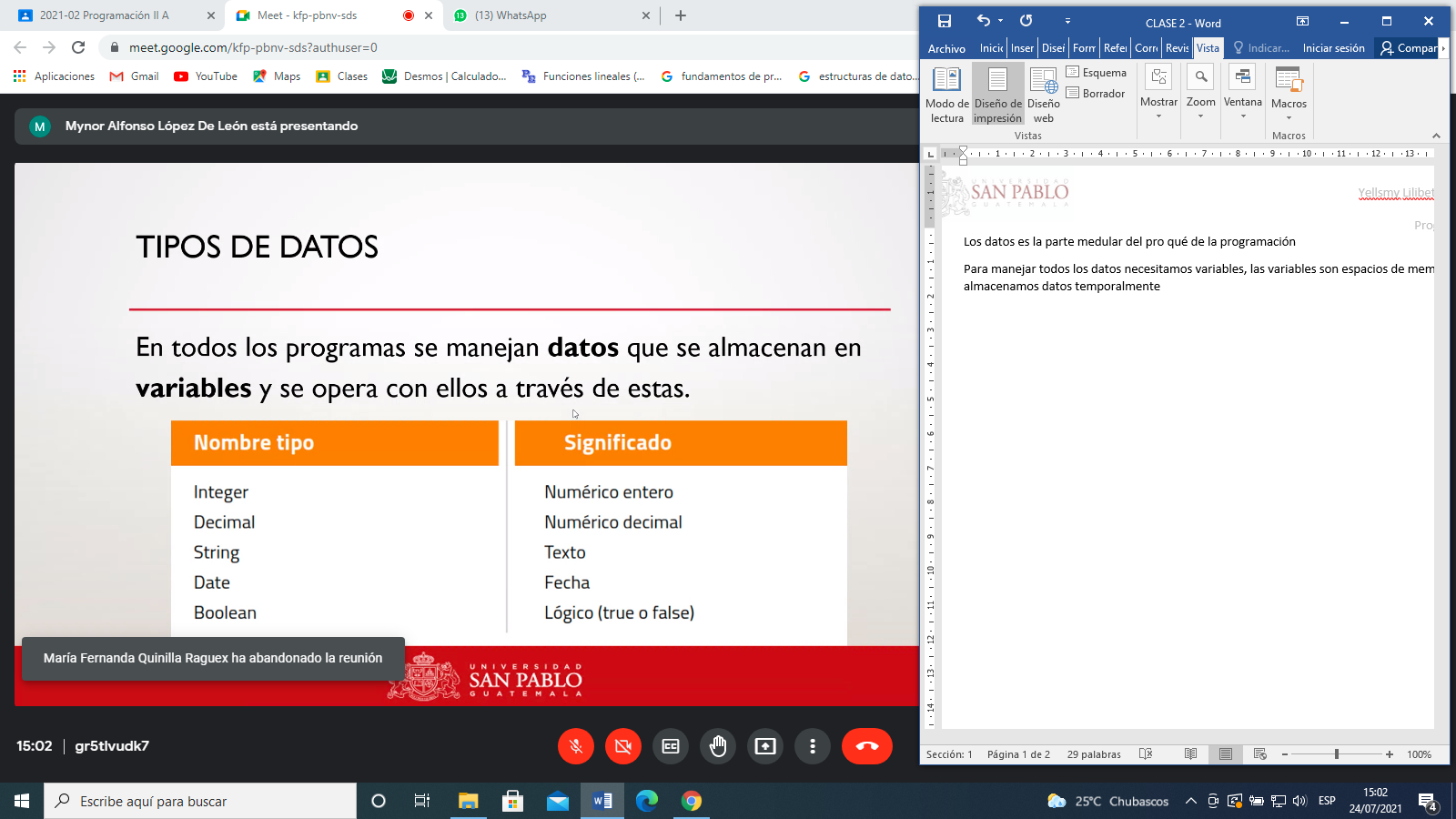
Los datos es la parte medular del pro qué de la programación

