# Examen Teórico – Desarrollar en Node.JS.

1. Node.js es un sistema del lado del \_servidor\_\_\_\_\_\_\_\_\_\_\_\_\_.

2. Node.js es un entorno de programación basado en el lenguaje de programación

1. Java
2. JavaScript
3. PHP
4. Python

3. Node.js es una plataforma creada sobre \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ para crear aplicaciones de red rápidas y escalables.

1. El motor V5 de safari
2. El motor V8 de Chrome
3. El motor V8 de Firefox
4. Ninguno de los anteriores

4. Qué comando utilizas para saber la versión de node.js que tienes instalada

1. node --version
2. node -version
3. node -v
4. node.js –version

5. ¿Cómo ejecutas la aplicación miappnode.js en node.js?

1. run miappnode.js
2. node miappnode.js
3. exec miappnode.js
4. execute miappnode.js

6. How Node based web servers are different from traditional web servers?

1. Node based server process request much faster than traditional server.
2. Node based server uses a single threaded model and can services much larger number of requests than traditional server like Apache HTTP Server.
3. There is no much difference between the two.
4. None of the above.

7. By default, npm installs any dependency in the local mode.

1. True
2. False

8. Which method of fs module is used to read a directory?

1. fs.readDirectory(path[, mode], callback)
2. fs.read(path[, mode], callback)
3. fs.readdir(path, callback)
4. None of the above.

9. What is Express?

1. Express is a application framework that provides a robust set of features to develop desktop based applications.
2. Express is a minimal and flexible Node.js web application framework that provides a robust set of features to develop web and mobile applications.
3. Both of the above.
4. None of the above.

10. Which of the following is true about readable stream?

1. Readable stream is used for read operation.
2. Output of readable stream can be input to a writable stream.
3. Both of the above.
4. None of the above.

11. Which of the following stands true for dns.lookup(hostname[, options], callback) method?

1. Resolves a hostname (e.g. 'google.com') into the first found A (IPv4) or AAAA (IPv6) record.
2. If options is not provided, then IP v4 and v6 addresses are both valid. If options is an integer, then it must be 4 or 6.
3. Both of the above.
4. None of the above.

12. All APIs of Node.JS are.?

1. Asynchronous
2. Synchronous
3. Both of the above.
4. None of the above.

13. Which of the following is true about internal binding with respect to domain module?

1. Error emmitter is executing its code within run method of a domain.
2. Error emmitter is added explicitly to a domain using its add method.
3. Both of the above.
4. None of the above

14. Which of the following is not a valid HTTP method?

1. get
2. put
3. post
4. header

15. Which of the following is true about EventEmitter.on property?

1. on property is used to fire event.
2. on property is used to bind a function with the event.
3. on property is used to locate an event handler.
4. None of the above.

16. Which of the following API creates a client?

1. net.createServer([options][, connectionListener])
2. net.connect(options[, connectionListener])
3. net.createConnection(port[, host][, connectListener])
4. None of the above.

17. Which of the following is true about spawn methd of child\_process module.

1. The spawn() method method launches a new process with a given command.
2. The spawn() method returns streams (stdout, stderr).
3. Both of the above.
4. None of the above.

18. - Which of the following is true about process global object?

1. The process object is an instance of EventEmitter.
2. process emits exit event when process is about to exit.
3. process emits uncaughtException when when an exception bubbles all the way back to the event loop.
4. All of the above.

19. Which method of fs module is used to write a file?

1. fs.write(path, flags[, mode], callback)
2. fs.writeFile(path, flags[, mode], callback)
3. fs.writePath(path, flags[, mode], callback)
4. None of the above.

20. Which of the following code can make a request to a web server?

1. http.request(options, callback)
2. http.createRequest(options, callback)
3. http.makeRequest(options, callback)
4. None of the above.

