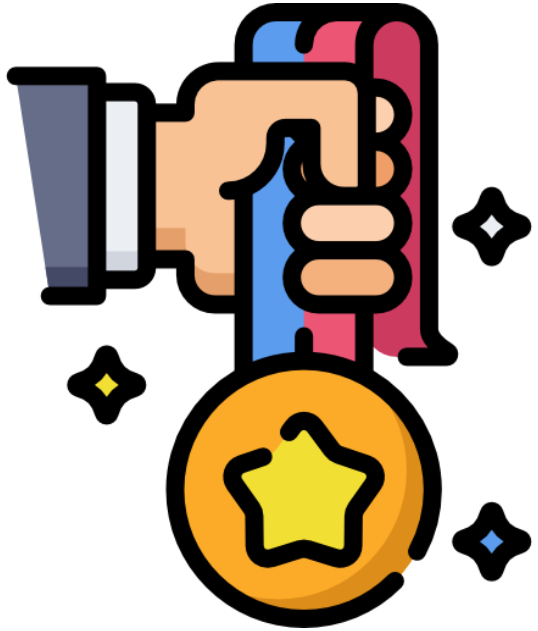
Estructura de Datos y Programación Orientada a Objetos

Proyecto: Implementación de clases del paquete Controlador

Semana 8

Logro de aprendizaje

- *Implementa la estructura de un proyecto utilizando el concepto de clase.*



Contenidos

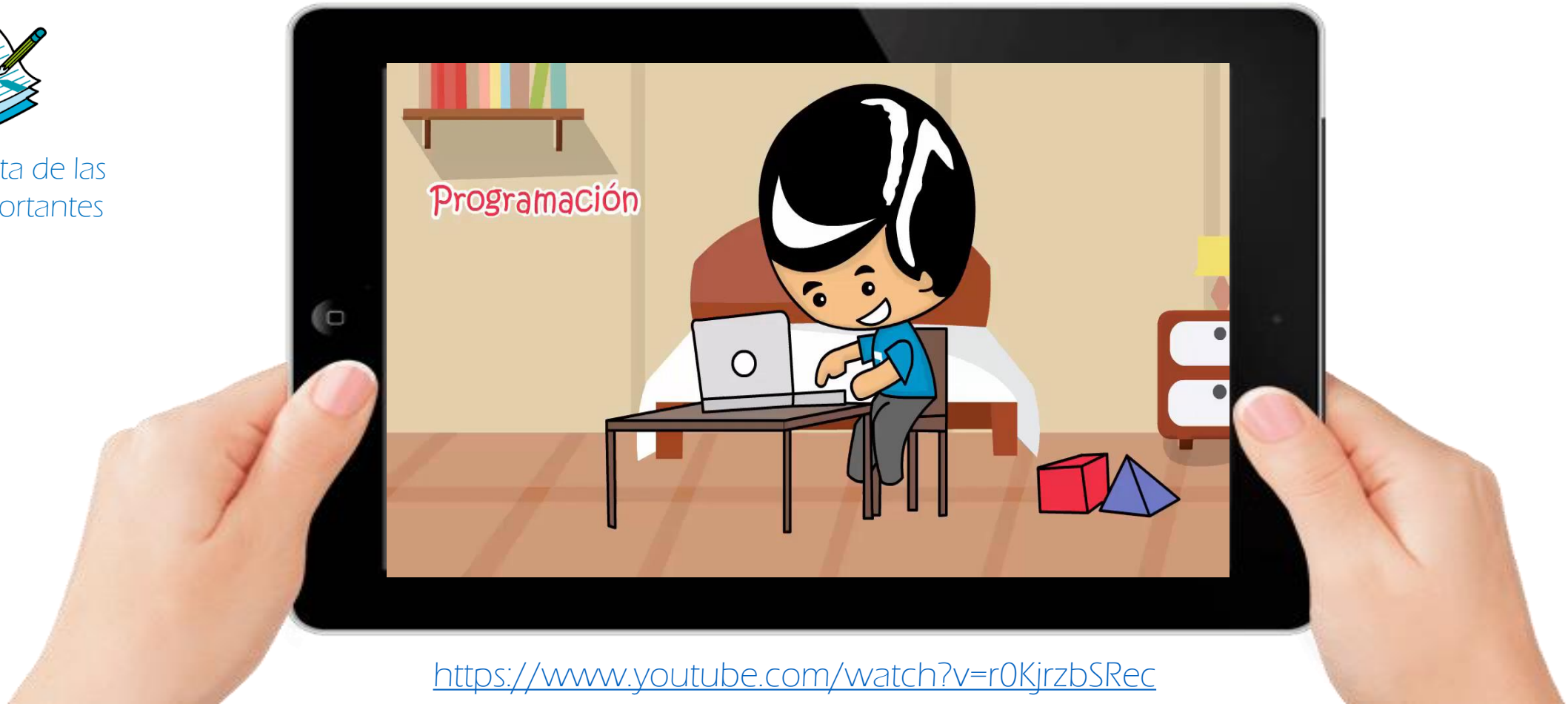
PROYECTO:
IMPLEMENTACIÓN
DE CLASES
DEL PAQUETE
CONTROLADOR

