



**Argentina  
programa  
4.0**



Ministerio de Economía  
**Argentina**

Secretaría de  
Economía del Conocimiento

***primero  
la gente***

# Clase 31: Node.js y Express

## Publicar un proyecto en la nube - Por dónde seguir

## Agenda de hoy

- A. Publicar un proyecto backend en la nube
- B. Opciones para publicar nuestro proyecto backend
  - a. Limitaciones de las nubes
  - b. Consignas del proyecto final
  - c. Por dónde seguir (Node JS)
- C. Otros conceptos importantes en backend
  - a. Manejo de Sockets
  - b. Consumir endpoint remotos
  - c. Manejo de Sesiones y Cookies
  - d. Por dónde seguir (Bases de Datos)



# **Publicar un proyecto backend en la nube**

# Publicar un proyecto backend en la nube

Hoy todo es una nube. Desde las más populares, como lo son [AWS](#), [Azure](#), o [GCP](#), hasta las menos resonantes pero con gran aceptación, todas se han vuelto una herramienta fundamental para alojar y desplegar aplicaciones.

En esta última entrega del curso, veremos cómo podemos publicar nuestros proyectos backend de forma gratuita en alguna de las nubes existentes.



# Publicar un proyecto backend en la nube

Cuando tenemos que publicar algún proyecto cloud en la nube, para una instancia como la nuestra que es utilizarlo como Portfolio y realizar experiencia en este terreno, debemos tener en cuenta algunos factores específicos:

- Ventajas de las nubes
- Plataformas gratuitas
- Pasos para publicar
- Conclusiones



# Publicar un proyecto backend en la nube

## Ventajas de la nube

Al elegir la nube para alojar nuestro proyecto backend, disfrutamos de numerosas ventajas. En primer lugar, no tenemos que preocuparnos por la infraestructura física, ya que la nube nos brinda una plataforma virtualizada y escalable que se adapta a nuestras necesidades.



# Publicar un proyecto backend en la nube

## Ventajas de la nube

Además, la nube nos ofrece un alto nivel de disponibilidad y redundancia, lo que garantiza que nuestra aplicación esté siempre en línea y accesible para los usuarios.





# Publicar un proyecto backend en la nube

## Plataformas gratuitas

Existen varias plataformas en la nube que ofrecen planes gratuitos para alojar proyectos backend desarrollados en **Node.js**.

Estas plataformas nos permiten desplegar nuestras aplicaciones de manera sencilla y sin incurrir en costos.



# Publicar un proyecto backend en la nube

## Plataformas gratuitas

Algunas de las opciones más populares incluyen a:

- Netlify
- Alibaba Cloud
- Render
- Fly.io
- Railway
- Glitch

Además de opciones varias que podemos encontrar en Google Cloud Platform, Microsoft Azure o Amazon Web Services.



# Publicar un proyecto backend en la nube

## Plataformas gratuitas

Estas plataformas proporcionan herramientas y servicios adicionales que facilitan la gestión y el monitoreo de nuestras aplicaciones en la nube.

Pero **un punto importante que debemos tener en cuenta** es que, **toda plataforma Cloud cambia constantemente sus *Términos y Condiciones***, por lo cual, lo que hoy puede ser gratuito en alguna plataforma, en algún tiempo indeterminado puede dejar de serlo.



# Publicar un proyecto backend en la nube

## Plataformas gratuitas

Estas plataformas proporcionan herramientas y servicios adicionales que facilitan la gestión y el monitoreo de nuestras aplicaciones en la nube.

Pero **un punto importante que debemos tener en cuenta** es que, **toda plataforma Cloud cambia constantemente sus *Términos y Condiciones***, por lo cual, lo que hoy puede ser gratuito en alguna plataforma, en algún tiempo indeterminado puede dejar de serlo.



# Publicar un proyecto backend en la nube

## Pasos para publicar un proyecto backend en la nube

Para publicar nuestro proyecto backend en la nube, seguimos algunos pasos clave. En primer lugar, creamos una cuenta en la plataforma de nuestra elección y configuramos nuestra aplicación.

Luego, utilizamos herramientas como la línea de comandos o la integración con sistemas de control de versiones para enviar nuestro código a la nube.



