

PROGRAMACIÓN ORIENTADA A OBJETOS

Grupo 2

Práctica 2

Integrantes Equipo 4

Jacobo Ochoa Ramirez
Juan Manuel Rodríguez Sánchez
Sebastián Valdivieso Zapata
Juan Jose Ospina Erazo
Esteban Vásquez Pérez

TABLA DE CONTENIDO

1. Descripción general de la solución (análisis, diseño, e implementación).
2. Descripción del diseño estático del sistema en la especificación UML (Diagrama de clases y objetos del sistema).
3. Descripción de la implementación de características de programación orientada a objetos en el proyecto (indicando los lugares y el modo en que se implementaron).
4. Descripción de cada una de las 5 funcionalidades implementadas (incluye la descripción de la funcionalidad, que objetos intervienen en su implementación con un breve modelo de la secuencia del proceso, y por último, Incorporar una captura de pantalla con los resultados que presenta al usuario).
5. Manual de usuario con los elementos necesarios para poder evaluar el correcto funcionamiento del sistema (nombres, contraseñas,etc). En caso de que no se pueda valorar por esta causa alguna funcionalidad esta NO será tomada en cuenta para su evaluación.

DESCRIPCIÓN GENERAL DE LA SOLUCIÓN

Implementamos una aplicación para gestionar el funcionamiento de un Gimnasio. Utilizamos un modelo que permite generar rutinas, planes de alimentación y llevar el registro de todos los datos necesarios para esta labor, además de otras funcionalidades esenciales.

DESCRIPCIÓN DEL DISEÑO ESTÁTICO

Diagrama UML: <https://app.diagrams.net/#G14lkD7mva55gPEUkK7ttofEQOB6qXj5b3>

Nuestro diseño cuenta con 21 clases de las cuales una es una interfaz "Plan" y una es abstracta "Persona".

CLIENTE: El usuario principal de nuestra aplicación. Tiene un plan de alimentación y un plan de entrenamiento, además de un gimnasio que es el que más frecuenta, pero puede cambiar.

ENTRENADOR: Entrenan a clientes de nivel principiante, intermedio y avanzado. Tiene una disponibilidad semanal dentro de un diccionario cuyas llaves son el día y valores son las horas de trabajo.

MOVIMIENTO: Puede ser un calentamiento, cardio, una única repetición de un ejercicio. Estos tienen Dificultades asociadas.

EJERCICIO: Subclase de Movimiento. Tiene repeticiones, sets y otros atributos y Tiempo. Se trabaja un solo tipo Muscular por ejercicio.

RUTINA: Está compuesta por varios ejercicios.

MÁQUINA: Las máquinas donde se hacen los movimientos (Mancuernas, bandas, caminadora, barras, y libre si no hay un equipamiento específico).

GINNASIO: Está compuesto de Máquinas, Entrenadores y Clientes. Tienen sedes en distintas ciudades y horarios disponibles.

ALIMENTO: Es un componente de una comida (Tomate, Zanahoria, Pollo...).

COMIDA: Está compuesta por múltiples alimentos.

PLAN ALIMENTACIÓN: Está compuesto por múltiples comidas para todos los días de la semana.

PLAN EJERCICIO: Está compuesto por múltiples ejercicios para todos los días de la semana.

HISTORIAL: Sirve para guardar los planes de alimentación y los planes de ejercicio que ha seguido un cliente, además de los gimnasios que ha visitado y con qué frecuencia.

DESCRIPCIÓN DE CARACTERÍSTICAS DE POO IMPLEMENTADAS

- Clases abstractas y métodos abstractos:
 - Clase abstracta(Persona): las clases Cliente y Entrenador heredaron de esta, sus atributos son *protected* y tiene el método abstracto toString.
 - Método abstracto(Persona-toString())

