

Administración del Conocimiento

Sistema experto.

Oscar Martínez Martínez | **UP150189**

Dr. José Eder Guzmán Mendoza

06 Agosto / 2018

# Descripción general del sistema experto.

Uno de los mayores problemas que se encuentran los reclutadores al buscar desarrolladores es el buscar buenas habilidades técnicas, entonces surge una pregunta ¿Qué tipo de habilidades técnicas y conocimientos técnico requiero para hacer una buena selección de profesionistas especializados en el frontend de aplicaciones web?

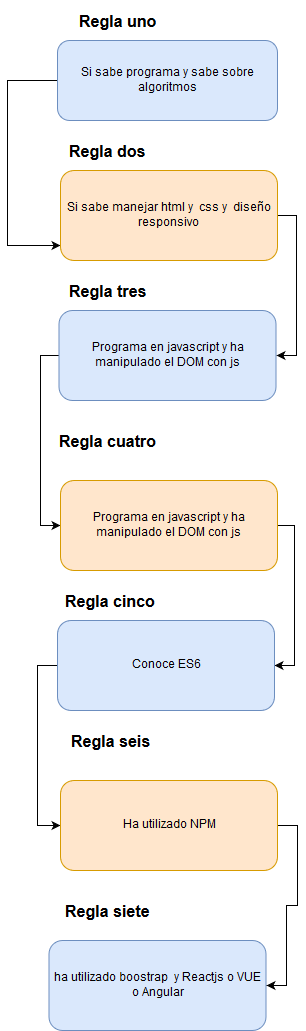
El desarrollo frontend ha evolucionado con lo largo de los años, en la actualidad, ya no solo es necesario saber html, css y javascript, es cierto que este conocimiento te da las bases pero en la actualidad el desarrollo frontned ha evolucionado tanto que se podría dar una sola carrera enfocado solo a esto. Es por eso que me he dado la tarea de analizar el proceso que se requiere para llegar a ser un verdadero profesional del frontend, analizar todos los conocimientos que se requieren, en nuestra primera versión solo analizamos ciertas variables características de los desarrolladores, en cada iteración de desarrollo nuestro sistema experto se volverá más eficiente, de esta forma crear un sistema experto funcional para seleccionar a los mejores profesionistas frontend técnicamente hablando. Y básicamente mi sistema experto responde a la pregunta ¿este candidato cuenta como las habilidades técnicas como desarrollador frontend? Y hay dos posibles salidas, este candidato cuenta con los conocimientos necesarios o no.

# Reglas de inferencia.

1. Si el candidato sabe programar y conoce sobre algoritmos de programación entonces tiene nociones de la programación.
2. Si el candidato sabe manejar html y darle estilo con css así como diseño responsivo, entonces saber maquetar para cualquier pantalla.
3. Si el candidato sabe programar en javascript y manejar el DOM de html con JS entonces puede hacer páginas interactivas y dinámicas.
4. Si el candidato conoce ES6 entonces esta actualizado en las nuevas especificaciones de javascript.
5. Si el candidato ha usado npm, entonces ha utilizado módulos de terceros.
6. Si el candidato sabe boostrap entonces ha manejado framworks web.
7. Si el candidato ha trabajo con React o Vue o Angular entonces ha hecho aplicaciones de una sola página.

Cuando todas nuestras variables (subrayadas en verde) se cumplan, la salida es positiva, el candidato cumple con todas las habilidades que se requieren para un desarrollador frontend, si una sola de estas variables no se cumple, el candidato no cuenta con las habilidades necesarias.

# Diagrama del sistema experto.



Programación del sistema experto.

El software está escrito con Javascript para correr en Nodejs, recibe entradas y salidas por la terminal.