# Publicar un proyecto backend en la nube

## Pasos para publicar un proyecto backend en la nube

A continuación, configuramos variables de entorno y otros parámetros necesarios para que nuestra aplicación funcione correctamente en el entorno de la nube.

Por último, desplegamos nuestra aplicación y la ponemos en funcionamiento en la nube.

Casi todas las nubes cuentan hoy con herramientas vía **Terminal** (npm), para realizar estos procesos.



# **Limitaciones de las nubes**

## Limitaciones de las nubes

Como bien venimos hablando, cada nube tiene su limitación en cuanto a qué servicios podemos utilizar gratuitamente, y cuáles requieren un pago.

Incluso en algunos casos, nos dan una capa gratuita bastante interesante, donde podremos publicar proyectos de media o alta complejidad, con múltiples bases de datos y mecanismos de seguridad. Pero esta publicación tendrá un límite de tiempo de gratuidad. Luego de ese límite, seguramente debamos comenzar a pagar un abono por los servicios utilizados.





# Limitaciones de las nubes

Es clave poder analizar todo esto con el mayor detenimiento posible, y hasta incluso prestar atención a posibles límites internos, como ser una cuota de uso (xxx MB / mes ó xx GB / mes).

Todo tiene su costo y, a diferencia de las redes sociales, el mantenimiento operativo de estos servicios específicos, no pueden vivir de cuotas publicitarias como sí lo pueden hacer Facebook, Twitter o LinkedIn.



# Limitaciones de las nubes

En otros escenarios, podemos tal vez montar una aplicación Node.js en nube de forma gratuita, pero no podemos utilizar una bb.dd SQL o MongoDB local.

Algunas pueden permitirnos algunas bb.dd locales con un tiempo limitado de activación diario, otras pueden permitir una conexión a una bb.dd remota como ser los clúster de MongoDB, y otras plataformas tal vez sólo permitan un archivo JSON que simule un set de datos a servir vía API RESTFUL.



# **Publicando un proyecto Cloud**

# Publicando un proyecto Cloud

**Veamos a continuación la sugerencia de la profe, para ver una plataforma Cloud donde podemos publicar algún proyecto basado en Node.js, de todos los que realizamos a lo largo de la cursada.**



# Espacio dedicado al proyecto final

Abrimos nuevamente un espacio para que sigas consultándonos sobre el proyecto final, nos cuentes cómo vas trabajando en éste, y/o te animes a mostrar tus avances compartiendo pantalla.



# **Por dónde seguir (Node JS)**

## Por dónde seguir (Node JS)

**Estamos llegando al final de la cursada. Pero esto no es todo, aún nos queda mucho camino por recorrer.**

Las bases de este curso te dejan los conocimientos más importantes de Node.js del lado del backend (los pilares).

Y contando con los pilares de esta fabulosa herramienta, puedes continuar el perfeccionamiento de la misma enfocándote en aprender el resto de las tecnologías satélite que la complementan.



*Node.js,  
Node.js  
everywhere...!*

# Más backend con Node.js

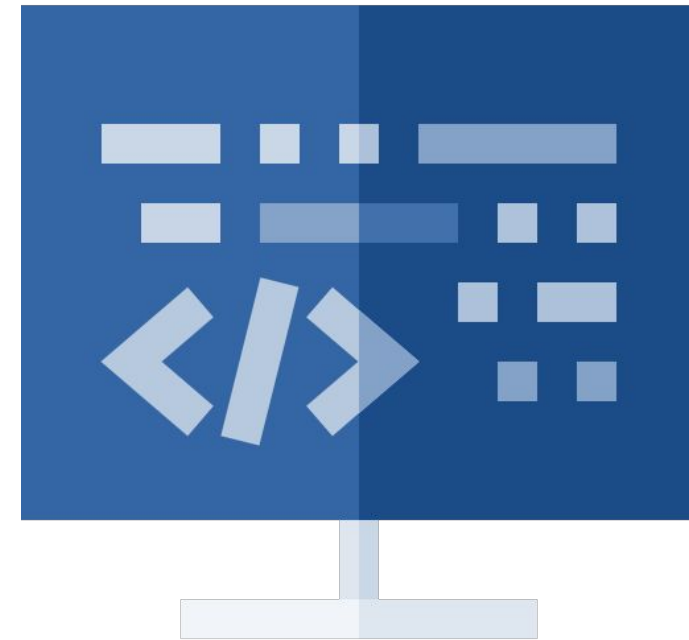


# Más backend con Node.js

## Bases de datos

Hay mucho más de backend en Node.js. En este curso nos enfocamos en explorar solo dos bases de datos específicas y de las más demandadas a nivel laboral: (MongoDB y MySQL)

Con las bases de MySQL aprendidas, puedes aprovechar y utilizar, por ejemplo, SQL Server, para entender cómo se integra este otro motor de bb.dd a Node.JS, utilizando el mismo Driver.



