|  |  |
| --- | --- |
| **TÍTULO:** node: origenes en filosofia  **ASIGNATURA:**  fundamentos de node.js  **FECHA:** 01/06/2021  **Modulo Nº:** 1,V2 | |
| **IDEAS CLAVE**  (Ideas principales, palabras clave, preguntas, etc.) | **APUNTES O NOTAS DE CLASE**   * Entorno de ejecucion para servidores que funcionen con node, creado en el 2009, * posee sus diferentes herramienta para faciltar ciertas tareas * escrito en c++, convierte el codigo a maquina en diferencia al navegador donde este lo interpreta en tiempo real. Todo esto gracias al motor V8 * Modulos: todol lo diferente a sintaxis seran modulos, estos modulos vienen por defecto y los demas se descargan a demas de que se pueden crear unos propios * orientar tu codigo a manera reactiva, similar a rxjs o node es similar a rxjs * bucle de eventos que va a estar ejecutandose, es es el que va tener los eventos escuchando a nuestros metodos |
| **RESUMEN**  Ideas nuevas que trae la utilizacion de JavaScript para el lado del servidor, la pricipal diferencia entre la ejecucion entre un entorno y es que en servidor sera compilado, en web es interpretado en tiempo real. | |

|  |  |
| --- | --- |
| **TÍTULO:** EventLoop: asincrona por diseño  **ASIGNATURA:**  fundamentos de node.js  **FECHA:** 01/06/2021  **Modulo Nº:** d | |
| **IDEAS CLAVE**  (Ideas principales, palabras clave, preguntas, etc.)  NODE JS PERMITE MANEJAR DIFERENTES FUNCIONES DE MANERA PARALELA | **APUNTES O NOTAS DE CLASE**   * Un bucle en ejecucuion constante, este eventloop gestiona y escucha de manera constantes lo eventos, esto forma una aplicación asincrona * el event queue, es la herramienta que alimenta al event loop, este almacena todos los eventos en una cola y le envia cada evento uno por uno para que haya alguna respuesta o solucion si no la hay los enviar al thread pool. * El **THREAD POOL** maneja la fuciones asincronas, este espera para allar su solucion, en este proceso se creara un hilo para manejar cada proceso asincrono y no afeactar el hilo principal de nuestra aplicación , una vez hayada la solucion de un proceso este lo enviera al event loop y este decide si terminar dicho evento o enviarlo a cola |
| **RESUMEN**  Toda la ejecucion del codigo en node sera ejecutado de manera asincrona, esto no bloquea ningun proceso en nuestra app.  Callbaks, promise, awai y asicn , son algunas herramientas utlizadas para estos procesos  esta manera de ejecucion de codigo puede traer problemas para la detencion de errores y seguridad | |

|  |  |
| --- | --- |
| **TÍTULO:** Monohilo: implicaciones en diseño y seguridad  **ASIGNATURA:**  fundamentos de node.js  **FECHA:** 01/06/2021  **Modulo Nº:** 1 | |
| **IDEAS CLAVE**  (Ideas principales, palabras clave, preguntas, etc.)  Recordar que la funcion setInterval se ejecuta de una manera infinita, pero su ejecucion va ser cada x tiempo  el valor de x va ser dado por nosotros | **APUNTES O NOTAS DE CLASE**  El problema de la ejecucion de un solo hilo en nuestra aplicación   * el problema de ser monohilo sera que si nuestro codigo detecta un error toda la ejecucion de nuestro codigo se va a detener. * Es clave estar pendiente de cualquier problema que pueda cotarlaejecucion de nuestro programa. * Comprobar que todo lo que pueda fallar no falle, * todo lo que pueda ser asincrono que lo sea |
| **RESUMEN**  Es importante estar pendiente de nuestras funciones, para concer que parametros entran en ellas,  y con eso evitar cualquier error,que pueda pararla ejecucion de nuestro programa.  Al parecer solo hay un hilo de ejecucion en node, los multi hilos seran par los procesos asincronos | |

