

ZERO To HERO

NODE.JS ASYNCHRONOUS





CLAUDINEY JUNIOR

- ARQUITETO DE SOLUÇÕES @ BANCO ORIGINAL
- COMMUNITY MANAGER @ CAQUICODERS
- LISTED SPEAKER @ NODEJS.ORG
- NERD 🧐

PROCESSO SÍNCRONO



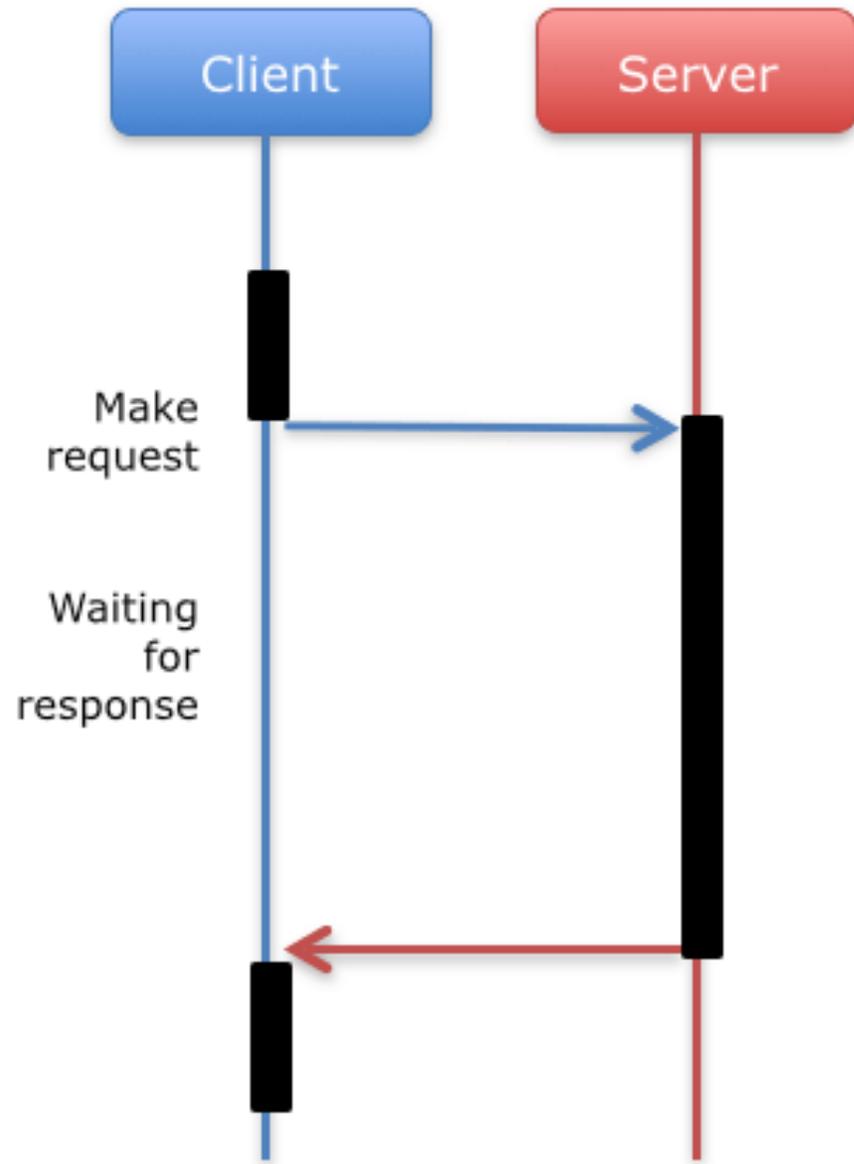
Processamento **síncrono**

+ **Requisições**

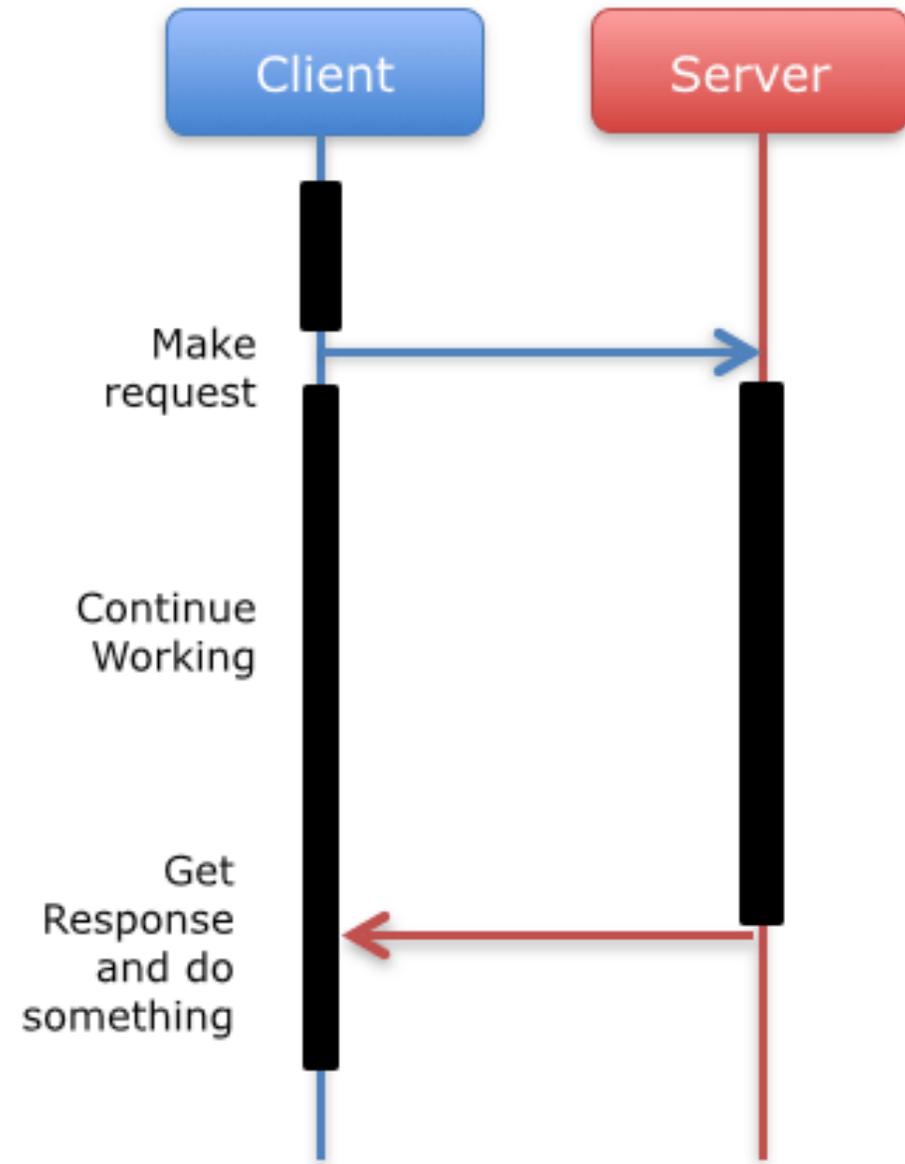


+ tempo de processador **ocioso**

Synchronous

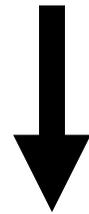


