

Universidad Autónoma de Baja California

PRÁCTICA 8

RELACIONES ENTRE CLASES



Autor:
Luis Eduardo Galindo Amaya
1274895

19 de abril de 2022

Fotos

```

<du/Escritorio/P00-2022/ Python 3.8.10 (default, Mar 15 2022, 12:22:00)
+ --
+ imports/
+ Práctica 1/
+ Práctica 2/
+ Práctica 3/
+ Práctica 4/
+ Práctica 5/
+ Práctica 6/
+ Práctica 7/
+ Práctica 8/
+ img/
+ src/
+ _jycache_/
+ Equipo.py
+ Jugador.py
+ Torneo.py
+ main.py
+ diagrams.org
+ template.sbl
+ template.org
+ template.pdf
+ template.synctex-gs
+ template.tex
+ README.org

Python 3.8.10 (default, Mar 15 2022, 12:22:00)
(GCC 9.4.0) on Linux
Type "help", "copyright", "credits" or "license()" for more information
>>> python.el: native completion setup loaded
>>> Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "/home/galindo/Escritorio/P00-2022/práctica 8/src/main.py", line 11
    me_5, in <module>
      torneoA = Torneo("Copa extranjera", "Nacional", 6)
TypeError: __init__() takes 3 positional arguments but 4 were given
>>>
>>>
>>> torneoA
<Torneo.Torneo object at 0x7f3704632d30>
>>> torneoA.agregar_equipo(EquipoA)
>>>

from Equipo import *
from Jugador import *
from Torneo import *

torneoA = Torneo("Copa extranjera", "Nacional")
torneoB = Torneo("Copa local", "Mexico")

EquipoA = Equipo("A", "1", "pepe")
EquipoB = Equipo("B", "1", "pepe2")
EquipoC = Equipo("C", "1", "pepe3")
EquipoD = Equipo("D", "1", "pepe4")

A = Jugador(1, "Pepe2", 1.50)
B = Jugador(2, "Pepe3", 1.70)
C = Jugador(3, "Pepe2", 1.80)
D = Jugador(4, "Pepe3", 1.50)
E = Jugador(5, "Pepe2", 1.30)
F = Jugador(6, "Pepe3", 1.40)
G = Jugador(7, "Pepe2", 1.50)
H = Jugador(8, "Pepe3", 1.60)

A.puntos_anoatados = 3
B.puntos_anoatados = 10
C.puntos_anoatados = 4
D.puntos_anoatados = 20
E.puntos_anoatados = 7
F.puntos_anoatados = 0
G.puntos_anoatados = 2
H.puntos_anoatados = 10

EquipoA.agregar_jugador(A)
EquipoB.agregar_jugador(B)
EquipoC.agregar_jugador(C)
EquipoD.agregar_jugador(D)
EquipoA.agregar_jugador(E)
EquipoB.agregar_jugador(F)
EquipoC.agregar_jugador(G)
EquipoD.agregar_jugador(H)

torneoA.agregar_equipo(EquipoA)

```

Código

Programa 1: main.py

```

1 class Equipo:
2
3     def __init__(self, nombre, divisi n ,
4         entrenador):
5         self.nombre = nombre
6         self.divisi n = divisi n
7         self.entrenador = entrenador
8         self.torneos_ganados = 0
9         self.torneos_perdidos = 0
10        self.jugadores = []
11
12    def agregar_jugador(self, jugador):
13        jugador.numero_uniforme = self.
14            jugadores_registrados()
15        self.jugadores.append(jugador)
16
17    def jugadores_registrados(self):
18        return len(self.jugadores)

```

```
18     def goles_totales(self):
19         return sum([x.puntos_ anotados for x in
                     self.jugadores])
```

Programa 2: Equipo.py

```
1  class Equipo:
2
3      def __init__(self, nombre, divisi n ,
4                  entrenador):
5          self.nombre = nombre
6          self.divisi n = divisi n
7          self.entrenador = entrenador
8          self.torneos_ganados = 0
9          self.torneos_perdidos = 0
10         self.jugadores = []
11
12     def agregar_jugador(self, jugador):
13         jugador.numero_uniforme = self.
14             jugadores_registrados()
15         self.jugadores.append(jugador)
16
17     def jugadores_registrados(self):
18         return len(self.jugadores)
19
20     def goles_totales(self):
21         return sum([x.puntos_ anotados for x in
22                     self.jugadores])
```

Programa 3: Jugador.py

```
1  class Jugador():
2      def __init__(self, numero_uniforme, nombre,
3                  estatura):
4          self.numero_uniforme = numero_uniforme
5          self.nombre = nombre
6          self.puntos_ anotados = 0
7          self.posici n = 0
8          self.estatura = 0
```

Programa 4: Torneo.py

```

1 class Torneo:
2     def __init__(self, nombre, regi n):
3         self.nombre = nombre
4         self.regi n = regi n
5         self.equipos_participantes = []
6         self.partidos_jugados = 0
7         self.partidos_pendientes = 0
8
9     def agregar_equipo(self, equipo):
10        self.equipos_participantes.append(equipo)

```

UML