# Más backend con Node.js

## Bases de datos

De la mano de MongoDB, otra base de datos similar a MongoDB es Apache Cassandra. Esta es una bb.dd NoSQL, la cual almacena datos en formato JSON-like, y está diseñada para manejar grandes volúmenes de datos.

Puedes integrar Cassandra con Node.js utilizando el conector **Cassandra-Driver**, para realizar las operaciones CRUD.



# Más backend con Node.js

## Bases de datos

De la mano de MongoDB, otra base de datos similar a MongoDB es Apache Cassandra. Esta es una bb.dd NoSQL, la cual almacena datos en formato JSON-like, y está diseñada para manejar grandes volúmenes de datos.

Puedes integrar Cassandra con Node.js utilizando el conector **Cassandra-Driver**, para realizar las operaciones CRUD.



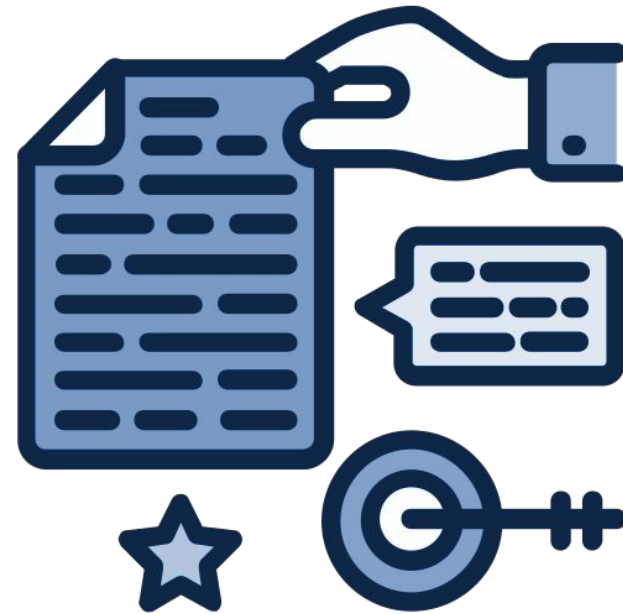
# Seguridad

# Más backend con Node.js

## Seguridad

Hemos aprendido a manejar los niveles de seguridad, integrando JWT como herramienta generadora de Token.

Además de esta, en la actualidad se utiliza **Passport.js** para el manejo de Login y Registro de usuarios. Es ideal que adquieras experiencia en el manejo de la misma.

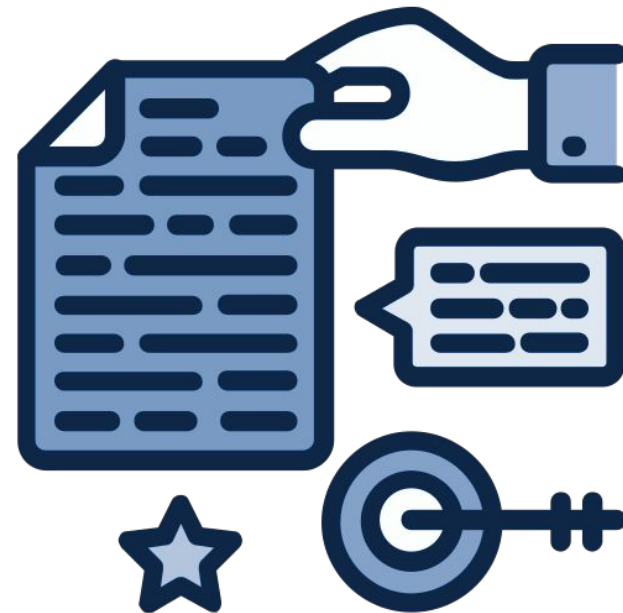


# Más backend con Node.js

## Seguridad

**Passport.js** nos permite crear y validar usuarios a través de un registro previo y el almacenamiento de contraseñas encriptadas en bb.dd dedicadas.

