|  |  |  |
| --- | --- | --- |
| Imagen que contiene dibujo, taza  Descripción generada automáticamente | **Carátula para entrega de prácticas** | |
| Facultad de Ingeniería | | Laboratorio de docencia |

Laboratorios de computación

salas A y B

*Profesor: Ing. Karina Garcia Morales \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Asignatura:\_Fundamentos de Programacion \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Grupo: \_20\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*No de Práctica(s): \_\_\_\_\_Practica 3 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Integrante(s): \_\_\_Mendoza Hernandez Juan Pablo \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*No. de Equipo de*

*cómputo empleado:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*No. de Lista: \_\_\_\_\_\_\_\_\_\_\_\_33\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Semestre: \_\_\_\_202-1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Fecha de entrega: \_\_\_\_\_\_\_04-10-21\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Observaciones: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

CALIFICACIÓN: \_\_\_\_\_\_

***\*Solución de Problemas y Algoritmos\****

***Objetivos.***

Elaborar algoritmos correctos y eficientes en la solución de problemas siguiendo las etapas de Análisis y Diseño pertenecientes al Ciclo de vida del software.

***Desarrollo de la practica***

***Conceptos.***

***Problema informático*:** Es definido así el conjunto de inconvenientes los cuales requieren una serie de soluciones, y cada solución pude requerir de múltiples pazos que pueden variar dependiendo de la circunstancia.

***ingeniería en software:*** Un ingeniero usualmente deberá de utilizar los principios de ingeniería en software para la resolución de problemas informáticos, por lo cual es de suma importancia que dichos principios sean sólidos, la IEEE define a la ingeniería en software como ***“La aplicación de un enfoque sistemático, disciplinado y cuantificable hacia el desarrollo, operación y mantenimiento del software".*** (S, 2015)

***métodos para la generación de software.***

Planeación y estimación del proyecto

Pruebas y mantenimiento (validación y verificación)

Análisis de requerimientos del sistema y software.

Codificación

Diseño de la estructura de datos, la arquitectura del programa y el procedimiento algorítmico.

***Ciclo de vida del software:*** Se llama así a todo el proceso de desarrollo del software, y de esta forma se definen las distintas fases de dicho desarrollo de forma organizada.

***Fases del ciclo de vida del software:***

***Requisitos:*** *También llamada definición de necesidades, se plantean los problemas que el software debe de ser capaz de solucionar.*

***Análisis:*** *Se consideran los requisitos del cliente y se examina cualquier restricción*

***Diseño:*** *Se plantea la arquitectura del software.*

***Codificación:*** *Se comienza con la creación del programa mediante el lenguaje seleccionado*

***Pruebas:*** *Se testea el software mediante pruebas de escritorio.*

***Mantenimiento:*** *Se mantiene el programa mediante y se le brinda soporte, adaptándolo a posibles nuevas situaciones.*

***Solución de problemas:*** La solución de problemas, es la metodología que será requerida para lograr con éxito la resolución de una variable, dicho de una forma más sencilla, la resolución de un problema será la lista de pasos que deberán seguirse para conseguir solucionar el percance que se plantea.

***Algoritmo:*** Un algoritmo es una serie de instrucciones secuenciales, las cuales permiten la ejecución o activación de un programa, algunos plantean que el algoritmo es mucho más importante que el propio lenguaje de programación.

***Actividades***

Como primera actividad, analizamos los problemas presentes en la práctica, como problema, se presento el siguiente caso

1. Determinar si un numero dado es positivo o negativo.

Durante este primer análisis de problema se solicitó encontrar alguna solución, para dicho problema, para ello primero teníamos que considerar las restricciones, en este caso particular la restricción seria que el número no podía ser igual a 0, forzosamente debería ser mayor o menor a 0, tras encontrar nuestra restricción procedimos a encontrar nuestro dato de salida, el cual seria el hecho de obtener si el valor seria positivo o negativo.

Numero positivo o negativo

Conjunto de Numero Reales

Después se nos mostro un ejemplo más, en este ejemplo se solicito obtener el mayor de dos números diferentes.

Para ello primero establecimos las restricciones, o valores críticos, los cuales eran que los números deberían ser distintos

Posteriormente definimos a los números como un conjunto perteneciente a los números reales, y definimos nuestros datos de salida como, decir cual de los dos números es mayor.

