

Elabora un pseudocódigo que solicite una cantidad de dólares y los convierta a pesos mexicanos

Algoritmo

cant_usd = 0

mxn = 0

tc = 0

Escribir "Ingresa la cantidad de dólares"

Leer cant_usd

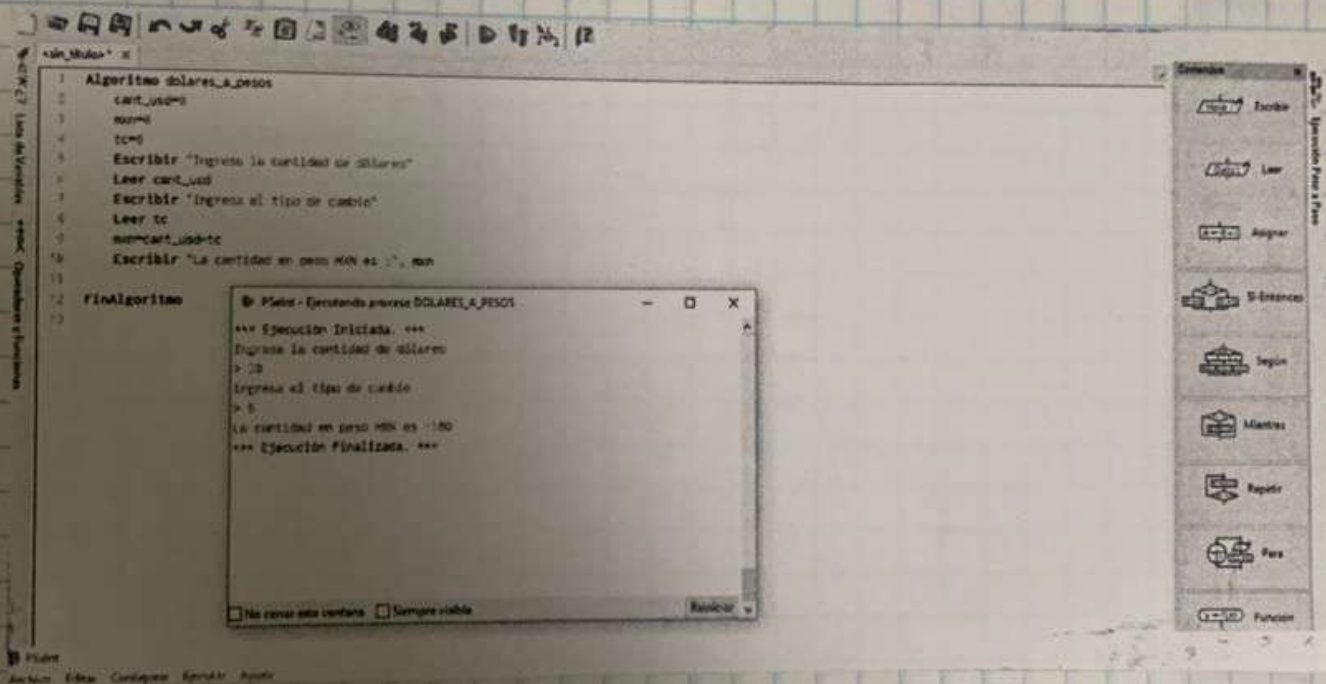
Escribir "Ingresa el tipo de cambio"

Leer tc

$mxn = cant_usd * tc$

Escribir "La cantidad en pesos MXN es :", mxn

FinAlgoritmo



Algoritmo salarios

salario = 0, aumento = 0;

Definir Opción como entero;

Escribir "selecciona tu puesto";

Escribir "1. sindicalización (20%)";

Escribir "2. de confianza (10%)";

Escribir "3. alto directivo (5%)";

Escribir "4. ejecutivo (0%)";

Leer Opción;

Escribir "Ingresa tu salario actual";

Leer salario;

