

Teoría:

Preguntas generales sobre HTTP/HTTPS:

¿Qué es HTTP y cuál es su función principal?

HTTP es un protocolo que sirve para poder tener una comunicación entre el cliente y el servidor, pudiendo definir como se envían los datos para que se pueda navegar y/o acceder a distintos sitios web.

¿Cuál es la diferencia entre HTTP y HTTPS?

La diferencia entre HTTP y HTTPS es la seguridad en el envío de datos, ya que HTTPS utiliza un paso adicional a HTTP que encripta los datos para su mayor seguridad.

¿Cómo funciona el proceso de cifrado en HTTPS?

El proceso de cifrado en HTTPS funciona de manera que encripta las conversaciones cliente/servidor para garantizar su seguridad.

¿Qué es un certificado SSL/TLS y cuál es su importancia en HTTPS?

El certificado SSL/TLS es un archivo digital que verifica la identidad y la autenticidad de un sitio web. Su importancia en el uso de HTTPS es que al usar ya sea SSL o TLS estos te aseguran que la comunicación cliente servidor sea segura encriptando los datos enviados luego de su validación.

¿Que es un método HTTP? ¿Podrías enumerar algunos de los mas utilizados?

Un método HTTP es una acción que el cliente solicita al servidor mediante una solicitud web para realizar una tarea específica como obtener, enviar, actualizar o eliminar.

algunos de los métodos mas utilizados son:

GET(sirve para solicitar datos).

POST(sirve para enviar datos).

Explica las diferencias entre los métodos HTTP GET y POST.

La diferencia entre estos métodos es que con GET puedes solicitar datos al servidor. cómo visitar alguna página web. al hacer esto los parámetros enviados se verán reflejados en la URL. En cambio, con POST se utiliza para el envío de datos como puede serlo un formulario. este se envía mediante un cuerpo de solicitud que si se combina junto con HTTPS sus datos serán cifrados.

¿Qué es un código de estado HTTP? ¿Podrías mencionar algunos de los más comunes y lo que significan?

Un código de estado es un número de 3 dígitos que el servidor envía en respuesta a una petición http. Este confirma el resultado de la petición en donde informa si se proceso correctamente o si hubo un error. Los más comunes pueden ser los mensajes de error como 404 Not Found(la url especificada no se encuentra en el servidor). 503 Service Unavailable(el servidor no se encuentra disponible temporalmente debido a una sobrecarga o mantenimiento).

¿Qué es una cabecera HTTP?

Las cabeceras HTTP son un conjunto de datos que se intercambian entre el cliente y el servidor como lo pueden ser las cookies o la autenticación. un ejemplo de cabecera podrían ser las cookies que sirven para mantener la información del estado o sesión del usuario.

¿Qué es un redirect (redirección) HTTP y cuándo es utilizado?

Un redirect HTTP es una forma de indicar que un recurso o pagina web ha sido movido a otra ubicación dando como respuesta una URL distinta.

Preguntas técnicas y de seguridad en HTTP/HTTPS:

¿Cómo se asegura la integridad de los datos en una conexión HTTPS?

La integridad de los datos en una conexión HTTPS se asegura mediante el archivo digital SSL/TLS la encriptación de datos y verifica con MAC que los datos enviados no hayan sido modificados(si esto pasa MAC alerta al receptor de que el mensaje fue alterado).

¿Qué diferencia hay entre una cabecera Authorization y una cabecera Cookie?

La cabecera se usa para enviar credenciales de autenticación en cada solicitud mientras que la cookie se utiliza para mantener la sesión del usuario.

Preguntas de implementación práctica:

¿Qué problemas podrías enfrentar al trabajar con APIs que dependen de HTTP, y cómo los resolverías?

Uno de los problemas que tendría es la transferencia literal de datos, ya que estos no se encriptan antes de ser enviados pueden ser obtenidos por terceros ya que http no cuenta con temas de seguridad. una de las formas para poder resolver este problema seria actualizar las URL de la API asi son redirigidas a HTTPS

¿Qué es un cliente HTTP?

Un cliente HTTP es una aplicación o sitio web que envia solicitudes a un servidor HTTP para interactuar con dicho servicio.

¿Mencionar la diferencia entre los clientes POSTMAN y CURL?

La diferencia entre estas es que Postman utiliza una interfaz grafica para hacer pruebas o desarrollar APIs mientras que con cURL lo utilizan mediante la linea de comandos

Preguntas de GIT

¿Qué es GIT y para qué se utiliza en desarrollo de software?

Git es un sistema de control que utilizan los desarrolladores para poder tener un seguimiento en los cambios realizados en su proyecto.

¿Cuál es la diferencia entre un repositorio local y un repositorio remoto en GIT?

Un repositorio local quiere decir un repositorio que solo se encuentra alojado en tu PC sin necesidad de conexión a internet, mientras que un repositorio remoto es uno que se encuentra en la nube(como GitHub), alojado en un servidor siendo accesible con internet.

