

ESCUELA POLITÉCNICA NACIONAL

Facultad de Sistemas

Desarrollo de Aplicaciones Web

Fecha: 12 – 06 – 2017 :GR2

Examen – 1er Bimestre

1) Crear un paquete en NPM y desplegar un servidor web

De los ejemplos en la clase, se debe de crear un nuevo paquete en npm. No se olvide que para esto usted debe de hacer login en su cuenta de npm.

Pasos

1. Iniciar el archivo package.json: (1 pt)
 - a. Con el comando npm init, inicie un nuevo proyecto, llene los datos correspondientes.
 - b. El nombre del paquete debe de ser establecido con el siguiente formato:
“aweb-examen-01-apellido-nombre”
Ejemplo: En mi caso mi nombre es Adrian y mi apellido es Eguez el paquete tendría el siguiente nombre:
“aweb-examen-01-eguez-adrian”
2. Dentro de este paquete va a levantar el servidor web llamado “express”, lo puede instalar con el siguiente comando: (1 pt)
 - a. npm install express -- save
 - b. Documentación de instalación: <http://expressjs.com/en/starter/installing.html>
3. Después de instalar el servidor cree un nuevo archivo llamado “index.js”. En este archivo es donde usted programará su servidor web. (1 pt)
 - a. Documentación de uso: <http://expressjs.com/en/starter/hello-world.html>
4. Express nos permite crear urls donde escuchar peticiones con el protocolo http. A continuación está un ejemplo donde el servidor tiene definido 1 ruta “/bienvenido” la cuál escucha solamente peticiones con el método http “GET” y responde con texto plano “Hello World”. Dentro del archivo **index.js** levante un servidor express para continuar con su examen como el del ejemplo de abajo pero levantado en el puerto “8080”. (2 pt)
 - a. Ejemplo:

```
const express = require('express')
const app = express()

app.get('/bienvenido', function (req, res) {
  res.send('Hello World!')
})

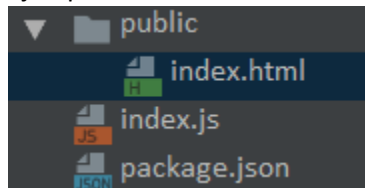
app.listen(3000, function () {
  console.log('Example app listening on port 3000!')
})
```

- b. MUY IMPORTANTE CAMBIAR EL PUERTO (DEL PUERTO 3000 AL PUERTO “8080”)
- c. Después de haber creado su servidor HTTP va a elaborar las siguientes rutas:

5. Ruta **Información personal**: Mediante Express va a crear una ruta a la url: **“/informacion”** que escuche peticiones solamente con el método http **“POST”**. Usamos el método post por razones académicas. (1 pt) (Documentación: <http://expressjs.com/en/starter/basic-routing.html>)
6. Respuesta de ruta **Información personal**: La respuesta a esta ruta debe de ser leída de un archivo que vamos a crear a continuación. El archivo debe de llamarse: **“Informacion.txt”** y debe de estar localizada en el mismo directorio que el archivo **“index.js”**. Este archivo tendrá la siguiente información: (1 pt)
 - a. Apellidos Nombres
 - b. Numero Unico
 - c. Hobby
 - d. Ejemplo del contenido del archivo **“Informacion.txt”** :
Eguez Sarzosa Vicente Adrian
200810275
Videojuegos
7. Lectura del archivo: Dentro de la ruta **“/Informacion”** vamos a leer el archivo dentro de la función callback como se muestra a continuación. Para la lectura del archivo puede guiarse del ejemplo hecho en clase. (1 pt)
 - a. Lectura del archivo:

```
app.post('/', function (req, res) {  
  Aqui leer archivo  
})
```

- b. NO OLVIDE CAMBIAR EL NOMBRE DE LA RUTA A **“/Información”**
8. Respuesta de la Ruta: Use el método `res.send(“texto a enviarse”)` para enviar el contenido del archivo **“Informacion.txt”** a los usuarios. (1 pt)
 - a. Ejemplo del método:
9. Ruta **Facultad de Sistemas**: Mediante Express va a crear una ruta a la url: **“/facultaddesistemas”** que escuche peticiones solamente con el método http **“GET”**. (1 pt)
10. El contenido de Facultad de Sistemas debe de estar una carpeta dentro del mismo directorio llamada **“public”** (1 pt)
 - a. Ejemplo:



- a. Ejemplo:
11. Se debe de configurar la carpeta **“public”** como una carpeta de archivos estáticos con el siguiente comando (1 pt)
 - a. `app.use(express.static('public'));`
 - b. Documentación: <http://expressjs.com/en/starter/static-files.html>
 - c. Ejemplo:

```
const express = require('express')_  
const app = express();  
app.use(express.static('public'))_
```

