

En el servidor creo los objetos de la clase persona y le añado los atributos que corresponden al nombre, al apellido y a la edad de una persona. Con `delayElements` le digo el tiempo que tiene que tardar en crear a una persona.

Sumando cuanto tardan todas las personas, tardan 16 segundos en total.

```
@Override
public Flux<Person> personList1() {
    Person person = new Person();
    person.setFirstName("Marina");
    person.setLastName("Aparicio");
    person.setAge(22);
    return Flux.just(person).delayElements(Duration.ofSeconds(1));
}

@Override
public Flux<Person> personList2() {
    Person person = new Person();
    person.setFirstName("Antonia");
    person.setLastName("Olivares");
    person.setAge(60);
    return Flux.just(person).delayElements(Duration.ofSeconds(3));
}

@Override
public Flux<Person> personList3() {
    Person person = new Person();
    person.setFirstName("Lopez");
    person.setLastName("Vega");
    person.setAge(36);
    return Flux.just(person).delayElements(Duration.ofSeconds(5));
}

@Override
public Flux<Person> personList4() {
    Person person = new Person();
    person.setFirstName("Amparo");
    person.setLastName("Duque");
    person.setAge(57);
    return Flux.just(person).delayElements(Duration.ofSeconds(7));
}
```

Añado las personas al controlador del Servidor para que pueda pasar al cliente.

```
@RestController
public class PersonListController {

    @Autowired
    ListPersonServiceI listPersonServiceI;

    @GetMapping("/person-list-1")
    public Flux<Person> personList1(){
        return listPersonServiceI.personList1();
    }

    @GetMapping("/person-list-2")
    public Flux<Person> personList2(){
        return listPersonServiceI.personList2();
    }

    @GetMapping("/person-list-3")
    public Flux<Person> personList3(){
        return listPersonServiceI.personList3();
    }

    @GetMapping("/person-list-4")
    public Flux<Person> personList4(){
        return listPersonServiceI.personList4();
    }
}
```

En el cliente llamo a las personas creadas en el servidor con webClient añadiendo el puerto del servidor. También, con subscribe hago que se muestre cada persona tardando los segundos indicados. En este caso se mostrarán solo las personas cuyo nombre comience por A con startsWith.

```
@Service
public class PersonService implements PersonI {

    @Override
    public Flux<Person> persons() {

        WebClient webClient = WebClient.create("http://localhost:8087");

        Flux<Person> person1 = webClient.get().uri("/person-list-1").retrieve().bodyToFlux(Person.class);
        Flux<Person> person2 = webClient.get().uri("/person-list-2").retrieve().bodyToFlux(Person.class);
        Flux<Person> person3 = webClient.get().uri("/person-list-3").retrieve().bodyToFlux(Person.class);
        Flux<Person> person4 = webClient.get().uri("/person-list-4").retrieve().bodyToFlux(Person.class);

        Flux<Person> personsListStartsA = Flux.concat(person1, person2, person3, person4)
            .filter(person -> person.getFirstName().startsWith("A"));

        personsListStartsA.subscribe(person -> System.out.println(person.getLastName() + ", " + person.getFirstName() + " tiene " + person.get/

        Flux<Person> personsList = Flux.concat(person1, person2, person3, person4);

        return personsList;
    }
}
```

Por último, con el controlador del cliente hago que las personas se muestren en personList.html

```
@Controller
public class PersonListController {

    @Autowired
    PersonService personService;

    @GetMapping("/person-list")
    public String personList(final Model model) {
        final Flux<Person> personList = personService.persons();
        model.addAttribute("personList", personList);
        return "personList";
    }
}
```

```
<html xmlns:th="http://www.thymeleaf.org">
```

```
    <head>
        <title>Lista de personas</title>
    </head>

    <body>
        <table>
            <thead>
                <tr>
                    <th>Nombre</th>
                    <th>Apellido</th>
                    <th>Edad</th>
                </tr>
            </thead>
            <tbody>
                <tr th:each="person : ${personList}">
                    <td><div th:text="${person.firstName}"></div></td>
                    <td><div th:text="${person.lastName}"></div></td>
                    <td><div th:text="${person.age}"></div></td>
                </tr>
            </tbody>
        </table>
    </body>
```

---

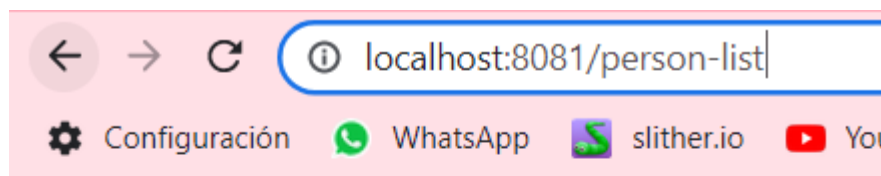
En la consola se muestran las personas cuyo nombre comienza por “A” tardando tantos segundos como se ha indicado.

```
WebFluxClient - WebFluxApplication [Spring Boot App] C:\Program Files\Java\jdk-11.0.10\bin\javaw.exe (6 oct. 2022 23:56:2)

:: Spring Boot :: (v2.7.4)

2022-10-06 23:56:28.434 INFO 3796 --- [main] c.n.bootcamp.webf
2022-10-06 23:56:28.439 INFO 3796 --- [main] c.n.bootcamp.webf
2022-10-06 23:56:31.758 INFO 3796 --- [main] o.s.b.web.embedde
2022-10-06 23:56:31.788 INFO 3796 --- [main] c.n.bootcamp.webf
Olivares, Antonia tiene 60 años
Duque, Amparo tiene 57 años
Olivares, Antonia tiene 60 años
Olivares, Antonia tiene 60 años
Duque, Amparo tiene 57 años
Duque, Amparo tiene 57 años
```

En el navegador se muestra una lista con todas las personas con la url localhost:8081/person-list.



Nombre	Apellido	Edad
--------	----------	------

Marina	Aparicio	22
--------	----------	----

Antonia	Olivares	60
---------	----------	----

Lopez	Vega	36
-------	------	----

Amparo	Duque	57
--------	-------	----

Se muestra que ha tardado en cargar tantos segundos como el total de lo que cargan todas las personas, en este caso 16 segundos.