¿Cómo se crea un nuevo repositorio en GIT y cuál es el comando para inicializarlo? Explica la diferencia entre los comandos git commit y git push.

Para crear un repositorio en git primero necesitas tener una carpeta con el cual vas a tener almacenado el proyecto, una vez seleccionada esa carpeta, abres git bash ubicado en ese archivo y pones el comando "git init"

"git commit" es un comando para poder guardar los cambios en tu repositorio local, esto sirve por si estas haciendo muchos cambios en tu proyecto lo puedes dividir por secciones con commits con un mensaje de la tarea terminada como "formularios terminados" por otro lado "git push" se usa para poder subir todos los cambios al repositorio remoto necesitando

conexión a internet para poder efectuar tal comando.

¿Qué es un "branch" en GIT y para qué se utilizan las ramas en el desarrollo de software?

"branch" en git es una manera de poder separar en secciones tu repositorio como lo pueden ser una branch de "login", "eventos", "alguna tarea especificada" la cual nos facilita a la hora de separar las tareas así los colaboradores pueden trabajar de una manera aislada teniendo su tarea en su rama especificada. Luego de tener todas las tareas completadas, estas se suben a la rama principal "main/master" así se ven todos los cambios reflejados juntos

¿Qué significa hacer un "merge" en GIT y cuáles son los posibles conflictos que pueden surgir durante un merge?

Hacer un merge en git significa combinar dos ramas como por ej hacer merge de la rama "login" a la rama "main" para poder subir de manera segura los cambios hechos desde la rama login hacia la rama main. Los conflictos en merge pasan cuando dos ramas tienen cambios que están en el mismo archivo y que git no puede resolver automáticamente, siendo este el caso git te mostrara donde se genera el conflicto dando indicadores donde que archivo pertenece a tal rama así lo resuelve el desarrollador.

¿Cómo se deshace un cambio en GIT después de hacer un commit pero antes de hacer push?

para deshacer un commit en git antes de hacer push se ejecuta el siguiente comando
git reset (--soft, --hard, --mixed) HEAD~1

soft mantiene los cambios para poder hacer modificaciones

hard para eliminar todos los cambios

mixed mantiene los cambios en tu carpeta pero los elimina del area de staging

¿Qué es un "pull request" y cómo contribuye a la revisión de código en un equipo?

"pull request" es una función de GitHub que te permite solicitar una revision y aprobacion de los cambios realizados en una rama para poder ser fusionados mediante el merge a otra rama como lo puede ser "main/master"

¿Cómo puedes clonar un repositorio de GIT y cuál es la diferencia entre git clone y git pull?

para clonar un repositorio de GIT lo que necesitas es abrir la git bash para poder poner el comando git clone seguido de la url del repositorio de GitHub al que quieres clonar.
git clone <https://github.com/HidalgoBrian/RepoPrueba.git>

las diferencias entre git clone y git pull son que con git clone puedes clonar un

repositorio junto con todas sus ramas y archivos mientras que con git pull puedes actualizar una rama de tu repositorio local

¿Qué es Node.js y por qué es una opción popular para el desarrollo backend?

Node.js es un entorno de ejecución que permite utilizar JavaScript en el servidor sin necesidad de un navegador. Esto sirve para crear aplicaciones del lado del servidor usando JavaScript

Node.js es una opción popular porque es fácil de aprender ya que se basa en uno de los lenguajes más conocidos en todo el mundo, por su manejabilidad tanto del servidor como del cliente ya que se basa en código JavaScript y tiene un gran rendimiento porque cuenta con el motor v8 que utiliza google chrome que es conocido por su alta velocidad de ejecución.

¿Cuál es la diferencia entre require() y import en Node.js?

require() es la forma tradicional de importar módulos de manera sincrónica con node la cual utiliza con CommonJs que es el sistema de módulos que se utilizaba antes de introducir los módulos modernos.

por otro lado, import es la versión actualizada de importar de manera asincrónica los módulos ya que cuentan con los módulos modernos de JavaScript.

¿Qué es npm y cuál es su función en el ecosistema de Node.js?

npm es una herramienta importante para node.js que sirve para gestionar las dependencias utilizadas en el proyecto como las librerías y/o módulos.

¿Cómo se inicializa un proyecto de Node.js usando npm y cuál es el propósito del archivo package.json?

para inicializar un proyecto con npm lo se debe hacer es abrir la terminal para colocar el siguiente comando

“npm init” ó “npm init -y”

El primer comando te da una serie de preguntas para definir el proyecto como el nombre, versión, descripción y al final creará un archivo package.json con los datos proporcionados. en cambio con el segundo comando con “-y” te dará todos los valores por defecto.

el archivo package.json sirve para gestionar las dependencias, obtener datos del proyecto, verificar las versiones de las dependencias y definir script personalizados (como comandos para ejecutar pruebas)

¿Qué son las dependencias en npm y cómo se instalan?

