

# APLICAÇÕES INFORMÁTICAS EM ENGENHARIA BIOMÉDICA

Application  
Development



# Full Stack Development

## Development

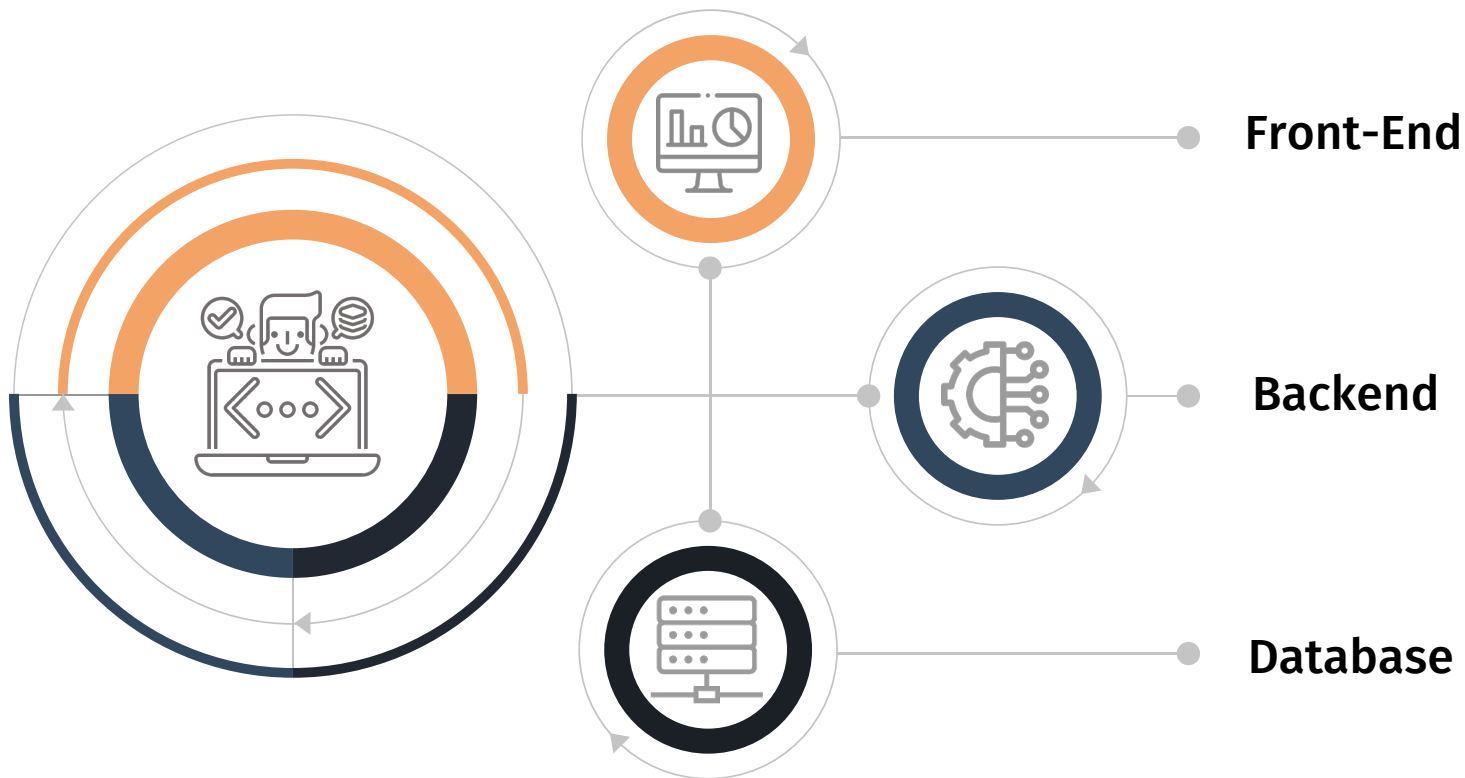
It refers to the front end (client side) and back end (server side) development of web application.



