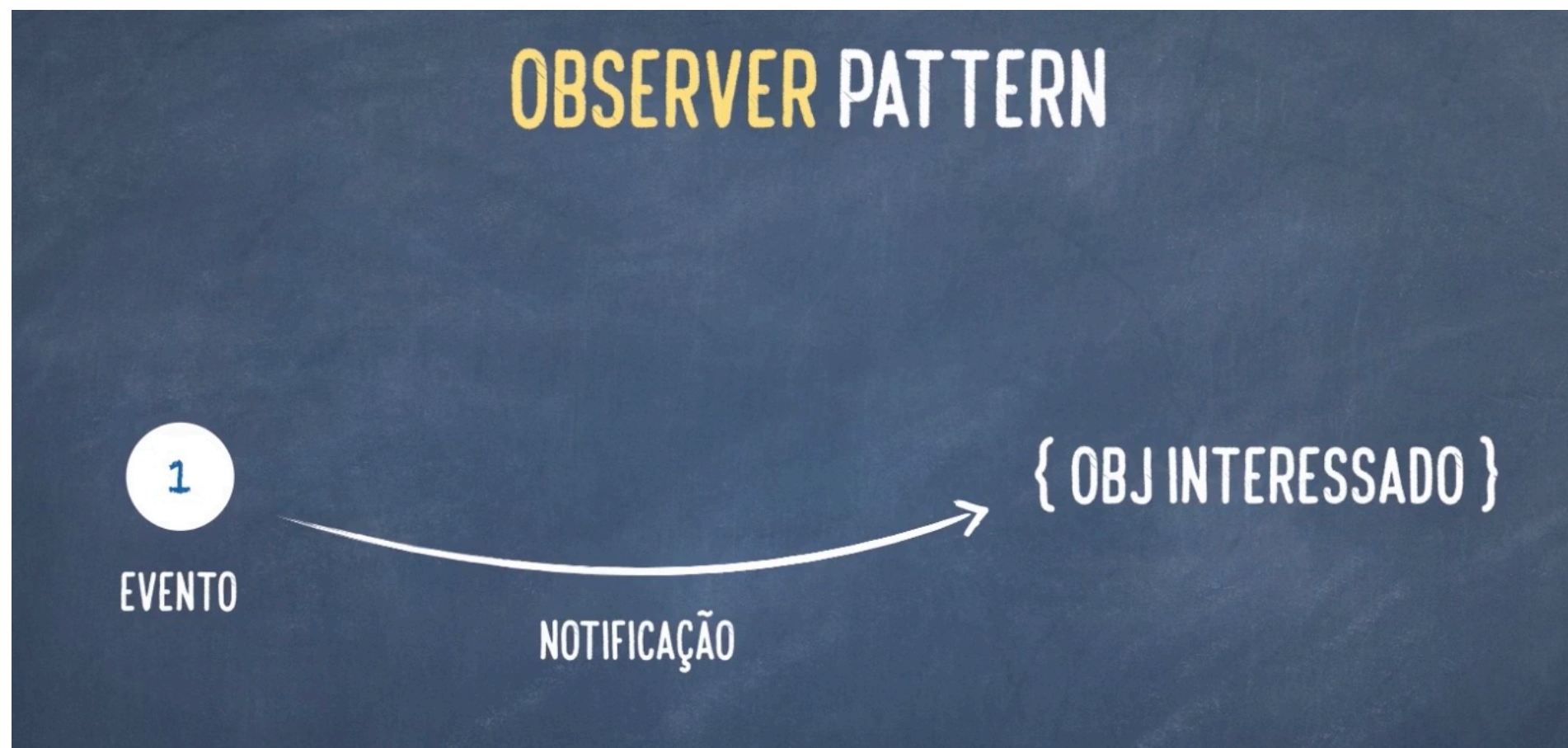


React programming e Http



A idéia de programação reativa é simples. Um evento acontece e os que estão interessados são notificados e reagem a ele.

Ela é baseada num padrão chamado de Observer. Nesse padrão existe o objeto que é o foco do interesse e outros objetos que estão interessados em mudanças nesse objeto, que são os listeners.

