

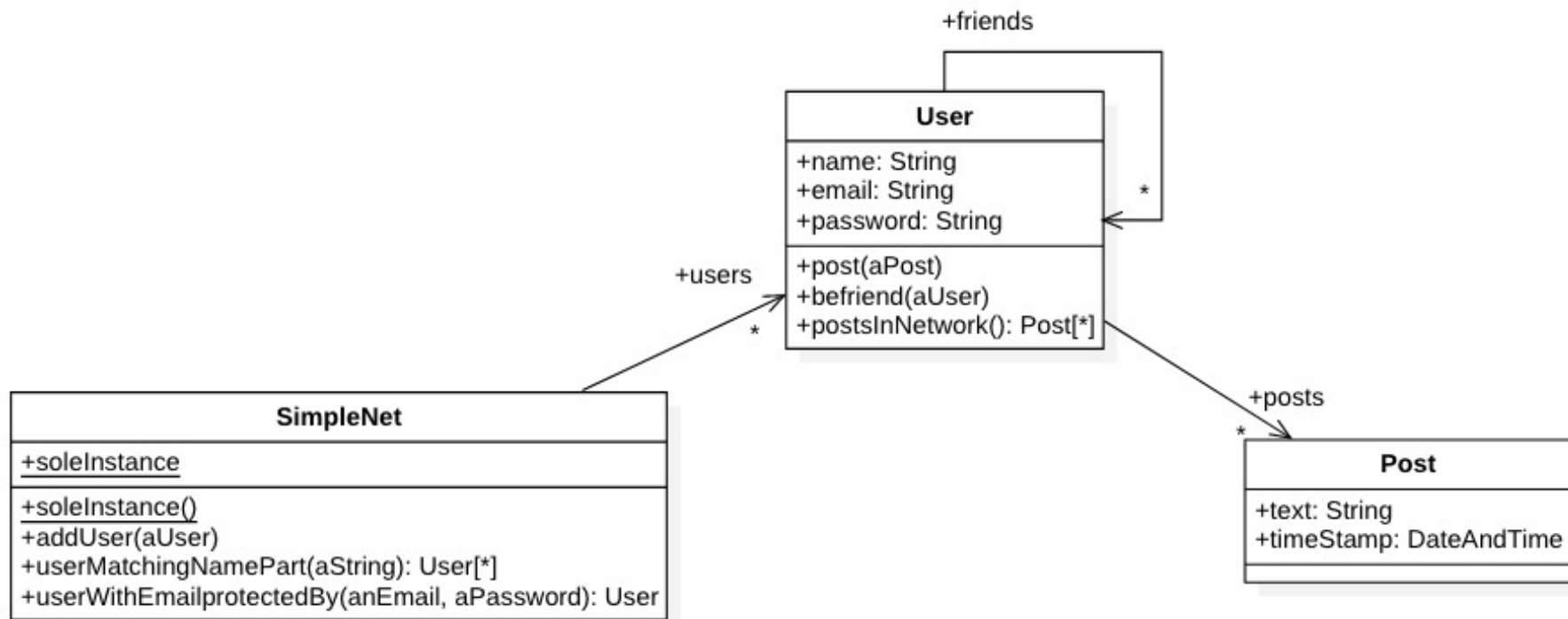
Un servicio REST

Usando los frameworks de caja negra Teapot y NeoJSON

Objetivos

- Explorar dos micro-frameworks: Teapot y NeoJSON
- Ambos usan composición como estrategia principal de instanciación
- Nos interesa ver la diferencia con lo que veníamos haciendo en Seaside
- Nos interesa ver otra forma en la que el mundo exterior interactúa con nuestro modelo de dominio

SimpleNet



SimpleNet REST

GET /users

```
[
  {
    "name": "Juana Delos Palotes",
    "email": "juanita@gmail.com"
  },
  {
    "name": "María Delos Palitos",
    "email": "maria@gmail.com"
  }
]
```

GET /users/juanita@gmail.com/posts

```
[
  {
    "text": "Mas contento que perro con dos colas !",
    "timeStamp": "2021-03-31T15:50:43.970369-03:00"
  },
  {
    "text": "Pasaré, pasará, pero el último quedará",
    "timeStamp": "2021-03-31T21:45:41.767842-03:00"
  }
]
```

