

Ejercicio 1: Recepción de hotel

- **Descripción:** Esta aplicación de escritorio Win32 actúa como un sistema de recepción de hotel. Su funcionalidad principal es capturar y mostrar el texto que el usuario introduce a través del teclado.
- **Mensajes de Windows usados:**
 - `WM_CHAR`: Captura cada carácter tecleado por el usuario.
 - `WM_PAINT`: Se encarga de repintar la ventana para mostrar el texto actualizado.
- **Actualización de estado:**
 - La variable `g_typed` de tipo `std::wstring` almacena la cadena de caracteres introducida por el usuario. Cuando se recibe un mensaje `WM_CHAR`, esta cadena se modifica: se añade el nuevo carácter o se elimina el último si se presiona la tecla de retroceso (`\b`). Después de cada modificación, se llama a `InvalidateRect` para solicitar un repintado de la ventana.

Ejercicio 2: Contador de entradas

- **Descripción:** Esta aplicación simple cuenta las "entradas vendidas" a través de clics del ratón, con el botón izquierdo para aumentar y el derecho para disminuir el contador.
- **Mensajes de Windows usados:**
 - `WM_LBUTTONDOWN`: Se activa al hacer clic con el botón izquierdo del ratón.
 - `WM_RBUTTONDOWN`: Se activa al hacer clic con el botón derecho del ratón.
 - `WM_PAINT`: Se encarga de dibujar el contador de entradas en la ventana.
- **Actualización de estado:**
 - La variable `g_clicks` de tipo `int` se usa para almacenar el número de clics. Cada vez que se recibe un mensaje `WM_LBUTTONDOWN`, `g_clicks` se incrementa. Si se recibe un mensaje `WM_RBUTTONDOWN`, `g_clicks` se decrementa, siempre que sea mayor que cero. Después de cada cambio, se llama a `InvalidateRect` para actualizar la visualización del contador.

Ejercicio 3: Página responsive

- **Descripción:** Esta aplicación demuestra cómo una ventana puede responder a los cambios de tamaño, repintando su contenido en consecuencia. La aplicación muestra información de tamaño del área del cliente en un panel.
- **Mensajes de Windows usados:**
 - `WM_SIZE`: Se dispara cada vez que la ventana cambia de tamaño.
 - `WM_PAINT`: Es el responsable de dibujar el contenido de la ventana, incluyendo el panel con la información del tamaño.
- **Actualización de estado:**
 - Las variables `g_cx` y `g_cy` almacenan el ancho y el alto del área del cliente de la ventana, respectivamente. Cuando se recibe un mensaje `WM_SIZE`, estas

variables se actualizan con los nuevos valores de tamaño proporcionados por el parámetro `lParam` del mensaje.

Ejercicio 4: Pizarra

- **Descripción:** Esta es una aplicación de dibujo básica que muestra un texto estático de bienvenida e instrucciones. El mensaje se dibuja directamente en la ventana cuando esta necesita ser repintada.
- **Mensajes de Windows usados:**
 - `WM_PAINT`: Se encarga de dibujar el mensaje de instrucciones en el centro de la ventana.
- **Actualización de estado:**
 - En este ejercicio, no hay variables de estado que se actualicen en respuesta a los mensajes, ya que el contenido de la ventana es fijo y no interactivo.

Ejercicio 5: Ascensor

- **Descripción:** Esta aplicación simula un panel de ascensor con tres botones. Al hacer clic en un botón, la aplicación registra cuál fue el último piso visitado y lo muestra en un área de texto.
- **Mensajes de Windows usados:**
 - `WM_CREATE`: Se utiliza para inicializar la posición y el tamaño de los rectángulos que representan los botones y el área de texto.
 - `WM_LBUTTONDOWN`: Captura los clics del botón izquierdo del ratón.
 - `WM_PAINT`: Se encarga de dibujar los botones y el área de texto con el último piso visitado.
- **Actualización de estado:**
 - La variable `ultimoPisoVisitado` de tipo `int` almacena el número del último botón presionado. Cuando se recibe un mensaje `WM_LBUTTONDOWN` y se detecta que el clic ocurrió dentro de uno de los rectángulos de los botones, el valor de `ultimoPisoVisitado` se actualiza al piso correspondiente.

Ejercicio 6: Caja registradora

- **Descripción:** Esta aplicación simula la entrada de un importe monetario en una caja registradora. El usuario puede introducir dígitos y una coma decimal, y el importe se muestra en la ventana.
- **Mensajes de Windows usados:**
 - `WM_CHAR`: Se utiliza para capturar cada pulsación de tecla, incluyendo dígitos y la coma decimal.
 - `WM_PAINT`: Dibuja el texto del importe actual en la ventana.
- **Actualización de estado:**
 - La variable `importe` de tipo `std::wstring` almacena la cadena del importe. Cuando se recibe un mensaje `WM_CHAR`, la cadena se actualiza. La aplicación añade el carácter introducido si es un dígito o una coma (solo la primera), y elimina el último carácter si se presiona la tecla de retroceso. Se

llama a `InvalidateRect` para solicitar un repintado de la ventana cada vez que la cadena cambia.

Ejercicio 7: Visor de eventos

- **Descripción:** Esta herramienta de depuración registra y muestra los últimos 10 mensajes de Windows que recibe la ventana. Es útil para entender el flujo de mensajes en una aplicación.
- **Mensajes de Windows usados:**
 - `WM_PAINT`: Se encarga de dibujar la lista de eventos.
 - **Otros mensajes:** La aplicación captura y muestra cualquier otro mensaje (excepto `WM_PAINT`, `WM_NCPAINT` y `WM_ERASEBKGD`), como `WM_LBUTTONDOWN`, `WM_MOUSEMOVE` y `WM_SIZE`.
- **Actualización de estado:**
 - Se utiliza un `std::deque` estático llamado `ultimosEventos` para almacenar los mensajes recientes. Antes del `switch` principal del `WndProc`, se procesa cada mensaje entrante, se formatea a una cadena y se añade al inicio del `deque`. Si el número de eventos supera 10, el evento más antiguo se elimina del final.

Ejercicio 8: Panel de mostrador

- **Descripción:** Esta aplicación muestra en paneles separados el nombre introducido por teclado, el número de clics del ratón y el tamaño actual de la ventana. Los paneles se redimensionan automáticamente al cambiar el tamaño de la ventana.
- **Mensajes de Windows usados:**
 - `WM_SIZE`: Captura los cambios de tamaño de la ventana.
 - `WM_LBUTTONDOWN`: Se activa con los clics del botón izquierdo del ratón.
 - `WM_CHAR`: Captura la entrada de texto del teclado.
 - `WM_PAINT`: Se encarga de dibujar los tres paneles con la información actualizada.
- **Actualización de estado:**
 - Se actualizan varias variables de estado:
 - `g_ancho` y `g_alto`: se actualizan con las nuevas dimensiones de la ventana cuando se recibe `WM_SIZE`.
 - `g_contadorClics`: se incrementa cada vez que se recibe `WM_LBUTTONDOWN`.
 - `g_nombre`: se modifica al recibir `WM_CHAR` para añadir o eliminar caracteres, reflejando el texto introducido por el usuario.