21. Can we create child processes in Node applications.

1. True
2. False

22. Which of the following command will show all the modules installed globally?

1. $ npm ls -g
2. $ npm ls
3. $ node ls -g
4. $ node ls

23. Which of the following module is required for exception handling in Node?

1. web module
2. net module
3. domain module
4. error module

24. What REST stands for?

1. REpresentational State Transfer
2. Resource Efficient State Transfer
3. Real Elegant State Transfer
4. Resource Elegant State Transfer

25. A stream fires error event when there is any error receiving or writing data.

1. False
2. True

# PUNTO 1. Examen Base – Desarrollar en Node JS.

1. Haz una clase llamada Persona que siga las siguientes condiciones:

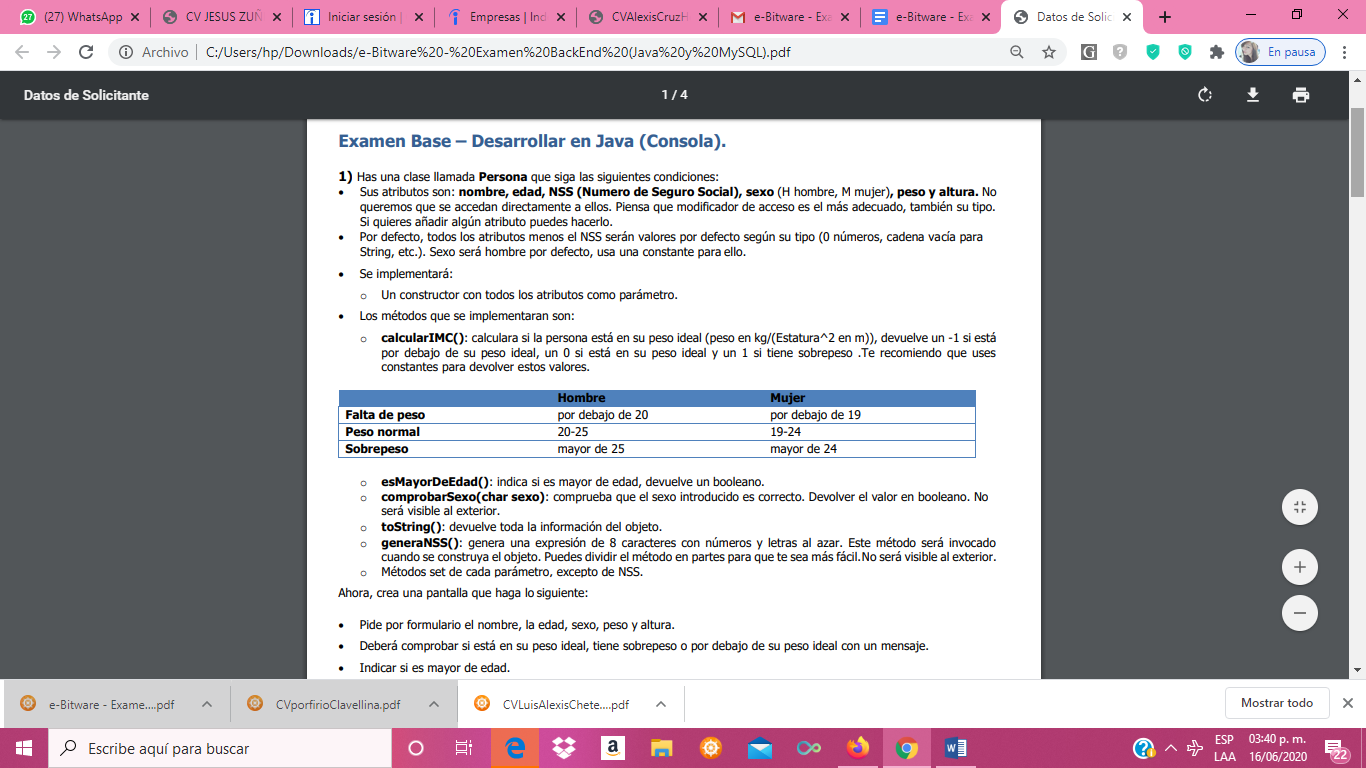
* Sus atributos son: nombre, edad, NSS (Numero de Seguro Social), sexo (H hombre, M mujer), peso y altura. No queremos que se accedan directamente a ellos. Piensa que modificador de acceso es el más adecuado, también su tipo. Si quieres añadir algún atributo puedes hacerlo.
* Por defecto, todos los atributos menos el NSS serán valores por defecto según su tipo (0 números, cadena vacía para String, etc.). Sexo será hombre por defecto, usa una constante para ello.

• Se implementará:

* Un constructor con todos los atributos como parámetro.

