

Trabajo realizado por Cristina Sandoval Laborde

Ejemplo de personalización componentes

Formulario de registro

Nombre:

Apellidos:

Número de telefono:

Correo electrónico:

Selecciona la ubicacion de la academia:

Selecciona tu género:

☐ Hombre

☐ Mujer

☐ Otro

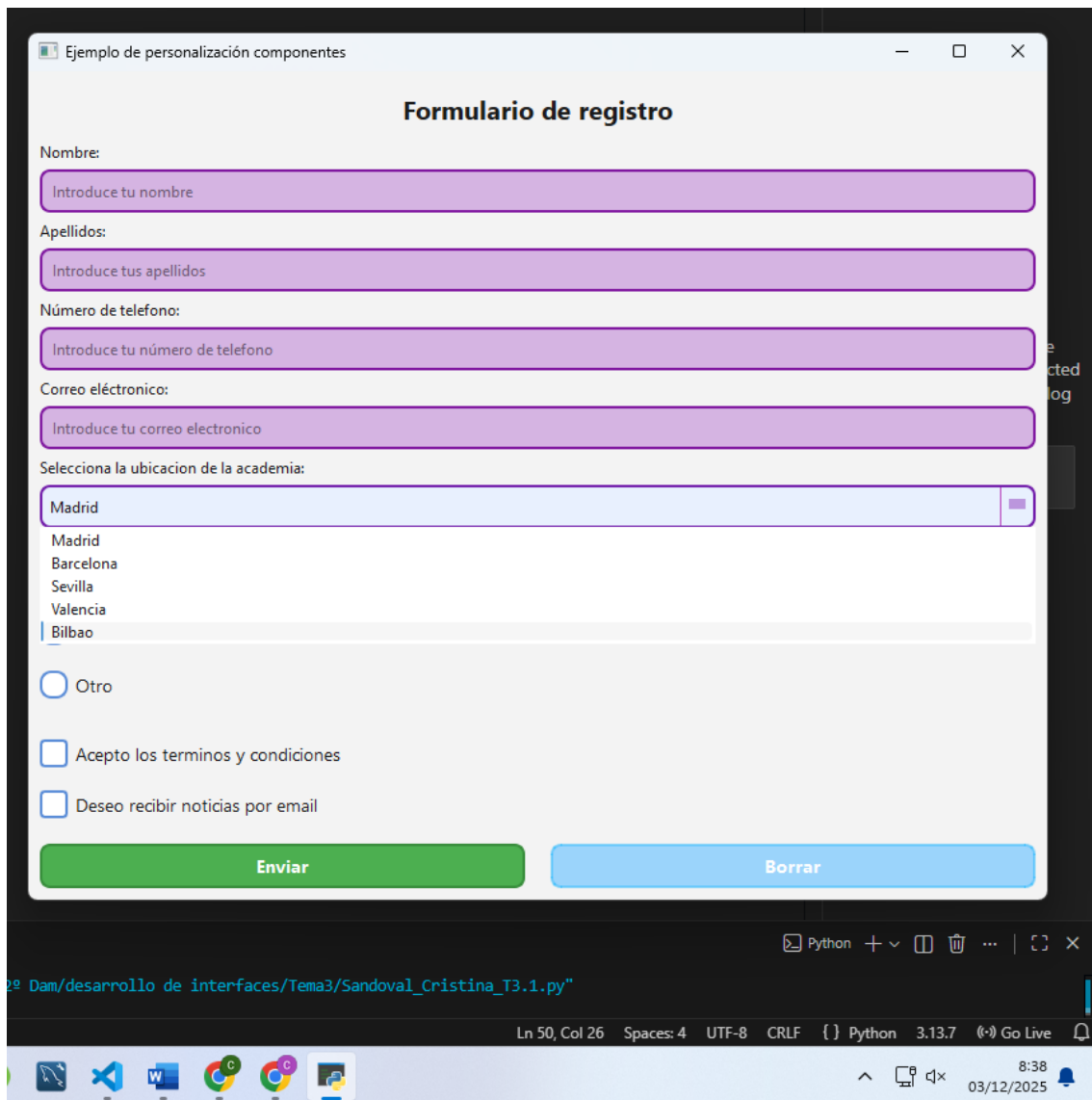
☐ Acepto los terminos y condiciones

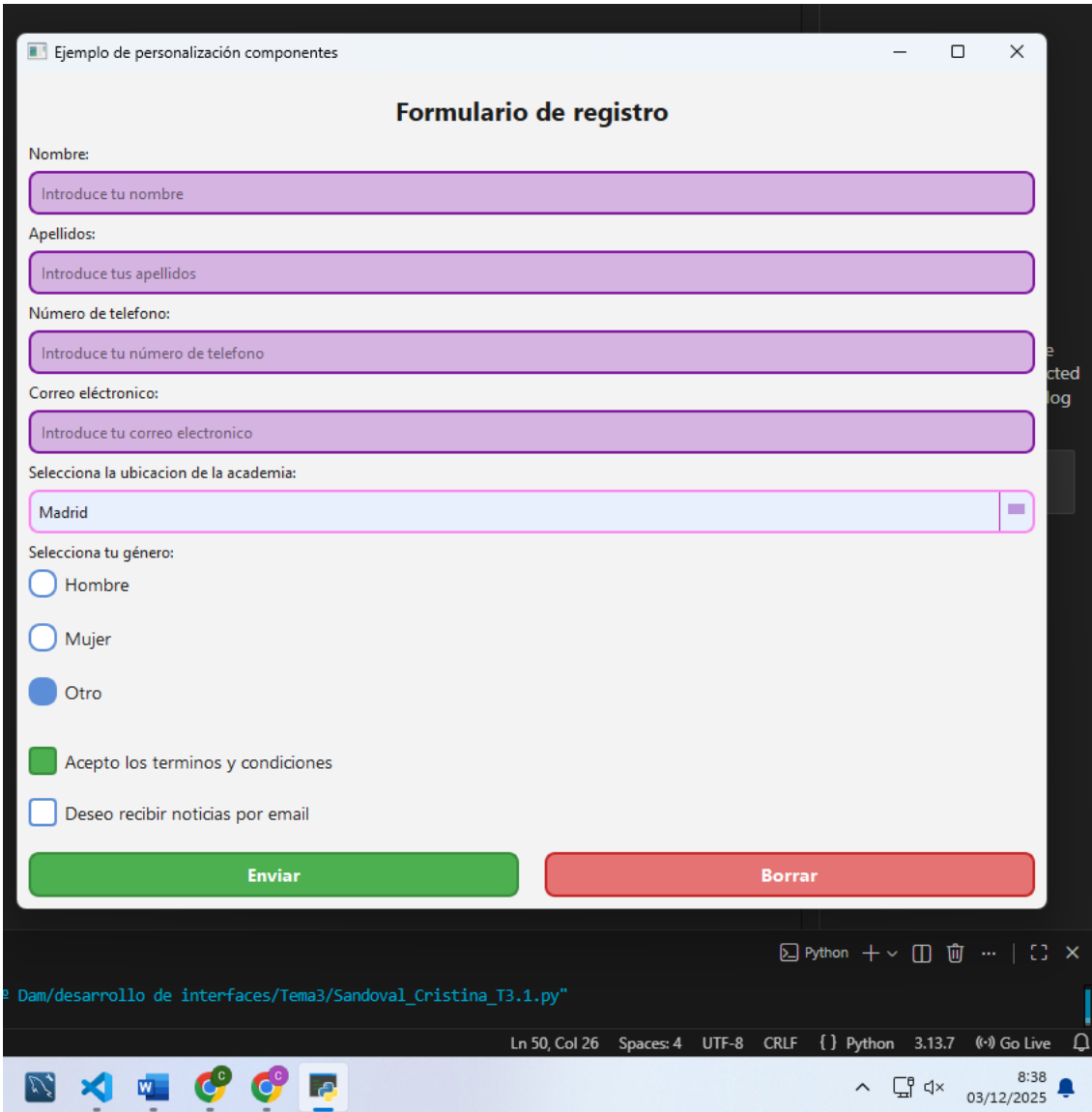
☐ Deseo recibir noticias por email

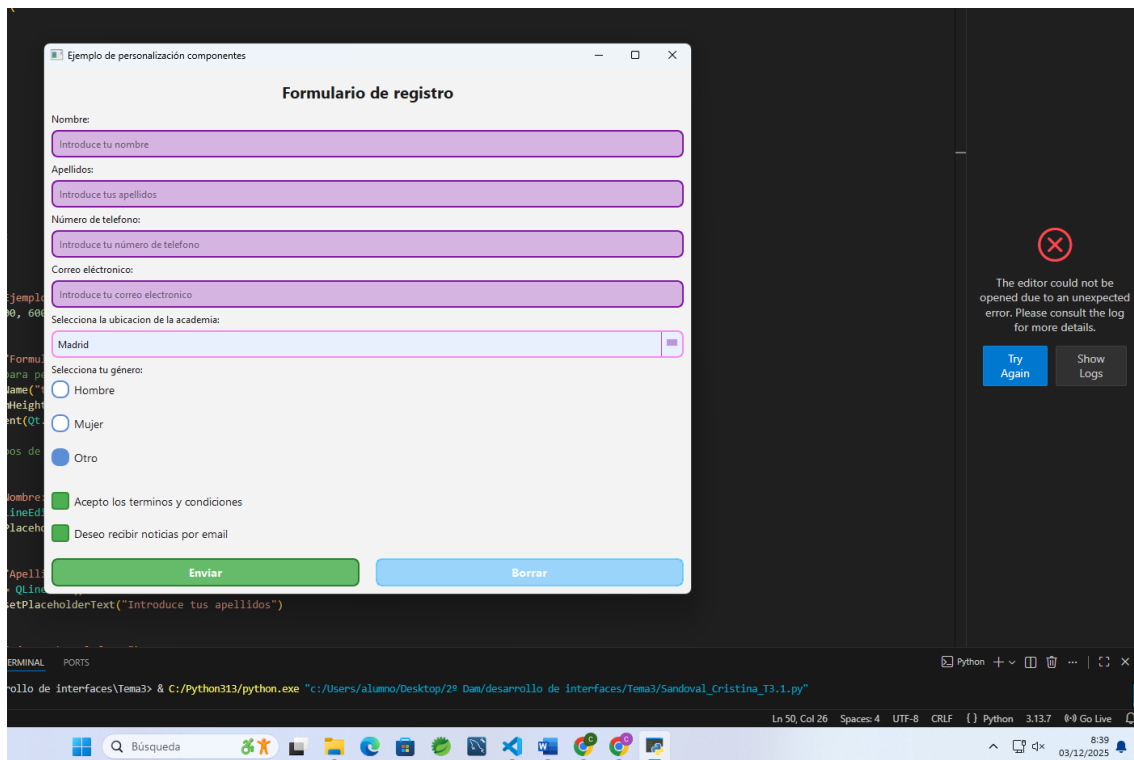
Dam/desarrollo de interfaces/Tema3/Sandoval_Cristina_T3.1.py

Ln 50, Col 26 Spaces: 4 UTF-8 CRLF { } Python 3.13.7 Go Live

8:38 03/12/2025







Qué widget ha sido más sencillo

El QLabel: Es el más simple porque solo muestra texto y la personalización es sencilla. Si le asignas un objectName es muy fácil aplicarle un estilo en el QSS.

Ejemplo del mio

```
# --- Titulo ---
self.titulo = QLabel("Formulario de registro")
#Darle un objectName para personalizar el estilo en el qss
