

Notas sobre la implementación del Ejercicio:

1. Otra opción a considerar para el modelo de Figuras, sería definir una interfaz común que solo defina los métodos comunes para todas las figuras (ej: getSuperficie, getPerimetro). Luego clases que la implementen, especifican estos métodos, y también definen sus métodos particulares como getDiametro para Circulo, getBase/getAltura para Triangulo etc..., de esta manera no es necesario retornar null en caso que la figura no posea la característica.

2. Dado que para el punto 1 se implementa una Factory de Figuras que recibe un parámetro indicando el tipo de figura. Se diseñó la API Rest siguiendo ese concepto, por lo cual el request del método post a *'/figuras'* es:

{ "tipo": "triangulo", "parametros": [2.5, 7.1] }.

Otra opción podría ser definir el método post para que el request sea por ejemplo:

{ "tipo": "triangulo", "base": 2.5, "altura": 7.1 }

Y tener en la Factory métodos particulares para crear cada Figura.

3. Dependiendo el uso se pueden agregar nuevos métodos a la API Rest como:

- Ingresar múltiples figuras mediante un POST request

```
[  
  { "tipo": "triangulo", "parametros": [2.5, 7.1] },  
  { "tipo": "cuadrado", "parametros": [3.0] },  
  { "tipo": "circulo", "parametros": [8.3] }  
]
```

- Obtener datos particulares de cada figura (superficie, base, diametro, ...) mediante un GET

GET: /figuras/{id}/{parametro}

- Obtener todas las figuras según el tipo