

UNIVERSIDAD ESTATAL DE BOLIVAR

**FACULTAD DE CIENCIAS ADMINISTRATIVAS, GESTION EMPRESARIAL E INFORMATICA**

**SOFTWARE**

# TEMA:

# TRABAJO GRUPAL

# ASIGNATURA:

ALGORITMOS Y LOGICA DE PROGRAMACION

# DOCENTE:

# MSC. ENRIQUE BAÑO LEON

# PARALELO/MODALIDAD/:

A-MATUTINA

# GRUPO 6

# AUTORES:

* ARIEL CALDERON
* DAVID AVILEZ
* XIOMARA PUNINA
* DERLIN GARCIA

# GUARANDA-BOLIVAR-ECUADOR

# PII 2023

**Trabajo Grupal De Algoritmos y Lógica de Programación**

**Objetivo:** Crear un algoritmo el cual de solución al problema plantado

**Problema:** Una empresa paga a sus 100 operarios semanalmente, de acuerdo con el número de horas trabajadas, a razón de P pesetas por hora y de 1.5.P pesetas por cada hora extra. Las horas extras son las que exceden de 40h. Hacer el organigrama que dado el número de horas trabajadas y el valor de P calcule el salario que le corresponde.

1. **Entender el problema**

El problema planteado en general busca según la cantidad de horas trabajadas determinar el tipo de horas trabajadas por los trabajadores de una empresa y en base a ese dato determinar su salario en pesetas.

Las entradas de datos serian el nombre del trabajador, y el número de horas trabajadas, mientras que las salidas son el valor del salario; la variable es el número de horas trabajadas.

1. **Análisis del problema**

Las partes de problema son la lectura de datos de entrada, la estructura de datos, las fórmulas a usar, la salida de datos.

Dentro de la lectura de datos de entrada se pedirá que se ingrese su nombre y la cantidad de horas de trabajo, la estructura de datos a usar es la condicional, se va a usar dos formulas una si las horas de trabajo son menores o iguales a 40horas y otra si las horas de trabajo son mayores a 40, finalmente en la salida de datos se aplicara el comando escribir para mostrar el salario

1. **Diseñar la solución**

El primer paso para resolver este problema es la declaración de variables , después como segundo paso es necesario realizar el ingreso de las entradas de datos ; lo principal es determinar el número de horas trabajadas para lo cual utilizaremos la estructura de datos condicional ( si entonces) para determinar si el número de horas trabajadas se pueden considerar horas extras , si el número de horas trabajadas es menor a 40 horas se considerara al salario por la siguiente formula (salario= horas de trabajo \* 1) , mientras que si se exceden las 40 horas el salario se aplicara la siguiente formula( salario horas extras=horas de trabajo\*1.5) , según esta asignación se mostrara el salario correspondiente.

1. **Escribir el seudocódigo**

Algoritmo CalcularSalario

// Declaración de variables

Definir horas\_trabajadas, valor\_pesetas, salario\_base, horas\_extra, salario\_total Como Entero

// Solicitar al usuario el número de horas trabajadas

Escribir "Ingrese el número de horas trabajadas:"

Leer horas\_trabajadas

// Solicitar al usuario el valor de Pesetas

Escribir "Ingrese el valor de Pesetas por hora:"

Leer valor\_pesetas

// Verificar si las horas trabajadas son menores o iguales a 40

Si horas\_trabajadas <= 40 Entonces

// Calcular el salario base sin horas extra

salario\_base = horas\_trabajadas \* valor\_pesetas

// No hay horas extra trabajadas

horas\_extra = 0

Sino

// Calcular el salario base considerando 40 horas

salario\_base = 40 \* valor\_pesetas

// Calcular las horas extra trabajadas

horas\_extra = horas\_trabajadas - 40

FinSi

// Calcular el salario total considerando las horas extra

salario\_total = salario\_base + (horas\_extra \* 1.5 \* valor\_pesetas)