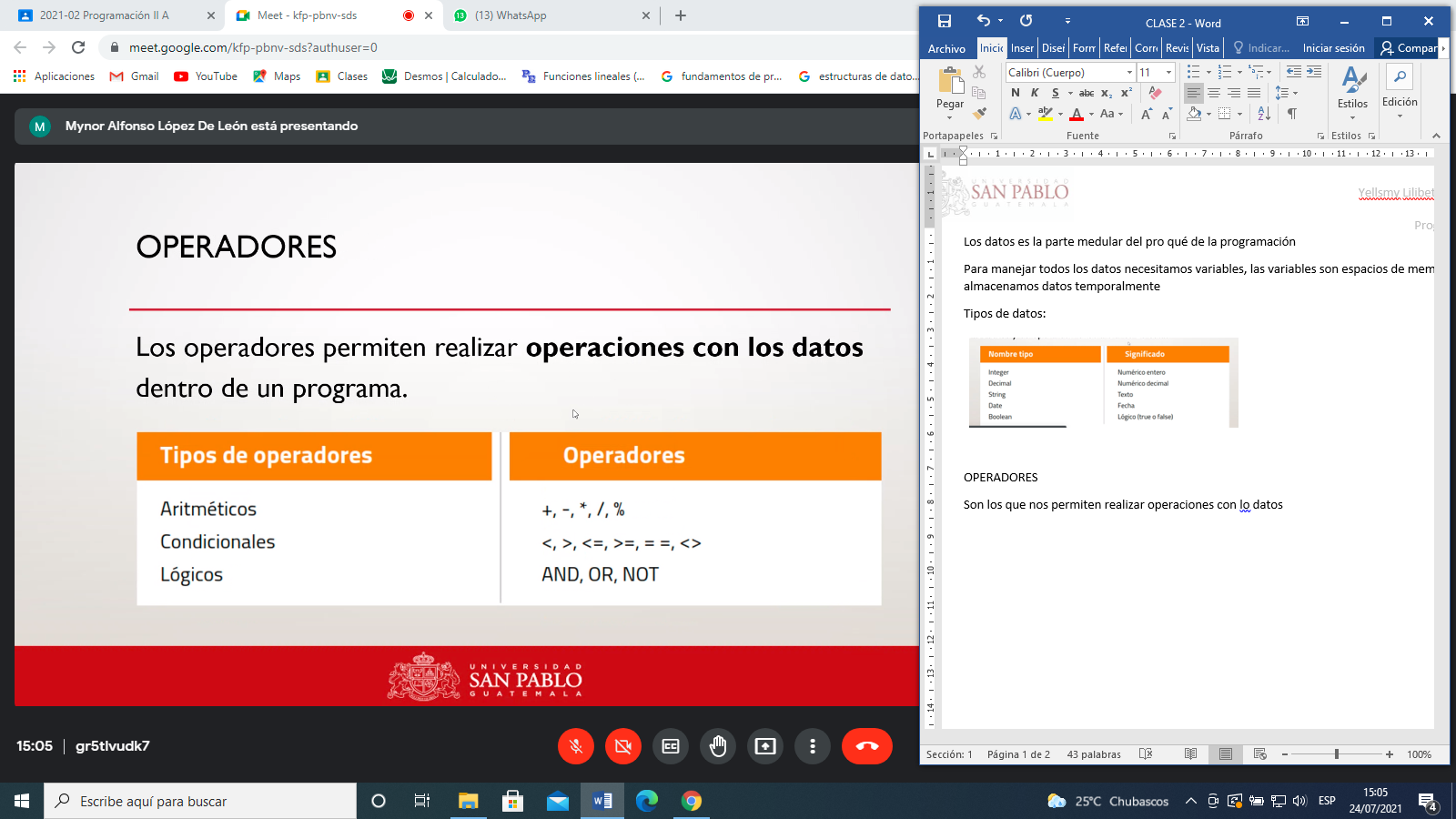
Para manejar todos los datos necesitamos variables, las variables son espacios de memoria donde almacenamos datos temporalmente

Tipos de datos:

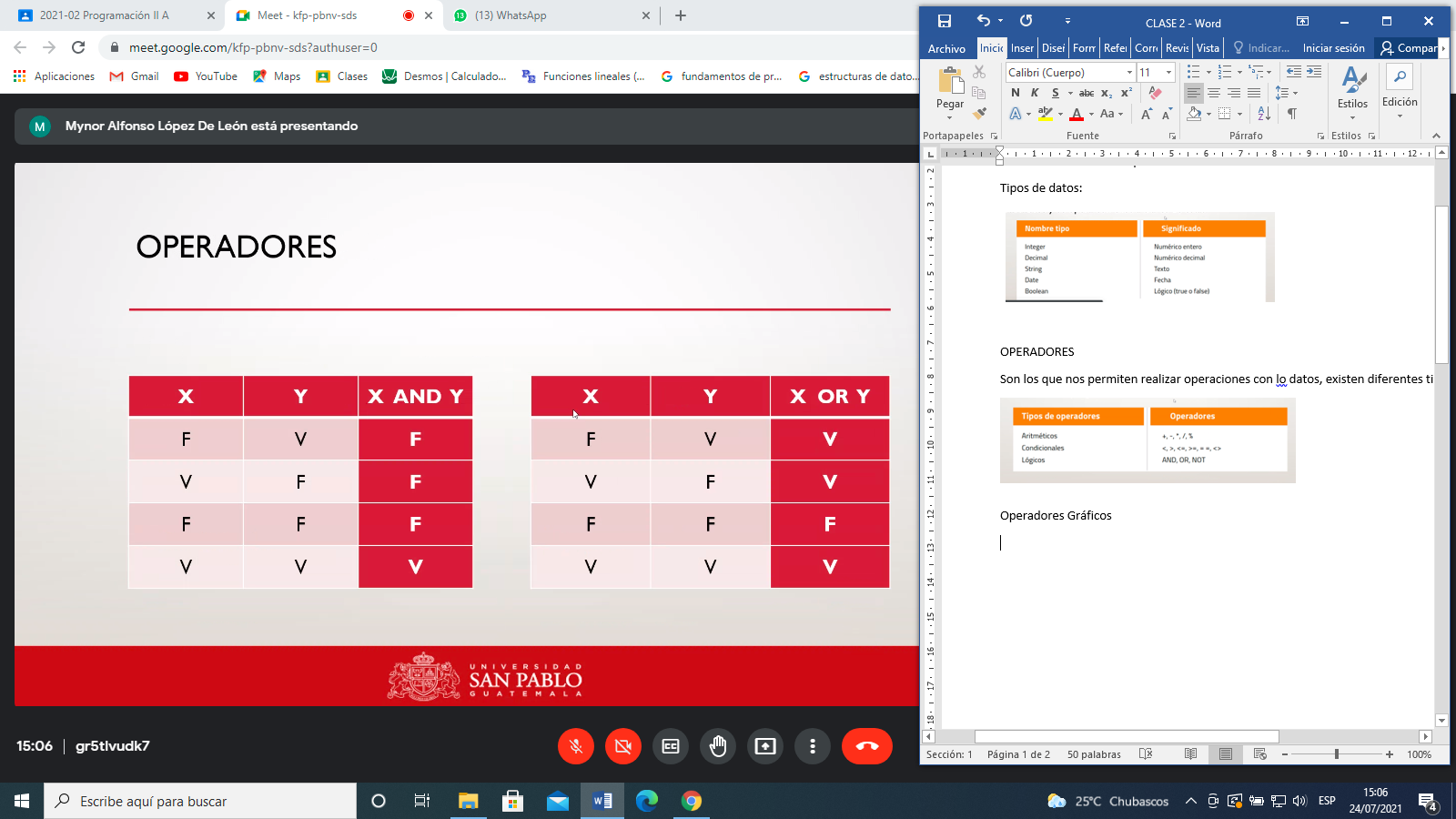


OPERADORES

Son los que nos permiten realizar operaciones con los datos, existen diferentes tipos de datos:



Operadores Gráficos



Es muy necesario usar los comentarios para registrar lo que estamos haciendo

TEOREMA DE LA ESTRUCTURA

Todo programa por complejo que sea puede ser diseñado con tres tipos de instrucciones básicas

Instrucciones secuenciales: Son aquellas que son continuas, una instrucción tras otra, por ejemplo, escribimos una instrucción por línea, y todas las líneas están numeradas.

Instrucciones alternativas: también llamadas condicionales, nos permite tomar las decisiones en un programa, de modo que ante una condición que se cumpla el programa puede tomar un camino o ejecutar sentencias, existen simples y multiples.

Simples: If, else

Multiples: Este tipo de instrucción que permite a partir de la evaluación de una condicional, realiza cierto tipo de instrucciones, por ejemplo, Switch

Switch (expresión)

Case valor 1:

Instrucciones

Case valor 2:

Instrucciones

:

Default

Instrucciones

End Switch

Instrucciones repetitivas: Permite ejecutar un bloque de sentencias varias veces en función del cumplimiento de una condición, por ejemplo:

Do While(condición)

Instrucción1

Instruccion2

:

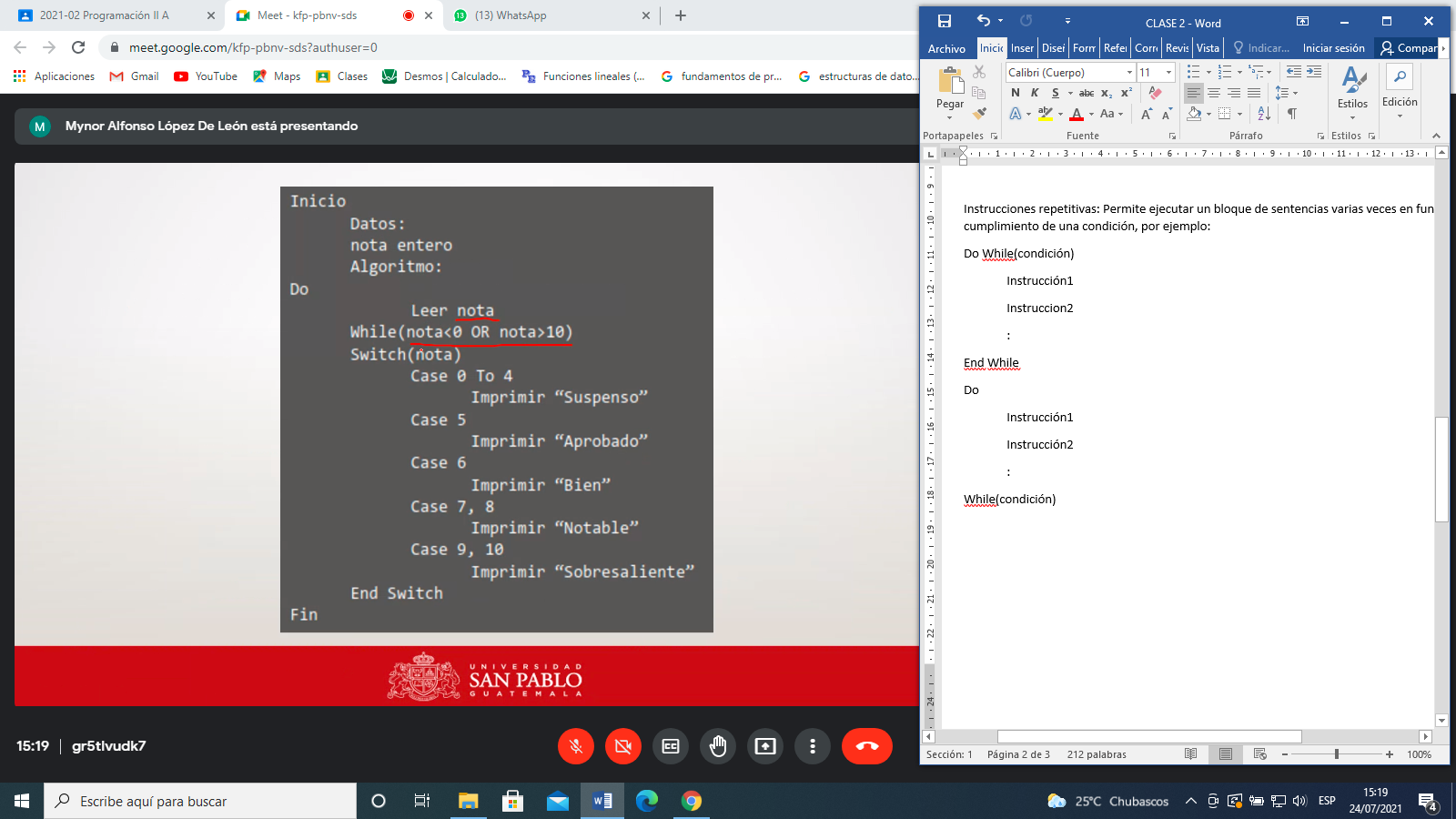
End While

Do

Instrucción1

Instrucción2

:

While(condición)

También existe el For como instrucción repetitiva, pero a diferencia de las anteriores, debemos saber cuántas veces se va ejecutar

PROGRAMACIÓN MODULAR

Representa una pequeña evolución de la programación estructurada. Se basa en la idea de divide y vencerás, la programación modular consiste en dividir un programa grande en bloques más pequeños (módulos), a fin de que el desarrollo pueda acometerse con mayor facilidad.

