



**Clase 20.** Programación Backend

# ***DBaaS & Firebase***



## **OBJETIVOS DE LA CLASE**

- Conocer el concepto de Base de Datos como un Servicio (DBaaS).
- Crear una cuenta en Mongo Atlas.
- Acceder a Mongo Atlas a través de clientes CLI, GUI y Web.
- Realizar un CRUD utilizando Firebase.

# ***CRONOGRAMA DEL CURSO***

Clase 19



Mongoose

---

---

---

Clase 20



DBaaS & Firebase

Clase 21



Clase 21 - Trabajo con  
datos: Mocks

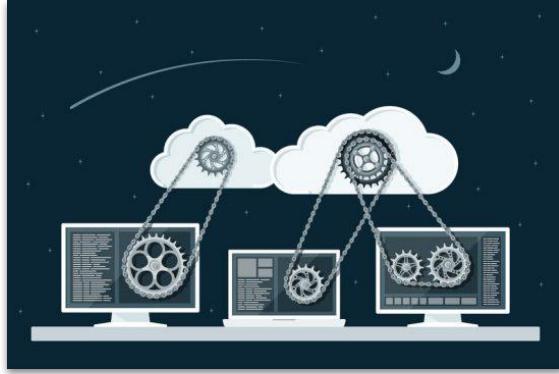
---

---

---

# **DBaaS: DataBase As A Service**





# **DBaaS: ¿en qué consiste?**

- Cuando hablamos de Base de Datos como Servicio (**DBaaS**) nos referimos a la **ejecución** y **gestión** de las **bases de datos, optimizadas** y **alojadas** en la infraestructura de un **proveedor** de **servicios cloud**.
- De esta manera, para gestionar las bases de datos en el cloud **debemos contar con un servicio «por detrás»** como PaaS o IaaS, para estar seguros de tener la infraestructura necesaria.



# **Modalidades y Configuraciones**

Podemos definir dos modalidades de servicio bien diferenciadas

- **Modelo clásico:** el cliente hace uso de la infraestructura física del proveedor para alojar sus bases de datos.
- **Alojamiento gestionado:** el cliente se desentiende de cualquier tarea de mantenimiento y gestión avanzada de la base de datos, que asumirá el proveedor.

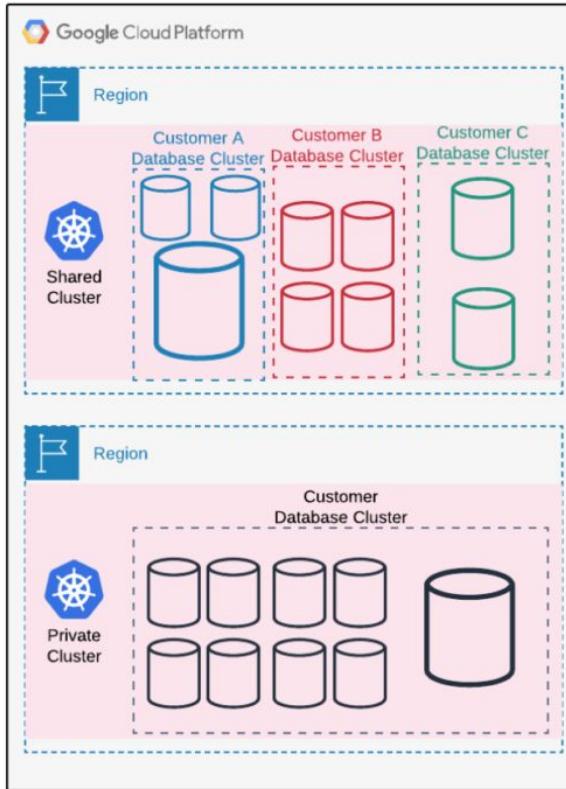
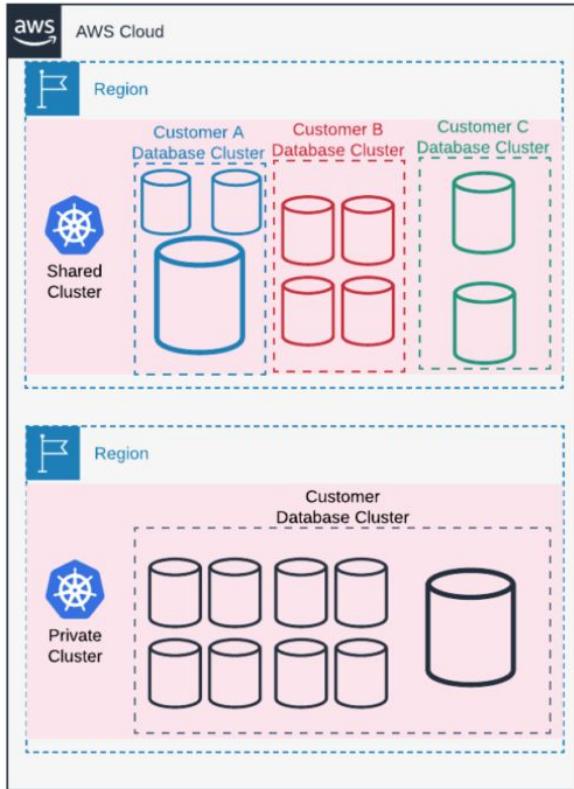


# *Ventajas del modelo DBaaS*



- **Se elimina la infraestructura física** de la ecuación ahorrando en costos, ya que el proveedor es responsable del mantenimiento y la disponibilidad de los sistemas. Los usuarios son responsables de sus propios datos.
- **Ahorro de costos generalizado**. Además de prescindir de las inversiones físicas, con **DBaaS** se puede tener menos personal dedicado a esta tarea, ahorrar en energía y aprovechar mejor el espacio físico.
- **Escalabilidad**. Con **DBaaS** podemos acceder a diferentes tarifas basadas principalmente en el rendimiento deseado y nuestras necesidades.
- **Personal cualificado**. A través de **DBaaS** se accede a expertos en bases de datos que se encargarán de todas las tareas de mantenimiento, actualización, seguridad y gestión.

# DBaaS : Esquemas



**CODER HOUSE**

# *MongoDB Atlas*

# mongoDB Atlas



**CODER HOUSE**

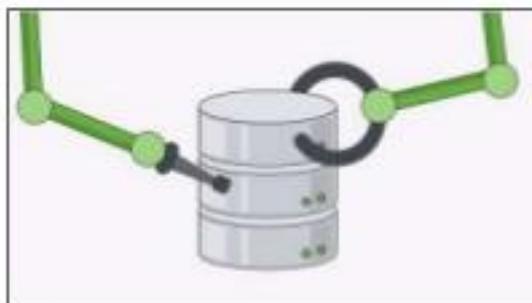
# **Atlas: el salto a la nube de MongoDB**

- **MongoDB Atlas** es un servicio de **Cloud Database** (Base de Datos en la Nube), que nos permite crear y administrar nuestra **MongoDB** desde cualquier lugar del mundo a través de su plataforma.
- MongoDB Atlas está orientado a ser **accesible desde el navegador** y fue desarrollado con el objetivo de aliviar el trabajo de los desarrolladores, al quitarles la necesidad de instalar y administrar entornos de Base de Datos.

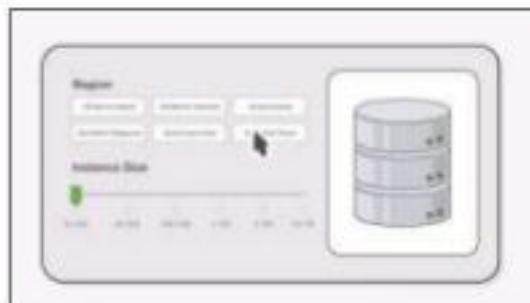
# **MongoDB Atlas: Características Destacadas**

- **Automatización:** una manera fácil de crear, lanzar y escalar aplicaciones en MongoDB.
- **Flexibilidad:** DBaaS con todo lo necesario para las aplicaciones modernas.
- **Seguridad:** varios niveles de seguridad disponibles.
- **Escalabilidad:** gran escalabilidad sin interrumpir la actividad.
- **Alta disponibilidad:** implementaciones con tolerancia a errores y autoreparación predeterminadas.
- **Alto rendimiento:** el necesario para las cargas de trabajo exigentes.

