

Temas vistos

Clase # 21 - Base de datos - 20/5

Temas

Práctica

Material Didáctico

Clase # 22 - Base de datos parte 2 | 22/5

Temas

Práctica

Material didáctico

Clase # 23 - Base de datos parte 3 | 27/5

Temas

Material didáctico

Clase # 24 - Introducción a NodeJS 1 - 29/5

Temas

Práctica

Material didáctico

Clase # 25 - NodeJS 2 - Módulos | 3/6

Temas

Material didáctico

Clase # 26 - NodeJS 3 - Rutas | 5/6

Temas

Práctica

Material didáctico

Clase # 27 - NodeJS 4 - Rutas parte 2 | 10/6

Temas

Material didáctico

Clase # 28 - NodeJS 5 | 12/6

Temas

Práctica

Material didáctico

Clase # 29 - NodeJS 6 | 19/6

Temas

Material didáctico

Clase # 30 - NodeJS 7 | 24/6

Temas

Material didáctico

Clase # 31 - NodeJS 8 | 26/6

Temas

Material didáctico

Clase # 32 - Presentación de trabajos | 1/7

Temas

Clase # 33 - Presentación de trabajos | 3/7

Temas

Clase # 21 - Base de datos - 20/5

Temas


- Qué es una base de datos
- Tipos de base de datos (relacional y no relacional)
- roadmaps
- Instalación de Xampp
- Lógica de una base de datos (ID, filas y columnas, datos, etc)
- Interfaz de base de datos
- Mi primer base de datos en **PHPMyAdmin** ([Deploy Fully Managed MySQL in the Cloud](#))
 - creación de base de datos prueba con tabla (id, nombre, email)
 - incorporar datos
 - tipos de campos

Práctica

Crear una base de datos en [Deploy Fully Managed MySQL in the Cloud](#) , con la siguiente tabla, agregar tipo de datos e ingresar al menos 3 productos (por ejemplo, notebooks)

- ID
- nombre
- urlImagen
- descripcion
- precio
- stock

Material Didáctico

- [Bases de datos relacionales vs. no relacionales: ¿qué es mejor? - Aukera](#)
- [Tipos de datos en MySQL para una base de datos SQL](#)
- [MySQL :: MySQL 8.0 Reference Manual :: 11 Data Types](#)
-  40 - Indices de Tablas (Primary Key, Unique, Plain Text, FullText) [MariaDB]
- [MariaDB vs MySQL, un Resumen sobre las Tecnologías de Base de Datos](#)

☰ Recursos (Documentación, Cursos, Desafíos, Juegos, ...)

- [Base de datos SQL vs NoSQL: diferencias, ventajas y mitos](#)
- 📁 MySQL
- Hostings de base de datos
 - [Clever Cloud](#)
 - alternativa de host de base de datos: <https://www.alwaysdata.com/en/>

Si existe un error de bloqueo de puertos

- [How to solve "Error: Apache shutdown unexpectedly"? - Stack Overflow](#)

Clase > <https://youtu.be/-U2d9K8BEpA>

Clase # 22 - Base de datos parte 2 | 22/5

Temas

- Sentencias MySQL usando la terminal
- Creación de relación entre 2 tablas mediante terminal

Práctica

- Hacer ejercicios de [SQLBolt - Learn SQL - SQL Lesson 1: SELECT queries 101](#)

Material didáctico

- 📁 Comandos SQL
- [Extensión de VSC](#)
- [Consola de mysql online](#)
- [El lenguaje SQL | Dataprix](#)
- 📺 CREANDO la BASE DE DATOS de Twitter con MySQL
- [SQL Basics Cheat Sheet | DataCamp](#)

☰ Recursos (Documentación, Cursos, Desafíos, Juegos, ...)

