****Javier Martín Aizpuru

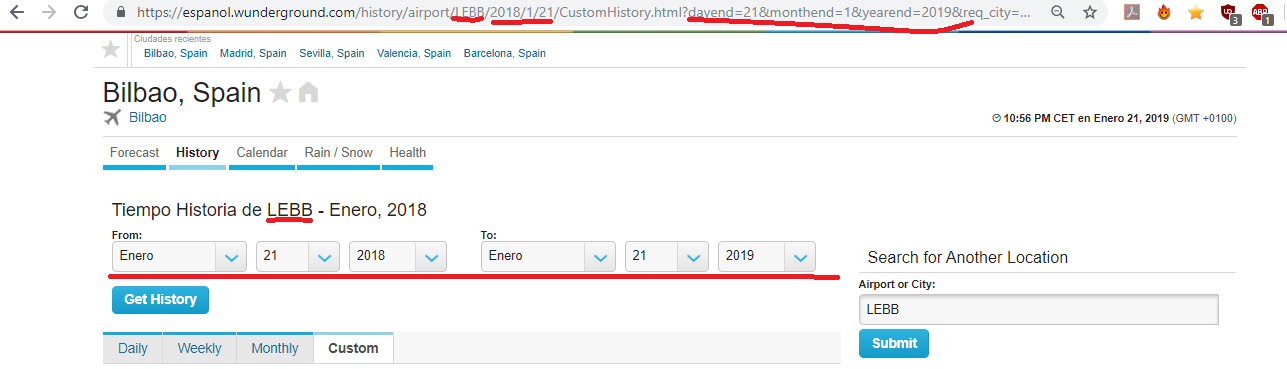
79076369T

**Informe de Cierre de Proyecto: “El tiempo”**

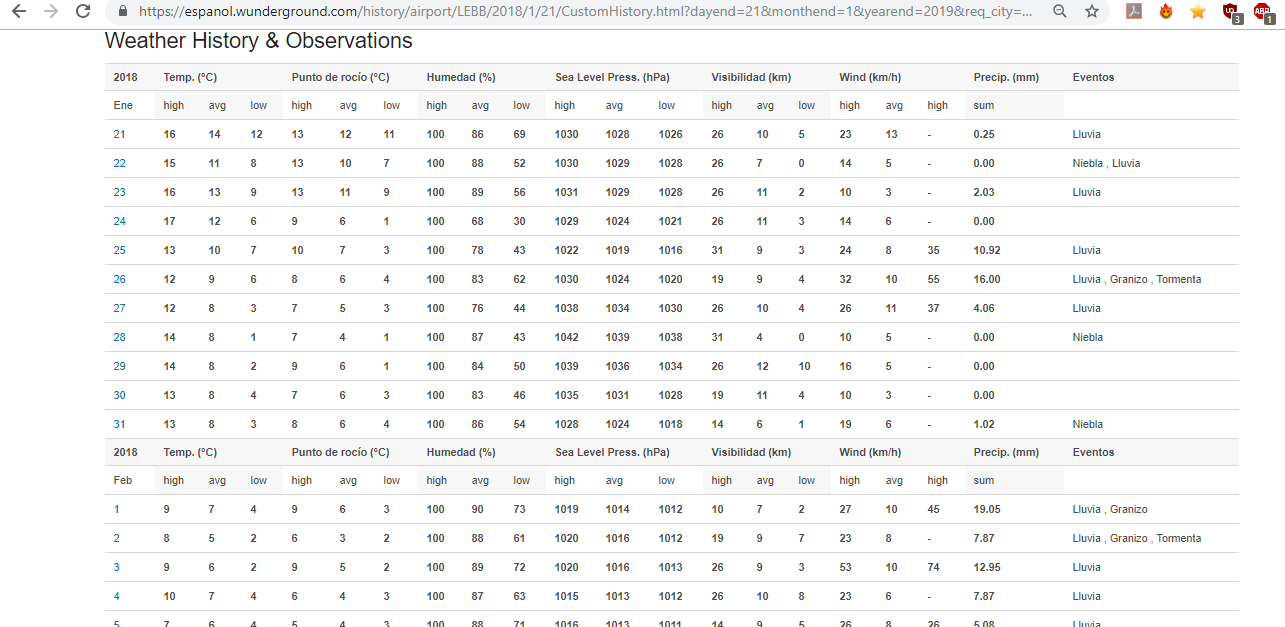
**Dedicación de tiempo al proyecto**

* Proceso de búsqueda de página web (*Weather Underground*) que ofrezca base de datos meteorológicos de distintas ciudades y permita el scrapping accesible para ello. Aproximadamente **3-4 horas** tras descartar otras opciones y localizar el método de acceso a los datos del apartado *Custom History* modificando la URL
* Inicio de proyecto en código java, construcción del método personalizado para el web scrapping personalizado por código de estaciones y el almacenamiento en estructuras de datos simples. Aproximadamente **5 horas** entre bibliografía, vídeos y el código.
* Creación de la Base de Datos *sqlite*, implementación de métodos de almacenamiento, actualización, reinicio y selección de los datos provenientes del scrapping. Aproximadamente **7 horas** hasta el funcionamiento idóneo de la base de datos y sus métodos
* Desarrollo de la interfaz gráfica y de la carga de datos desde la BBDD a una JTable. Ventanas de búsqueda, ventanas de inicio y ventanas de resultado. Alrededor de **6 horas** para la primera versión.
* Añadido de funciones básicas sobre los datos de búsqueda recogidos en la JTable. **2 horas**
* Interfaz gráfica: Nuevas ventanas de ayuda para los códigos, y la ventana de predicción. Por otro lado, un algoritmo simple de predicción para el día siguiente al actual en base a 4 parámetros. **6 horas**

