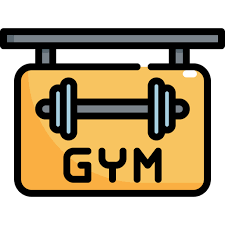
**Proyecto 1**



**PowerLab**

**Gestión de Gimnasios “PowerLab”**

**Fecha de entrega: Domingo 28 de setiembre 2025.**

**Porcentaje de la calificación del curso:** 10%.

**Objetivo general:**

Aplicar la definición y el uso de colecciones tipo arreglo unidimensional (vector), punteros, memoria dinámica y relaciones entre clases, al diseño e implementación de un programa o aplicación para la gestión administrativa de una cadena de Gimnasios.

**Objetivos específicos:**

* Resolver el problema en forma eficiente y haciendo uso de los principios del Paradigma de Programación Orientada a Objetos (POO).
* Realizar un adecuado diseño de clases.
* Definir e implementar correctamente las relaciones entre clases.
* Definir y utilizar adecuada y eficientemente arreglos unidimensionales dinámicos.
* Administrar eficientemente el uso de memoria dinámica.
* Implementar correctamente el código requerido en el lenguaje C++.
* Crear una interfaz de usuario en consola que permita interactuar con la aplicación de forma amigable e intuitiva.
* Implementar todas las funcionalidades de la aplicación que se especifican.

**Gestión de Gimnasios “PowerLab”:**

Debido al acelerado crecimiento de la cadena de gimnasios “**PowerLab**”, los dueños han visto la necesidad de crear un sistema computacional para administrar los datos de todas las sedes o sucursales del gimnasio en el país y le ha solicitado a su compañía el diseño e implementación del sistema. Por el momento, la cadena cuenta con 10 sucursales, no obstante, se espera un crecimiento que no supere las 30 sucursales.

**Funcionalidades**

El sistema debe permitir ingresar sucursales, para cada sucursal es requerido conocer el código, provincia, cantón, correo electrónico y teléfono, entre otros datos

El sistema debe permitir ingresar nuevos instructores para cada sucursal, de los instructores se debe registrar: número de cédula, nombre completo, teléfono, correo, fecha de nacimiento, y una o varias de las siguientes especialidades: CrossFit, HIIT, TRX, Pesas, spinning, Cardio, Yoga, Zumba (un instructor puede tener más de una especialidad).

Adicionalmente el sistema debe permitir registrar a los clientes de cada sucursal. De cada cliente se requiere conocer: cédula de identidad, nombre completo, teléfono de contacto, correo electrónico, fecha de nacimiento, sexo, fecha de inscripción, entre otros más.

Cada cliente debe tener registrado un instructor asociado, quien lo guía durante su entrenamiento en la sucursal especifica. Los instructores se encargan, además, de realizar **mediciones**  a los clientes, así como de la elaboración de **rutinas de ejercicios**.

Los clientes deben ser valorados periódicamente por su instructor asignado, para lo cual se genera y registra un **reporte de medición**, el cual contempla la siguiente información:

Nombre del cliente, id del cliente, fecha de la medición, nombre del instructor, peso, estatura, porcentaje de grasa, porcentaje de músculo, edad metabólica, porcentaje de grasa visceral (grasa que cubre los órganos), medidas corporales( cintura, cadera, pecho y muslo), IMC, entre otros datos más a definir.

El Índice de Masa Corporal (IMC) lo debe calcular el sistema según los datos ingresados para el cliente. El sistema debe generar rangos de IMC, según la siguiente tabla:

|  |  |
| --- | --- |
| **Clasificación** | **IMC (kg/m²)** |
| Delgadez severa | <16,00 |
| Delgadez moderada | 16,01 - 16,99 |
| Delgadez leve | 17,00 - 18,49 |
| Normal | 18,5 - 24,99 |
| Pre-obesidad | 25,00 - 29,99 |
| Obesidad leve | 30,00 - 34,99 |
| Obesidad media | 35,00 - 39,99 |
| Obesidad mórbida | ≥40,00 |

El Ministerio de Salud establece que cuando una persona padece obesidad leve, media o mórbida, debe ser considerado como un paciente de alto riesgo.