## Developers

They have the ability to design complete web application and websites. They work on the frontend, backend, database and debugging of web application or websites.

# Components



# FRONT-END

It is the visible part of website or web application which is responsible for user experience. The user directly interacts with the front end portion of the web application or website. The front end portion can be built by using some languages:



**HTML**



**CSS**



**Javascript**



1

HTML stands for Hypertext Markup Language

2

HTML elements are used to define the structure and content of a web page

3

HTML code consists of elements, which are enclosed in angle brackets (< >)

4

Some common HTML elements include headings (<h1> to <h6>), paragraphs (<p>), links (<a>), images (<img>), and lists (<ul> and <ol>)

5

Tags can also include attributes, which provide additional information about an element (e.g. the src attribute specifies the source URL of an image)

# HTML



```
<!DOCTYPE html>
<html>
  <head>
    <title>My Web Page</title>
  </head>
  <body>
    <h1>Welcome to my Web
Page!</h1>
    <p>This is an example of a
paragraph.</p>
    <a
href="https://www.example.com">Click
here</a> to visit another website.
  </body>
</html>
```

## Welcome to my Web Page!

This is an example of a paragraph.

[Click here](https://www.example.com) to visit another website.



1 CSS stands for Cascading Style Sheets

2 It is used to style and format web pages written in HTML

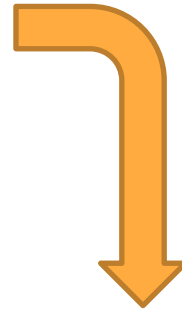
3 Selectors are used to target specific HTML elements, while declarations are used to specify how the targeted elements should be styled

4 CSS can be used to change the color, font, layout, and other visual aspects of a web page

5 CSS can be included in an HTML document using the `<style>` element, or in a separate file using the `<link>` element

# CSS

```
<head>
  <title>My Web Page</title>
  <style>
    h1 {
      color: blue;
      font-size: 36px;
      text-align: center;
    }
    p {
      color: gray;
      font-size: 24px;
      text-align: justify;
    }
    a {
      color: red;
      text-decoration: none;
    }
  </style>
</head>
...
```



**Welcome to my Web Page!**

This is an example of a paragraph.

[Click here](#) to visit another website.



# CSS Frameworks



## Bootstrap

Spotify, Twitter,  
LinkedIn



## Foundation

Mozilla, Disney,  
National  
Geographic



## Materialize

Samsung, Lyft,  
Shopify, The  
Guardian



## Bulma

BMW, Doodle,  
Algolia



## Semantic UI

Amazon Web  
Services, Microsoft  
Teams, Expedia

CSS frameworks are pre-designed libraries of CSS code that can be used to style web pages quickly and easily. Frameworks typically include a set of predefined styles for common HTML elements, such as buttons, forms, and navigation menus.



1

JavaScript is a programming language that is primarily used to add interactivity and dynamic behavior to web pages.

2

JavaScript is a client-side language, meaning it runs in the user's web browser rather than on the server.

3

It is used for a variety of purposes, such as creating interactive forms, animating graphics, and manipulating web page content.

4

It is supported by all major web browsers and can be used on both desktop and mobile devices.

5

JavaScript is often used in conjunction with HTML and CSS to create dynamic, responsive websites and web applications.