Num1:

Num:2

Números no pueden sr iguales

Conjunto de números reales

***Por ultimo se mostró un tercer ejemplo.***

En este ejemplo se solicitó obtener el factorial de un número.

Para ello nuevamente como primer paso definimos los valores críticos, el cual seria que el número de entrada debería ser mayor a 0 y no podía ser negativo.

Luego como datos de entrada teníamos que el conjunto de entrada E está dado por el conjunto de los números naturales o por el cero.

Luego como dato de salida teníamos que el conjunto de salida S está conformado por el conjunto de los números naturales.

Conjunto de numero naturales

Conjunto de numero naturales o 0

Tras esto se nos pidió encontrar la solución al siguiente problema.

***Problema***: Solicitar un valor al usuario y seguirlo solicitando hasta que este sea múltiplo de 5

***Restricciones***: El numero deberá ser un número real.

***Datos de Entrada:*** El conjunto de entrada “E” este dado por el conjunto de números reales.

E pertenece a R donde

Num pertenece a R (-∞,∞)

***Datos de Salida;*** El conjunto de salida S, esta conformado por, todos los números reales que sean múltiplos de 5.

S pertenece a R, donde

Res pertenece a todos los R múltiplos de 5.

***Diagrama.***

Conjunto de números Reales

Números reales múltiplos de 5

Tras esto inicio un siguiente problema, este problema era el cómo encontrar novi(a)

***Problema***: Como encontrar Novi(a)

***Restricciones:*** Que la persona no te corresponda, (por razones personales o generales) que la persona se encuentre en una relación.

***Datos de entrada:*** características físicas, emocionales y sociales que busque el individuo (altura, edad, situación emocional)

***Datos de salida:*** Relación estable y bonita.

***Diagrama.***

características buscadas en el individuo

Relación estable y bonita

Tras esto pasamos a ver un poco de los algoritmos, sus principales características y el como son utilizados para la resolución de problemas.

Igual se agregó un nuevo concepto.

***Variable***: se define como características.

Tras esto se utilizo como ejemplo el siguiente caso. Para visualizar algoritmos

**Plantear un algoritmo que determine si un numero es negativo o positivo**.

***Restricciones***: El numero no puede ser 0

***Datos de entrada***: Numero real

***Datos de salida***: Numero positivo o negativo

Tras esto se nos mostro el algoritmo, pero igual se solicitó crear uno nuevo o completar el algoritmo mostrado.

***Algoritmo creado.***

1. Inicio
2. Solicitar numero al usuario
3. Si el numero es igual a 0 regresar al paso 2 de lo contrario ir al paso 4
4. Si el numero ingresado en mayor a 0, decir que es positivo e ir al paso 6
5. Si el numero es menor a 0 decir que es negativo
6. Fin del algoritmo

Tras esto vimos las pruebas de escritorio, las cuales nos ayudan a demostrar el buen funcionamiento de nuestro algoritmo, a la par vimos la importancia de los valores críticos, los cuales son nuestras restricciones.

***Se solicito realizar pruebas de escritorio con 3 valores***

|  |  |  |
| --- | --- | --- |
| ***iteración*** | ***X*** | ***Salida*** |
| ***1*** | 3 | ***Positivo*** |
| ***2*** | -2 | ***Negativo*** |
| ***3*** | 0 | ***Valor no valido*** |

***Tras esto pasamos el siguiente ejemplo.***

***Obtener el mayor de dos numero***

***Datos de entrada:*** *Número real.*

***Restricciones:*** *Los números deben ser diferentes*

***Datos de salida:*** *La impresión del número más grande****.***

***Dominio:*** *Todos los números reales.*

Tras esto se solicito que creamos un algoritmo que solucionara el problema.

1. Inicio
2. Solicitar primer número al usuario
3. Leer primer numero
4. Solicitar 2 numero al usuario, si el numero es = al 1 numero regresar al paso 2, de lo contrario seguir al paso 5
5. Leer 2 numero
6. SI el 1 numero es mayor al 2, “Leer numero 1 es mayor” ir al paso 8
7. Si el numero 2 es mayor, “Leer numero 2 es mayor”
8. Fin del algoritmo.

