**Node.js**

***JavaScript – Novidades ES6***

* Let e Const
  + **var** – define elementos de escopo global ou escopo de função, se for dentro de funções
  + **Let e const** – definem elementos de escopo local
* Template Literals
  + Similar a strings, com crases (`..`) no lugar de apostrofes (‘..’) ou aspas (“..”)
  + Permite textos em múltiplas linhas
  + Permite a inserção de expressões por meio da construção $ { expr }
* Arrow Functions
  + Arrow Functions são uma sintaxe simplificada para a definição de funções

***Promises***

Uma promise é um mecanismo que recebe uma função que por sua vez apresenta dois call-backs:

* Uma operação resolve a ser executada caso a função seja bem sucedida
* Uma operação reject a ser executada caso a função falhe

As promises buscam simplificar a programação assíncrona. Recursos similares realizados, anteriormente, via bibliotecas como: jQuery ou deferred.js

**Estados de uma promise** – Possíveis estágios de uma promise no tempo

* Pendente (pending) – A operação disparada ainda está sendo executada
* Resolvida (resolved) – A operação teve êxito e executa-se a função resolve
* Rejeitada (rejected) – A operação falhou e executa-se a função reject

***Arquitetura da web - visão geral***

O protocolo HTTP é a forma como clientes e servidores se comunicam na rede. As requisições e as respostas obedecem aos padrões estabelecidos pelo protocolo HTTP.

A requisição HTTP é um pacote de dados enviado pela rede pelo cliente web para o servidor web e identifica o recurso solicitado. A resposta HTTP é formada por pacotes de dados enviados pelo servidor web para o cliente web com os recursos solicitados.