Según la Organización Mundial de la Salud, los clientes deben consumir 0.8 gramos de proteína por cada kilogramo de peso. Si es hombre y hace ejercicio esta cantidad debe aumentar a un valor entre 1.7 - 2.5 y si es mujer y hace ejercicio a un valor entre 1.6 – 1.8. Para calcular la cantidad de vasos de agua de 250 ml que debe tomar una persona, se debe dividir el peso en kilos entre 7.

Al finalizar la cita de medición se debe generar un reporte con todos los datos del cliente: mediciones, cálculos, IMC, recomendación de cuántos vasos con agua debe tomar al día(calculado por el sistema), así como la cantidad de ingesta de proteínas requerida según su peso (calculado por el sistema), entre otros

El sistema debe permitir registrar un **historial** de mediciones realizadas a cada cliente. Dicho historial de medición podrá ser consultado para visualizar la evolución del cliente. De cada cliente se registra un máximo 10 mediciones.

El sistema además debe permitir que el instructor genere la **rutina** actual del cliente a partir de una batería de ejercicios previamente registrados y clasificados en el sistema. La rutina se divide en ejercicios para las siguientes áreas del cuerpo: pecho y tríceps, bíceps, piernas y espalda.

*Nota: No se guardan rutinas anteriores asignadas al cliente, únicamente se mantiene en el sistema la rutina actual, la cual puede cambiar en el momento que así lo considere el instructor.*

El gimnasio brinda en cada sucursal un máximo de 8 clases grupales, entre las siguientes: crossFit, HIIT, TRX , pesas, spinning , cardio, yoga, zumba. *Nota: Observe que las clases corresponde con las mismas especialidades de los instructores.*

Cada clase grupal tiene un código, una capacidad o cupo máximo, un salón especifico, instructor *(de la especialidad correspondiente),* un horario entre otros.

El sistema debe permitir matricular clientes en las diferentes clases grupales con un máximo de 3 clases por persona *(si una clase está llena no se permite la matricula).*

**Instrucciones Generales:**

* Se evaluará tanto la funcionalidad como la eficiencia y la aplicación adecuada y correcta del POO en el diseño e implementación del sistema.
* Con el objetivo de agilizar las pruebas y el desarrollo por parte del programador, se le recomienda crear alguna función que alimente el sistema con datos “**quemados”[[1]](#footnote-1)**. Sin embargo, aun así, el sistema debe contar con las opciones del menú necesarias para ingresar nuevos datos por parte del usuario *(nuevas sucursales, clientes, instructores, clases, ejercicios, rutinas, mediciones, entre otros)*.
* Los arreglos y los objetos que se utilicen deben crearse con asignación dinámica de memoria.
* Las colecciones que manejen arreglos unidimensionales o similares deben implementarse tal como se ha hecho en el curso. No podrán utilizarse bibliotecas del lenguaje que soporten estructuras de datos o métodos similares, ni hacer uso de la librerías de la STL.
* El proyecto debe entregarse con diagrama de clases y relaciones y con código fuente completo.
* Cualquier programa que no compile o que tenga errores de compilación recibirá una calificación de cero.
* El programa no deberá descontinuarse abruptamente en ningún momento, por lo cual se deberá realizar una adecuada validación de tipo y consistencia de los datos de entrada.
* La interfaz del sistema deberá ser amigable y fácil de entender y utilizar por el usuario. Recuerde que el usuario debe tener información suficiente para poder utilizar el sistema.
* Las opciones para el acceso a las funcionalidades del sistema deben estar agrupadas en menús y submenús, de forma lógica e intuitiva para el usuario.
* El proyecto se debe realizar individualmente o en equipos de máximo 2 estudiantes.
* Solo se reciben y revisan las entregas que se realicen por medio del Aula Virtual. Por lo tanto, verifique entregar la versión correcta y completa.
* No se recibirá ningún proyecto después de la fecha y hora establecidas.
* Solo uno de los dos miembros del equipo debe entregar el proyecto. Se debe incluir en un archivo de texto el nombre y cedula de los integrantes del equipo.
* Queda a criterio del profesor solicitar defensa oral del proyecto, dentro o fuera del horario de clase.
* Cualquier proyecto que no sea de elaboración propia y original de los estudiantes o haya sido copiado o adaptado de otro origen (plagio), de manera parcial o total, se calificará con nota 0 (cero) y se procederá de acuerdo con lo establecido al respecto en los reglamentos aplicables y vigentes de la universidad.