// Mostrar el salario total

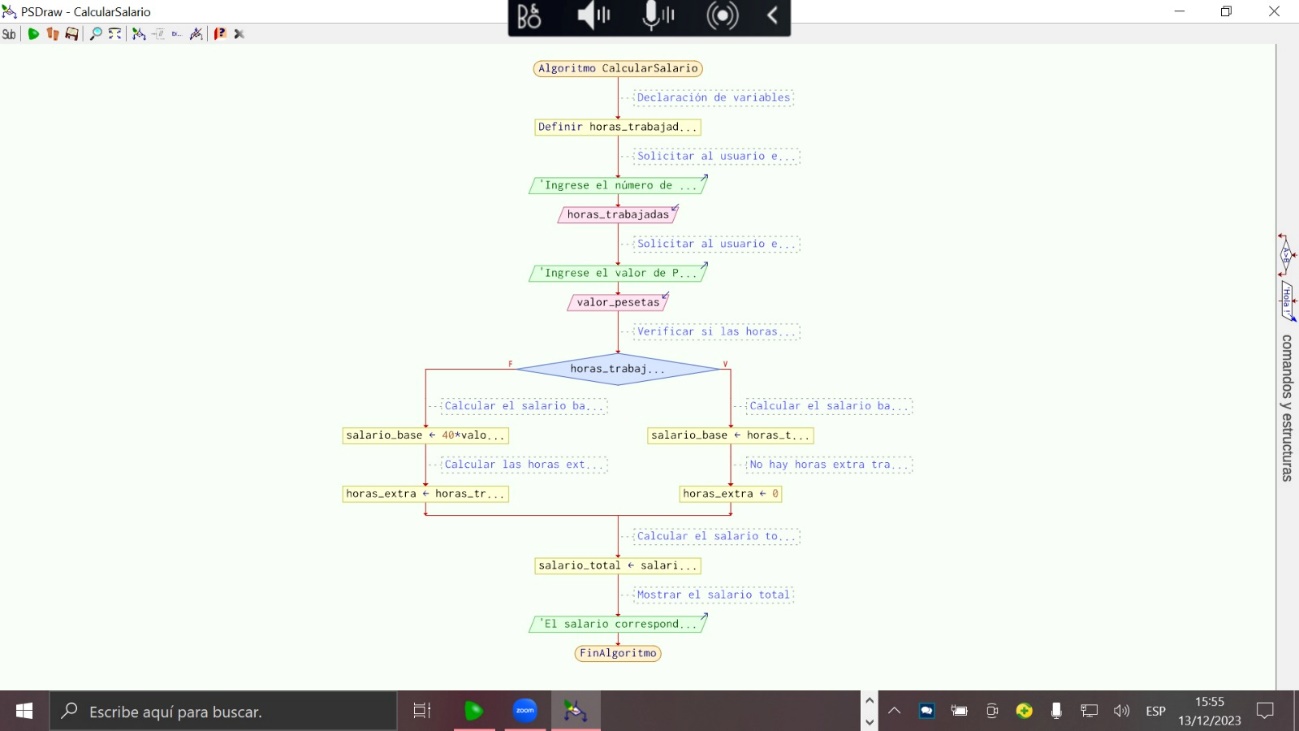
Escribir "El salario correspondiente es:", salario\_total