Según Opción Hacer

1.

aumento = salario * 0.20;

2.

aumento = salario * 0.10;

3.

aumento = salario * 0.05;

4.

aumento = salario * 0;

Fin Según

Escribir "Tu aumento es: \$", aumento;

Fin Algoritmo

Algoritmo Empresa

```

1  salario=0;
2  aumento=0;
3  Definir opciones Como Entero;
4  Escribir "selecciona tu puesto";
5  Escribir "1. Sindicalización(20%)";
6  Escribir "2. De confianza(10%)";
7  Escribir "3. Alto directivo(5%)";
8  Escribir "4. Ejecutivo(0%)";
9  Leer opciones;
10 Escribir "Ingresa tu salario actual:";
11 Leer salario;
12
13 Según opciones Hacer
14     1: aumento=salario*0.20;
15     2: aumento=salario*0.10;
16     3: aumento=salario*0.05;
17     4: aumento=salario*0;
18 FinSegún
19
20 Escribir "Tu aumento es: $", aumento;
21 FinAlgoritmo

```

PSeint - Ejecutando proceso EMPRESA

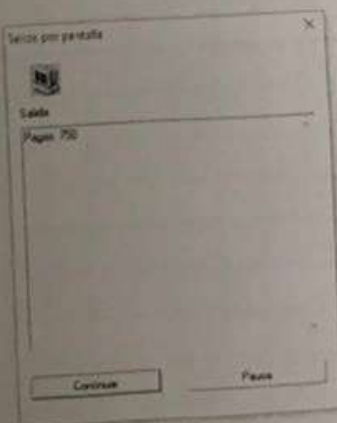
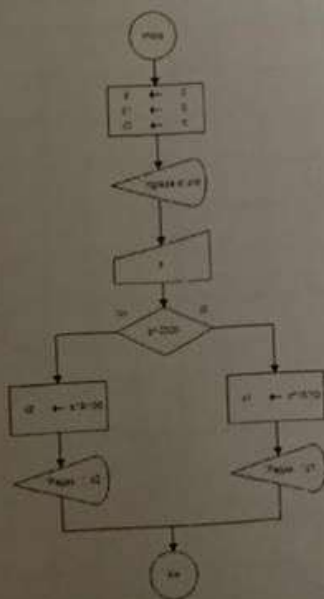
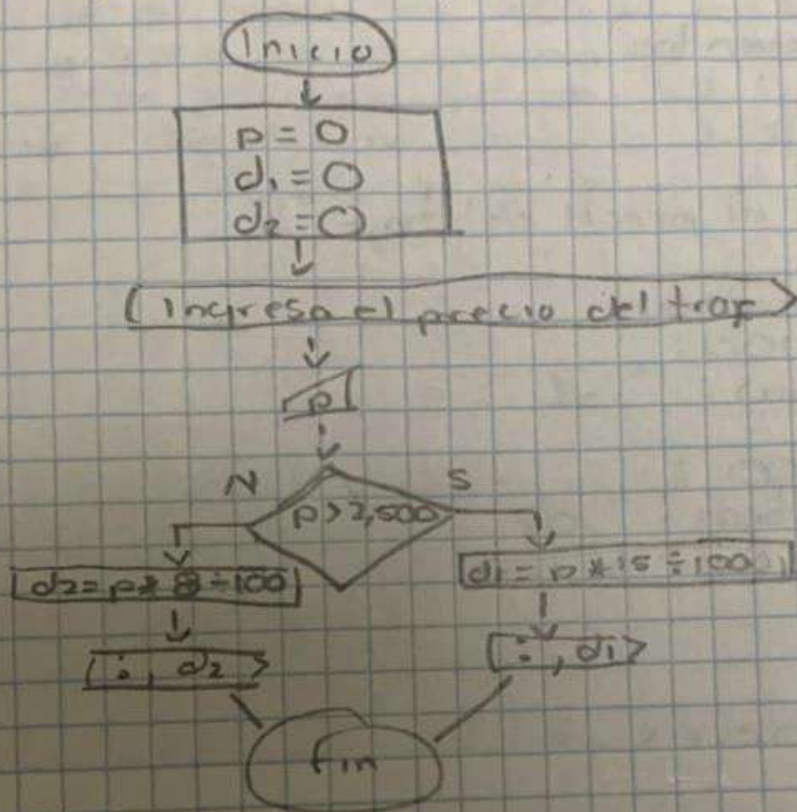
```

*** Ejecución Iniciada ***
selecciona tu puesto
1. Sindicalización(20%)
2. De confianza(10%)
3. Alto directivo(5%)
4. Ejecutivo(0%)
> 1
Ingresa tu salario actual:
> 800
Tu aumento es: $160
*** Ejecución Finalizada. ***

