

{dev/talles}

django

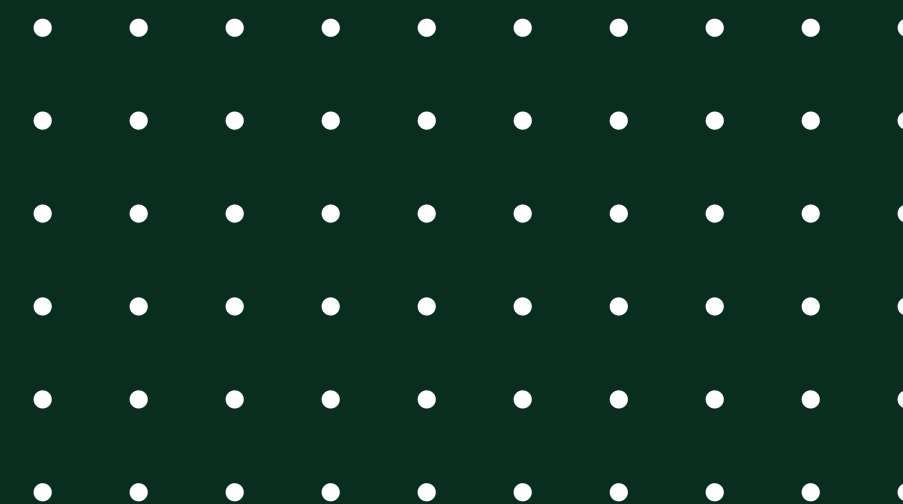
# Introducción a Django Rest Framework



Por: Ricardo Cuéllar

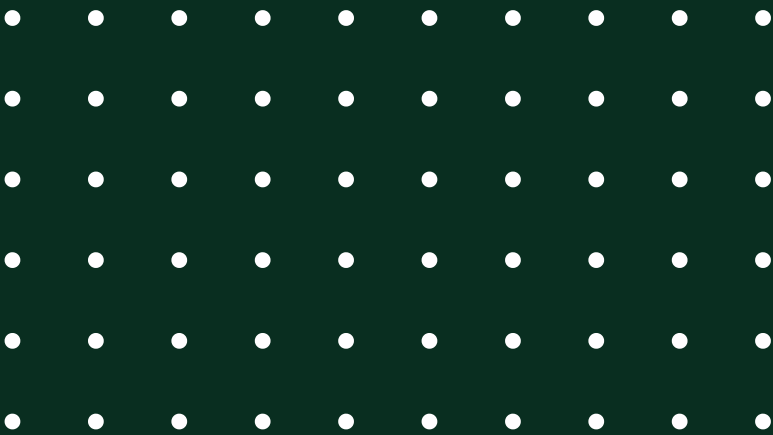
# ¿Qué es Django Rest Framework?

- Es una extensión de **Django** para crear APIs RESTful
- Convierte tu backend en un proveedor de datos.
- Incluye herramientas como autenticación, permisos, serialización, paginación, etc.
- Es muy usado en proyectos reales.