**Tabla de evaluación y calificación del proyecto:**

Para cada rubro especificado a continuación se verificará el cumplimiento y validación de todo lo especificado anteriormente en el documento.

Es necesario que se haga un correcto, adecuado y eficiente diseño e implementación de clases y relaciones entre clases, de acuerdo con el POO.

Como se mencionó anteriormente, el sistema debe permitir al usuario ingresar todos los datos requeridos para probar las funcionalidades.

|  |  |
| --- | --- |
| **Rubro** | **Puntos** |
| **Ingreso de Datos Generales**   * Ingreso de **sucursales** * Ingresos de **clientes** *(vincularlos a sucursales ya existentes)*   ***(2 ptos.)***   * Ingreso de **instructores** *(vincularlos a sucursales ya existes)*   ***(2 ptos.)*** | **4** |
| **Informe Clientes**   * Lista general de todos los clientes por sucursal específica. ***(3 ptos.)*** * Detalle de un cliente específico *(nombre completo, id, teléfono, correo, fecha de nacimiento, sexo, fecha de inscripción, sucursal a la que pertenecen y el instructor asignado)* ***(3 ptos.)*** * Lista de clientes por instructor específico ***(3 ptos.)*** | **9** |
| **Informe Instructores**   * Lista de instructores por sucursal ***(5 ptos.)*** * Lista de instructores por especialidad ***(5 ptos.)*** | **10** |
| **Historial de mediciones por cliente**   * El sistema permite ingresar el registro de medición a un cliente especifico *(el cliente debe existir previamente en el sistema, así como el instructor asignado).* ***(2 ptos.)*** * Hecho lo anterior se debe poder buscar un cliente y mostrar su historial de mediciones, organizadas por fecha, con la opción de mostrar el detalle de cualquiera de sus reportes de medición. ***(13 ptos.)*** | **15** |
| **Reporte IMC**   * Se debe mostrar por sucursal especifica, cuántos y cuáles clientes se ubican en cada uno de los rangos de contextura según el IMC. | **9** |
| **Rutinas**   * El sistema debe permitir ingresar ejercicios a la “batería de ejercicios”. ***(4 ptos.)*** * El sistema debe permitir la generación de una nueva rutina a un cliente especifico *(a partir de la “batería de ejercicios”).* Se debe poder buscar cualquier cliente de cualquier sucursal y mostrar la rutina generada. ***(14 ptos.)*** | **18** |
| **Matricula de Clases Grupales**   * El sistema debe permitir crear **clases grupales** por sucursal especifica. Debe permitir visualizar dichas **clases grupales**, con todo su detalle *(tipo de clase, instructor, horario, capacidad o cupo máximo, cupos disponibles, cantidad de matriculados etc).* ***(5 ptos.)***   *Nota: El instructor asignado a la clase debe existir previamente en el sistema y poseer la especialidad que concuerde con el tipo de clase grupal.*   * El sistema debe permitir matricular clientes en las **clases grupales** impartidas en las diferentes sucursales. Para verificar estos, se debe permitir visualizar la lista de clientes matriculados por clase especifica. ***(10 ptos.)*** *Nota: Las clases y los clientes deben existir previamente, se debe verificar que haya cupo antes de realizar la matricula.* * Se debe permitir visualizar las clases matriculadas por cliente especifico ***(10 ptos.)*** | **25** |
| Buen uso de memoria dinámica en objetos y colecciones | **5** |
| Interfaz clara, agradable e intuitiva. | **5** |
| **Total** | **100** |

1. El concepto **"datos quemados"** (también llamado *hardcoded data*) se refiere a cuando los valores o datos están **escritos directamente en el código fuente**, en vez de cargarse desde un archivo de configuración, base de datos, o entrada del usuario. [↑](#footnote-ref-1)