

# **Estructura de Datos y Programación Orientada a Objetos**

**Proyecto: Implementación de clases del paquete**

**Controlador**

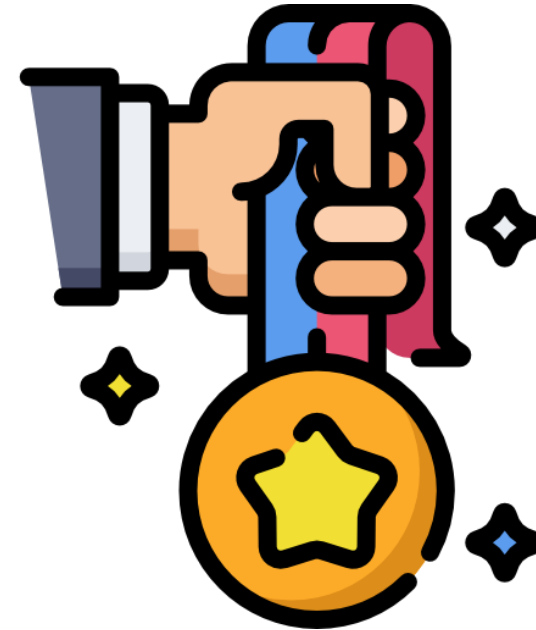
**Semana 10**



# Logro de aprendizaje



- Implementa las clases del proyecto usando el concepto de herencia, encapsulamiento y polimorfismo.



# Contenidos



PROYECTO:

IMPLEMENTACIÓN

DE CLASES DEL

PAQUETE

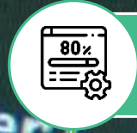
CONTROLADOR



Programación de los formularios utilizando las herramientas de Programación Orientada a Objetos



Abstracción



Encapsulamiento



Herencia



Polimorfismo

INICIO

# Observemos el siguiente vídeo



Tomar nota de las  
ideas importantes



[https://www.youtube.com/watch?v=oBO01Cx\\_YwQ&ab\\_channel=FredyGee](https://www.youtube.com/watch?v=oBO01Cx_YwQ&ab_channel=FredyGee)

k

INICIO

# Observemos el siguiente vídeo



Tomar nota de las  
ideas importantes



INICIO

# Herencia



## Herencia

La herencia crea nuevas clases partiendo de clases existentes, que tendrá todos los atributos y los métodos de su "superclase" o "clase padre".

Además se le podrán añadir otros atributos y métodos propios.

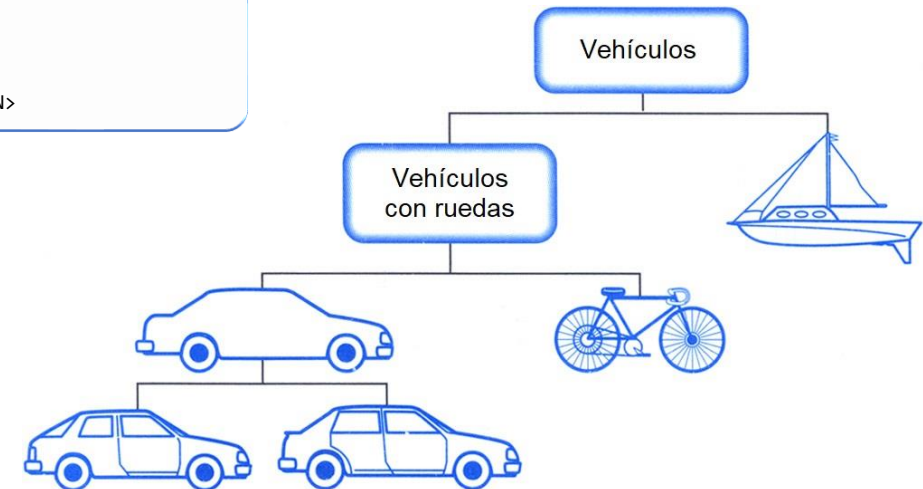
## Clases

**CLASE PADRE:** Clase de la que desciende o deriva una clase.

**SUBCLASE:** Clase descendiente de otra.

## Sintaxis

```
class ClaseDerivada(ClaseBase):  
    <declaración-1>  
    .  
    .  
    .  
    <declaración-N>
```



# Aplicación:



## ACTIVIDAD 1:



**Consigna:** Trabajo Individual: Desarrollo de ejercicios

- Cada alumno deberá realizar los ejercicios propuestos por el docente.



