RESUELVE EL PROBLEMA**: Usuario |
| **VARIABLES**  nomUsuario: String // almacena el nombre del usuario que se ingresa  mensajeBien: String // almacena el mensaje de saludo final |
| **NOMBRE ALGORITMO**: saludar\_al\_usuario    **PROCESO DEL ALGORITMO**   1. *Mostrar* “Ingrese su nombre: ” 2. *Leer* nomUsuario 3. mensajeBien ← “Bienvenido ” + nomUsuario 4. *Mostrar* mensajeBien |

* **Ejercicio 13:**

**Definición del Problema**: Calcular el perímetro y área de un rectángulo.

**Análisis:**

* Datos de Entrada: La base y altura del rectángulo.
* Datos de Salida: El perímetro y área del rectángulo.
* Proceso:

¿Quién debe realizar el proceso?: Un usuario

¿Cuál es el proceso que realiza…? Sumar todos los lados del rectángulo (L1+L2+L3+L4) para obtener el perímetro. Y multiplicar la base por la altura (bxh) para obtener el área.

**Diseño:**

|  |
| --- |
| **ENTIDAD QUE RESUELVE EL PROBLEMA**: Usuario |
| **VARIABLES**  baseRectangulo: Real // Almacena el valor de la base que se ingrese  alturaTriangulo: Real // Almacena el valor de la altura que se ingrese  areaRectangulo: Real // Almacena el resultado del radio  perimetroRect: Real // Almacena el resultado del perimetro |
| **NOMBRE ALGORITMO**: obtener\_perímetro  obtener\_area  **PROCESO DEL ALGORITMO**   * + - 1. *Mostrar* “Ingrese la base:”       2. *Mostrar* “Ingrese la altura:”       3. *Leer* baseRectangulo       4. *Leer* alturaRectangulo       5. perimetroRect ← baseRectangulo + baseRectangulo + alturaRectangulo + alturaRectangulo       6. *Mostrar* perimetroRect       7. areaRectangulo ← baseRectangulo \* alturaRectangulo       8. *Mostrar* areaRectangulo |

* **Ejercicio 14:**

**Definición del Problema**: Se debe obtener la hipotenusa de un triángulo rectángulo.

**Análisis:**

* Datos de Entrada: Sus dos catetos.
* Datos de Salida: Cuanto vale su hipotenusa.
* Proceso:

¿Quién debe realizar el proceso?: Un usuario

¿Cuál es el proceso que realiza…? Usar el teorema de Pitágoras (h**²** = ca**²** + co**²**) para sacar la hipotenusa.

**Diseño:**

|  |
| --- |
| **ENTIDAD QUE RESUELVE EL PROBLEMA**: Usuario |
| **VARIABLES**  cateOpuesto: Real // Almacena el valor del co que se ingrese  cateAdyacente: Real // Almacena el valor del ca que se ingrese  hipotenusa: Real // Almacena el resultado de la hipotenusa |
| **NOMBRE ALGORITMO**: hallar\_hipotenusa    **PROCESO DEL ALGORITMO**   * + - 1. *Mostrar* “Ingrese Cateto opuesto”       2. *Mostrar* “Ingrese Cateto adyacente”       3. *Leer* cateOpuesto       4. *Leer* catAdyacente       5. hipotenusa ← (cateOpuesto\*\*2 + cateAdycente\*\*2)\*\*(1/2)       6. *Mostrar* hipotenusa |

* **Ejercicio 15:**

**Definición del Problema**: Calcular la suma, resta, multiplicación y división a partir de dos números.

**Análisis:**

* Datos de Entrada: Los dos números, A y B.
* Datos de Salida: Los resultados de la suma, resta, multiplicación y división.
* Proceso:

¿Quién debe realizar el proceso?: Un usuario

¿Cuál es el proceso que realiza…? Primero debe multiplicar los dos números (AxB), dividir (A/B), después sumar (A+B) y restar (A-B).