# MongoDB Atlas



Servicio automatizado



DBaaS bajo demanda



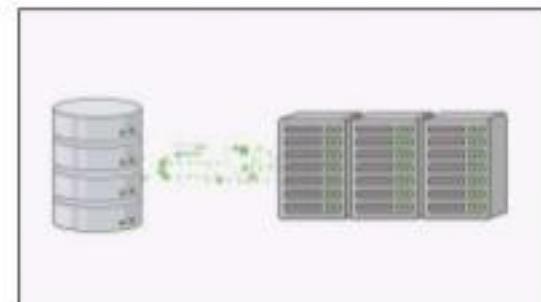
Escalabilidad elástica



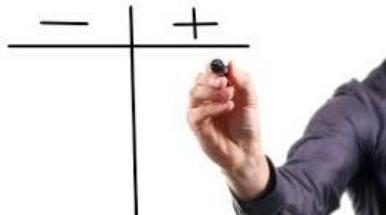
Seguro



Alta disponibilidad



Recuperación de desastres



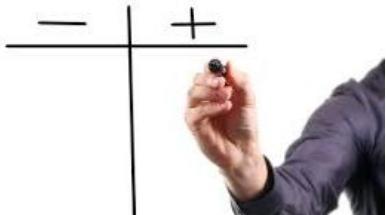
# *Ventajas de MongoDB Atlas*

## Ejecución

- Puesta en marcha de un clúster en segundos.
- Implementaciones replicadas y sin interrupción.
- Total escalabilidad: escalado horizontal o vertical sin interrumpir la actividad.
- Revisiones automáticas y actualizaciones simplificadas.

## Protección y seguridad

- Autenticación y cifrado.
- Copias de seguridad continuas con recuperación temporal.
- Supervisión detallada y alertas personalizadas.



# **Ventajas de MongoDB Atlas**

## **Libertad de movimiento**

- Modelo de planes de precio según demanda: se factura por hora.
- Compatible con diferentes tipos de servicios de nube (AWS, GCP, Azure).
- Parte de un paquete de productos y servicios para todas las fases de la aplicación.

*La configuración se realizará desde una cuenta free. MongoDb Atlas también ofrece una versión de pago.*

# MongoDB Atlas Website: <https://www.mongodb.com/cloud/atlas>

The screenshot shows the MongoDB Atlas website. At the top, there's a navigation bar with links for Managed MongoDB Hosting, Cloud, Software, Pricing, Learn, Solutions, Docs, Contact, Sign In, and a prominent green Try Free button. A banner at the top encourages users to "Nominate a project for the MongoDB Innovation Awards today!" Below the banner, the main heading "MongoDB Atlas" is displayed in large, bold, teal letters. A descriptive paragraph highlights the service as the most innovative cloud database service, mentioning unmatched data distribution and mobility across AWS, Azure, and Google Cloud, built-in automation for resource and workload optimization, and more. A green "Start free" button is located below this text. At the bottom left, there's a link for users who already have an account: "Already have an account? Log in here →". On the right side of the page, a detailed interface for selecting a cloud provider and region is shown. It features sections for "Cloud Provider & Region" (with options for AWS, Google Cloud, and Azure), "Multi-Cloud, Multi-Region & Workload Isolation" (with a toggle switch currently set to OFF), and a grid of regional cluster options. The "N. Virginia (us-east-1)" cluster is highlighted with a green border and labeled as "FREE TIER AVAILABLE". Other regions listed include Stockholm (eu-north-1), Hong Kong (ap-east-1), Ireland (eu-west-1), Tokyo (ap-northeast-1), Seoul (ap-northeast-2), N. California (us-west-1), London (eu-west-2), Paris (eu-west-3), Singapore (ap-southeast-1), and Mumbai (ap-south-1). Each region entry includes a small flag icon and a star rating.

Managed MongoDB Hosting | Docs

Nominate a project for the MongoDB Innovation Awards today!

mongoDB Cloud Software Pricing Learn Solutions Docs

Contact Sign In Try Free

# MongoDB Atlas

The most innovative cloud database service on the market, with unmatched data distribution and mobility across AWS, Azure, and Google Cloud, built-in automation for resource and workload optimization, and so much more.

Start free

Already have an account? [Log in here →](#)

Cloud Provider & Region

AWS, N. Virginia (us-east-1)

aws Google Cloud Azure

Multi-Cloud, Multi-Region & Workload Isolation (M10+ clusters)

Distribute data across clouds or regions for improved availability and local read performance, or introduce replicas for workload isolation. [Learn more](#)

OFF

Create a free tier cluster by selecting a region with [FREE TIER AVAILABLE](#) and choosing the M0 cluster tier below.

★ Recommended region

NORTH AMERICA		EUROPE	ASIA
N. Virginia (us-east-1) ★ <a href="#">FREE TIER AVAILABLE</a>	Stockholm (eu-north-1) ★	Hong Kong (ap-east-1) ★	
Ohio (us-east-2) ★	Ireland (eu-west-1) ★ <a href="#">FREE TIER AVAILABLE</a>	Tokyo (ap-northeast-1) ★	
N. California (us-west-1)	London (eu-west-2) ★	Seoul (ap-northeast-2)	
Oregon (us-west-2) ★	Paris (eu-west-3) ★	Singapore (ap-southeast-1) ★ <a href="#">FREE TIER AVAILABLE</a>	
Montreal (ca-central-1)	Frankfurt (eu-central-1) ★ <a href="#">FREE TIER AVAILABLE</a>	Mumbai (ap-south-1) <a href="#">FREE TIER AVAILABLE</a>	

**CODER HOUSE**

# *¿Cómo crear y configurar una cuenta en MongoDB Atlas?*



# Configuración de cuenta



**mongoDB. Atlas**

[Get started free](#)  
No credit card required

 [Sign up with Google](#)

or

Your Company (optional)

How are you using MongoDB? 

Your Work Email

First Name

Last Name

Password

8 characters minimum

I agree to the [terms of service](#) and [privacy policy](#).

[Get started free](#)

Already have an account? [Sign In](#).

1. Nos dirigimos a la página oficial de [\*\*MongoDB Atlas\*\*](#):
2. Seleccionamos **START FREE** y nos registramos con un correo. También podemos ingresar con Google.
3. Luego nos redireccionará a la próxima ventana donde **continuamos haciendo click en Create cluster**.



# Configuración de cuenta



Daniel's Org - 2020-... Access Manager Support Billing All Clusters Daniel

Project 0

DATA STORAGE

Clusters

Triggers

Data Lake

SECURITY

Database Access

Network Access

Advanced

Feature Requests

Flex Consulting Available

Want help from a MongoDB expert?  
We now offer remote consulting engagements.

Get Flex Consulting Now

Charts

Create a cluster

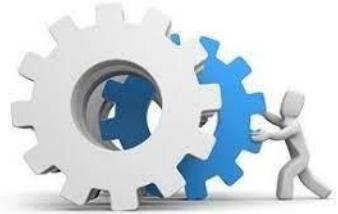
Choose your cloud provider, region, and specs.

Once your cluster is up and running, live migrate an existing MongoDB database into Atlas with our [Live Migration Service](#).

