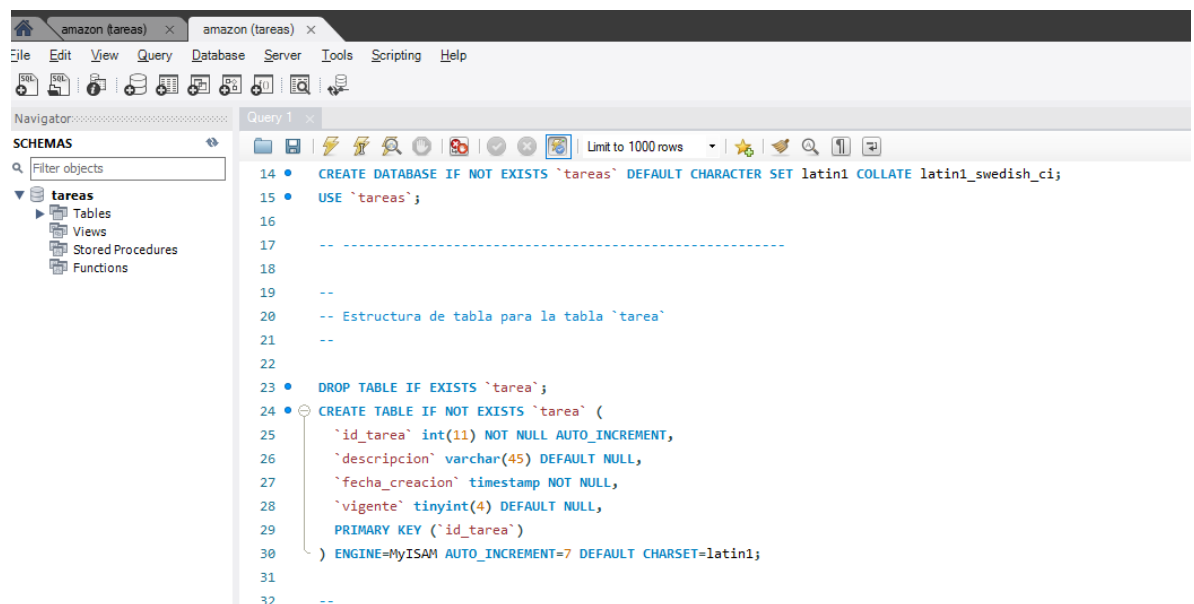


## Instructivo de despliegue

### 1. Configuración de la base de datos.

Lo primero que se debe realizar es instalar el motor de base de datos de mysql, luego con el programa de mysql workbench (u otro programa para manejar base de datos my sql), configurar el acceso motor de base de datos instalado, luego en explorador archivos vamos a la carpeta donde se encuentra este documento y buscamos el archivo “tareas.sql”, lo abrimos con block de notas y copiamos todo su contenido.

Regresamos al programa de gestión de base de datos de su preferencia buscamos la opción de ejecutar consultas y pegamos el código que se copió del archivo “tareas.sql”, luego ejecutamos el script, y con esto creamos la base de datos.



### 2. Configuración del Back.

#### 2.1. Descargamos el código fuente.

Vamos a la ubicación donde queremos descargar el código fuente, con git instalado y desde una terminal ejecutamos el siguiente comando.

git clone <https://github.com/cristiansrc/tareas.git>

#### 2.2. Configuración del origen.

Con netbeans abrimos el proyecto y buscamos el paquete:

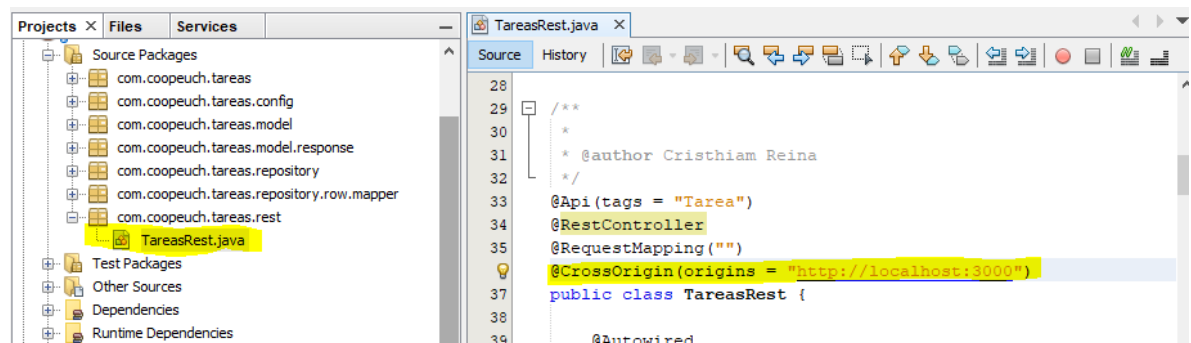
com.coopeuch.tareas.rest

Y abrimos el archivo TareasRest.java, ahí vamos a la línea 36:

```
@CrossOrigin(origins = "http://ec2-3-139-71-41.us-east  
2.compute.amazonaws.com:3000")
```

En esa línea cambiamos “http://ec2-3-139-71-41.us-east 2.compute.amazonaws.com” por la ip o el host del servidor donde va a quedar desplegado el front de la aplicación, por ejemplo, puede ser localhost, quedaría de la siguiente forma:

```
@CrossOrigin(origins = "http://localhost:3000")
```



### 2.3. Configuración de la base de datos.

En el proyecto dentro de netbeans buscamos la carpeta Other Sources y desplegamos su contenido, luego desplegamos el contenido del paquete <default.properties> y abrimos el archivo “application.properties”.

Ahí tenemos que configurar 3 propiedades “spring.datasource.url”, “spring.datasource.username” y “spring.datasource.password”.

En la propiedad “spring.datasource.url” se debe cambiar “database-1.cexxr4cyzpox.us-east-2.rds.amazonaws.com” por la ip o el host donde tenemos la base de datos, por ejemplo, localhost, el resto de la cadena debe quedar igual, la propiedad debe quedar de la siguiente forma:

spring.datasource.url=jdbc:mysql://localhost/tareas?useUnicode=true&useJDBCCompliantTimezoneShift=true&useLegacyDatetimeCode=false&serverTimezone=UTC

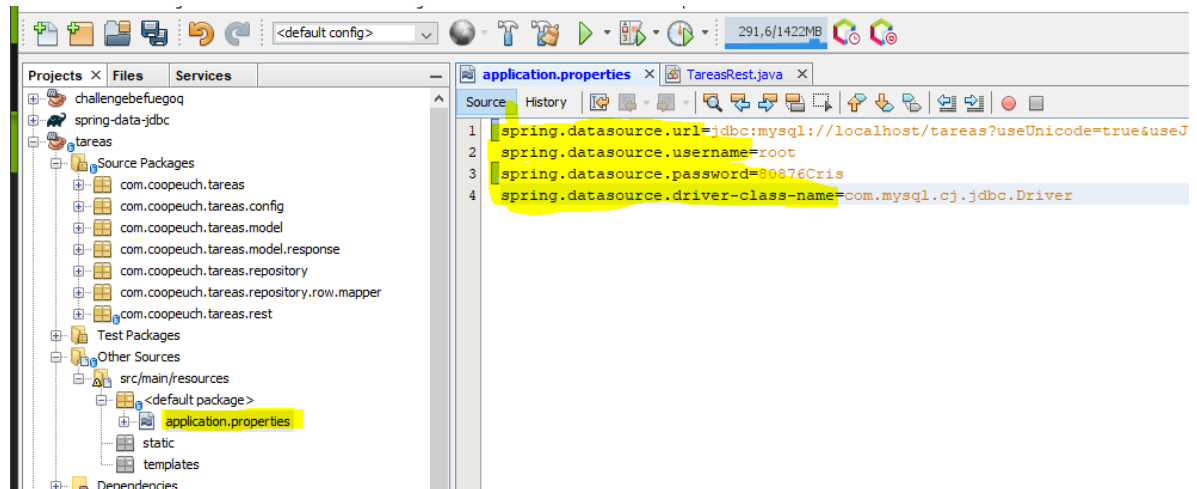
Para la propiedad “spring.datasource.username” tenemos que dejar el usuario de la base de datos que se encuentra configurado en el motor de base de datos para acceder a la base de datos de tareas, la propiedad quedaría de la siguiente forma:

spring.datasource.username=root

Por ultimo configuramos la propiedad “spring.datasource.password” y ahí configuramos la contraseña de la base de datos, quedaria de la siguiente forma:

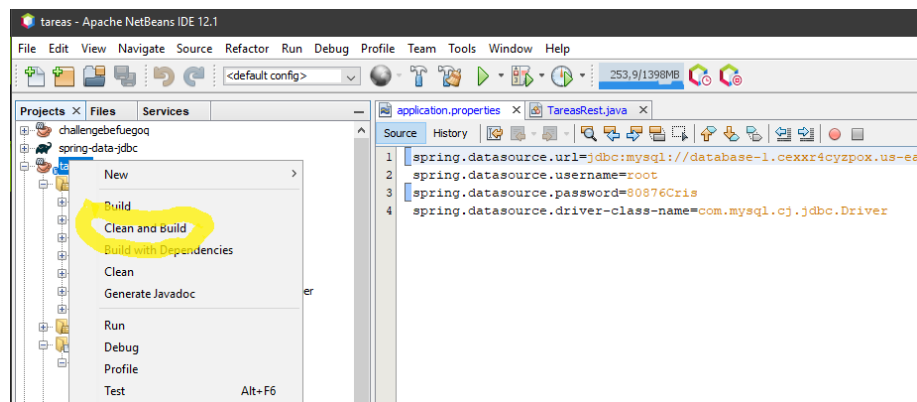
spring.datasource.password=80876Cris

La propiedad “spring.datasource.driver-class-name” se debe dejar tal cual como se descarga del repositorio.



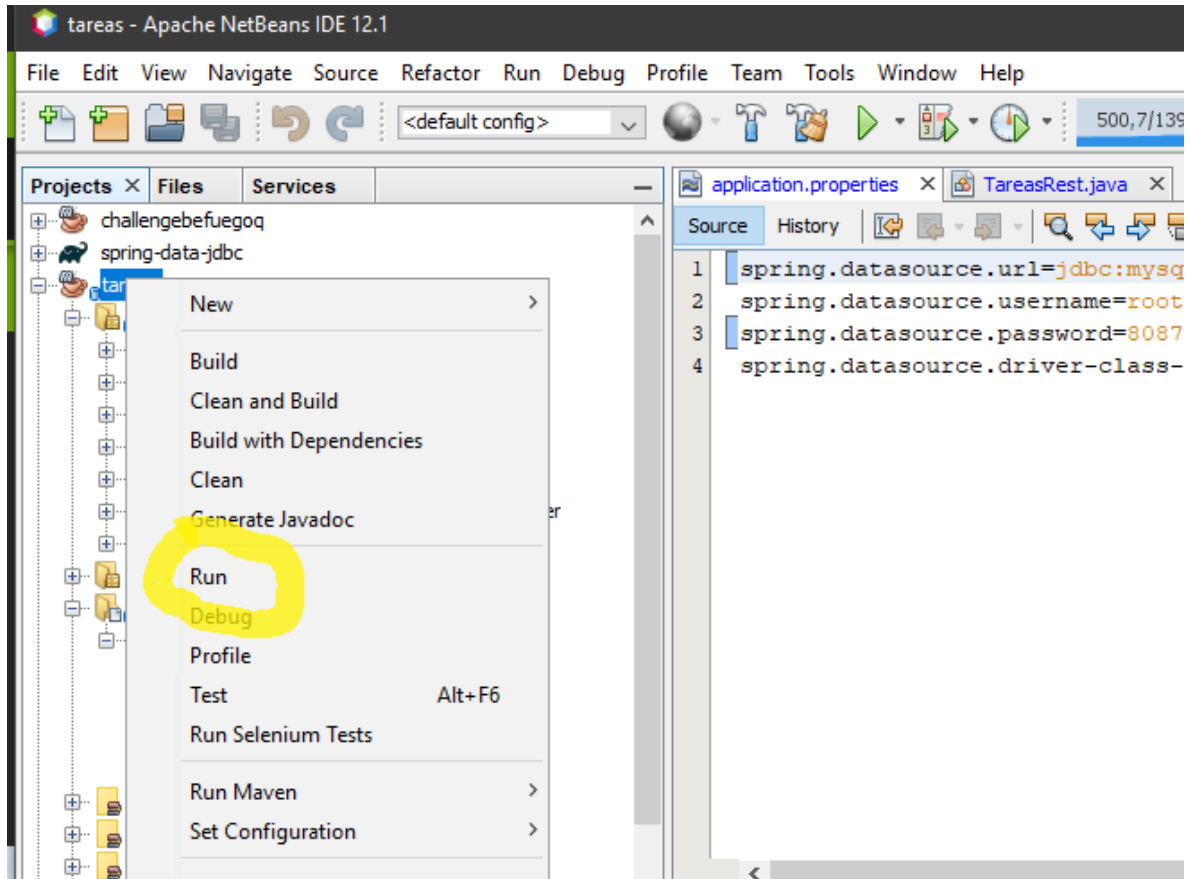
## 2.4. Compilación de la aplicación.

