Explanation of classes

The system classes, for which there is a class diagram in a separate document (**Class Diagram.pdf**), are as follows:

Class	Explanation
BigDataTFG	It is the main class of the system, which acts as the Interface (JFrame) of the system. The class contains elements of the ResourceBundle type (in particular, the one called “ rb ”) that are used on the system's initial screen to choose the language, whose dialogue is coded on line 57 [1]. rb.getString (...) method takes the name of a label and returns its value in the selected language (example, line 92). Some labels are numeric or alphanumeric and are not translated. Then all the buttons, combos, etc. are loaded onto the main screen. jButton1ActionPerformed -> method that downloads CSV files from the open internet repository jButton2ActionPerformed -> method to decompress previously downloaded CSV files jButton3ActionPerformed -> method to load the previously decompressed file into MongoDB jButton6ActionPerformed-> method that reads the selected radio button at the bottom (data mining) and takes care of doing what is appropriate (clean, graph or create the .arff files) jButton7ActionPerformed->method that opens the created .arff file and , depending on the selected action, executes the corresponding data mining algorithm jButton8ActionPerformed->method that opens a web browser window and displays the location of the selected station jComboBox2ActionPerformed-> filter by country jComboBox3ActionPerformed -> filter by station jButton9ActionPerformed, jButton10ActionPerformed and jButton11ActionPerformed -> methods to create the

[1]Whenever a line number is indicated, it will be approximate since the version may vary.

	database at the region, country or station level jButton12ActionPerformed -> method to load annual data into an existing database jButton4ActionPerformed -> method that makes specific fields visible when choosing a data mining technique (bottom right) jButton5ActionPerformed->method that saves the selected station to build the classification model (only with stations) jButton13ActionPerformed-> method that executes the sorting task getPgrAvance -> progress bar when downloading getPgrAvance1-> progress bar when decompressing getPgrAvance2 -> progress bar when importing to MongoDB main method that starts everything and launches the interface.
Classify	Class responsible for performing a classification (prediction) of a variable for a station for which there was not much historical data, training the model with data from other stations. previously created .arff view file is read with the data from the selected stations (with sufficient data), then an intermediate testing file that Weka needs is created (something transparent to the user), and finally the classification algorithm is executed (invoking methods from a Weka package), indicating beforehand what the class attribute is.
Final Collection N	This class loads user-specified data for a specific year (using `jButton12ActionPerformed` in the main class) into a database for a region, country, or station (previously created by the user). The data is saved already aggregated by month.
QueryGraph	Class to display a specific graph with information about a variable (minimum temperature , maximum temperature or precipitation), within a database and for a time range.
Where to query	Class that is responsible for querying a specific collection in a database (e.g., the year 2026 in Jordan) to view a specific field within it (e.g., rainfall). The method of this class is overridden to allow different configurations regarding the number of parameters.
CreateMongoDB	Class that creates a database (it only creates the

	database with its corresponding name and some empty collections) of region, country or season based on the user's selection (jButton9ActionPerformed, jButton10ActionPerformed and jButton11ActionPerformed buttons in the main class)
Decompressor	Class that decompresses the specified file
Download File	This class downloads files from the website. It has a method of the same name that receives the path and URL and downloads the file. The URL is constructed in the main class, and this method is called from there using the URL already constructed with the corresponding year.
Arff File	.arff file that will be the minable view . fileArff method creates the minable view for time series analysis. fileArffClass method creates the minable view for the classification part. ArffTest file method creates a test file, a sort of trick, to tell Weka what you want to predict.
FormatDate	Class that puts all the dates of a collection into the same unified format (ISODate) so that the process of aggregating by months in the ColeccionFinalN class is easier .
ReadMongoRunnable	This class manages the CSV import progress bar to MongoDB (called “getPgrAvance2” in the main class). While the bar isn't perfect, it provides a good approximation of the current state of each upload.
MongoDB	It's a class that handles saving CSV files to MongoDB . It has a single method called `MongoDB` that takes a CSV file, a database, and a collection, and stores the CSV file in that collection within the database. The collection will always have the following field structure: ID_estacion,fecha,elemento,valor_dato,M_flag,q_flag,s_flag,observ_time It is important to clarify that this method invokes the mongoimport command and you must correctly specify the path of the file system where it is located (e.g., “C:\\ Program Files\\ MongoDB \\Server\\4.2\\ bin \\”
TimeSeriesTFG	Class responsible for performing time series data mining, that is, predicting the values of a variable for a specific year based on the values of previous years. This class reads the previously created .arff view file , which contains data from a specific database used to train the model, and executes the specified Weka method on the indicated variable (minimum temperature, maximum temperature, or precipitation). There are two types of analysis, also selectable by the user: univariate and multivariate .
ViewDatabases	MongoDB databases created by the system user.

ViewCollections	Class responsible for storing all collections from the database “ tfg ” in a Combo box (“ tfg ” is a database created in MongoDB to manage this system).
See Table	MongoDB database and collection and putting the results into a Combo box in the interface.
English	File containing the interface label values in English
Spanish	File containing the interface label values in Spanish