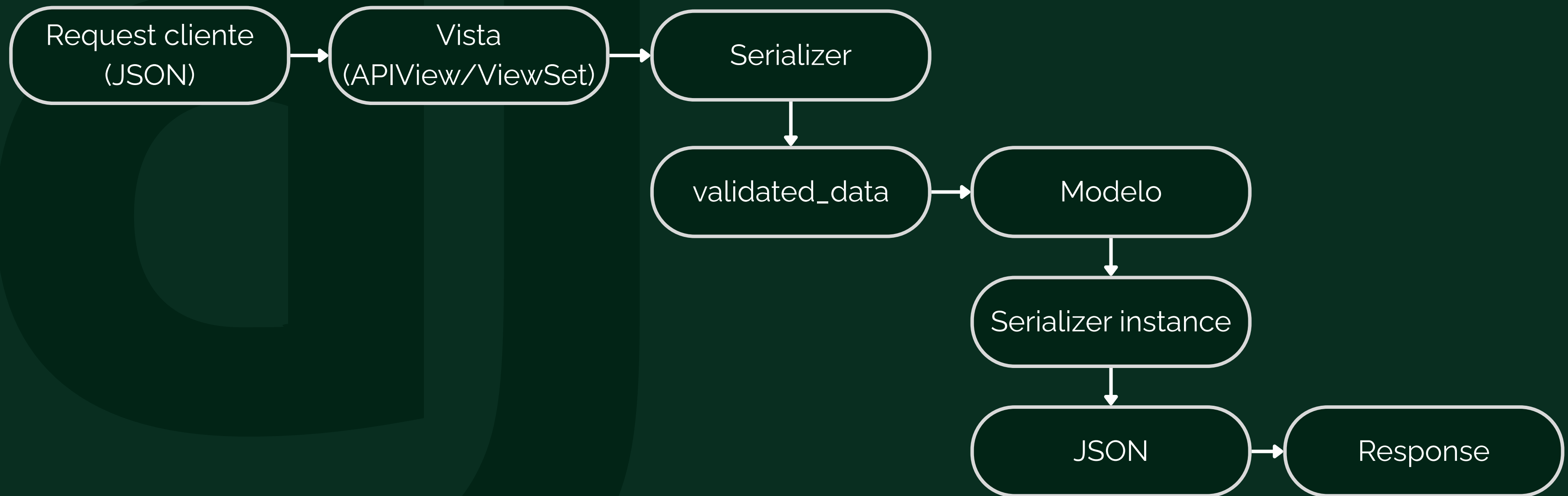
# Diferencias con Django tradicional

Django Tradicional	Django Rest Framework (DRF)
Pensado para HTML Templates	Pensado para APIs
Usa views que retornan plantillas y HttpResponseRedirect	Usa APIView, ViewSet, Response
Validaciones con forms o ModelForms	Validaciones con Serializers
Más trabajo a mano para exponer JSON	CRUD rápido



