

Preguntas problematizadoras:  
Sin considerar el ámbito tecnológico,

¿cómo definirías a un cliente?

Sería es que da las peticiones (usuario)

¿has sido un cliente alguna vez?

si

¿Cuáles son las partes del modelo cliente-servidor?

Servidor, usuario, peticiones y respuesta

Identifica las partes del modelo cliente-servidor y relacionalas con la información presentada en cada caso.

Puedes apoyarte de las preguntas

¿Quién es el cliente? ¿Quién es el servidor?

El que compra productos o el que quiere el servicio (clientes)

el mesero y cocinero, el servidor de amazon (servidor)

¿Se te ocurre alguna otra situación con la que podrías asociar el modelo cliente-servidor?

Cuando estamos pagando en el banco, cuando pedimos algo en la tienda, cuando compramos fruta

¿Crees que sea posible que el cliente y el servidor se encuentren dentro de la misma máquina?

Si

Realiza un esquema de cómo se vería el modelo en cada una de las situaciones presentadas, incluyendo los casos de error y ok.

Usuario → petición → (si la petición es correcta) → entonces → se da el productos

|  
(si la petición es incorrecta) → retornamos el menu para que elija un nuevo producto

Usuario → petición → (si la petición es correcta) → entonces → Se vende el producto

|  
(si la petición es incorrecta) → Retornamos que el producto esta agotado

Instrucciones:

1. Elige el lugar donde deseas poner tu proyecto. Te recomendamos crearle su propia carpeta en un lugar fácil de recordar.

2. Con ayuda de tu terminal instala [node.js](#) 1. Un dato curioso de este ingrediente como recordará es que nos ayuda a correr Javascript fuera de un navegador. (Si quieres saber más sobre Javascript visita el apartado en la parte inferior de la página).

3. Verifica que la instalación haya sido exitosa y de ser así, ahora es momento de auxiliarnos de [npm](#) 2 para hacer la instalación de [express.js](#) 3 . Una forma usual de realizar la instalación es a través de: [npm](#) 4 install [express](#) 5 --save

De ser necesario ejecutar lo anterior como superusuario y espera hasta que llegué a su punto de cocción.

Una vez que ya tenemos [express.js](#) 6 listo. Procedemos a incorporar nuestro [archivo index.js](#) 7 , a la carpeta del proyecto, este archivo contiene las directivas para levantar nuestro servidor. El cual luce de la siguiente manera:

Imagen del archivo, podemos poner como ejercicio que identifiquen las partes del middleware, etc.

5. ¡Ya casi tenemos todo listo! solo falta agregar el sitio web al gusto, el primer paso para esto es crear la carpeta que albergará estos archivos dentro de nuestro proyecto. Ya que se trata de archivos que una vez que son lanzados no cambian (html y css), es decir son [estáticos](#) 8, pero que además serán visibles para todos los usuarios llamaremos a nuestra carpeta public. (Un buen chef realiza el paso anterior y las posteriores creaciones de archivos y carpetas a través de la terminal)

6. Si quieres ser más ordenado y crees estar listo para dar un paso más para perfeccionar tus habilidades en proyectos web, separa los archivos de cada tipo en carpetas diferentes: una carpeta para imágenes llamada img, una carpeta para los archivos que sirven como estructura llamada [html](#) 9 y otra para los estilos llamada [css](#) 10. (Si existieran archivos de otro tipo, por ejemplo : javascript. Tendrían su propia carpeta y hay algunas convenciones en cuanto a nombre, en este caso el nombre de la carpeta usualmente es js).

4.Nota: Si decides realizar este paso es posible que debas hacer algunas modificaciones en cuanto a rutas dentro de tus archivos html. Es importante familiarizarse con lo anterior ya que a medida que los proyectos crezcan tendrás más archivos y te será más fácil ubicarlos si se encuentran en carpetas separadas.

7. Estamos a unos pocos pasos de terminar, ya sólo debemos indicar que la carpeta que tiene nuestros archivos estáticos es public. Esto se realiza dentro de nuestro archivo index.js. A través de la siguiente línea:  
app.use(express.static('public'));

Agrega la línea y guarda los cambios.