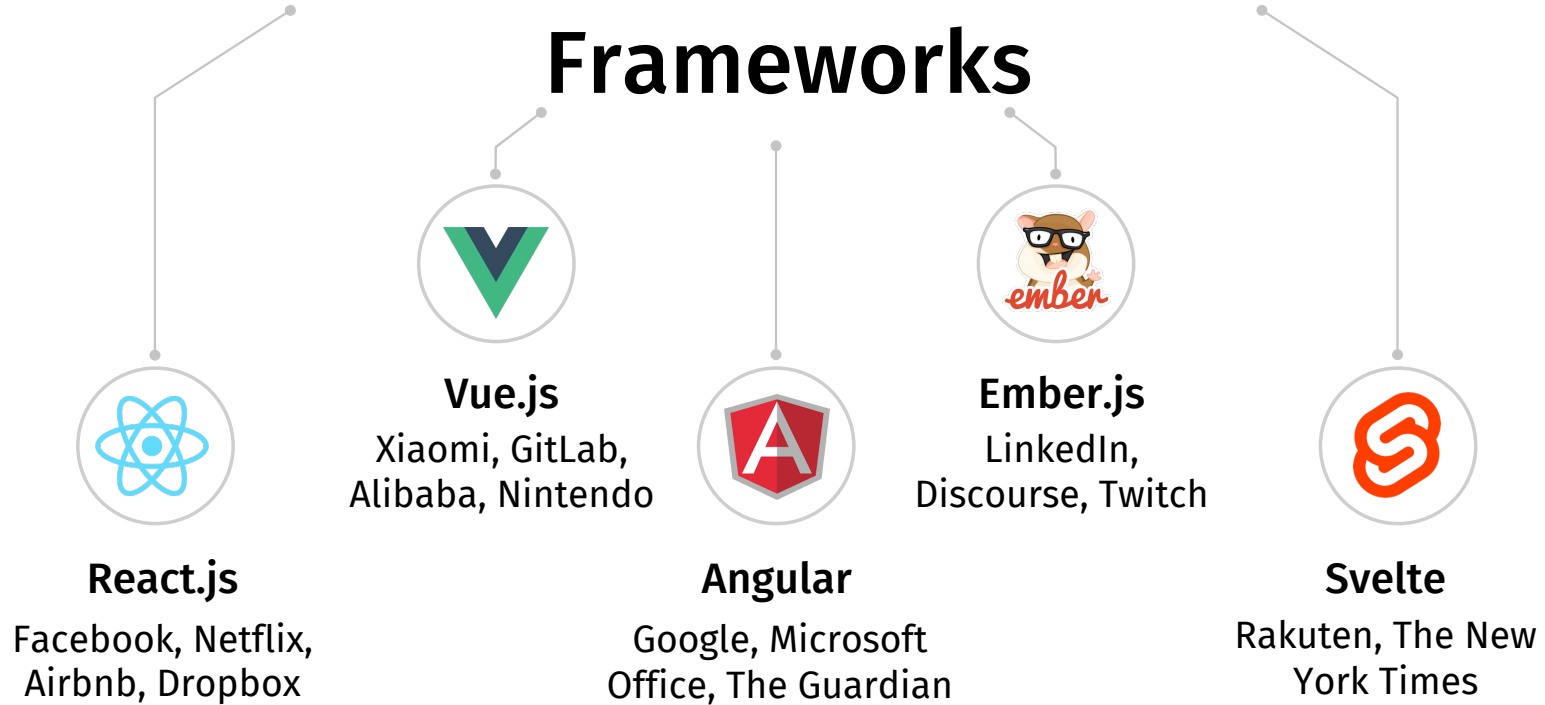
html

```
<p id="my-paragraph">Click the button to change me!</p>  
<button id="my-button">Click me!</button>
```

javascript

```
// get the paragraph and button elements  
var paragraph = document.getElementById('my-paragraph');  
var button = document.getElementById('my-button');  
  
// add click eventlistener to the button element  
button.addEventListener('click', function() {  
    // change the text content of the paragraph element  
    paragraph.textContent = "I've been changed!";  
});
```

