**Anexo (MWT2A11\_Anexo)**

**Caso 1**

Te encuentras en un restaurante en el que tu mesa tendrá asociado a un mesero.

Una vez que has leído la carta y decidiste que ordenar, entonces llamas al mesero

para realizar una PETICIÓN de la comida que has elegido.

Posteriormente, el mesero lleva tu pedido para que lo preparen y obtiene tu comida

una vez que es preparada (a la que con fines de este ejemplo llamaremos

RECURSO). Una vez que el mesero tiene tu RECURSO, debe volver a tu mesa

para retornar una RESPUESTA, pensemos en 2 casos particulares:

● El restaurante se ha quedado sin los ingredientes necesarios para

preparar tu platillo. La RESPUESTA que recibirás del mesero será

negativa, es decir ha ocurrido un ERROR y en consecuencia no

podrás obtener tu RECURSO.

● La situación contraria es en la que todo ha salido bien y obtendremos

una respuesta positiva del mesero, a la que llamaremos, por ejemplo,

OK. De manera que el RECURSO que pediste llegará a ti para que

puedas consumirlo.

**Caso 2**

Quieres pedir un artículo en una tienda en línea, pensemos por ejemplo en Amazon.

Vas a la página y encuentras lo que estás buscando e incluso cosas que no estás

buscando. Decides hacer una PETICIÓN de un producto.

Consideremos el caso aislado en el que existe una oferta muy llamativa en ese

producto que estás ordenando y muchas personas quieren ordenarlo al mismo

tiempo también. Al presionar click sobre comprar pueden ocurrir 2 cosas:

● Si aún hay artículos en el stock, te dejará seguir con el proceso y como

RESPUESTA te dirá que todo se ha llevado a cabo de manera

correcta, es decir OK y posteriormente a tu casa llegará el RECURSO

que ordenaste.

● Si fuiste desafortunado y se ha agotado el producto, únicamente se te

indicará que ya no hay más productos, lo que finalmente asociaremos

con un ERROR ya que no se te enviará el RECURSO que has

solicitado.

**Preguntas problematizadoras:**

● Sin considerar el ámbito tecnológico, ¿cómo definirías a un cliente? ¿has sido

un cliente alguna vez?

Un cliente es aquella parte que solicita “PETICIONES” o en este caso que se realice una tarea especial para obtener un servicio o recurso (producto), si he sido cliente por ejemplo me gusta comer sushi entonces en el restaurante como cliente solicito al camarero mi pedido el cual informara a el chef para poder realizarlo y el mesero pueda entregarme mi “PETICIÓN”.

También por ejemplo he comprado productos en línea en donde soy un cliente y solicito a la página poder corroborar mi pago para obtener así un producto.

● ¿Cuáles son las partes del modelo cliente-servidor? y con tus propias

palabras define cada una de ellas.

Las dos partes principales son: el cliente que para mí es aquella parte que emite solicitudes, por ejemplo solicita un recurso y que puede proporcionar información relevante para que este recurso pueda ser obtenido como por ejemplo calcular el costo de envió de un paquete únicamente ingresando mi código postal, en este caso el cliente proporciona el código postal y le solicita al servido un recurso que en este caso es un costo, el servidor es la otra gran parte de este modelo y es aquella que procesa información en este caso es quien realiza el calculo del costo para nuevamente poder ser enviado hacia el cliente, maneja toda la parte conocida como “Lógica de negocio”.

● Identifica las partes del modelo cliente-servidor y relaciónalas con la

información presentada en cada caso. Puedes apoyarte de las preguntas

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

**Caso 1**

En el caso 1 nosotros como comensales del restaurante somos el cliente, los cuales realizamos una petición al mesero que en este caso es el mesero el cual toma nota y lo lleva al chef a quien le pregunta si existen los ingredientes necesarios “recursos” para elaborar el platillo y que así el mesero pueda entregarnos una respuesta a nuestra petición, el chef ya con la información del mesero determina si es posible o no crear el platillo y con ello darle una respuesta al mesero que es la que nuevamente nos entregará, en la cual puede ser que si nos traiga el platillo ya que si hay existencia de los elementos necesarios o una respuesta negativa en donde nos menciona que no es posible realizar dicho platillo.