Ventajas:

Simplificación del desarrollo

Menor número de errores

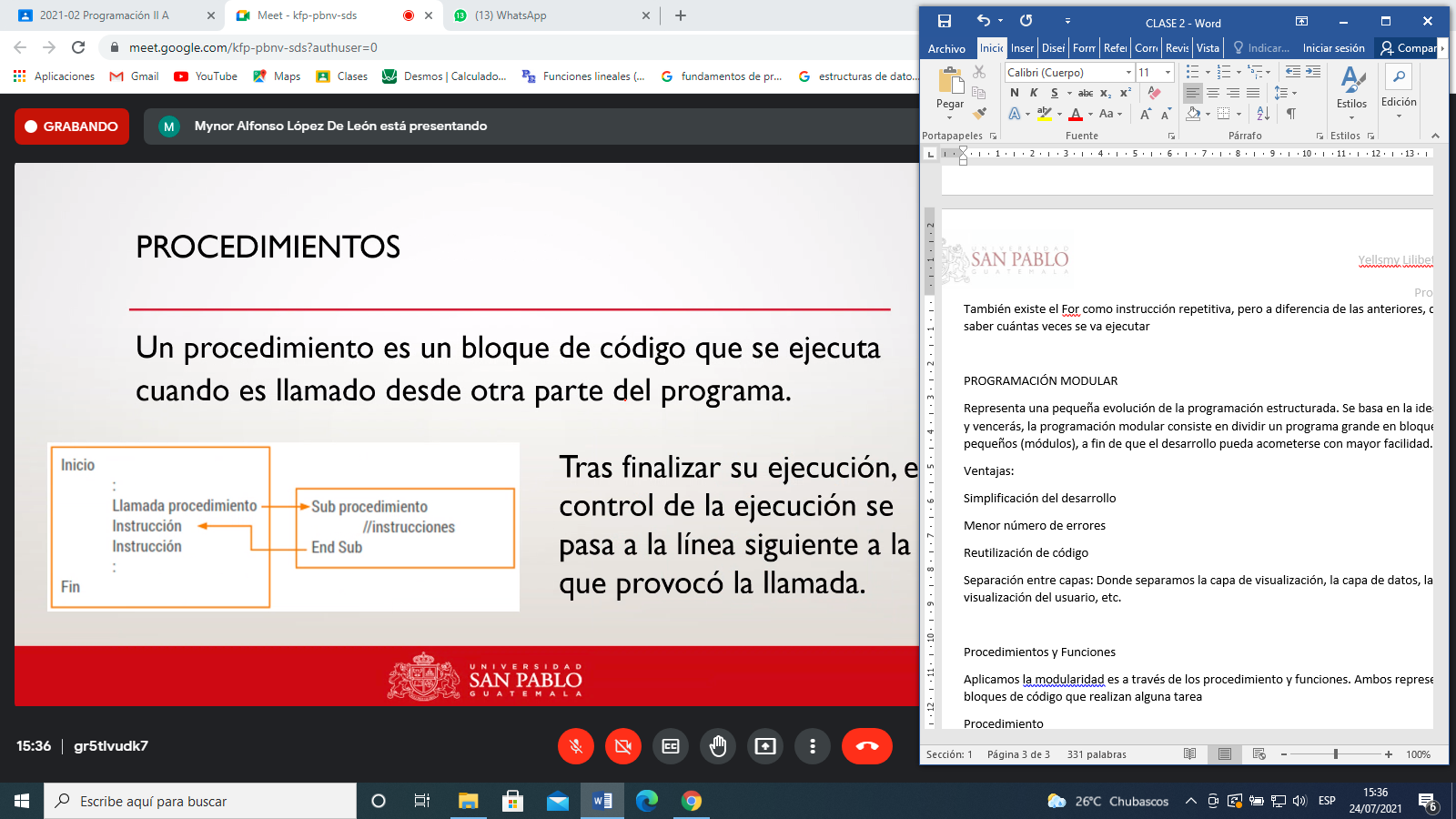
Reutilización de código

Separación entre capas: Donde separamos la capa de visualización, la capa de datos, la capa de visualización del usuario, etc.

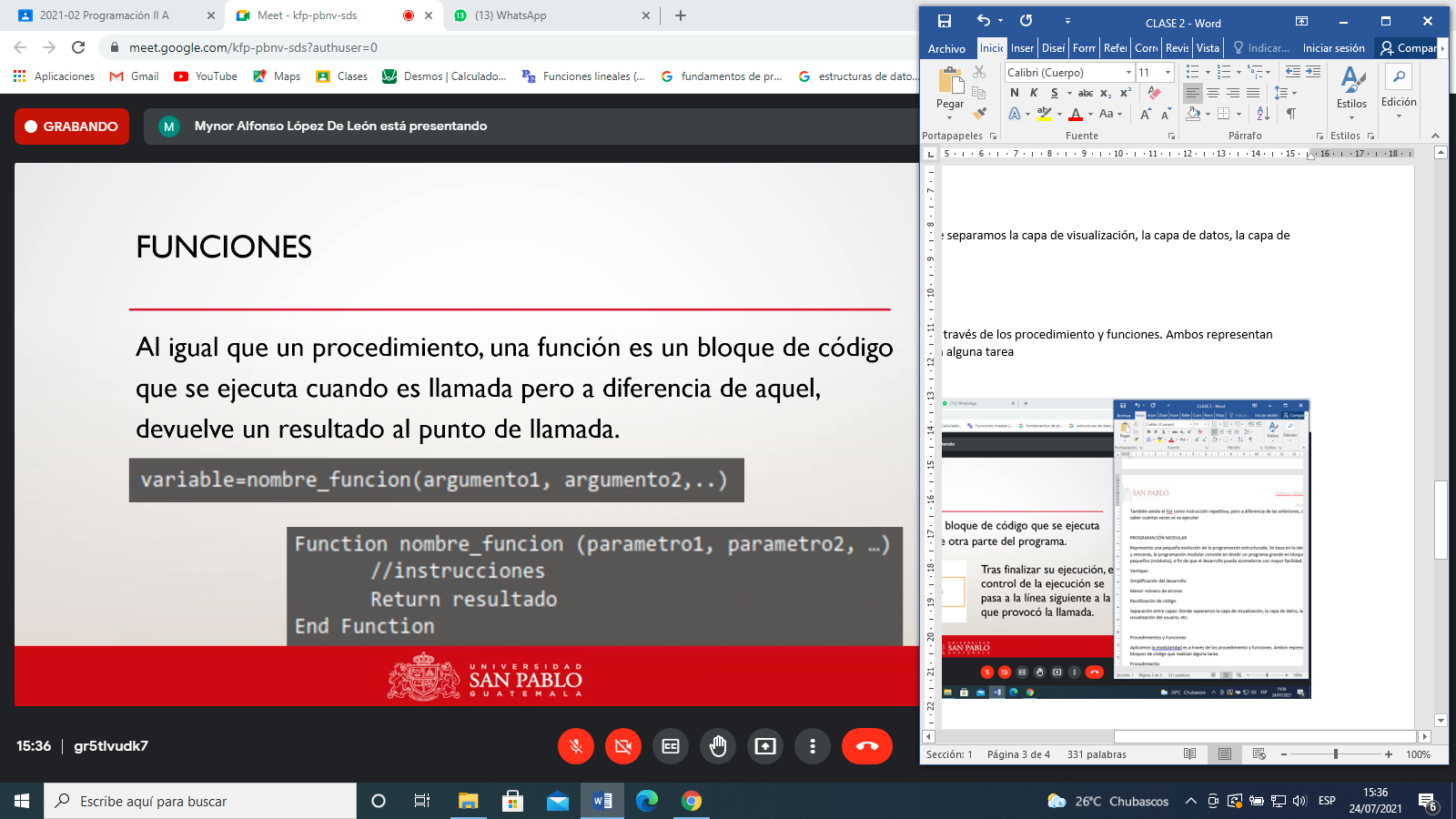
Procedimientos y Funciones

Aplicamos la modularidad es a través de los procedimiento y funciones. Ambos representan bloques de código que realizan alguna tarea

Procedimiento:



Función:



PROGRAMACIÓN ORIENTADA A OBJETOS

La mayoría de los lenguajes de programación actuales son orientados a objetos, esto permite que se puedan aplicar una serie de características propias de la POO, encaminadas a reutilizar código que sea más fácil de mantener y más fiable.

Un lenguaje orientado a objetos se basa en el uso de clases y objetos para implementar la funcionalidad de la aplicación.

Un objeto es como una caja negra que expone al exterior una serie de funciones con las que poder realizar operaciones, así como una serie de parámetros que permiten configurar características del objeto.

Se definen clases para separar cada una de las operaciones que se deban hacer.

OBJETOS:

A las funciones y procedimientos del objeto se le llama métodos, mientras que a las características se les conoce como propiedades.

Propiedades de una persona: Color de ojos, cabello, tipo de cabello, color de piel, tamaño, forma del rostro, cómo iba vestido, tamaño de ojos, todas las características que tiene una persona.

//asignación de un valor a una propiedad

Variable\_objeto.propiedad = valor

Métodos de una persona: hablar, correr, comer, caminar, saltar, todo lo que puede hacer la persona.

//recuperación del valor de una propiedad

Valor = variable\_objeto.propiedad

La variable objeto es cuando tenemos un espacio de memoria donde guardamos un objeto.

Para llamar a los métodos de un objeto utilizamos la sintaxis:

Variable\_objeto.metodo(argumento1, argumento2,…)

Resultado = variable:objeto.metodo(argumento1,…)

CLASES

El lugar donde se define el código de los métodos y propiedades es la clase. A través de éstos se establece el comportamiento de un objeto.

Cuando nos referimos a personas, estamos refiriéndonos a una clase, es decir, persona es una clase.

Definir una clse mediante pseudocódigo:

Estructura:

Class Nombre clase

//declaración de propiedades

//definición de métodos

End Class

Ejemplo:

Class Persona

//Altura, Peso

//Habla, canta, corre

End Class

Crear un objeto de una clase:

Estructura:

Objeto = New Nombreclase()

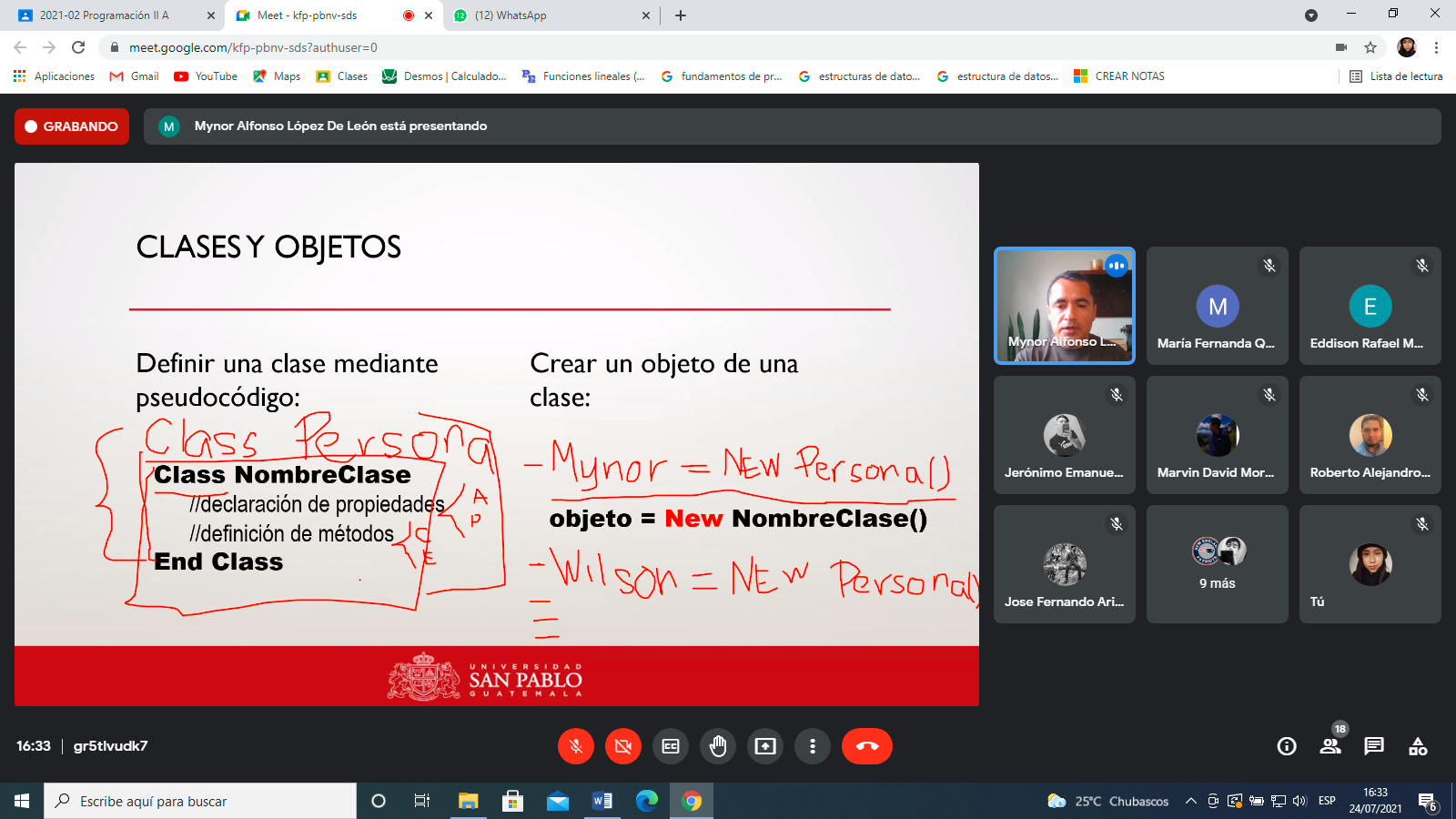
Ejemplo:

Mynor = New Persona()

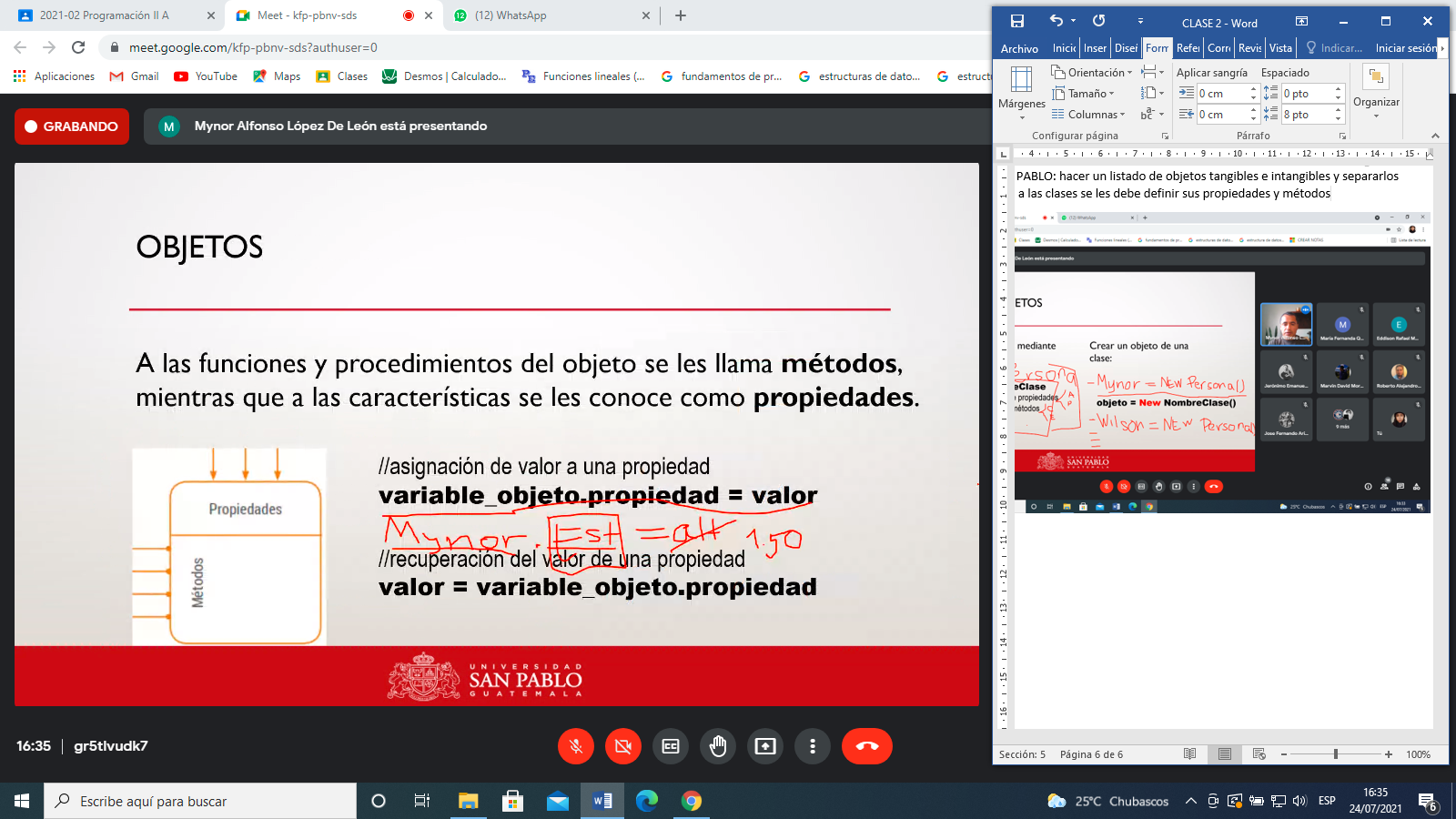
Wilson = New Persona()

TAREA:

UNIVERSIDAD DE SAN PABLO: hacer un listado de objetos tangibles e intangibles y separarlos entre clases y objetos,



a las clases se les debe definir sus propiedades y métodos, para los objetos se debe asignar valores a las propiedades



Definir al menos 15 clases y 30 objetos