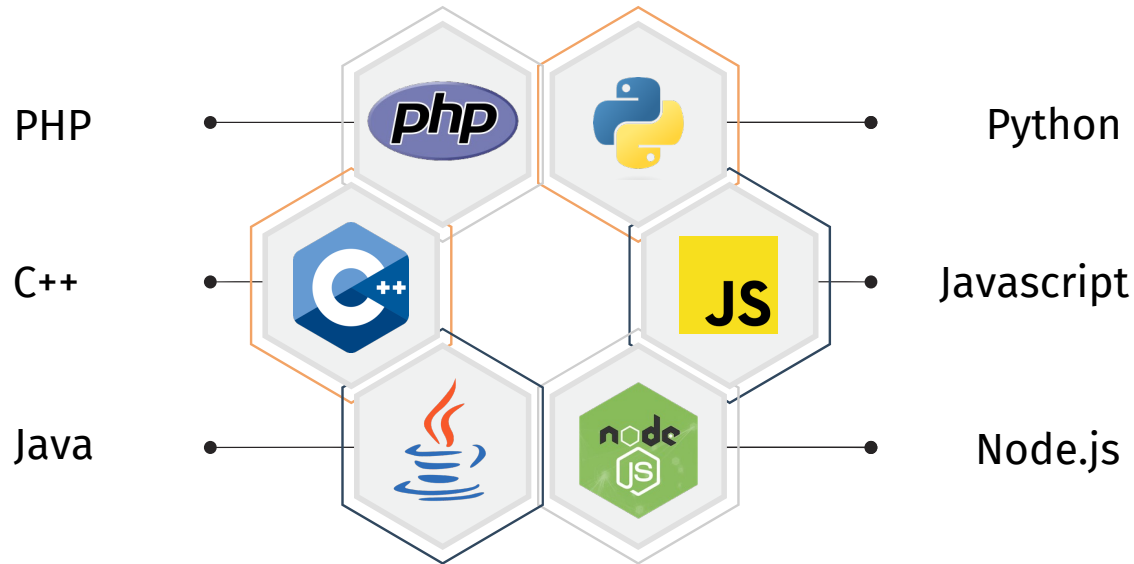
# Font-end Frameworks



Frontend frameworks are pre-written, reusable code libraries that provide developers with a set of tools, components, and styles for building web user interfaces. They typically include HTML, CSS, and JavaScript code, as well as a set of templates and components that can be used to quickly create responsive and visually appealing web pages.

# BACK-END

Back end It refers to the server-side development of web application or website with a primary focus on how the website works. It is responsible for managing the database through queries and APIs by client-side commands. The back end portion is built by using some libraries, frameworks, and languages.