- Método abstracto(Plan-toString())
- Método abstracto(Plan-crearPlanSemanal(ObjetivoCliente)): Lo utilizan las clases PlanEjercicio y PlanAlimentacion para la creación de objetos con elementos random bajo unos parámetros y para la clase PlanEjercicio es fundamental para la funcionalidad 3
- Interfaces:
 - Interfaz(Plan): PlanEjercicio y PlanAlimentacion heredan de esta. Tiene los métodos abstractos toString() y crearPlanSemanal()
- Herencia:
 - Herencia simple(Movimiento) le hereda a la clase Ejercicio, la cual utiliza múltiples veces *super()* y métodos get y set
 - Interfaz y herencia abstracta anteriormente mencionadas
- Ligadura dinámica:
 - En el método generarPlanEjercicio() le hace un set al atributo planEjercicio del Plan entregado por el método crearPlanSemanal(). Finalmente se pide el toString de planEjercicio que es un objeto de tipo Plan, por ligadura dinámica utiliza el toString de la clase PlanEjercicio.
- Atributos de clase y métodos de clase:
 - Atributos de clase(Cliente -> listaClientes): ArrayList con todos los clientes creados
 - Atributos de clase(Entrenador -> listaEntrenadores): ArrayList con todos los entrenadores creados
 - Atributos de clase(Movimiento -> listaMovimiento): ArrayList con todos los movimientos creados
 - Atributos de clase(Gimnasios -> listaGimnasios): ArrayList con todos los gimnasios creados
 - Atributos de clase(Rutina -> NUMEJERCICIOS): Número máximo de ejercicios por rutina
 - Método de clase(Movimiento ->)
 - Método de clase(Movimiento -> filtrarMovimientosPorTipo)
 - Método de clase(Movimiento -> filtrarMovimientosPorDificultad)
 - Método de clase(Ejercicio -> generarEjercicio)
 - Método de clase(Rutina -> generarRutina)
 - Método de clase(Main -> reservarGimnasio)
 - Método de clase(Main -> recomendarPlanAlimentacion)
 - Método de clase(Main -> recomendarPlanEjercicio)
 - Método de clase(Main -> recomendarEntrenadores)

- Constante:
 - PlanAlimentación -> NUMCOMIDAS: número de comidas en un día (3)
 - Rutina -> NUM EJERCICIOS: número máximo de ejercicios que se realizan en una rutina (7)
- Encapsulamiento:
 - private: La mayoría de los atributos de las clases están en Private
 - protected: Los atributos de las clases Movimiento y Persona están en Protected
 - public: algunos atributos de clase están public, algunos atributos de la clase Alimento están public, entre otros
- Sobrecarga de métodos:
 - crearPlanSemanal(ObjetivoCliente, NivelCliente) es sobrecarga de crearPlanSemanal(ObjetivoCliente).
- Sobrecarga de constructores:
 - Movimiento (String, String, String, TipoEjercicio, Maquina, NivelCliente, boolean) tiene como sobrecarga Movimiento(String, TipoEjercicio, Maquina, NivelCliente) , Movimiento(String, String, String, TipoEjercicio, Maquina, NivelCliente) y Movimiento(String, String, String, TipoEjercicio, Maquina) - Todos estos constructores se crearon con el fin de implementar el método generarEjercicio() sin que se crearan objetos duplicados
 - Rutina (String) tiene como sobrecarga Rutina(String, ArrayList<Ejercicio>, TipoEjercicio) y Rutina(ArrayList<Ejercicio>, TipoEjercicio) para la creación de objetos bajo distintas circunstancias.
- Manejo de this() y this.
 - Uso de this()(Movimiento): El constructor Movimiento(String nombre, TipoEjercicio , Maquina, NivelCliente) utilizar this para usar el constructor Movimiento (String, String, String, TipoEjercicio, Maquina, NivelCliente, boolean)
 - Uso de this()(Cliente): El constructor Cliente() utiliza this para usar el constructor Cliente(String, Gimnasio, int, double, double, String, int, Entrenador, PreferenciaAlimenticia, NivelCliente nivelCliente, ObjetivoCliente objetivoCliente)
 - Uso de this: Se utiliza múltiples veces en los constructores y métodos get y set de las clases, a parte de estos se utiliza para métodos de clases como Historial, Entrenador, Gimnasio, entre otros.
- Enum

- Hay siete clases enum, un ejemplo de su uso es en el método en PlanEjercicio crearPlanSemanal() para definir los días de la semana y como los tipos de ejercicios que es posible realizar.

DESCRIPCIÓN DE LAS FUNCIONALIDADES

1. Recomendar Plan Ejercicio:

Esta funcionalidad recomienda un plan con rutinas para realizar en el transcurso de la semana. Utiliza información del usuario como su objetivo y el nivel de experiencia para crear una serie de rutinas adecuadas.

En esta funcionalidad interactúan las clases *Cliente*, *PlanEjercicio*, *Rutina*, *Movimiento* y *Ejercicio*. El usuario hace click derecho en el botón Recomendar Plan Ejercicio. Ingresar su nombre, identificación y el nombre que le desea dar al plan de entrenamiento.