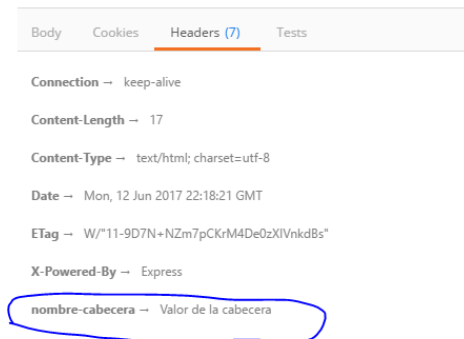
- d. **“public”** es el directorio donde se encuentra la carpeta.

12. Ruta **Cabeceras**: Mediante Express va a crear una ruta a la url: **“/cabeceras”** que escuche peticiones solamente con el método http **“POST”**. (1 pt)
13. Respuesta de ruta **Cabeceras**: Utilizar el método **“append”** para colocar las siguientes cabeceras: (6 pt)

- a. El formato de las cabeceras es así: **“nombre-cabecera: valor de la cabecera”**
- b. Ejemplo:

```
res.append('nombre-cabecera', 'Valor de la cabecera');
```

En postman responde:

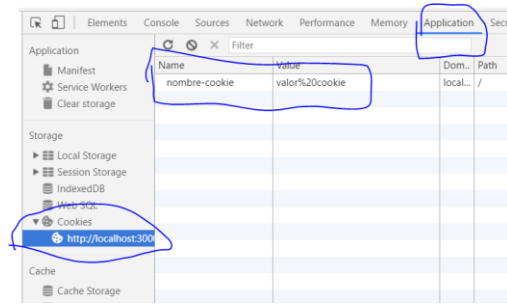


- c. Cabecera 1, enviar la hora y fecha del servidor : **“fecha: FechaServidor”** (puede utilizar **“new Date()”** para enviar la fecha del servidor)
- d. Cabecera 2, enviar el método del cliente: **“metodo: método-del-request”** (para esta cabecera va a necesitar el método **“req.method”** Documentación: <http://expressjs.com/en/4x/api.html#req.method>)
- e. Cabecera 3, enviar la ip del cliente: **“dirección-ip: ip-del-request”** (para esta cabecera va a necesitar el método **“req.ip”** Documentación: <http://expressjs.com/en/4x/api.html#req.ip>)
- f. Cabecera 4, enviar la url Original del cliente: **“url-original: url-del-request”** (para esta cabecera va a necesitar el método **“req.originalUrl”** Documentación: <http://expressjs.com/en/4x/api.html#req.originalUrl>)
- g. Cabecera 5, enviar el protocolo del cliente: **“protocolo: protocolo-del-request”** (para esta cabecera va a necesitar el método **“req.protocol”** Documentación: <http://expressjs.com/en/4x/api.html#req.protocol>)
14. Crear cookies con ruta **SetCookie**: Mediante Express va a crear una ruta a la url: **“/setcookie”** que escuche peticiones solamente con el método http **“GET”**. (1 pt)
15. Crear cookies con ruta **SetCookie**: para crear la ruta va a utilizar el método **“res.cookie(“nombre-del-cookie”, “valor del cookie”);”** (Documentación: <http://expressjs.com/en/4x/api.html#res.cookie>) (3 pt)
- a. Ejemplo:

```
res.cookie("nombre-cookie", "valor cookie");
```

1. click derecho inspeccionar
2. application, y cookies

En el navegador:

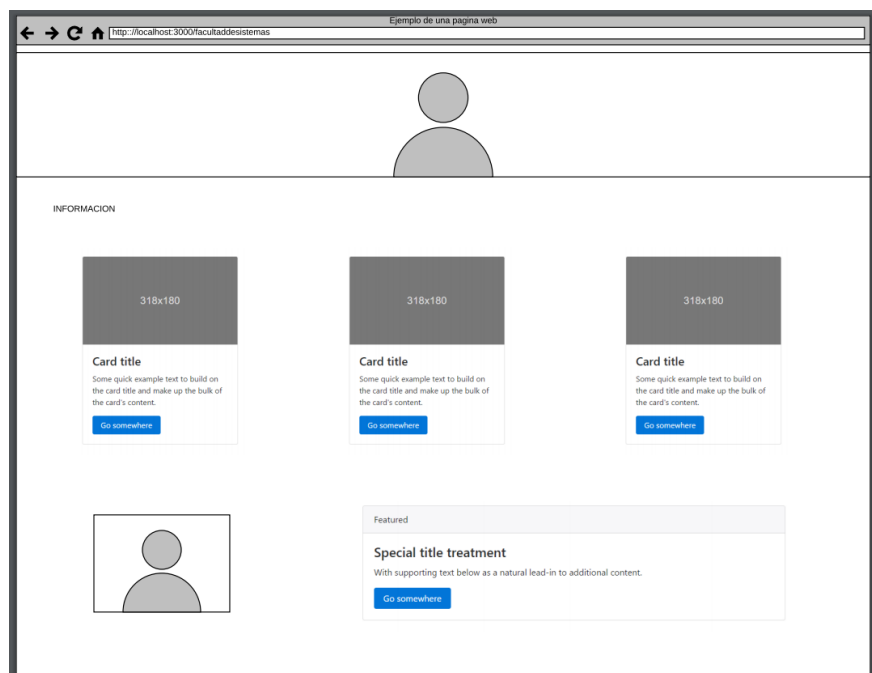


- b. La cookie va a colocar su apellido y su edad con el siguiente formato: “nombre-cookie:valor-cookie” en mi caso mi apellido es eguez y mi edad es 28, el cookie debería ser: “eguez:28”
 - c. Después de usar el método “**res.cookie()**” responder con el método “**res.send()**” el texto: “Cookie seteada”
16. Eliminar cookies con ruta **ClearCookie**: Mediante Express va a crear una ruta a la url: “/clearcookie” que escuche peticiones solamente con el método http “GET”. (1 pt)
17. Eliminar cookies con ruta **ClearCookie**: Para eliminar un cookie puede utilizar el método “**res.clearCookie()**” (Documentación: <http://expressjs.com/en/4x/api.html#res.clearCookie>) (1 pt)
- a. Ejemplo:

```
res.clearCookie("nombre-cookie");
```
 - b. Después de usar el método “**res.clearCookie()**” responder con el método “**res.send()**” el texto: “Cookie eliminada”

2. Pagina HTML

Dentro de la página HTML llamada “index.html” localizada en el directorio “**public**” va a realizar el siguiente trabajo:



1. Use bootstrap (1 pt)

2. Use una hoja de estilos con el nombre “estilos.css” dentro de la carpeta publica para sus estilos. (1 pt)
3. Linkee la hoja de estilos a su archivo “index.html” (estilos.css)

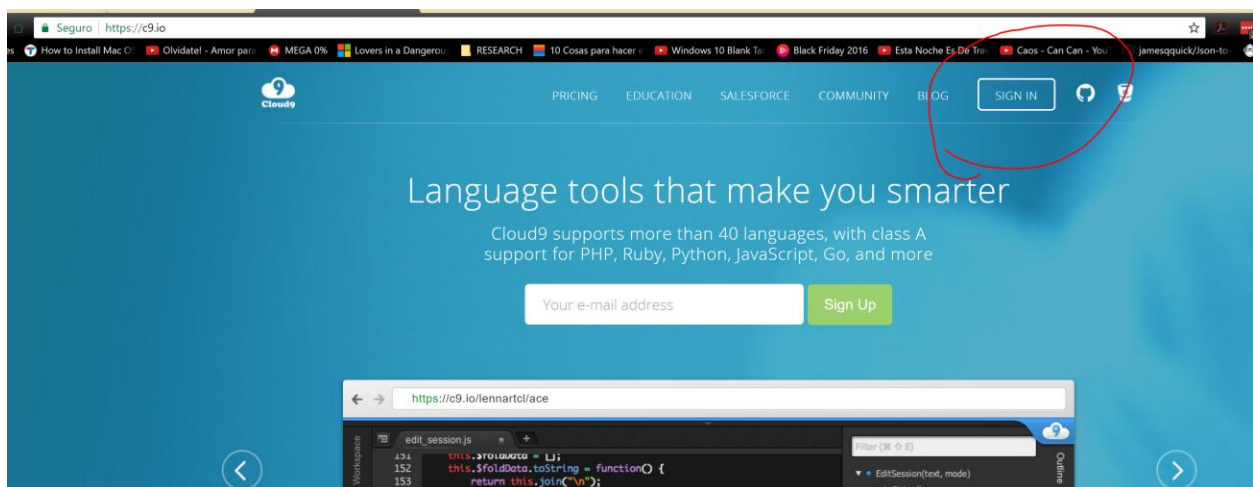
```
<link rel="stylesheet" href="estilos.css">
```
4. La imagen de cabecera debe de tener maximo una altura de 300px (utilice clases css) (1 pt)
5. La información de las cartas con imágenes debe de ser sus tres materias favoritas de la facultad, descripcion y debe ser responsive (4 pt)
6. En la parte de abajo colocar una foto de usted en la parte izquierda (puede ser la de facebook) y en la parte de la derecha otra carta con un boton de enlace hacia su facebook (si es que no tiene enviar solo a <http://www.facebook.com>). Además de esto escriba una breve descripción suya. (4 pt)
7. Utilice animaciones para cambiar el tamaño y color del subtitulo de la carta de debajo de su información personal (por ejemplo de tamaño 22 a 10 y de color rojo a azul y que vuelva a su estado original, que sea infinita la animacion) (4 pt)
8. NO OLVIDE QUE SUS ARCHIVOS DE **IMÁGENES** PUEDEN ESTAR DENTRO DE LA CARPETA “Public” Y SE LAS PUEDE REFERENCIAR.
9. LEVANTAR LA APLICACIÓN EN EL PUERTO “8080”

Publish en npm:

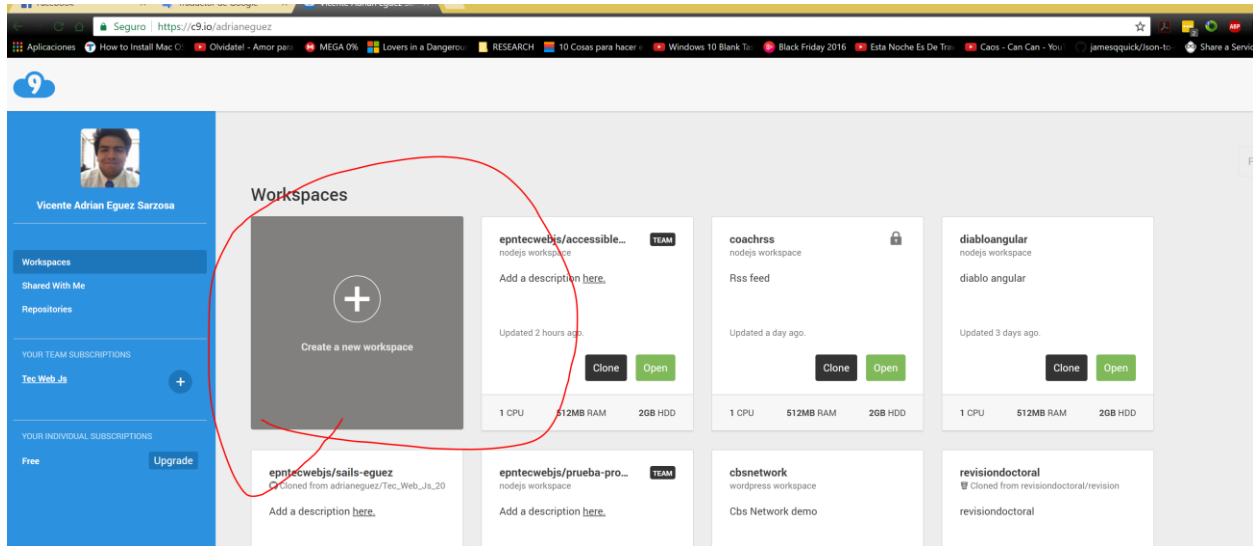
Su prueba debe de estar subida a npm, no se olvide que el comando para iniciar sesión es “npm login” y para publicar su paquete es “npm publish”.

Deploy en cloud9:

a. Logearse en Cloud 9



b. Crear un nuevo workspace en Cloud 9



EL NOMBRE DEL **WORKSPACE** DEBE DE SEGUIR EL SIGUIENTE ESTANDAR:

“examen-aplicaciones-web-”+apellido+nombre

SI MI NOMBRE ES **ADRIAN EGUEZ** ENTONCES EL REPOSITORIO DEBE DE LLAMARSE ASÍ:

“examen-aplicaciones-web-eguezadrian”

No se olviden de seleccionar el repositorio de **Node.js** y copiar el enlace de su repositorio

Create a new workspace

Workspace name

examen-aplicaciones-web-eguezadrian

Description

examen-aplicaciones-web-eguezadrian

Team

Tec Web Js

Hosted workspace

Clone workspace

Remote SSH workspace

Salesforce



Private

This is a workspace for your eyes only



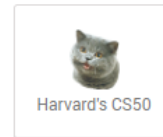
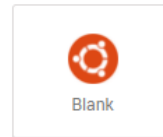
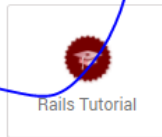
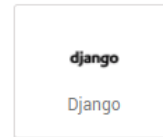
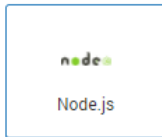
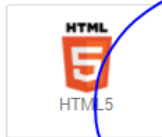
Public

