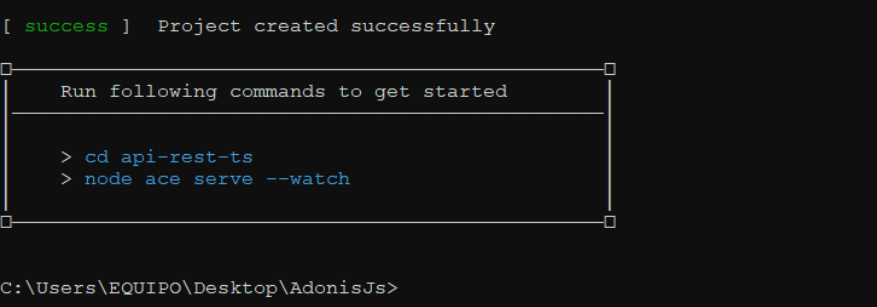
# Paso 5: INSTALACIÓN DE ORM Y DEPENDENCIAS PARA LA BASE DE DATOS.



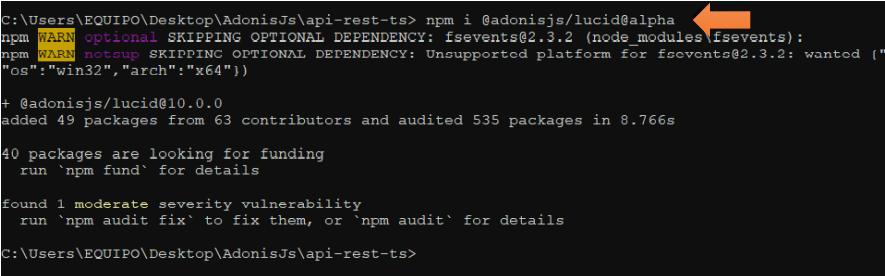
Como podemos observar, entramos a la carpeta usando los comando que nos imprime por la terminal.

**cd api-rest-ts**



Escribimos el siguiente comando para la instalación del ORM que vamos a usar en AdonisJS:

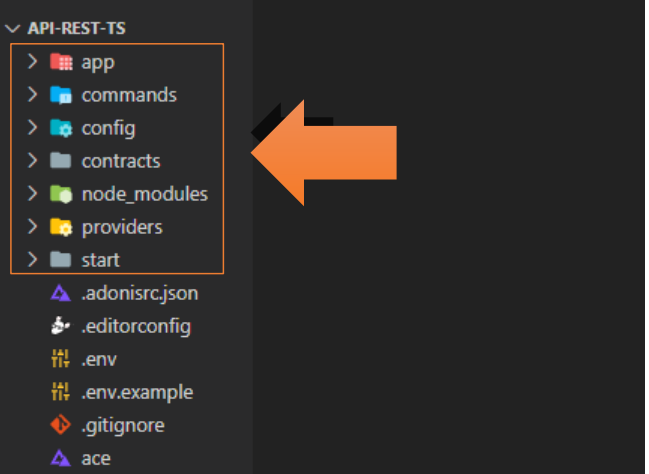
**npm i @adonisjs/lucid@alpha**



Instalación completada con éxito.

# Paso 6: PROGRAMACIÓN DE API REST

Para programar la api, vamos a usar Vs Code, pero también son libres de usar cualquier otro editor de código, ya sea Sublime Text 3, VIM, WebStorm, Atom.

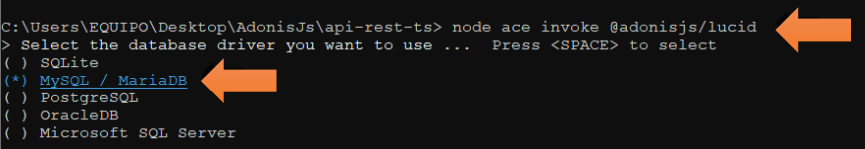


Como podemos observar, no tenemos la carpeta llamada database por lo que no podemos hacer migración a nuestra base de datos. Para eso tenemos que ingresar un comando para poder hacer migraciones, crear modelos etc.