El método generarPlanEjercicio() utiliza la información ya establecida del Cliente, como el objetivo y el nivel de experiencia. Este llama al método crearPlanSemanal() de la clase PlanEjercicio y le entrega la información anteriormente mencionada. Para cada día de la semana se va a determinar un tipo de ejercicio predominante (2 días se realizará empujar, halar y pierna, y el día restante es de cardio). Estos tipos se van a repartir de manera aleatoria a través de la semana.

Para cada día se va a generar una rutina con el método generarRutina(TipoEjercicio, ObjetivoCliente, NivelCliente), que va a filtrar una lista estática con todos los objetos de tipo movimiento que se han creado. Lo hace por medio de los métodos filtrarMovimientosPorTipo() y filtrarMovimientoPorDificultad() para obtener solo los objetos que cumplen con los parámetros de tipo y dificultad. Los movimientos los convierte en ejercicios con el método generarEjercicio() que establece una serie de repeticiones y series basadas en el objetivo brindado. Finalmente agrega un ejercicio de calentamiento, hasta 5 del tipo seleccionado y uno de estiramiento.

Con las rutinas de los 7 días se crea un objeto de PlanEjercicio, que es entregado como objeto de tipo Plan a generarPlanSemanal(). Este plan se define como el plan de ejercicio del cliente y después se retorna el toString de este.

Archivo	Procesos y Consultas	Ayuda
Recomendar Plan Ejercicio	Añadir Alimento	
Buscar Alimento por Calorías, Proteínas y Alergenos	Buscar Gimnasios por Ciudad	
Consultar maquinas disponibles		

Gymbro - Usuario

Archivo	Procesos y Consultas	Ayuda
---------	----------------------	-------

Recomendar Plan Ejercicio

Recibe un Plan de Ejercicio que recomiendan nuestros mejores entrenadores.

Información	Datos
Nombre usuario	Oswaldo
Identificación	43
Nombre para el plan	Buff

Aceptar Borrar

2. Añadir alimento

Esta funcionalidad añade un conjunto de alimentos con los que el usuario puede crear una comida que se ajuste a su plan de alimentación, basándose en sus necesidades calóricas, proteicas, de carbohidratos, grasas y sus alérgenos, si es que tiene.

En esta funcionalidad interactúan las clases *Cliente*, *Alimento*, *Comida* y *PlanAlimentacion*.

El usuario hace click en el botón Añadir Alimento, y se abrirá una ventana en la que el usuario debe añadir un alimento basándose en sus calorías, proteínas, carbohidratos, grasas, y alérgeno. El programa guardará los alimentos ingresados por el usuario en la base de datos y mostrará una pantalla confirmando el guardado.

Archivo

Procesos y Consultas

Ayuda

Añadir Alimento

Añada un alimento con el cual podrá preparar una comida que se ajuste a su Plan de Alimentación y su objetivo.

Requerimientos	Datos
Nombre	Mantequilla de maní
Calorías	588
Proteínas	25
Carbohidratos	20
Grasas	10
Alérgeno	FRUTOS SECOS

Aceptar

Borrar

Archivo

Procesos y Consultas

Ayuda

¡Alimento añadido exitosamente!

MANTEQUILLA DE MANÍ

Calorías: 588.0
Proteínas: 25.0
Carbohidratos: 20.0
Grasas: 10.0
Alérgeno: FRUTOS SECOS

3. Buscar alimento

Esta funcionalidad crea un conjunto de alimentos que el usuario puede elegir para crear una comida basándose en sus requerimientos nutricionales.

El usuario debe ingresar un máximo de calorías, un mínimo de proteínas y algún alérgeno que tenga. El programa elegirá los alimentos que cumplan con estas características, y descartará los demás. Luego mostrará una pantalla con los alimentos que cumplen con las características.

The screenshot shows a web application window titled "Gymbro - Usuario". It has a navigation bar with three buttons: "Archivo", "Procesos y Consultas", and "Ayuda". Below the navigation bar, there is a section titled "Buscar Alimentos por Calorías, Proteínas y Alergeno" with a subtitle "Escoge entre una lista de alimentos que cumplen tus requerimientos calóricos y proteínicos, descarta los alimentos a los que seas alérgico/a." Below this, there is a form with two columns: "Requerimientos" and "Datos". The "Requerimientos" column has three labels: "Máximo de calorías", "Mínimo de proteínas", and "Alergeno". The "Datos" column has three input fields with the values "500", "12", and "Frutos secos" respectively. Below the form, there are two buttons: "Aceptar" and "Borrar".