Build a Cluster



**CODER HOUSE**



# Configuración de cuenta



 MONGODB ATLAS

## Choose a path. Adjust anytime.

Available as a fully managed service across 60+ regions on AWS, Azure, and Google Cloud

Dedicated Multi-Cloud & Multi-Region Clusters

For teams developing world-class applications that require multi-region resiliency or ultra-low latency:

- ✓ Includes all features from Shared and Dedicated Clusters
- ✓ Replicate data across clouds and regions
- ✓ Globally distributed read and write operations
- ✓ Control data residency at the document level

[Create a cluster](#)

Starting at \$0.13/hr\*  
\*estimated cost \$88.56/month

Dedicated Clusters

For teams building applications that need advanced development and production-ready environments:

- ✓ Includes all features from Shared Clusters
- ✓ Auto-scaling
- ✓ Network isolation
- ✓ Realtime performance metrics

[Create a cluster](#)

Starting at \$0.08/hr\*  
\*estimated cost \$56.84/month

Shared Clusters

For teams learning MongoDB or developing small applications:

- ✓ Highly available auto-healing cluster
- ✓ End-to-end encryption
- ✓ Role-based access control

[Create a cluster](#)

Starting at FREE

[Advanced Configuration Options](#)

[Dismiss](#)

CLUSTERS > CREATE A STARTER CLUSTER

## Create a Starter Cluster

Welcome to MongoDB Atlas! We've recommended some of our most popular options, but feel free to customize your cluster to your needs. For more information, check our [documentation](#).

Cloud Provider & Region

AWS, N. Virginia (us-east-1) ▾

AWS  Google Cloud  Azure

★ Recommended region

ASIA	NORTH AMERICA	EUROPE
Mumbai (ap-south-1)	N. Virginia (us-east-1) ★	Frankfurt (eu-central-1) ★
Singapore (ap-southeast-1) ★	Oregon (us-west-2) ★	Ireland (eu-west-1) ★

AUSTRALIA

Sydney (ap-southeast-2) ★

Cluster Tier

M0 Sandbox (Shared RAM, 512 MB Storage) [Encrypt](#)

Additional Settings

MongoDB 4.4, No Backup

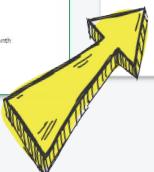
Cluster Name

Cluster0

FREE

Free forever! Your M0 cluster is ideal for experimenting in a limited service. You can upgrade to a production cluster anytime.

[Back](#) [Create Cluster](#)



**CODER HOUSE**



# Configuración de cuenta



The screenshot shows the MongoDB Atlas Cluster creation interface. At the top, there's a banner stating "We are deploying your changes" and "DANIEL'S ORG - 2020-08-04 > PROJECT 0". Below this, the "Clusters" section is visible. A cluster named "Cluster0" is listed under a "SANDBOX" category, showing it's at Version 4.4. To the right of the cluster details, a red box highlights a message: "Your cluster is being created" followed by the note "New clusters take between 1-3 minutes to provision." On the left sidebar, under "DATA STORAGE", the "Clusters" tab is selected. Other tabs include "Triggers", "Data Lake", and "REALM". Under "SECURITY", tabs for "Database Access", "Network Access", and "Advanced" are shown. At the bottom, there are "Feature Requests" and "System Status: All Good".

Nos redirecciónará a un dashboard donde el clúster aún se seguirá creando, pero podemos explorar mientras se crea en segundo plano.

# Configuración de cuenta

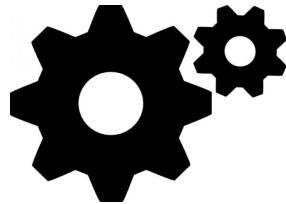


Una vez finalizada la creación realizamos lo siguiente:

The screenshot shows the MongoDB Atlas interface with the 'Clusters' tab selected. A yellow arrow points from the 'CONNECT' button on a cluster card to a larger callout box titled 'Connect to Cluster0'. The callout box contains the following steps:

- ① Add a connection IP address
  - Add Your Current IP Address
  - Add a Different IP Address
  - Allow Access from Anywhere
- ② Create a Database User
  - This first user will have **atlasAdmin** permissions for this project.
  - Keep your credentials handy, you'll need them for the next step.
  - Username: ex. dbUser
  - Password: ex. dbUserPassword
  - Autogenerate Secure Password
  - Show
  - Create Database User

At the bottom of the callout box, there is a 'Choose a connection method' button. The footer of the interface includes links for Privacy, Atlas Blog, Contact Sales, and an Upgrade button. A 'Feature Requests' section is also present at the bottom left.



# Configuración de cuenta



**Punto 1:** MongoDB Atlas nos ofrece una seguridad de conexión por IP, esto quiere decir que podemos configurarlo de 2 maneras:

- **Add Your Current IP Address:** opción para poner nuestra IP, pero cada vez que cambiemos la PC tenemos que volver a configurar.
- **Add a Different IP Address:** para configurar una IP que permita las conexiones de cualquier PC, podemos colocar la IP 0.0.0.0/0.

1 Whitelist your connection IP address

IP Address	Description (Optional)
0.0.0.0/0	An optional comment describing this entry
<input type="button" value="Cancel"/> <input type="button" value="Add IP Address"/>	



# Configuración de cuenta



## Punto 2: Configuración de usuario de acceso.

### 1 Add a connection IP address

✓ An IP address has been added to the IP Access List. Add another address in the [IP Access List tab](#).

### 2 Create a Database User

This first user will have [atlasAdmin](#) permissions for this project.

Keep your credentials handy, you'll need them for the next step.

Username

daniel

Password

.....

Autogenerate Secure Password

SHOW

Create Database User





# Configuración de cuenta

Al realizar todo lo anterior, va quedando así:

Connect to Cluster0

Setup connection security > Choose a connection method > Connect

You need to secure your MongoDB Atlas cluster before you can use it. Set which users and IP addresses can access your cluster now. [Read more](#)

You're ready to connect. Choose how you want to connect in the next step.

① Add a connection IP address  
✓ An IP address has been added to the IP Access List. [Add another address in the IP Access List tab.](#)

② Create a Database User  
✓ A MongoDB user has been added to this project. *Not yours? Create one in the MongoDB Users tab.*  
**You'll need your MongoDB user's credentials in the next step.**

**Choose a connection method**

# ***MongoDB Atlas con distintos clientes***



# Opciones de conexión



Connect to Cluster0

✓ [Setup connection security](#) Choose a connection method [Connect](#)

Choose a connection method [View documentation](#)

Get your pre-formatted connection string by selecting your tool below.

- 1 [Connect with the mongo shell](#)  
Interact with your cluster using MongoDB's interactive Javascript interface
- 2 [Connect your application](#)  
Connect your application to your cluster using MongoDB's native drivers
- 3 [Connect using MongoDB Compass](#)  
Explore, modify, and visualize your data with MongoDB's GUI

[Go Back](#) [Close](#)

Después del paso anterior procedemos a obtener la **cadena de conexión**. Nos aparecerán 3 opciones, de las cuales seleccionaremos una conexión mediante:

- 1. Mongo CLI**
- 2. Node.js**
- 3. Mongo Compass GUI**

# OPTION 1

# Cliente Mongo Shell



Connect to Cluster0

✓ Setup connection security ✓ Choose a connection method Connect

I do not have the mongo shell installed I have the mongo shell installed

1 Select your operating system and download the mongo shell

Windows ▾

Download mongo shell (4.4.4) or Copy download URL

2 Add <your mongo shell's download directory>/bin to your \$PATH variable

3 Run your connection string in your command line

Use this connection string in your application:

```
mongo "mongodb+srv://cluster0.nfdif.mongodb.net/myFirstDatabase" --username daniel
```

Replace myFirstDatabase with the name of the database that connections will use by default. You will be prompted for the password for the Database User, daniel. When entering your password, make sure all special characters are URL encoded.