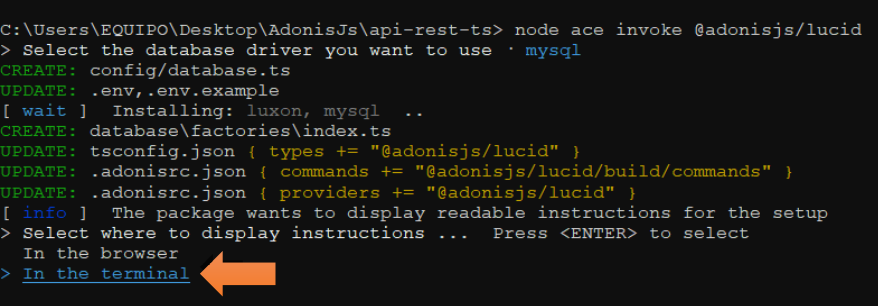
**node ace invoke @adonisjs/lucid**

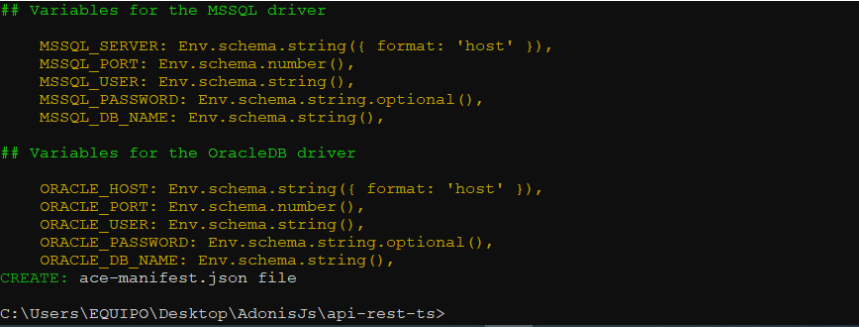
Escribimos el comando anterior y damos Enter, escogemos Mysql / MariaDB, pero si quieres usar otra base de datos también es válido, solo cambiarán los parámetros de configuración en el archivo .env.

Para seleccionar la base datos presionamos barra espaciadora y damos Enter.



A continuación, esperamos a que la instalación se termine de completar y damos Enter a la opción In the terminal para mostrar la información por la terminal.



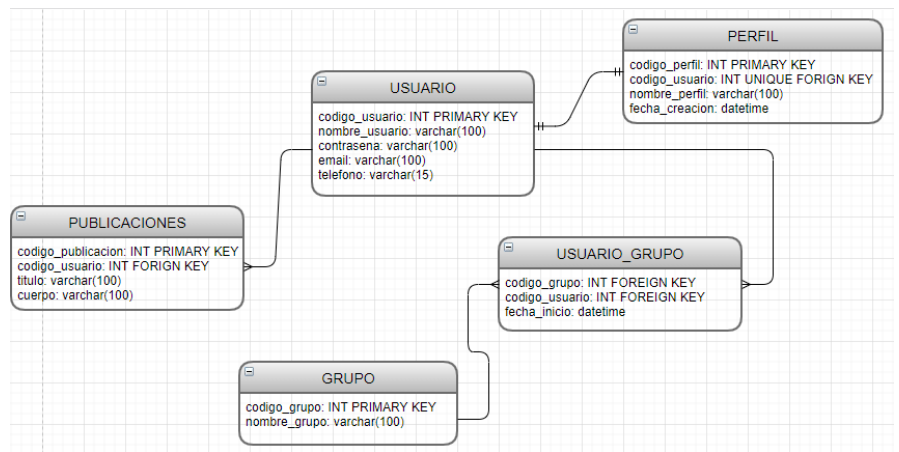


Ahora como podemos observar, si nos muestra la carpeta database.