Requerimientos	Datos
Máximo de calorías	500
Mínimo de proteínas	12
Alergeno	Frutos secos

The screenshot shows the same web application window as above, but now it displays a list of food items that match the search criteria. The list is centered on the screen and contains the following items: "Huevo", "Pollo", "Carne de res", "Pescado", "Atun", "Camarones", "Queso", and "Avena".

- Huevo
- Pollo
- Carne de res
- Pescado
- Atun
- Camarones
- Queso
- Avena

4. Buscar gimnasio por ciudad

Esta funcionalidad busca una sede para un gimnasio basándose en el nombre del mismo y en la ciudad en la que está ubicado el usuario. El usuario ingresará el nombre del gimnasio y la ciudad en la que lo busca, y el programa le dirá las sedes disponibles.

The screenshot shows a web application window titled 'Gymbro - Usuario'. It has a navigation bar with three buttons: 'Archivo', 'Procesos y Consultas', and 'Ayuda'. Below the navigation bar is a section titled 'Buscar Gimnasio' with a subtitle 'Ingrese la información requerida para buscar un gimnasio'. This section contains two tabs: 'Requerimientos' and 'Datos'. Under 'Requerimientos', there is a label 'Nombre de gimnasio' followed by a text input field containing 'gymbro'. Under 'Datos', there is a label 'Ciudad' followed by a dropdown menu showing 'Medellín'. At the bottom of the form are two buttons: 'Aceptar' and 'Borrar'.

The screenshot shows the same web application window. The 'Buscar Gimnasio' section now displays the search results: 'Prado', 'Robledo', and 'Laureles' listed vertically.

5. Consultar máquinas disponibles

Esta funcionalidad consulta las máquinas disponibles en cada una de las sedes de los gimnasios. El usuario deberá marcar la casilla que le interese, y el programa le devolverá las máquinas disponibles en las sedes elegidas.

The screenshot shows the same web application window. Below the navigation bar is a section titled 'Marca los campos que necesites:'. This section contains two groups of checkboxes. The first group is labeled 'Nombre' and includes three options: 'Smartfit' (checked), 'Gymbro' (checked), and 'Bodytech' (unchecked). The second group is labeled 'Ciudad' and includes two options: 'Bogotá' (checked) and 'Medellín' (unchecked). At the bottom of the form is a button labeled 'Aceptar'.



Manual de usuario

Preámbulo: Al abrir el programa se abrirá una ventana, en la que principalmente se dará una bienvenida al programa, junto con el objetivo del mismo. También se verá una breve hoja de vida de los desarrolladores, con su nombre, fecha de nacimiento y gustos, esta hoja de vida cambiará de desarrollador al pasar el cursor por encima de las fotos. Finalmente habrá un botón para ingresar al programa.



Al dar click en el botón de ingresar, se abrirá otra pantalla con cinco botones principales, que son las cinco funcionalidades del programa, definidas como: *Recomendar Plan Ejercicio*, *Añadir Alimento*, *Buscar Alimento por Calorías*, *Proteínas y Alérgenos*, *Buscar Gimnasios por Ciudad* y *Consultar máquinas disponibles*. El usuario puede dar click en cada una de ellas para realizar una acción


diferente. Adicionalmente, hay tres botones más pequeños en la parte superior de la pantalla, que están definidos como *Archivo*, *Procesos y Consultas* y *Ayuda*. El uso de estos botones será descrito a continuación.


Archivo	Procesos y Consultas	Ayuda
Recomendar Plan Ejercicio		Añadir Alimento
Buscar Alimento por Calorías, Proteínas y Alergenos	Buscar Gimnasios por Ciudad	
Consultar maquinas disponibles		

Archivo

Al dar click en el botón de Archivo, se abrirá una pantalla con dos botones principales, *Aplicación* y *Salir*.

Si se da click en el botón de Aplicación, se abrirá una ventana emergente que dará una breve descripción de la aplicación. Si se da click en el botón salir, la aplicación se cerrará.

