Taller I

(22/03 – 05/04)

# Gestión de Eventos y Asistentes en C++

## Descripción del Problema

Un organizador de eventos necesita un programa para gestionar la información de los eventos que organiza y llevar un registro de los asistentes a cada evento. El sistema debe permitir la creación de nuevos eventos, el registro de asistentes, la consulta de la lista de asistentes y la generación de informes.

## Requerimientos Técnicos

El programa debe cumplir con los siguientes requisitos técnicos:

### Programación Orientada a Objetos (POO)

* Utilizar clases para representar eventos y asistentes.
* Aplicar conceptos de encapsulamiento, abstracción, herencia (si es necesario) y polimorfismo (opcional).

### Uso de Punteros y Arrays

* Emplear punteros y arrays para gestionar la información de eventos y asistentes de manera dinámica.
* Garantizar una gestión adecuada de la memoria asignada y liberada.

### Interfaz de Usuarios

* Proporcionar opciones para crear eventos, registrar asistentes, consultar la lista de asistentes y generar informes.

### Persistencia de Datos

* Implementar funciones para cargar y guardar datos en un archivo, de modo que la información se conserve entre sesiones.

## Detalles Adicionales

### Tipos de Eventos:

* Los eventos podrían clasificarse en diferentes tipos, como conferencias, talleres, seminarios, conciertos, etc. Cada tipo de evento podría tener atributos específicos, como duración, ubicación, tema, etc.

### Tipos de Asistentes:

* Los asistentes podrían ser de diferentes tipos, como estudiantes, profesionales, invitados especiales, etc. Cada tipo de asistente podría tener atributos específicos, como nombre, edad, ocupación, empresa/institución, etc.

### Información en los Informes:

* Los informes generados podrían incluir datos como:
* Lista de eventos programados.
* Lista de asistentes registrados para cada evento.
* Estadísticas sobre la asistencia a los eventos (por ejemplo, número total de asistentes, asistencia promedio por evento, etc.).
* Detalles sobre los asistentes (por ejemplo, edad promedio, ocupaciones más comunes, etc.).
* Información sobre eventos específicos, como la duración, la ubicación y el tema.

## Entregables

* Se debe subir un link de github con el proyecto en visualización publica a campus virtual con un plazo máximo hasta el 05/04/2024.
* El programa debe ser capaz de compilar con g++ (buscar información sobre mingw o utilizar codespace de github para realizar el proyecto).
* Dentro del README del repositorio se debe incluir toda la información sobre los objetos que se crean en el main durante el inicio del programa. Por ejemplo:

*admin1: { nombre:”Jose”; contraseña:”123123”; edad:”30”; correo:”*[*admin@admin.cl*](mailto:admin@admin.cl)*”}*

* Dentro del README del repositorio se debe incluir la información relacionada al equipo de trabajo (Max 2 integrantes: Nombre completo, rut y correo institucional)

## Evaluación

| Criterio | Puntaje |
| --- | --- |
| Diseño y estructura del programa | 0/10 |
| Implementación de clases y métodos | 0/10 |
| Uso correcto de punteros y arrays | 0/10 |
| Funcionalidad y precisión del programa | 0/10 |
| Persistencia de datos | 0/5 |
| Cumplimiento de los requisitos solicitados | 0/5 |
| Presentación y organización del código | 0/5 |
| Total | 0/55 |

* Fecha de entrega 05/04 23.59 a través de campus virtual. Consultas -> jose.veas@ce.ucn.cl
* Se evaluará con nota 1 en caso de:
* Copia
* No entregar documentación
* Programa no compila
* No entregar en fechas indicadas

\*El profesor se reserva el derecho de solicitar revisión con alumno en caso de encontrarse con situaciones sospechosas que podrían modificar la evaluación.