Diagrama de clase del Proyecto

Implementación de clases

Observemos el siguiente vídeo

Tomar nota de las ideas importantes



https://www.youtube.com/watch?v=r0KjrzbSRec

INICIO

La POO

¿QUÉ ES LA PROGRAMACIÓN ORIENTADA A OBJETOS?

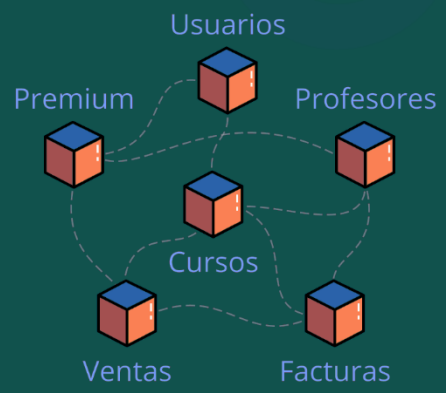
Es un paradigma de programación que organiza las funciones en entidades llamadas objetos.

Los objetos se crean a partir de una plantilla llamada clase. Cada objeto es una instancia de su clase.

CLASE → INSTANCIACIÓN → OBJETO

Los objetos tienen **datos (atributos)** y **funcionalidades (métodos).**

En una aplicación los objetos están separados **pero se comunican entre ellos.**



ATRIBUTOS

```
Nombres
Apellidos
Correo
Contraseña
Premium
```

MÉTODOS

```
Editar perfil
Iniciar sesión
Cerrar sesión
Cambiar contraseña
Pasar a premium
```

Puedes programar con este paradigma **en la mayoría de lenguajes.**

Conceptos básicos de la POO

¿Qué es un objeto?

¿Objeto?

Algo / Cosa

Que se quiere representar pero no es suficiente con una variable de los tipos más básicos.

Cualidades de un Objeto

Descripción del objeto

¿Cómo es?

¿Qué tiene?

A las cualidades se les conoce como Atributos

- Color
- Modelo
- Ventanas
- Puertas
- Llantas
- ...

Capacidad de hacer algo (Comportamiento)

¿Qué hace?