**Diseño:**

|  |
| --- |
| **ENTIDAD QUE RESUELVE EL PROBLEMA**: Usuario |
| **VARIABLES**  numeroA, numeroB: Real // Almacenan el valor de los dos números que se ingresen  resultadoSuma: Real // Almacena el valor que da la operación suma  resultadoResta: Real // Almacena el valor que da la operación resta  resultadoMulti: Real // Almacena el valor que da la operación multiplicacion  resultadoDivi: Real // Almacena el valor que da la operación division |
| **NOMBRE ALGORITMO**: sumar\_numeros  restar\_numeros  multiplicar\_numeros  dividir\_numeros  **PROCESO DEL ALGORITMO**   1. *Mostrar “*Ingresar el primer número” 2. *Mostrar “*Ingresar el segundo número” 3. *Leer* numeroA 4. *Leer* numeroB 5. resultadoSuma ← numeroA + numeroB 6. *Mostrar* resultadoSuma 7. resultadoResta ← numeroA – numeroB 8. *Mostrar* resultadoResta 9. resultadoMulti ← numeroA \* numeroB 10. *Mostrar* resultadoMulti 11. resultadoDivi ← numeroA / numeroB 12. *Mostrar* resultadoDivi |

* **Ejercicio 16:**

**Definición del Problema**: Se debe convertir una temperatura Fahrenheit en grados Celsius.

**Análisis:**

* Datos de Entrada: La fórmula que cambia los Fahrenheit a Celsius.
* Datos de Salida: Los grados Celsius.
* Proceso:

¿Quién debe realizar el proceso?: Un usuario

¿Cuál es el proceso que realiza…? Se debe realizar la fórmula de conversión de Fahrenheit a Celsius (ºC = (ºF – 32) / 1.8).

**Diseño:**

|  |
| --- |
| **ENTIDAD QUE RESUELVE EL PROBLEMA**: Usuario |
| **VARIABLES**  gradoCel: Real // Almacena el resultado de la conversión a Celsius  tempFahre: Real // Almacena el valor de los Fahrenheit que se ingresa  numeroA: Real // Almacena el valor 32  numeroB: Real // Almacena el valor 1.8 |
| **NOMBRE ALGORITMO**: convertir\_a\_grados\_celsius    **PROCESO DEL ALGORITMO**   1. *Mostrar “*Ingresar el valor de la temperatura Fahrenheit” 2. *Leer* tempFahre 3. numeroA ← 32 4. numeroB ← 1.8 5. gradoCel ← (tempFahre – numeroA) / numeroB 6. *Mostrar* gradoCel |

* **Ejercicio 17:**

**Definición del Problema**: Se debe hallar la distancia entre el personaje Link y la caja de tesoro.

**Análisis:**

* Datos de Entrada: Las coordenadas de Link (𝑥1, 𝑦1) y las coordenadas de la caja de tesoro (𝑥2, 𝑦2).
* Datos de Salida: La distancia entre ambos objetos.
* Proceso:

¿Quién debe realizar el proceso?: Un usuario

¿Cuál es el proceso que realiza…? Se realiza un teorema de Pitágoras (h**²** = ca**²** + co**²**) ya que se proyecta entre ambas coordenadas un triángulo rectángulo. Primero se debe saber el valor de cada cateto para después hallar su hipotenusa.

**Diseño:**

|  |
| --- |
| **ENTIDAD QUE RESUELVE EL PROBLEMA**: Usuario |
| **VARIABLES**  coordenadaLinkX: Real // Almacena la coordenada en X de Link  coordenadaLinkY: Real // Almacena la coordenada en Y de Link  coordenadaTesX: Real // Almacena la coordenada en X del tesoro  coordenadaTesY: Real // Almacena la coordenada en Y del tesoro  Distancia: Real // Almacena la distancia que hay entre Link y el tesoro |
| **NOMBRE ALGORITMO**: hallar\_distancia    **PROCESO DEL ALGORITMO**   1. coordenadaLinkX ← 1 2. CoordenadaLinkY ← 1 3. CoordenadaTesX ← 2 4. CoordenadaTesY ← 4 5. Distancia ← ((coordenadaTesY - coordenadaLinkY)^2 + (coordenadaTesX - coordenadaLinkX)^2)^½ 6. *Mostrar* distancia |

* **Ejercicio 18:**

**Definición del Problema**: Se debe obtener las raíces de una ecuación de segundo grado y las posibles discriminantes de la misma.