• Los métodos que se implementarán son:

o calcular IMC(): calculara si la persona está en su peso ideal (peso en kg/(Estatura^2 en m)), devuelve un -1 si está por debajo de su peso ideal, un 0 si está en su peso ideal y un 1 si tiene sobrepeso .Te recomiendo que uses constantes para devolver estos valores.



o esMayorDeEdad(): indica si es mayor de edad, devuelve un booleano.

o comprobarSexo(char sexo): comprueba que el sexo introducido es correcto. Devolver el valor en booleano. No será visible al exterior.

o toString(): devuelve toda la información del objeto.

o generaNSS(): genera una expresión de 8 caracteres con números y letras al azar. Este método será invocado cuando se construya el objeto. Puedes dividir el método en partes para que te sea más fácil. No será visible al exterior. o Métodos set de cada parámetro, excepto de NSS.

Ahora, crea una pantalla que haga lo siguiente:

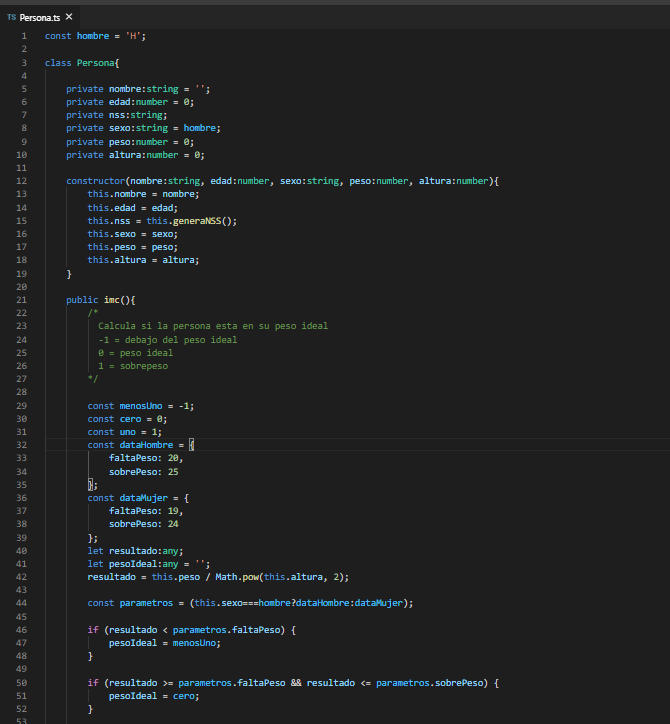
• Pide por formulario el nombre, la edad, sexo, peso y altura.

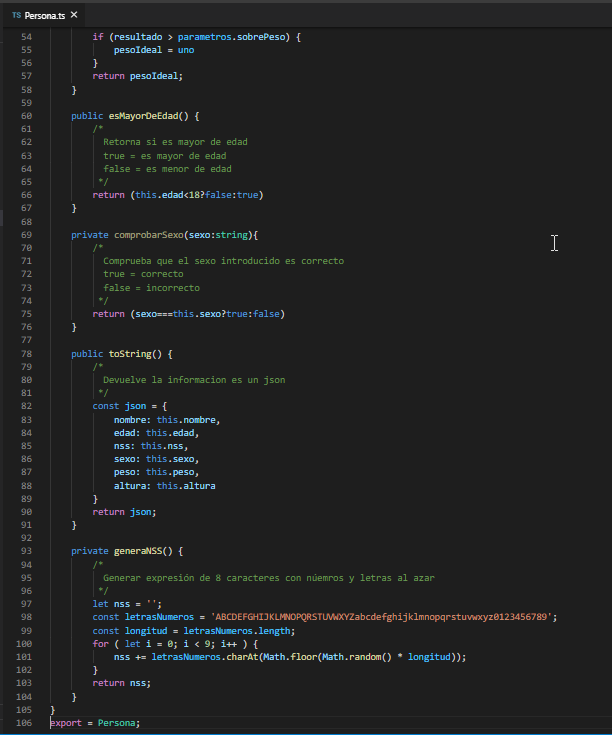
• Deberá comprobar si está en su peso ideal, tiene sobrepeso o por debajo de su peso ideal con un mensaje.

• Indicar si es mayor de edad.

• Por último, mostrar toda la información ingresada. Puedes usar métodos, para que te sea más fácil

**Evidencia** **- Examen Base – Desarrollar en Node JS.**





**PUNTO 2. Examen Servicios REST – Node JS.**

Desarrollar API REST con JSON que permita consumir dar de alta, consultar y modificar un cliente.

