**LABORATORIO NO. 03**

“Lenguajes y Paradigmas”

**PRÁCTICA EN LABORATORIO**

1. Realice un algoritmo y un diagrama de flujo para el siguiente problema:

Una empresa necesita calcular el monto del salario de tres empleados que pertenecen a la Gerencia General, para ello le solicitan realizar un programa que solicite al usuario la siguiente información: Nombre del empleado, Código del Empleado, Fecha de Ingreso a la Empresa, Salario Base, Bonificación de Ley, Número de Horas Extras Trabajadas y Descuentos en el mes.

En base a eso calcular lo siguiente:

* Monto de Horas Extras = Salario por hora \* 1.5 \* Número de Horas Extras.
* Salario por Hora = Salario Base/22/8
* Total de Ingresos = Salario Base + Bonificación de Ley + Monto de Horas Extras.
* IGSS = Salario Base \* 4.83%
* Total de Egresos = Descuentos en el mes + IGSS
* Salario Final = Total Ingresos – Total Egresos

Al usuario solo se le debe mostrar Total de Ingresos, Total de Egresos y el Salario Final.

**Análisis:**

**Entradas:**

* Nombre del empleado
* Código del Empleado
* Fecha de Ingreso a la Empresa
* Salario Base
* Bonificación de Ley
* Número de Horas Extras Trabajadas
* Descuentos en el mes.

**Salidas:**

* Total ingresos
* Total egresos
* Salario final

**Procesos:**

* Monto de Horas Extras = Salario por hora \* 1.5 \* Número de Horas Extras.
* Salario por Hora = Salario Base/22/8
* Total de Ingresos = Salario Base + Bonificación de Ley + Monto de Horas Extras.
* IGSS = Salario Base \* 4.83%
* Total de Egresos = Descuentos en el mes + IGSS
* Salario Final = Total Ingresos – Total Egresos

**Diseño:**

Program Calculo Empleados

{

String nombre, código, fecha;

Decimal salario, bonificación, horas\_extra, descuentos, total\_ingresos, total\_egresos, salario\_final, monto\_horas\_extra, salrio\_hora, igss;

Write (“Ingrese el nombre del empleado:”);

Read(nombre);

Write (“Ingrese el código del empleado:”);

Read(código);

Write (“Ingrese la fecha de ingreso del empleado:”);

Read(fecha);

Write (“Ingrese el salario base del empleado:”);

Read(salario);

Write (“Ingrese la bonificación de ley:”);

Read(bonificacion);

Write (“Ingrese el número de horas extra trabajadas:”);

Read(horas\_extra);

Write (“Ingrese los descuentos del mes”);

Read(descuentos);

Salario\_hora = salario/22/8;

Monto\_horas\_extra = salario\_hora \* 1.5 \* horas\_extra;

Total\_ingresos = Salario + bonificación + Monto\_horas\_extra;

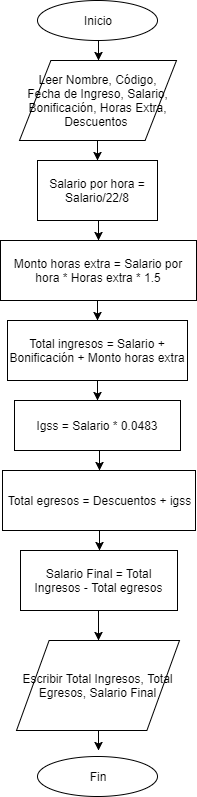
Igss = Salario \* 0.0483;

Total\_egresos= descuentos + igss;

Salario\_final = total\_ingresos – total egresos;

Write (“El total de ingresos del empleado es:” + total\_ingresos + “. El total de egresos del empleado es: “+ total\_egresos + “. Y el salario final del empleado es: ” + Salario final );

}



**TAREA**

1. Marque con una X la casilla que considere que coincide

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Lenguaje** | **Declarativo** | **Imperativo** | **Orientado a Objetos** | **Intérprete** | **Compilador** |
| C# |  |  | X |  | x |
| Java |  |  | X | x | x |
| Javascript |  | X | X | x |  |
| Basic |  | X |  |  | x |
| Fortran |  | X |  |  | x |
| PHP |  | X | X | x |  |
| Perl |  | X | X | x | x |
| Python |  | X | X | x |  |
| C |  | X |  |  | x |
| Haskell | x |  |  |  | x |
| SQL | x |  |  | x |  |
| Ruby |  |  | X | x |  |
| Pascal |  | X |  |  | x |

2. Realice el algoritmo y el diagrama de flujo del siguiente problema:

Se necesita determinar la proyección de gasolina que necesitará una Estación, por lo cual se sabe que un vehículo tipo Sedán llena en promedio 6 galones, SUV 12 galones, Motocicleta 4 galones, Pickup 12 galones, Camión 20 galones, Buses 20 galones.