**Recursos:** Pc o Laptop, Python, VisualStudioCode

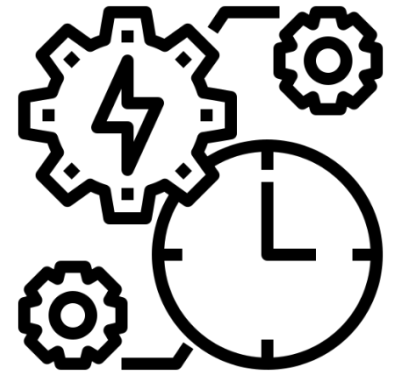


**Tiempo:** 40 minutos

# Ejercicios



- Implemente la clase Animal, que tenga como atributos nombre, numero\_patas y tamaño. En el método `__init__` cargar los atributos por teclado y en otro método mostrar sus datos. Luego implemente la clase Perro y la clase Gato que hereden de la clase Animal. Finalmente cree un objeto de las clases hijas uno para cada clase.
- Implemente la clase Animal, que tenga como atributos nombre, numero\_patas y tamaño. En el método `__init__` cargar los atributos por teclado y en otro método mostrar sus datos. Luego implemente la clase Canino que tiene como atributo raza y la clase Perro que tiene como atributo numero\_vacunas, la clase Canino hereda de la clase Animal y la clase Perro hereda de la clase Canino. Finalmente cree dos objetos de la clase Perro.

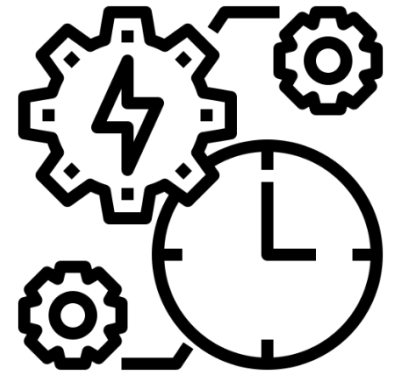




# Ejercicios



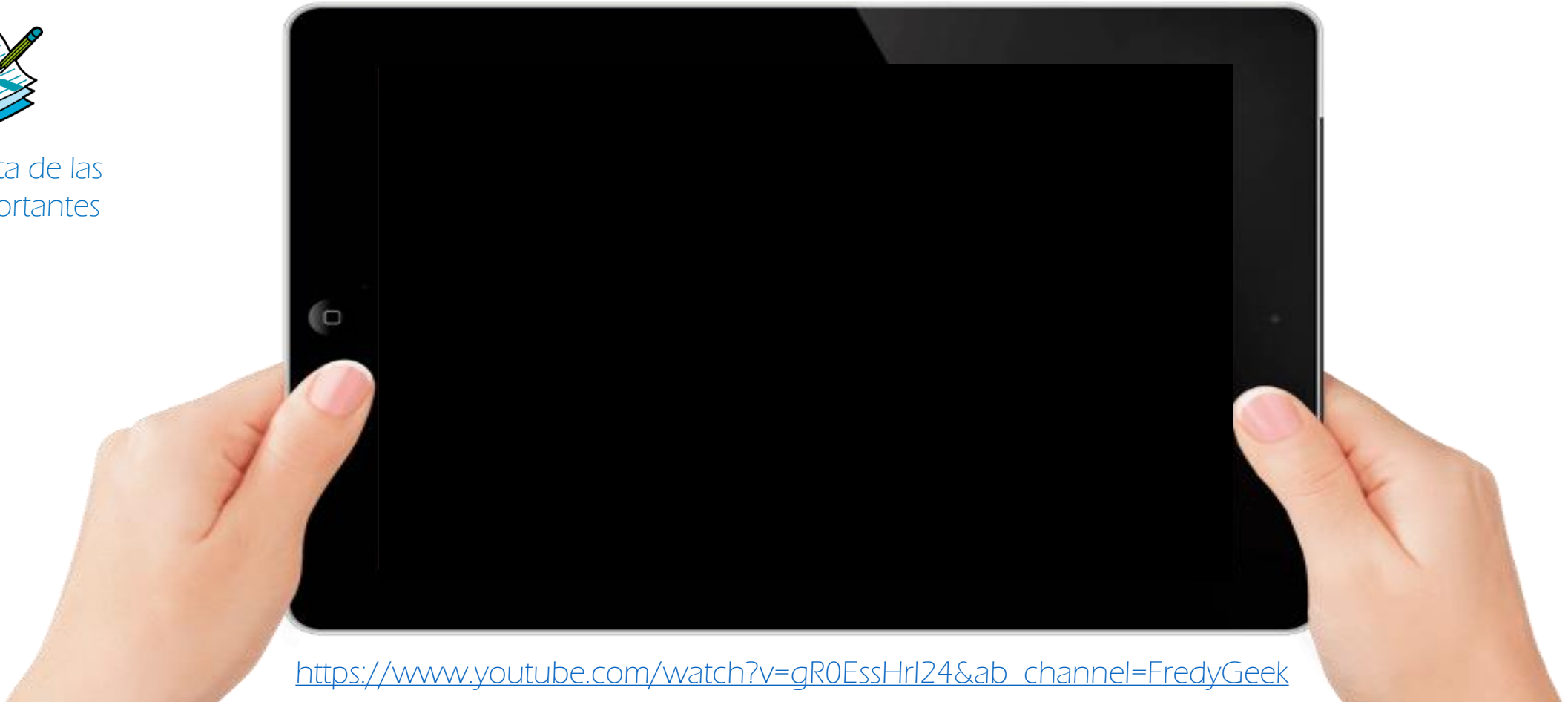
- Implementar una clase Persona que tenga dos atributos: nombre y edad. Definir dos métodos para la carga por teclado y su impresión. Implementar una segunda clase llamada Empleado que herede de la clase Persona y agregue un atributo sueldo y muestre si debe pagar impuestos (sueldo superior a 2500). Crear un objeto de cada clase.
- Implemente una clase Cuenta y dos subclases CuentaAhorro y CuentaPlazoFijo. Definir los atributos titular y monto de la clase Cuenta.
  - La clase CuentaAhorro tienen un nombre de titular y un monto. La clase CuentaPlazoFijo añade un plazo de imposición en días y una tasa de interés. La clase CuentaAhorro no genera interés, la clase CuentaPlazoFijo si genera intereses.
  - En el bloque principal del programa definir un objeto de la clase CuentaAhorro y otro de la clase CuentaPlazoFijo.