|  |  |
| --- | --- |
| **TÍTULO:** Variables de entorno  **ASIGNATURA:**  fundamentos de node.js  **FECHA:** 01/06/2021  **Modulo Nº:** d | |
| **IDEAS CLAVE**  (Ideas principales, palabras clave, preguntas, etc.)  El software no deben guardad valores como credenciales o token de seguirdad | **APUNTES O NOTAS DE CLASE**  Las variables de entorno son vari bles que no van dentro del software,estas son definidas dentro de nuestra maquina y son llamadas por nuestro software  por buena practica es bueno nombrar anuestrs variable de entorno en mayusculas y separar nuestra variables por giones al piso |
| **RESUMEN**  Recordemos que una variable de entorno es dato que esta en nuestro sistema y no dentro de el software | |

|  |  |
| --- | --- |
| **TÍTULO:** Herramientas para ser más felices: Nodemon y PM2  **ASIGNATURA:**  fundamentos de node.js  **FECHA:** 01/06/2021  **Modulo Nº:** 1 | |
| **IDEAS CLAVE**  (Ideas principales, palabras clave, preguntas, etc.) | **APUNTES O NOTAS DE CLASE**  Nodemon: recarga en caliente el codigo escrito en node js, este modulo es una dependecia de desarrollo  Pm² es una herramienta que nos ayuda a tener el codigo total mente monitorizado en producion |
| **RESUMEN**  Estas herramientas son para facilitar el desarrollo de node y la administracion de nuestros servidores hechos en node. | |

|  |  |
| --- | --- |
| **TÍTULO:** Callbaks  **ASIGNATURA:**  fundamentos de node.js  **FECHA:** 02/06/2021  **Modulo Nº:** 2 | |
| **IDEAS CLAVE**  (Ideas principales, palabras clave, preguntas, etc.)  Las funciones en javascript son elementos de primer nivel | **APUNTES O NOTAS DE CLASE**  SetTimeOut: el lo mas parecido a una funcicon asincrona, ya decidimos en cuanto tiempo se debe ejecutar.  Un callback es la anidacion de una funcion dentro de otra, esta es recibida como parametro. es ejecutada cuando se realiza el llamado dentro de la funcion  un problema que llega con la asincronia es el no saber cuanto tarda un proceso en completarse para ejecutar el siguiente, |
| **RESUMEN** | |

|  |  |
| --- | --- |
| **TÍTULO:** Callbalck hell: refactorizar o sufrir  **ASIGNATURA:**  fundamentos de node.js  **FECHA:** 02/06/2021  **Modulo Nº:** 2 | |
| **IDEAS CLAVE**  (Ideas principales, palabras clave, preguntas, etc.)  La recursividad consiste en en llamar la funcion dentro de si misma ayudados por control de flujo con los parametros ligeramente cambiados | **APUNTES O NOTAS DE CLASE**  ¿Que es el callback hell?  Es un antipatron que se origina al momento de intentar de trabajar con la logica asincrona en el ambito de nodejs, este consiste en la anidacion de varios callbacks lo que genera dificultad al momento de leerlo debbugear dicho codigo.  Existen diferentes tecnicas y herramientas que pueden evitar la creacion de callbacks hell, como: la modularizacion, las promise, y los async await.  Otra buena solucion es la recursividad en las funcione, la funciones recursivas son aquellas que se llaman dentro de simismas con controles de flujo como el if |
| **RESUMEN**  Lo unico malo es ver la creacion de un callback hell y no evitar su creacion  da click [Aquí](https://carbon.now.sh/HRcV3hQCZyJqDEEpuM93) para ingresar al recurso  metodos en los que le podemos hayar soluciones al callback hell   * promoesas, async-await, y recursividad al momento de escribir codigo. | |

