Proxy

Um proxy é capaz de interceptar diversos tipos de operações em um objeto ou função definido como alvo. É útil para realizar processos de auditoria e performance, sem a necessidade de misturar regras de negócio.

```
1.  let book = {
2.    title: "Domain-Drive Design",
3.    author: "Eric Evans",
4.    pages: 450
5.  };
```

Criando um objeto alvo

```
1.  let book = {
2.    title: "Domain-Drive Design",
3.    author: "Eric Evans",
4.    pages: 450
5.  };
6.  let bookProxy = new Proxy(book);
```

Criando um proxy em torno do objeto alvo

```
1.  let book = {
2.    title: "Domain-Drive Design",
3.    author: "Eric Evans",
4.    pages: 450
5.  };
6.  let bookProxy = new Proxy(book, {
7.  });
```

Definindo um handler para o proxy

Traps

constructor: disparado ao invocar o construtor do objeto
get: disparado ao consultar uma determinada propriedade do objeto
set: disparado ao definir uma determinada propriedade no objeto

Outras traps: apply, defineProperty, deleteProperty, enumerate, getOwnProperty, getPrototypeOf, has, isExtensible, ownKeys, setPrototypeOf

```
1.  let book = {
2.    title: "Domain-Drive Design",
3.    author: "Eric Evans",
4.    pages: 450
5.  };
6.  let bookProxy = new Proxy(book, {
7.    get(target, propertyName) {
8.      return target[propertyName];
9.    }
10. });
```

Interceptando a consulta de uma propriedade do objeto alvo

```
1.  let book = {
2.    title: "Domain-Drive Design",
3.    author: "Eric Evans",
4.    pages: 450
5.  };
6.  let bookProxy = new Proxy(book, {
7.    set(target, propertyName, value) {
8.      return target[propertyName] = value;
9.    }
10. });
```

Interceptando a definição de uma propriedade do objeto alvo


Reflect

Proporciona uma forma de executar diversos tipos de operações em um objeto alvo.

Existem diversas operações que podem ser utilizadas para interagir com um objeto ou uma função:

constructor(target, args): invoca o construtor do objeto
get(target, propertyName):consultar uma determinada propriedade do objeto
set(target, propertyName, value): define uma determinada propriedade no objeto
apply(target, this, args): invoca uma função

Outras armadilhas: defineProperty, deleteProperty, enumerate, getOwnProperty, getPrototypeOf, has, isExtensible, ownKeys, setPrototypeOf

```
1.  let book = {
2.    title: "Domain-Drive Design",
3.    author: "Eric Evans",
4.    pages: 450
5.  };
6.  let bookProxy = new Proxy(book, {
7.    get(target, propertyName) {
8.      return Reflect.get(target, propertyName);
9.    }
10. });
```

Interceptando a consulta de uma propriedade do objeto alvo

```
1.  let book = {
2.    title: "Domain-Drive Design",
3.    author: "Eric Evans",
4.    pages: 450
5.  };
6.  let bookProxy = new Proxy(book, {
7.   set(target, propertyName, value) {
8.    return Reflect.set(target, propertyName, value);
9.   }
10. });
```

Interceptando a definição de uma propriedade do objeto alvo