Las dependencias en npm son librerías o módulos que el proyecto necesita para funcionar. Para instalar estas dependencias se utiliza un comando en la terminal como “npm install”, “npm install --only=dev”. El primer comando es para instalar las dependencias de producción y el segundo es solo para las dependencias de desarrollo.

¿Cómo puedes gestionar versiones específicas de paquetes en npm y para qué sirve el archivo package-lock.json?

Para instalar una versión específica en npm se puede usar el comando “npm install paquete@versión” que te dará la versión proporcionada. seguido de esto puedes configurar el package.json y en las versiones de dependencias configurarlo con “^” delante de la versión para poder actualizar automáticamente la versión sin sumar el primer número. Ej: 4.0.0 si le ponemos el ^ al principio solo actualiza justo antes de la 5.0.0

El package-lock-json contiene las versiones exactas del proyecto. Este archivo se genera automáticamente cuando hacemos un npm install. Su tarea es asegurar que las versiones de las dependencias se mantengan iguales para evitar problemas de compatibilidad.

¿Qué es nest.js cómo se usa en Node.js para construir aplicaciones backend?

Nest.js es un framework de Node.js que se utiliza para desarrollar aplicaciones backend. Este framework está construido con TypeScript pero se puede utilizar con JavaScript. en todo caso nest facilita el uso de node ya que nos da una estructura en formato mvc y un código fácil de entender y mantener.

¿Cómo se manejan errores en Node.js y cuál es la diferencia entre callbacks, promesas y async/await para manejar código asíncronico?

En Node.js los errores se manejan de distintas maneras según el estilo asíncronico que se use. en el caso de Callbacks es el más tradicional en el cual le pasas primero el argumento de error, en las promesas es más legible y permite concatenar todo hacia el error(pos si pasa algo ya te redirige hacia el error) y luego está async/await que hace que el código asíncronico se lea como sincrónico, y simplifica el manejo de errores.

Actividad practica número 1:

¿Qué sucede cuando hacemos el GET por segunda vez, luego de haber ejecutado el POST?

cuando hice el get por segunda vez luego de hacer ejecutar el post puedo visualizar mis datos proporcionados en el archivo json

GET https://reclutamiento-dk POST https://reclutamiento-dk + No environment

https://reclutamiento-dev-procontacto-default-rtdb.firebaseio.com/reclutier.json Save Share

GET https://reclutamiento-dev-procontacto-default-rtdb.firebaseio.com/reclutier.json Send

Params Authorization Headers (6) Body Scripts Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

Body Cookies Headers (8) Test Results 200 OK 247 ms 1.39 KB

Pretty Raw Preview Visualize JSON

```
49  },
50  "-0BbaDK_MPrkwt9JL0cF": {
51    "age": 22,
52    "birthday": "2001/12/18",
53    "documentNumber": 43863355,
54    "documentType": "DNI",
55    "name": "Ivone",
56    "surname": "Corleto"
57  },
58  "-0Bc1lfarf9LnRSn0Xo4": {
59    "age": 22,
60    "birthday": "2002/06/30/",
61    "documentNumber": 44318250,
62    "documentType": "DNI",
63    "name": "Brian",
64    "suraname": "Hidalgo"
65  }
66 }
```

Home Workspaces API Network Search Postman Invite Upgrade

My Workspace New Import GET https://reclutamiento-dk +

https://reclutamiento-dev-procontacto-default-rtdb.firebaseio.com/reclutier.json Save Share

GET https://reclutamiento-dev-procontacto-default-rtdb.firebaseio.com/reclutier.json Send

Params Authorization Headers (6) Body Scripts Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

Body Cookies Headers (8) Test Results 200 OK 910 ms 1.25 KB

Pretty Raw Preview Visualize JSON

```
1  {
2    "-0BbUVxgEqJYmZ0Z1Ad": {
3      "age": 30,
4      "birthday": "16/11/1993",
5      "documentNumber": 12345678,
6      "documentType": "Dni",
7      "name": "Lucas",
8      "suzaname": "Seguza"
9    },
10   "-0BbUZ14k40h_VGPOMg6": {
11     "age": 29,
12     "birthday": "16/09/1995",
13     "documentNumber": 20123456781,
14     "documentType": "CUIT",
15     "name": "Juan",
16     "suzaname": "Perez"
17   },
18   "-0BbUc6GkeikiUMoIfEE": {
19     "age": 29,
20     "birthday": "16/09/1995",
21     "documentNumber": 20123456781,
22     "documentType": "CUIT",
23     "name": "Juan",
24     "suzaname": "Perez"
25   }
26 }
```

Postbot Ctrl Alt P

Online Find and replace Console Postbot Runner Start Proxy Cookies Vault Trash