Luego compilamos la aplicación, damos clic derecho sobre el proyecto y damos “clean and build”



## 2.5. Despliegue de la aplicación desde netbeans.

Si se desea desplegar la aplicación en servidor, pasar al paso 2.6, para desplegar el proyecto solo se debe dar clic derecho sobre el proyecto en netbeans y luego run.



## 2.6. Despliegue de la aplicación sin ide.

Para desplegar en una maquina distinta, se debe ingresar al proyecto desde el buscador de archivos del sistema operativo, entrar al proyecto y luego ingresar a la carpeta target, buscamos el archivo tareas-0.0.1-SNAPSHOT.jar, y luego lo enviamos a la maquina y la ubicación deseada, luego desde una terminal acceder a la ubicación y luego colocar el comando:

```
Java -jar tareas-0.0.1-SNAPSHOT.jar
```

## 3. Configuración del front.

### 3.1. Descargamos el código fuente.

Vamos a la ubicación donde se desea descargar el código fuente de la aplicación y git instalado, desde una terminal ejecutamos el siguiente comando:

git clone <https://github.com/cristiansrc/tareas-front.git>

### 3.2. Configuramos la url de los servicios.

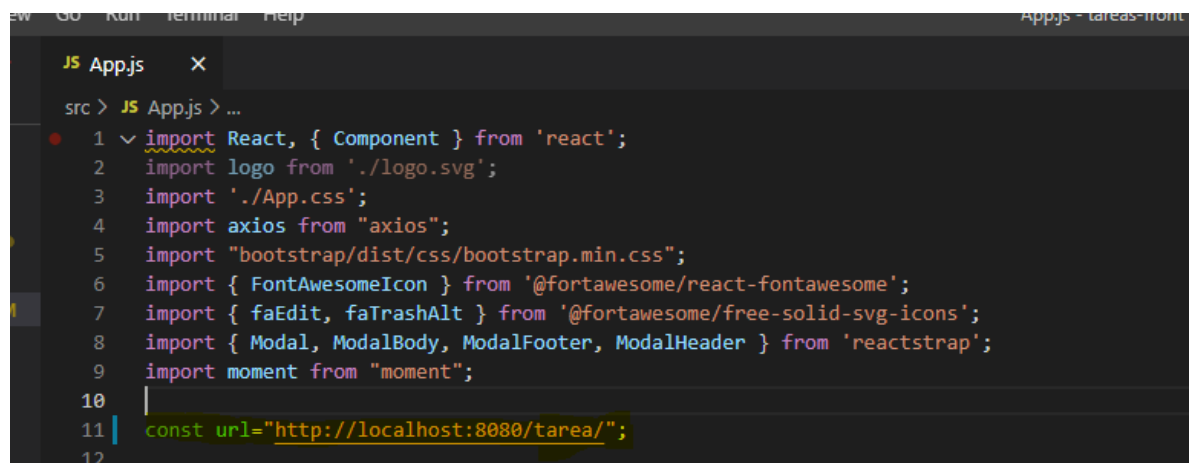
Entramos a la carpeta tareas-front y buscamos el archivo App.js y lo abrimos con el bloc de notas y buscamos la línea numero 11 que contiene lo siguiente:

```
const url="http://ec2-3-139-71-41.us-east-2.compute.amazonaws.com:8080/tarea/";
```

Aqui vamos a cambiar "http://ec2-3-139-71-41.us-east-2.compute.amazonaws.com" por la ip o el host del servidor donde vamos a dejar los servicios de la aplicación, por ejemplo, si los vamos a dejar en una maquina local, podemos dejar "localhost" y quedaría de la siguiente forma:

```
const url="http://localhost:8080/tarea/";
```

Y por último guardamos el archivo.

A screenshot of a code editor window titled 'App.js'. The editor shows a list of imports at the top of the file, including React, axios, bootstrap, FontAwesomeIcon, and others. Line 11 is highlighted, showing the updated URL: `const url="http://localhost:8080/tarea/";`. The line number 11 is visible on the left margin. The editor has a dark theme and a file explorer on the left side.

### 3.3. Despliegue.

Desde una terminal vamos a la ubicación de la carpeta de tareas-front y con nodejs instalado ejecutamos el comando npm install.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\crist\Documents\Procesos de seleccion\Apiux\Codigo fuente\tareas-front> npm install
npm WARN bootstrap@4.5.3 requires a peer of jquery@1.9.1 - 3 but none is installed. You must install a
npm WARN create-react-context@0.3.0 requires a peer of react@^0.14.0 || ^15.0.0 || ^16.0.0 but none is
```

En la misma terminal ejecutamos el comando `npm start`.

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\crist\Documents\Procesos de seleccion\Apiux\Codigo fuente\tareas-front> npm start
```

Luego de ejecutar el comando debe aparecer lo siguiente

```

src\App.js
Line 2:8:      'logo' is defined but never used      no-unused-vars
Line 34:5:    Do not mutate state directly. Use setState()  react/no-direct-mutation-state
Line 35:5:    Do not mutate state directly. Use setState()  react/no-direct-mutation-state
Line 84:67:   Expected '===' and instead saw '=='            egegeg
Line 97:5:    Do not mutate state directly. Use setState()  react/no-direct-mutation-state
Line 98:5:    Do not mutate state directly. Use setState()  react/no-direct-mutation-state
Line 98:57:   Expected '===' and instead saw '=='            egegeg
Line 116:5:   Do not mutate state directly. Use setState()  react/no-direct-mutation-state
Line 116:57:  Expected '===' and instead saw '=='            egegeg
Line 196:34:  Expected '===' and instead saw '=='            egegeg

```

Search for the [keywords](#) to learn more about each warning.

To ignore, add `// eslint-disable-next-line` to the line before.

Con esto se despliega la aplicación.

←

→

↺

🏠

⚠️ No seguro

ec2-3-139-71-41.us-east-2.compute.a...

🔍

☆

🔥

⋮

🗑️

👤

⋮

Agregar Tarea

ID	Descripcion	Fecha de creacion	Vigente	Acciones
7	prueba	2018-02-12 4:27 am	Si	<div>✎</div> <div>🗑️</div>
4	prueba	2018-02-12 11:59 pm	Si	<div>✎</div> <div>🗑️</div>