**Descripción por fases del proyecto**

****

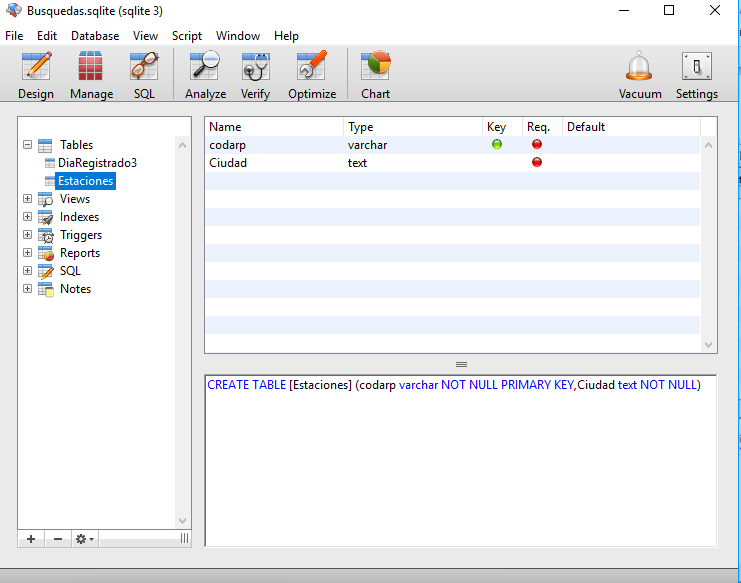
Página desde la que se parte para el algoritmo de scrapping que permite la manipulación de la URL, como se indica en la imagen de arriba en base a los parámetros de **fechas iniciales y finales y código de estación.**



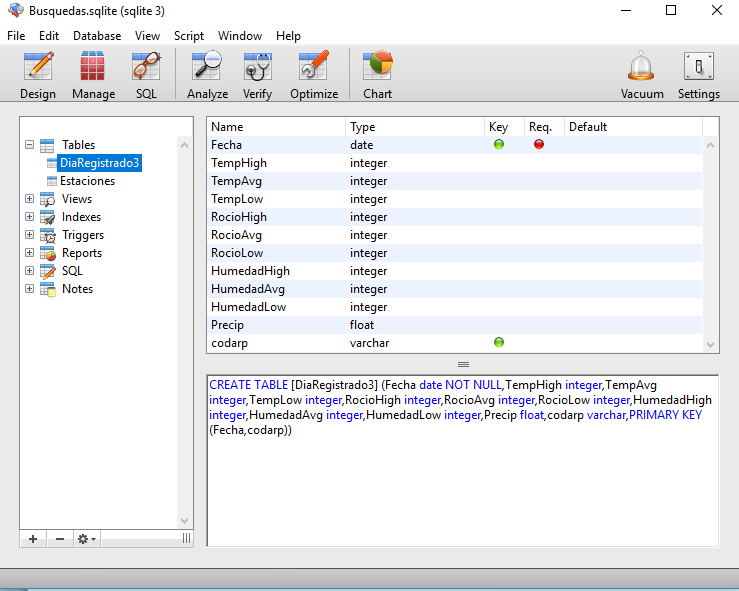
Los datos se devuelven en este formato de tabla, para lo cual hay que proceder a leer cada fila, descartando las que no interesen (títulos de cada mes, años..) y almacenando el resto de datos en la Base de Datos. Para ello se ha utilizado la librería **JSoup** y la extensión para Chrome **uBlock Origin** que permite localizar de manera más fácil los elementos en la inspección de web.

**Bases de datos**

La estructura de las tablas es la siguiente:

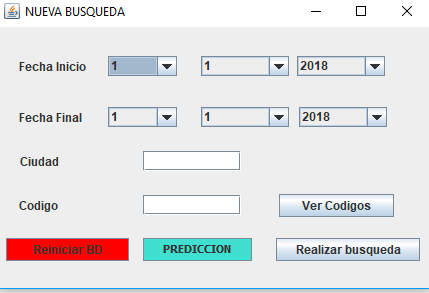


Cómo se aprecia en las imágenes se ha utilizado **SQLiteManager** para la creación de la base de datos, la visualización de las operaciones a tiempo real. Las tablas Estaciones y DiaRegistrado están relacionados por el código de estación (*codarp*) que actúa como clave externa o *foreign key* en la tabla de DiaRegistrado. Se usó la librería **sqlite-jdbc**