***Pruebas de escritorio.***

|  |  |  |  |
| --- | --- | --- | --- |
| ***Iteración*** | ***1 numero*** | ***2 numero*** | ***Salida*** |
| ***1*** | ***7*** | ***6*** | ***Numero 7 es mayor*** |
| ***2*** | ***3*** | ***3*** | ***Valor no valido*** |
| ***3*** | ***-12*** | ***-5*** | ***Numero -5 es mayor*** |

***Tras esto pasamos al 3 ejemplo.***

***Obtener el factorial de un número.***

***El factorial de un número está dado por el producto de ese número por cada uno de los números anteriores hasta llegar a 1. El factorial de 0 (0!) es 1.***

***Datos de entrada:*** *Numero entero*

***Restricciones:*** *El numero de entrada debe ser entero y no puede ser negativo*

***Datos de salida:*** *La impresión del factorial del número****.***

***Algoritmo.***

1. ***Inicio***
2. ***Solicitar un numero entero.***
3. ***Si el numero entero es menor a 0 regresar al punto 1***
4. ***Si el numero entero es mayor a 0, asignar como variable entera contador.***
5. ***Si la variable contadora es < o = al numero entero de entrada realizar lo siguiente.***
   1. Se multiplica el valor de la variable contador con el valor de la variable factorial. El resultado se almacena en la variable factorial
   2. Se incrementa en uno el valor de la variable contador.
   3. Regresar al punto 4
6. **Si la variable contadora no es menor o igual al número entero se muestra el resultado almacenado en la variable factorial.**
7. **Fin del algoritmo.**

***Pruebas de escritorio.***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***iteración*** | ***X*** | ***Factorial*** | ***Contador*** | ***Salida*** |
| ***1*** | ***0*** | ***1*** | ***2*** | ***El factorial de 0 es 1*** |
| ***2*** | ***2*** | ***1*** | ***3*** | ***El factorial de 2 es 2*** |
| ***3*** | ***3*** | ***6*** | ***4*** | ***El factorial de 3 es 6*** |

***Valor crítico.***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***iteración*** | ***X*** | ***Factorial*** | ***Contador*** | ***Salida*** |
| ***1*** | ***-2*** | ***1*** | ***2*** | ***Valor no valido*** |

***Tarea.***

1. ***describe las salidas de los ejercicios finales y la comparación entre ambos algoritmos., realicen una propuesta para un óptimo algoritmo.***

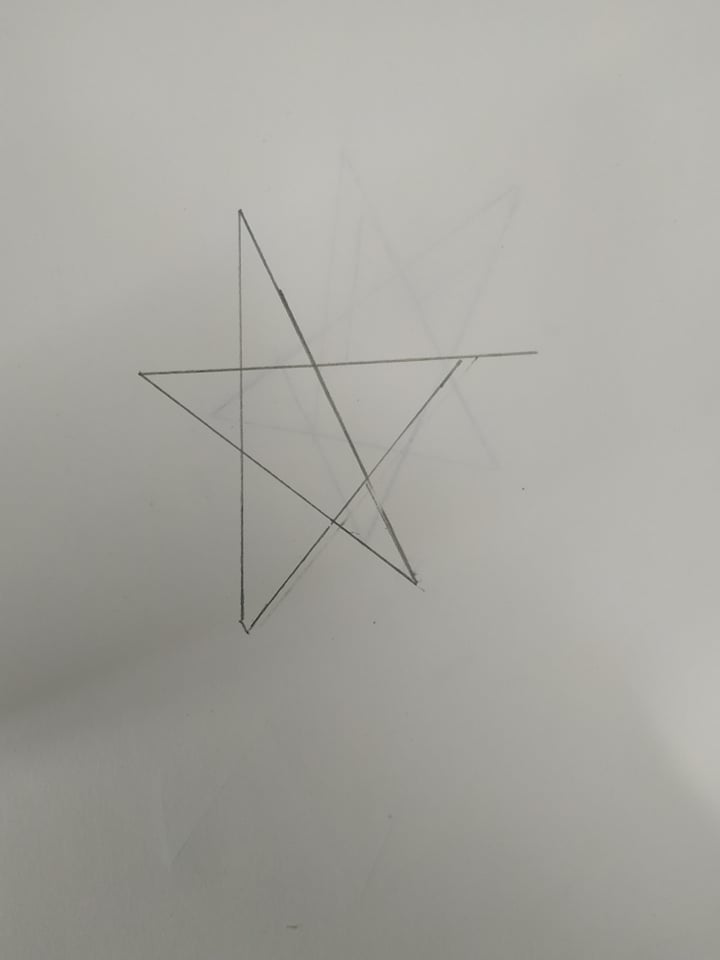
***Problema:*** Seguir el algoritmo para obtener una figura