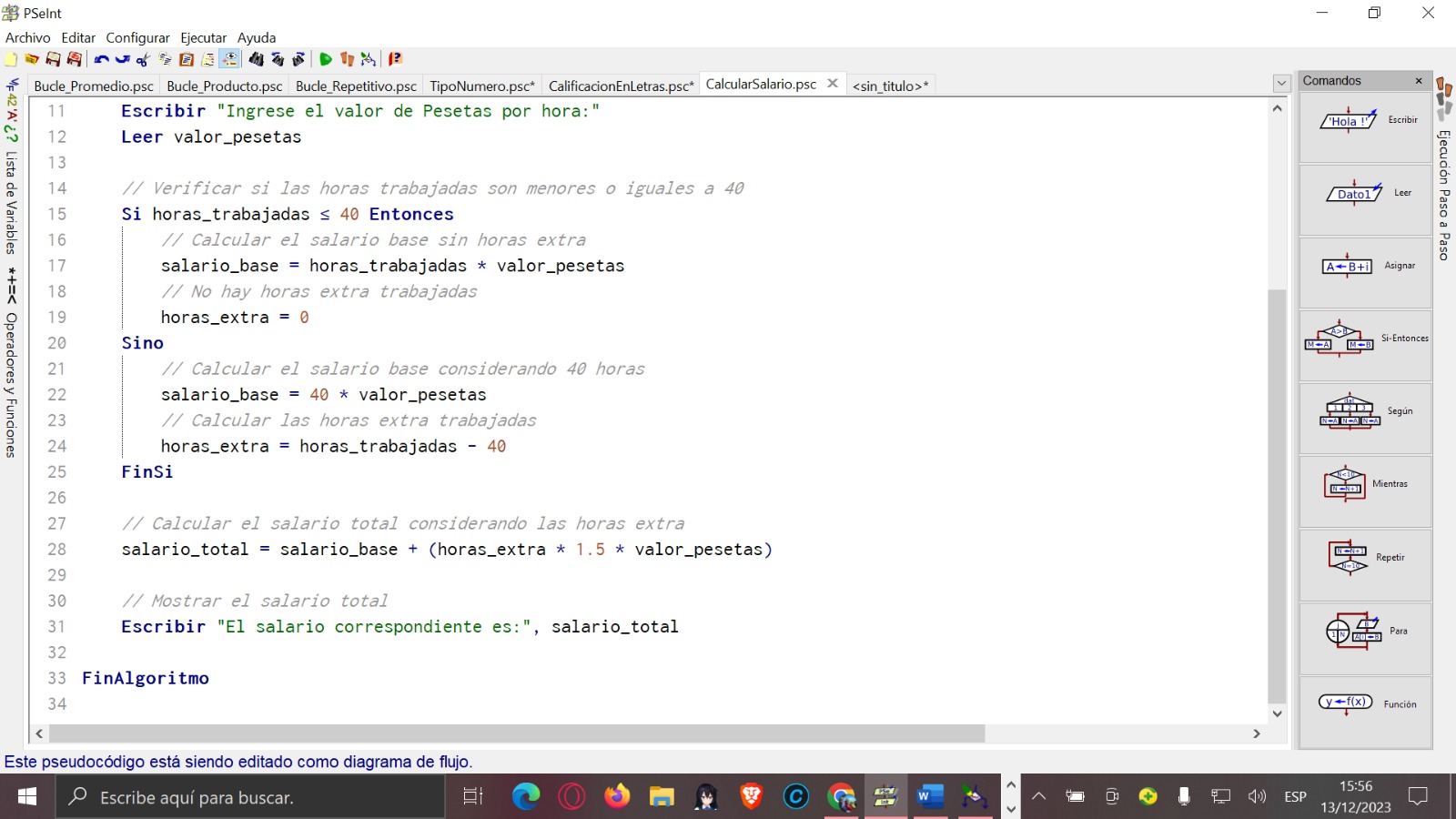
