



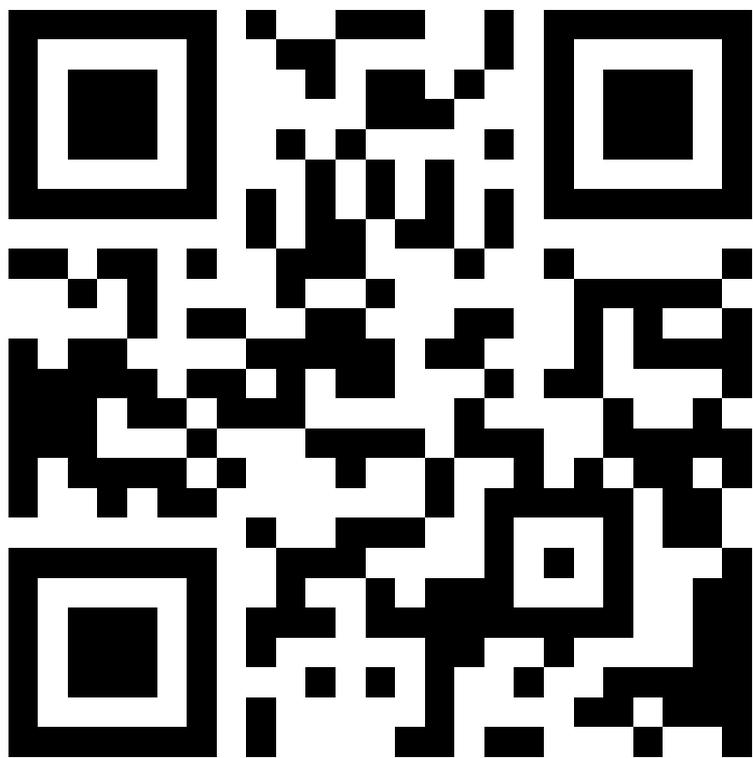
# Lidiando con formatos y unidades en nuestra API REST

**Matuu Varela**

#PyConAr2018 - 23/11/18

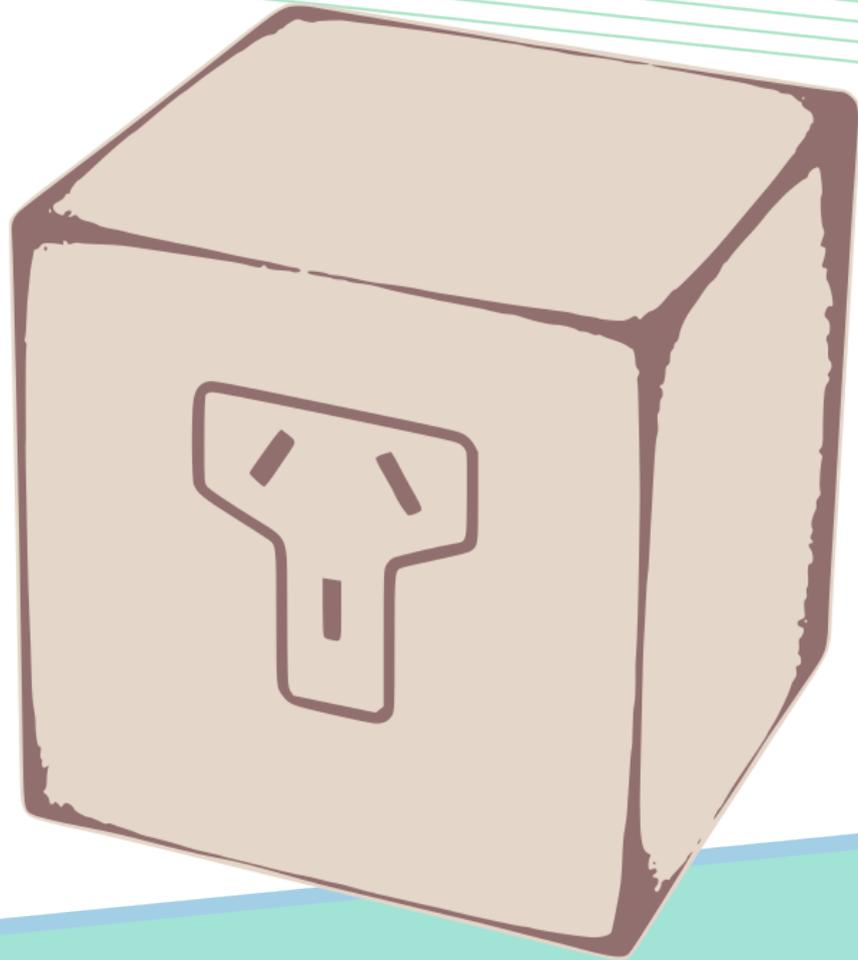
# Hola...

