“Festivos anuales provinciales de España”

Descripción

El conjunto de datos resultante tras aplicar la ejecución del *script* conforma una serie de características correspondientes con las fiestas anuales nacionales, autonómicas y locales de las distintas provincias de España.

Representación gráfica

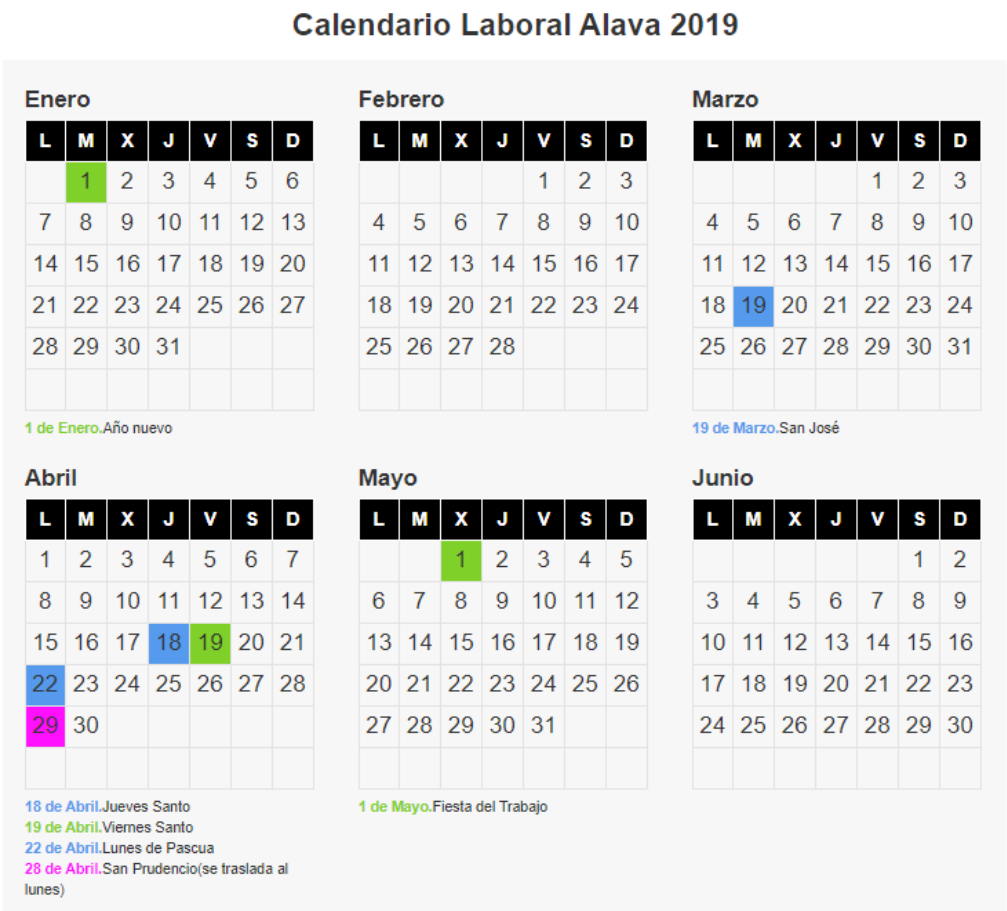

Figura 1: Calendario laboral correspondiente a los meses de enero a junio de 2019, en Álava.

Contenido

Cada registro del conjunto de datos se corresponde con un día festivo que cuenta con los siguientes atributos:

- **Name**: el nombre en cuestión del día festivo.
- **Date**: fecha en la que tiene lugar la festividad.
- **Province**: provincia en la que se lleva a cabo el día festivo.
- **Type**: el tipo de fiesta a la que se refiere, Nacional, Autonómica o Local.
- **Moved Date From**: día asignado a la festividad, previo a su modificación. Gran parte de los registros no contendrán valor para este atributo.

Los autores de la web *CalendariosLaborales*, *Happy Publi S.L.,* han recolectado información acerca de los días festivos desde el año 2006 hasta el año actual, 2019.

Para obtener los datos del dataset resultante se ha hecho uso del lenguaje de programación Python, junto con la librería *Selenium,* además de técnicas de *Web Scraping* para obtener la información contenida en las páginas HTML.

Contexto

El contexto del dataset se corresponde con los festivos anuales, como se ha comentado previamente en la descripción, nacionales, autonómicos y locales, de las distintas provincias de España desde el año 2006 hasta el 2019.

Agradecimientos

Los datos han sido recolectados desde la web online de *CalendariosLaborales* de manera pública y gratuita.

Inspiración

Los datos obtenidos podrían utilizarse de manera ampliamente diversa, como, por ejemplo, con la finalidad de predecir los futuros días festivos y automatizar los procesos de asignación de los propios días. Así mismo, cabe destacar que es un conjunto de datos atípico y original.

Otro tipo de utilización podría ser un *recomendador* de días semanales. Por ejemplo, se tomemos como suposición que se escoge una actividad de ocio, tal como asistir a clases de música. Los días festivos, no se acude a dicha clase,

pero, sin embargo, si se abona el importe correspondiente. De esta manera, el *recomendador* nos indicaría, dada una fecha inicial, qué días semanales podrían resultar más adecuados para tomar las clases, con la finalidad de *perder* el menor número de clases posible.

Licencia

Debido a que el propio aviso legal de la web indica que queda terminantemente prohibida la reproducción de elementos o contenidos realizados con ánimo de lucro o fines comerciales, los datos resultantes estarán bajo la licencia de **CC BY-NC-SA 4.0**, con la finalidad de evitar posibles problemas penales e incumplimientos de la ley.

Código fuente y dataset

Tanto el código utilizado para la obtención de los datos, como los propios datos obtenidos, están disponibles dentro del repositorio de *Github*:

https://github.com/AlejandroSuau/TDLC_Practice1

Técnicas y prevenciones para evitar bloqueos

Con la finalidad de evitar ser bloqueados y hacer un uso ético y correcto del *web scraping*, hay una serie de técnicas que podemos aplicar. Algunas de ellas han podido ser desarrolladas en esta práctica, tales como, el hecho de realizar una espera aleatoria de N segundos entre cada petición a una url distinta para no sobrecargar el servidor de peticiones o la utilización de un user-agent en la cabecera, en este caso, por parte de la librería *selenium*. Sin embargo, otras de las técnicas, como el uso de un servidor proxy que varíe las IPs, la utilización de diversos user-agent, o la modificación de parámetros en la cabecera, como el *http-referer*, no han podido ser implementadas.

Recursos

1. Subirats, L., Calvo, M. (2018). Web Scraping. Editorial UOC.

2. Masip, D. El lenguaje Python. Editorial UOC.

3. Lawson, R. (2015). Web Scraping with Python. Packt Publishing Ltd. Chapter2.Scraping the Data.

4. Simon Munzert, Christian Rubba, Peter Meißner, Dominic Nyhuis. (2015). Automated Data Collection with R: A Practical Guide to Web Scraping and Text Mining. John Wiley & Sons.