***Datos de Entrada:*** Hoja tamaño carta en limpio, regla y lápiz.

***Datos de salida:*** Figura correcta.

***Algoritmo.***

1. Dibuja una V invertida. Empieza desde el lado izquierdo, sube, y baja hacia el lado derecho, no levantes el lápiz.
2. Ahora dibuja una línea en ángulo ascendente hacia la izquierda. Debe cruzar la primera línea más o menos a 1/3 de la altura. Todavía no levantes el lápiz del papel.
3. Ahora, dibuja una línea horizontal hacia la derecha. Debe cruzar la V invertida más o menos a 2/3 de la altura total. Sigue sin levantar el lápiz.
4. Dibuja una línea en un ángulo descendente hasta el punto de inicio. Las líneas deben unirse.
5. Ahora ya puedes levantar el lápiz del papel. Has terminado la estrella de 5 puntas



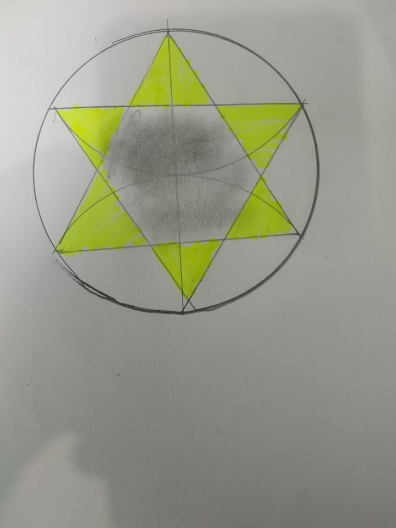
***Problema***: seguir el algoritmo para obtener una figura

***Datos de entrada***: Hoja tamaño carta en limpio, regla y lápiz.

***Datos de Salida***: Figura correcta.

***Algoritmo.***

1. Empieza dibujando un círculo con un compás. Coloca un lápiz en el compás. Coloca la punta del compás en el centro de una hoja de papel.
2. Ahora gira el compás, mientras mantienes la punta apoyada en el papel. El lápiz dibujará un círculo perfecto alrededor de la punta del compás.
3. Marca un punto en la parte superior del círculo con el lápiz. Ahora, coloca la punta del compás en la marca. No cambies el radio del compás con que hiciste el círculo.
4. Gira el compás para hacer una marca en el propio círculo hacia la izquierda. Haz una marca también en el lado derecho.
5. Ahora, coloca la punta del compás en uno de los puntos. Recuerda no cambiar el radio del compás. Haz otra marca en el círculo.
6. Continúa moviendo la punta del compás a las otras marcas, y continúa hasta que tengas 6 marcas a la misma distancia unas de otras. Ahora, ya puedes dejar tu compás a un lado.
7. . Usa una regla para crear un triángulo que empiece en la marca superior del círculo. Coloca el lápiz en la marca superior. Ahora dibuja una línea hasta la segunda marca por la izquierda. Dibuja otra línea, ahora hacia la derecha, saltándote la marca de la parte más baja. Complementa el triángulo con una línea hacia la marca superior. Así completarás el triángulo.
8. Crea un segundo triángulo empezando en la marca en la base del círculo. Coloca el lápiz en la marca inferior. Ahora conéctala con la segunda marca hacia la izquierda. Dibuja una línea recta hacia la derecha, saltándote el punto superior. Completa el segundo triángulo dibujando una línea hasta la marca en la parte inferior.
9. . Borra el círculo. Has terminado de dibujar tu estrella de 6 puntos.