# Observemos el siguiente vídeo



Tomar nota de las  
ideas importantes



INICIO

# Encapsulamiento



## Encapsulamiento

- Los atributos de un objeto pueden ocultarse (superficialmente) para que no sean accedidos desde fuera de la definición de una clase. Para ello, es necesario nombrar los atributos con un prefijo de doble subrayado: `__atributo`

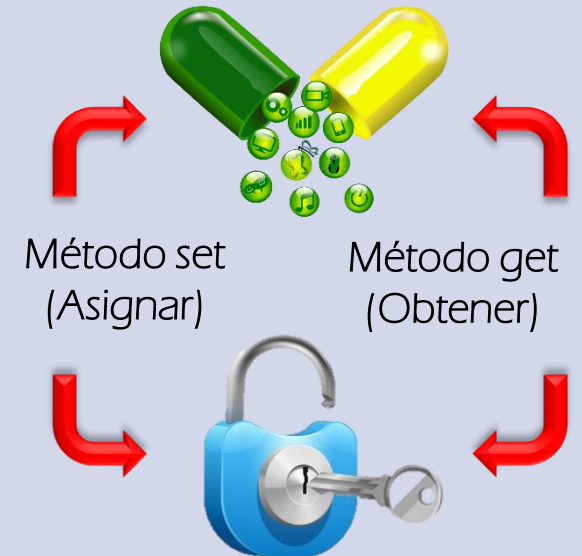


## Encapsulamiento

- Privado.



¿Qué necesito hacer para acceder o manipular su contenido?



# Aplicación:



## ACTIVIDAD 3:



**Consigna:** Trabajo Individual: Desarrollo de ejercicios

- Cada alumno deberá realizar los ejercicios propuestos por el docente.