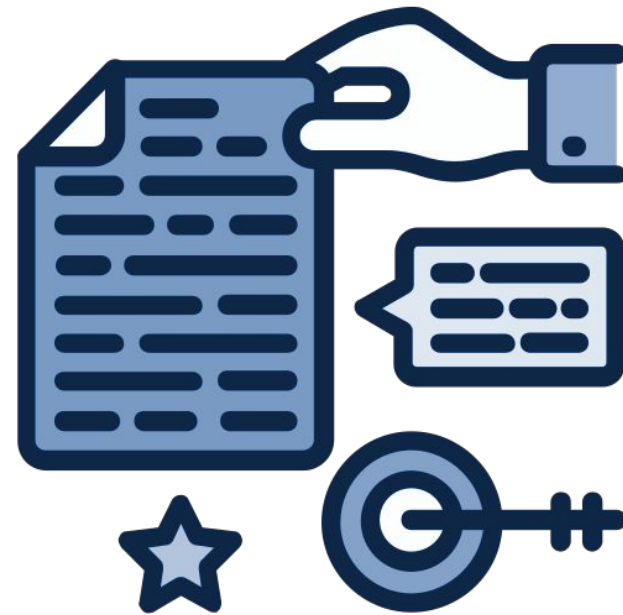
También nos da soporte para integrar validaciones de usuarios a través de cuentas de correo de: Facebook, Github, Google o Microsoft, entre otras opciones.



# Más backend con Node.js

## Seguridad

Y dentro de esta gama de validación y registro de usuarios, también tenemos a **OAuth 2.0**; otra opción completamente integrable a Node.js, la cual nos permite validar mediante el servicio de esta plataforma, el registro de usuarios genéricos o a través de cuentas existentes de las redes sociales más populares.

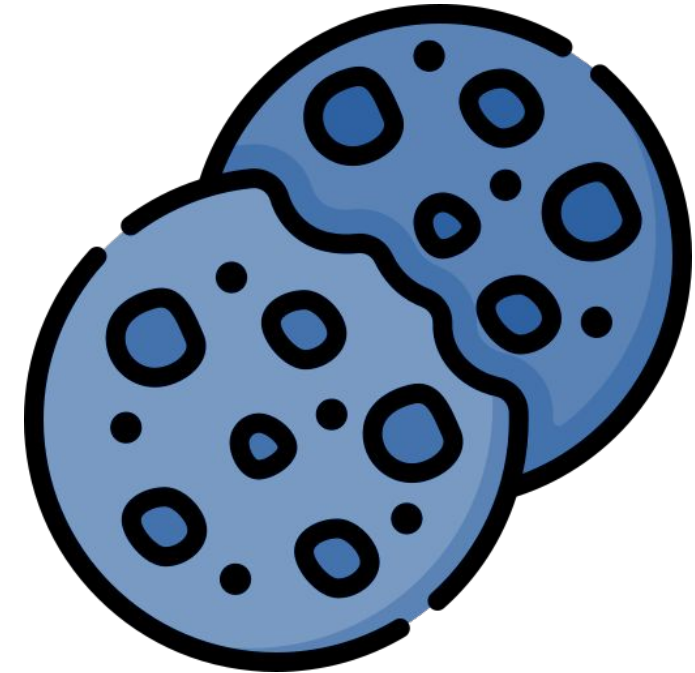


# Más backend con Node.js

## Seguridad

Incluso, todos los servicios anteriores relacionados a la seguridad, se pueden complementar con el manejo de sesiones y cookies.

Estas son opciones disponibles para trabajar desde Node.js, de una forma fácil y rápida de implementar.





# Mensajería

# Más backend con Node.js

## Mensajería

Existen también varias opciones para integrar mecanismos de mensajería en aplicaciones Node.js. Por ejemplo, algunos servicios válidos y muy requeridos hoy, son aquellos que permiten el envío de SMS.

Entre los más populares encontramos a **Twilio**, **Nexmo** y **Plivo**. Pero son servicios de terceros que brindan sus propias APIs que debemos consumir.