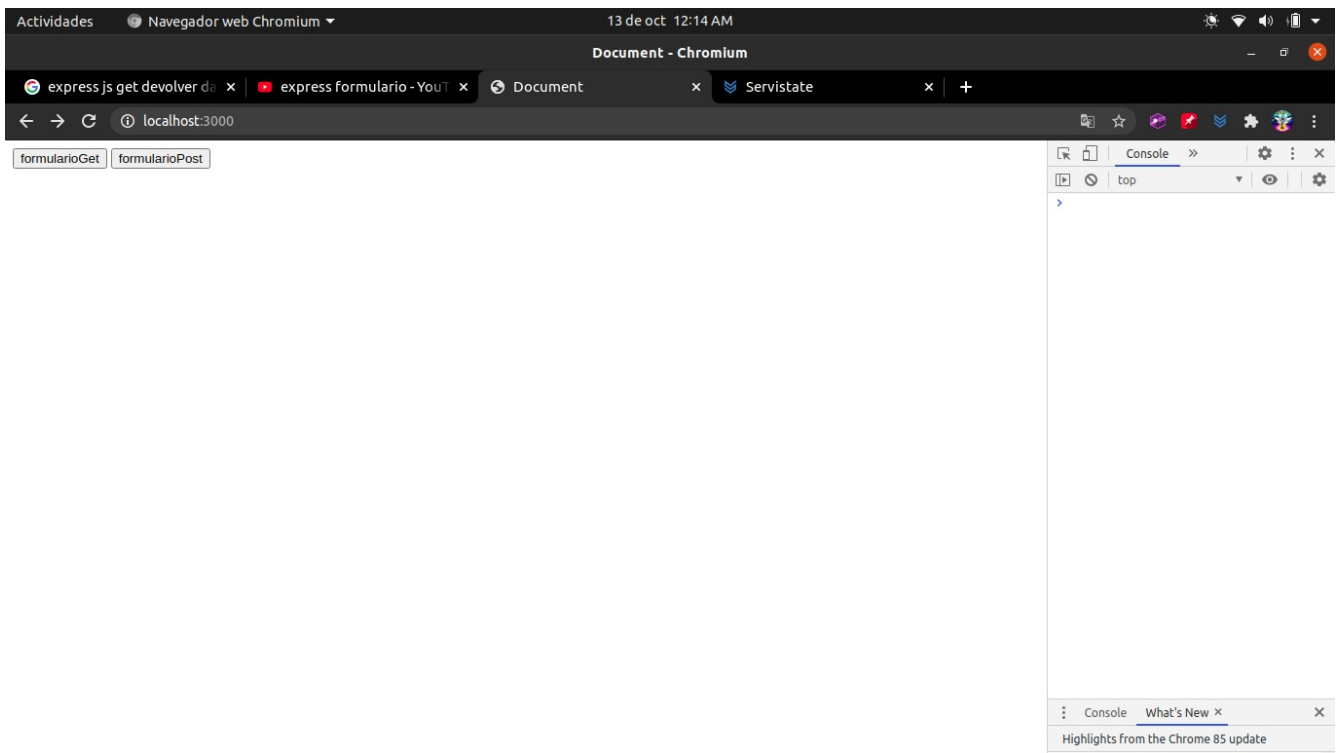
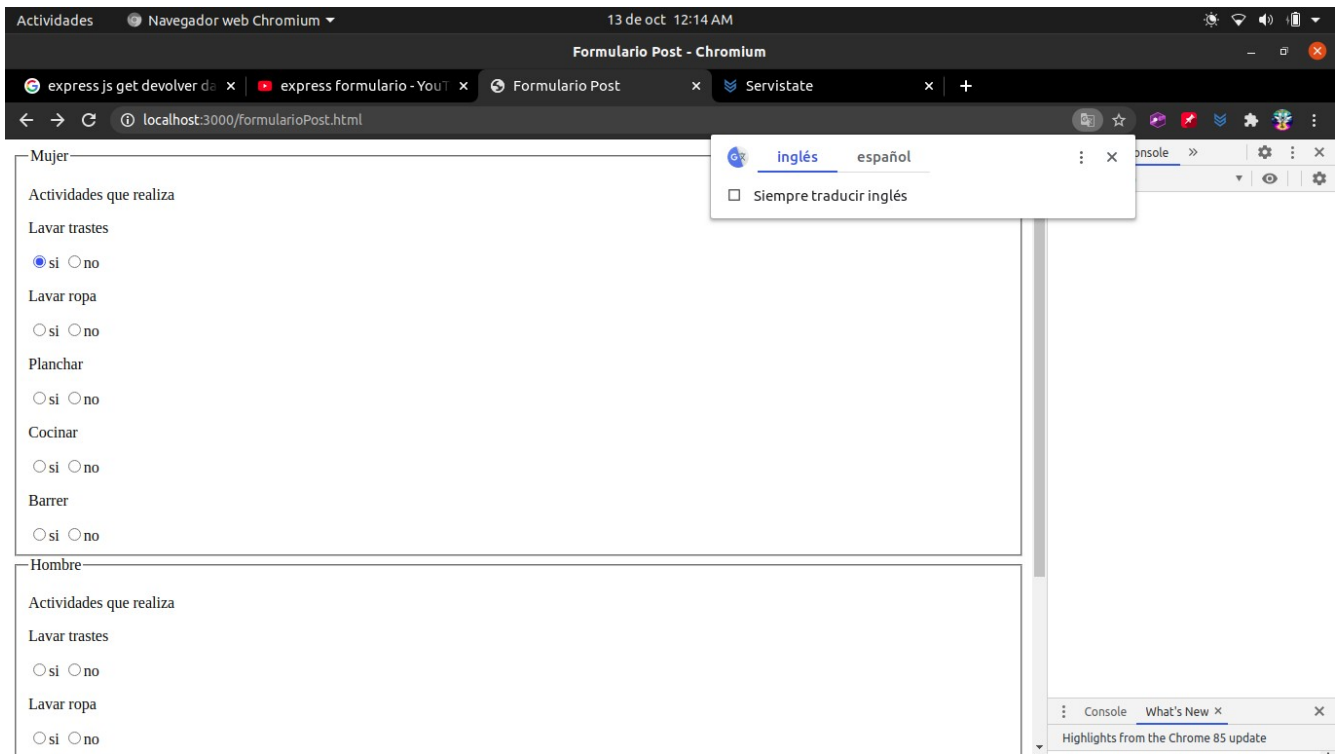
8. Ahora es momento de probar nuestra creación y verificar que tenga buen sabor. Corre el archivo [index.js](#) 11 en tu terminal a través del comando: [node index.js](#) 12.

