

# TAREA 3 y 4 — Análisis y Mejora de la Interfaz de Usuario (Win32 C++)

**Alumno:** Ignacio Delgado

**Profesora:** Mónica Nano

**Materia:** PGE

**Fecha:** 18/08/2025

**Título del trabajo:** Unificación del flujo de compra de pasajes (inspirado en GoToGate) en una sola pantalla con selección de asientos y pago

---

## Punto 1: Software analizado y problemas de usabilidad encontrados

El software elegido para el análisis es **Go To Gate**, una plataforma de compra de pasajes aéreos utilizada a nivel internacional.

Si bien cumple con su objetivo principal (permitir reservar y pagar vuelos), en la práctica presenta varios problemas de usabilidad que hacen que la experiencia de compra sea **más difícil, confusa y propensa a errores**.

### Problemas principales identificados:

- **Flujo dividido en múltiples pantallas:** el proceso está separado en tres etapas (1. Datos personales, 2. Selección de asientos, 3. Pago). Esto obliga al usuario a navegar hacia adelante y atrás, perdiendo contexto de lo que ya completó.
- **Riesgo de pérdida de datos:** si ocurre un error en la conexión o el sistema expira la sesión, se pierde todo lo que se había cargado en pasos previos.
- **Aumento de la carga cognitiva:** el usuario debe recordar qué completó en cada pantalla, en lugar de tener una visión global del proceso.
- **Selector de asientos poco intuitivo:** se presenta en una página aparte, con un diseño que no siempre es claro ni amigable.
- **Validaciones tardías:** los errores en los datos ingresados (por ejemplo, en la edad o en el pago) recién se detectan al final, cuando ya se pasó por varias pantallas.

**Conclusión:** El principal problema de Go To Gate es que **divide un proceso que debería ser continuo en varias pantallas**. Esto aumenta la frustración, el tiempo invertido y la posibilidad de perder el trabajo hecho.

---

## Punto 2: Propuesta de mejora

Para resolver estos problemas, propusimos y diseñamos un **prototipo en Win32 (Visual Studio 2022)** que unifica el flujo de compra en una **única interfaz integrada**.

### Mejoras implementadas en nuestro prototipo:

- **Pantalla única:** todos los pasos (datos personales, elección de asiento y pago) se encuentran en la misma aplicación, evitando saltos entre pantallas y reduciendo el riesgo de perder información.
- **Resumen del vuelo:** al inicio se muestra claramente el vuelo elegido con su precio, para que el usuario tenga siempre la referencia.
- **Formulario de datos personales más simple:** se agregaron campos claros (nombre, apellido, edad, sexo) con validaciones inmediatas para evitar errores posteriores.
- **Mapa de asientos más intuitivo:** al presionar un botón se abre una ventana auxiliar con una grilla de 20 asientos, fáciles de seleccionar y con confirmación inmediata.
- **Pago en la misma vista:** se incorporó la carga de tarjeta directamente, con validaciones locales (número de tarjeta válido, vencimiento correcto, CVV) y enmascarado del número para mayor seguridad.
- **Confirmación clara:** antes de finalizar, el sistema muestra un resumen del vuelo, los datos ingresados y el asiento elegido, de forma que el usuario sabe exactamente qué está comprando.

### Beneficios de estas mejoras:

- Se elimina el riesgo de perder datos por saltar entre páginas.
- Se reduce el tiempo de compra, ya que todo ocurre en un solo flujo.
- El usuario recibe **feedback inmediato** en cada acción (ejemplo: validación de tarjeta o asiento elegido).
- El proceso es más **intuitivo y visual**, con menos fricción.

---

## Punto 3: Explicación detallada del prototipo

El prototipo se implementó en **Win32 con C++**, siguiendo la arquitectura tradicional de aplicaciones de escritorio Windows.

### Diseño de la aplicación:

1. **Ventana principal:** contiene el resumen del vuelo, formulario de datos personales, campos de pago y botones de acción (Elegir asiento, Confirmar compra, Cancelar).
2. **Ventana de asientos (auxiliar):** despliega una rejilla de 20 botones (asientos), con Aceptar/Cancelar. Una vez confirmado, el asiento se refleja en la ventana principal.
3. **Confirmación final:** al completar los datos y pago, se muestra un resumen en la misma ventana, simulando la confirmación de la compra.

### Cómo se implementaron las funcionalidades clave:

- **Callbacks:** cada ventana tiene un procedimiento de ventana que recibe los mensajes de Windows (clic de botones, cierre, creación de controles). Esto nos permitió manejar cada acción del usuario en tiempo real.
- **Bucle principal (message loop):** mantiene activa la aplicación y procesa todos los eventos de entrada (teclado, ratón, etc.), asegurando la fluidez de la experiencia.
- **Eventos Paint:** se usaron para redibujar la interfaz cuando fue necesario (por ejemplo, al actualizar el asiento seleccionado en la pantalla principal).
- **Funciones de conversión y validación:** se programaron utilidades para validar los campos:
  - Verificación del número de tarjeta con el algoritmo de Luhn.
  - Validación de fecha de vencimiento en formato MM/AA.
  - Validación del CVV como campo numérico.
  - Enmascaramiento del número de tarjeta, mostrando solo los últimos 4 dígitos.

### Síntesis del prototipo:

- Resolvimos el principal problema de Go To Gate (flujo fragmentado) al integrar todo en una sola aplicación.
- El prototipo refleja un **proceso continuo, rápido y seguro**, sin depender de múltiples pasos que puedan hacer perder información.

- Se aplicaron técnicas de programación propias de Win32 (callbacks, loop de mensajes, eventos paint) y validaciones básicas para simular una experiencia de compra **más confiable y amigable**.