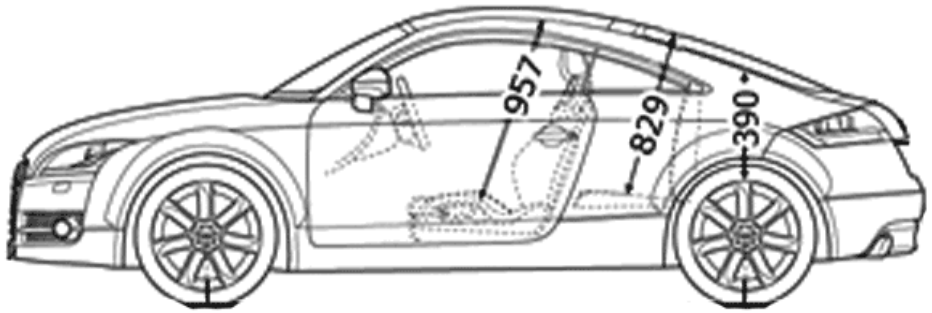
- Encender
- Apagar
- Acelerar
- Frenar
- Abrir puertas
- Cerrar puertas

A los comportamientos se les conoce como Métodos o Funciones

Clases

¿Qué es una clase?

Molde para obtener objetos con la misma estructura



Para obtener un objeto a partir de una clase, se la debe INSTANCIAR.

Objetos de la misma clase: tienen los mismos métodos y atributos, pero los atributos pueden tener distintos valores.

Elementos y características de la POO

Elementos

- Materiales necesarios para diseñar y programar un sistema.

Características

- Herramientas disponibles para construir el sistema a partir de los materiales.

- Visto con más claridad

Elementos:

OBJETO: Auto
ATRIBUTOS: Color, Modelo, etc.
MÉTODOS: Encender, Apagar, etc.

Características:

Abstracción
Encapsulamiento
Herencia
Polimorfismo

Creando nuestro primer objeto

Sintaxis:

```
class <nombre clase>:
    <acciones>
```

Ejemplo

```
class Auto:
   pass

coche1 = Auto()

coche1.color = "Rojo"
coche1.modelo = "Yaris"
coche1.puertas = 4
coche1.llantas = 4
coche1.velocidades = 5

print("Color: ", coche1.color)
print("Modelo: ", coche1.modelo)
print("Puertas: ", coche1.puertas)
print("Llantas: ", coche1.llantas)
print("Velocidades: ", coche1.velocidades)
```

Creando atributos y métodos

Sintaxis:

```
def <nombre metodo>(self):
    <acciones>
```

Ejemplo

```
class Auto:
    color = ""
    modelo = ""
    puertas = 0
    llantas = 0
    velocidades = 0

    def mostrarDatos(self):
        print("Color: ", self.color)
        print("Modelo: ", self.modelo)
        print("Puertas: ", self.puertas)
        print("Llantas: ", self.llantas)
        print("Velocidades: ", self.velocidades)

coche1 = Auto()
coche1.color = "Azul"
coche1.modelo = "Yaris"
coche1.puertas = 4
coche1.llantas = 4
coche1.velocidades = 5
coche1.mostrarDatos()
```

Creando atributos y métodos

Sintaxis:

```
def <nombre metodo>(self):
    <acciones>
```

Ejemplo

```
class Auto:

    def inicializar(self, color, modelo, puertas, llantas, velocidades):
        self.color = color
        self.modelo = modelo
        self.puertas = puertas
        self.llantas = llantas
        self.velocidades = velocidades

    def mostrarDatos(self):
        print("Color: ", self.color)
        print("Modelo: ", self.modelo)
        print("Puertas: ", self.puertas)
        print("Llantas: ", self.llantas)
        print("Velocidades: ", self.velocidades)

coche1 = Auto()
coche1.inicializar("Rojo", "Yaris", 4, 4, 5)
coche1.mostrarDatos()
```

Creando atributos y métodos

Sintaxis:

```
def <nombre metodo>(self):
    <acciones>
```

Ejemplo

```
class Auto:

    def inicializar(self):
        self.color = input("Ingrese el color del auto: ")
        self. modelo = input("Ingrese el modelo del auto: ")
        self.puertas = int(input("Ingrese el número de puertas del auto: "))
        self. llantas = int(input("Ingrese el número de llantas del auto: "))
        self.velocidades = int(input("Ingrese las velocidades del auto: "))

    def mostrarDatos(self):
        print("Color: ", self.color)
        print("Modelo: ", self.modelo)
        print("Puertas: ", self.puertas)
        print("Llantas: ", self.llantas)
        print("Velocidades: ", self.velocidades)

coche1 = Auto()
coche1.inicializar()
coche1.mostrarDatos()
```