Si elegimos la **opción 1** obtendremos el string de conexión para vincular la base nube con el cliente CLI

## Mongo Shell:

```
mongo  
"mongodb+srv://cluster0.nfdif.mongodb.net/myFirstDatabase"  
--username daniel
```

# OPTION1

# *Cliente Mongo Shell*



A screenshot of a terminal window titled "Cmder" showing a MongoDB shell session. The session starts with connecting to a MongoDB cluster using the command "mongo 'mongodb+srv://cluster0.nfdif.mongodb.net/myFirstDatabase' --username daniel --password daniel123". It then displays the MongoDB shell version (v4.4.4) and the connection details for the cluster. An implicit session is created with a unique ID. The MongoDB server version (4.4.4) is shown, along with the available databases: "admin" (0.000GB) and "local" (3.990GB). The session ends with the MongoDB Enterprise prompt "MongoDB Enterprise atlas-bwvi2w-shard-0:PRIMARY>".

```
C:\cmder_mini
λ mongo "mongodb+srv://cluster0.nfdif.mongodb.net/myFirstDatabase" --username daniel --password daniel123
MongoDB shell version v4.4.4
connecting to: mongodb://cluster0-shard-00-00.nfdif.mongodb.net:27017,cluster0-shard-00-01.nfdif.mongodb.net:270
17,cluster0-shard-00-02.nfdif.mongodb.net:27017/myFirstDatabase?authSource=admin&compressors=disabled&gssapiServ
iceName=mongodb&replicaSet=atlas-bwvi2w-shard-0&ssl=true
Implicit session: session { "id" : UUID("ae15ccc7-c0dc-4e8a-9ffc-e230097777e6") }
MongoDB server version: 4.4.4
MongoDB Enterprise atlas-bwvi2w-shard-0:PRIMARY> show dbs
admin 0.000GB
local 3.990GB
MongoDB Enterprise atlas-bwvi2w-shard-0:PRIMARY>
```

At the bottom of the terminal window, the taskbar shows the "mongo.exe" application is running. In the bottom right corner, there is a watermark for "CODER HOUSE".

# OPTION 2

# Cliente Node.js



Connect to Cluster0

✓ Setup connection security ✓ Choose a connection method Connect

1 Select your driver and version

DRIVER VERSION

Node.js 3.6 or later

2 Add your connection string into your application code

Include full driver code example

```
mongodb+srv://daniel:<password>@cluster0.nfdif.mongodb.net/myFirstDatabase?  
retryWrites=true&w=majority
```

Replace <password> with the password for the **daniel** user. Replace **myFirstDatabase** with the name of the database that connections will use by default. Ensure any option params are URL encoded.

Having trouble connecting? [View our troubleshooting documentation](#)

Si elegimos la **opción 2** obtendremos el string de conexión para vincular la base nube con **Node.js**:

mongodb+srv://daniel:<password>@cluster0.nfdif.mongodb.net/myFirstDatabase?  
retryWrites=true&w=majority

# OPTION 2

# Cliente NodeJS



A screenshot of the Visual Studio Code interface. The left sidebar shows the file structure with 'index.js' selected. The main editor area contains a Node.js script for connecting to a MongoDB Atlas database. The terminal at the bottom shows the command 'node index.js' being run and the output 'Base de datos conectada'. The status bar at the bottom indicates the code is in JavaScript mode.

```
const mongoose = require('mongoose')

/* ----- */
/* Conexión a la base de datos test en Mongo Atlas */
/* ----- */

//mongoose.connect('mongodb://localhost/mibase',{
mongoose.connect('mongodb+srv://daniel:daniel123@cluster0.nfdif.mongodb.net/test?retryWrites=true&w=majority',{
    useNewUrlParser: true,
    useUnifiedTopology: true
},err => [
    if(err) throw new Error(`Error de conexión a la base de datos ${err}`)
    console.log('Base de datos conectada')
])

PS C:\Cursos\Coderhouse\CursoBackend\Clase21\nodeMongoAtlas> node index.js
Base de datos conectada
```

File Edit Selection View Go Run Terminal Help index.js - nodeMongoAtlas - Visual Studio Code

EXPLORER OPEN EDITORS NODEMONGOATLAS

index.js index.js package-lock.json package.json

TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE

1: node

Ln 10, Col 11 Spaces: 4 UTF-8 CRLF JavaScript Go Live Prettier

CODER HOUSE

# OPTION 3

# Cliente Mongo Compass



Connect to Cluster0

✓ Setup connection security ✓ Choose a connection method Connect

I do not have MongoDB Compass I have MongoDB Compass

1 Select your operating system and download MongoDB Compass

Windows 64-bit (7+) ▾

Download Compass (1.26.0) or Copy download URL

2 Copy the connection string, then open MongoDB Compass.

mongodb+srv://daniel:<password>@cluster0.nfdif.mongodb.net/test

You will be prompted for the **daniel** user's (Database User) username. When entering your password, make sure that any special characters are [URL encoded](#).

Having trouble connecting? [View our troubleshooting documentation](#)

Si elegimos la **opción 3** obtendremos el string de conexión para vincular la base nube con el cliente

## GUI Mongo Compass:

mongodb+srv://daniel:<password>@cluster0.nfdif.mongodb.net/test

# OPTION 3

# Cliente Mongo Compass



The screenshot shows the MongoDB Compass application interface. On the left, there's a sidebar with 'MongoDB Compass - Connect' (Connect, View, Help), 'New Connection' (selected), 'Favorites', and 'Recents'. The main area has a 'New Connection' dialog with a red box around the connection string input field containing 'mongodb+srv://daniel.daniel123@cluster0.nfdif.mongodb.net/test'. Below it is a 'CONNECT' button. To the right is the main dashboard showing 'Local' (3 DBS, 8 COLLECTIONS) and a 'HOSTS' section listing 'cluster0-shard-00-01.nfdif...', 'cluster0-shard-00-02.nfdif...', and 'cluster0-shard-00-00.nfdif...'. It also shows a 'CLUSTER' section for a 'Replica Set (atlas-bwv12w-... 3 Nodes)' and an 'EDITION' section for 'MongoDB 4.4.4 Enterprise'. The right side features tabs for 'Databases' (selected), 'Performance', and 'CREATE DATABASE'. A table lists databases: 'admin' (0.0B, 0 collections, 0 indexes), 'config' (0.0B, 1 collection, 0 indexes), and 'local' (0.0B, 7 collections, 0 indexes). At the bottom, there's a 'MongoSH Beta' prompt.

# Cliente Web

The screenshot shows the MongoDB Atlas web interface. On the left, the 'Clusters' section is visible, featuring a search bar, a 'Sandbox' cluster entry, and tabs for 'CONNECT', 'METRICS', and 'COLLECTIONS'. A large red arrow points from the 'COLLECTIONS' tab towards the main content area. In the center, the 'Cluster0' section is displayed under the 'Collections' tab. It shows a database named 'test' containing a collection named 'colección'. The 'colección' document is shown with its ID and body. A red box highlights the 'colección' document in the list.

DATABASES: 1 COLLECTIONS: 1

+ Create Database

NAMESPACES

test.colección

COLLECTION SIZE: 49B TOTAL DOCUMENTS: 1 INDEXES TOTAL SIZE: 32KB

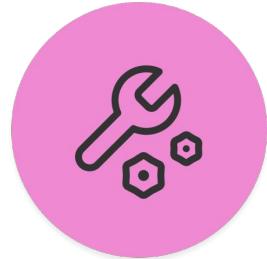
Find Indexes Schema Anti-Patterns Aggregation Search Indexes

QUERY RESULTS 1-1 OF 1

```
_id: ObjectId("6047952f4b905f5a7a318e6b")
body: "Soy un documento"
```

System Status: All Good  
©2021 MongoDB, Inc. Status Terms Privacy Atlas Blog Contact Sales  
Feature Requests





# ***MongoDB Atlas con distintos clientes***

*Tiempo: 10 minutos*



- 1) Conectarse a la base MongoDB Atlas con los clientes **CLI**, **Compass** y **Node.js**
- 2) Mediante la consola CLI insertar estos documentos en una base llamada 'ecommerce', colección 'usuarios':

```
[ { nombre: 'Lucas', apellido: 'Blanco', dni: '30355874' }, { nombre: 'María', apellido: 'García', dni: '29575148' }, { nombre: 'Tomas', apellido: 'Sierra', dni: '38654790' }, { nombre: 'Carlos', apellido: 'Fernández', dni: '26935670' } ]
```
- 3) A través de un proyecto **Node.js**, listar estos datos representándolos en la consola.
- 4) Con el mismo proyecto, incorporar un usuario más:  
`nombre: 'Federico', apellido: 'Perez', dni: '320118321' }`  
Utilizar sintaxis de Promesas con `async await` e `import` para la importación de módulos.
- 5) Con **Compass** borrar al usuario llamado Tomas.
- 6) Con **Mongo CLI** actualizar el usuario llamado 'Carlos' al nombre 'Juan Carlos' y luego listar los documentos finales.