# Más backend con Node.js

## Mensajería

En estos casos, tal vez sea más difícil probar algunos servicios, pero si tal vez conviene leer la documentación de ellos para entender a nivel teórico, cómo se deben implementar, saber qué opciones existen, y cómo cobran el uso de los servicios tercerizados.

A mayor conocimiento, más chances tendremos en el mercado laboral, y menos deberemos aprender al insertarnos en éste.



# Más backend con Node.js

## Mensajería

Otras plataformas que debemos conocer también, son las que nos permiten enviar mensajes de correo electrónico. Aquí encontramos al módulo **Nodemailer**, como el candidato ideal para estos procesos.

Este nos permite conectarnos a proveedores de servicios de correo electrónico vía (*SMTP, SendGrid, Gmail, entre otros tantos*), de forma sencilla y rápida.



# Más backend con Node.js

## Mensajería

Con un código simple y conciso, podremos implementar soluciones de envío de mensajes de validación de usuarios a un correo electrónico, para recuperar una contraseña, o para registrar por primera vez un usuario en una plataforma.

Nodemailer es muy fácil de implementar y usar, y se complementa muy bien con Passport.js u OAuth 2.0.

```
Nodemailer

const nodemailer = require('nodemailer');

const transporter = nodemailer.createTransport({
  service: 'Gmail',
  auth: {
    user: 'tucorreo@gmail.com',
    pass: 'tupassword',
  },
});

const mailOptions = {
  from: 'tucorreo@gmail.com',
  to: 'destinatario@example.com',
  subject: '¡Hola!',
  text: 'Este es un mensaje de prueba.',
};

transporter.sendMail(mailOptions, (error, info) => {
  if (error) {
    console.error('Error al enviar el correo:', error);
  } else {
    console.log('Correo enviado:', info.response);
  }
});
```

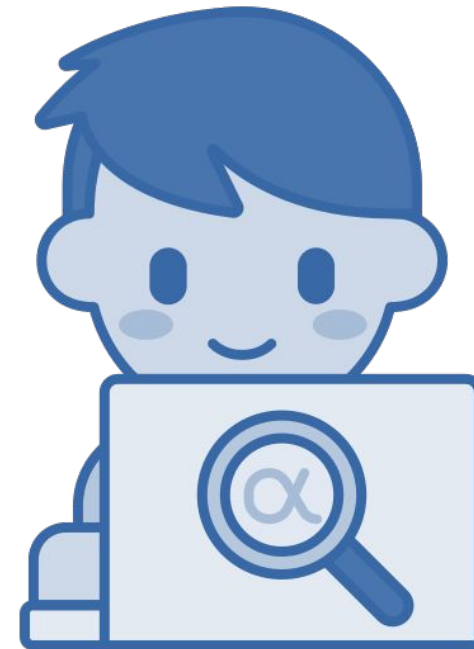
# Servicios

# Más backend con Node.js

## Servicios

Puedes también generar experiencia creando algunos servicios rápidos para los usuarios.

Entre ellos podemos mencionar a algunos populares, como ser, un acortador de URL's, un generador de códigos QR, o un generador de Token numérico como los de las aplicaciones de home banking.



# Más backend con Node.js

## Acortador de URLs

Un acortador de URL's es un servicio fácil y práctico de realizar, recibiendo una URL extensa como parámetro, almacenando la misma en una bb.dd, y retornando caracteres alfanuméricos creados con **Crypto.randomUUID()**.

Puedes tomar la primera parte del UUID generado, y entregarlo como el acortador de URL. Luego generas un segundo Endpoint que reciba el token UUID y retorne la URL vinculada a este.

```
URL Shortener

const URLShortener = Crypt.randomUUID();

const shortenerFirstPart = URLShortener.spli("-");

console.log(shortenerFirstPart[0]);
```





# Más backend con Node.js

## Generador de códigos QR

El módulo **qrcode** facilita la generación de códigos QR utilizando Node.js. Al instalarlo, solo debemos enviarle un texto específico, el cual puede provenir de un String simple, una URL, una estructura de datos JSON, entre otras.