Asynchronous



Processamento **síncrono**

+ tempo de processador **ocioso**



+ **hardware**

Processamento síncrono

+ hardware



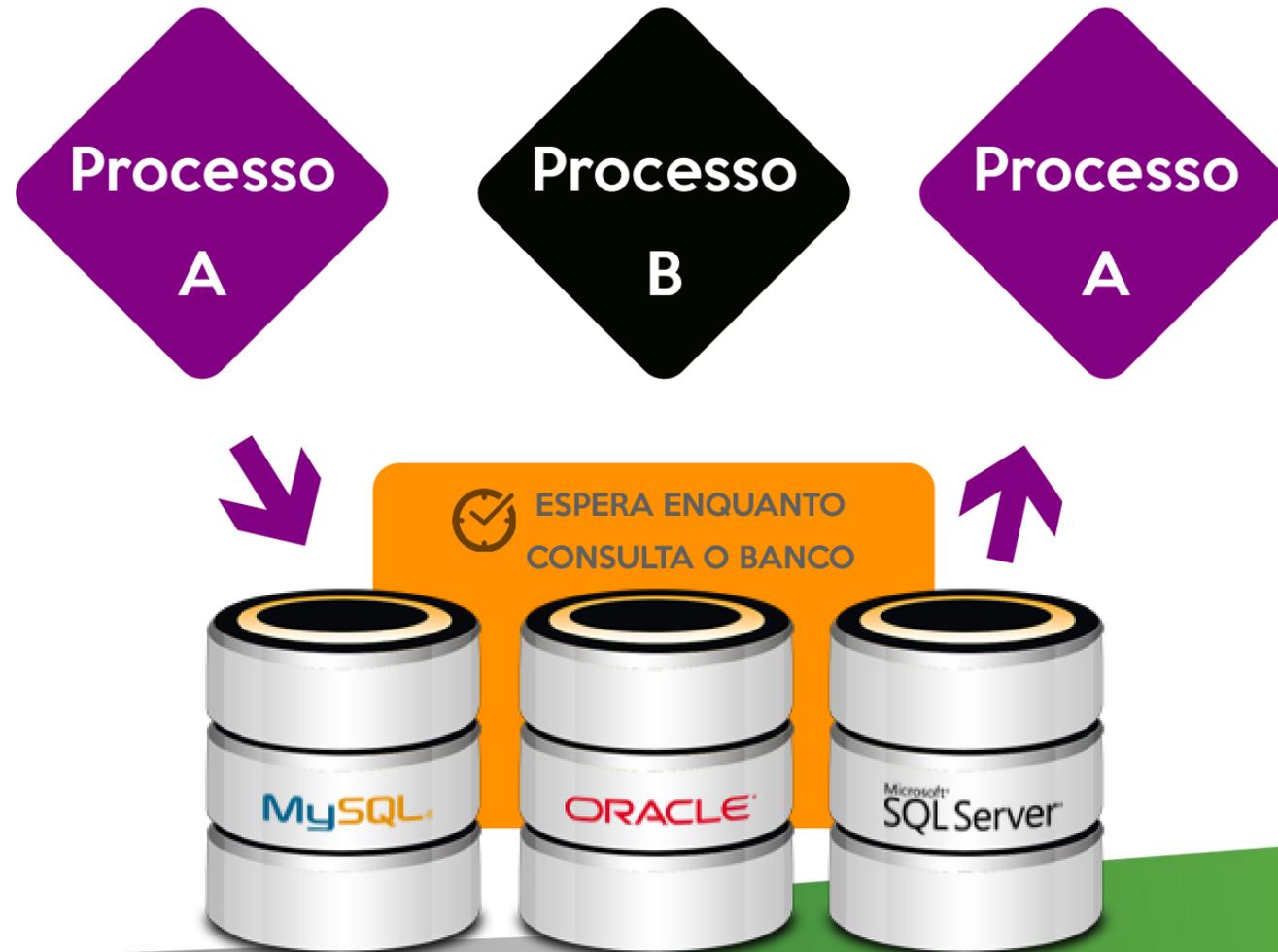
+ \$\$\$\$\$\$\$\$\$\$

A meme featuring a close-up of a man with glasses and a disapproving expression, likely from the TV show 'The Office'. The text is overlaid on the image in a bold, white, sans-serif font with a black outline.