**Recursos:** Pc o Laptop, Python, VisualStudioCode

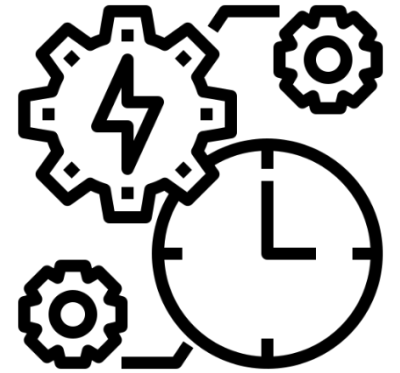


**Tiempo:** 40 minutos

# Ejercicios



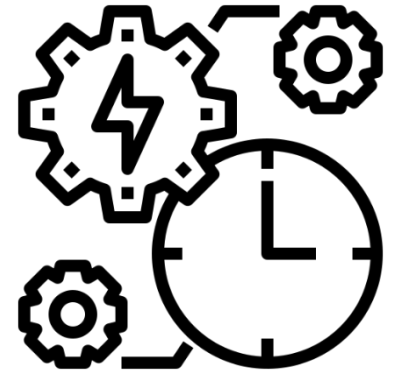
- Implementar una clase que represente una persona, que tenga como atributos código, nombre y edad. En el método `__init__` cargar los atributos por teclado y luego ocultar cada uno de los atributos. Cree un objeto de la clase persona y usando `print` muestre el código, nombre y edad del objeto creado.
- Implementar una clase que represente una persona, que tenga como atributos código, nombre y edad. En el método `__init__` cargar los atributos por teclado y luego ocultar cada uno de los atributos. Crear los métodos `get` y `set` para acceder y modificar los atributos fuera de la clase. Cree un objeto de la clase persona y usando `print` muestre el código, nombre y edad del objeto creado.



# Ejercicios



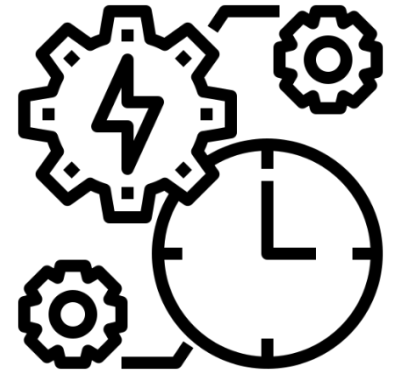
- Implementar una clase que represente una persona, que tenga como atributos código, nombre y edad. En el método `__init__` cargar los atributos por teclado, otro método para mostrar datos y un método para saber si la persona es mayor de edad. Luego ocultar cada uno de los atributos y también los métodos. Crear los métodos `get` y `set` para acceder y modificar a los métodos y atributos fuera de la clase. Cree un objeto de la clase persona.
- Implementar una clase que represente un rectángulo, que tenga como atributos su largo y su ancho. En el método `__init__` cargar los atributos por teclado, en otro método calcular su área y otro método para calcular su perímetro. Luego ocultar cada uno de los atributos y también los métodos. Crear los métodos `get` y `set` para acceder y modificar a los métodos y atributos fuera de la clase. Cree un objeto de la clase rectángulo.



# Ejercicios



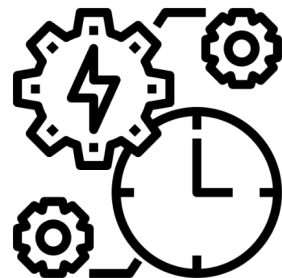
- Implementar una clase que represente un triángulo, que tenga como atributos sus tres lados. En el método `__init__` cargar los atributos por teclado, en otro método imprimir el valor del lado mayor y otro método que muestre si es equilátero o no. Luego ocultar cada uno de los atributos y también los métodos. Crear el método `get` para acceder a los métodos y atributos fuera de la clase. Cree un objeto de la clase triángulo.



# Ejercicios



- Cree el `PaqueteAlumno2`, dentro creamos el archivo `claseAlumno.py`, declare la clase `Alumno` que permita registrar un nombre y dos notas, los atributos deben estar encapsulados, declare sus métodos de acceso `get` y `set`, debe crear un método `promedio` que retorne la nota promedio. Cree el archivo `formularioAlumno.py`, declare la clase `FormularioAlumno`, cree tres objetos de la clase `Alumno`, y luego de ingresar sus datos visualice la data ingresada, además del promedio obtenido. Debe crear también el formulario `frmAlumno.ui`.



MainWindow

PROCESAR

LIMPIAR

SAIR

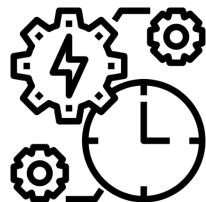
|          |             |
|----------|-------------|
| Nombre   | : Vanessa   |
| Nota 1   | : 20        |
| Nota 2   | : 15        |
| Promedio | : 17.5      |
| Nombre   | : Valentina |
| Nota 1   | : 12        |
| Nota 2   | : 15        |
| Promedio | : 13.5      |
| Nombre   | : Cecilia   |
| Nota 1   | : 12        |
| Nota 2   | : 20        |
| Promedio | : 16.0      |