- Pampeano, vivo en San Rafael (Mza).
- Ingeniero en informática.
- Miembro de la Asociación Civil Python Argentina (Associate!!)
- Desarrollo en Python/Django desde hace más de 5 años.
- Trabajo en Kilimo (Córdoba)

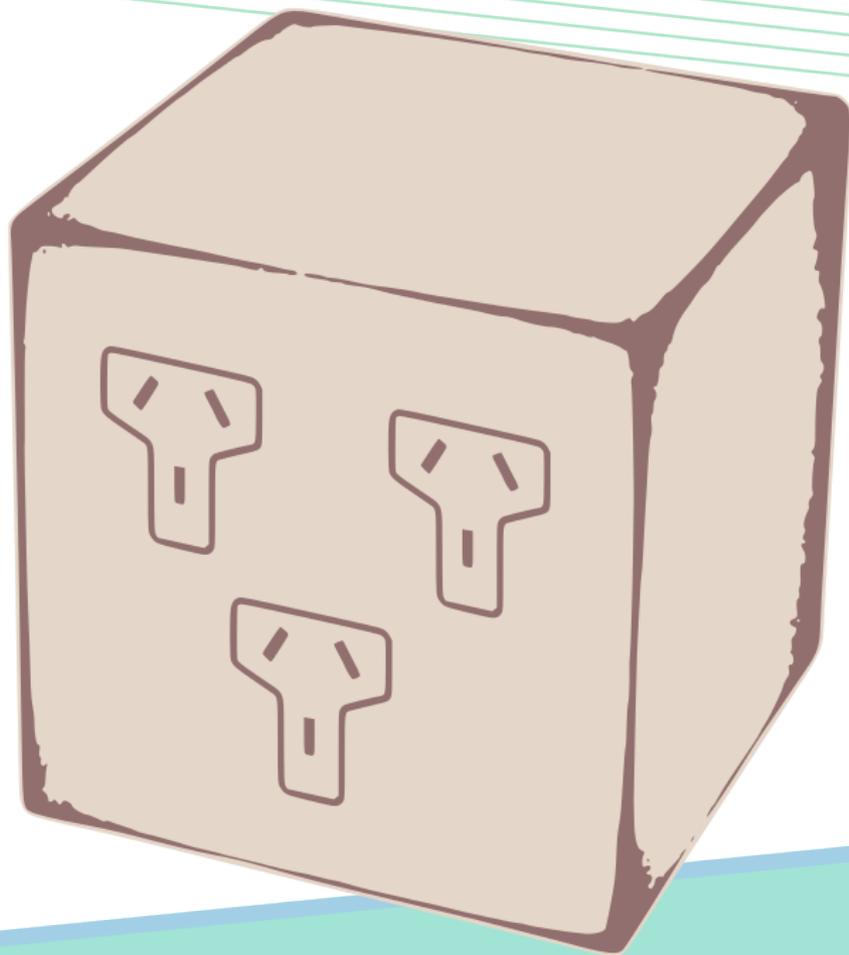


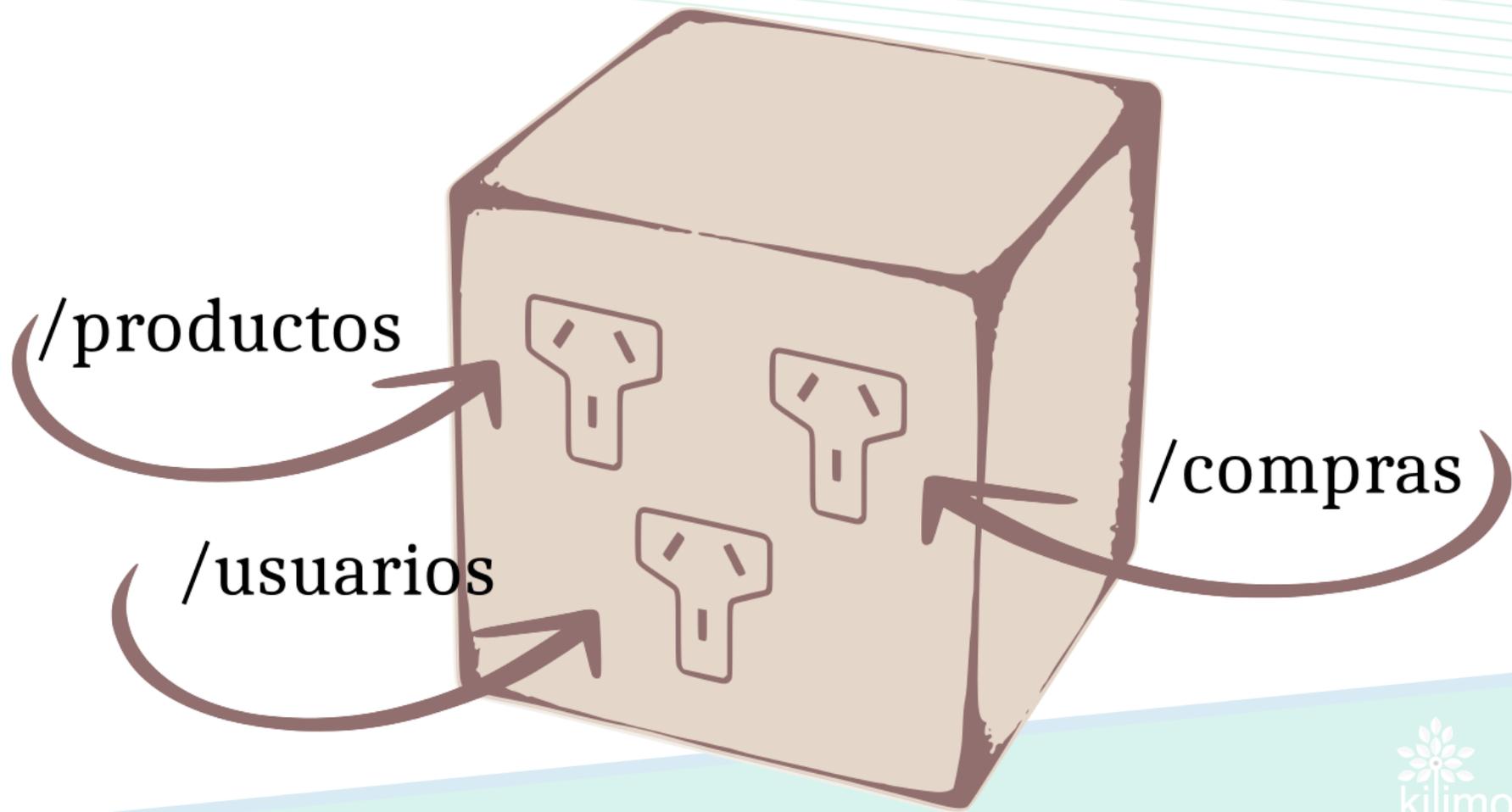
<https://goo.gl/Uu645x>

# API REST



# Recursos





# Serializadores

```
class Producto:
```

```
    nombre = models.CharField(max_length=100)  
    codigo = models.CharField(max_length=100)  
    descripcion = models.CharField(max_length=200)  
    precio = models.DecimalField(max_digits=10, decimal_places=2)  
    activo = models.BooleanField(default=True)
```

```
{  
  "nombre": "Remera PyCon 2018",  
  "codigo": "AR123470669",  
  "descripcion": "Remera oficial de PyCon 2018 CABA",  
  "precio": 400,  
  "activo": true  
}
```