# BREAK

¡5/10 MINUTOS Y VOLVEMOS!



# *Firebase*

**CODER HOUSE**



Firebase

# ¿Qué es y cómo usarlo?

- Firebase es una **plataforma** para el **desarrollo de aplicaciones web y móviles** desarrollada por James Tamplin y Andrew Lee en 2011 y adquirida por Google en 2014, empezando con su producto base: **base de datos en tiempo real**.
- Firebase permite que, en lugar de hacer peticiones AJAX, el usuario se conecte a la base de datos y automáticamente envíe los datos.
- Firebase puede ser administrado por cualquier aplicación backend y hay múltiples dependencias disponibles para lograr la conexión en cualquier plataforma.



# ***Firebase: Ventajas***



- No necesitamos casarnos con Firebase, **se usa lo que se necesita** (por ejemplo quiero sólo Cloud Functions y Remote Config, pero no Real Time database).
- **Usa Cloud Storage:** base de datos para que usuarios puedan compartir ficheros e imágenes, sin necesidad de hacer bases de datos propias, que para imágenes a veces puede ser un poco 'tedioso'.
- **Usa Cloud Functions:** con esto nos ahorraremos toda la infraestructura de backend. Es lo que más cobra Google, ya que sabe que es en lo que más ahorraremos. Con el plan Blaze con las CF puedes hacer llamadas a tu API, no hay firewalls.

# **Firebase Website: <https://firebase.google.com/>**

The screenshot shows the official Firebase website (<https://firebase.google.com/>) displayed in a web browser. The page has a blue header with the Firebase logo and navigation links for Products, Casos de uso, Precios, Documentación, Comunidad, and Asistencia. A search bar and language selection (Español) are also present. The main content area features a large white text area with the headline "Firebase helps you build and run successful apps" and a subtext "Backed by Google and loved by app development teams - from startups to global enterprises". Below this are three buttons: "Comenzar", "Try demo", and "Watch video". To the right, there is a stylized illustration of two people working on a laptop, with a rocket launching and arrows indicating success. The bottom right corner of the image contains the text "CODER HOUSE" in a yellow box.

Firebase

firebase.google.com

Buscar

Español – ... Ir a la consola

Firebase

Productos Casos de uso Precios Documentación Comunidad Asistencia

Buscar

Español – ... Ir a la consola

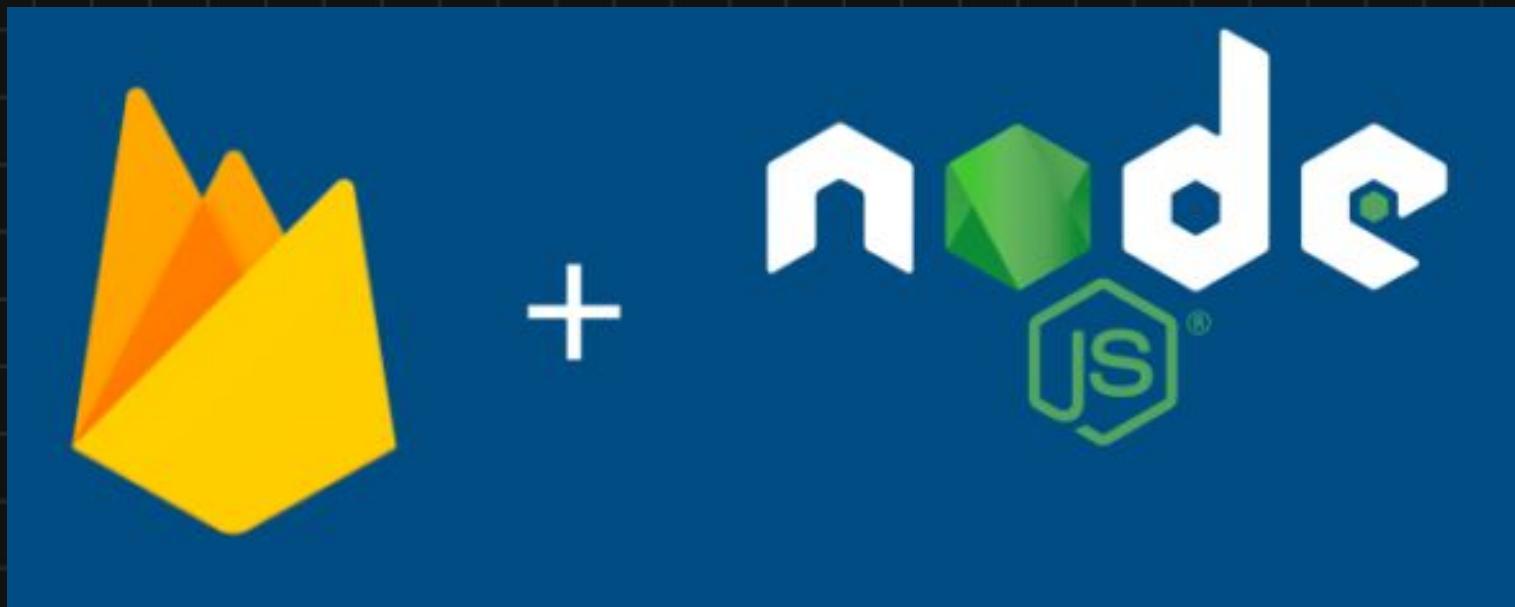
Firebase helps you build and run successful apps

Backed by Google and loved by app development teams - from startups to global enterprises

Comenzar Try demo Watch video

CODER HOUSE

# *Firebase y Node.js*

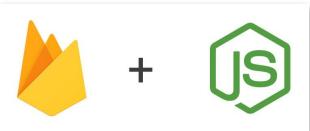




# **CRUD API usando Firebase**



- A modo de ejemplo de interacción de **Node.js con Firebase**, vamos a crear las operaciones necesarias que son: Crear, Leer, Actualizar y Borrar (o **CRUD** por sus siglas en inglés).
- En **Firebase** utilizaremos la opción **Cloud Firestore**. Esta base de datos NoSQL nos permite guardar los datos en forma de documentos y colecciones. Como ya habíamos visto, esto es similar al concepto de filas y columnas en las bases de datos SQL.
- Vamos a seguir a continuación una serie de pasos para configurar y poder acceder a **Cloud Firestore** de **Firebase**.



# Creando un proyecto en Firebase

Vamos a empezar a configurar nuestro entorno, ingresando en la Consola de Firebase: <https://console.firebaseio.google.com/u/0/>

A screenshot of a web browser displaying the Firebase console homepage. The URL in the address bar is https://console.firebaseio.google.com/u/0/. The page features a blue header with the Firebase logo and navigation links. Below the header, there's a section titled 'Tus proyectos de Firebase' (Your Firebase projects) featuring an illustration of a person working at a desk. A white card contains a red-bordered button labeled '+ Agregar proyecto' (Add project). To the right of this card is another white box labeled 'BaseFirebase' with the ID 'carbon-nucleus-264418'. At the bottom of the page, there are two cards: one for 'Explorar un proyecto de demostración' (Explore a demonstration project) and another for 'Los proyectos de Firebase son contenedores para tus apps' (Firebase projects are containers for your apps), which includes icons for iOS, Analytics, and more information.