# Ejercicios



- Cree el **PaqueteEmpleado**, dentro creamos el archivo **claseEmpleado.py**, declare la clase **Empleado**, declare los siguientes atributos como privados: `codigo (int)`, `nombre (String)`, `horas (int)`, `tarifa (float)` e implemente los siguientes métodos:
  - Un constructor que inicialice a los atributos
  - Métodos de acceso: `set/get` para cada atributo
  - Un método que retorne el sueldo bruto ( $\text{horas} * \text{tarifa}$ )
  - Un método que retorne el descuento (11% del sueldo bruto)
  - Un método que retorne el sueldo neto ( $\text{sueldo bruto} - \text{descuento}$ )
- Cree el archivo **formularioEmpleado.py**, declare la clase **FormularioEmpleado**, cree un objeto de la clase **Empleado** con datos fijos, cree un método `mostrarDatos` que imprima los atributos del empleado, el sueldo bruto, el descuento y el sueldo neto. Debe crear también el formulario `frmEmpleado.ui`.



MainWindow

PROCESAR

LIMPIAR

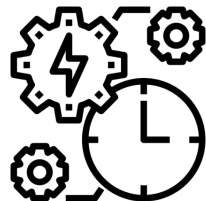
SALIR

|              |            |
|--------------|------------|
| Código       | : 123      |
| Nombre       | : Kenny    |
| Horas        | : 5        |
| Tarifa       | : S/. 20   |
| Sueldo Bruto | : S/. 100  |
| Descuento    | : S/. 11.0 |
| Sueldo Neto  | : S/. 89.0 |

# Ejercicios



- Cree el `PaqueteRegistroAlumnos`, dentro creamos el archivo `claseRegistroAlumnos.py`, declare la clase `RegistroAlumnos`, declare los siguientes atributos como privados: `nombre` (String), `apellidoPaterno` (String), `apellidoMaterno` (String), `codigo` (String) e implemente los siguientes métodos:
  - Un constructor que inicialice a los atributos
  - Métodos de acceso: `set/get` para cada atributo
- Cree el archivo `formularioRegistroAlumnos.py`, declare la clase `FormularioRegistroAlumnos`, cree un objeto de la clase `RegistroAlumnos`. Debe crear también el formulario `frmRegistroAlumnos.ui`.



The screenshot shows a window titled "MainWindow" with a form titled "Registro de Alumnos". The form has four input fields: "Nombre:", "Apellido Paterno:", "Apellido Materno:", and "Código:". To the right of these fields are four buttons: "REGISTRAR", "MOSTRAR", "LIMPIAR", and "SALIR". Below the input fields, there is a section titled "Relación de alumnos:" which contains a text area displaying the following information:

|                  |           |
|------------------|-----------|
| Nombre           | : KENNY   |
| Apellido Paterno | : MALLQUI |
| Apellido Materno | : BARRERA |
| Código           | : D17036  |

# Aplicación:



## ACTIVIDAD 2:



**Consigna:** Trabajo Individual: Elaboración de formularios para el Proyecto

- Cada alumno deberá realizar los ejercicios propuestos por el docente.