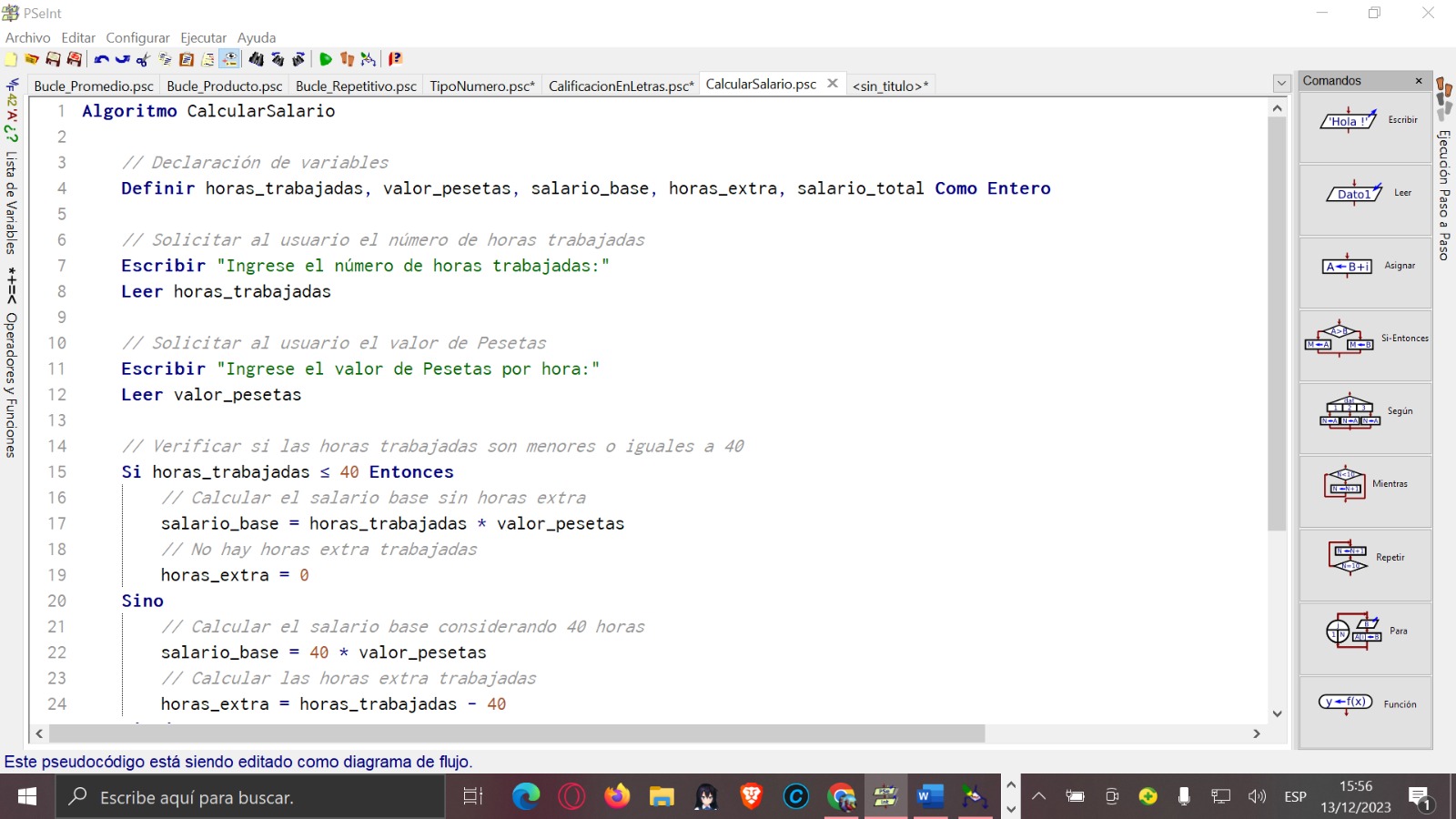
FinAlgoritmo