*const* readline = require('readline')

*const* util = require('util');

*const* colors = require('colors');

*const* rl = readline.createInterface({ input: process.stdin, output: process.stdout });

rl.question[util.promisify.custom] = (*arg*) *=>* {

return new Promise((*resolve*) *=>* {

rl.question(arg, resolve);

});

};

*const* questionPromise = util.promisify(rl.question);

async *function* main() {

*let* programmingVar = false;

*let* v1 = false;

do {

*const* answer = await questionPromise('¿Sabe programación? \nA)Si \nB)No \nR= '.green);

if (answer.toLowerCase() === 'a') {

programmingVar = true;

v1 = false;

} else if (answer.toLowerCase() === 'b') {

programmingVar = false;

v1 = false;

} else {

*console*.log('Escribe una respuesta correcta'.rainbow);

v1 = true;

}

} while (v1);

*let* algorithmVar = false;

*let* p2 = false;

do {

*const* answer2 = await questionPromise('¿Sabes sobre Algoritmos? \nA)Si \nB)No \nR= '.yellow);

if (answer2.toLowerCase() === 'a') {

algorithmVar = true;

p2 = false;

} else if (answer2.toLowerCase() === 'b') {

algorithmVar = false;

p2 = false;

} else {

*console*.log('Escribe una respuesta correcta'.rainbow);

p2 = true;

}

} while (p2);

//Evaluación de la regla 1

*const* rule1 = (programmingVar && algorithmVar);

*let* htmlVar = false;

*let* cssVar = false;

*let* p3 = false;

do {

*const* answer3 = await questionPromise('¿Haz usado html5 y css? \nA)Si \nB)No \nR= '.green);

if (answer3.toLowerCase() === 'a') {

htmlVar = true;

cssVar = true;

p3 = false;

} else if (answer3.toLowerCase() === 'b') {

htmlVar = false;

cssVar = false;

p3 = false;

} else {

*console*.log('Escribe una respuesta correcta'.rainbow);

p3 = true;

}

} while (p3);

*let* reponsiveVar = false;

*let* p4 = false;

do {

*const* answer4 = await questionPromise('¿Haz hecho diseño reponsivo con css? \nA)Si \nB)No \nR= '.yellow);

if (answer4.toLowerCase() === 'a') {

reponsiveVar = true;

p4 = false;

} else if (answer4.toLowerCase() === 'b') {

reponsiveVar = false;

p4 = false;

} else {

*console*.log('Escribe una respuesta correcta'.rainbow);

p4 = true;

}

} while (p4);

//Evaluación de la regla2

*const* rule2 = (htmlVar && cssVar && reponsiveVar);

*let* javascriptVar = false;

*let* p5 = false;

do {

*const* answer5 = await questionPromise('¿Haz escrito programas con javascripr? \nA)Si \nB)No \nR= '.red);

if (answer5.toLowerCase() === 'a') {

javascriptVar = true;

p5 = false;

} else if (answer5.toLowerCase() === 'b') {

javascriptVar = false;

p5 = false;

} else {

*console*.log('Escribe una respuesta correcta'.rainbow);

p5 = true;

}

} while (p5);

*let* domVar = false;

*let* p6 = false;

do {

*const* answer6 = await questionPromise('¿Haz manipulado el DOM de html desde javascript? \nA)Si \nB)No \nR= '.red);

if (answer6.toLowerCase() === 'a') {

domVar = true;

p6 = false;

} else if (answer6.toLowerCase() === 'b') {

domVar = false;

p6 = false;

} else {

*console*.log('Escribe una respuesta correcta'.rainbow);

p6 = true;

}

} while (p6);

*const* rule3 = (javascriptVar && domVar);

*let* emacscript6 = false;

*let* p7 = false;

do {

*const* answer7 = await questionPromise('¿Conoces emacscript 6? \nA)Si \nB)No \nR= '.red);

if (answer7.toLowerCase() === 'a') {

emacscript6 = true;

p7 = false;

} else if (answer7.toLowerCase() === 'b') {

emacscript6 = false;

p7 = false;

} else {

*console*.log('Escribe una respuesta correcta'.rainbow);

p7 = true;

}

} while (p7);

*const* rule4 = (emacscript6);

*let* npmVar = false;

*let* p8 = false;

do {

*const* answer8 = await questionPromise('¿Conoces instalado modulos o librerias desde npm ? \nA)Si \nB)No \nR= '.green);

if (answer8.toLowerCase() === 'a') {

npmVar = true;

p8 = false;

} else if (answer8.toLowerCase() === 'b') {

npmVar = false;

p8 = false;

} else {

*console*.log('Escribe una respuesta correcta'.rainbow);

p8 = true;

}

} while (p8);

*const* rule5 = (npmVar);