**Análisis:**

* Datos de Entrada: La ecuación de segundo grado y su fórmula general.
* Datos de Salida: Las raíces de la ecuación y sus discriminantes.
* Proceso:

¿Quién debe realizar el proceso?: Un usuario

¿Cuál es el proceso que realiza…? Primero se obtiene las posibles discriminantes (𝑏2−4𝑎𝑐), para saber cuántas raíces tendrá la ecuación y ocupar la estructura de control SEGÚN a fin de, si me sale una discriminante se efectúe una serie de acciones (sacar las raíces), si me sale otra distinta efectuar otro tipo de acciones y así sucesivamente.

La fórmula general ( 𝑥=−𝑏±√𝑏2−4𝑎𝑐2𝑎) para sacar las raíces.

**Diseño:**

|  |
| --- |
| **ENTIDAD QUE RESUELVE EL PROBLEMA**: Usuario |
| **VARIABLES**  coeficienteA: Real // almacena  coeficienteB: Real // almacena  coeficienteC: Real  raiz1: Real  raiz2: Real  discriminante: Real  mensaje: String |
| **NOMBRE ALGORITMO**: obtener\_raices  obtener\_discriminantes  **PROCESO DEL ALGORITMO**  2. *Mostrar* “Ingresar el termino cuadratico”  3. *Leer* coeficienteA  4. *Mostrar* “Ingresar el temino lineal”  5. *Leer* coeficienteB  6. *Mostrar* “Ingresar el termino idependiente”  7. *Leer* coeficienteC  8. discriminante ← coeficienteB^2 – 4\*coeficienteA\*coeficienteC  9. **Según\_sea** (discriminante) hacer  10. **caso** discriminante>0:  11. raiz1 ← -coeficienteB – (coeficienteB^2 – 4\*coeficienteA\*coeficienteC)^(1/2) / (2\*coeficienteA)  12. raiz2 ← -coeficienteB + (coeficienteB^2 – 4\*coeficienteA\*coeficienteC)^(1/2) / (2\*coeficienteA)  13. mensaje ← “Las raíces son”+ raiz1+raiz2  14. sentencia de ruptura  15. **caso** discriminante=0:  16. raiz1 ← -coeficienteB + (coeficienteB^2 – 4\*coeficienteA\*coeficienteC)^(1/2) / (2\*coeficienteA)  17. mensaje ← “La raíz es”+ raiz1  18. sentencia de ruptura  19. **caso** discriminante<0:  20. mensaje ← “No tiene raices”  21. sentencia de ruptura  22. otros:  23. mensaje ← “ERROR”  24. sentencia de ruptura  25. **fin\_segun**  26. *Mostrar* mensaje |

**CONCLUSIÓN:**

**FUENTES BIBLIOGRÁFICAS:**

* clase 03 09 04 24 Fudnamentos de Programación Orientada a objetos: <https://youtu.be/YHpW_HMP_PE?si=PmhigKipfKlWiWaD>
* 14 Estructuras de Control Iterativas: <https://virtual.unju.edu.ar/mod/resource/view.php?id=300666>
* 11 Bifurcaciones - Expresiones lógicas - Estructuras selectivas: <https://virtual.unju.edu.ar/mod/resource/view.php?id=299585>
* Clase 02 Aplicación Análisis y diseño operadores aritméticos y bifurcaciones: <https://youtu.be/bMJhygO43cA?si=UAbmVd5YzrE0Drhi>
* 14 Estructuras de Control Iterativas: <https://virtual.unju.edu.ar/mod/resource/view.php?id=300666>