

# Proyecto Final de Python

Comisión 18695

Francisco Martín Carranza



## Búsqueda de datos con ciertos requisitos

Se plantea la necesidad de no permitir que el usuario introduzca datos inválidos en la búsqueda de deportistas. Por ejemplo, números o símbolos donde va el nombre o apellido, o letras donde sólo se permiten números.

Para esto, se utilizará la variable **error** que almacenará el error según corresponda. Por ejemplo para un futbolista:

- “Debe ingresar un nombre válido”: Si se introduce algo que no sea una letra (se realiza la excepción de los espacios para más de un nombre).
- “Debe ingresar un apellido válido”: Igual al caso del nombre.
- “Debe ingresar una edad válida”: Si el dato introducido no es del tipo entero.

Primero se presenta la función completa, y luego se detalla cada sección:

```
def lista_futbolistas(request):
    futbolistas = None
    error = None
    if request.method == 'GET':
        nombre = request.GET.get('nombre', '')
        apellido = request.GET.get('apellido', '')
        edad = request.GET.get('edad', None)
        if nombre == '' and apellido == '' and edad == '':
            futbolistas = Futbolista.objects.all()
        elif nombre:
            if nombre.replace(' ', '').isalpha():
                futbolistas = Futbolista.objects.filter(nombre = nombre)
            else:
                error = 'Debe ingresar un nombre válido'

        elif apellido:
            if apellido.replace(' ', '').isalpha():
                futbolistas = Futbolista.objects.filter(apellido = apellido)
            else:
                error = 'Debe ingresar un apellido válido'

        elif edad:
```

```

    try:
        edad = int(edad) #Por si no se introducen solo números
        futbolistas = Futbolista.objects.filter(edad = edad)
    except:
        error = 'Debe ingresar una edad válida'

    return render(request, 'AppCoder/lista_futbolistas.html', {'futbolistas':
futbolistas, 'error': error})

```

Se inicializan las variables en cero al llamar la función:

```

def lista_futbolistas(request):
    futbolistas = None
    error = None

```

Se filtran y obtienen los campos según corresponda, solo si el método de la solicitud es **GET**:

```

if request.method == 'GET':
    nombre = request.GET.get('nombre', '')
    apellido = request.GET.get('apellido', '')
    edad = request.GET.get('edad', None)

```

Se considera que si el usuario no introduce ninguna variable a buscar, quiere presentar todos los futbolistas (o deportistas) en su totalidad:

```

if nombre == '' and apellido == '' and edad == '' :
    futbolistas = Futbolista.objects.all()

```

Caso contrario, comienza una verificación en cadena para ver si el usuario introdujo nombre, o si introdujo apellido, o por último si introdujo una edad a buscar.

Para el caso del **nombre**, se verifica si la variable nombre, luego de suprimir los espacios, tiene solamente letras con la función **.isalpha()**. Se permiten los espacios para aquellos que tengan segundos nombres. En ese caso, se considera que se puede filtrarse el nombre con el dato introducido:

```

elif nombre:
    if nombre.replace(' ','').isalpha():
        futbolistas = Futbolista.objects.filter(nombre = nombre)
    else:
        error = 'Debe ingresar un nombre válido'

```

Se considera igual para el **apellido**. Sin esto, no sería posible introducir más de un apellido:

```
elif apellido:
    if apellido.replace(' ','').isalpha():
        futbolistas = Futbolista.objects.filter(apellido = apellido)
    else:
        error = 'Debe ingresar un apellido válido'
```

Por último, para la edad se realiza un **try vs except**. Debe filtrar el campo **edad** en la búsqueda siempre y cuando sea un valor entero. Si se da un error, entrará en la condición del **except**.

```
elif edad:
    try:
        edad = int(edad)
        futbolistas = Futbolista.objects.filter(edad = edad)
    except:
        error = 'Debe ingresar una edad válida'
```

La última sentencia es el **return** donde se devuelven bajo **render** los objetos de **futbolistas**, el **error** (si existiera), se regresa a la misma URL del listado de futbolistas, y claro, el **request** de la solicitud. Si el método no hubiera sido **GET**, como en las primeras líneas de función, entraría directamente a la misma línea a continuación:

```
return render(request, 'AppCoder/lista_futbolistas.html',
{'futbolistas': futbolistas, 'error': error})
```

Claro que si el método no fuera **GET** significa que el usuario está ingresando por primera vez a la URL.

Si no hubiera error, o si no se encuentran futbolistas que cumplan los campos introducidos, o si es la primera vez que ingresa el usuario se muestra el mensaje *“No se encontraron futbolistas”*. Esto puede verse en detalle en el código del HTML donde se invoca al listado de deportistas.

Ejemplos:

1. Ingreso por primera vez a la URL:

← → ↻ ⓘ 127.0.0.1:8000/futbolistas/

Futbolistas Basquetbolistas Tenistas About me

[Crear Nuevo Futbolista](#)

Nombre a buscar...

Apellido a buscar...

Edad a buscar...

Buscar

**No se encontraron futbolistas**

2. Al seleccionar “*Buscar*” sin introducir campos:

---

[Crear Nuevo Futbolista](#)

Nombre a buscar...

Apellido a buscar...

Edad a buscar...

Buscar

Listorti José María [Detalle](#) | [Borrar](#) | [Actualizar](#)

Messi Lionel Andrés [Detalle](#) | [Borrar](#) | [Actualizar](#)

Ronaldo Cristiano [Detalle](#) | [Borrar](#) | [Actualizar](#)

3. Se introduce un **nombre** no válido:

---

[Crear Nuevo Futbolista](#)

José María!

Apellido a buscar...

Edad a buscar...

Buscar

Listorti José María [Detalle](#) | [Borrar](#) | [Actualizar](#)

Messi Lionel Andrés [Detalle](#) | [Borrar](#) | [Actualizar](#)

Ronaldo Cristiano [Detalle](#) | [Borrar](#) | [Actualizar](#)

---

[Crear Nuevo Futbolista](#)

Nombre a buscar...

Apellido a buscar...

Edad a buscar...

Buscar

**Debe ingresar un nombre válido**

4. Se introduce un **apellido** no válido:

---

[Crear Nuevo Futbolista](#)

Nombre a buscar...

Messi7

Edad a buscar...

Buscar

Listorti José María [Detalle](#) | [Borrar](#) | [Actualizar](#)

Messi Lionel Andrés [Detalle](#) | [Borrar](#) | [Actualizar](#)

Ronaldo Cristiano [Detalle](#) | [Borrar](#) | [Actualizar](#)

[Crear Nuevo Futbolista](#)

Nombre a buscar...

Apellido a buscar...

Edad a buscar...

Buscar

**Debe ingresar un apellido válido**

5. Se introduce una **edad** no válida:

---

[Crear Nuevo Futbolista](#)

Nombre a buscar...

Apellido a buscar...

48f

Buscar

Listorti José María [Detalle](#) | [Borrar](#) | [Actualizar](#)

Messi Lionel Andrés [Detalle](#) | [Borrar](#) | [Actualizar](#)

Ronaldo Cristiano [Detalle](#) | [Borrar](#) | [Actualizar](#)

---

[Crear Nuevo Futbolista](#)

Nombre a buscar...

Apellido a buscar...

Edad a buscar...

Buscar

**Debe ingresar una edad válida**

6. Ejemplo de búsqueda exitosa:

---

[Crear Nuevo Futbolista](#)

José María

Apellido a buscar...

Edad a buscar...

Buscar

Listorti José María [Detalle](#) | [Borrar](#) | [Actualizar](#)

Messi Lionel Andrés [Detalle](#) | [Borrar](#) | [Actualizar](#)

Ronaldo Cristiano [Detalle](#) | [Borrar](#) | [Actualizar](#)

---

### [Crear Nuevo Futbolista](#)

Listorti José María [Detalle](#) | [Borrar](#) | [Actualizar](#)