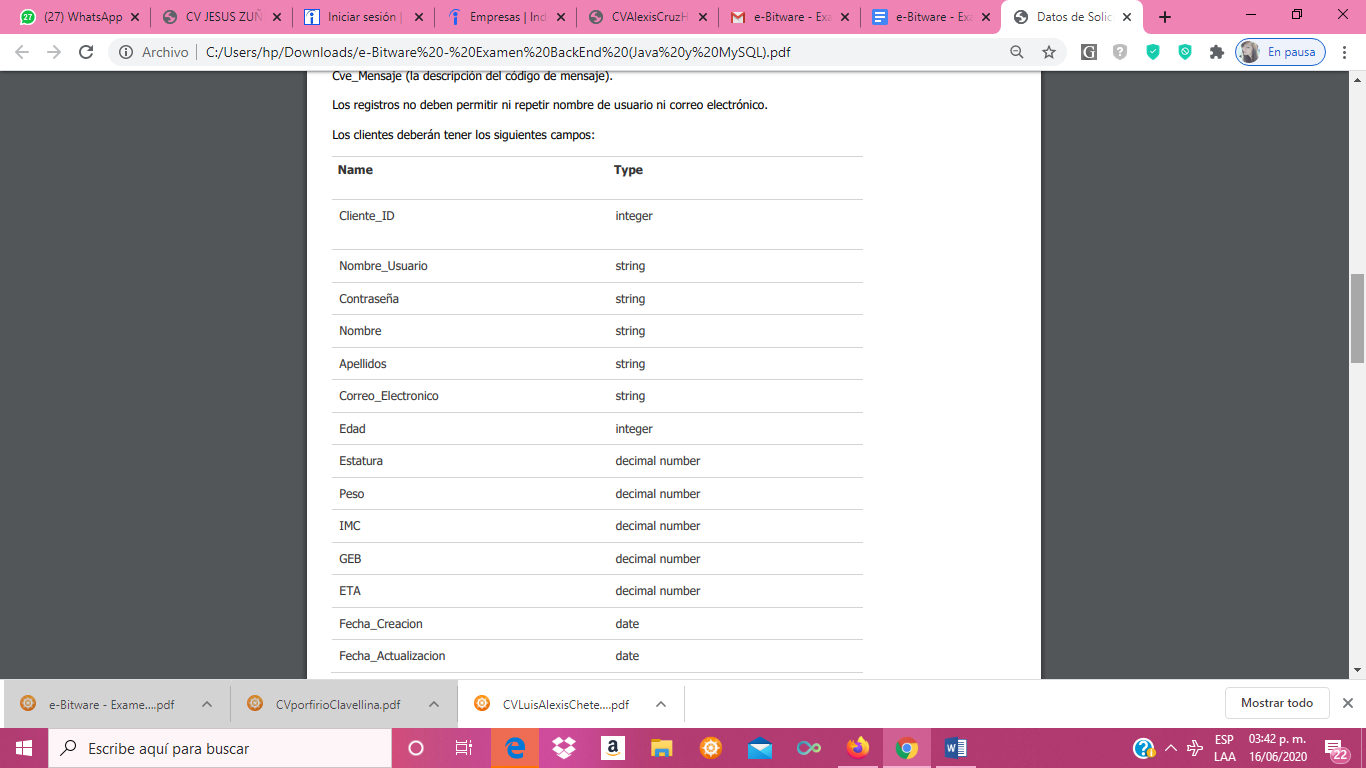
La respuesta de la API para cada petición deberá considerar los siguientes campos base dentro de la respuesta JSON:

Cve\_Error (donde 0 es exitoso y cual error es con un valor negativo).

Cve\_Mensaje (la descripción del código de mensaje).

Los registros no deben permitir ni repetir nombre de usuario ni correo electrónico.