|  |  |
| --- | --- |
| **TÍTULO:** Promesas  **ASIGNATURA:**  fundamentos de node.js  **FECHA:** 02/06/2021  **Modulo Nº:** 2 | |
| **IDEAS CLAVE**  (Ideas principales, palabras clave, preguntas, etc.)  Las promesas se crean para dejar de tranabjar con sincronia el infierno de los callbacks | **APUNTES O NOTAS DE CLASE**  ¿que son las promesas? Una promesa como lo dice su nombre es algo que va pasar en un futuro y nosotros vamos a estar al pendiente de su respuesta , es una funcion que puede tener una respuesta negativa o positiva.  Las se contruyen apatir de la palabra recervada **promise,** esta recibe una funcion que lleva dos parametros estos son resolve y reject:   * resolve = resuelve la promesa, este paremtro se ejecuta en la funcion then * reject = rechaza la promesa y devuelve error, este parametro se ejecuta en la funcion con el paremetro catch   una promesa tiene tres estados, pendiente, resolve y reject  Importante: aunque se pueden utilizar promesas sin el catch, es importantetenerlo ya que es el el que captura todos los erroes en nuestros procesos.  Lecturas importantes:   * [Explicacion de promesas en platzi](https://platzi.com/blog/que-es-y-como-funcionan-las-promesas-en-javascript/) |
| **RESUMEN**  La promesa es una herramienta que nos permite manejar el asincronismo de una manera mucho mas facil ya que fue la solucion al callback hell  estes promesas tienen tres estados, resolve, reject y pendiente  recordemos que como su nombre lo dice es algo que promete su repuesta en el futuro, que esta puede ser positiva o negativa | |

|  |  |
| --- | --- |
| **TÍTULO:** Async/await  **ASIGNATURA:**  fundamentos de node.js  **FECHA:** 02/06/2021  **Modulo Nº:** 2 | |
| **IDEAS CLAVE**  (Ideas principales, palabras clave, preguntas, etc.) | **APUNTES O NOTAS DE CLASE**  ¿Qué es async/await?  Las funciones async con su palabra reservada await es la forma mas facil de escrivbier codigo asicrono, ya que su extructura es muy similar a las funciones sincronas de antallo, el resultado de estas funciones asincronas devuelven una promesa lo que nos permitira estar pendiente a su resultado  ¿Como funcionan estas palabras clave dentro del asincronismo?  Estas palabras clave son agredas antes de la definicon de una funcion(Async) y dentro de la funcion para una determinada tarea (Await)  ¿Cuando utilizar?  Estas funciones deben ser utlizadas cuando deban ejecutar procesos que tomen mucho tiempo de ejecucion o lectura o escritura, ademas si necesitan la informacion de manera inmedianta |
| **RESUMEN**  Con async await podemos tratar funciones asicronas a nivel visual como si fueran funciones sicronas  en async await espera que el proceso anterior termine para seguir con el siguiente | |

|  |  |
| --- | --- |
| **TÍTULO:** P  **ASIGNATURA:**  fundamentos de node.js  **FECHA:** 03/06/2021  **Modulo Nº:** 3 | |
| **IDEAS CLAVE**  (Ideas principales, palabras clave, preguntas, etc.) | **APUNTES O NOTAS DE CLASE** |
| **RESUMEN** | |

|  |  |
| --- | --- |
| **TÍTULO:** P  **ASIGNATURA:**  fundamentos de node.js  **FECHA:** 03/06/2021  **Modulo Nº:** 3 | |
| **IDEAS CLAVE**  (Ideas principales, palabras clave, preguntas, etc.) | **APUNTES O NOTAS DE CLASE** |
| **RESUMEN** | |

|  |  |
| --- | --- |
| **TÍTULO:** P  **ASIGNATURA:**  fundamentos de node.js  **FECHA:** 03/06/2021  **Modulo Nº:** 3 | |
| **IDEAS CLAVE**  (Ideas principales, palabras clave, preguntas, etc.) | **APUNTES O NOTAS DE CLASE** |
| **RESUMEN** | |

|  |  |
| --- | --- |
| **TÍTULO:** P  **ASIGNATURA:**  fundamentos de node.js  **FECHA:** 03/06/2021  **Modulo Nº:** 3 | |
| **IDEAS CLAVE**  (Ideas principales, palabras clave, preguntas, etc.) | **APUNTES O NOTAS DE CLASE** |
| **RESUMEN** | |

|  |  |
| --- | --- |
| **TÍTULO:** P  **ASIGNATURA:**  fundamentos de node.js  **FECHA:** 03/06/2021  **Modulo Nº:** 3 | |
| **IDEAS CLAVE**  (Ideas principales, palabras clave, preguntas, etc.) | **APUNTES O NOTAS DE CLASE** |
| **RESUMEN** | |