# Back-end Frameworks



**Express**  
(usada com Node.js)  
Uber, Netflix,  
PayPal



**Django**  
Instagram, Spotify,  
Dropbox



**Laravel**  
Buzzfeed, BBC, Quora



**Spring**  
Ticketmaster, Wix,  
Philips

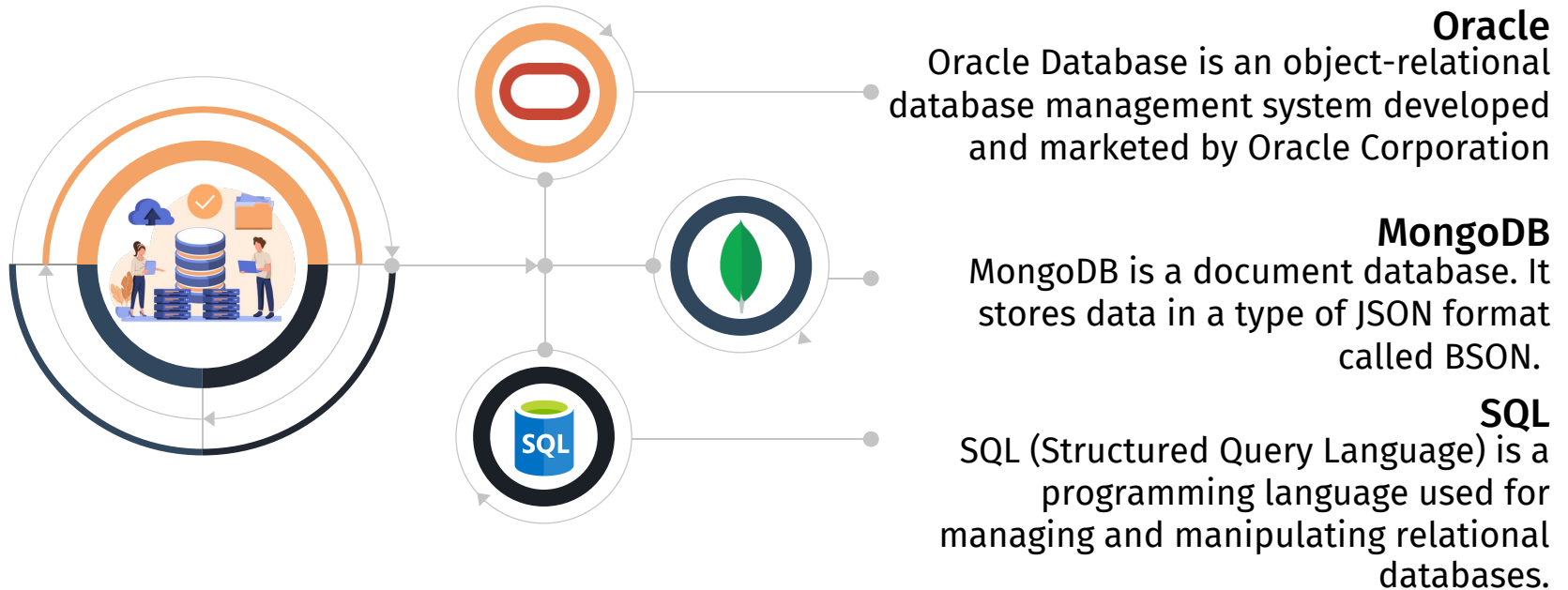


**Rails**  
Airbnb, Twitch,  
GitHub

Other back-end program/scripting languages are C, Ruby, REST, GO etc.

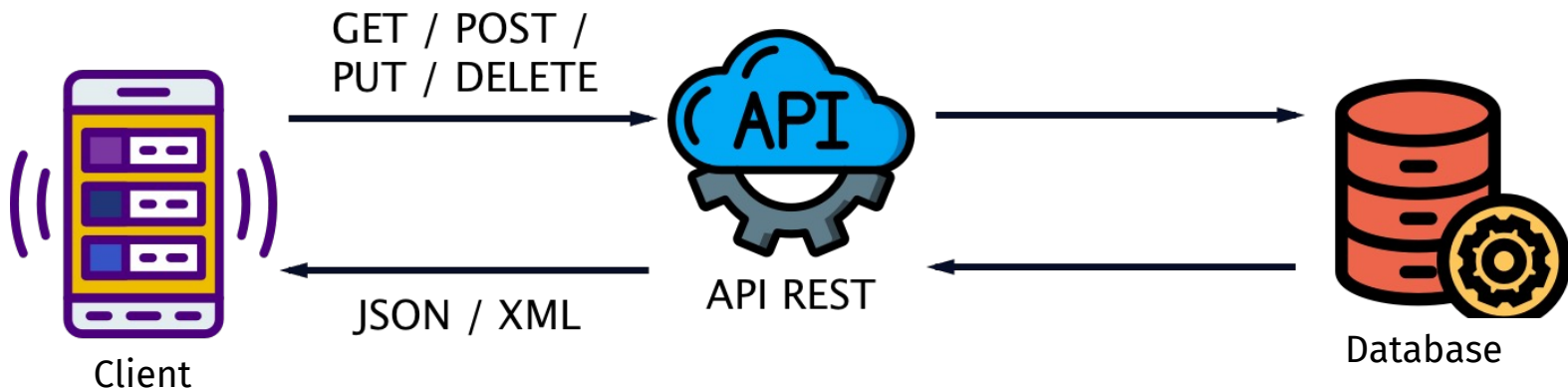
# Database Technologies

Database is the collection of interrelated data which helps to inserting and deleting the data from database and organizes the data in the form of tables, views, schemas, reports etc.