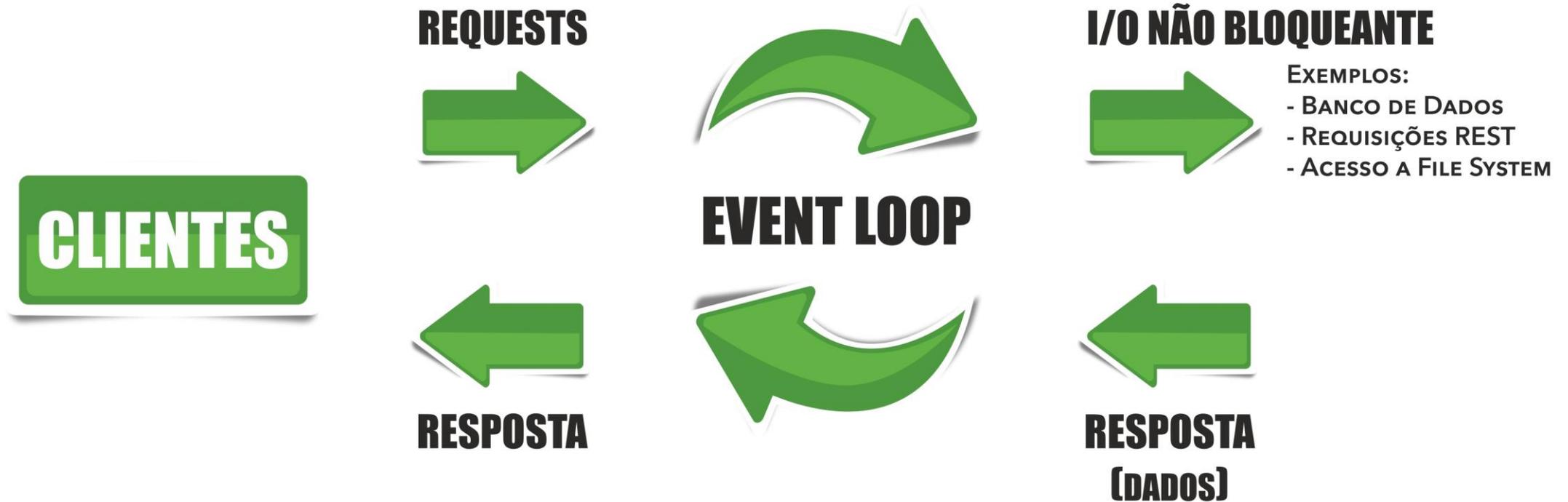
TIME IS MONEY

FALSE. MONEY IS MONEY

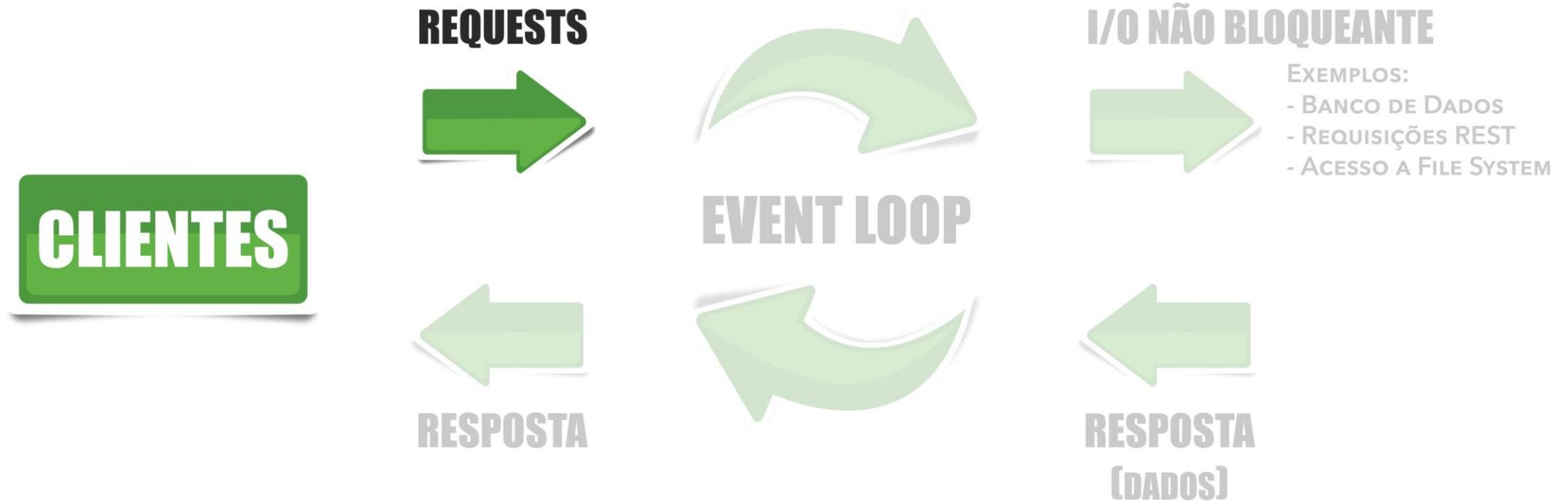
PROCESSO ASSÍNCRONO



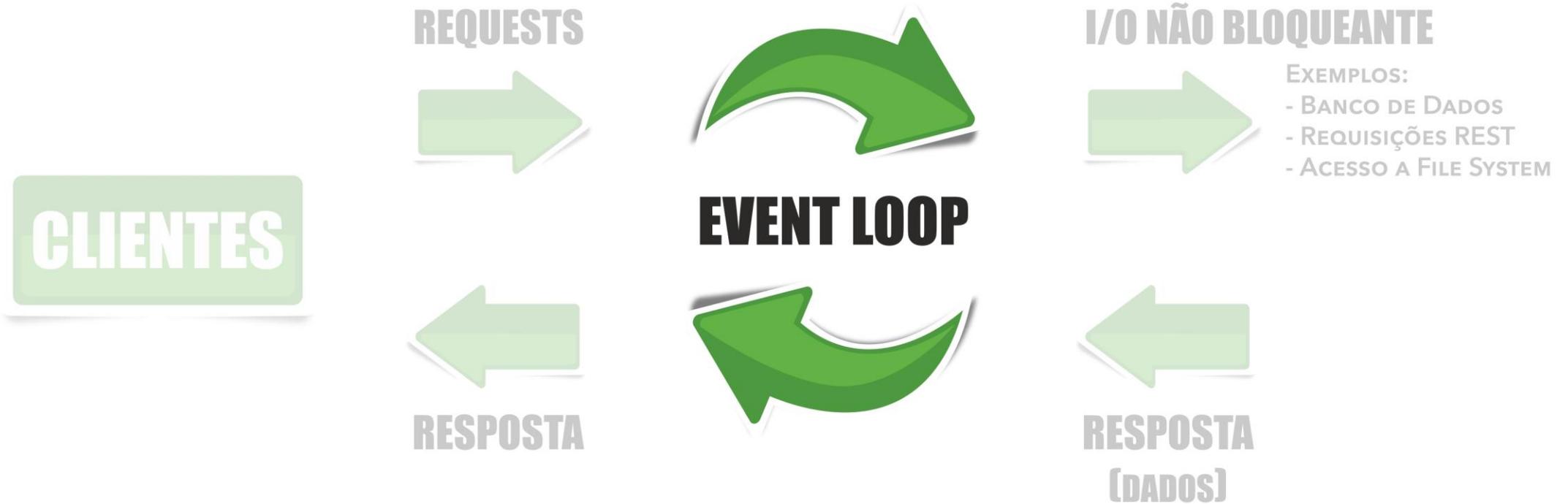
Event Loop



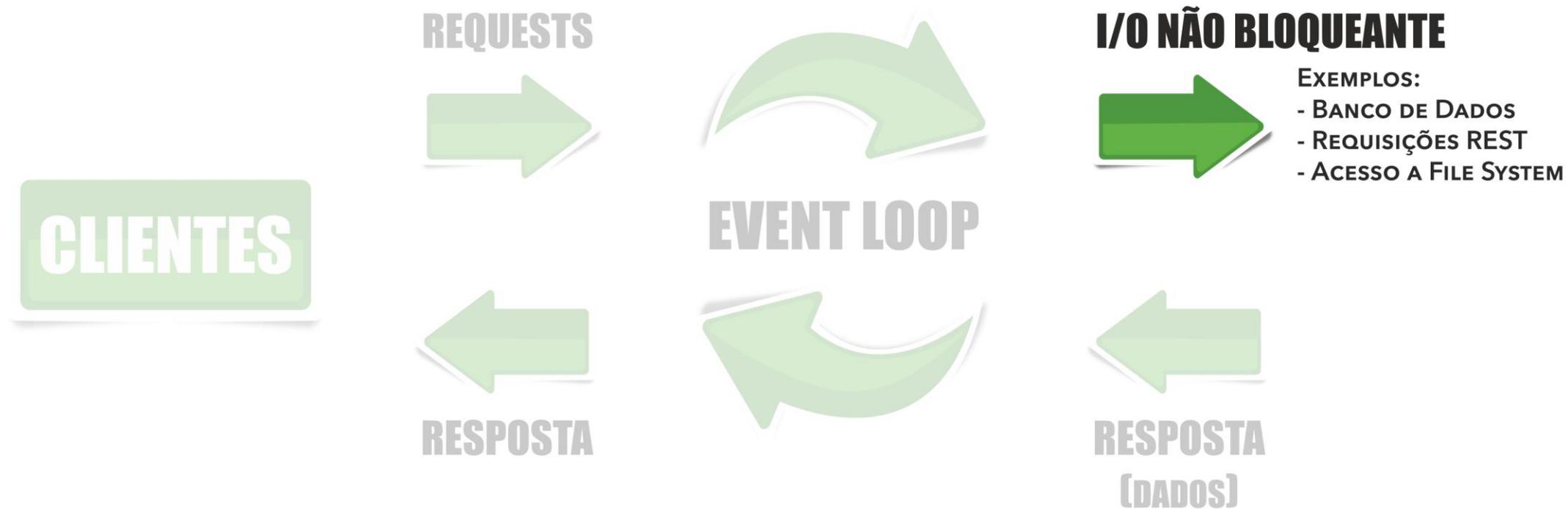
Event Loop



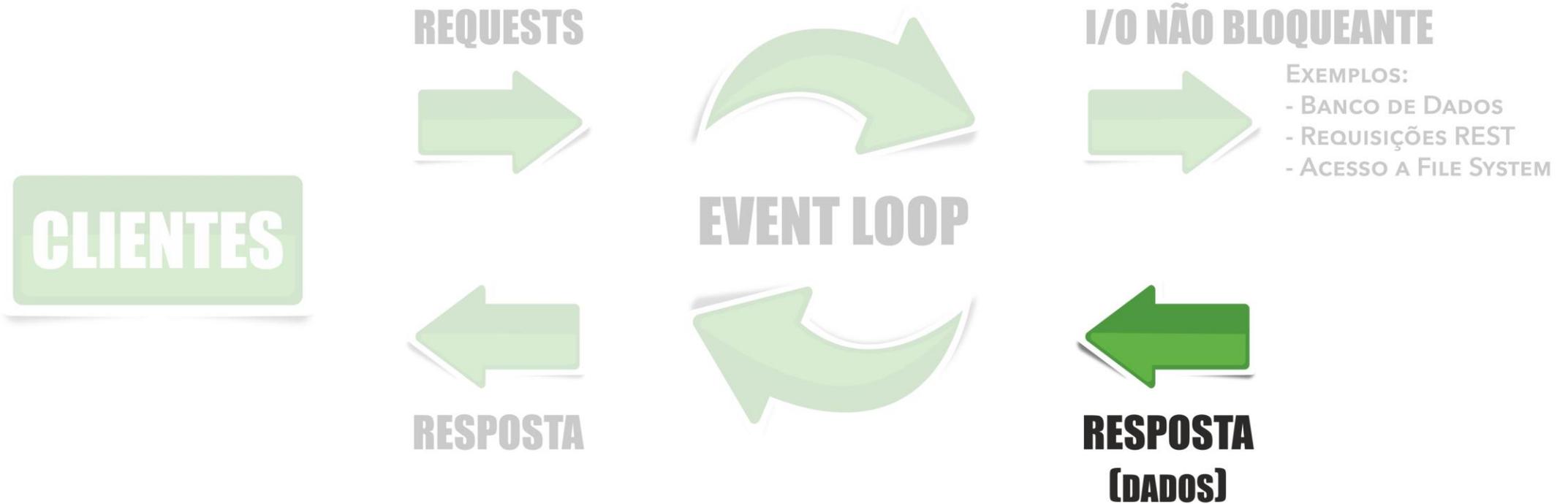
Event Loop



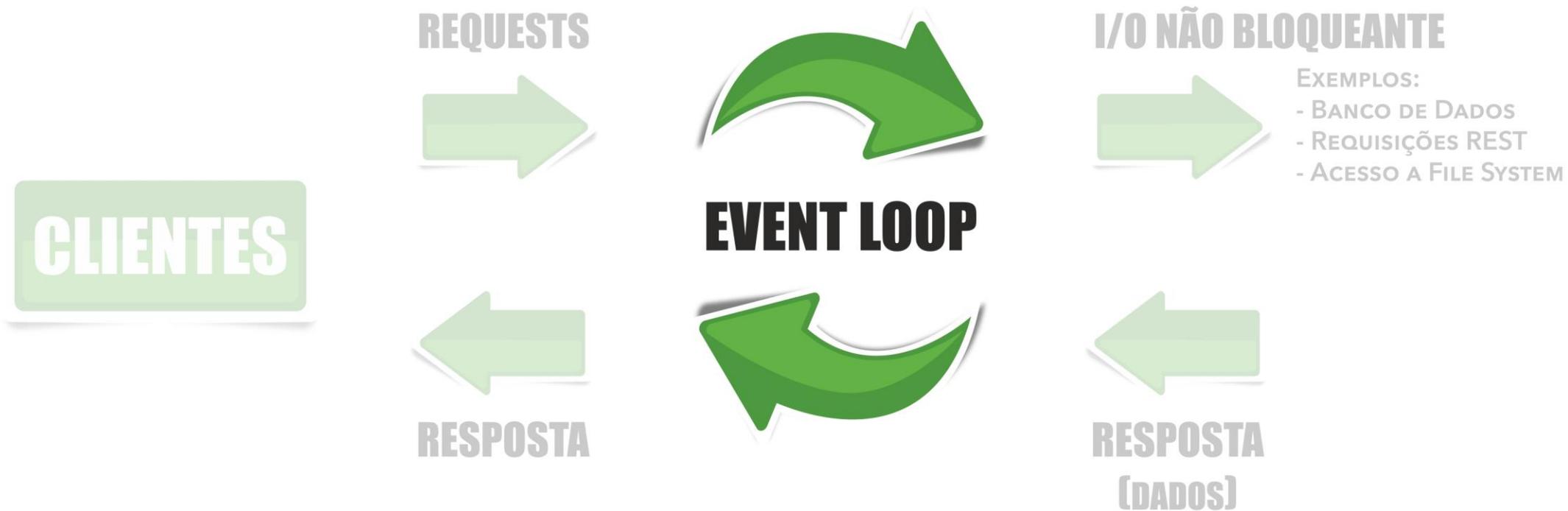
Event Loop



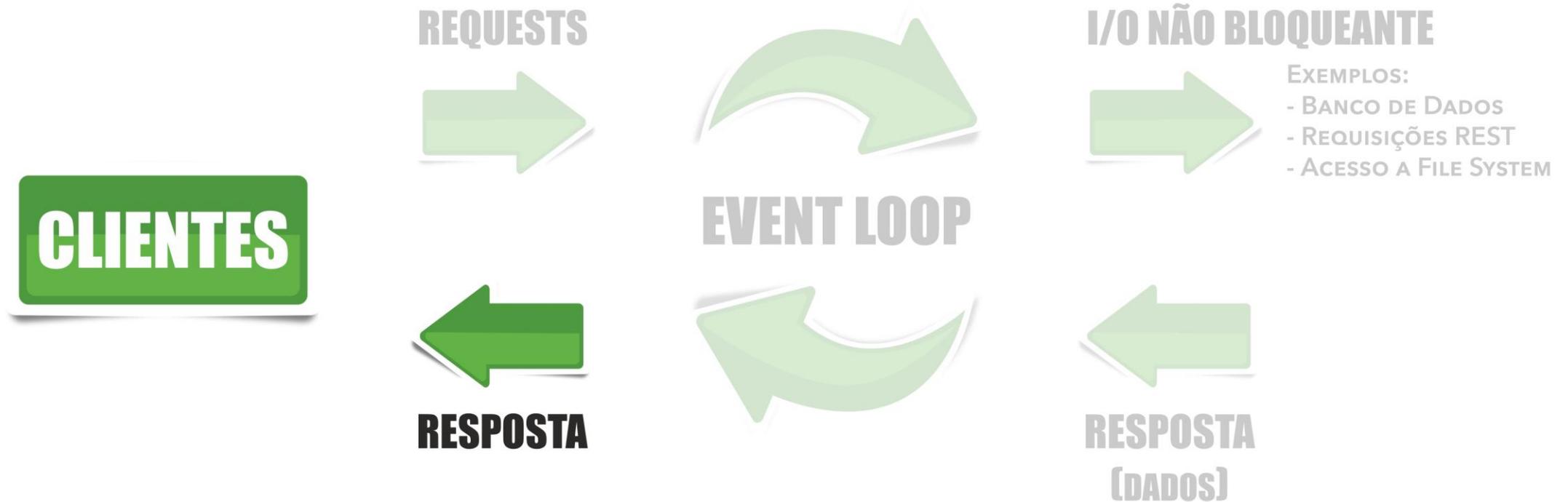
Event Loop



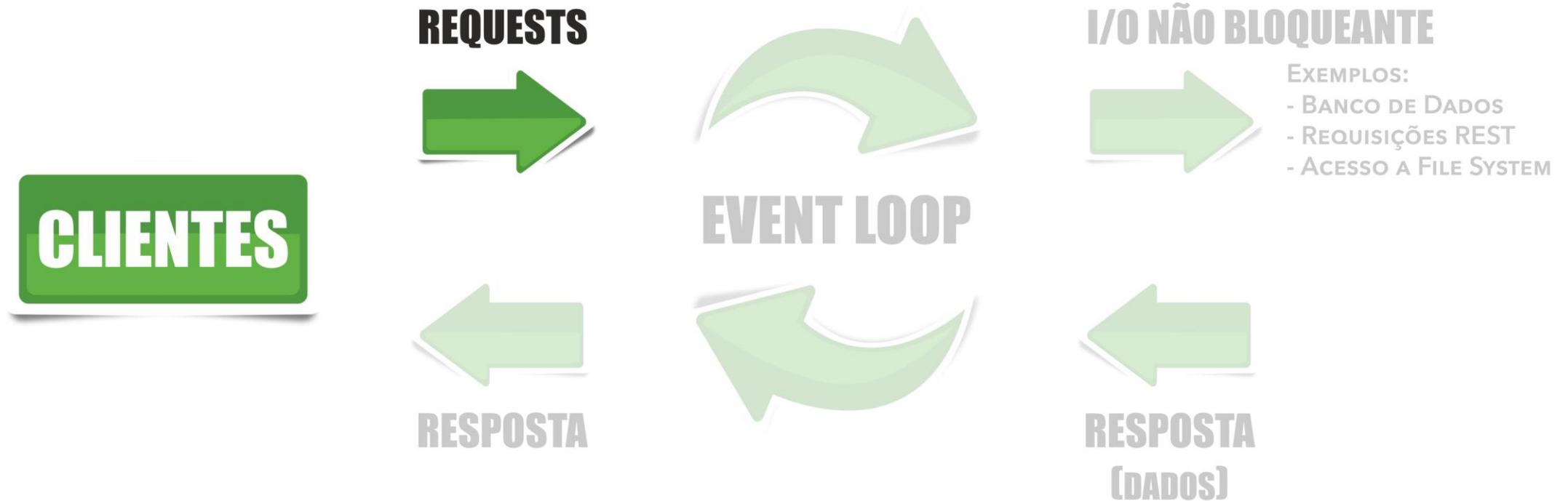
Event Loop



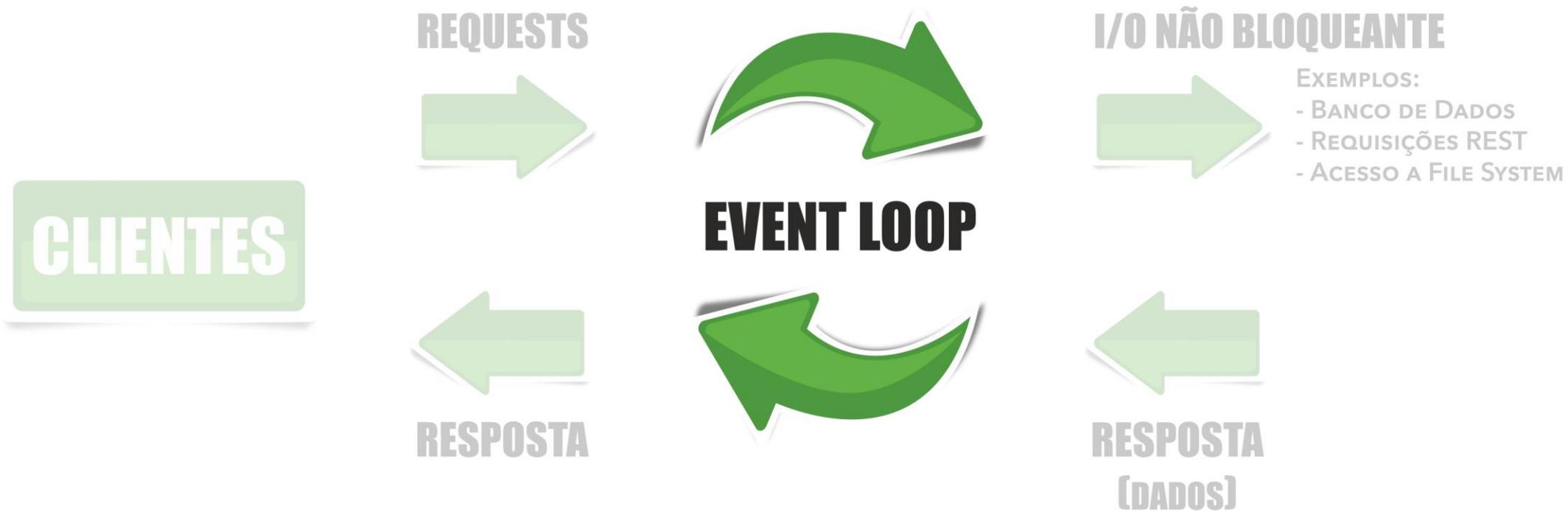
Event Loop



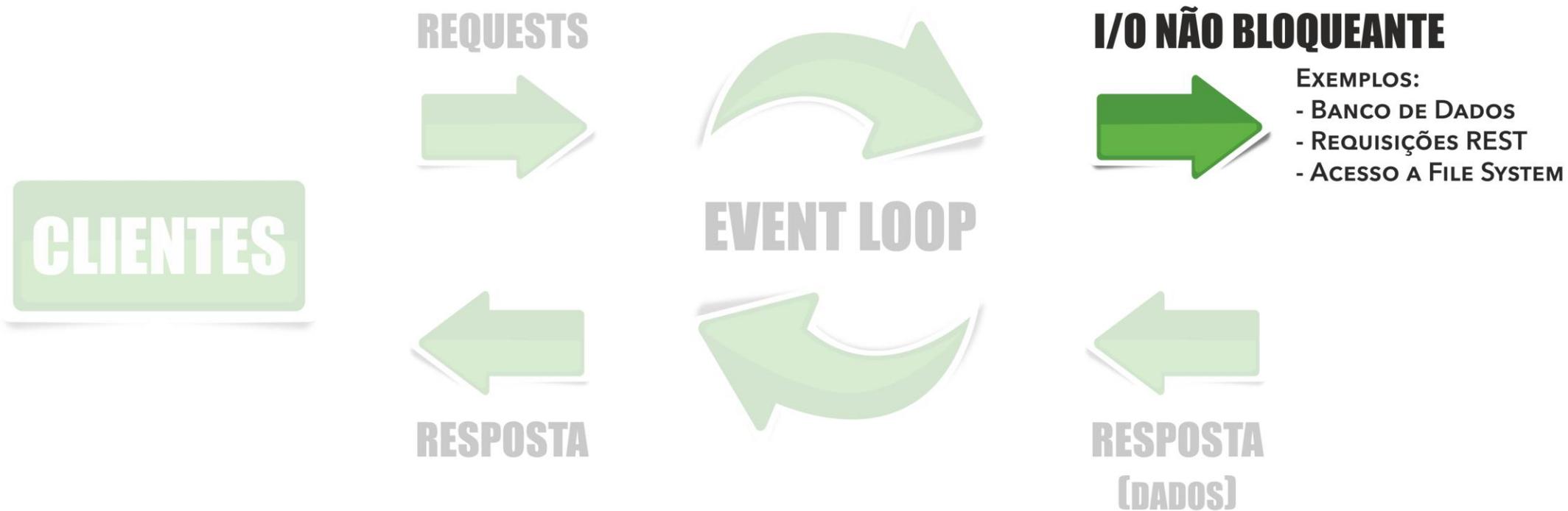
Event Loop



