Sara Maria Ramírez Pérez
U0276188

SDI - Segundo parcial

7-3-22

¿Qué es Node.js?

a. Un servidor web multiplataforma
b. Un lenguaje de programación
c. Un Framework que permite ejecutar JS del lado del servidor
d. Una plataforma de software que permite ejecutar JS del lado del servidor

¿Cómo gestiona Node.js las operaciones I/O?

a. De forma síncrona y sin bloqueo
b. De forma asíncrona y sin bloqueo
c. De forma asíncrona y sólo se bloquean algunas operaciones
d. De forma síncrona, asíncrona y con bloqueo

¿Qué tipo de escalabilidad usa NodeJS para las aplicaciones?

a. Horizontal
b. Vertical
c. Transversal
d. Perpendicular

NodeJS mejora la concurrencia porque su arquitectura se basa:

a. Ejecución multihilo con un bucle de eventos
b. Ejecución multihilo y un modelo I/O asíncrono
c. Un solo hilo de ejecución con un bucle de eventos
d. Un solo hilo de ejecución sin bucles de eventos

Son características de una comunicación asíncrona:

a. Las operaciones no son bloqueantes
b. Las operaciones se ejecutan de forma secuencial
c. Las operaciones son bloqueantes
d. Se realizan mediante el sistema de callback (retrollamada) y/o promesas

14-3-22

Es un framework de desarrollo de aplicaciones web minimalista y flexible para Node.js:

a. NPM
b. Express
c. Swig
d. Javascript

En una aplicación Node.js, el fichero package.json contiene:

a. La configuración y los metadatos de la aplicación
b. El nombre del servidor
c. Las dependencias
d. Todos los ficheros JS de la aplicación

Sara Maria Ramirez Pérez
U0276188

En NodeJS la función require se utiliza para:

a. Crear una nueva aplicación Node
b. Definir módulos
c. Eliminar módulos de la aplicación
d. Importar módulos

¿Cómo se denominan las funciones que se ejecutan cuando se recibe la petición en una app Node?

a. App (Aplicaciones)
b. Handler (Manejadores)
c. Routing (Enrutamiento)
d. Module (Módulos)

Forma de declarar un módulo en Node.js:

a. module.exports=function(){…}
b. module.imports=function(){…}
c. module.exports={…}
d. module.require=function(){…}

Forma de incluir un módulo en Node.js:

a. module.require=function(){…}
b. require("./routes/users.js")(app);
c. var express=require('express');
d. module.import={}

Para obtener los valores de los parámetros enviados en una petición POST en NodeJS se utiliza:

a. req.body.<clave_parámetro>
b. req.params.<clave_parámetro>
c. req.query.<clave_parámetro>
d. req.form.query.<clave_parámetro>

Para obtener los valores de los parámetros enviados en una petición GET en NodeJS se utiliza:

a. req.body.<clave_parámetro>
b. req.params.<clave_parámetro>
c. req.query.<clave_parámetro>
d. req.form.query.<clave_parámetro>

Una plantilla Swig recibe un atributo con clave "nombre" ¿Cómo se insertaría el valor de ese atributo en un elemento H1?

a. <h1>{nombre}</h1>
b. <h1>{{nombre}}</h1>
c. <h1>${nombre}</h1>
d. <h1>#{nombre}</h1>

21-3-22

¿Cuál es la principal diferencia al definir un módulo como un objeto y no como una clase o función en una aplicación Node.js?

- **a. Al incluir el módulo varias veces, todos retornan la referencia al mismo objeto**
- b. Al incluir el módulo varias veces, todos retornan la referencia a diferentes objetos.
- c. Al incluir el módulo varias veces, no hay diferencia al definir módulos como clases, funciones u objetos.

¿Cuál sería la mejor alternativa para utilizar un módulo desde otros módulos en una aplicación Node.js?

- a. Obtener el objeto/función siempre que vaya a utilizar el módulo.
- **b. Obtener el objeto/función una vez y enviarlo como parámetro a otros módulos.**
- c. Obtener el objeto/función y almacenarlo en variables de la app.