*let* boostrapVar = false;

*let* p9 = false;

do {

*const* answer9 = await questionPromise('¿haz implementado boostrap? \nA)Si \nB)No \nR= '.green);

if (answer9.toLowerCase() === 'a') {

boostrapVar = true;

p9 = false;

} else if (answer9.toLowerCase() === 'b') {

boostrapVar = false;

p9 = false;

} else {

*console*.log('Escribe una respuesta correcta'.rainbow);

p9 = true;

}

} while (p9);

*const* rule6 = (boostrapVar);

*let* frameworksWeb = false;

*let* p10 = false;

do {

*const* answer10 = await questionPromise('¿haz trabajado con React o Vue o Angular? \nA)Si \nB)No \nR= '.green);

if (answer10.toLowerCase() === 'a') {

frameworksWeb = true;

p10 = false;

} else if (answer10.toLowerCase() === 'b') {

frameworksWeb = false;

p10 = false;

} else {

*console*.log('Escribe una respuesta correcta'.rainbow);

p10 = true;

}

} while (p10);

*const* rule7 = (frameworksWeb);

if (rule1 && rule2 && rule3 && rule4 && rule5 && rule6 && rule7) {

*console*.log('Este candidado tiene todas las habilidades frontend necesarias'.blue)

} else {

*console*.log('Este candidado no tiene todas las habilidades frontend necesarias'.bgRed)

}

rl.close();