- [Learn SQL: Queries Cheatsheet | Codecademy](#)





Clase > <https://youtu.be/Zyj-kTB-QOc>

Clase # 23 - Base de datos parte 3 | 27/5

Temas

- Relaciones (1 a 1, 1 a muchos, muchos y muchos)
 - Usando phpMyAdmin
 - Consultas a tablas relacionadas
 - Uso de JOINS, subconsultas, funciones escalares

Material didáctico

- **Relaciones**
 - [Planificación relaciones](#)
 - [Relaciones de tablas](#)
 - [Relaciones entre tablas de bases de datos – Styde.net](#)
 - [Uno a muchos | Bases de datos en la enseñanza](#)
 - [SQL para principiantes: Parte 3 - Relaciones de base de datos](#)
 - [mysql - ¿Cómo establecer relación 1:n en base de datos SQL?](#)
 -   Crear una relación de uno a muchos en MySQL con PhpMyAdmin (CUR...
 -  Como Relacionar Tablas en Mysql con phpMyAdmin
 - [Cómo unir 3 tablas \(o más\) en SQL | LearnSQL.es](#)
- **Avanzado**
 - [Sentencia JOIN en SQL - Wikipedia, la enciclopedia libre](#)
 - [¿Qué significa CASCADE, SET NULL, RESTRICT, NO ACTION al crear relaciones?](#)
 - [Oracle Clave primaria vs Clave única vs Índice único - Stack Overflow en español](#)
- [MariaDB JOIN - AlphaCodingSkills](#)
-  Ejemplos según el PPT de la clase 3 de SQL 2024

clase > <https://youtu.be/zLBoMOipmno>

Clase # 24 - Introducción a NodeJS 1 - 29/5


Temas

- **NodeJS**
 - Instalación
 - Qué es
 - ventajas
 - Hola mundo (ejemplo)
 - Instalación Express usando NPM
 - sitio estático usando Express
 - usando carpeta public para servir sitio web
 - correr fetch del lado del cliente vs lado servidor


Práctica

- Recrear el servidor web que se mostró en clase, usando Express.

Material didáctico




- [Introduction to Node.js - GeeksforGeeks](#)
 -  NodeJS
 - [¿Por qué demonios usaría Node.js? Un tutorial caso por caso | Toptal](#)

Repaso Asincronismo

- [Async in JS: How does it work | Stack Diary](#)
- [JavaScript Internals: Under The Hood of a Browser | by Vlad Ostrenko | Better Programming](#)
-  Como funciona JavaScript, event loop, call stack, task queue [Descarga PDF...]
 - [Synchronous vs Asynchronous JavaScript – Call Stack, Promises, and More](#)

☰ Recursos (Documentación, Cursos, Desafíos, Juegos, ...)


NodeJs - Su funcionamiento técnico

- [Código bloqueante y no bloqueante con NodeJS | Arsys](#)
 -  [How Node.js Works | Mosh](#)
 -  [Non-blocking I/O and how Node uses it, in friendly terms: blocking vs async ...](#)
 - [Node.js Single Thread Mechanism](#)
 - [How Node.js overcome the problem of blocking of I/O operations ? - GeeksforGeeks](#)
 - [If Javascript Is Single Threaded, How Is It Asynchronous? - DEV Community](#)


Express

- [express - npm](#)
- [Ejemplo "Hello World" de Express](#)
- [Servicio de archivos estáticos en Express](#)

NPM

-  [#4 - NODE JS - ¿Qué es el package.json? - Introducción práctica PASO A PA...](#)
- Licencias: [Licencia de software libre](#)
- [¿Que diferencia hay entre NPM y YARN?](#)
- [¿Tiene sentido Yarn ahora que tenemos npm 5? | campusMVP.es](#)

Framework para hacer apps de escritorio

- [Electron | Build cross-platform desktop apps with JavaScript, HTML, and CSS.](#)

Clase > <https://youtu.be/HbOJF-8ljhs>

Clase # 25 - NodeJS 2 - Módulos | 3/6

Temas

- Métodos de peticiones y códigos de respuesta
- Módulos
 - Nativos (ej: fileSystem)
 - propios

☰ Recursos (Documentación, Cursos, Desafíos, Juegos, ...)