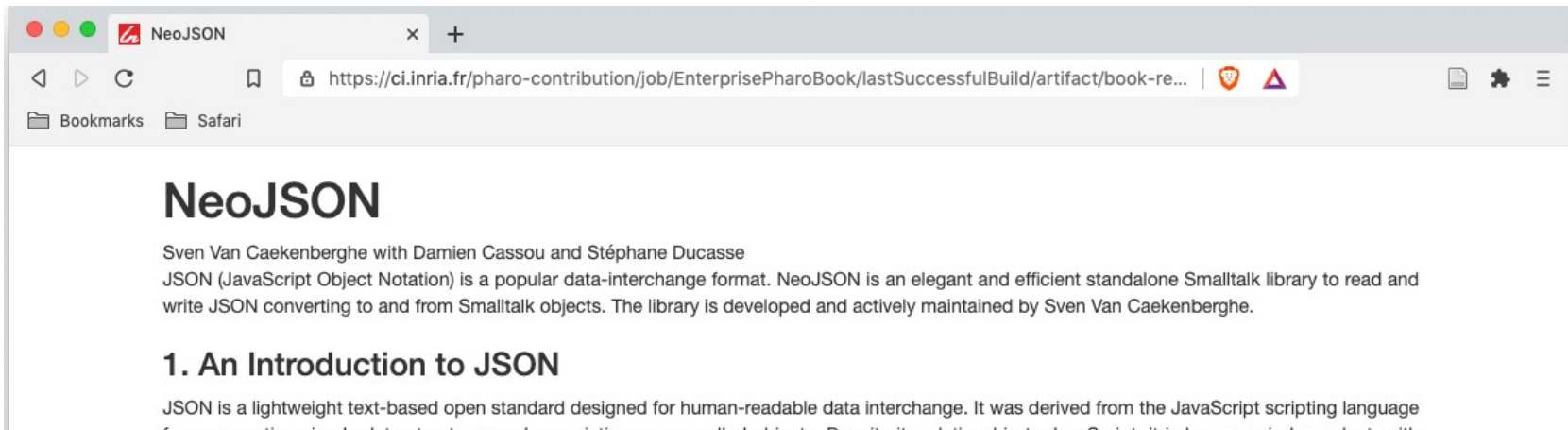
POST /users/juanita@gmail.com/posts/Me%20tomo%20unos%20minutos

Plan

- La nueva clase SimpleNetREST es nuestro servicio web
- Accede al modelo (que comparte con la app Web) via “SimpleNet soleInstance”
- Vamos a utilizar NeoJson para transformar nuestros User y Post en JSON
- Vamos a utilizar Teapot para implementar la API rest
- Vamos a utilizar PostMan para hacer las pruebas

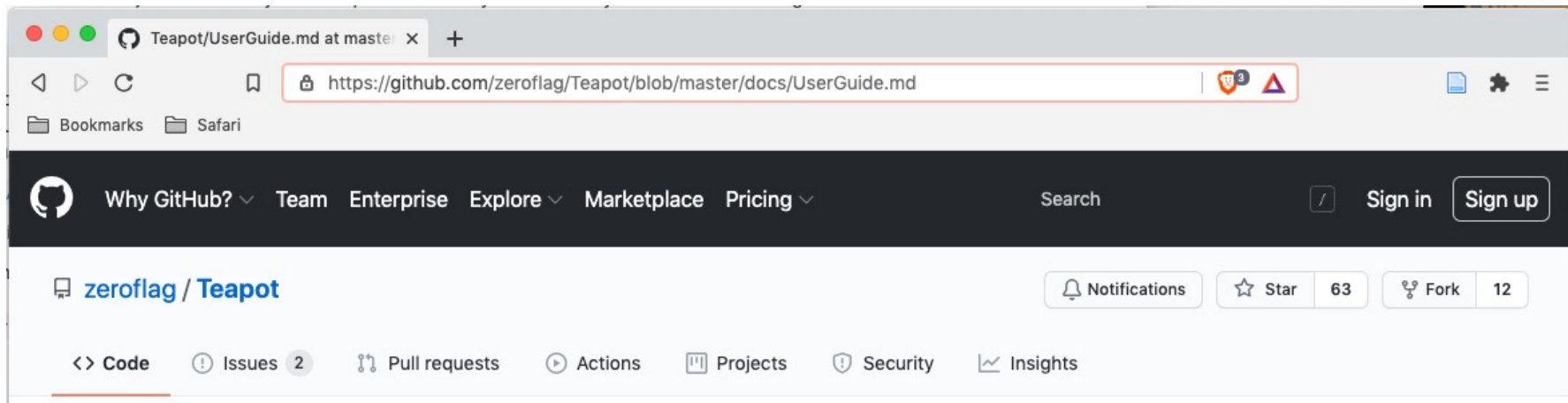
NeoJSON

- (Micro) Framework para mapear objetos desde/a JSON
- Bien documentado, simple, y flexible
- Podemos instalarlo desde el catálogo y desde el repositorio Github
 - <https://github.com/svenvc/NeoJSON>



Teapot

- (Micro) Framework para implementar servicios REST
- Bien documentado, simple, flexible y minimalista
- Podemos instalarlo desde el catálogo o desde Github
 - <https://github.com/zeroflag/Teapot>



Veamos ...

- Veamos que implica utilizar Teapot y NeoJON
- Prestamos atención a:
 - Que instanciamos
 - Que mensajes enviamos
 - Como retornan el control a nuestro código
- Tenemos el código a mano

Plan de ruta ...

