Piloto de Monetización CSFJ

Documentación Técnica para Desarrolladores

Alejandro Caicedo

26 de noviembre de 2025

Índice

1. Descripción General

Aplicación web para gestionar y calcular costos de items relacionados con actividades de monetización de la Convocatoria San Francisco Javier (CSFJ). Construida completamente en C++17 con un servidor HTTP integrado, sin dependencias externas.

1.1. Tecnologías Utilizadas

- **Backend:** C++17 con servidor HTTP nativo (sockets)
- **Frontend:** HTML5, CSS3, JavaScript vanilla
- **Build System:** CMake 3.16+
- **Compatibilidad:** Windows (Winsock2), Linux/macOS (POSIX sockets)

2. Características Principales

- **Agregar items** con lista desplegable de 10 categorías predefinidas:
  - Hora docente, Viáticos, Transporte, Material didáctico, Refrigerio
  - Alquiler de espacio, Equipamiento, Servicios profesionales, Publicidad, Certificaciones
  - Opción «Otro...» para items personalizados
- **Visualizar tabla** con cálculo automático de totales por item y total general
- **Editar items** existentes
- **Exportar a CSV** para análisis en Excel u otras herramientas
- **Formateo de moneda** en tiempo real con separadores de miles

3. Estructura del Proyecto

```
pilotoDeMonetizacionCSFJ/
|-- CMakeLists.txt          # Configuracion de compilacion CMake
|-- README.md               # Documentacion en markdown
|-- documentacion.tex       # Este archivo
|-- src/
|   +-- main.cpp            # Servidor HTTP y logica principal
   (~750 lineas)
|-- templates/
|   |-- index.html          # Pagina principal con formulario y
   tabla
|   +-- edit.html           # Pagina de edicion de items
+-- static/
    |-- styles.css          # Estilos CSS
    +-- formatter.js        # JavaScript para formateo de moneda
```

4. Requisitos del Sistema

- **CMake** 3.16 o superior
- **Compilador C++17:** Visual Studio 2019+, MinGW (MSYS2), GCC, o Clang
- **Sin dependencias externas** - solo librerías estándar del sistema

5. Compilación

5.1. Usando CMake (recomendado)

```
# Navegar al directorio del proyecto
cd pilotoDeMonetizacionCSFJ

# Generar archivos de build
cmake -B build -S .

# Compilar en modo Release
cmake --build build --config Release
```

El ejecutable se genera en:

- **Visual Studio:** `build/Release/pilotoDeMonetizacionCSFJ.exe`
- **MinGW:** `build/pilotoDeMonetizacionCSFJ.exe`

5.2. Compilación rápida con g++ (MinGW/MSYS2)

```
g++ -std=c++17 -Wall -Wextra -O2 src/main.cpp -lws2_32 -o
   pilotoDeMonetizacionCSFJ.exe
```

5.3. En Linux/macOS

```
g++ -std=c++17 -Wall -Wextra -O2 src/main.cpp -o
   pilotoDeMonetizacionCSFJ
```

6. Ejecución

1. Ejecutar el servidor:

   ```
   ./build/Release/pilotoDeMonetizacionCSFJ.exe
   ```

2. Abrir en el navegador: http://localhost:8080
3. Para detener el servidor: presionar `Ctrl+C` en la terminal

7. Guía de Uso

7.1. Agregar un Item

1. Seleccionar una categoría del dropdown o elegir «Otro...» para nombre personalizado
2. Ingresar la cantidad (número entero positivo)
3. Ingresar el costo unitario (el formato de moneda se aplica automáticamente)
4. Hacer clic en «Agregar»

7.2. Editar un Item

1. Hacer clic en el botón «Editar» de la fila correspondiente
2. Modificar los campos deseados
3. Hacer clic en «Guardar cambios»

7.3. Exportar a CSV

1. Hacer clic en el botón «Exportar CSV»
2. Se descargará un archivo `items.csv` con todos los items y totales

8. Documentación Técnica

8.1. Rutas HTTP Implementadas

Método	Ruta	Descripción
GET	`/`	Página principal con tabla de items
GET	`/index.html`	Alias de la página principal
GET	`/edit?index=N`	Página de edición del item en posición N
GET	`/export`	Descarga archivo CSV
GET	`/static/*`	Archivos estáticos (CSS, JS)
POST	`/submit`	Agregar nuevo item
POST	`/update`	Actualizar item existente

Cuadro 1: Rutas HTTP implementadas en el servidor

8.2. Almacenamiento de Datos

- Los datos se almacenan **únicamente en memoria** durante la ejecución
- Al detener el servidor, todos los datos se pierden
- Estructura de item: `{nombre, cantidad, costoUnitario}`
- Acceso thread-safe mediante mutex

8.3. Formato de Moneda

El formateo se realiza tanto en backend (C++) como en frontend (JavaScript):

- Separadores de miles alternados: ' y ,
- Siempre 2 decimales
- Ejemplo: `1'234,567.89`

9. Guía de Modificación

9.1. Agregar nuevas categorías al dropdown

Editar `templates/index.html`, buscar el elemento `<select id=ïtemDropdown»` y agregar:

```
<option value="Nueva Categoria">Nueva Categoria</option>
```

9.2. Cambiar el puerto del servidor

Editar `src/main.cpp`, buscar la constante:

```
constexpr int PORT = 8080;
```

Cambiar `8080` por el puerto deseado.

9.3. Agregar una nueva ruta HTTP

En `src/main.cpp`, dentro de la función `handleRequest()`:

1. Agregar la condición para la nueva ruta:

```
else if (method == "GET" && path == "/nueva-ruta") {
    // Tu logica aqui
    response = "HTTP/1.1 200 OK\r\n...";
}
```

2. Recompilar el proyecto

9.4. Modificar estilos

Editar `static/styles.css`. Los cambios se reflejan al recargar la página (no requiere recompilar).

9.5. Agregar persistencia de datos

Para guardar datos de forma permanente, se necesitaría:

1. Agregar una librería de base de datos (SQLite recomendado) o
2. Implementar guardado/lectura a archivo JSON/CSV
3. Modificar las funciones de agregar/editar/eliminar items

10. Limitaciones Conocidas

- **Sin persistencia:** Los datos existen solo mientras el servidor está activo
- **Sin autenticación:** Cualquier usuario en la red puede acceder
- **Single-threaded:** Procesa una solicitud a la vez (suficiente para uso individual)
- **Sin HTTPS:** Las conexiones no están cifradas

11. Solución de Problemas

11.1. El puerto 8080 está ocupado

```
# Ver que proceso usa el puerto
netstat -ano | findstr :8080

# Terminar el proceso (reemplazar PID con el numero obtenido)
taskkill /PID <PID> /F
```

O cambiar el puerto en `src/main.cpp` (ver sección de modificación).

11.2. Error de compilación: ws2_32 not found

Asegurarse de estar en Windows y que el compilador tiene acceso a las librerías del sistema. Con MinGW, verificar que MSYS2 esté correctamente instalado.

11.3. Los templates no se encuentran

Ejecutar el servidor desde el directorio raíz del proyecto, donde se encuentran las carpetas `templates/` y `static/`.

11.4. Los cambios en HTML/CSS no se reflejan

Limpiar la caché del navegador (`Ctrl+Shift+R`) o abrir en modo incógnito.

Licencia

Proyecto interno de la Convocatoria San Francisco Javier.