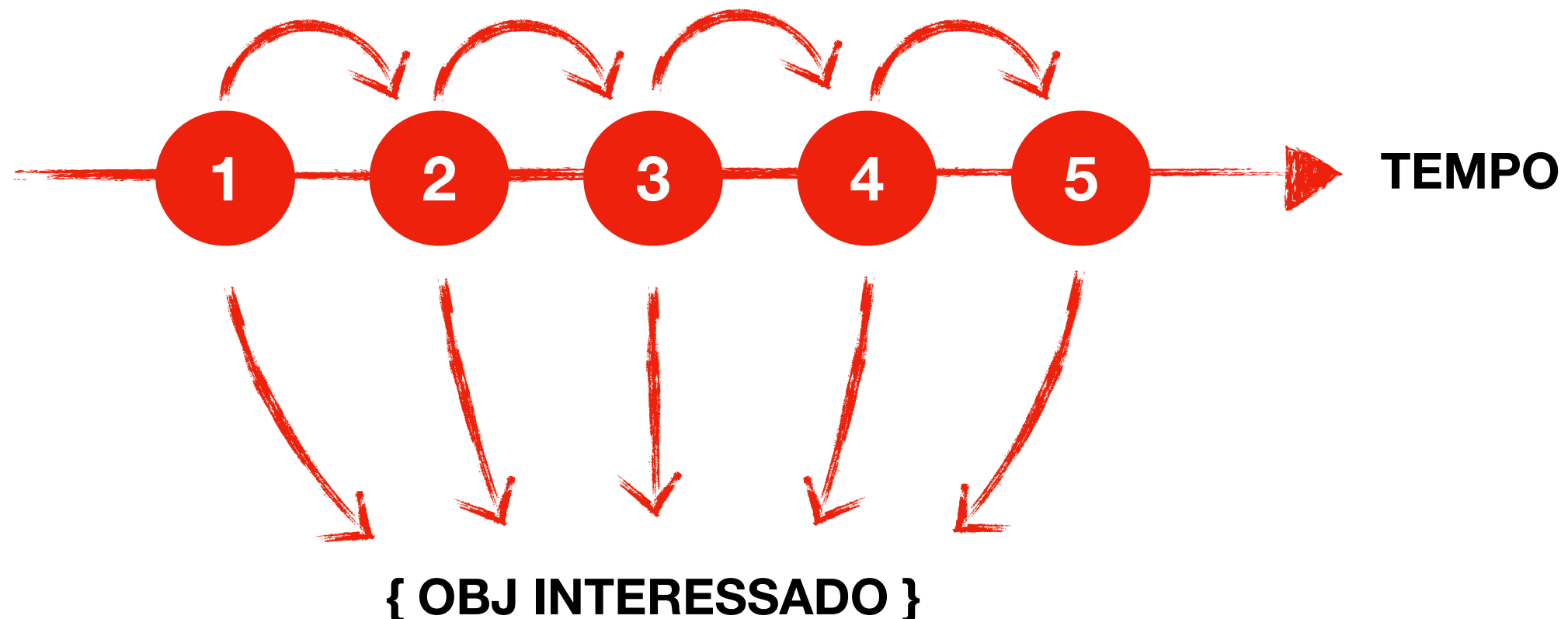


React programming e Http



Em react programming os eventos acontecem em uma cadeia chamada stream, ou seja, uma sequencia de eventos.

Cada evento gera uma notificação aos objetos interessados. Assim a stream vai ser consumida item a item até que não restem mais objetos.

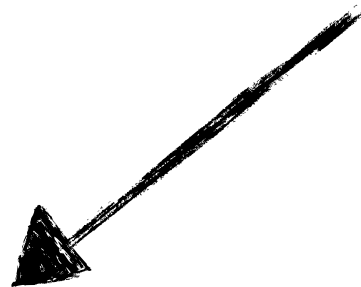


React programming e Http



É a combinação de dois padrões:

Iterator + Observable



Item a item na stream



**Notifica os listeners
interessados**

React programming e Http



Quando ocorre algum evento que muda o estado de um objeto, os listeners que estão inscritos como interessados são avisados e aí executam alguma ação.

Em angular, várias partes do framework são implementadas usando esse conceito. Por exemplo, ao realizar uma chamada http ao backend, parte da aplicação vai se inscrever e esperar pela resposta (evento). Quando o evento acontecer, a aplicação reage e extrai os dados necessários.

React programming e Http



A biblioteca que o angular usa para implementar react programming é o rxjs. Aqui o objeto principal é Observable. Ele possui uma série de funções ou operadores para inscrição e manipulação de eventos.

Tratamento de erros



Um objeto Observable não executa uma ação imediatamente, na verdade, quando o implementamos estamos fazendo apenas a configuração dele. A execução só acontecerá após “alguém” se inscrever nesse objeto.

```
listarCategorias(): Observable<Categoria[]>{  
  return this.http.get<Categoria[]>(`${CINEMA_API}/categorias`);  
}
```

Existem duas forma de tratar erros no Angular. Uma é através da implementação de uma interface ErrorHandler, criando uma classe que faz uma tratamento de erros global, declarando-a na lista de providers.

A outra maneira é usar o operador pipe. Pipe espera então por uma função (catchError) que recebe um response e devolve um outro objeto Observable.

Tarefa



Crie um site que apresente uma lista de alunos matriculados em uma escola. Essa lista deve ser lida de um serviço backend (json-server). Devem ser apresentados os seguintes dados dos alunos: Matrícula, Nome, Idade e Turma. O site deve ter no mínimo a seguinte estrutura:

