

# Controlando o Tempo

com Typescript e Event Sourcing

# Lucas Santos

Solutions Architect  
**nextcode**



**Microsoft**<sup>®</sup>  
Most Valuable  
Professional



@khaosdoctor



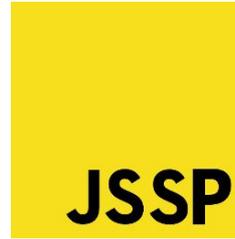
lsantos.me



/khaosdoctor



@\_staticvoid



@khaosdoctor



Isantos.me



/khaosdoctor



@\_staticvoid

O que é Event  
Sourcing?

Capture all changes to an application state as a sequence of events.



@khaosdoctor



Isantos.me



/khaosdoctor



@\_staticvoid

# Event Sourcing

- Padrão de arquitetura
- Apareceu pela primeira vez em um artigo de Martin Fowler em 2005
- A entidade tem um estado inicial
- O estado inicial só pode ser alterado por eventos
- Tudo é uma stream de eventos
- Base de padrões como Flux e Redux



@khaosdoctor



Isantos.me



/khaosdoctor



@\_staticvoid

O que isso implica?

# O que isso implica?

- Facilidade de debugging
- Log automático
- Capacidade de desfazer alterações de forma simples
- Reconstrução de estados anteriores!



@khaosdoctor



lsantos.me



/khaosdoctor



@\_staticvoid

# O que isso implica?

- Facilidade de debugging
- Log automático
- Capacidade de desfazer alterações de forma simples
- **Reconstrução de estados anteriores!**



@khaosdoctor



Isantos.me



/khaosdoctor



@\_staticvoid

# Viagem no tempo!



@khaosdoctor



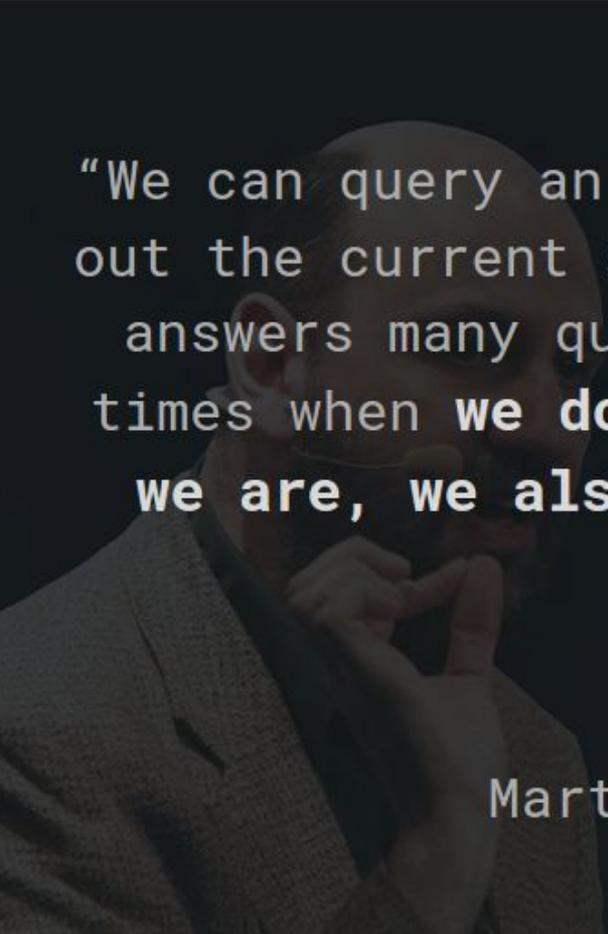
Isantos.me



/khaosdoctor



@\_staticvoid



“We can query an application's state to find out the current state of the world, and this answers many questions. However there are times when **we don't just want to see where we are, we also want to know how we got there.**”

...

Martin Fowler, 2005



@khaosdoctor



lsantos.me



/khaosdoctor



@\_staticvoid

# Um guia simples

- Toda entidade começa com um estado inicial
- Toda entidade começa com um array de eventos
  - Um único evento "criado" deve estar presente
- O estado final é obtido a partir da redução do array de eventos



@khaosdoctor



lsantos.me



/khaosdoctor



@\_staticvoid

Um exemplo prático

# Controle de rotas de navios



@khaosdoctor



lsantos.me



/khaosdoctor



@\_staticvoid

# Controle de rotas de navios



```
Navio:  
  ID: 1bef45c  
  Nome: Black Pearl  
  Events: []  
  Estado: {  
    Loc: null  
  }
```



@khaosdoctor



lsantos.me



/khaosdoctor



@\_staticvoid

# Controle de rotas de navios

Evento:

- ID: 1abc34f
- Nome: criado
- Loc: Santos
- Data: 2018-11-05



Navio:

```
ID: 1bef45c
Nome: Black Pearl
Events: [ criado ]
Estado: {
  Loc: Santos
}
```



@khaosdoctor



lsantos.me



/khaosdoctor



@\_staticvoid

# Controle de rotas de navios



@khaosdoctor



lsantos.me

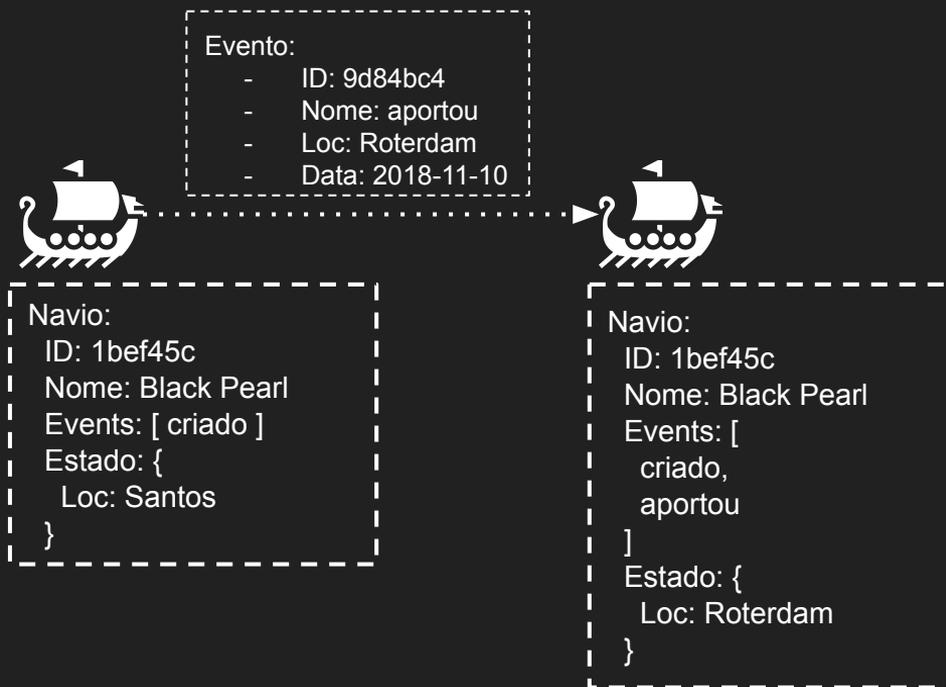


/khaosdoctor



@\_staticvoid

# Controle de rotas de navios



@khaosdoctor



lsantos.me



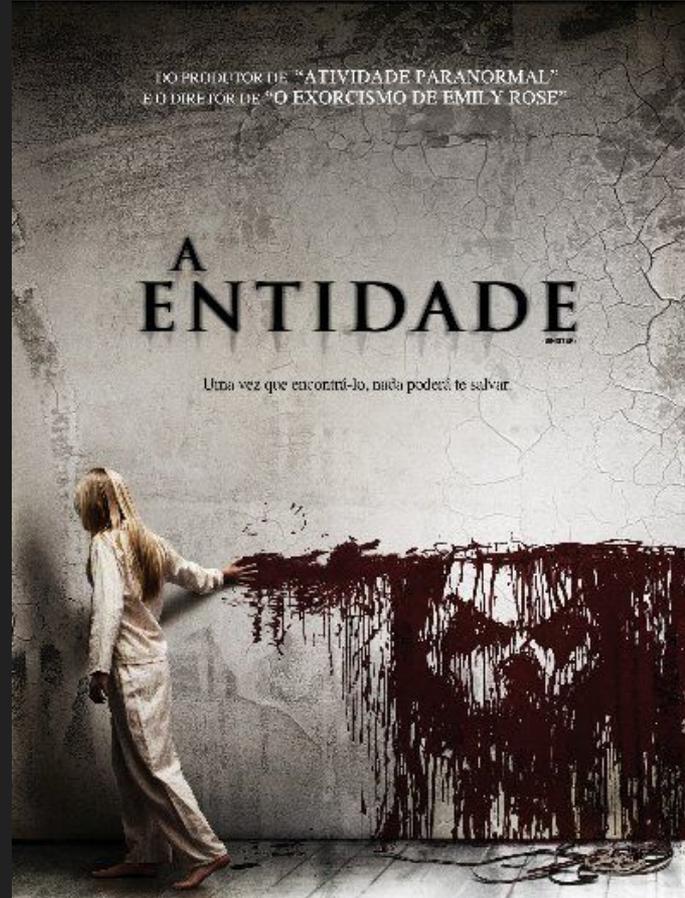
/khaosdoctor



@\_staticvoid

Estado

# Entidade



# Anatomia de uma entidade

- ID
- Propriedades estáticas
- Array de eventos
- Objeto de estado
- Data de criação
- Data da última modificação
- Data da remoção (se aplicável)

```
{
  _id: '1ebcef2427586bef',
  events: [
    {
      id: '4ebcef24bac36bf0',
      name: 'ship-arrived-at-port',
      data: {
        portName: 'Rotterdam'
      },
      timestamp: '2018-11-10T21:45:00',
      user: 'lsantos'
    }
  ],
  state: {
    id: '1ebcef2427586bef',
    name: 'Black Pearl',
    createdAt: '2018-11-05T23:45:00',
    updatedAt: '2018-11-10T21:45:00',
    updatedBy: 'lsantos',
    deletedAt: null
  }
}
```



@khaosdoctor



lsantos.me



/khaosdoctor



@\_staticvoid

Eventos

# Anatomia de um evento

- ID
- Data de criação
- ID do usuário que o criou
- Dados que foram modificados

```
{
  id: '4ebcef24bac36bf0',
  name: 'ship-arrived-at-port',
  data: {
    portName: 'Rotterdam'
  },
  timestamp: '2018-11-10T21:45:00',
  user: 'lsantos'
}
```



@khaosdoctor



lsantos.me



/khaosdoctor



@\_staticvoid

Reducers

# Reducers

Estado  
inicial



@khaosdoctor



lsantos.me

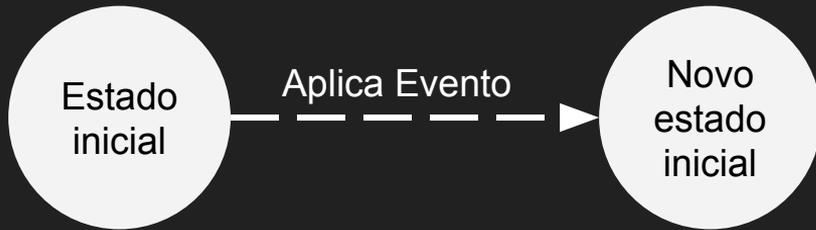


/khaosdoctor



@\_staticvoid

# Reducers



@khaosdoctor



lsantos.me



/khaosdoctor



@\_staticvoid

# Reducers



@khaosdoctor



lsantos.me

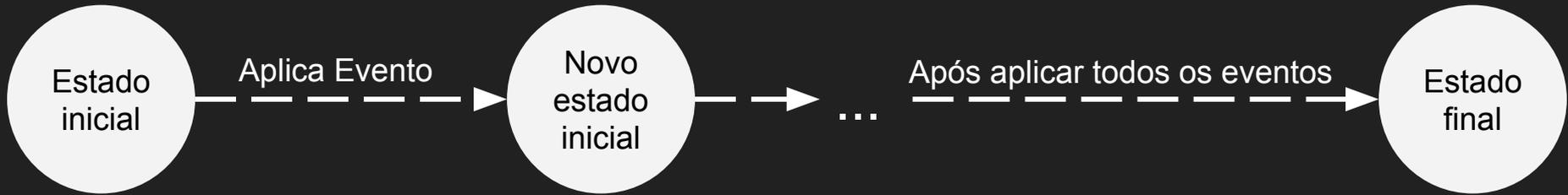


/khaosdoctor



@\_staticvoid

# Reducers



@khaosdoctor



lsantos.me



/khaosdoctor



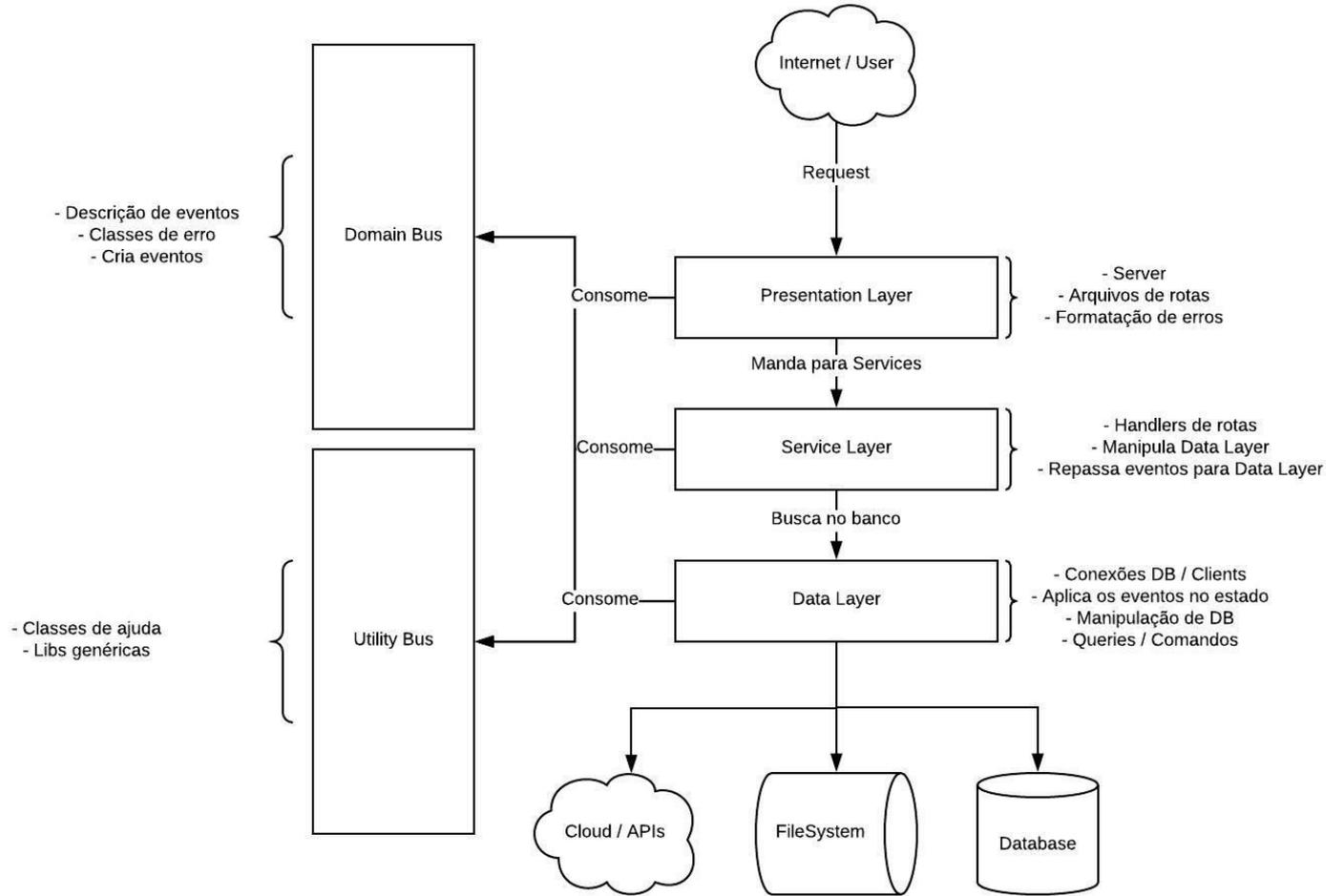
@\_staticvoid

# Arquitetura



D.D.D

Domain Driven Design



@khaosdoctor



Isantos.me



/khaosdoctor



@\_staticvoid

Por que Typescript?



# Por que Typescript?

- Tipos ajudam a saber o retorno de cada função
- Evita erros em runtime
  - Entidades têm estados iniciais e finais bem definidos
  - Eventos têm tipos definidos e precisam seguir uma interface
- Tipagem ajuda a manter conceitos de P.O.O
- Intellisense no VSCode <3



@khaosdoctor



Isantos.me



/khaosdoctor



@\_staticvoid

# Um evento

```
interface IEvent<TData> {
  id: string
  name: string,
  data: TData,
  timestamp: Date
}

export class ShipArrivedEvent<InitialData> implements IEvent<InitialData> {
  readonly id: string
  static readonly name: string = 'ship-arrived-at-port'
  readonly data: TData
  readonly timestamp: Date

  constructor(data: InitialData, user: string) {
    this.id = new ObjectId().toHexString()
    this.data = data
    this.user = user
    this.timestamp = new Date()
  }

  static commit (state: Ship, event: ShipArrivedEvent): Ship {
    state.portName = event.data.portName
    state.updatedAt = event.timestamp
    state.updatedBy = event.user
    return state
  }
}
```

# Uma entidade

```
import ShipArrivedAtPortEvent from './ShipArrivedAtPortEvent'  
import ShipWasCreatedEvent from './ShipWasCreatedEvent'  
  
export default class Ship {  
  name: string | null = null  
  portName: string | null = null  
  updatedAt: Date | null = null  
  updatedBy: string | null = null  
  
  static create(name: string, user: string) {  
    return [  
      new ShipWasCreatedEvent({ name }, user)  
    ]  
  }  
  
  checkInPort (portName: string, user: string) {  
    return [  
      new ShipArrivedAtPortEvent({ portName }, user)  
    ]  
  }  
  
  get state() {  
    return {  
      name: this.name,  
      portName: this.portName,  
      updatedAt: this.updatedAt,  
      updatedBy: this.updatedBy  
    }  
  }  
}
```

# Um reducer

```
import cloneDeep from 'lodash.clonedeep'

interface ICommitFunction<Entity, Event> {
  (state: Entity, event: Event): Entity
}

interface IEvent<TData> {
  id: string
  name: string,
  data: TData,
  timestamp: Date
}

export class Reducer<EntityInterface> {
  private knownEvents: { [ key: string ]: ICommitFunction<EntityInterface, any> }

  constructor(knownEvents: { [ key: string ]: ICommitFunction<EntityInterface, any> }) {
    this.knownEvents = knownEvents
  }

  reduce(state: EntityInterface, events: IEvent<any>[]): EntityInterface {
    return events.reduce<EntityInterface>((state: EntityInterface, event: IEvent<any>) => {
      const clonedState = cloneDeep(state)
      return this.knownEvents[ event.name ](clonedState, event)
    }, state)
  }
}
```



```
import { Reducer } from './reducer'  
import Ship from './ship'  
import { ShipWasCreatedEvent, ShipArrivedEvent } from './events'  
  
const shipReducer = new Reducer<Ship>({  
  [ShipWasCreatedEvent.name]: ShipWasCreatedEvent.commit,  
  [ShipArrivedEvent.name]: ShipArrivedEvent.commit  
})
```



@khaosdoctor



Isantos.me



/khaosdoctor



@\_staticvoid

```
const events = [
  ...Ship.create('Black Pearl', 'lsantos'),
  ...Ship.checkInPort('Santos', 'lsantos')
]
/*
[
  ShipWasCreatedEvent {
    id: 'd043fc61a0ab5e088a458b6b',
    name: 'ship-was-created',
    data: { name: 'Black Pearl' },
    timestamp: 2018-10-16T22:10:06.022Z,
    user: 'lsantos'
  },
  ShipArrivedEvent {
    id: 'a2707db386e25529e25ee882',
    name: 'ship-arrived-at-port',
    data: { portName: 'Santos' },
    timestamp: 2018-10-16T20:54:57.122Z,
    user: 'lsantos'
  }
]
*/
```



@khaosdoctor



lsantos.me



/khaosdoctor



@\_staticvoid



```
const ship = shipReducer.reduce(new Ship(), events)
console.log(ship.state)
/*
{
  name: 'Black Pearl',
  portName: 'Santos',
  updatedBy: 'lsantos',
  updatedAt: 2018-10-16T20:54:57.122Z
}
*/
```



@khaosdoctor



lsantos.me

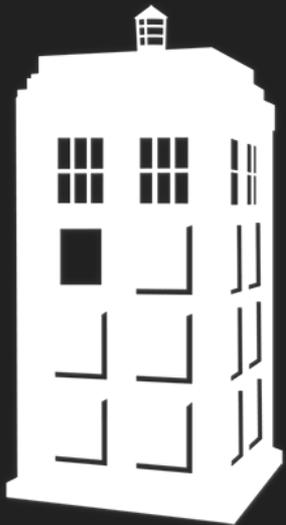


/khaosdoctor



@\_staticvoid

Muito complicado...



# TARDIS

<https://github.com/nxcd/tardis>



# PARADOX

<https://github.com/nxcd/paradox>



@khaosdoctor



Isantos.me



/khaosdoctor



@\_staticvoid

Time to code!



Reducer?



```
import { Reducer } from '@nxcd/tardis'
import Ship from './Person'
import ShipWasCreated from './PersonWasCreatedEvent'

const shipReducer = new Reducer<Ship>({
  [ShipWasCreated.eventName]: ShipWasCreated.commit
})

const events = [ ...Ship.create('Black Pearl', 'lsantos') ]

const ship = shipReducer.reduce(new Ship(), events)
console.log(ship.state)
/*
{
  name: 'Black Pearl',
  portName: 'Santos',
  updatedBy: 'lsantos',
  updatedAt: 2018-10-16T20:54:57.122Z
}
*/
```



@khaosdoctor



lsantos.me



/khaosdoctor



@\_staticvoid

E para salvar isso tudo?

```
{
  "_id" : ObjectId("5bd37ca5bfde5b0680325d59"),
  "events" : [{
    "id" : "5bd37ca5e94b6c0680d31b05",
    "name" : "production-order-was-created",
    "data" : {
      "id" : ObjectId("5bd37ca5bfde5b0680325d59"),
      "createdAt" : ISODate("2018-11-20T00:00:00.000-02:00"),
      "product" : {
        "id" : "PRODUTOXPPTO",
        "description" : "Rolo de plástico",
        "conversionType" : "D"
      }
    },
  },
  {
    "timestamp" : ISODate("2018-10-26T17:44:21.282-03:00"),
    "user" : "lsantos"
  }
],
  {
    "id" : "5bd37d1be94b6c0680d31b06",
    "name" : "production-order-status-was-updated",
    "data" : {
      "status" : "producing"
    },
    "timestamp" : ISODate("2018-10-26T17:46:19.494-03:00"),
    "user" : "lsantos"
  }
],
  "state" : {
    "id" : ObjectId("5bd37ca5bfde5b0680325d59"),
    "product" : {
      "parentProduct" : null,
      "id" : "PRODUTOXPPTO",
      "description" : "Rolo de plástico",
      "conversionFactor" : NumberInt("1000")
    },
    "producedAmount" : NumberInt("0"),
    "createdAt" : ISODate("2018-11-20T00:00:00.000-02:00"),
    "status" : "producing",
    "updatedAt" : ISODate("2018-10-26T17:46:19.494-03:00"),
    "updatedBy" : "lsantos",
    "deletedAt" : null
  }
}
```

Eventos

```
{
  "_id" : ObjectId("5bd37ca5bfde5b0680325d59"),
  "events" : [{
    "id" : "5bd37ca5e94b6c0680d31b05",
    "name" : "production-order-was-created",
    "data" : {
      "id" : ObjectId("5bd37ca5bfde5b0680325d59"),
      "createdAt" : ISODate("2018-11-20T00:00:00.000-02:00"),
      "product" : {
        "id" : "PRODUTOXPTO",
        "description" : "Rolo de plástico",
        "conversionType" : "D"
      }
    },
    "timestamp" : ISODate("2018-10-26T17:44:21.282-03:00"),
    "user" : "lsantos"
  },
  {
    "id" : "5bd37d1be94b6c0680d31b06",
    "name" : "production-order-status-was-updated",
    "data" : {
      "status" : "producing"
    },
    "timestamp" : ISODate("2018-10-26T17:46:19.494-03:00"),
    "user" : "lsantos"
  }
  ],
  "state" : {
    "id" : ObjectId("5bd37ca5bfde5b0680325d59"),
    "product" : {
      "parentProduct" : null,
      "id" : "PRODUTOXPTO",
      "description" : "Rolo de plástico",
      "conversionFactor" : NumberInt("1000")
    },
    "producedAmount" : NumberInt("0"),
    "createdAt" : ISODate("2018-11-20T00:00:00.000-02:00"),
    "status" : "producing",
    "updatedAt" : ISODate("2018-10-26T17:46:19.494-03:00"),
    "updatedBy" : "lsantos",
    "deletedAt" : null
  }
}
```

Estado

# Nada é de graça

## Prós:

- É possível saber, exatamente, como uma entidade estava em qualquer ponto do tempo
- Logs nativos da implementação
- D.D.D força atenção ao comportamento ao invés da implementação
- Event Streams, por só permitirem inserção, são muito rápidas
- Desenvolvimento utilizando DDD se torna muito rápido após o primeiro "set" estar pronto

## Contras:

- Falta de ferramentas com vasto tempo de desenvolvimento
- Não existe muita teoria sobre o assunto
- Custo de implementação
- Custo de inovação, Event Sourcing é ainda muito novo
- Pode ser complicado à primeira vista
- Limite (ajustável) de 15Mb para documentos no Mongo (1 evento pesa +/- 723B)
- Entidades com muitos eventos podem causar dois problemas (contornáveis)
  - Estouro de tamanho de documento
  - Demora no reduce



@khaosdoctor



Isantos.me



/khaosdoctor



@\_staticvoid

## <refs>

- <https://martinfowler.com/eaaDev/EventSourcing.html>
- <https://medium.com/@gabrielqueiroz/vamos-falar-sobre-event-sourcing-276ae66106f7>
- <https://microservices.io/patterns/data/event-sourcing.html>
- <https://github.com/nxcd/tardis>



@khaosdoctor



Isantos.me



/khaosdoctor



@\_staticvoid



# T. HANKS



/khaosdoctor



Isantos.me



@khaosdoctor



@\_staticvoid