******

***Crear un algoritmo optimo***

***Problema:*** *Dibujar un circulo*

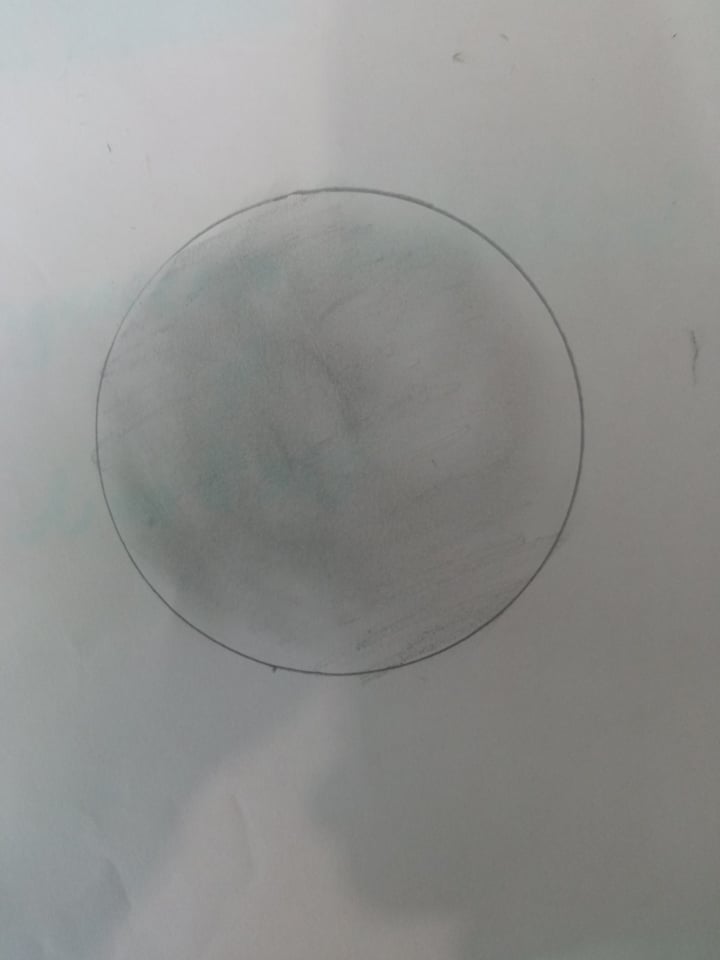
***Datos de entrada:*** *Necesitamos compas, lápiz, papel y las medidas del circulo*

***Restricciones:*** *La figura debe de ser un circulo*

***Dato de salida****: Circulo.*

***Algoritmo***

1. ***Inicio***
2. ***Coloque el lápiz en el compás***
3. ***Decida la medida de su circunferencia***
4. ***Coloque la punta metálica del Compas en el centro del papel***
5. ***Realice un movimiento circular sin despegar la punta metálica del papel***
6. ***Al terminar quite la punta metálica del papel***
7. ***Rellene el circulo***
8. ***Fin del algoritmo.***

******

1. ***¿Qué se necesita para dar solución a un problema?***

¿Que se necesita para la solución de un problema?

1. Necesitamos definir de forma clara el problema.
2. Definir las causas del problema que queremos resolver.
3. Pensar en posibles soluciones. Formas de resolver las causas que generan el problema.
4. Evaluar y crear una forma de actuar eficaz. Elaborar un plan
5. Implementar el plan.
6. Evaluarlos resultados.

Si nos damos cuenta, la forma de resolver un problema es la misma forma mediante la cual nosotros creamos algoritmos.

1. **Describe las fases del ciclo de vida del desarrollo de software explicadas en clase e ilustra**

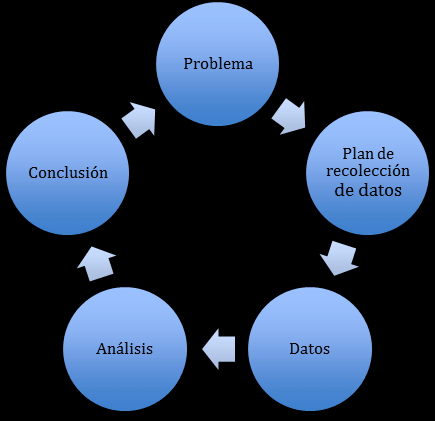
***¿Pare empezar que es el ciclo de vida de un software?***

El ciclo de vida del software contempla todas las fases por las cuales deberá de pesar un software para lograr ser validado como un producto eficaz, y que sea sustentable.

Es de suma importancia que el ciclo de vida se realice de forma adecuada puesto a que, de no ser así, el software podría presentar varios fallos que impedirían su correcto funcionamiento, e implicarían un gran costo para su reparación.