MÉTODO __init__ DE LA CLASE:

Método `__init__`

- El método **`__init__`** es un método especial de una clase en Python.
- El objetivo fundamental del método **`__init__`** es inicializar los atributos del objeto que creamos.
- Básicamente el método **`__init__`** remplaza al método inicializar que habíamos hecho en la clase anterior.

Ventajas

- El método **`__init__`** es el primer método que se ejecuta cuando se crea un objeto.
- El método **`__init__`** se llama automáticamente. Es decir, es imposible de olvidarse de llamarlo ya que se llamará automáticamente.

Sintaxis

```
def __init__([parámetros]):

    [algoritmo]
```

Aplicación:

ACTIVIDAD 1:

Consigna: Trabajo Individual: Desarrollo de ejercicios

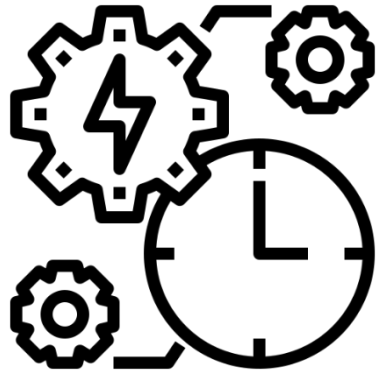
- Cada alumno deberá realizar los ejercicios propuestos por el docente.

Recursos: Pc o Laptop, Python

Tiempo: 50 minutos

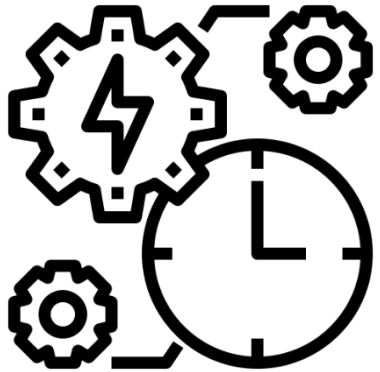
Ejercicios

- Implemente una clase llamada Celular que tenga como mínimo 5 atributos. Definir los siguientes métodos: para inicializar y para mostrar datos.
- Implemente una clase llamada Casa que tenga como mínimo 5 atributos. Definir los siguientes métodos: para inicializar y para mostrar datos.
- Implemente una clase llamada Operacion que tenga como atributos num1 y num2. Definir los siguientes métodos: para inicializar, para sumar, para restar, para multiplicar, para dividir y para mostrar resultado.
- Implementar una clase llamada Alumno que tenga como atributos su nombre y su nota. Definir los métodos para inicializar sus atributos, imprimirlos y mostrar un mensaje si está APROBADO (nota mayor o igual a 13). Definir dos objetos de la clase Alumno.



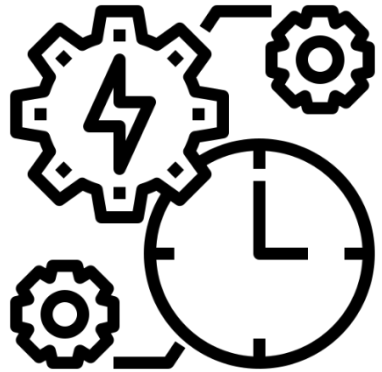
Ejercicios

- Confeccionar una clase que permita carga el nombre y la edad de una persona. Mostrar los datos cargados. Imprimir un mensaje si es mayor de edad (edad >= 18).
- Desarrollar un programa que cargue los lados de un triángulo e implemente los siguientes métodos: inicializar los atributos, imprimir el valor del lado mayor y otro método que muestre si es equilátero o no. El nombre de la clase llamarla Triangulo.
- Implementar una clase que represente un empleado, que tenga como atributos su nombre y su sueldo. En el método __init__ cargar los atributos por teclado y luego en otro método ver sus datos y por último un método que imprima un mensaje si debe pagar impuestos (si el sueldo supera a 2250).