- Externos (ej: [colors](#))
- Desestructuración de objetos
- reinstalar los paquetes al no tener node_modules
- uso de watch para no reiniciar el servidor

Material didáctico

HTTP y métodos

- [Cabeceras HTTP para Dummies | Envato Tuts+](#)
- [Métodos de petición HTTP](#)
- [Códigos de estado de respuesta HTTP](#)
- [HTTP Cats](#)
- [Generalidades del protocolo HTTP](#)

Módulos

- [CommonJS vs ES Modules - Javascript en español - Lenguaje JS](#)
 - [CommonJS vs. ES Modules: Modules and Imports in NodeJS](#)
 - [REQUIRE vs IMPORT ⚡ COMMON JS vs ES MODULES 🤔 CJS vs ES...](#)
 - [The requested module does not provide an export named in JS | bobbyhadz](#) (errores comunes)
- [FileSystem](#)
 - otros ejemplos de FS - [Node.js Examples](#)
- [Cómo utilizar la Desestructuración de Arreglos y Objetos en JavaScript](#)
- [Módulos en Node.js](#)

Clase > <https://youtu.be/J5b1vNgiqdE>

Temas

- rutas ([Direccionamiento básico de Express](#)) (get, put, post, delete)
 - parámetros
 - params (nos permite rutas dinámicas, ej: usuarios/:id)
 - query (URL con variables, ej "?mensaje=hola&costo=1000")
 - body (Para tomar los valores de un formulario)
- Middlewares (su aplicación)
 - Ejemplo de `express.urlencoded()` y `express.json()`

Práctica

- Recrear el crud realizado en clase

Material didáctico

Rutas

- [Direccionamiento básico de Express](#)
- [Rutas / Routing con NodeJS y Express - Tutorial Stack MEAN](#)
- Manejo de rutas modulando [Direccionamiento de Express](#)
- [¿Cuáles son las ventajas e inconvenientes de usar PATCH para actualizar recursos?](#) (PATCH vs PUT).

Parámetros

- [¿Para qué sirve `app.use\(express.json\(\)\)` en Express?](#) (menciona el uso de `req.body`)
- [What are `express.json\(\)` and `express.urlencoded\(\)`? - Stack Overflow](#)
- [Método de respuesta JSON en Express.js.](#)
- [¿Qué podemos hacer con el `express.urlencoded\(\)`?](#)
- [Parámetros de ruta en express](#)
 - [Parámetro Query en Express](#)
 - [Parámetro Params en Express](#)
 - [req.body en Express](#)

Solicitud y respuesta

- [¿Qué son los parámetros "res" \(respuesta\) y "req" \(solicitud\) en las funciones Express?](#)
- [Códifica Su Primera API con Node.js y Express: Comprensión REST API](#)

Clase > <https://youtu.be/BoRufBSc8Ck>

Clase # 27 - NodeJS 4 - Rutas parte 2 | 10/6

Temas

- CRUD con una DB SQL
- DB
 - Configuración base de datos
 - Pool de conexiones
- Paquetes
 - [mysql2 - npm](#)
 - [express - npm](#)

Material didáctico

Base de datos

Paquete mysql2

- [Pool de conexiones - Javascript en el servidor: Node.js](#)
- [MySQL Wildcard Characters](#)
- [How to filter query results in MySQL](#)

APIS

- [¿Qué es una api rest? ¿Cómo funciona? ¿En que tipo de web utilizarlas?](#)
- [Ten una REST API Falsa Funcionando Usando json-server | Envato Tuts+](#)
 - Fakes API

☰ Recursos (Documentación, Cursos, Desafíos, Juegos, ...)

- [dummyJSON - API con información random para testear](#)
- [JSONplaceholder - API con información random para testear](#)

Clase > https://youtu.be/luQK_1qjVI0

Clase # 28 - NodeJS 5 | 12/6

Temas

- Conexión con front para desplegar los productos y tener un buscador
- Creación de filtros

Práctica

Recrear lo visto en clase para luego poder aplicarlo en el proyecto

Material didáctico

Base de datos

Paquete mysql2

- [How to filter query results in MySQL](#)

Middlewares

- [Qué es un Middleware en Express](#)
- [Middleware en Express JS](#)
- [Express's json\(\) and urlencoded\(\) Explained - DEV Community](#)
 - [express.json\(\)](#)
 - [express.urlencoded](#)
 - [What does 'extended' mean in express 4.0? - node.js](#)
- [Getting the Request Body in Express - Mastering JS](#)

URL

- [FormData - Referencia de la API Web | MDN](#)
- [URLSearchParams - Referencia de la API Web | MDN](#)