# Firebase: configuración del proyecto

The screenshot shows the Firebase Project Configuration interface. A red arrow labeled '1' points to the 'Configuración del proyecto' button in the left sidebar. A red arrow labeled '2' points to the 'Cuentas de servicio' tab in the top navigation bar. A large red box labeled '3' highlights the 'Fragmento de configuración de Admin SDK' code editor. A red box labeled '4' highlights the 'Generar nueva clave privada' button at the bottom of the code editor.

1

2

3

4

Configuración del proyecto

Cuentas de servicio

SDK de Firebase Admin

Fragmento de configuración de Admin SDK

```
var admin = require("firebase-admin");

var serviceAccount = require("path/to/serviceAccountKey.json");

admin.initializeApp({
  credential: admin.credential.cert(serviceAccount),
  databaseURL: "https://carbon-nucleus-264418.firebaseio.com"
});
```

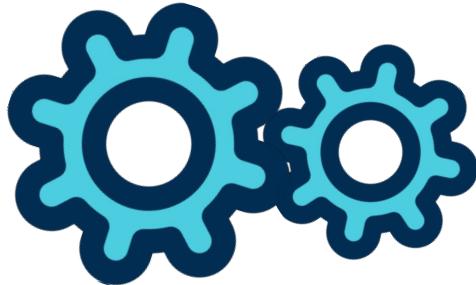
Generar nueva clave privada



# Firebase: vista de la base de datos (Firestore Database)

The screenshot shows the Firebase Cloud Firestore console interface. On the left, a sidebar menu lists various services: Authentication, Cloud Firestore (which is highlighted with a red box and has a red arrow pointing to it), Realtime Database, Storage, Hosting, Functions, and Machine Learning. Below these are sections for Lanzamiento y supervisión (Crashlytics, Performance, Test Lab) and Analytics (Dashboard, Realtime, Events, Conve...). The main content area is titled "Cloud Firestore" and contains tabs for Datos, Reglas, Índices, and Uso. Under the "Datos" tab, there is a section for the project "carbon-nucleus-264418" with a "Iniciar colección" button. A large circular icon representing a database is centered in the main area. At the bottom, the text "Tu base de datos está lista. Solo tienes que agregar datos." is displayed. The footer indicates the location "Ubicación de Cloud Firestore: nam5 (us-central)" and shows a "Spark" section with "Gratis USD 0 por mes".

# *Configuración del proyecto Node.js*



# *Configuración del proyecto Node.js*



1. Creamos un proyecto Node.js con **npm init -y**
2. Instalamos el paquete npm para trabajar con Firebase en la carpeta de nuestro proyecto: **npm i firebase-admin**
3. Incluimos en el proyecto el archivo JSON descargado desde el botón **Generar nueva clave privada** de la configuración de nuestro servidor en modo admin.
4. Generamos el archivo **server.js** y escribimos el código de conexión hacia la base de datos Firebase como se detalla a continuación:



# Configuración del proyecto Node.js



File Edit Selection View Go Run Terminal Help carbon-nucleus-264418.firebaseio-admin

EXPLORER ...

> OPEN EDITORS

✓ NODEFIREBASE

✓ db

{ carbon-nucleus-264418.firebaseio-adminsdk-fe63c-52c4c25c90.json

1 {  
2    "type": "service\_account",  
3    "project\_id": "carbon-nucleus-264418",  
4    "private\_key\_id": "52c4c25c909d5ba34733fdd19eb",  
5    "private\_key": "-----BEGIN PRIVATE KEY-----\n",  
6    "client\_email": "firebase-adminsdk-fe63c@carbon-nucleus-264418.iam.gserviceaccount.com",  
7    "client\_id": "101202556690464271736",  
8    "auth\_uri": "https://accounts.google.com/o/oauth2/auth",  
9    "token\_uri": "https://oauth2.googleapis.com/token",  
10    "auth\_provider\_x509\_cert\_url": "https://www.googleapis.com/oauth2/v1/certs",  
11    "client\_x509\_cert\_url": "https://www.googleapis.com/robot/v1/metadata/x509/firebase-adminsdk-fe63c@carbon-nucleus-264418.iam.gserviceaccount.com"}  
12 }  
13 }

```
const admin = require("firebase-admin");

const serviceAccount = require(
  "./db/carbon-nucleus-264418.firebaseio-adminsdk-fe63c-52c4c25c90.json");

admin.initializeApp({
  credential: admin.credential.cert(serviceAccount),
  databaseURL: "https://carbon-nucleus-264418.firebaseio.com"
});
```

Fragmento de configuración de Admin SDK

Node.js  Java  Python  Go

```
var admin = require("firebase-admin");

var serviceAccount = require("path/to/serviceAccountKey.json");

admin.initializeApp({
  credential: admin.credential.cert(serviceAccount),
  databaseURL: "https://carbon-nucleus-264418.firebaseio.com"
});
```

Generar nueva clave privada



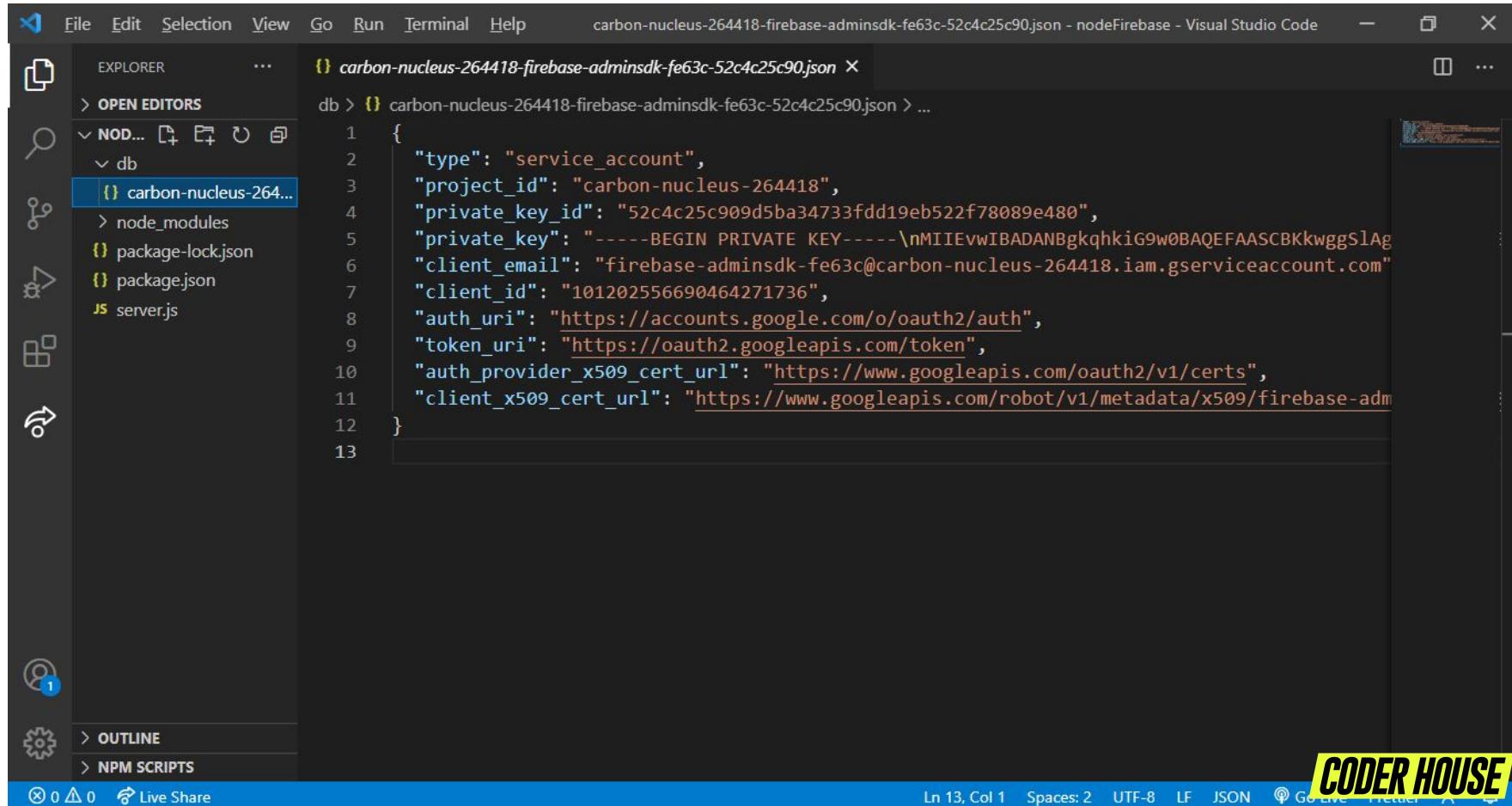
# Proyecto CRUD Node.js Firebase: Creación del proyecto

