

TAREA 3.1 - PERSONALIZACIÓN VISUAL DE COMPONENTES CON QSS

Tarea realizada por Daniel García Méndez

The screenshot shows a Windows application window titled "Formulario Gimnasio". The main title bar is dark blue with white text. Below it, the window has a light gray header area with the title "Formulario de Inscripción". The form consists of several input fields and buttons. At the top, there are four input fields with placeholder text: "Introduce tu nombre", "Introduce tu DNI", "Introduce tu teléfono", and "Introduce tu dirección". Below these is a checkbox labeled "Acepto los términos y condiciones". A section titled "selecciona tu tipo de membresía:" contains two radio buttons: "Membresía Básica" and "Membresía Premium". Another section titled "selecciona una actividad:" has a single input field with placeholder text "Selecciona una actividad" and a green "Enviar" button below it.

- **qué widget ha sido más sencillo:** El **QLabel** porque es un widget simple que no requiere tener mucho conocimiento para usarlo, solo pones **QLabel("texto a mostrar")** y funcionaría todo correctamente.
- **qué widget ha sido más complejo:** El widget que me han sido los botones, debido a sus pseudo estados.
- **qué pseudo-estados se han usado y por qué**
En esa hoja de estilos he usado varios pseudo-estados para que los widgets cambien de aspecto según el estado o la interacción del usuario.

Qué pseudo-estados hay

En mi código aparecen estos pseudo-estados:

- QLineEdit:focus
- QCheckBox::indicator:hover
- QCheckBox::indicator:checked
- QRadioButton::indicator:hover
- QRadioButton::indicator:checked
- QComboBox:hover
- QPushButton:hover
- QPushButton:pressed

Por qué se usa cada uno

- **:focus** en **QLineEdit** sirve para resaltar el campo que tiene el foco de teclado, esto cambia el borde y fondo para que el usuario sepa en qué caja está escribiendo.
- **:hover** en **QCheckBox::indicator**, **QRadioButton::indicator**, **QComboBox** y **QPushButton** al pasar el ratón por encima, se afina el borde o el color para indicar que el usuario ha interactuado con este.
- **:checked** en los indicadores de **QCheckBox** y **QRadioButton** diferencia de forma visual el estado marcado del no marcado (se cambia el fondo y el color del borde a verde o azul según el caso).
- **:pressed** en **QPushButton** crea el efecto de presionar el botón mientras se mantiene pulsado.

Ejemplos de pseudo estados utilizados en el código:

```
QRadioButton::indicator {
    width: 16px;
    height: 16px;
    border-radius: 8px;
    border: 2px solid #05eefb;
    background: #ecf0f1;
}

QRadioButton::indicator:hover {
    border: 2px solid #3498db;
}

QRadioButton::indicator:checked {
    background: #3498db;
    border: 2px solid #3498db;
}
```

```
QCheckBox::indicator {
    width: 18px;
    height: 18px;
    border-radius: 4px;
    border: 2px solid #db470d;
    background: #ecf0f1;
}

QCheckBox::indicator:hover {
    border: 2px solid #3498db;
}

QCheckBox::indicator:checked {
    background: #2ecc71;
    border: 2px solid #2ecc71;
}
```

```
QComboBox::hover {
    border: 2px solid #2ecc71;
}
QComboBox::drop-down {
    width: 24px;
    border-left: 1px solid #bdc3c7;
}

QComboBox::down-arrow {
    background-color: #2980b9;
}
```

```
QPushButton:hover {
    background-color: #27ae60;
}
QPushButton:pressed {
    background-color: #1e8449;
}
```

Webgrafía:

<https://experts-deny-b9a.craft.me/I08WXLxkQnVpWv>