# PASO 7. CREACIÓN DE MODELO RELACIONAL PARA MIGRAR A LA BASE DE DATOS

Para este ejemplo vamos a usar el siguiente modelo entidad relación:



En este modelo, vemos que un usuario tiene 1 perfil (1:1), un usuario tiene muchas publicaciones (1:N) y que un usuario pertenece a muchos grupos y un grupo puede tener muchos usuarios (N:M)

Para crear cada una de estas tablas vamos a la terminal y dentro del proyecto, escribimos:

node ace make:migration usuario

node ace make:migration perfil

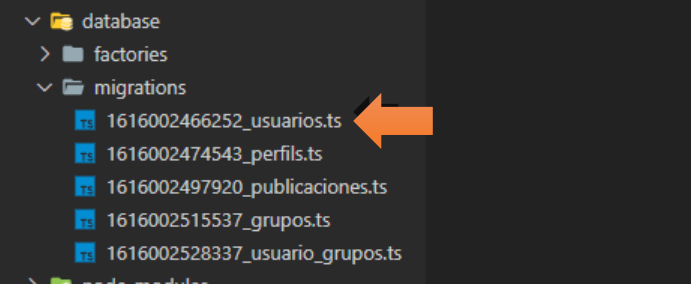
node ace make:migration publicaciones

node ace make:migration grupo

node ace make:migration usuario\_grupo

Una vez haya generado todas migraciones, nos dirigimos a la carpeta database y realizamos las relaciones e inserción de campos.

Una vez haya generado todas migraciones, nos dirigimos a la carpeta database y realizamos las relaciones e inserción de campos.



Abrimos la tabla **usuarios.ts** la cual debe aparecer de la siguiente forma:

import BaseSchema from '@ioc:Adonis/Lucid/Schema'

export default class Usuarios extends BaseSchema {

protected tableName = 'usuarios'

public async up () {

this.schema.createTable(this.tableName, (table) => {

table.increments('id')

table.timestamps(true)

})

}

public async down () {

this.schema.dropTable(this.tableName)

}

}

y lo modificamos de la siguiente forma:

import BaseSchema from '@ioc:Adonis/Lucid/Schema'

export default class Usuarios extends BaseSchema {

protected tableName = 'usuarios'

public async up () {

this.schema.createTable(this.tableName, (table) => {

table.integer('codigo\_usuario').primary().unsigned() // codigo\_usuario llave primaria

table.string('nombre\_usuario', 100).notNullable()

table.string('contrasena', 100).notNullable()

table.string('email', 100).notNullable()

table.string('telefono', 15).notNullable()

table.timestamps(false)

})

}

public async down () {

this.schema.dropTable(this.tableName)

}

}

Abrimos la tabla perfils.ts y hacemos la siguiente modificación:

import BaseSchema from '@ioc:Adonis/Lucid/Schema'

export default class Perfils extends BaseSchema {

protected tableName = 'perfils'

public async up () {

this.schema.createTable(this.tableName, (table) => {

table.integer('codigo\_perfil').primary().unsigned()

table.string('nombre\_perfil', 100).notNullable()

table.date('fecha\_creacion').notNullable()

table.integer('codigo\_usuario').unsigned().unique().index('codigo\_usuario')

table.foreign('codigo\_usuario').references('usuarios.codigo\_usuario').onDelete('cascade')

table.timestamps(false)

})

}

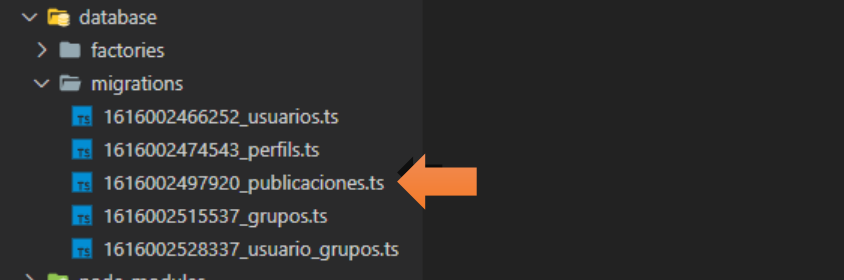
public async down () {

this.schema.dropTable(this.tableName)

