



# PROYECTO EURO 2024

**Andres Arrrieta**

```
import requests
import json
import urllib.request
import random
import string
import atexit
import os
import matplotlib.pyplot as plt
from bokeh.plotting import figure, show
```

Inclusion de librerías necesarias para la ejecución del proyecto, para que pueda correr el proyecto tuve que instalar previamente el request pip y bokeh pip

```
# Clase "Cliente", se guardan los datos capturados desde la API
```

```
class Cliente:
```

```
    def __init__(self, nombre, cedula, edad, tipo_entrada, nombre_estadio):
        self.nombre = nombre
        self.cedula = cedula
        self.edad = edad
        self.tipo_entrada = tipo_entrada
        self.nombre_estadio = nombre_estadio
        self.id_boleto = self.generar_id_boleto() # Se genera un id de boleto aleatorio

    def generar_id_boleto(self): # Funcion para generar un id aleatorio para el boleto
        caracteres = string.ascii_letters + string.digits
        id_boleto = ''.join(random.choice(caracteres) for _ in range(8))
        return id_boleto
```

```
# Clase "Equipo", se guardan los datos capturados desde la API
```

```
class Equipo:
```

```
    def __init__(self, id, code, name, group):
        self.id = id
        self.code = code
        self.name = name
        self.group = group
```

```
# Clase "Stadium", se guardan los datos capturados desde la API
```

```
class Stadium:
```

```
    def __init__(self, id, name, city, capacity_general, capacity_vip, restaurants):
        self.id = id
        self.name = name
        self.city = city
        self.capacity_general = capacity_general
```

# Clases

Se genera un id de boleto aleatorio cada vez que se crea un cliente

Funcion para generar un id aleatorio para el boleto

se guardan los datos capturados desde la API

# Clase "Restaurant", se guardan los datos capturados desde la API

```
class Restaurant:
```

```
    def __init__(self, name, products):
        self.name = name
        self.products = products
```

# Clase "Product", se guardan los datos capturados desde la API

```
class Product:
```

```
    def __init__(self, name, quantity, price, adicional, stock):
        self.name = name
        self.quantity = quantity
        self.price = float(price) + float(price)*0.16
        self.adicional = adicional
        self.stock = stock
```

# Clase "Match", se guardan los datos capturados desde la API

```
class Match:
```

```
    def __init__(self, match_data, equipos, estadios):
        self.id = match_data['id']
        self.number = match_data['number']
        self.home = self.get_equipo_by_id(match_data['home']['id'], equipos)
        self.away = self.get_equipo_by_id(match_data['away']['id'], equipos)
        self.date = match_data['date']
        self.group = match_data['group']
        self.estadio = self.get_estadio_by_id(match_data['stadium_id'], estadios)
```

```
    def get_equipo_by_id(self, equipo_id, equipos):
        for equipo in equipos:
            if equipo.id == equipo_id:
                return equipo
```

```
    def get_equipo_by_id(self, equipo_id, equipos):
        for equipo in equipos:
            if equipo.id == equipo_id:
                return equipo
        return None
```

```
    def get_estadio_by_id(self, estadio_id, estadios):
        for estadio in estadios:
            if estadio.id == estadio_id:
                return estadio
        return None
```

```
# Funcion que carga todos los datos, hace tres llamadas a la API y captura todos los datos necesarios
def cargar_datos():
    # Cargar los equipos desde la API
    url_equipos = "https://raw.githubusercontent.com/Algoritmos-y-Programacion/api-proyecto/main/teams.json"
    equipos = cargar_equipos(url_equipos)

    # Cargar los estadios y restaurantes desde la API
    url_estadios = "https://raw.githubusercontent.com/Algoritmos-y-Programacion/api-proyecto/main/stadiums.json"
    estadios = cargar_estadios(url_estadios)

    # Cargar los partidos desde la API
    url_partidos = "https://raw.githubusercontent.com/Algoritmos-y-Programacion/api-proyecto/main/matches.json"
    partidos = cargar_partidos(url_partidos, equipos, estadios)

    return equipos, estadios, partidos
```

# Funcion para cargar los datos de la API

**A partir de ahora el código se centra en la creación de muchas funciones para que el programa funcione**

**Entonces mostrando las bases de este proyecto se dará una serie de pasos a seguir para que pueda disfrutar lo máximo de este proyecto**

# 1. Familiarizate con todas las opciones que te ofrece el menu principal

```
--> Euro 2024 - Venta de Tickets <--
```

```
Menu:
```

1. Búsqueda de partidos.
2. Venta de entradas.
3. Validar boleto.
4. Buscar productos.
5. Comprar productos.
5. Mostrar estadísticas.
7. Salir.

## 2.Create un Usuario para comprar tu boleto al partido que deseas

```
Elija una opcion: 2
Bienvenido al sistema de venta de entradas
Nombre del cliente: Andres Arrieta
Cédula del cliente: 30496624
Edad del cliente: 19
----- Partidos Disponibles -----
```

3.Previamente puedes ver todos los partidos disponibles pero estos igual se te mostraran cuando quieras realizar la compra de los mismo

```
Elija una opcion: 1
Elija el filtro (pais, estadio, fecha): pais
Nombre del pais: England
Partidos de England:
Equipo local: Serbia vs Equipo visitante: England
Equipo local: Denmark vs Equipo visitante: England
Equipo local: England vs Equipo visitante: Slovenia
```



**4. Eliges el ID del partido que quieres comprar como tambien el tipo de entrada que desees tener (Recuerda que la entrada VIP te permite comprar productos en el restaurante)**

-----  
ID del partido al que desea comprar: 1cc60753-36e3-4a3e-bbdd-cd482220588a  
Tipo de entrada que desea comprar (General/VIP): VIP



[illegible]

Asiento: Fila 100, Columna 100

Su ID de entrada es: XB46SGjU

## 6.Despues de tener el asiento deseado podemos verificar nuestro boleto

```
1. Busqueda de partidos.  
2. Venta de entradas.  
3. Validar boleto.  
4. Buscar productos.  
5. Comprar productos.  
6. Mostrar estadísticas.  
7. Salir.  
Elija una opcion: 3  
Ingrese ID del Boleto: XB46SGjU  
El boleto es válido. Cliente: Andres Arrieta
```

# 7. Si se posee una entrada VIP se puede comprar productos ,como el consumo alcoholico esta arraigada a la edad del cliente

```
-----  
Nombre del producto que desea comprar ('fin' para finalizar): Inteligente Acero Zapatos  
Nombre del producto que desea comprar ('fin' para finalizar): Increible Plástico Pantalones  
Nombre del producto que desea comprar ('fin' para finalizar): Ergonómico Algodón Pollo  
Nombre del producto que desea comprar ('fin' para finalizar): fin  
Productos seleccionados:  
Nombre: Inteligente Acero Zapatos  
Cantidad: 6182  
Precio (con IVA): 968.6  
Adicional: alcoholic  
Stock: 351  
-----  
Nombre: Increible Plástico Pantalones  
Cantidad: 339  
Precio (con IVA): 379.32  
Adicional: package  
Stock: 249  
-----  
Nombre: Ergonómico Algodón Pollo  
Cantidad: 14465  
Precio (con IVA): 737.76  
Adicional: alcoholic  
Stock: 353  
-----  
Monto total: 2085.6800000000003  
¿Desea proceder con la compra? (Si/No): █
```

# Diagrama de Clases