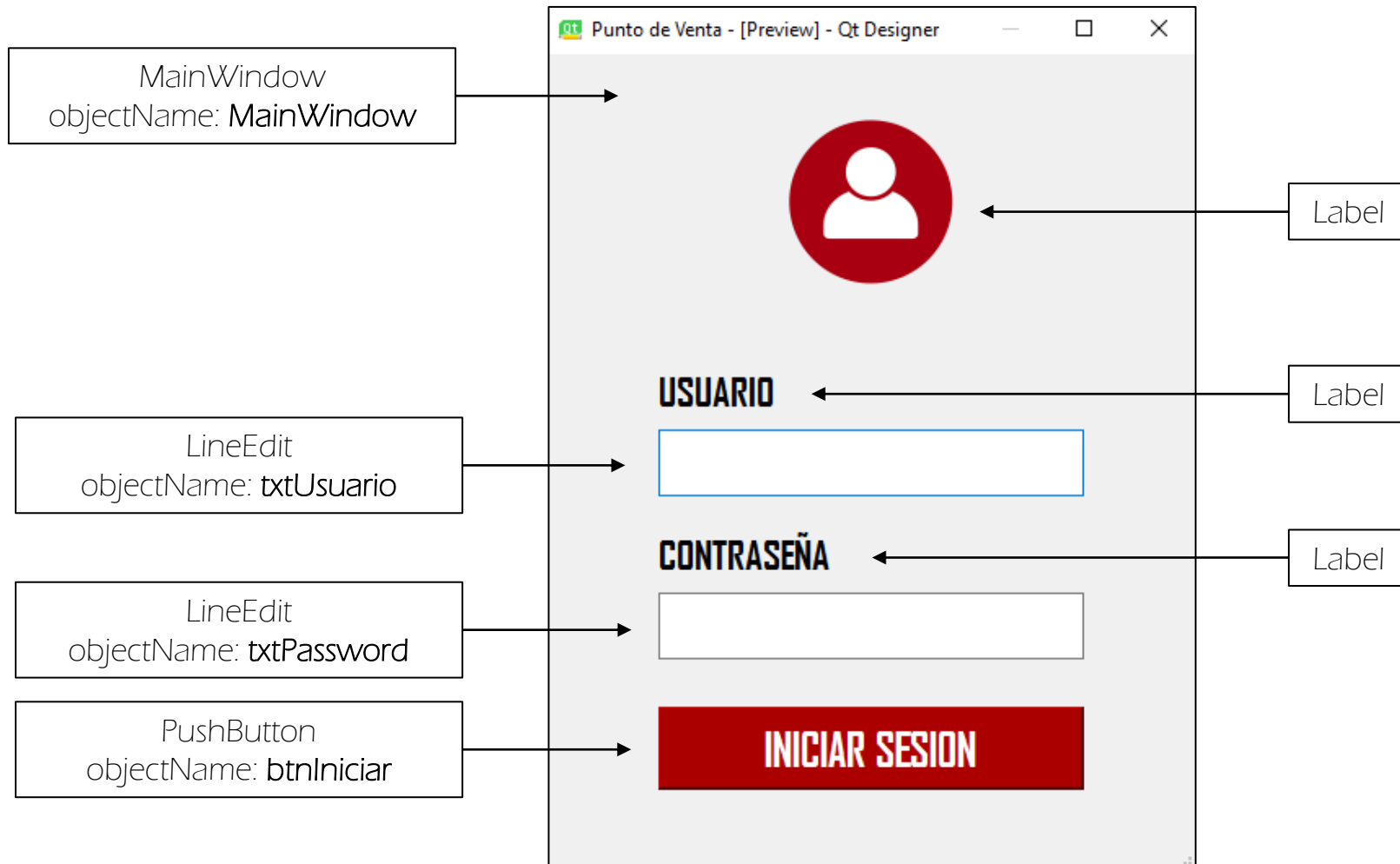
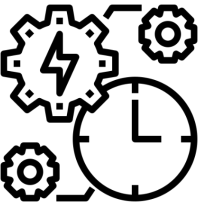