En MongoDB, una colección:

- a. Define la estructura de los documentos
- b. Cada documento se almacena como una colección
- **c. Almacena documentos que pueden tener estructuras diferentes.**

Es una de las principales ventajas de MongoDB

- a. Ideal para almacenar datos estructurados
- b. Muy potente en el manejo de transacciones
- **c. No necesita definir un esquema previo**
- d. Soporte flexible para la definición de relaciones entre entidades

28-03-22 Tema 9

¿Cuál es el objetivo principal de los servicios web?

- **a. Permitir la interoperabilidad entre aplicaciones**
- b. Permitir un acoplamiento fuerte entre aplicaciones
- c. Permitir la comunicación solo entre aplicaciones desarrollados en el mismo lenguaje
- d. Permitir desplegar aplicaciones en nube

¿Por qué los servicios web son adecuados para entornos Web?

- a. Emplean tecnologías estándares como CORBA, RMI, DCOM, etc
- **b. Emplean tecnologías estándares cono HTTP, SOAP, XML, JSON, etc**
- c. No son escalables y no permite la computación distribuida
- d. No se pueden utilizar en aplicaciones de escritorio

Rest

¿En qué consiste el principio HETEOAS de una API REST?

- **a. Los mensajes deben contener enlaces a otros recursos de la API**
- b. La API tiene una sintaxis universal para identificar los recursos (URI)
- c. Los recursos deben ser declarados como cacheables o no cacheables

¿En qué filosofía se basa REST?

a. Se basa en la llamada a procedimientos/funciones remotas
b. Se basa en una filosofía orientada a Recursos
c. Se basa en que un mismo recurso solo puede representarse en único formato, ejemplo JSON
d. Todas son correctas

Una petición web a un recursos REST

a. Debe retornar un código de respuesta estándar
b. Debe retornar una respuesta en formato estándar
c. Todas las anteriores son correctas
d. Todas las anteriores son incorrectas

¿Qué es JSON?

a. Es un formato ligero de almacenamiento e intercambio de datos
b. Es un formato que solo puede ser usado con JAVASCRIPT
c. Es un formato menos eficiente que XML
d. Es un formato que sólo puede ser usado en aplicaciones web

04-04-22 Tema 10

Es un estándar basado en XML para el intercambio de información entre aplicaciones en entornos descentralizados y distribuidos.

a. SOAP
b. WSDL
c. UDDI
d. HTTP

Los servicios web SOAP permiten el intercambio de información en formato.

a. JSON
b. XML
c. XML y JSON
d. XLS

Se utiliza para describir la funcionalidad que proporciona un servicio web SOAP.

a. ENVELOP
b. WSDL
c. UDDI
d. HTTP

Sección en la que se añade información relativa a la autenticación o a un ID de transacción de un sobre SOAP

a. Body
b. Header
c. Envelope
d. Footer

Sara Maria Ramírez Pérez
U0276188

Seminario repaso

¿Qué es Node.js?

a. Un servidor web multiplataforma
b. Un lenguaje de programación
c. Un Framework de software que permite
d. Una plataforma de software que permite ejecutar JS del lado del servidor

¿Cómo gestiona Node.js las operaciones de I/O?

a. De forma síncrona y sin bloqueo
b. De forma asíncrona y sin bloqueo
c. De forma asíncrona y sólo se bloquean algunas operaciones
d. De forma síncrona, asíncrona y con bloqueo

¿Qué tipo de escalabilidad usa Node.js para las aplicaciones?

a. Horizontal
b. Vertical
c. Circular
d. Perpendicular

Node.js mejora la concurrencia porque su arquitectura se basa:

a. Ejecución multihilo con un bucle de eventos
b. Ejecución multihilo y un modelo I/O asíncrono
c. Un solo hilo de ejecución con un bucle de eventos
d. Un solo hilo de ejecución sin bucles de eventos

Es un framework de desarrollo de aplicaciones web minimalista y flexible para Node.js:

a. Mongo
b. Express
c. Swig
d. Javascript

Son características de una comunicación asíncrona:

a. Las operaciones no son bloqueantes
b. Las operaciones se ejecutan de forma secuencial
c. Las operaciones son bloqueantes
d. Sólo se realizan mediante el sistema de callback (retrollamadas)