}

}

Abrimos la tabla publicaciones.ts



Hacemos la siguiente modificación de la tabla:

import BaseSchema from '@ioc:Adonis/Lucid/Schema'

export default class Publicaciones extends BaseSchema {

protected tableName = 'publicaciones'

public async up () {

this.schema.createTable(this.tableName, (table) => {

table.integer('codigo\_publicacion').primary().unsigned()

table.string('titulo', 100).notNullable()

table.string('cuerpo', 100).notNullable()

table.integer('codigo\_usuario').unsigned().index('codigo\_usuario')

table.foreign('codigo\_usuario').references('usuarios.codigo\_usuario').onDelete('cascade')

table.timestamps(false)

})

}

public async down () {

this.schema.dropTable(this.tableName)

}

}

Abrimos la tabla grupos.ts y hacemos la siguiente modificación:

export default class Grupos extends BaseSchema {

protected tableName = 'grupos'

public async up () {

this.schema.createTable(this.tableName, (table) => {

table.integer('codigo\_grupo').primary().unsigned()

table.string('nombre\_grupo',100).notNullable()

table.timestamps(false)

})

}

Abrimos la tabla usuario\_grupos.ts y hacemos la siguiente modificación:

export default class UsuarioGrupos extends BaseSchema {

protected tableName = 'usuario\_grupos'

public async up () {

this.schema.createTable(this.tableName, (table) => {

table.integer('codigo\_usuario').unsigned().index('codigo\_usuario')

table.integer('codigo\_grupo').unsigned().index('codigo\_grupo')

table.date('fecha\_inicio').notNullable()

table.foreign('codigo\_usuario').references('usuarios.codigo\_usuario').onDelete('cascade')

table.foreign('codigo\_grupo').references('grupos.codigo\_grupo').onDelete('cascade')

table.timestamps(false)

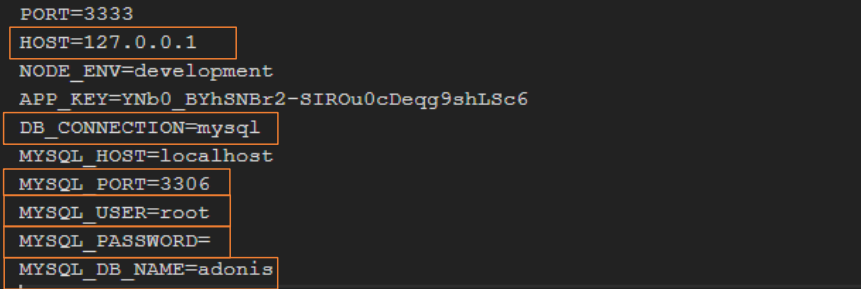
})

}

# Paso 8: MIGRAMOS NUESTRAS TABLAS AL MOTOR DE BASE DE DATOS.

Antes de migrar nuestras tablas a la base de datos, vamos a configurar nuestro archivo .env

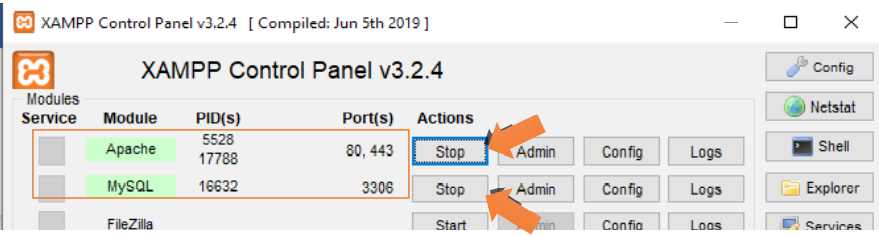
Debemos dejar configurado el archivo .env de la siguiente forma:



Ojo que la base de datos se debe llamar **adonis o nombrarla según la configuración**, Entonces activamos nuestro motor de base de datos local, para eso abrimos xampp y activamos Apache y Mysql, Si tienes otro motor de base de datos corriendo también es válido.