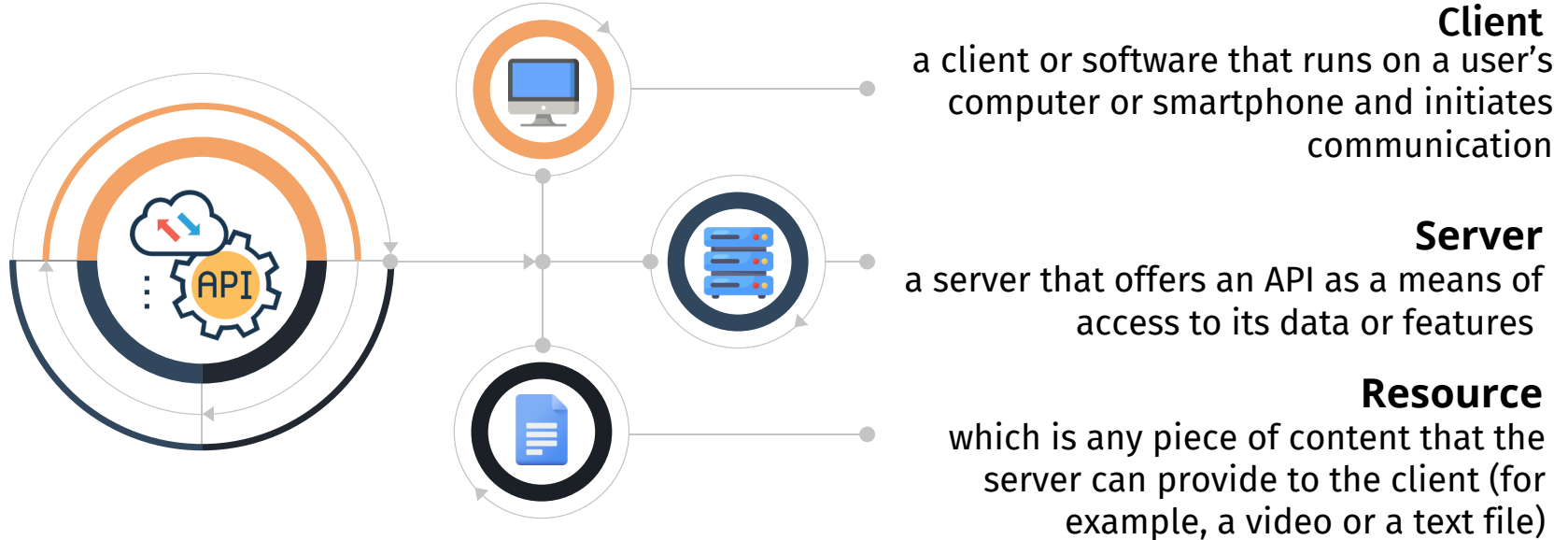
# REST API

REST is short for **Representational State Transfer**, an architectural style for building web services that interact via an HTTP protocol. A REST client can interact with each resource by sending an HTTP request.





# REST API concepts



To get access to a resource, the client sends an **HTTP request**.

In return, the server generates an **HTTP response** with encoded data on the resource.

# REST API methods

Any REST request includes four essential parts: an HTTP method, an endpoint, headers, and a body. An **HTTP method** describes what is to be done with a *resource*. There are four basic methods also named CRUD operations:

**POST**

to Create a resource

**GET**

to Retrieve a resource



**PUT**

to Update a resource

**DELETE**

to Delete a resource

# Request structure

Endpoint	<code>https://apiurl.com/review/new</code>
HTTP Method	<code>POST</code>
HTTP Headers	<code>content-type: application/json</code> <code>accept: application/json</code> <code>authorization: Basic abase64string</code>
Body	<pre>{   "review" : {     "title" : "Great article!",     "description" : "So easy to follow.",     "rating" : 5   } }</pre>

