

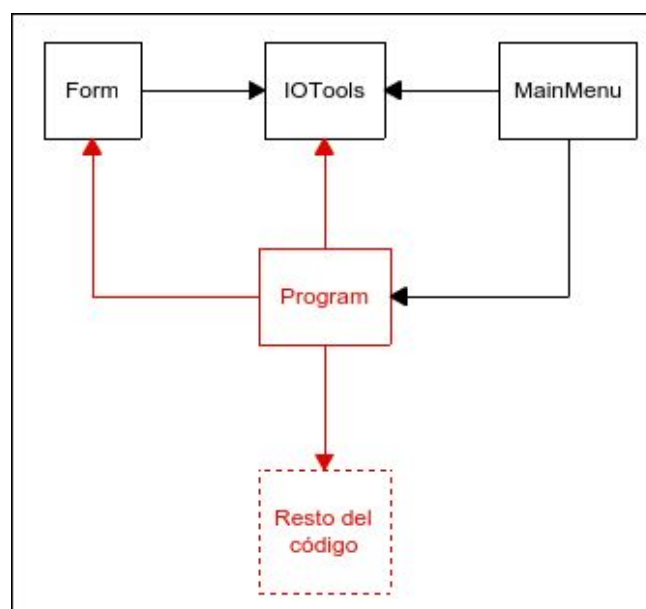
Especificaciones sobre el laboratorio

1.Introducción

El estudiante dispondrá de 3 archivos:

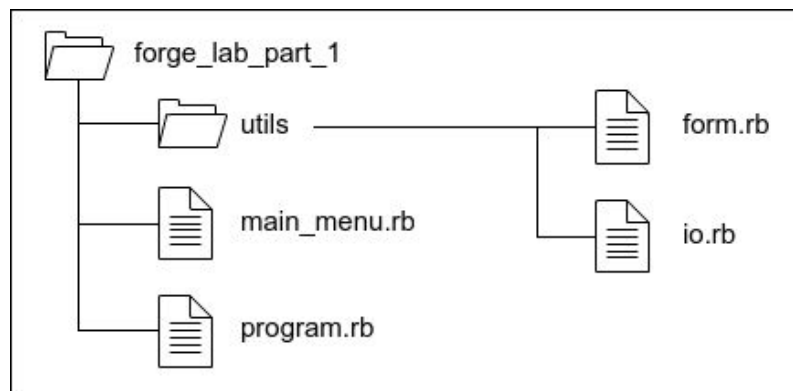
- main_menu.rb
- form.rb
- io.rb

Diagrama de comunicación entre archivos:



Las partes **rojas** en el diagrama, es lo que el estudiante debe realizar.

Los archivos tienen que estar ubicados en un directorio, de la siguiente manera:



A continuación, se detallarán las clases dadas y lo que se espera de la clase Program.

2. Clase MainMenu (main_menu.rb)

Esta clase es la que será ejecutada para poder ejecutar el laboratorio.
MainMenu se comunica con las clases IO y Program.

MainMenu se encarga de desplegar las opciones correspondientes para que el usuario pueda elegir una.

3. Clase Form (form.rb)

Form es una clase que permite solicitar información al usuario mediante la consola.
Cada vez que se quiera pedir información al usuario, se deberá crear una nueva instancia de Form, con los parámetros adecuados.

Especificación de los métodos:

- ***initialize(title = "", fields = {})***

Al crear una nueva instancia de Form, se podrá especificar un título y un conjuntos de datos.

Luego de almacenar la información dada, se imprimirá el título dado.

Tipos de los parámetros:

- title: String
- fields: Hash

- ***select_from_list(selection_text, list)***

Este método permite seleccionar un elemento de una lista dada.

El usuario deberá ingresar la posición del elemento en la lista (comenzando en 1).

Si la posición ingresada no es válida, se mostrará un error, pidiendo que ingrese nuevamente una posición.

- ***ask_for(*attrs)***

Este métodos permite pedirle al usuario los datos que se desean.

Recibe como parámetros una lista de los datos a solicitar.

- ***get_data***

Este métodos permite obtener los valores que el usuario ingresó para cada dato solicitado.