 Aplicación ×



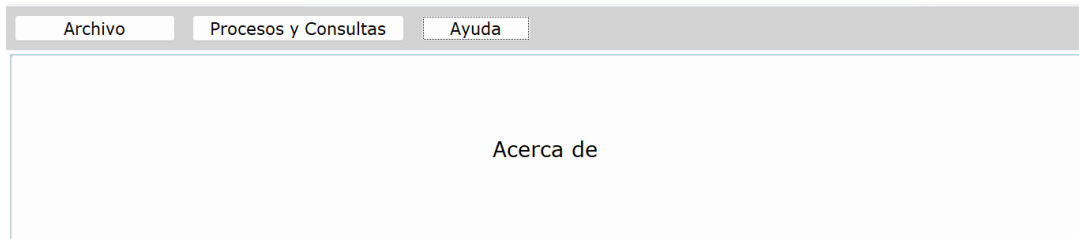
Gymbro consiste en una aplicación que tiene como fin gestionar el funcionamiento de un gimnasio.

Con esta podrás realizar procesos y consultas que llevarán tu disciplina y consistencia con tus planes a otro nivel, trabajando de la mano de los mejores entrenadores y sus recomendaciones.

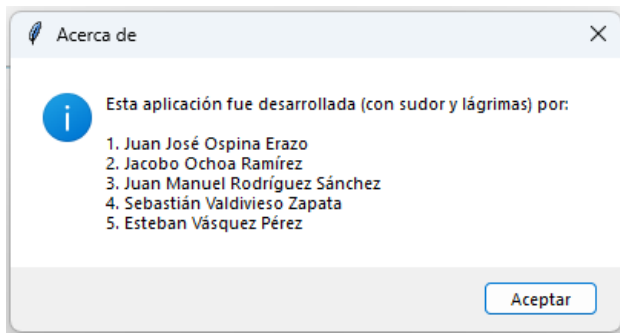
Aceptar

Ayuda

Al dar click en el botón Ayuda, se abrirá una pantalla con un único botón: *Acerca de*.



Al dar click en el botón Acerca de, se abrirá una ventana emergente con el nombre de los desarrolladores.



1. Recomendar Plan Ejercicio:

La opción de recomendar un plan de ejercicio requiere que le digas tu nombre, identificación y el nombre que le quieres dar a tu plan de ejercicio

Recomendar Plan Ejercicio

Recibe un Plan de Ejercicio que recomiendan nuestros mejores entrenadores.

Información	Datos
Nombre usuario	Duvan
Identificación	1
Nombre para el plan	Musculoso

Aceptar Borrar

Si utilizas una identificación mayor que 50, el programa va a generar un error, porque el usuario no se va a encontrar en la base de datos.

Duvan, basándonos en tu nivel de habilidad y tu objetivo, hemos creado el plan Musculoso para ti

-----TU PLAN DE EJERCICIO-----

LUNES: HALAR
Caminata lenta = Caminadora ---> 11x1
Oblicuos en polea alta = Polea ---> 10x2
Extension sentada de espalda = Remo ---> 12x2
Remo con agarra interno = Remo ---> 10x2
Flexion lateral de tronco en polea baja = Polea ---> 13x2
Pullover con polea alta = Polea ---> 10x2
Tibial anterior = Rodillo ---> 12x2

MARTES: EMPUJAR
Caminata lenta = Caminadora ---> 13x1
Remo a ambas manos = Remo ---> 12x2
Press plano de pecho = Barra ---> 13x3
Press inclinado de hombro = Barra ---> 13x3
Press inclinado de pecho = Barra ---> 11x2
Press de banca con mancuernas = Banco de abdominales ---> 9x2
Vasto lateral = Rodillo ---> 9x3

MIÉRCOLES: EMPUJAR
Caminata lenta = Caminadora ---> 11x1
Remo a ambas manos = Remo ---> 11x3
Press de banca con mancuernas = Banco de abdominales ---> 13x2
Press inclinado de hombro = Barra ---> 10x3
Press inclinado de pecho = Barra ---> 13x2
Press plano de pecho = Barra ---> 11x2
Lumbares = Banco de abdominales ---> 11x3

JUEVES: CARDIO
Caminata lenta = Caminadora ---> 10x1
Prueba de resistencia en bicicleta fija = Bicicleta ---> 11x2
Abdominales = Banco de abdominales ---> 9x3
Correr = Caminadora ---> 13x2
Vasto interno = Rodillo ---> 12x3

Los planes de ejercicios están diseñados para adaptarse a tus habilidades y objetivo de entrenamiento, cada uno es único. Los días de trabajo pueden ser de halar, empujar, para piernas o de cardio. Todas las rutinas tienen hasta 7 ejercicios e incluyen uno de calentamiento al inicio de la sesión y uno de estiramiento al final.