The screenshot shows the Visual Studio Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** packagejson - nodeFirebase - Visual Studio Code.
- Explorer Panel (Left):** Shows the project structure:
  - EXPLORER
  - > OPEN EDITORS
  - NOD...
  - db
  - { carbon-nucleus-264...
  - > node\_modules
  - { package-lock.json
  - { package.json (highlighted)
  - JS server.js
- Editor Area (Center):** Displays the contents of the package.json file.

```
1 {  
2   "name": "nodefirebase",  
3   "version": "1.0.0",  
4   "description": "",  
5   "main": "server.js",  
6   "scripts": {  
7     "test": "echo \\"Error: no test specified\\" && exit 1",  
8     "start": "node server.js"  
9   },  
10  "keywords": [],  
11  "author": "",  
12  "license": "ISC",  
13  "dependencies": {  
14    "firebase-admin": "^9.5.0"  
15  }  
16}  
17
```
- Bottom Status Bar:** Shows file status (0 changes), Live Share icon, and file statistics (Ln 14, Col 31, Spaces: 2, UTF-8, LF, JSON).
- Bottom Right Logo:** CODER HOUSE

# Proyecto CRUD Node.js Firebase: JSON credenciales



The screenshot shows the Visual Studio Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** carbon-nucleus-264418-firebase-adminsdk-fe63c-52c4c25c90.json - nodeFirebase - Visual Studio Code.
- Explorer Panel:** Shows the project structure:
  - NODE\_MODULES
  - db
  - carbon-nucleus-264418 (selected)
    - node\_modules
    - package-lock.json
    - package.json
    - server.js

- Editor Panel:** Displays the contents of the selected JSON file:

```
1  {
2    "type": "service_account",
3    "project_id": "carbon-nucleus-264418",
4    "private_key_id": "52c4c25c909d5ba34733fdd19eb522f78089e480",
5    "private_key": "-----BEGIN PRIVATE KEY-----\nMIIEvwIBADANBgkqhkiG9w0BAQEFAASCBKkwggSlAg",
6    "client_email": "firebase-adminsdk-fe63c@carbon-nucleus-264418.iam.gserviceaccount.com",
7    "client_id": "101202556690464271736",
8    "auth_uri": "https://accounts.google.com/o/oauth2/auth",
9    "token_uri": "https://oauth2.googleapis.com/token",
10   "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",
11   "client_x509_cert_url": "https://www.googleapis.com/robot/v1/metadata/x509.firebaseio-adminsdk-fe63c.iam.gserviceaccount.com"
12 }
13 }
```
- Bottom Status Bar:** Ln 13, Col 1, Spaces: 2, UTF-8, LF, JSON, Go LIVE, Predictor.
- Bottom Right Logo:** CODER HOUSE

# Proyecto CRUD Node.js Firebase: Conexión a la base

The screenshot shows a Visual Studio Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** server.js - nodeFirebase - Visual Studio Code.
- Explorer View (Left):** Shows the project structure:
  - OPEN EDITORS: JS server.js
  - NOD... (selected):
    - db
    - { } carbon-nucleus-264...
    - > node\_modules
    - { } package-lock.json
    - { } package.json
  - JS server.js
- Code Editor (Center):** Displays the `server.js` file content:

```
1 const admin = require("firebase-admin");
2
3 const serviceAccount = require(
4   "./db/carbon-nucleus-264418-firebase-adminsdk-fe63c-52c4c25c90.json"
5 );
6
7 admin.initializeApp({
8   credential: admin.credential.cert(serviceAccount),
9   databaseURL: "https://carbon-nucleus-264418.firebaseio.com"
10 });
11
12 console.log('Base Firebase conectada!')
13
14
15
16
17
18
19
20
21
```
- Bottom Status Bar:** Ln 13, Col 1, Spaces: 4, UTF-8, CRLF, JavaScript, Live Share, Go LIVE, Fpreter.
- Bottom Right Logo:** CODER HOUSE

# Proyecto CRUD Node.js Firebase: CREATE

The screenshot shows the Visual Studio Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** server.js - nodeFirebase - Visual Studio Code.
- Explorer View (Left):** Shows the project structure:
  - OPEN EDITORS: server.js
  - NOD... (selected):
    - db
    - { } carbon-nucleus-264...
    - > node\_modules
    - { } package-lock.json
    - { } package.json
  - server.js
- Code Editor (Right):** Displays the server.js code for a CRUD operation using Firestore.

```
26  CRUD()
27
28  async function CRUD() {
29
30      const db = admin.firestore();
31      const query = db.collection('usuarios')
32
33      /* ----- */
34      /* CREATE */
35      try {
36          //const doc = query.doc()    // para generación automática de id
37          let id = 1
38          let doc = query.doc(`#${id}`) // para generación manual de id
39          await doc.create( { nombre: 'Jose', dni: 11223344 })
40          id++
41          doc = query.doc(`#${id}`) // para generación manual de id
42          await doc.create( { nombre: 'Ana', dni: 22334455 })
43          id++
44          doc = query.doc(`#${id}`) // para generación manual de id
45          await doc.create( { nombre: 'Diego', dni: 33445566 })
46
47          console.log('Datos insertados')
48      } catch (error) { console.log(error) }
49
50      /* ----- */
```
- Bottom Status Bar:** Ln 13, Col 1, Spaces: 4, UTF-8, CRLF, JavaScript, Go Live, F12, etc.
- Bottom Right Logo:** CODER HOUSE

# Proyecto CRUD Node.js Firebase: READ ALL y READ ID

The screenshot shows the Visual Studio Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** server.js - nodeFirebase - Visual Studio Code.
- Left Sidebar (Explorer):** Shows the project structure:
  - NODE\_MODULES
  - db (with carbon-nucleus-264...)
  - node\_modules
  - package-lock.json
  - package.json
  - server.js (selected)
- Right Sidebar:** Includes a terminal window showing logs and a preview pane.
- Code Editor:** The file "server.js" contains the following code:

```
/* ----- */
/* READ ALL */
try{
    const querySnapshot = await query.get()
    let docs = querySnapshot.docs;

    const response = docs.map((doc) => ({
        id: doc.id,
        nombre: doc.data().nombre,
        dni: doc.data().dni,
    }))
    console.log(response);
} catch (error) { console.log(error) }

/* ----- */
/* READ ID */
try{
    let id = 2
    const doc = query.doc(`#${id}`)
    const item = await doc.get()
    const response = item.data()
    console.log(response);
} catch (error) { console.log(error) }
```
- Bottom Status Bar:** Ln 13, Col 1, Spaces: 4, UTF-8, CRLF, JavaScript, Go LIVE, Fpreter.
- Bottom Right Logo:** CODER HOUSE