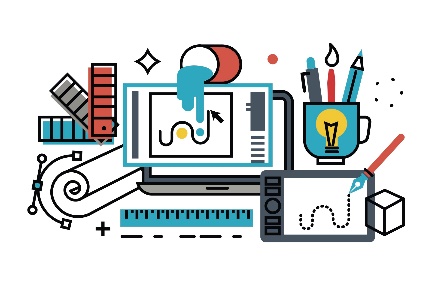
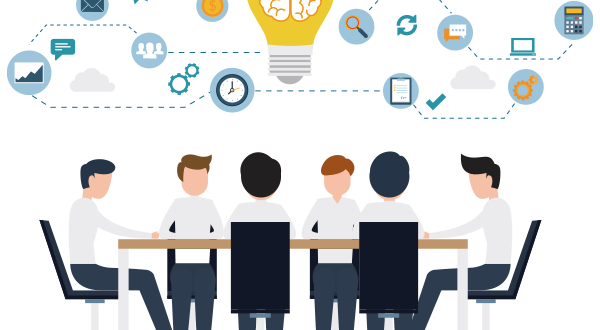
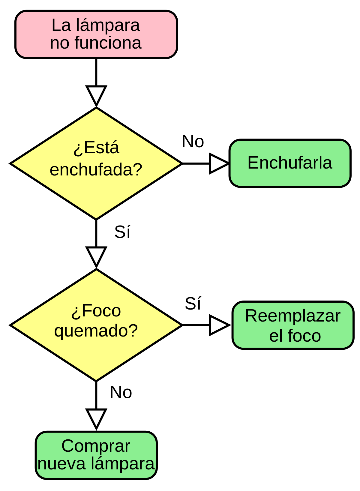
***Fases del ciclo de vida del software.***

***Análisis:*** Consiste en señalar específicamente los problemas para los cuales se diseñará el software, para de esta forma conseguir definir las especificaciones que el sistema va a requerir.



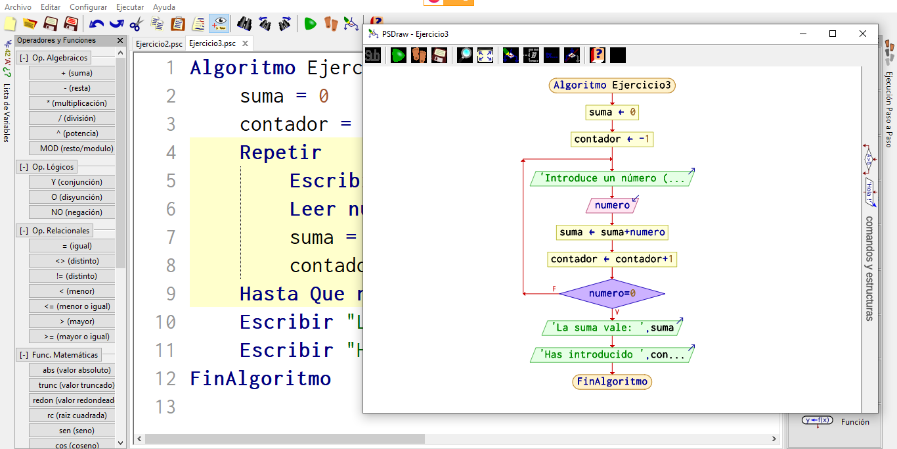


**Diseño**: En esta fase se estudian las posibles estructuras que el software poseerá con el fin de lograr cumplir su tarea, durante esta fase se crean los posibles algoritmos y se refinan para lograr conseguir el más eficaz, aumentando el coste beneficio.



***Implementacion.***

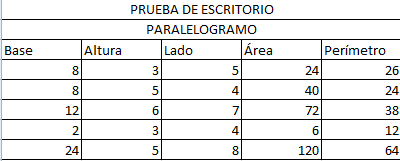
Durante esta etapa, se selecciona el lenguaje de programacion y el algoritmo mas adecuado para el tipo de sofware que vamos a contruir, tras esto se inicia el proceso de programacion, durante esta etapa igual se deberan de desarollar una serie de pruebas controladas para ver que el sofware sigua su camino organicamente y no presente fallos que mas adelante causes conflicto.



***Pruebas***:

Durante esta etapa, se busca encontrar fallos que se pudieron haber generado en las etapas anteriores, para ello se utiliza el software en un entorno controlado, esto se asemeja mucho a la fase beta de un videojuego, donde se presenta un previo para poder ver el funcionamiento y los aspectos mejorables a la par que los aspectos que requieran volver a plantearse.

A qui entrarían las pruebas de escritorio.

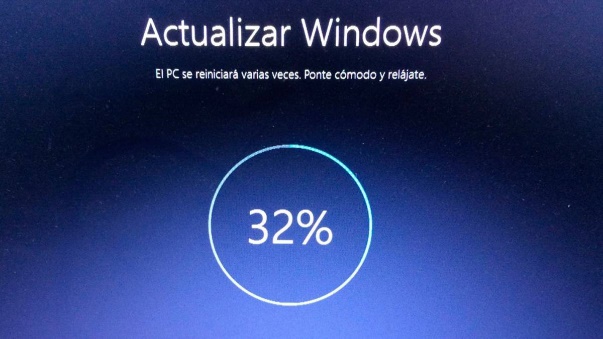
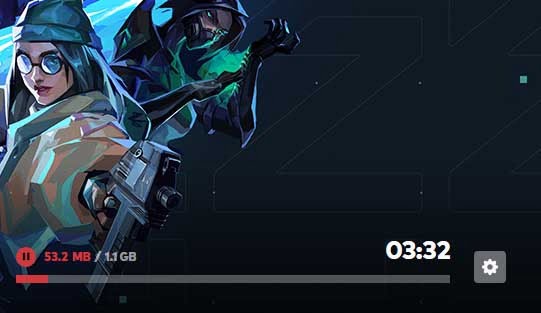
***Instalacion o despliegue***

Durante esta etapa se prepara todo para poder poner en marcha al sofware, aquí se consideran aspectos como la compatibilidad con dispositivos y se generan las especificaciones requeridas para que el soware funcione de forma correcta.



**Mantenimiento**: Esta es una de las etapas mas importantes del sofware, pues el sofware no se debe desgastar con el tiempo pues debe mantenerse funcional el maximo de tiempo posible, para ello se requiere de un buen mantenimiento el cual tiene como principo 3 pilares.

1. Eliminar los defectos detectados durante su vida útil (mantenimiento correctivo).
2. Adaptarlo a nuevas necesidades (mantenimiento adaptativo).
3. Añadirle nuevas funcionalidades (mantenimiento perfectivo).



1. ***Analizar las  siguientes problemáticas, desglosando y dando solución al problema identificando claramente su conjunto de entrada (datos de entrada), el conjunto de salida (datos de salida) y restricciones, si es que tiene.***
2. ***En una playa el estacionamiento cobra $ 2.5 por hora o fracción. Determinar cuánto debe pagar un cliente por el estacionamiento de su vehículo, conociendo el tiempo de estacionamiento en horas y minutos.***

***Datos de entrada:*** se cobra 2.5 la hora o fracción, una hora contiene 60 minutos, para convertir minutos a horas, se dividen los minutos entre 60

***Restricciones:*** una hora no puede tener menos de 60 minutos, el precio no puede cambiar de 2.5 la hora

***Datos de salida****: Total a pagar*

***Algoritmo***

1. ***Inicio***
2. ***Solicitar minutos al usuario***
3. ***Leer minutos***
4. ***Si los minutos son menores a 60, leer “total a pagar 2.5” e ir al paso 9***
5. ***SI los minutos son mayores a 60 entoces dividir los minutos entre 60***
6. ***Leer resultado como horas***
7. ***Si las horas son mayores a 1 multiplicar 2.5 x horas***
8. ***Leer total a pagar***
9. ***Fin del algoritmo.***

***Pruebas de escritorio***

|  |  |  |  |
| --- | --- | --- | --- |
| ***Iteracion*** | ***Horas*** | ***Coste*** | ***Total a pagar*** |
| ***1*** | ***2 horas*** | ***2.5 x 2*** | ***5*** |
| ***2*** | ***1 hora*** | ***2.5*** | ***2.5*** |
| ***3*** | ***30 minutos*** | ***2.5*** | ***2.5*** |

1. ***Dados dos números realizar suma, resta, producto y división (considerar  el dividendo  ≠ 0).***

***Datos de entrada:*** Los números deben ser enteros

***Restricciones:*** *Los números no pueden ser menores a 0, el dividendo no puede ser 0, por lo que el 2 valor no puede ser 0,*

***Datos de salida:*** resultado de la suma, resta, multiplicación y división

***Algoritmo***

1. ***Inicio***
2. ***Solicitar primer número al usuario***
3. ***Leer número 1.***
4. ***Solicitar 2 numero al usuario si el numero es = 0 regresar al paso 2 de lo contrario ir al paso 5***
5. ***Leer numero 2***
6. ***Sumar numero 1 + numero 2***
7. ***Leer la suma es igual a (1 numero + 2 numero)***
8. ***Restar numero 1 – numero 2***
9. ***Leer la resta es igual a (numero 1 – numero 2)***
10. ***Multiplicar numero 1 – numero 2***
11. ***Leer el producto es igual a (numero 1 \* numero 2)***
12. ***Dividir numero 1 entre numero 2***
13. ***Leer el cociente es igual a (numero 1 / numero 2)***
14. ***Fin del algoritmo.***

***Pruebas de escritorio.***

***Realice las pruebas por separado para mayo comodidad.***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***iteración*** | ***Primer numero*** | ***Segundo numero*** | ***Operación*** | ***Salida*** |
| ***1*** | ***5*** | ***3*** | ***(1 numero + 2 numero)*** | ***8*** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Iteracion*** | ***Primer numero*** | ***Segundo Numero*** | ***Operación*** | ***Salida*** |
| ***1*** | ***4*** | ***2*** | ***(1 numero – 2 numero)*** | ***2*** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Iteracion*** | ***Primer numero*** | ***Segundo Numero*** | ***Operación*** | ***Salida*** |
| ***1*** | ***5*** | ***3*** | ***(1 numero \* 2 numero)*** | ***15*** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Iteracion*** | ***Primer numero*** | ***Segundo Numero*** | ***Operación*** | ***Salida*** |
| ***1*** | ***10*** | ***5*** | ***(1 numero / 2 numero)*** | ***2*** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Iteacion*** | ***Primer numero*** | ***Segundo Numero*** | ***Operación*** | ***Salida*** |
| ***1*** | ***10*** | ***0*** | ***(1 numero / 2 numero)*** | ***-------------***  ***Valor no valido*** |

1. ***Dados tres números; si el primero es negativo, deberá realizar el producto de los tres, y en caso contrario realizará  la suma.***

***Datos de entrada:*** *3 números enteros*

***Restricciones:*** *Los números deben de ser enteros.*

***Datos de salida****: suma o multiplicación de los tres números*

***Algoritmo****.*

1. ***Inicio***
2. ***Solicitar primer número,***
3. ***Leer primer numero***
4. ***Solicitar 2 numero.***
5. ***Leer segundo numero***
6. ***Solicitar 3 numero***
7. ***Leer tercer numero***
8. ***Si el primer numero es menor a 0 entonces***
   1. *Multiplicar (1 numero\* 2 número\* 3 numero)*
   2. *Leer “el producto es igual a (1 numero \*2 numero\* 3 numero)”*
   3. *Ir al paso 10*
9. ***Si el 1 numero no es menor a 0, entonces*** 
   1. *Sumar (1 numero + 2 numero + 3 numero)*
   2. *Leer “la suma es igual a (1 numero + 2 numero + 3 numero)”*
   3. *Ir al paso 10*
10. ***Fin del algoritmo****.*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Iteracion*** | ***1 numero*** | ***2 numero*** | ***3 numero*** | ***Operación*** | ***Salida*** |
| ***1*** | ***-2*** | ***3*** | ***5*** | ***(1 numero\* 2 numero\* 3 numero)*** | ***-30*** |
| ***2*** | ***2*** | ***4*** | ***3*** | ***(1 numero + 2 numero + 3 numero)*** | ***9*** |
| ***3*** | ***1*** | ***0*** | ***3*** | ***(1 numero + 2 numero + 3 numero)*** | ***4*** |

***Referencias bibliograficas***

S, A. (junio de 2015). *Ingeniería de Software.* Obtenido de http://ing-software-verano2014.blogspot.com/

FACULTAD DE INGENIERIA. (09 de 2021). *Laboratorio de computacion Sala A y B*. Obtenido de <http://lcp02.fi-b.unam.mx/>

A. (2019, 30 marzo). *¿Qué Es Un Algoritmo En Programación Y Para Qué Sirve?* Avantel LTpro. https://www.avantel.co/blog/tecnologia/que-es-un-algoritmo-programacion-para-que-sirve/

***Link de Gitup***