2. Añadir Alimento:

Para usar la funcionalidad Añadir Alimento, se debe dar click en el botón *Añadir Alimento* en la pantalla principal de Procesos y Consultas que contiene las funcionalidades. Luego de

esto el programa abrirá una ventana donde el usuario debe ingresar los alimentos que desee, incluyendo su nombre, calorías, proteínas, carbohidratos, grasas, y si tiene o no algún alergeno. Luego de esto, el usuario debe dar click en el botón *Aceptar* para guardar el alimento, o en *Borrar* para descartarlo. En caso de que se dé click en *Aceptar*, el programa guardará el alimento en su base de datos.

Añadir Alimento

Añada un alimento con el cual podrá preparar una comida que se ajuste a su Plan de Alimentación y su objetivo.

Requerimientos	Datos
Nombre	Mantequilla de maní
Calorías	524
Proteínas	25
Carbohidratos	20
Grasas	17
Alergeno	frutoseco

Aceptar

Borrar

¡Alimento añadido exitosamente!

MANTEQUILLA DE MANÍ
Calorías: 524.0
Proteínas: 25.0
Carbohidratos: 20.0
Grasas: 17.0
Alergeno: FRUTOSECO

3. Buscar Alimento por Calorías, Proteínas y Alergeno

Para usar la funcionalidad de *Buscar Alimento por Calorías, Proteínas y Alergeno*, el usuario deberá ingresar un máximo de calorías, un mínimo de proteínas, y si tiene o no algún alergeno, si es que le aplica. Después de hacer esto, el usuario debe dar click en *Aceptar*, y el programa le devolverá una pantalla con los alimentos que puede usar para crear una comida.

Buscar Alimentos por Calorías, Proteínas y Alergeno

Escoge entre una lista de alimentos que cumplen tus requerimientos calóricos y proteínicos, descarta los alimentos a los que seas alérgico/a.

Requerimientos

Máximo de calorías

Mínimo de proteínas

Alergeno

Datos

Aceptar

Borrar

Buscar Alimentos por Calorías, Proteínas y Alergeno

Escoge entre una lista de alimentos que cumplen tus requerimientos calóricos y proteínicos, descarta los alimentos a los que seas alérgico/a.

Requerimientos

Máximo de calorías

Mínimo de proteínas

Alergeno

Datos

450

20

lactosa

Aceptar

Borrar

Pollo
Carne de res
Pescado
Atun
Queso

4. Buscar Gimnasios por Ciudad

El usuario digita el nombre del gimnasio al que desea asistir, en la base de datos se encuentran disponibles las opciones: Smartfit, Gymbro y Bodytech.

Posteriormente se elige la ciudad (Medellin o Bogota) y se presiona el botón *Aceptar*.

La aplicación va a mostrar la lista de gimnasios disponibles.

Buscar Gimnasio

Ingrese el nombre del gimnasio que desea consultar y elija su respectiva sede.

Requerimientos

Nombre de gimnasio

Datos

Smartfit

Ciudad

Bogota

Aceptar

Borrar

Sedes Smartfit Bogota

La Candelaria

20 de Julio

La Merced

5. Buscar Máquinas por Sede:

El usuario deberá elegir la funcionalidad dando click en el botón correspondiente. Al hacer esto se abrirá una pantalla donde aparecerán dos filtros, uno correspondiente al gimnasio que se desee usar (según el nombre), y otro correspondiente a la ciudad (Bogotá o Medellín). El usuario deberá marcar las casillas que le interesa y luego dar click en aceptar. Al hacer esto el programa le devolverá una pantalla en la que aparecerán las sedes que son acordes al interés del usuario, y las máquinas disponibles en esas sedes.

Marca los campos que necesites:

Nombre

☒ Smartfit

☐ Gymbro

☒ Bodytech

Ciudad

☒ Bogotá

☐ Medellín

Aceptar

Máquinas Disponibles

Sede Smartfit en La Candelaria

Rodillo

Remo

Aductores

Sede Smartfit en 20 de Julio

Caminadora

Banco de abdominales

Elíptica

Sede Smartfit en La Merced

Caminadora

Banco de abdominales

Elíptica

Sede Bodytech en Santafe

Rodillo

Elíptica

Caminadora