La primera parte corresponde a lo que lo va a ejecutar (nuestro entorno de ejecución) y la segunda al archivo con terminación .js que tiene las pautas para levantar el servidor.

9. La terminal debe desplegar un mensaje donde indica que tu aplicación está corriendo en el puerto número [3000](#)13 , como lo indicamos en index.js. Ve al [localhost](#) 14 , que hace referencia a tu mismo ordenador en la dirección 127.0.0.1 de tu navegador ¡No olvides incluir el puerto! y visualiza tu sitio web desplegado.

10. Si llegaste hasta este paso es hora de que disfrutes y presumas tu creación con tu familia, amistades o con quien tu quieras. Con el pequeño detalle de que se encuentre observando la misma computadora que tú.

11. ¡El siguiente paso es conquistar el mundo! lo cual es posible a través de una plataforma de hosting que permita que todos en cualquier lugar vean tu sitio web.



Actividades Navegador web Chromium 13 de oct 12:15 AM

Formulario Get - Chromium

express.js get devolver da x express formulario - YouT x Formulario Get x Servistate x +

localhost:3000/formularioGet.html

Mujer

Actividades que realiza

Lavar trastes

☐ si ☐ no

Lavar ropa

☐ si ☐ no

Planchar

☐ si ☐ no

Cocinar

☐ si ☐ no

Barrer

☐ si ☐ no

Hombre

Actividades que realiza

Lavar trastes

☐ si ☐ no

Lavar ropa

☐ si ☐ no

Console What's New x

Highlights from the Chrome 85 update

Actividades Navegador web Chromium 13 de oct 12:15 AM

Formulario Get - Chromium

express.js get devolver da x express formulario - YouT x Formulario Get x Servistate x +

localhost:3000/formularioGet.html

☐ si ☐ no

Cocinar

☐ si ☐ no

Barrer

☐ si ☐ no

Hombre

Actividades que realiza

Lavar trastes

☐ si ☐ no

Lavar ropa

☐ si ☐ no

Planchar

☐ si ☐ no

Cocinar

☐ si ☐ no

Barrer

☐ si ☐ no

Enviar

Console What's New x

Highlights from the Chrome 85 update

Visual Studio Code interface showing the `index.js` file in the `FORMULARIOS CON EXPRESS` project. The Explorer sidebar shows the file structure. The main editor displays the following JavaScript code:

```
33
34 app.post('/formularioPost', (req,res)=>{
35     var trastes = req.body.trastes;
36     console.log(trastes);
37     console.log('hola mundo');
38     res.send("desde formulario de post"+trastes);
39
40 });
41
42 app.put('/formularioPut', function (req, res) {
43     res.send('desde put');
44 });
45
46 app.delete('/formularioDelete', function (req, res
47     res.send('desde /formularioDelete');
48 });
49
```

The status bar at the bottom indicates: Ln 38, Col 48 Spaces: 4 UTF-8 LF JavaScript Port: 5500.