A través del texto recibido, generamos una URL que retorne una imagen QR la cual tendrá embebida la información en cuestión.

```
QR Code

const qr = require('qrcode');

// Generar un código QR a partir de un texto
qr.toDataURL('Hola, mundo!', (err, url) => {
  if (err) {
    console.error('Error al generar el código QR:', err);
    return;
  }

  console.log('URL del código QR:', url);
});
```



# Más backend con Node.js

## Generador de códigos numéricos

El generador de códigos numéricos es ideal para demostrar la misma interacción que utiliza una aplicación móvil actual la cual complementa a los home banking.

El código numérico se podrá mostrar en una aplicación frontend, y a través de un segundo endpoint, validar su efectividad. Incluso puedes complementarlo con Nodemailer para que envíe el código a un correo electrónico específico.

```
Token Number  
  
const tokenNumber = parseInt(Math.random() * 1000000);  
  
console.log(tokenNumber);
```

# Manejo de socket

# Más backend con Node.js

## Sockets

El manejo de sockets con Node.js es especialmente útil en aplicaciones en tiempo real que requieren comunicación bidireccional entre el servidor y el cliente. Socket.io es el módulo más popular en Node.js, y podremos crear:

- aplicaciones colaborativas
- Videojuegos multiplayer
- Aplicaciones de seguimiento en tiempo real
- Aplicaciones de streaming en tiempo real



# Más backend con Node.js

## Sockets

Dentro de los ejemplos que más se utiliza Socket.io, es para la creación de un chat en tiempo real.

Esta tecnología es aplicable en terrenos actuales como los de ChatGPT, WhatsApp, el seguimiento de vehículos en tiempo real, entre otros.

No debemos dominarla a la perfección, pero sí entender cómo funciona y experimentarla con un ejemplo simple.



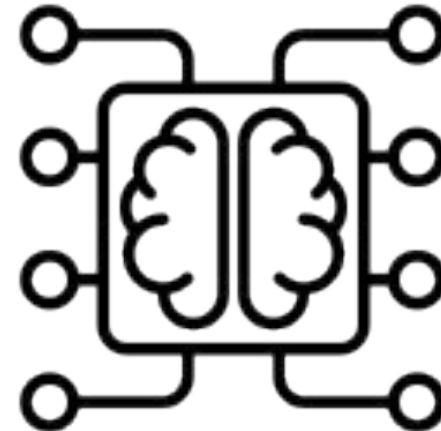
# Contenedores

# Más backend con Node.js

## Contenedores

Los contenedores son una forma de encapsular y distribuir aplicaciones junto con todas sus dependencias en un entorno aislado y portátil.

A diferencia de las máquinas virtuales, los contenedores no requieren un sistema operativo completo, lo que los hace más livianos y rápidos de implementar.



# Más backend con Node.js

## Contenedores

**Kubernetes** es una plataforma de orquestación de contenedores de código abierto ampliamente utilizada para automatizar y gestionar la implementación, escalado y administración de aplicaciones en contenedores.

Esta plataforma distribuye y equilibra automáticamente las cargas de trabajo en función de las necesidades de la aplicación, lo que permite una mayor disponibilidad, escalabilidad y tolerancia a fallos.





# Más backend con Node.js

## Contenedores

**Docker** es una plataforma de virtualización de contenedores que simplifica el desarrollo y despliegue de aplicaciones.

Empaqueta las dependencias de una aplicación en un contenedor independiente y liviano, garantizando la portabilidad y reproducibilidad del entorno.

Así podremos ejecutar aplicaciones de manera consistente en diferentes entornos, desde el desarrollo hasta la producción, logrando una mayor eficiencia en el uso de recursos.



**Por dónde seguir  
(Bases de datos)**

## Más backend con Bases de Datos

De la mano de MySQL hemos recorrido un gran camino con este, el cual no solo nos deja un poder de conocimiento importante, sino que lo podemos aprovechar para utilizar cualquier otro motor alternativo que maneje también el lenguaje SQL, y así conseguir experiencia con otros motores.



## Más backend con Bases de Datos

Para terminar de complementar todo lo aprendido de la mano de MySQL, te recomendamos agregar a tus conocimientos:

- Funciones almacenadas
- Procedimientos almacenados
- Usuarios y Roles



# Más backend con Bases de Datos

## Funciones almacenadas

Recordemos que, una función almacenada, es un bloque de código reutilizable que realiza una tarea específica y puede ser invocado desde una consulta para obtener un resultado.