Los clientes deberán tener los siguientes campos:



1. Realiza el Siguiente POST para agregar un Cliente

POST /NutriNET/Cliente

Con la siguiente estructura JSON (Diferentes Valores)

{“Nombre": "Juan",

"Apellidos": "Perez Hernandez",

"Nombre\_Usuario": "Ju",

"Correo\_Electronico":

"jperez@hotmail.com",

"Contraseña": "juanitoperez"

1. Realiza un Get (Tendrá la opción de todos o por ID)

GET /NutriNET/Cliente

3. Realiza el siguiente PUT, para modificar el peso y la estatura de un cliente

PUT /NutriNET/Cliente/59

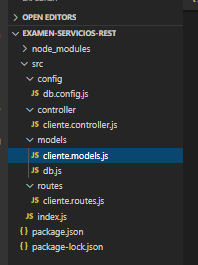
{ "Edad": 39 "

Estatura": 1.80

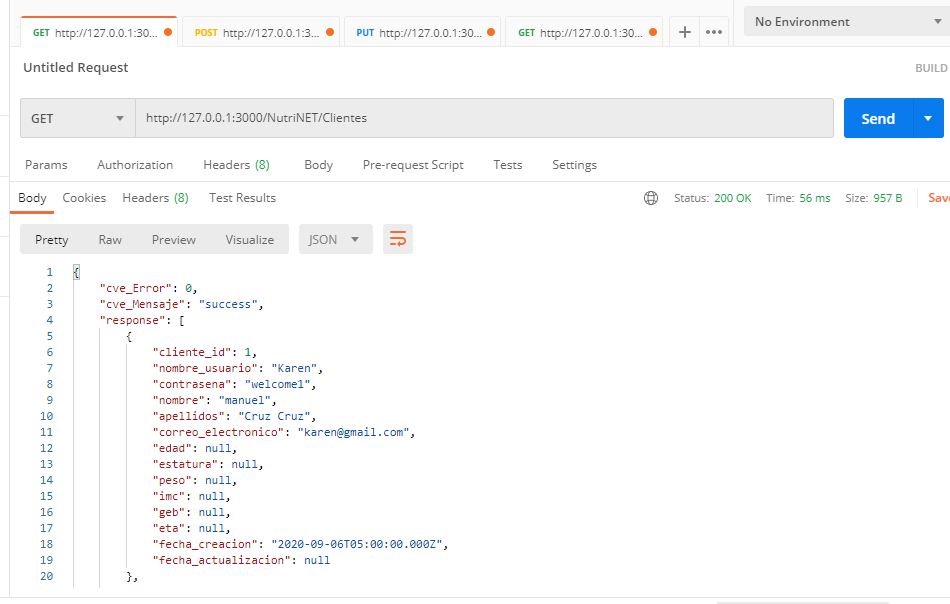
"Peso": 60 "

GEB": 1500 }

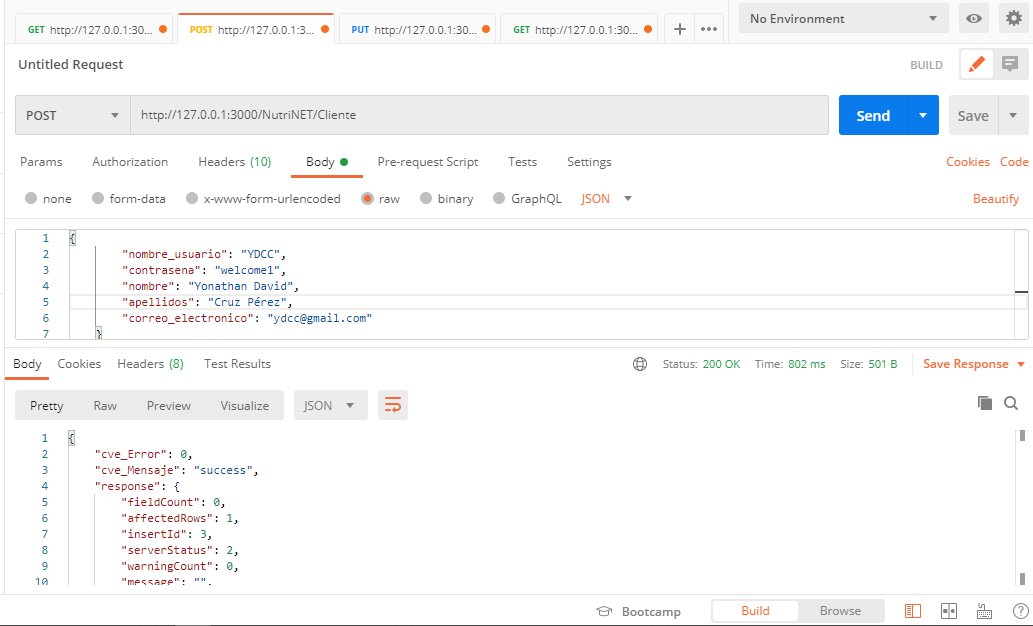
**Evidencia - Examen Servicios REST – Node JS.**