```

☐ No cerrar esta ventana ☐ Siempre visible Reiniciar

Almacenes el oro pinto distinguido tiene una promoción a todos los trajes que tienen un precio superior a 2,500\$. Se les aplicará un descuento de 15%, a todos los demás se les aplicará solo el 8%. Elabora diagrama de flujo y pseudocódigo que resuelvan el problema.



Para los requerimientos se usan los siguientes módulos

- Usuarios: Son personas que se conectan al sistema para hacer uso de los servicios que este les proporciona.
- Productos: Registra y gestiona los productos en venta con detalles como precio, código y categoría.
- Proveedores: Maneja la información de los proveedores que surten los productos.
- Inventario: Controla las existencias, entradas y salidas de productos.
- POS (Producto de Venta): Procesa las ventas, calcula totales, genera tickets y acepta pagos.
- Reportes: Genera informes sobre ventas, inventario y actividad del sistema.
- Tipos de Usuario: Define roles y permisos dentro del sistema (administrador, vendedor).

Segunda etapa

Diseño rápido: Esquema inicial del sistema

Primera pantalla: (Login) usuario y contraseña con botón de entrar.

Segunda pantalla: Menú principal con altas y bajas, consultas, punto de venta e inventario.

Tercera etapa

Construcción del prototipo: Versión funcional

Login	Menú Principal
Usuario: <input type="text"/>	<input type="button" value="Altas"/> <input type="button" value="Bajas"/> <input type="button" value="POS"/>
Contraseña: <input type="password"/>	<input type="button" value="Consultas"/> <input type="button" value="Inventario"/>
<input type="button" value="Ingresar"/>	

Cuarta etapa.

Una pantalla de login es una interfaz que se utiliza para controlar el acceso a un sistema aplicación o sitio web.

Sirve para:

- Autenticar la identidad del usuario
- Controlar el acceso al sistema
- Proteger la seguridad y privacidad
- Asignar permisos y roles
- Registrar la actividad del usuario
- Proporcionar una experiencia personalizada
- Cumplir con regulaciones de seguridad y privacidad.

La pantalla login típica incluye:

- Campo para ingresar el nombre de usuario
- Campo para ingresar la contraseña
- Botón para iniciar sesión
- Enlace para recuperar la contraseña olvidada
- Enlace para crear una nueva cuenta (si corresponde)

Una pantalla login es una parte fundamental de la seguridad y el control de acceso en cualquier sistema, aplicación o sitio web.

Tener diferentes tipos de usuarios para acceder a un sistema informático sirve para varios propósitos.

Seguridad: Al asignar diferentes niveles de acceso, se puede controlar qué información y funcionalidades pueden acceder cada tipo de usuario, reduciendo el riesgo de acceso no autorizado o mal uso de la información.

Control de acceso: Los diferentes tipos de usuarios permiten controlar quién puede acceder a qué información y funcionalidades, lo que ayuda a mantener la integridad y la confidencialidad de la información.