# Response structure

In response, the server sends not the sought-for resource itself, but its **representation** — a machine-readable description of its current state. The same resource can be represented in different formats, but the most popular ones are XML and [JSON](#).

```
HTTP/1.1 200
Content-Type: application/json

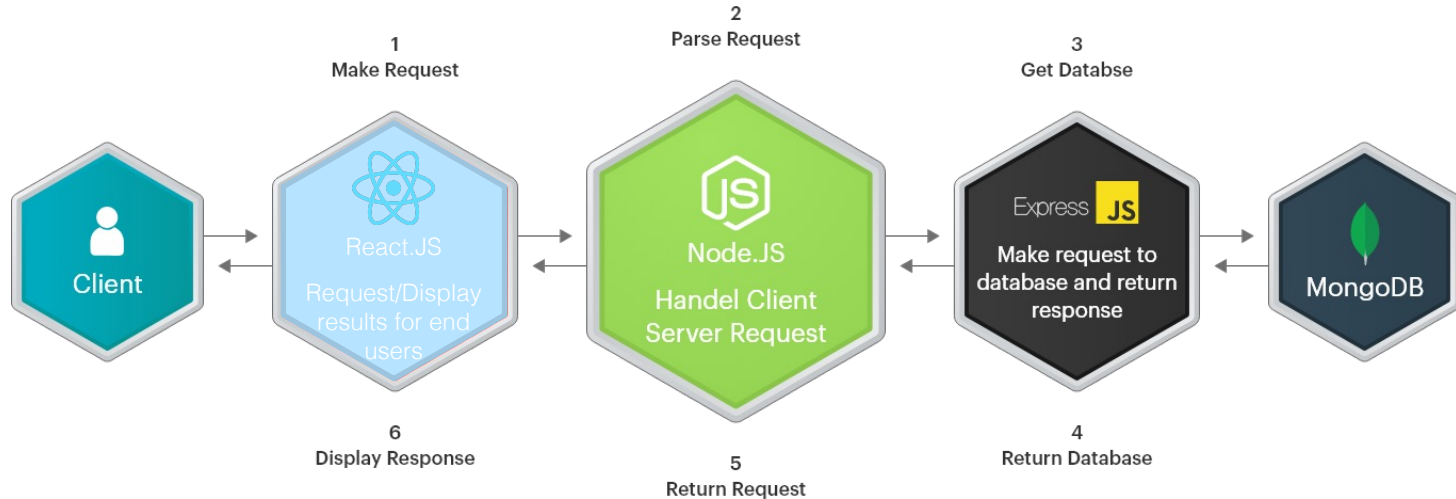
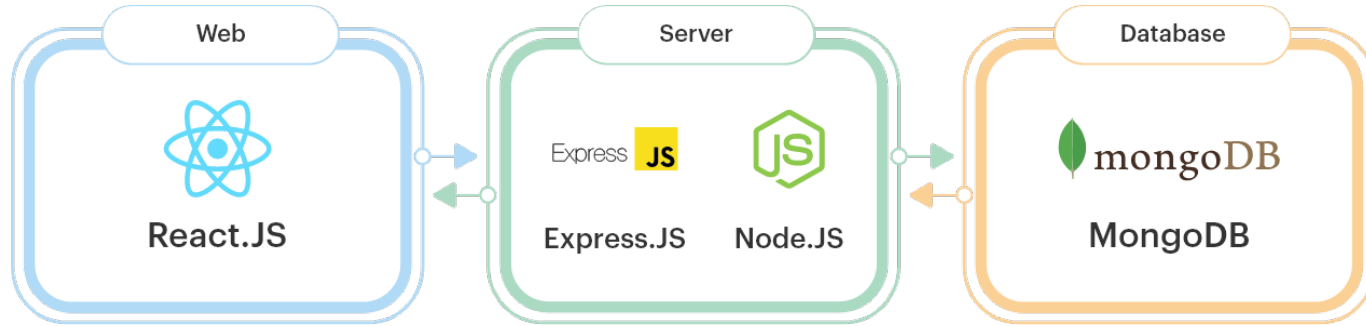
{
  "id": 10,
  "name": "shirt",
  "color": "red",
  "price": "$23"
}
```

## HTTP Status Codes



# Arquitecture overview

MERN stack as an example



# APLICAÇÕES INFORMÁTICAS EM ENGENHARIA BIOMÉDICA