Solicite al usuario la cantidad de vehículos Sedán, SUV, Motos, Pickup, Camión y Buses que llegaron en un día determinado, luego solicite cuántos días tiene el mes que desean proyectar.

Muestre al usuario el total de Galones de combustible que necesitan para abastecerse por ese mes.

**Análisis:**

**Entradas:**

* Cantidad sedán
* Cantidad Suv
* Cantidad motos
* Cantidad pickup
* Cantidad camiones
* Cantidad buses
* Días del mes

**Salidas:**

* Total galones

**Procesos:**

* Cantidad sedan \* 6
* Cantidad suv \* 12
* Cantidad motos \* 4
* Cantidad pickup \* 12
* Cantidad camión \* 20
* Cantidad bus \* 20
* Suma cantidades \* días del mes

**Diseño:**

Program Proyección del mes

{

Int Sedan, SUV, Pickup, Moto, Camion, Bus, Dia, Galones;

Write (“Ingrese la cantidad de vehículos sedán:”);

Read(Sedan);

Write (“Ingrese la cantidad de vehículos SUV:”);

Read(SUV);

Write (“Ingrese la cantidad de Motos:”);

Read(Moto);

Write (“Ingrese la cantidad de Pickup:”);

Read(Pickup);

Write (“Ingrese la cantidad de Camiones:”);

Read(Camion);

Write (“Ingrese la cantidad de Buses:”);

Read(SUV);

Write (“Ingrese la cantidad de días del mes:”);

Read(Dia);

Sedan = Sedan \* 6;

SUV = SUV \* 12;

Moto = Moto \* 4;

Pickup = Pickup \* 12;

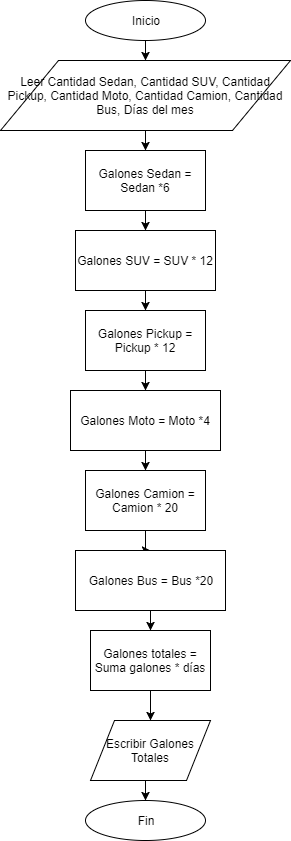
Camion = Camion \* 20;

Bus = Bus \* 20;

Galones = (Sedan + SUV + Moto + Pickup + Camion + Bus)\*Dia;

Write(“La cantidad de combustible en galones que se necesita para el mes es de: ” + Galones + “ galones”);

}



3. Realice el algoritmo y diagrama de flujo del siguiente programa

Su familia está pensando en realizar la inversión en adquirir una casa nueva, por lo cual le solicitan que haga un programa que permita saber el monto de las mensualidades que deben pagar.

El programa deberá solicitarle al usuario el monto actual del valor de la casa (Ejemplo: Q. 400,000)

También la tasa de interés anual (Ejemplo: 7%)

Finalmente debe ingresar el número de años que durará el crédito (Ejemplo: 20 años)

El programa debe ser capaz de devolver el monto mensual que se debe pagar con la siguiente fórmula:



Donde Pn es el monto del valor de la casa, i es la tasa de interés (mensual) y n es el número de meses que durará el crédito.

Nota: Sea cuidadoso porque los datos al usuario se le piden en años, pero se deben mostrar el resultado en meses, haga las conversiones necesarias.

**Análisis:**

**Entradas:**

* Valor actual de la casa
* Tasa de interés anual
* Años que durará el crédito

**Salidas:**

* Monto mensual a pagar

**Procesos:**

* Convertir interés anual a interés mensual
* Convertir años que durará el crédito a meses
* (Valor de la casa \* interés mensual) / (1 - (1 + interés mensual)) ^ (-meses)

**Diseño:**

Program Interes

{

Double Valor\_casa, Interes\_anual, Años, Meses, Interes\_mensual, Monto\_Mensual ;

Write (“Ingrese el valor de la casa:”);

Read(Valor\_casa);

Write (“Ingrese el porcentaje de interés anual:”);

Read(Interes\_anual);

Write (“Ingrese el número de años que durará el crédito:”);

Read(Años);

Interes\_anual = Interes anual/100

Interes\_mesual = Interes\_anual / 12;

Meses = Años \* 12;

Monto\_Mensual = (Valor\_casa \* Interes\_mensual) / (1 – Math.Pow( (1 + Interes\_mesual), (- Meses) );

Write (“El monto a pagar por mes es de: ” + Monto\_Mensual);

}