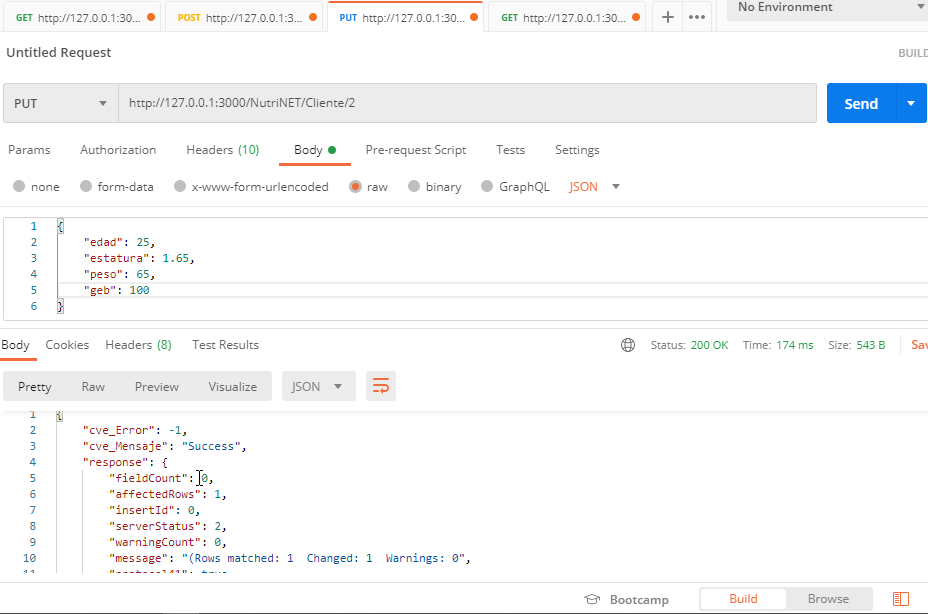
**GET**



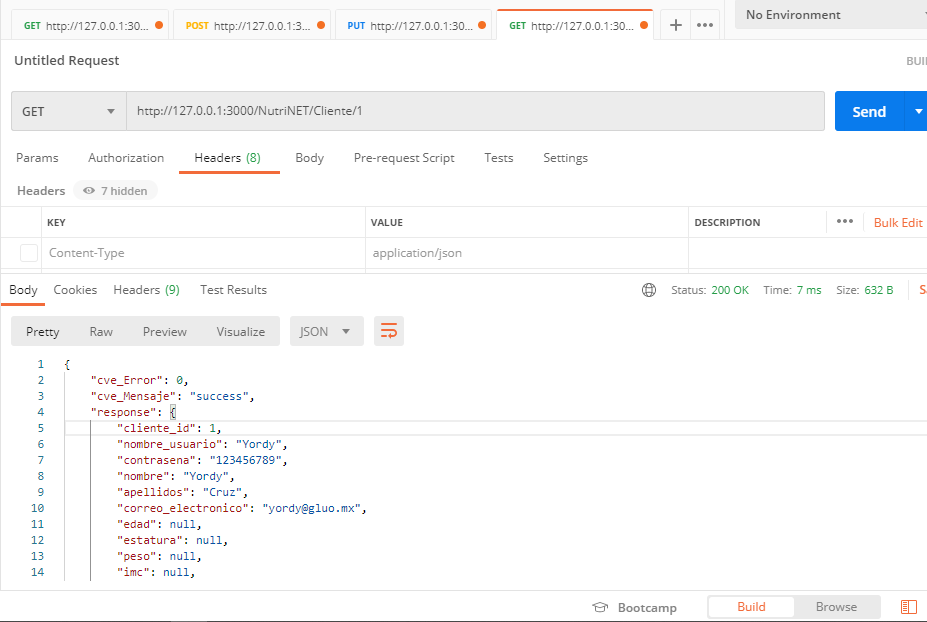
**POST**



**PUT**

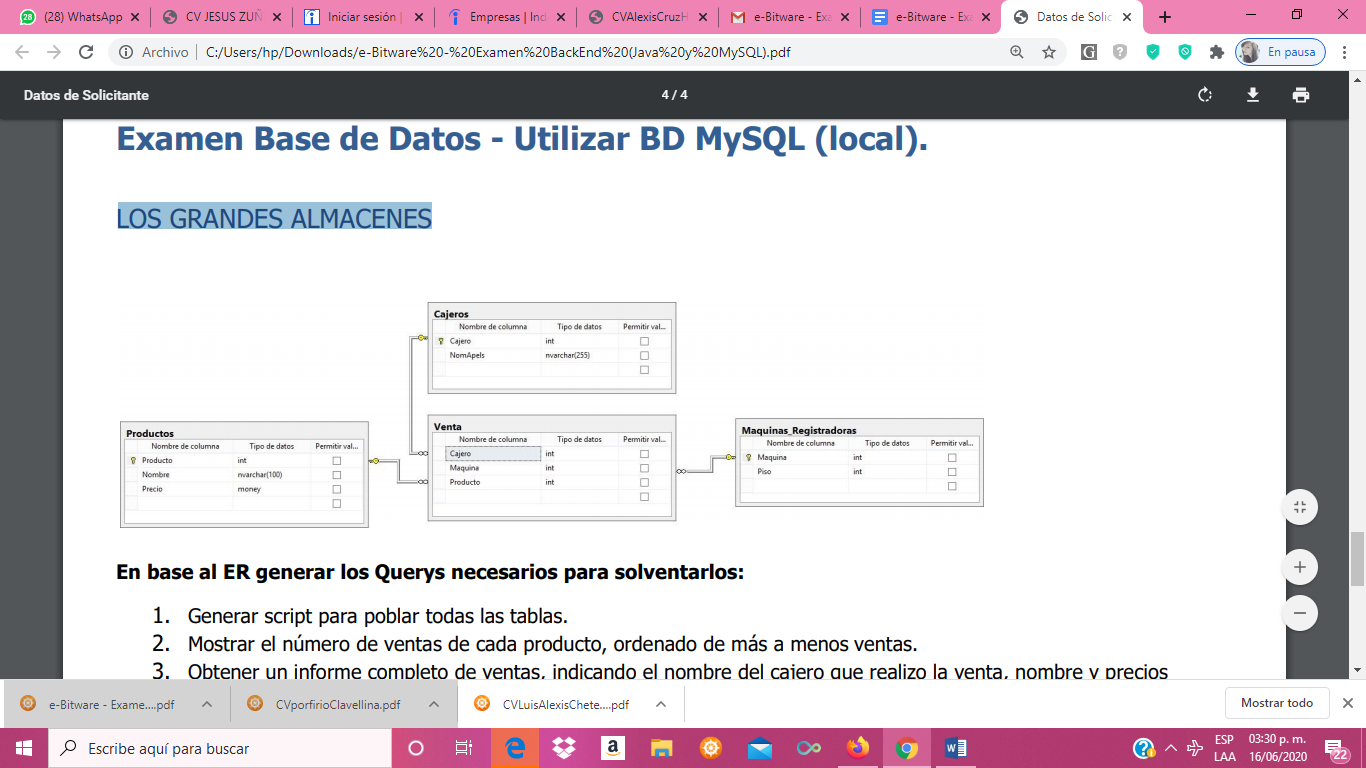


**GET:ID**



# PUNTO 3. Examen Base de Datos - Utilizar BD MySQL (local).

LOS GRANDES ALMACENES

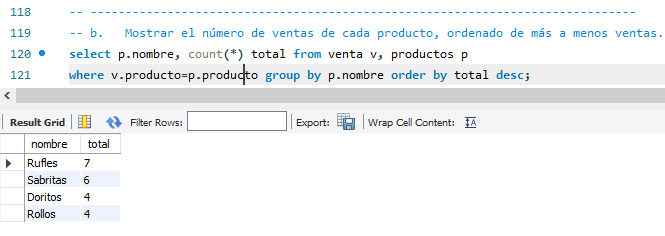


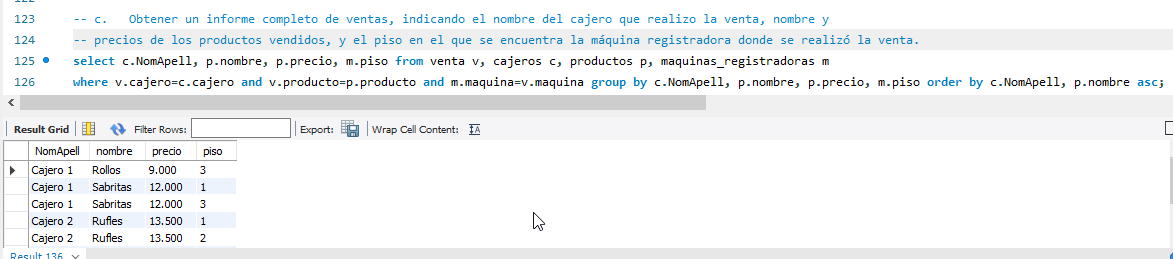
En base al ER generar los Querys necesarios para solventarlos:

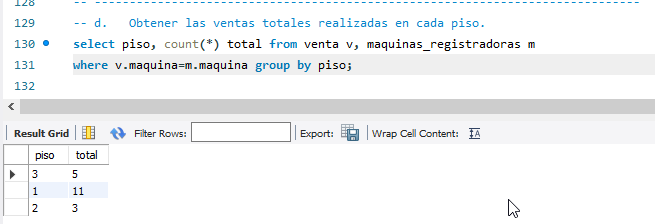