Si no tienen xampp instalado puede descargarlo de la siguiente página:

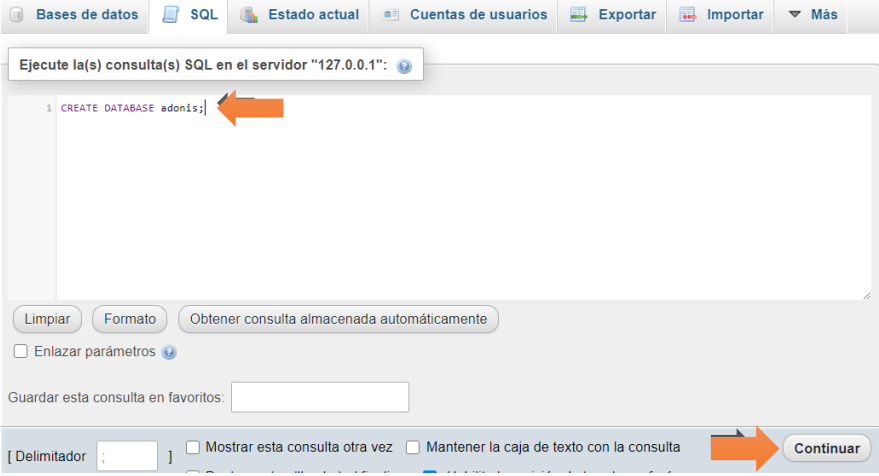
<https://www.apachefriends.org/es/download.html>



Ahora nos dirigimos a un navegador y escribimos:

<http://localhost/phpmyadmin/>

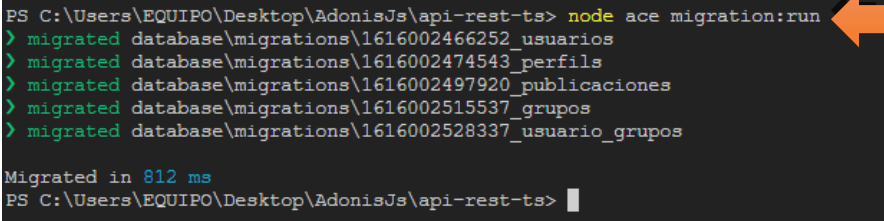




Una vez se haya creado la base de datos con éxito, procedemos a migrar nuestras tablas.

Para poder migrar nuestras tablas, escribimos el siguiente comando:

**node ace migration:run**



Resultado:



En caso de salir el error *ER\_NOT\_SUPPORTED\_AUTH\_MODE: Client does not support authentication protocol requested by server; consider upgrading MySQL client,* esto puede suceder con MySQL 8, hay que instalar por consola el paquete MySQL2, y hacer la referencia al llamado de ese paquete.

Para instalar ese paquete usamos: **npm i mysql2**

Luego debemos entrar por la carpeta config y abrir el archivo database.ts. En este archivo buscamos donde dice **client: mysql** y agregamos el 2 a mysql, quedando así: **client: mysql2**

# PASO 9: CREACIÓN DE MODELOS

Para la creación de nuestros modelos y poder usar el ORM Lucid de Adonis, vamos a escribir los siguientes comandos:

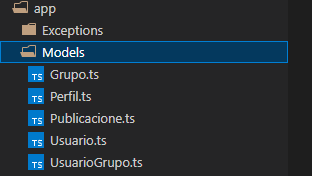
node ace make:model Usuario

node ace make:model Perfil

node ace make:model Publicaciones

node ace make:model Grupos

node ace make:model usuario\_grupos



# PASO 10: CREACIÓN DE LOS CONTROLADORES

Para la creación de los controladores escribimos los siguientes comandos:

node ace make:controller Usuario

node ace make:controller Perfil

node ace make:controller Publicaciones

node ace make:controller Grupos

node ace make:controller GrupoUsuario

A continuación escribimos la lógica de la API, OJO cada controlador contendrá un método para verificar que la llave primaria no se repita.