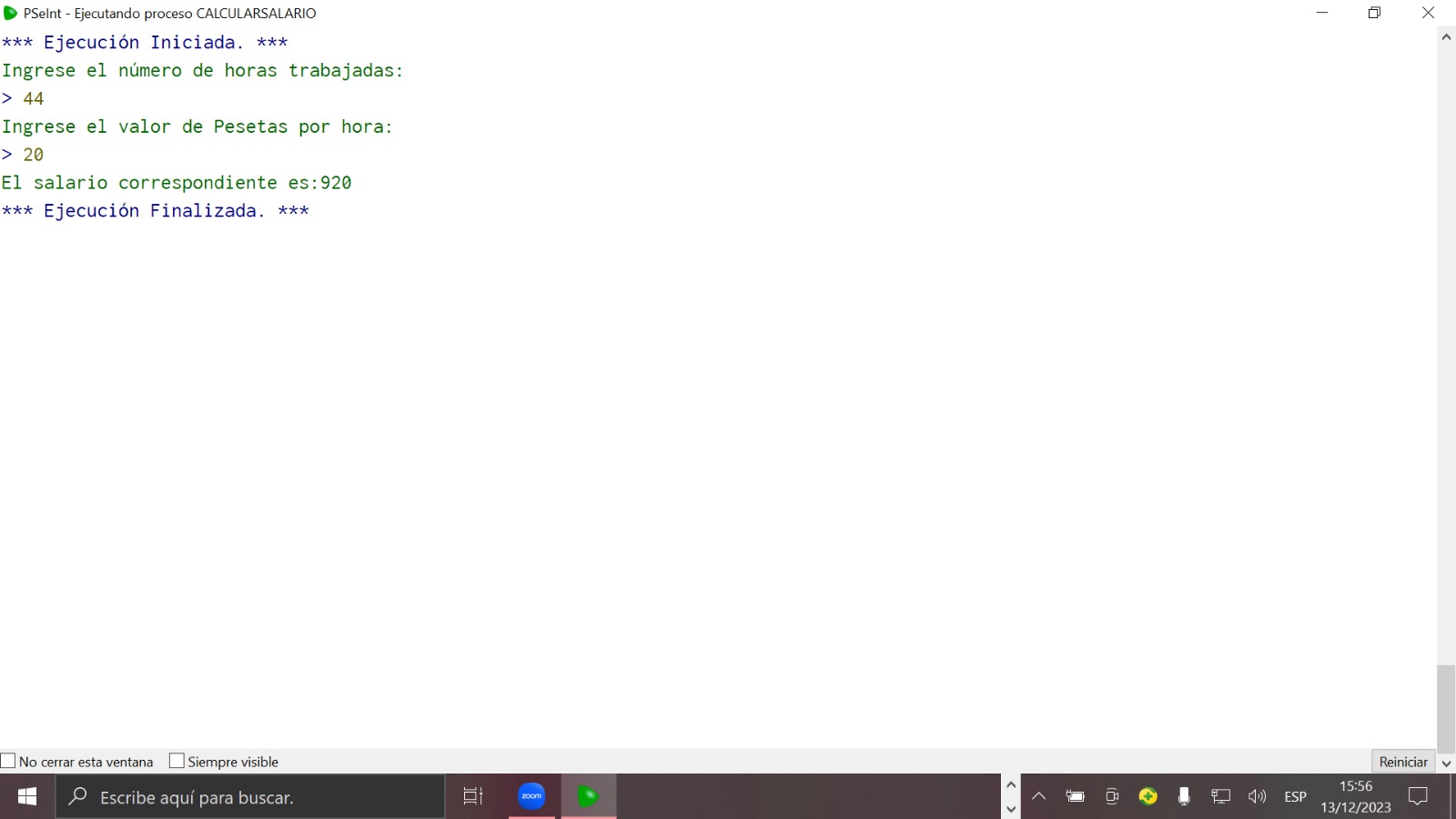
1. **Revisar y refinar**

En esta parte se reviso que todas las instrucciones de este algoritmo se encuentren correctamente ordenadas, además que tengan todos los signos correctamente ubicados, que no haya faltas ortográficas en la redacción de los procesos y que el algoritmo cumpla lo que solicita el problema planteado.

1. **Construir el diagrama de flujo**



1. **Implementar el código**



1. **Probar y depurar**