Ejemplo de uso:

```

form = Form.new('Ingresa la información del usuario',
               first_name: 'Nombre', last_name: 'Apellido')

form.ask_for(:first_name, :last_name)

colors = ['Rojo', 'Verde', 'Azul', 'Amarillo']
color = form.select_from_list('Ingresa el color favorito: ',
                              colors)

puts 'Datos del usuario:'
puts "Nombre: #{form.get_data[0]}"
puts "Apellido: #{form.get_data[1]}"
puts "Color favorito: #{color}"

```

3.Clase IOTools (io.rb)

Esta clase permite interactuar con la consola.

Tiene métodos para solicitar información en la consola, mostrar errores en la consola, desplegar un menú y desplegar una lista.

4.Clase Program (program.rb)

La clase Program será que se utilizará MainMenu y se deberá comunicar con el resto de las clases que haga el estudiante.

Dicha clase deberá contar con los siguientes métodos:

- ***initialize***

Al crearse una instancia de Program, deberá:

- solicitar el nombre para el campeonato
- solicitar la cantidad de jugadores por equipo (5, 7 u 11)
- crear un nuevo campeonato con los datos solicitados

Nota: el método no recibe parámetros

- ***championship_can_be_played***

Consulta si el campeonato puede comenzar.

Si no existen equipos en el sistema, deberá retornar el siguiente mensaje de error:

'No hay equipos para disputar el campeonato'.

Si la cantidad de equipos no es par, deberá retornar el siguiente mensaje de error:

'La cantidad de equipos disponibles debe ser par'.

Si todos los equipos no tienen al menos la cantidad especificada de jugadores, deberá retornar el siguiente mensaje de error:

'Hay equipos que tienen menos de la cantidad de jugadores requeridos'.

Nota: el método no recibe parámetros

- ***championship_start***

Si el campeonato no comenzó aún, deberá comenzar y generar el fixture.

Nota: el método no recibe parámetros

- ***championship_name***

Retorna el nombre del campeonato.

Nota: el método no recibe parámetros

- ***add_team***

Permitirá crear un nuevo equipo al campeonato a partir de información ingresada por la consola.

Nota: el método no recibe parámetros

- ***add_player***

Permitirá agregar un nuevo jugador al campeonato a partir de información ingresada por la consola.

Nota: el método no recibe parámetros

- ***add_player_to_team***

Muestra la lista de jugadores y equipos del campeonato, y permite agregar un jugador a un equipo.

Nota: el método no recibe parámetros

- ***display_players***

Muestra todos los jugadores del campeonato.

Cada jugador deberá ser mostrado de la siguiente manera:

nombre_del_jugador (número_de_cédula)

Nota: el método no recibe parámetros

- ***display_teams***

Muestra todos los equipos del campeonato.

Cada equipo deberá ser mostrado de la siguiente manera:

nombre_del_equipo (cantidad_de_jugadores jugadores)

Nota: el método no recibe parámetros

- ***next_match***

Muestra el siguiente partido y permite ingresar el resultado del mismo, dejando finalizado el partido.

Nota: el método no recibe parámetros

- ***display_fixture***

Muestra el fixture.

Visualización:

nombre_equipo_1 vs nombre_equipo_2
nombre_equipo_1 vs nombre_equipo_2
nombre_equipo_1 vs nombre_equipo_2
nombre_equipo_1 vs nombre_equipo_2
...

Nota: el método no recibe parámetros

- ***display_team_players***

Muestra los jugadores de un equipo.

El usuario deberá poder elegir el equipo desde una lista de los equipos del campeonato.

Los jugadores serán mostrados igual que en ***display_players***.

Nota: el método no recibe parámetros

- ***print_table***

Deberá mostrar a tabla de posiciones:

Visualización:

Equipo	PJ	PG	PE	PP	Puntos
<i>nombre_equipo</i>	0	0	0	0	0
<i>nombre_equipo</i>	0	0	0	0	0
<i>nombre_equipo</i>	0	0	0	0	0
<i>nombre_equipo</i>	0	0	0	0	0
<i>nombre_equipo</i>	0	0	0	0	0
<i>nombre_equipo</i>	0	0	0	0	0

Nota: el método no recibe parámetros