Clase > https://youtu.be/dW4_CXLSL6U

Clase # 29 - NodeJS 6 | 19/6

Temas

- Paquetes
 - multer básico (para subir imágenes al servidor)
- Cómo registrar un producto con una imagen
- Tablas relacionadas (productos y categorías, o por ej cuentas y roles)
- Arquitectura
 - controladores
 - router
 - helper (para manejar los filtros)

Material didáctico

Multer

- [Subir archivos con Multer en Node.js y Express](#) (hasta la parte de mongo)
- [File Upload Using Multer in Node.js and Express.js](#)
- [multer - npm](#)


Clase > <https://youtu.be/XWhnB7smyk8>

Temas


- JWT (JSON Web token)
 - teoría
 - práctica
 - Se puede usar localStorage, pero para mayor seguridad, usamos cookies con [httponly](#)

Material didáctico


JWT

-  [JWT: JSON Web Token | ¿Son tan buenos como parecen?](#) 🤔
- [¿Qué es JWT y cómo se diferencia de Cookies y Sesiones?](#)
- [JWT tokens and security - working principles and use cases](#)
 - [XSS \(Cross-Site Scripting\) vulnerabilities: principles, types of attacks, exploitations and security best practices](#)
- [Best Practices for Using JWT](#)
- Bearer
 - [Autenticación con Bearer Token – Documentación API New Bytes](#)
 - [Autenticación y autorización basada en token \(JWT Bearer\) con ASP.NET Core | Rafael Neto](#)
 - [Bearer Authentication](#)

JWT Avanzado

-  [10 - JWT \(Json web token\) - Node Js ¡desde cero! \(Link con token\)](#)
- [Email Verification - GeeksforGeeks](#)
- [How to Build A Passwordless Authentication with Email and JWT | HackerNoon](#)
(verificación de email)

JWT vs Cookies

-  [Autenticación con JWT y Cookies](#) 🍪🔑
- [Difference between Session Cookies vs. JWT \(JSON Web Tokens\), for session management.](#)
- [Cookies or JWT..? 🤔 - DEV Community](#)

Clase > <https://youtu.be/z5uiynUMgLc>

Clase # 31 - NodeJS 8 | 26/6

Temas

Deploy

- Subir a github
- Subir DB a clever cloud
- Deploy en render

Seguridad

- **express-session** para mantener la sesión de login abierta
- **bcrypt** para almacenar contraseñas seguras.

Material didáctico

Deploy

Durante el deploy, en por ej, render o etc, si aparece el mensaje de advertencia:
Warning: connect.session() MemoryStore is not designed for a production environment, as it will leak memory, and will not scale past a single process.

- [La explicación de ese mensaje](#)
- **Express-session**
 - Propiedades necesarias

☰ Recursos (Documentación, Cursos, Desafíos, Juegos, ...)

- saveUninitialized [When to use saveUninitialized and resave in express-session - Stack Overflow](#) | [\(en español\)](#)
- resave: [user-content-resave](#)
- secret: [Secure Session](#)
- [Introducción a Cookies y Sesiones con Node.js](#)
- [How to Manage Session using Node.js and Express | CodeForGeek](#)
- [Cookie and Session \(I\): The basic concept | by Alysa Chan | Medium](#)
- [Cookie and Session \(II\): How session works in express-session | by Alysa Chan | Medium](#)
- [Which is Better Node.js Session Management Libraries? cookie-parser vs express-session vs cookie-session](#)
- **Bcrypt**
 - [Encriptar contraseñas con Node - Parzibyte's blog](#)
 - <https://www.itsecurityguru.org/wp-content/uploads/2023/11/Time-to-Crack-bcrypt-hashed-passwords.jpg>
 - [User Authentication and Authorization: Password Authentication Cheatsheet | Codecademy](#)
 - [How to securely store passwords? - DEV Community](#)
 - [bcrypt - npm](#)

Clase > <https://youtu.be/hhO7Nglg7QY>

Clase # 32 - Presentación de trabajos | 1/7

Temas

- Presentación del trabajo de los equipos

Clase > <https://youtu.be/FvcrXWlxkf4>

Temas

- Presentación del trabajo de los equipos

Clase > <https://youtu.be/43g9gkjMEYw>