# ¿Cómo funciona DRF?

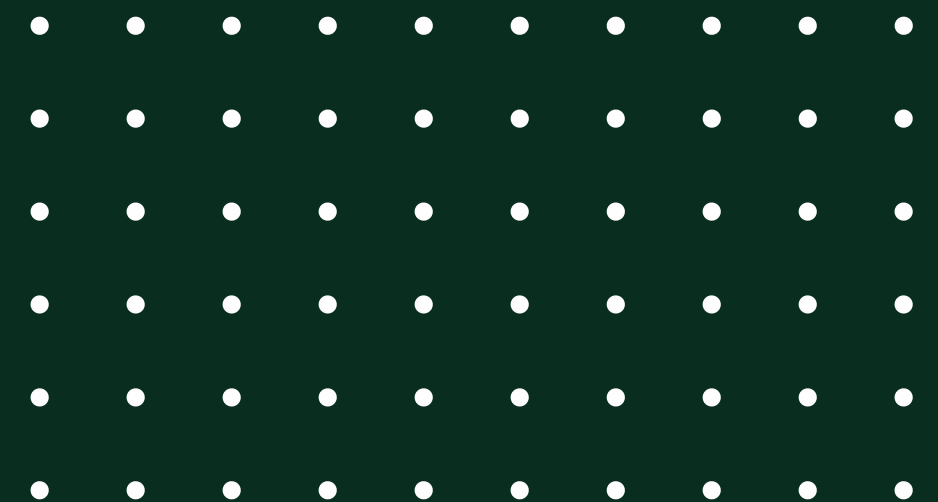
## Flujo básico



# ¿Qué es un serializer?

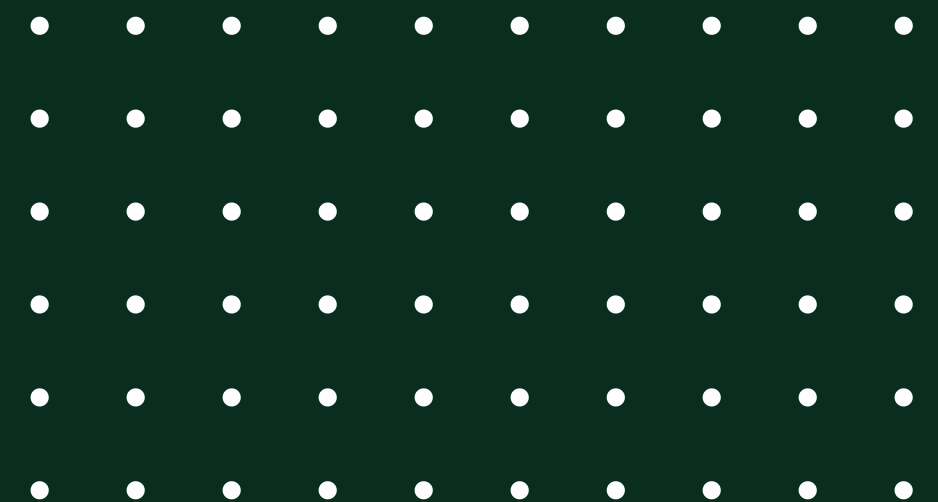
## El corazón de DRF

- Traducen datos entre JSON y modelos de Django.
- Hacen validación automática (tipos, longitudes, reglas).
- Deciden que campos exponer y cómo nombrarlos.
- Pueden anidar otros serializers para relaciones.
- Permite lógica extra.



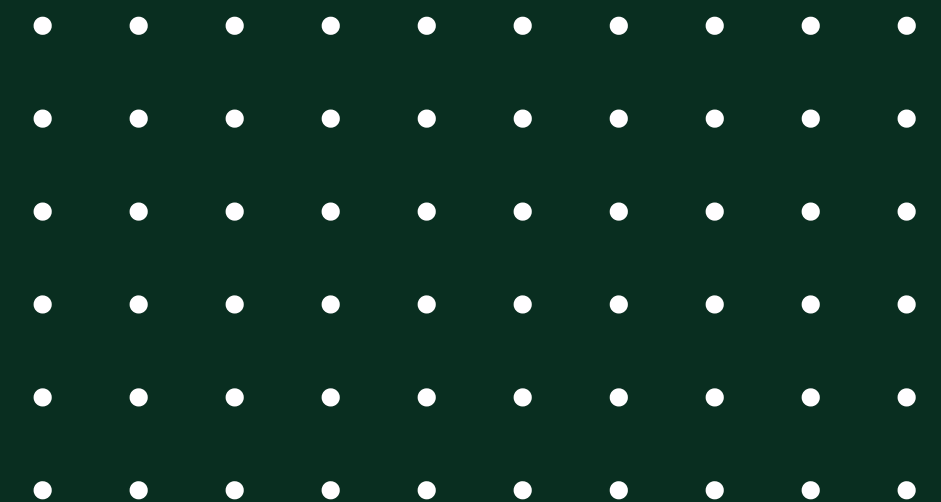
# ¿Por qué usar DRF?

- Todo el poder de **Django** en la palma de tu mano.
- Ahorra tiempo: CRUD completo en pocas líneas.
- Seguridad: permisos y autenticación integrada.
- API navegable para probar endpoints.
- Usado en productos y startups.
- Se puede personalizar todo.



# Desventajas y limitaciones

- Curva de aprendizaje mayor que Django tradicional. (Ya tienes avance)
- Mucha magia.
- Menos rápido que otros frameworks.
- Mucho código para lógica más personalizada.
- Dependiente al ORM de Django.
- Puede ser muy grande para proyectos muy pequeños.





# iVamos a practicar!