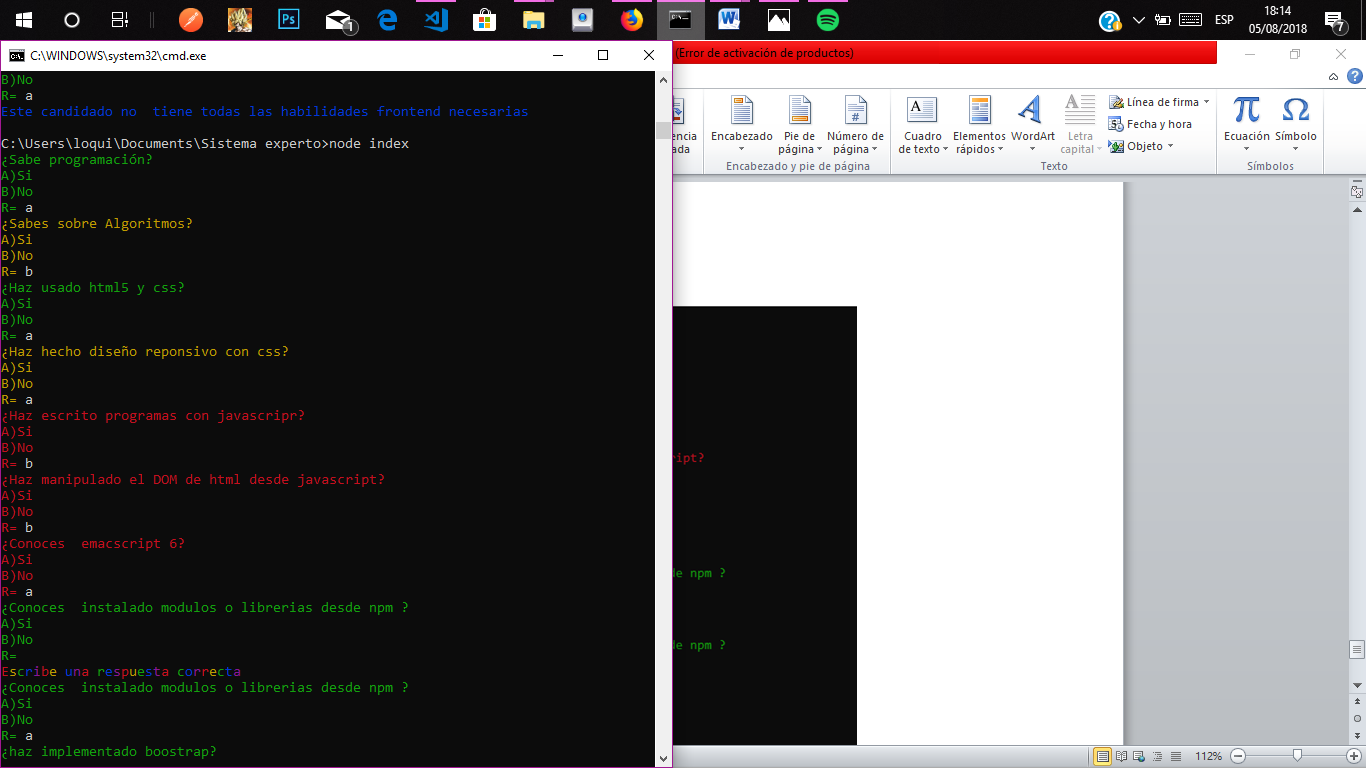
}

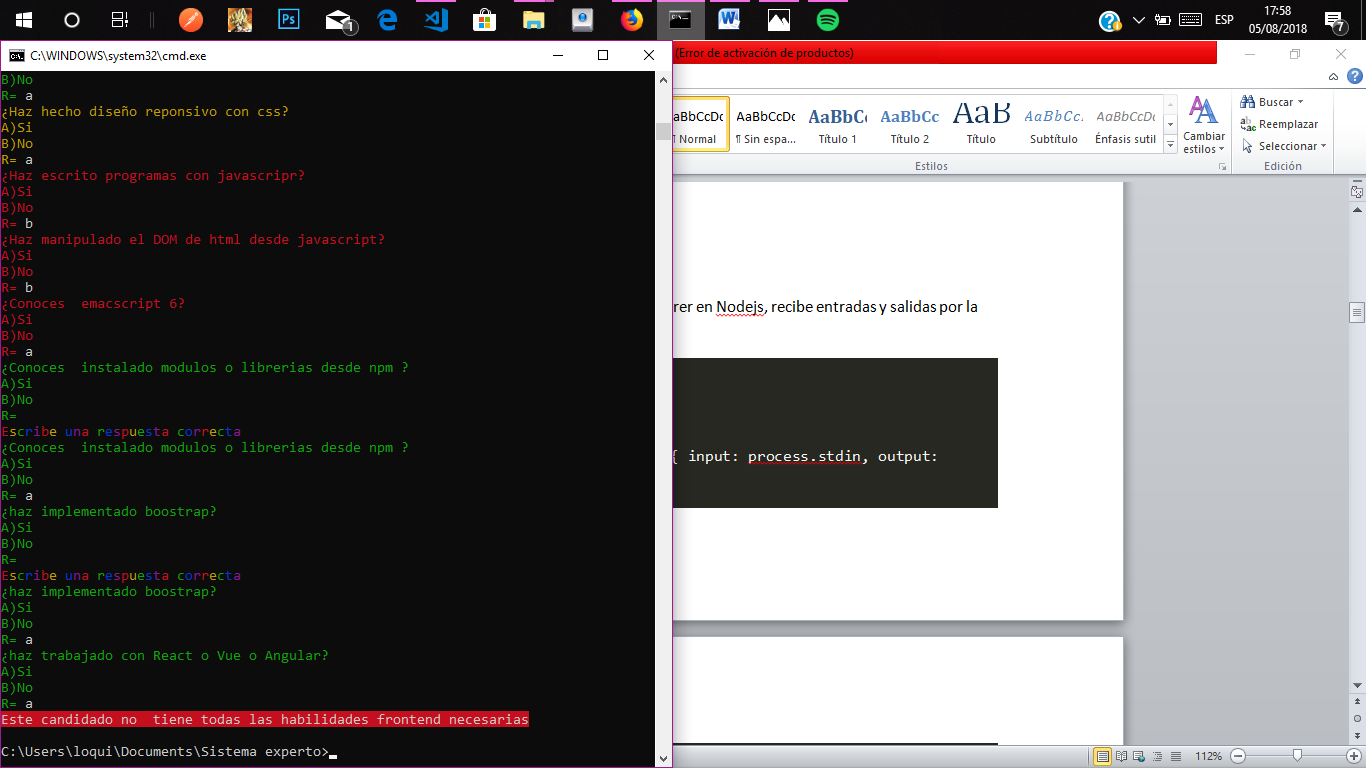
(async () *=>* {

await main();

})()

# Uso de hechos.





Repositorio en GitHub con el código fuente, documento y diagrama.