This will create a workspace for everybody to see

Clone from Git or Mercurial URL (optional)

e.g. ajaxorg/ace or git@github.com:ajaxorg/ace.git

Choose a template



Create workspace

c. Instalar su paquete

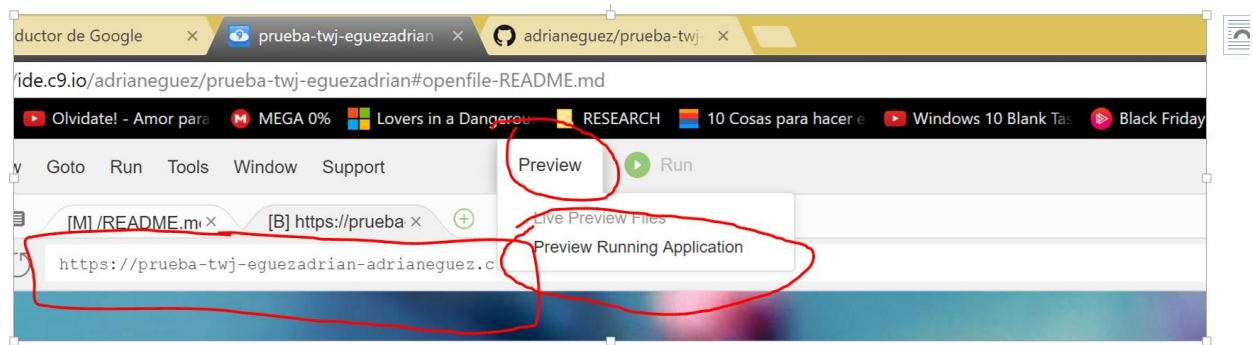
Correr el comando: "npm install aweb-examen-01-eguez-adrian" dependiendo del nombre de su paquete.

d. Pruebe su trabajo antes de terminar

Para probar su trabajo diríjase dentro de la carpeta “**node_modules**” y a su carpeta “**aweb-examen-01-eguez-adrian**”

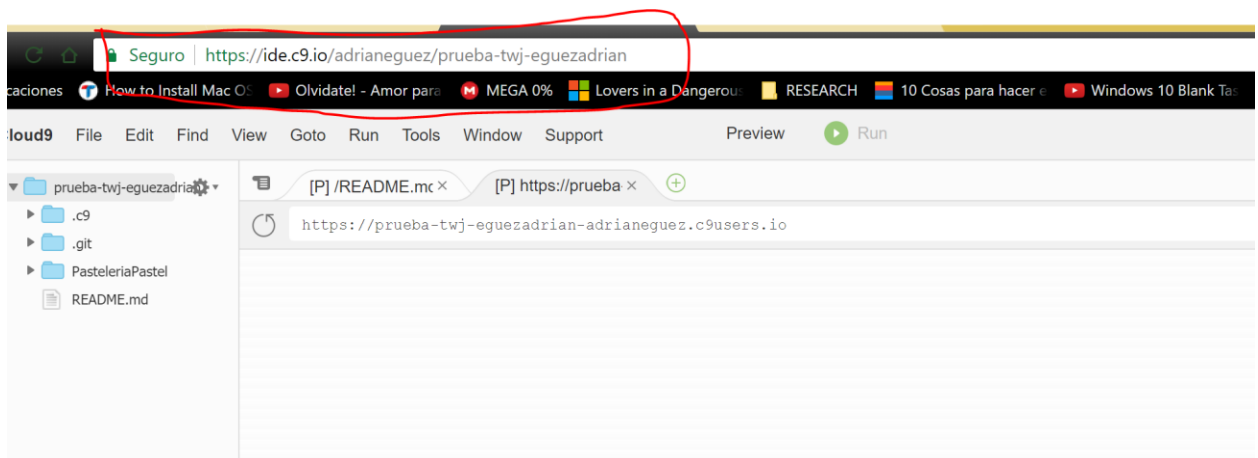
Ejemplo:

```
adrianeguez:~/workspace $ cd node_modules/  
adrianeguez:~/workspace/node_modules $ cd examen-web/  
adrianeguez:~/workspace/node_modules/examen-web $ ls  
index.js node_modules/ package.json public/  
adrianeguez:~/workspace/node_modules/examen-web $ node index.js  
Example app listening on port 3000!  
□
```



2) Subir al Aula Virtual el Link de su Workspace:

A la final usted deberá de presentar el enlace de su Workspace:



Ejemplo:

<https://ide.c9.io/adrianeguez/examen-aplicaciones-web-eguezadrian>