self.titulo.setObjectName("titulo")
self.titulo.setMinimumHeight(40)
self.titulo.setAlignment(Qt.AlignCenter)
```

```
QLabel#titulo {
    font-size: 20px;
    font-weight: bold;
}
```

Qué widget ha sido más complejo

El más completo para mí ha sido el QPushButton con los botones (Enviar y Borrar), porque requieren una personalización mayor con varios estados y deben reaccionar cuando se pasa el ratón por encima o hace clic.

Qué pseudo-estados se han usado y por qué

En el QPushButton

:hover → cuando el ratón pasa por encima.

:pressed → cuando el usuario hace clic.

Para que cambien de color, según se pone el ratón encima o se hace clic.

Mis ejemplos:

```
QPushButton#btnEnviar:hover {
    background-color: #66BB6A;
    border-color: #2E7D32;
}

QPushButton#btnEnviar:pressed {
    background-color: #388E3C;
    border-color: #1B5E20;
}

QPushButton#btnBorrar {
    background-color: rgb(16, 119, 30);
    color: white;
    padding: 6px 12px;
    border: 2px solid rgb(12, 109, 3);
    border-radius: 8px;
    font-size: 14px;
    font-weight: bold;
}

QPushButton#btnBorrar:hover {
    background-color: #E57373;
    border-color: #C62828;
}

QPushButton#btnBorrar:pressed {
    background-color: #C62828;
    border-color: #B71C1C;
}
```

En el caso del CheckBox

QCheckBox::indicator:hover

- Se activa cuando el ratón pasa por encima del checkbox.
- Cambia el color del borde para dar una sensación de interacción.

QCheckBox::indicator:checked

- Se aplica cuando el checkbox está **marcado**.
- Cambia el fondo a verde (background-color) y el borde a un verde más oscuro.
- Así el usuario ve claramente que está activado.

```
QCheckBox::indicator:hover {  
    border-color: #7AA8F0;  
}  
  
QCheckBox::indicator:checked {  
    background-color: #4CAF50;  
    border: 2px solid #388E3C;  
}
```

Webgrafia

<https://experts-deny-b9a.craft.me/WAWe2diIHficZK>