Gestión de permisos: Al asignar permisos específicos a cada tipo de usuario, se puede controlar qué acciones pueden realizar cada usuario, lo que ayuda a prevenir errores o acciones no autorizadas.

Personalización: Los diferentes tipos de usuarios permiten personalizar la experiencia de usuario, mostrando solo la información y funcionalidades relevantes para cada tipo de usuario.

Auditoría y seguimiento: Al tener diferentes tipos de usuarios, se puede realizar un seguimiento de las acciones realizadas por cada usuario, lo que ayuda a identificar posibles problemas de seguridad o mal uso de la información.

Cumplimiento de regulaciones: En algunos casos, la asignatura de diferentes tipos de usuarios es requerida por regulaciones o leyes, como la Ley de Protección de Datos Personales.

Mejora de la productividad: Al asignar los permisos y acceso adecuados a cada tipo de usuario, se puede mejorar la productividad, ya que los usuarios pueden acceder a la información y funcionalidades que necesitan para realizar su trabajo de manera eficiente.

Tipos de usuarios en sistemas informáticos:

Administrador: tiene acceso total al sistema y puede realizar cambios y configuraciones.

Usuario estándar: tiene acceso a la información y funcionalidades básicas del sistema.

Usuario avanzado: tiene acceso a funcionalidades adicionales y puede realizar acciones más complejas.

Usuario de solo lectura: tiene acceso a la información del sistema, pero no puede realizar cambios.

Usuario invitado: tiene acceso limitado al sistema y puede realizar acciones específicas.

Resumen

Metodología del desarrollo de software: Es un marco de trabajo utilizado para estandarizar, optimizar y controlar el proceso de desarrollo de software, asegurando productos de alta calidad.

Metodología de prototipos: Se basa en la creación de prototipos funcionales para facilitar la comunicación con los usuarios y clientes permitiendo evaluar y mejorar el sistema antes de su desarrollo final.

Etapas del proceso:

1. **Recolección y análisis de requisitos:** Identificación de las necesidades básicas del sistema.
2. **Diseño rápido:** Creación de un esquema inicial del sistema.
3. **Construcción del prototipo:** Desarrollo de una versión funcional con características esenciales.
4. **Evaluación del prototipo:** Los usuarios prueban e interactúan con el prototipo, proporcionando retroalimentación.
5. **Refinamiento del prototipo:** Ajustes basados en la retroalimentación de los usuarios.
6. **Implementación del producto final:** Desarrollo completo del sistema con mejoras y validaciones finales.

Metodología de desarrollo de software y de prototipos (etapas)
La metodología de desarrollo de software es un enfoque estructurado y sistemático para diseñar, desarrollar, probar y mantener software. Estas metodologías proporcionan un marco de trabajo para los equipos de desarrollo de software, asegurando que el proceso de desarrollo sea eficiente, eficaz y de alta calidad.

Cada metodología tiene sus propias ventajas y desventajas, y la elección de la metodología adecuada dependerá del proyecto específico, del equipo de desarrollo y de las necesidades del cliente. La metodología de prototipos es un enfoque de diseño y desarrollo de software que se centra en la creación de prototipos para evaluar y refinar las ideas y requisitos del proyecto.

El objetivo principal de la metodología de prototipos es crear el modelo tangible y funcional del sistema o producto que se está desarrollando, para que los usuarios y stakeholders puedan interactuar con él y proporcionar retroalimentación.

1. Definición del problema: Identificar el problema o necesidad que se quiere resolver.

2. Creación del prototipo: Desarrollar un prototipo inicial del sistema o producto.

3. Evaluación del prototipo: Presentar el prototipo a los usuarios y stakeholders para recopilar retroalimentación.

4. Refinamiento del prototipo: Realizar cambios y ajustes en el prototipo basándose en la retroalimentación recibida.

5. Iteración: Repetir las etapas de evaluación y refinamiento hasta que el prototipo cumpla con los requisitos y expectativas.