Visual Studio Code interface showing the `formularioGet.html` file in the `FORMULARIOS CON EXPRESS` project. The Explorer sidebar shows the file structure. The main editor displays the following HTML code:

```
6 <title>Formulario Get</title>
7 <body>
8 <form action="/formularioGet" method="get">
9     <fieldset>
10         <legend>Mujer</legend>
11         <p>Actividades que realiza</p>
12         <p>Lavar trastes</p>
13         <input type="radio" name="trastes" value="true" >si
14         <input type="radio" name="trastes" value="false">no
15
16         <p>Lavar ropa </p>
17         <input type="radio" name="ropa" value="true">si
18         <input type="radio" name="ropa" value="false">no
19
20         <p>Planchar</p>
21         <input type="radio" name="planchar" value="true">si
22
```

The status bar at the bottom indicates: Ln 59, Col 31 Spaces: 4 UTF-8 LF HTML Port: 5500.

Actividades Visual Studio Code 13 de oct 12:16 AM

formularioPost.html - Formularios con express - Visual Studio Code

File Edit Selection View Go Run Terminal Help

formularioGet.html formularioPost.html JS index.js resultado.html in ...

formulario > formularioPost.html > html > body > form > fieldset > input

```
5 <meta name="viewport" content="width=device-width, initial-scale=1.0
6 <title>Formulario Post</title>
7 </head>
8 <body>
9 <form action="/formularioPost" method="post">
10
11 <fieldset>
12 <legend>Mujer</legend>
13 <p>Actividades que realiza</p>
14 <p>Lavar trastes</p>
15 <input type="radio" name="trastes" value="true" >si
16 <input type="radio" name="trastes" value="false" >no
17
18 <p>Lavar ropa </p>
19 <input type="radio" name="ropa" value="true">si
20 <input type="radio" name="ropa" value="false">no
```

Ln 55, Col 54 (7 selected) Spaces: 4 UTF-8 LF HTML Port : 5500

Actividades Visual Studio Code 13 de oct 12:16 AM

index.js - Formularios con express - Visual Studio Code

File Edit Selection View Go Run Terminal Help

formularioGet.html formularioPost.html JS index.js resultado.html in ...

JS index.js > app.post('/formularioPost') callback

```
1 const express = require('express')
2 const app = express()
3 const port = 3000
4 var parser = require('body-parser');
5 app.use(parser());
6 var resultadosmujeres= new Array();
7 var resultadoshombres= new Array();
8
9 app.use(express.static('formulario'));
10 var datos= [];
11 app.get('/', (req, res) => {
12
13
14 });
15 app.get('/formularioGet', (req,res)=>{
16   var trastes = req.body.trastes;
17   console.log(trastes);
```

Ln 38, Col 48 Spaces: 4 UTF-8 LF JavaScript Port : 5500

Actividades Navegador web Chromium 13 de oct 12:17 AM

Servistate - Chromium

Formulario Get x Servistate x +

Servistate HTTP Editor & REST API Client chrome-extension://mmdjghedkfbdhbjhmfbbgjaihmmhkeg/index.html#/request

SERVISTATE QUICK REQUEST REQUEST

HISTORY

- GET http://localhost:3000/formul... 200
- GET http://localhost:3000/formul... 200
- DELETE http://localhost:3000/for... 200
- DELETE http://localhost:3000/for... 404
- DELETE http://localhost:3000/for... 200
- PUT http://localhost:3000/formul... 200
- PUT http://localhost:3000/formul... 404
- POST http://localhost:3000/form... 200
- POST http://localhost:3000/form... 404
- POST http://localhost:3000/ 200
- POST http://localhost:3000/form... 404

REQUEST Save Why Preview

GET http://localhost:3000/formularioGet Send

HEADERS URL PARAMS REDIRECTS: ON DETAILS STATS ASSERTIONS ASSIGNMENTS

Header Value Add

Send Request

RESPONSE STATUS 200 OK 16 B 14 ms

BODY RESPONSE HEADERS REQUEST HEADERS COOKIES HTTP

FORMATTED RAW PREVIEW Copy

1 esto esundefined

VARIABLES Add

Actividades Navegador web Chromium 13 de oct 12:17 AM

Servistate - Chromium

Formulario Get x Servistate x +

Servistate HTTP Editor & REST API Client chrome-extension://mmdjghedkfbdhbjhmfbbgjaihmmhkeg/index.html#/request

SERVISTATE QUICK REQUEST REQUEST

HISTORY

- GET http://localhost:3000/formul... 200
- POST http://localhost:3000/form... 200
- GET http://localhost:3000/formul... 200
- DELETE http://localhost:3000/for... 200
- DELETE http://localhost:3000/for... 404
- DELETE http://localhost:3000/for... 200
- PUT http://localhost:3000/formul... 200
- PUT http://localhost:3000/formul... 404
- POST http://localhost:3000/form... 200
- POST http://localhost:3000/form... 404
- POST http://localhost:3000/ 200
- POST http://localhost:3000/form... 404

REQUEST Save Why Preview

POST http://localhost:3000/formularioPost Send

HEADERS URL PARAMS REDIRECTS: ON DETAILS STATS ASSERTIONS ASSIGNMENTS

Header Value Add

FORM DATA URL-ENCODED RAW

Key Value Add

Send Request

RESPONSE STATUS 200 OK 33 B 16 ms

BODY RESPONSE HEADERS REQUEST HEADERS COOKIES HTTP

FORMATTED RAW PREVIEW Copy

VARIABLES Add



Actividades Navegador web Chromium 13 de oct 12:18 AM

Servistate - Chromium

Formulario Get x Servistate x +

Servistate HTTP Editor & REST API Client chrome-extension://mmdjghedkfbdhbjhmfefbgjaihmmhkeg/index.html#/request

SERVISTATE QUICK REQUEST REQUEST

HISTORY

- POST http://localhost:3000/form... 200
- PUT http://localhost:3000/formul... 200
- GET http://localhost:3000/formul... 200
- GET http://localhost:3000/formul... 200
- DELETE http://localhost:3000/for... 200
- DELETE http://localhost:3000/for... 404
- DELETE http://localhost:3000/for... 200
- PUT http://localhost:3000/formul... 200
- PUT http://localhost:3000/formul... 404
- POST http://localhost:3000/form... 200
- POST http://localhost:3000/form... 404
- POST http://localhost:3000/ 200
- POST http://localhost:3000/form... 404

REQUEST Save Why Preview

PUT http://localhost:3000/formularioPut Send

HEADERS URL PARAMS REDIRECTS: ON DETAILS STATS ASSERTIONS ASSIGNMENTS

Header Value Add

FORM DATA URL-ENCODED RAW

Key Value Add

Send Request

RESPONSE STATUS 200 OK 9 B 11 ms

BODY RESPONSE HEADERS REQUEST HEADERS COOKIES HTTP

FORMATTED RAW PREVIEW Copy

VARIABLES Add

Actividades Navegador web Chromium 13 de oct 12:18 AM

Servistate - Chromium

Formulario Get x Servistate x +

Servistate HTTP Editor & REST API Client chrome-extension://mmdjghedkfbdhbjhmfefbgjaihmmhkeg/index.html#/request

SERVISTATE QUICK REQUEST REQUEST

HISTORY

- PUT http://localhost:3000/formul... 200
- DELETE http://localhost:3000/for... 200
- POST http://localhost:3000/form... 200
- GET http://localhost:3000/formul... 200
- GET http://localhost:3000/formul... 200
- DELETE http://localhost:3000/for... 200
- DELETE http://localhost:3000/for... 404
- DELETE http://localhost:3000/for... 200
- PUT http://localhost:3000/formul... 200
- PUT http://localhost:3000/formul... 404
- POST http://localhost:3000/form... 200
- POST http://localhost:3000/form... 404
- POST http://localhost:3000/ 200
- POST http://localhost:3000/form... 404

REQUEST Save Why Preview

DELETE http://localhost:3000/formularioDelete Send

HEADERS URL PARAMS REDIRECTS: ON DETAILS STATS ASSERTIONS ASSIGNMENTS

Header Value Add

FORM DATA URL-ENCODED RAW

Key Value Add

Send Request

RESPONSE STATUS 200 OK 23 B 11 ms

BODY RESPONSE HEADERS REQUEST HEADERS COOKIES HTTP

FORMATTED RAW PREVIEW Copy

VARIABLES Add

