Seccion 1

**Responde las siguientes en tus propias palabras**

1. ¿Qué es HTTP y en donde se utiliza?

Hypertext Markup Language,

1. ¿Menciona los Verbos HTTP más utilizados?
2. En breves palabras , ¿ En qué consiste la arquitectura cliente-servidor?
3. ¿Qué tipos de relaciones hay en SQL?

Primarias y foráneas

1. ¿Que hace el siguiente comando “**git commit**”?

Salva cambios en el repositorio local

1. ¿Como se compone MVC?

Modelo, Vista, Controlador

1. ¿En qué consiste la Programación Orientada a Objetos?

Programar para expresar las cosas en forma más realista (POO)

1. ¿Qué es **REGEX**  y que usos le darías ?
2. ¿Qué hacen los siguientes comandos de consola?
   1. **CP copiar**
   2. **MKDIR crear una nueva carpeta**
   3. **CD cambiar de carpeta**
   4. **LS lista de archivos**
   5. **TOUCH**
   6. **TOP**
3. ¿Qué es un Middleware?

Tipo Api que permite visualizar en tiempo real información que se encuentra en base de datos, servidores, etc

1. ¿Que es una “promesa” en javascript?

Es un objeto devuelto que se engachan a callbacks.

1. ¿Como creas un nuevo repositorio en git?

git init

git commit -m “descripción”

git remote add origin “url del repositorio que utilizaremos”

git push origin master

Sección 2

**Resuelve los siguientes ejercicios de lógica, no importando el lenguaje de programación. Para el código puedes crear un gist publico (** [**https://gist.github.com/**](https://gist.github.com/) **)  y poner el enlace debajo de cada problema o subirlo a un repositorio en github.**

1. Given an input string, reverse the string word by word. For example, given s = "the sky is blue", return "blue is sky the".
2. Given a string, find the length of the longest substring without repeating characters. For example, the longest substring without repeating letters for "abcabcbb" is "abc", which the length is 3. For "bbbbb" the longest substring is "b", with the length of 1.
3. Given two words, beginWord and endWord, and a wordList of approved words, find the length of the shortest transformation sequence from beginWord to endWord such that:

* Only one letter can be changed at a time
* Each transformed word must exist in the wordList.
* Return the length of the shortest transformation sequence, or 0 if no such transformation sequence exists.

Note: beginWord does not count as a transformed word. You can assume that beginWord and endWord are not empty and are not the same.

Example:  For beginWord = "hit", endWord = "cog", and wordList = ["hot", "dot", "lot", "log", "cog"], the output should be wordLadder(beginWord, endWord, wordList) = 5.

The shortest transformation is "hit" -> "hot" -> "dot" -> "dog" -> "cog" with a length of 5

Sección 3

1. Crea una instancia en Heroku , Netlify , Firebase  o cualquier otra plataforma de cloud que exponga una aplicación web que:
   1. La aplicación deberá imprimir la latitud y longitud del usuario, así como su User Agent.
   2. Comparte la URL del repo en Github o Bitbucket.
   3. Comparte la URL donde esta hosteado el proyecto