

<https://github.com/ProgramacionFuncionalReactiva/Talleres>

Para realizar el taller de esta semana siga los siguientes pasos:

1. Genere el archivo CSV para los siguientes datos de estudiantes:

nombre,edad,calificacion,genero

Andrés,10,20,M

Ana,11,19,F

Luis,9,18,M

Cecilia,9,18,F

Katy,11,15,F

Jorge,8,17,M

Rosario,11,18,F

Nieves,10,20,F

Pablo,9,19,M

Daniel,10,20,M

2. Genere la tabla en MYSQL para esta información.

3. Elabore un programa que inyecte los datos del archivo CSV a la base de datos.

4. En el mismo programa agregue la funcionalidad para obtener de la base de datos todos los registros de Estudiantes.

5. Documente en un repositorio de github (archivo README.md) y suba el código en el mismo repositorio.

6. Comparta el enlace del repositorio.

<https://github.com/PFR-Computacion-O25F26/b2-tc-eda-estadistica-descriptiva-b-evelynlizzethe>

<https://slides.com/lisantiago/b2s14-base-de-datos>

<https://github.com/ProgramacionFuncionalReactiva/Talleres>

```
mysql -h db -u root -ppassword pfr_db
```

Como ejecuto este programa aqui en codspace?

```
package bim2.semana13.lectura

import doobie._
import doobie.implicits._
import cats.effect._

// Usar IOApp.Simple es la forma moderna en Scala 3 / Cats Effect 3
object EjemploBase extends IOApp.Simple {

  case class Movie(id: Int, originalTitle: String)

  // Marcado como private para quitar la alerta "Access can be private"
  private val xa = Transactor.fromDriverManager[IO](
    driver = "com.mysql.cj.jdbc.Driver",
    url = "jdbc:mysql://localhost:3306/my_db",
    user = "root",
    password = "integrador",
    logHandler = None
  )

  private def find(id: Int): ConnectionIO[Option[Movie]] =
    sql"SELECT ID, ORIGINAL_TITLE FROM MOVIE WHERE ID = $id".query[Movie].option

  private def listAllMovies(): ConnectionIO[List[Movie]] =
    sql"SELECT ID, ORIGINAL_TITLE FROM MOVIE".query[Movie].to[List]

  // El método run reemplaza al cuerpo del "App" tradicional
  override def run: IO[Unit] = {
    for {
      // Usamos for-comprehension para manejar los efectos de forma segura
      res    <- find(4).transact(xa)
      _      <- IO.println(s"Película encontrada: ${res.getOrElse("No existe")}")

      movies <- listAllMovies().transact(xa)
      _      <- IO.println("Lista de películas:")
      _      <- IO(movies.foreach(println))

      // Aquí podrías añadir el insert de la misma forma
    } yield ()
  }
}
```

Ya tengo la tabla MOVIES en la base de datos pfr_db que se detalla en el README.md