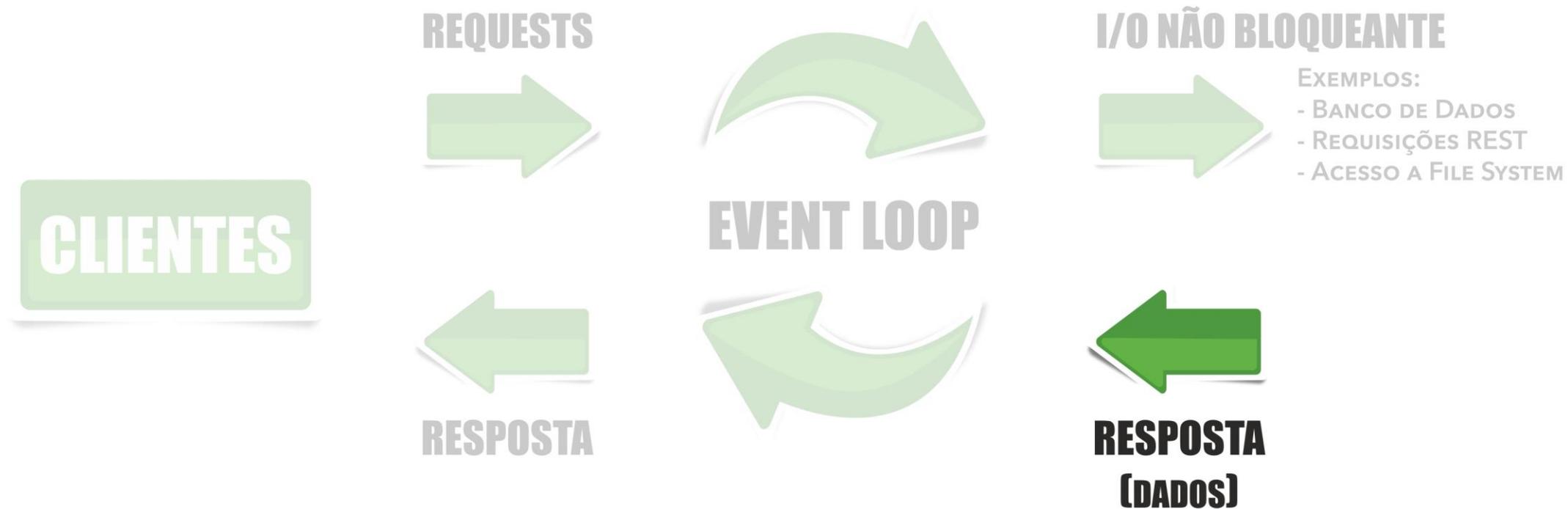
Event Loop



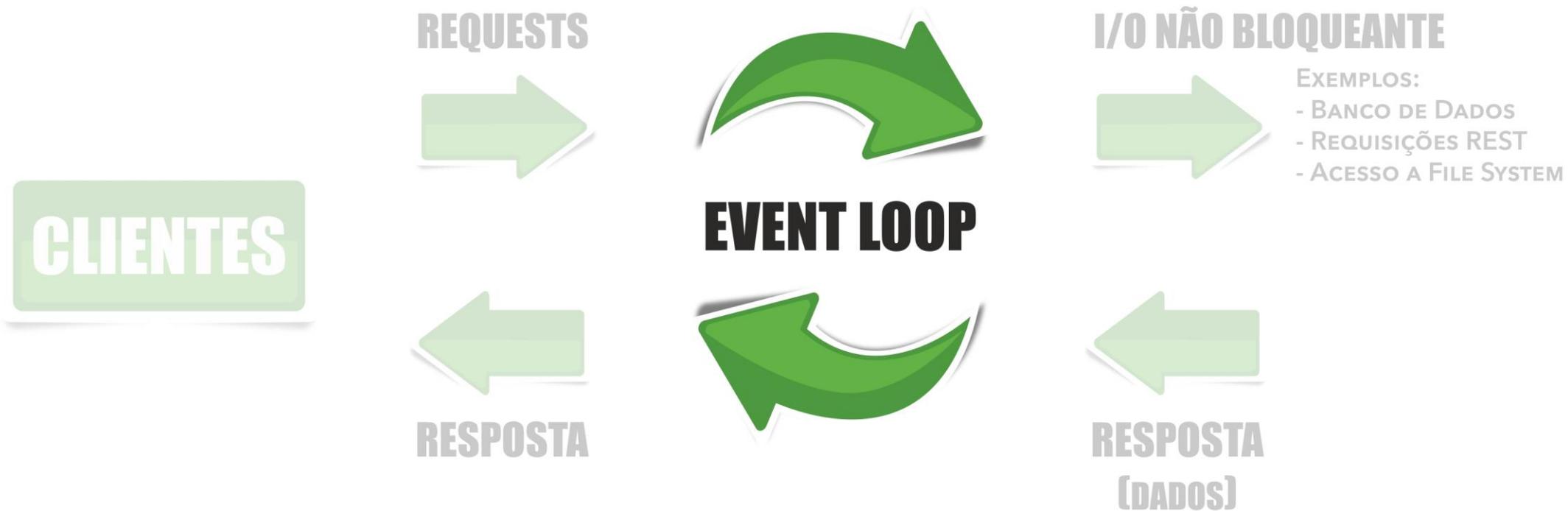
Event Loop



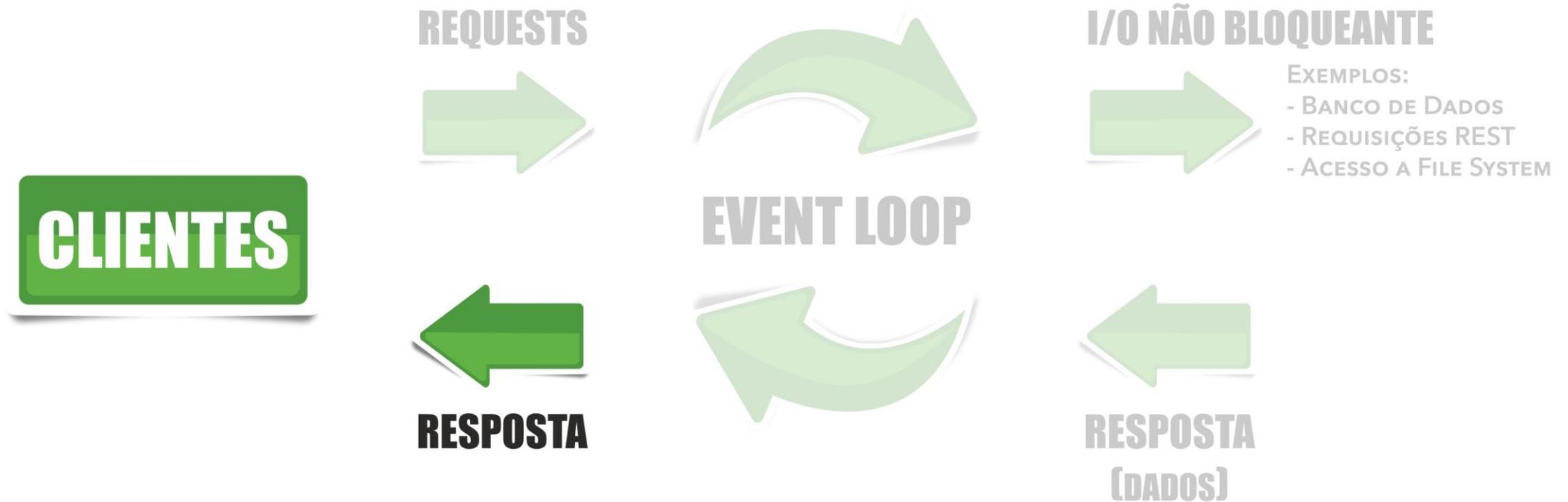
Event Loop



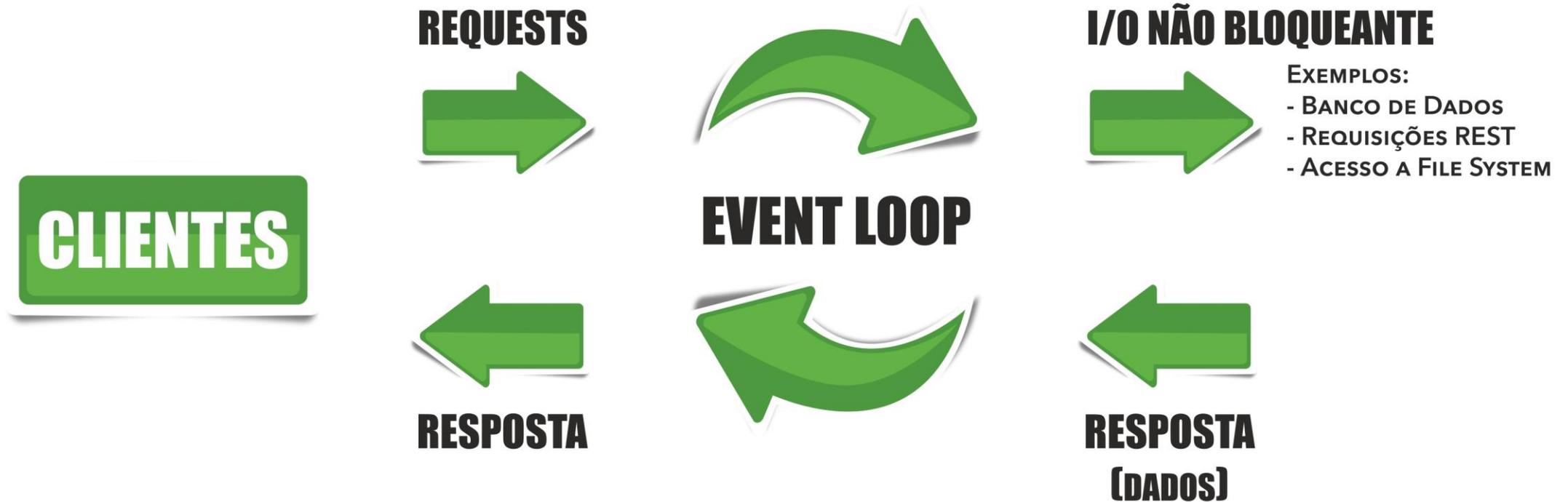
Event Loop



Event Loop



Event Loop



Javascript **Functions**

First class **objects**

Têm **propriedades e métodos**

Diferença: Elas podem ser **invocadas!!!**



Javascript **Functions**

First class **objects**

Têm **propriedades e métodos**

Diferença: Elas podem ser **invocadas!!!**

Javascript **Functions**

PODEM SER PASSADAS COMO **PARÂMETRO** PARA
OUTRAS FUNÇÕES



Callback Functions