**Recursos:** Pc o Laptop, Python

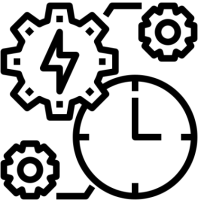
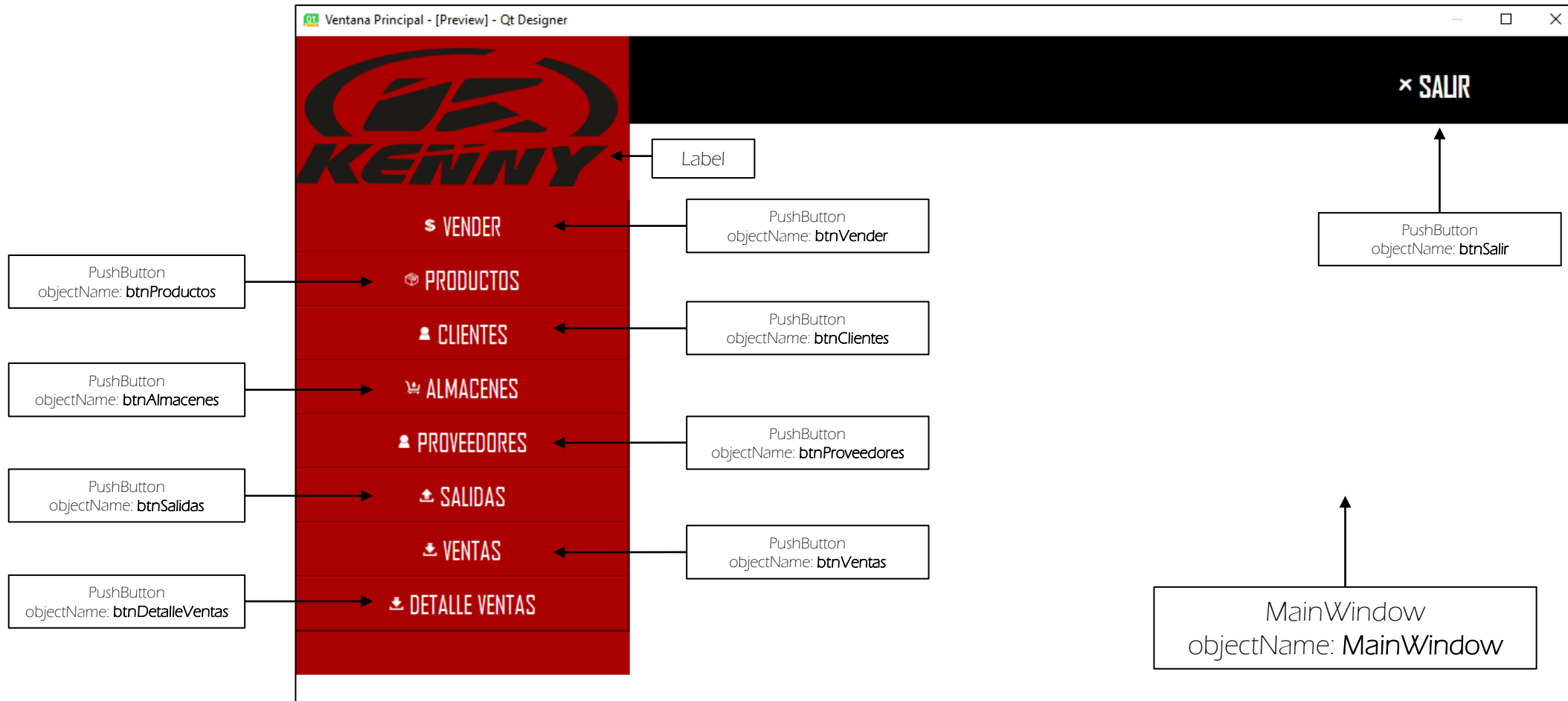


**Tiempo:** 50 minutos

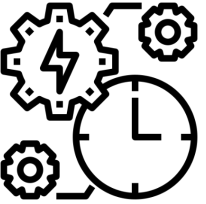
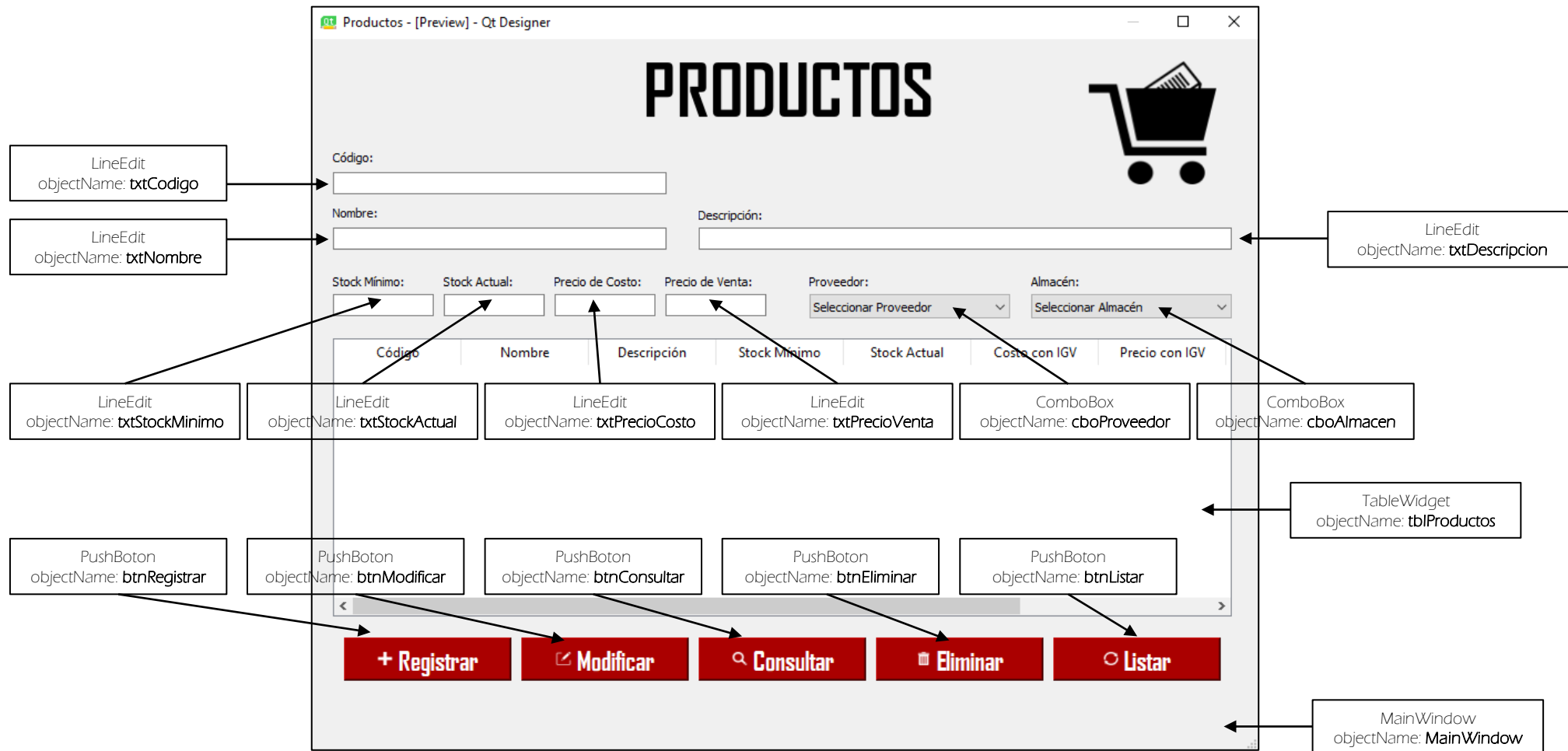
# Ejercicios



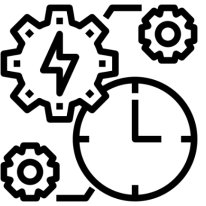
# Ejercicios



# Ejercicios



# Ejercicios



APLICACIÓN

# Ejercicios



Productos - [Preview] - Qt Designer

## PUNTO DE VENTA KENNY

Venta de Productos Tecnológicos

Nº

Código Cliente:  Nombres y Apellidos:

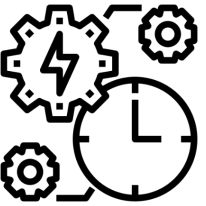
Código Producto:  Descripción:  Stock:  Precio:  Cantidad:

| Nro | Código Producto | Descripción | Precio | Cantidad | Total |
|-----|-----------------|-------------|--------|----------|-------|
|     |                 |             |        |          |       |

Subtotal

IGV (18%)

Total a Pagar



APLICACIÓN



# Evaluación Continua 3



Escuela de Tecnología

Desarrollo de Sistemas de Información

Curso: EDD & POO

Profesor: Kenny Vladimir Mallqui Barrera

Alumno:

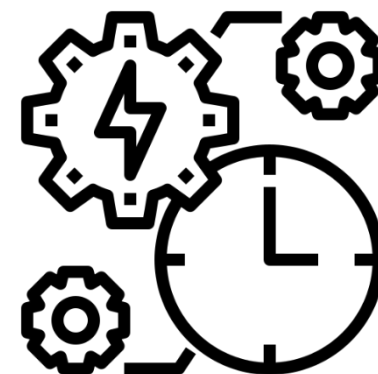
Nota:  
(números y letras)

Ciclo: II

Fecha: 15 / 11 / 22

Semestre: 2022 - IIIE

Duración: 90 min.

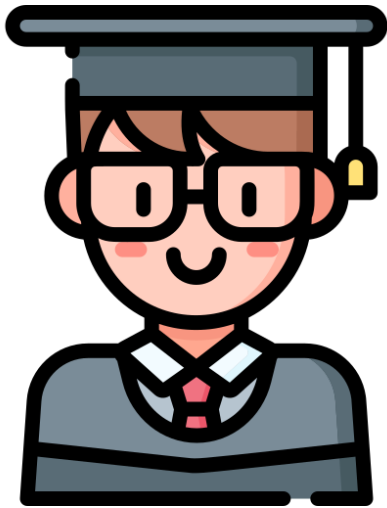


## Evaluación Continua 3

### Recomendaciones:

- Lee bien cada ítem y responde aquello que se te solicita.
- No olvides poner tu nombre antes de entregar esta evaluación.
- Recuerda revisar la ortografía.





1

¿Qué aprendimos hoy?

2

¿Por qué el tema tratado es importante en mi formación como programador?



**Muchas Gracias**  
**por su atención**  
¿Alguna pregunta?  
**¿No?**  
**Excelente**