Ejercicios

- 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.
- 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.



Aplicación:

ACTIVIDAD 2:

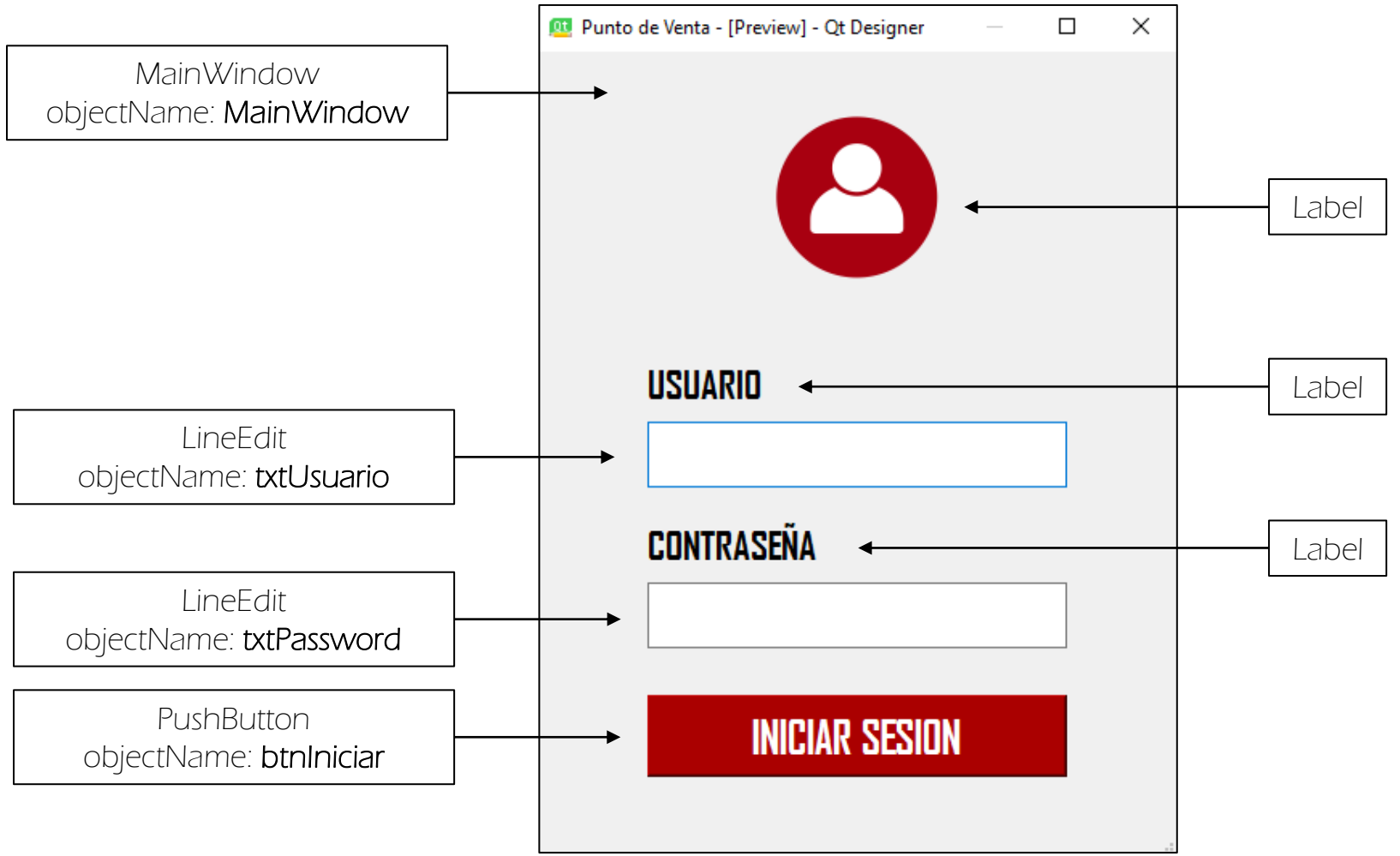
Consigna: Trabajo Individual: Elaboración de formularios para el Proyecto

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

Recursos: Pc o Laptop, Python

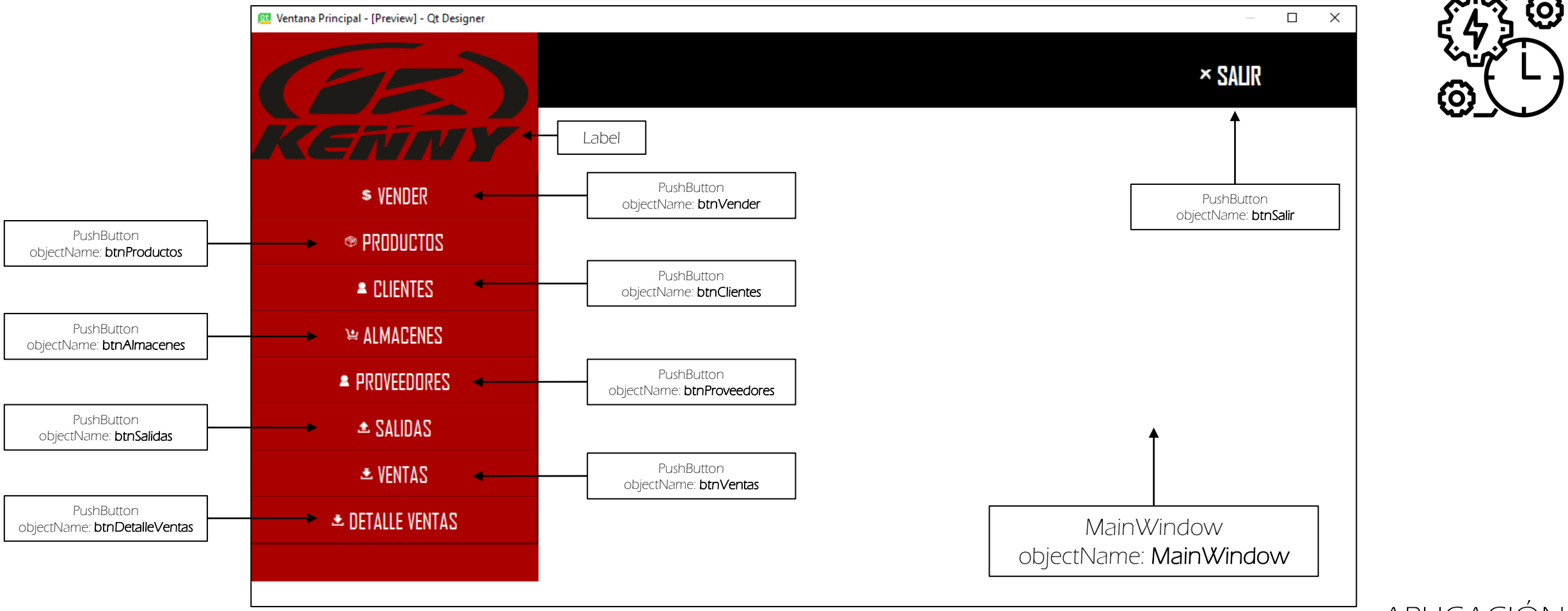
Tiempo: 50 minutos

Ejercicios



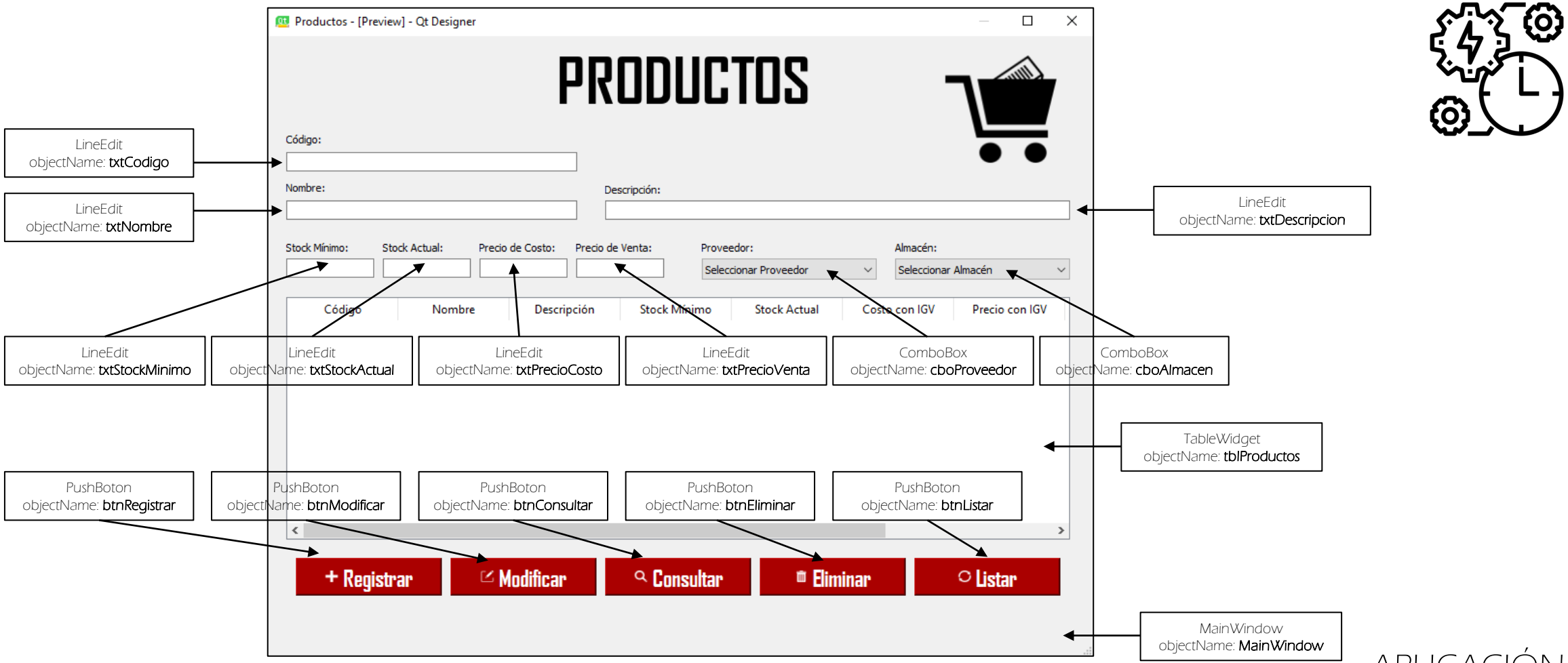
APLICACIÓN

Ejercicios



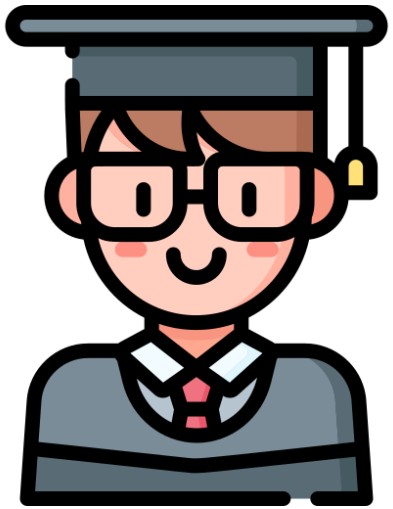
APLICACIÓN

Ejercicios



APLICACIÓN

Término



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