```
myFunction(arg, function(error, result) {  
    if(error) return false;  
    doSomething(result);  
});
```

Callback hell

```
1 function hell(win) {
2   // for listener purpose
3   return function() {
4     loadLink(win, REMOTE_SRC+'/assets/css/style.css', function() {
5       loadLink(win, REMOTE_SRC+'/lib/async.js', function() {
6         loadLink(win, REMOTE_SRC+'/lib/easyXDM.js', function() {
7           loadLink(win, REMOTE_SRC+'/lib/json2.js', function() {
8             loadLink(win, REMOTE_SRC+'/lib/underscore.min.js', function() {
9               loadLink(win, REMOTE_SRC+'/lib/backbone.min.js', function() {
10                loadLink(win, REMOTE_SRC+'/dev/base_dev.js', function() {
11                  loadLink(win, REMOTE_SRC+'/assets/js/deps.js', function() {
12                    loadLink(win, REMOTE_SRC+'/src/' + win.loader_path + '/loader.js', function() {
13                      async.eachSeries(SCRIPTS, function(src, callback) {
14                        loadScript(win, BASE_URL+src, callback);
15                      });
16                    });
17                  });
18                });
19              });
20            });
21          });
22        });
23      });
24    });
25  };
26 }
```

A character in a white martial arts gi is shown in a dynamic pose, performing a blue energy attack. The character is positioned on the left side of the image, with their right arm extended forward, emitting a bright blue, glowing sphere of energy. The background is dark, making the character and the energy attack stand out.

Promise

Representação de um **valor** que pode estar **disponível agora, no futuro ou nunca.**

**MINHA MÃE FALAVA
"NA VOLTA A GENTE COMPRA"**



E TO ESPERANDO ATÉ HOJE

4 estados de uma promise

- **Pending**
- **Fulfilled**
- **Rejected**
- **Settled**



#Estrutura

```
new Promise((resolve, reject) => {  
  if(error) reject('Deu erro')  
  resolve('Deu certo');  
})
```

Uso

```
Promise()  
  .then(resultado => console.log(resultado))  
  .catch(error => console.error(error));
```

async/await

Usado para tratar de **promises**



```
async function funcaoAssincrona() {  
    const resultado = await doSomething()  
};
```

```
# Precisa executar  
funcaoAssincrona();
```

Dúvidas?



/in/claudiney-junior



claudiney.info@gmail.com



/claudineyjuniior

Inscreva-se



ATÉ A PRÓXIMA!