## Deserialización

```
class Producto:  
    nombre = models.CharField(max_length=100)  
    codigo = models.CharField(max_length=10)  
    descripcion = models.TextField(max_length=1000)  
    precio = models.DecimalField(max_digits=10, decimal_places=2)  
    activo = models.BooleanField(default=True)
```

Objeto  
Python

Serialización

```
"nombre": "Remera PyCon 2018",  
"codigo": "AR123470669",  
"descripcion": "Remera oficial de PyCon 2018 CABA",  
"precio": 400,  
"activo": true
```

JSON

# Campos (Field)

Manejan la conversión entre los datos primitivos y los valores internos (python)

SISTEMA

# Fields

`to_representation(self, obj)`

`to_internal_value(self, data)`

EXTERIOR

# Problema

Mi sistema maneja datos en unidades y formatos específicos, pero tengo que brindar esos datos en otras unidades y formatos diferentes

# Problema

Ejemplo:

- Un usuario ingresa temperatura en °C, y pero otro desea visualizarlas en °F.
- Algunos usuarios ingresan la velocidad del viento en Km/h, otros en Mph, pero el sistema los procesa en m/s.

**Tip**

**Convertir la información en la  
frontera del sistema**

Veamos como crear nuestros propios  
**field** para manejar estas  
conversiones.

```
class VientoField(serializers.Field):
    """
    Campo para el manejo de los datos de viento
    """
    def to_representation(self, value):
        return viento_para_api(self.perfil, value)

    def to_internal_value(self, data):
        return viento_desde_api(self.perfil, data)
```

```
def viento_para_api(perfil, valor):  
    """  
    Transformar el valor de viento a la unidad esperada por el cliente.  
    """  
    if perfil.viento_pref == 'kph':  
        return valor  
    else:  
        return kms_a_mph(valor)
```

```
def viento_desde_api(perfil, valor):  
    """  
    Transformar el viento informado por el  
    cliente a la unidad esperada.  
    """  
    if perfil.viento_pref == 'kph':  
        return valor  
    else:  
        return mph_a_kms(valor)
```

Otro ejemplo...

```
class LluviaField(serializers.Field):  
    """  
    Campo para el manejo de los datos de lluvia  
    """  
    def to_representation(self, value):  
        return lluvia_para_api(self.perfil, value)  
  
    def to_internal_value(self, data):  
        return lluvia_desde_api(self.perfil, data)
```

## ¿Cómo usamos nuestros *fields*?

```
class RegistroSerializer(serializers.ModelSerializer):  
    temperatura = TemperaturaField()  
    viento = VientoField()  
    lluvia = LluviaField()  
  
    class Meta:  
        model = Registro  
        fields = ('temperatura', 'viento', 'lluvia')
```

**¿Y el contexto?**

En la 'view':

```
def get_serializer_context(self):  
    """  
    Inserta en el contexto del serializador,  
    el request y el usuario.  
    """  
    return {  
        'user': self.request.user,  
        'request': self.request  
    }
```

En el 'serializer':

```
self.context["user"].perfil
```

## Resumen:

- Crear tus propios 'fields' para manejar las transformaciones.
- Utilizar tus *fields* para crear los serializadores.
- Incluir en el contexto del serializador el contexto de la 'request' (para saber quien envia o solicita los datos).

**¿Lo vemos en acción?**



<https://goo.gl/q6g1sZ>



**¡MUCHAS GRACIAS!**

Twitter: @matuvarela

Telegram: @m4tuu