Nosotros = cliente

Chef = servidor

Mesero = Intermediario o código intermedio

**Caso 2**

En el segundo casi de igual manera nosotros somos el cliente, la página de Amazon es el intermediario que me permite ingresar la información necesaria para realizar una petición (en este caso comprar un producto) y el servidor de Amazon es la parte que revisa si existe en inventario o no el articulo que deseamos comprar, para darnos una respuesta afirmativa en caso de poder comprar el producto o negativa en caso de que marque un aviso de producto fuera de stock.

Nosotros = cliente

Servidores de Amazon = servidor

Vista de la página = intermediario entre el cliente y el servidor

● ¿Se te ocurre alguna otra situación con la que podrías asociar el modelo

cliente-servidor?

Creo que puede ser aplicado en muchas partes como por ejemplo negocios físicos de venta, negocios online de venta (como Amazon), así como los cajeros automáticos de los bancos.

● ¿Crees que sea posible que el cliente y el servidor se encuentren dentro de la

misma máquina?

Siguiendo la analogía de los caso anteriores diría que no se puede o simplemente no es deseable que el cliente y el servidor se encuentren dentro de la misma máquina ya que eso puede causar grandes conflictos, por ejemplo: en una computadora normal guardar toda la información que tiene Amazon sería imposible, si fuera el caso que pudiéramos acceder a la información del lado del cliente, seria algo muy negativo para la empresa ya que podríamos tener acceso a toda la información personal de sus clientes como números de tarjeta direcciones y eso es para nada deseable.

Aunque siento que se podría para etapas de prueba de un software solamente para ver cómo están funcionando las cosa y aun así lo creo poco probable.

● Realiza un esquema de cómo se vería el modelo en cada una de las

situaciones presentadas, incluyendo los casos de error y ok.

**Caso 1**

Petición: Solicitar platillo

Mesero (Intermediario)

Cliente

En caso de si existir los ingredientes necesarios el mesero llevará el platillo al cliente.

Llevar datos de petición

Ok

Chef (Servidor)

Mediante los datos obtenidos por el mesero determina si se puede realizar el platillo o no

Error

Por otro lado, si no están los suficientes ingredientes el mesero informará al cliente que no es posible realizar el platillo

**Caso 2**

Formulario de Amazon (Intermediario)

Petición: Comprar producto

Cliente

En caso de si existir el producto la interfaz mostrará que la compra se realizó exitosamente.

Llevar datos de petición

Ok

Servidor de Amazon (Servidor)

Mediante los datos obtenidos por el servidor determina si existe o no stock del producto.

Error

Por otro lado, si no hay stock del producto la interfaz mandará un mensaje de error informando que no hay en existencia.

**Segunda Parte**

Servidor a la Express.js / Servidor estilo Express.js

**Ingredientes:**

* Sitio web al gusto
* Archivo index.js
* node.js
* npm
* express.js

**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.js1. Un dato curioso de este

ingrediente como recordarás 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 npm2 para hacer la instalación de express.js3. Una forma

usual de realizar la instalación es a través de: npm4 install express5 --save

De ser necesario ejecutar lo anterior como superusuario y espera hasta que

llegué a su punto de cocción.

**4.** Una vez que ya tenemos express.js6 listo. Procedemos a incorporar

nuestro archivo index.js7 ,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 estáticos8, 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 HTML9 y otra

para los estilos llamada CSS10. (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).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.js11 en tu terminal a través del comando:

node index.js12.

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 300013, como lo indicamos en index.js. Ve al

localhost14, 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.