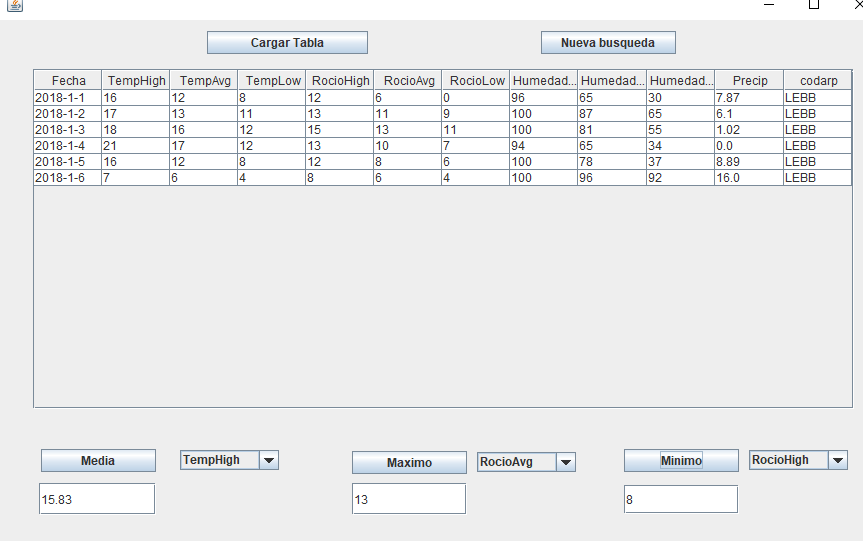
**Interfaz gráfica**

* **Ventana principa**l (*VentanaInicio*): Permite las funciones de búsqueda de nuevos datos, acceso a la ventana y función de predicción o al recurso de ayuda sobre los códigos. También se da la opción de reinicio de BBDD para eliminar las anteriores búsquedas

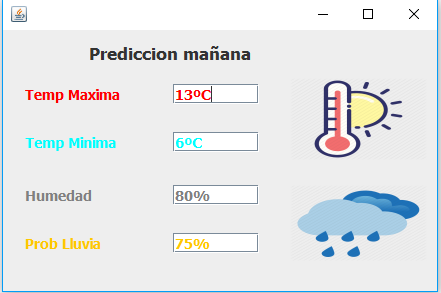


* **Ventana de resultados** (*VentanaTable*): Muestra los resultados en una JTable al presionar un botón, y en la parte inferior permite realizar varias funciones inmediatas sobre los datos buscados hasta que el usuario quiera realizar una nueva búsqueda volviendo al menú principal.

\* Para la carga de la JTable se ha utilizado la librería **r2xml**



* **Ventana de predicción** (*VentanaPrediccion*): Muestra la predicción para el día siguiente en base a los parámetros de temperatura máxima, temperatura mínima, humedad y probabilidad de lluvia.



* **Algoritmo de predicción**: Algoritmo muy básico que toma los resultados del mismo mes de la fecha actual pero del **año pasado** y calcula las medias de las **temperaturas máximas, mínimas, humedad** y calcula la **probabilidad de lluvia** en función del número de días que llovió en aquel mes. (Hubiera sido interesante perfeccionarlo más pero la planificación de tiempo no fue la indicada para ello)

***Librerías y recursos utilizados***

Aunque tanto las librerías, como las imágenes, bases de datos y demás recursos están subidos en **Github**, a continuación se muestran los links de descarga de los archivos **.jar** de las librerías.

* **Sqlite-jdbc** <https://bitbucket.org/xerial/sqlite-jdbc/downloads/>
* **R2xml**  <https://sourceforge.net/projects/finalangelsanddemons/>
* **JSoup** <https://jsoup.org/download>

**Valoración general sobre el desarrollo del proyecto**

Considerando por un lado que el proyecto mantiene algo de la idea inicial de predicción de datos del tiempo, creo que gran parte del proyecto se ha quedado en la parte de toma de datos, almacenamiento y el asunto de la manipulación de estos datos en cuanto a algoritmos complejos de *Machine Learning* apenas aparece.

Sin embargo, teniendo en cuenta que, al haberse reducido de 2 a 1 integrante y no haber llevado a cabo la planificación adecuada (quizás por sobreestimación de tiempo disponible para el proyecto) creo que el resultado el proyecto es interesante para el nivel de ProgIII y contiene lo esencial: interfaz gráfica, idea de predicción, toma de datos, base de datos y la idea original de la toma de datos para su posterior manipulación.

Por otro lado, este proyecto resulta interesante para hacer de puente al Machine Learning habiendo tenido ya experiencias sobre todo con la toma, limpieza y almacenamiento de datos para objetivos o proyectos futuros relacionados con el tema.

En cuanto al proceso de trabajo, me ha parecido muy interesante y hasta emocionante, el poder desarrollar una idea desde su origen e ir paso a paso, resultando en una experiencia agradable y muy útil para el futuro.