* 1. Generar script para poblar todas las tablas.
  2. Mostrar el número de ventas de cada producto, ordenado de más a menos ventas.
  3. Obtener un informe completo de ventas, indicando el nombre del cajero que realizo la venta, nombre y precios de los productos vendidos, y el piso en el que se encuentra la máquina registradora donde se realizó la venta.
  4. Obtener las ventas totales realizadas en cada piso.
  5. Obtener el código y nombre de cada cajero junto con el importe total de sus ventas.
  6. Obtener el código y nombre de aquellos cajeros que hayan realizado ventas en pisos cuyas ventas totales sean inferiores a los 5000 pesos.

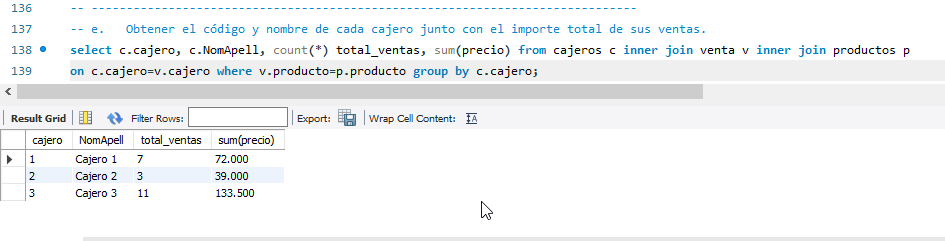
**Evidencia - Examen Base de Datos - Utilizar BD MySQL (local).**

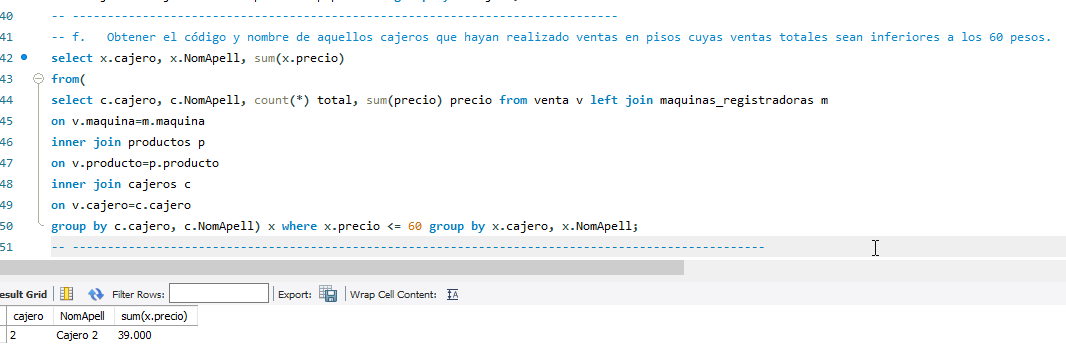












# PUNTO 4. Realizar la experiencia al usuario con Angular versión 5 en adelante.

**Evidencia**