Aprenderlas nos ayudará a conocer el lenguaje SQL enfocado en el ámbito de la programación.



# Más backend con Bases de Datos

## Usuarios y Roles

La administración de usuarios y roles en MySQL es el proceso de asignar y gestionar los permisos y privilegios de los usuarios, permitiendo un control preciso sobre el acceso y las acciones que pueden realizar en la base de datos.

Es totalmente complementable con Node.js y con los diferentes perfiles de usuarios pensados a nivel seguridad.



# Plataformas Cloud

# Plataformas Cloud

**PaaS:** Plataforma como servicio (PaaS) es un modelo en la nube que ofrece a los desarrolladores un entorno completo para desarrollar, probar y desplegar aplicaciones sin tener que preocuparse por la infraestructura subyacente. *(el lugar donde publicamos nuestra aplicación backend)*

**SaaS:** Software como servicio (SaaS) es un modelo de entrega de software en la nube en el que las aplicaciones están alojadas y gestionadas por un proveedor de servicios, y los usuarios acceden a ellas a través de la web. *(Google Drive, OneDrive, Dropbox, entre otras plataformas de uso frecuente)*

**IaaS:** Infraestructura como servicio (IaaS) es un modelo en la nube que proporciona a los usuarios una infraestructura informática completa, incluyendo servidores virtuales, almacenamiento y redes, permitiendo a los usuarios desplegar y gestionar sus propias aplicaciones y sistemas operativos.

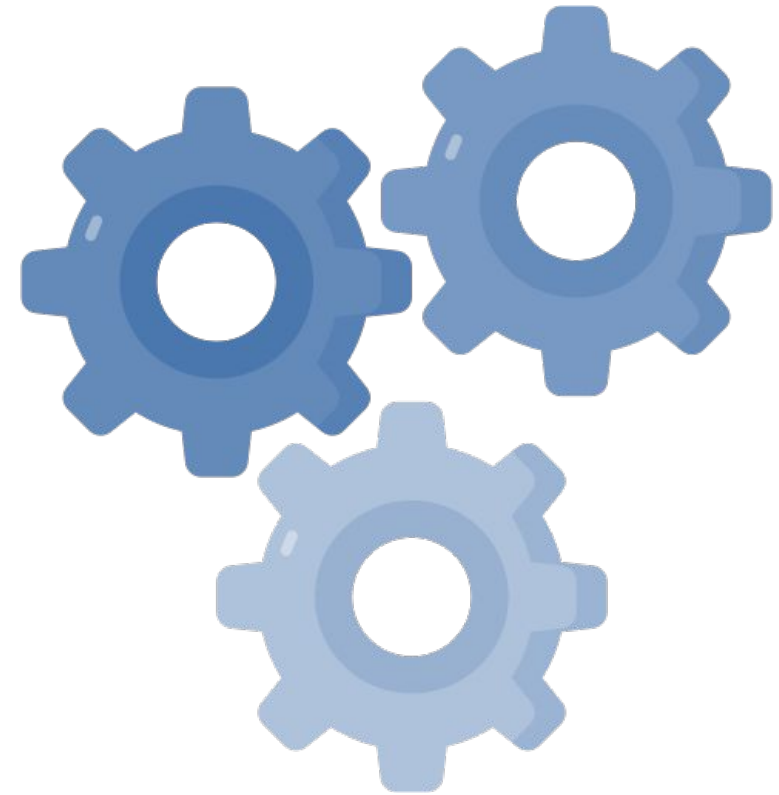




# Plataformas Cloud

No podremos conocer en detalle todos los servicios PaaS, IaaS y SaaS del mercado, pero sí es clave poder diferenciar a cada uno de ellos fácilmente, entendiendo qué son y qué servicios podemos utilizar o crear para estos.

Es muy amplio el mercado de servicios de backend actual, y es muy importante nunca dejarnos estar, en relación a seguir adquiriendo conocimientos y seguir practicando, para estar listas a la primera oportunidad de inserción laboral que nos surja.



# Gracias por elegirnos para tu formación profesional

***"El aprendizaje constante y la pasión por la programación backend te llevarán a alcanzar nuevos horizontes en el emocionante mundo de la tecnología."***



# Muchas gracias.



Ministerio de Economía  
**Argentina**

Secretaría de  
Economía del Conocimiento

*primero  
la gente*