# Proyecto CRUD Node.js Firebase: UPDATE y DELETE

The screenshot shows a Visual Studio Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** server.js - nodeFirebase - Visual Studio Code.
- Left Sidebar (Explorer):** Shows the project structure:
  - NODE\_MODULES
  - db (carbon-nucleus-264...)
  - node\_modules
  - package-lock.json
  - package.json
  - server.js (selected)
- Central Area (Code Editor):** Displays the `server.js` file content. The code handles UPDATE and DELETE operations on a Firestore database. It uses `query.doc(`${id}`)` to target specific documents and `doc.update()` or `doc.delete()` to perform the respective actions. Error handling is included with `try`, `catch`, and `console.log`.
- Right Sidebar:** Shows the Firebase Realtime Database interface with a list of users and their details.
- Bottom Status Bar:** Ln 13, Col 1, Spaces: 4, UTF-8, CRLF, JavaScript, Go LIVE, F12, etc.
- Bottom Right Logo:** CODER HOUSE

# Proyecto CRUD Node.js Firebase: Salida de consola

File Edit Selection View Go Run Terminal Help server.js - nodeFirebase - Visual Studio Code

EXPLORER OPEN EDITORS NOD... db carbon-nucleus-264... node\_modules package-lock.json package.json server.js

server.js > CRUD

```
26  CRUD()
27
28  async function CRUD() {
29
30      const db = admin.firestore();
31      const query = db.collection('usuarios')
```

TUTORIALS RECENT CHANGES

TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE 1: powershell

```
PS C:\Cursos\Coderhouse\CursoBackend\Clase21\nodeFirebase> node .
Base Firebase conectada!
Datos insertados
[
  { id: '1', nombre: 'Jose', dni: 11223344 },
  { id: '2', nombre: 'Ana', dni: 22334455 },
  { id: '3', nombre: 'Diego', dni: 33445566 }
]
{ nombre: 'Ana', dni: 22334455 }
El usuario ha sido actualizado WriteResult {
  _writeTime: Timestamp { _seconds: 1615491362, _nanoseconds: 340986000 }
}
El usuario ha sido borrado exitosamente WriteResult {
  _writeTime: Timestamp { _seconds: 1615491362, _nanoseconds: 629838000 }
}
[
  { id: '2', nombre: 'Ana', dni: 55443322 },
  { id: '3', nombre: 'Diego', dni: 33445566 }
]
```

OUTLINE NPM SCRIPTS

Ln 94, Col 5 Spaces: 4 UTF-8 CRLF JavaScript Go LIVE Fpreter A

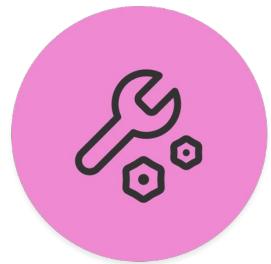
CODER HOUSE

# Proyecto CRUD Node.js Firebase: Consola Web

The screenshot shows the Firebase Cloud Firestore interface. On the left, a sidebar lists various services: Authentication, Cloud Firestore (selected), Realtime Database, Storage, Hosting, Functions, Machine Learning, Lanzamiento y supervisión, Analytics, Participación, and Extensions. At the bottom of the sidebar, it says "Spark Gratis USD 0 por mes" and "Actualizar". The main area is titled "Cloud Firestore" and shows the "Datos" tab selected. It displays a hierarchical view: "BaseFirebase > usuarios > 2". The "usuarios" collection contains three documents, numbered 2 and 3. Document 2 has fields "dni: 55443322" and "nombre: "Ana"". A modal window is open over the collection list, showing options to "Iniciar colección" (Create collection) and "Agregar documento" (Add document). The URL in the browser bar is "console.firebaseio.google.com/u/0/project/carbon-nucleus-264418/firestore/data~2Fusuarios~2F2".

Ubicación de Cloud Firestore: nam5 (us-central)

**CODER HOUSE**



# ***Node.js con Firebase***

*Tiempo: 10 minutos*



Realizar un proyecto en Node.js que acceda a una base de datos Firebase ya configurada.

- 1) **Agregar** los colores red, green, blue dentro de una colección llamada 'colores' con el formato { nombre: color }
- 2) **Listar** todos los colores disponibles.
- 3) **Modificar** el color blue por navy.
- 4) **Borrar** el color green

### A tener en cuenta:

- Implementar estas funciones utilizando Promises en las funciones de Firebase con sintaxis async/await, utilizando la importación en formato ES Modules (import)
- Verificar la información de la base de datos con la consola de Firebase.



# **SEGUNDA ENTREGA DEL PROYECTO FINAL**

Deberás entregar el avance de tu aplicación eCommerce Backend correspondiente a la segunda entrega de tu proyecto final.

# **SEGUNDA ENTREGA DEL PROYECTO FINAL**

**Formato:** link a un repositorio en Github con el proyecto cargado.

**Sugerencia:** no incluir los node\_modules

Proyecto  
Final



**>>Consigna:** Basándose en los contenedores ya desarrollados (memoria, archivos) desarrollar dos contenedores más (que cumplan con la misma interfaz) que permitan realizar las operaciones básicas de CRUD en MongoDb (ya sea local o remoto) y en Firebase. Luego, para cada contenedor, crear dos clases derivadas, una para trabajar con Productos, y otra para trabajar con Carritos.

# **SEGUNDA ENTREGA DEL PROYECTO FINAL**

**Formato:** link a un repositorio en Github con el proyecto cargado.

**Sugerencia:** no incluir los node\_modules

Proyecto  
Final



## **>>Aspectos a incluir en el entregable:**

- a. A las clases derivadas de los contenedores se las conoce como DAOs (Data Access Objects), y pueden ir todas incluidas en una misma carpeta de 'daos'.
- b. En la carpeta de daos, incluir un archivo que importe todas las clases y exporte una instancia de dao de productos y una de dao de carritos, según corresponda. Esta decisión se tomará en base al valor de una variable de entorno cargada al momento de ejecutar el servidor (opcional: investigar el uso de imports dinámicos).
- c. Incluir un archivo de configuración (config) que contenga los datos correspondientes para conectarse a las bases de datos o medio de persistencia que corresponda.

## **>>>Opcional:**

- d. Hacer lo mismo para bases de datos relacionales: MariaDB/SQLite3.

# SEGUNDA ENTREGA DEL PROYECTO FINAL

**Formato:** link a un repositorio en Github con el proyecto cargado.

**Sugerencia:** no incluir los node\_modules

Proyecto  
Final



```
import { promises as fs } from 'fs'

class ContenedorArchivo {
    constructor(ruta) {
        this.ruta = ruta;
    }

    async listar(id) {
        const objs = await this.listarAll()
        const buscado = objs.find(o => o.id === id)
        return buscado
    }

    async listarAll() {
        try {
            const objs = await fs.readFile(this.ruta)
            return JSON.parse(objs)
        } catch (error) {
            return []
        }
    }
}
```

## Ejemplos

```
import ContenedorArchivo from "../..../contenedores/ContenedorArchivo"

class ProductosDaoArchivo extends ContenedorArchivo {
    constructor() {
        super('DB/productos.json')
    }

    async desconectar() {
    }

    export default ProductosDaoArchivo
}
```

```
src
└── contenedores
    ├── ContenedorArchivo.js
    ├── ContenedorFirebase.js
    ├── ContenedorMemoria.js
    └── ContenedorMongoDb.js
└── daos
    ├── CarritosDaoArchivo.js
    ├── CarritosDaoFirebase.js
    ├── CarritosDaoMem.js
    └── CarritosDaoMongoDb.js
└── productos
    ├── ProductosDaoArchivo.js
    ├── ProductosDaoFirebase.js
    ├── ProductosDaoMem.js
    └── ProductosDaoMongoDb.js
└── index.js
└── utils
    ├── config.js
    └── main.js
└── server.js
```

*¿*PREGUNTAS?



# ***¡MUCHAS GRACIAS!***

Resumen de lo visto en clase hoy:

- DBaaS
- Mongo Atlas
- Firebase
- Integración con Node.js



***OPINA Y VALORA ESTA CLASE***

**#DEMOCRATIZANDOLAEDUCACIÓN**