En una aplicación Node.js, el fichero package.json contiene:

a. La configuración, las dependencias y los metadatos de la aplicación
b. El nombre del servidor
c. Solo las dependencias de la aplicación
d. Todos los ficheros JS de la aplicación

En una aplicación Node.js la función require se utiliza para:

a. Crear una nueva aplicación Node.js
b. Definir módulos

c. Eliminar módulos de la aplicación
d. Importar módulos

¿Cómo se denominan las funciones que se ejecutan cuando se recibe la petición HTTP en una app Node.js?

a. App (Aplicaciones)
b. Handler (Manejadores)
c. Routing (Enrutamiento)
d. Module (Módulos)

Para obtener los valores de los parámetros enviados en una petición POST en Node.js se utiliza:

a. req.body.<clave_parámetro>
b. req.params.<clave_parámetro>
c. req.query.<clave_parámetro>
d. req.form.query.<clave_parámetro>

Es una forma de declarar un módulo en Node.js:

a. module.exports=function(){…}
b. module.imports=function(){…}
c. module.define={…}
d. module.require=function(){…}

Forma de incluir un módulo en Node.js:

a. module.require=function(){…}
b. require("./routes/users.js")[app];
c. var express = require('express');
d. module.import={}

Una plantilla Twig recibe un atributo con clave "nombre", ¿cómo se insertaría el valor de ese atributo en un elemento H1?

a. <h1>{nombre}</h1>
b. <h1>{{nombre}}</h1>
c. <h1>${nombre}</h1>
d. <h1>#{nombre}</h1>

Para obtener los valores de los parámetros enviados en una petición GET en Node.js se utiliza:

a. req.body.<clave_parámetro>
b. req.params["clave_parámetro"]
c. req.query.<clave_parámetro>
d. req.form.query.<clave_parámetro>

¿Cuál es el objetivo principal de los servicios web?

a. Permitir la interoperabilidad entre aplicaciones.
b. Permitir un acoplamiento fuerte entre aplicaciones
c. Permitir la comunicación sólo entre aplicaciones desarrolladas en el mismo lenguaje
d. Permitir la escalabilidad entre aplicaciones

¿Por qué los servicios web son adecuados para entornos web?

a. Emplean tecnologías estándares como CORBA, RMI, DCOM, etc.
b. Emplean tecnologías estándares como HTTP, SOAP, XML, etc.
c. No son escalables y no permite la computación distribuida.
d. Facilita el desarrollo de aplicación web

¿En qué filosofía se basa REST?

a. Se basa en la llamada a procedimientos/funciones remotas
b. Se basa en que un mismo recurso solo puede representarse en un único formato, ejemplo JSON
c. Se basa en una filosofía orientada a recursos
d. Se basa en una filosofía orientada a mensajes

En un mecanismo que le permite a una aplicación realizar transferencias seguras de datos en dominios cruzados entre navegadores y servidores web:

a. JSON
b. AJAX
c. CORS
d. Access-Control

Una petición web a un recurso REST

a. Debe retornar un código de respuesta estándar
b. Debe retornar una respuesta en formato estándar
c. Las respuestas a y b son correctas
d. Las respuestas a y b con incorrectas

Es un estándar basado en XML para el intercambio de información entre aplicaciones en entornos descentralizados y distribuidos:

a. SOAP
b. WSFL
c. UDDI
d. REST

Se utiliza para describir la funcionalidad que proporciona un servicio web SOAP:

a. Envelop
b. WSDL
c. UDDI
d. XML

En MongoDB, una colección:

a. Define la estructura de los documentos
b. Cada documento se almacena como una colección
c. Almacena documentos que pueden tener estructuras diferentes
d. Un documento almacena varias colecciones

Lenguaje de dominio específico de JHipster que sirve para describir las entidades y sus relaciones de una aplicación web:

a. DSL
b. JDL
c. JDSL
d. JSL