Apéndice A

Manuales

A.1. Manual de despliegue e instalación

Este proyecto trabaja con herramientas externas, por lo que el usuario necesitará instalar en su máquina las aplicaciones `SemanticMerge` y `gMaster` previamente a los siguientes pasos de la instalación del proyecto. Para la utilización del primer software, se debe adquirir o solicitar una clave de acceso, o bien, utilizar la prueba gratuita de 30 días. Tras ello, deberá descargarse los dos archivos ejecutables correspondientes a los formateadores de texto así como el archivo `JAR` correspondiente al *parser* externo. El *link* del repositorio puede encontrarse en el **Apéndice B**. Los archivos necesarios para descargar se encuentran en la carpeta `Executables`.

Según donde hayamos instalado dichas aplicaciones, deberemos dirigirnos al directorio `/gmaster`, el cual si hemos utilizado la ruta predeterminada se encontrará en la ruta `C:/Users/{User}/AppData/Local/`. Una vez en dicho directorio, debemos acceder al subdirectorio `/config` y crear un archivo con nombre `externalparsers.conf`. Este archivo tiene la función de enlazar formatos no detectados con sus correspondientes *parsers* externos para `SemanticMerge`. Para mayor comodidad, se recomienda guardar el archivo `JAR` descargado en la carpeta asociada a `SemanticMerge`, la cual se encuentra en la misma ruta y al mismo nivel que la asociada a `gMaster`, de nuevo, si hemos utilizado la ruta predeterminada.

En nuestro caso, debemos asociar la extensión `.mdl` a la ruta absoluta del *parser*. Concretamente, el contenido del fichero será similar al siguiente bloque de texto:

```
.mdl=java -jar C:\Users\Propietario\AppData\Local\semanticmerge\
mvntfg-1.0-jar-with-dependencies.jar
```

A.2. Manual de usuario.

A.2.1. Programa de adición de nombres de las vistas y los delimitadores.

Una vez descargados todos los archivos, se puede comenzar a trabajar con la herramienta. Puesto que la herramienta tiene como objetivo optimizar el control de versiones, se debe crear un repositorio o utilizar uno ya existente para enlazarlo con `gMaster`. Tras ello, deberemos seleccionar los ficheros `Vensim` que queremos utilizar y modificarlos a través del primer programa que modifica los archivos, en este caso, añadir los nombres de las vistas en las ecuaciones y añadir los delimitadores. Las dos interfaces gráficas de los formateadores son prácticamente idénticas, por lo que se debe comprobar que se está utilizando aquella con el fondo de color amarillo. Tras procesar el archivo, se mantendrá un archivo a modo de *backup* del archivo sin procesar con extensión `.2mdl` y el archivo procesado conservará tanto el nombre como la extensión.

La interfaz inicial, tal y como se aprecia el la figura A.1, consta de únicamente un botón. Dicho botón, `Open file`, como su propio nombre indica, sirve para abrir el archivo. Sabremos que el archivo se ha cargado cuando la interfaz se modifique tal y como se muestra en la figura A.2. En caso de abrir un fichero cuyo formato no es `.mdl` o cualquier otro error, se notificará al usuario a través de esa etiqueta con un llamativo color rojo. En caso de seleccionar un archivo al que ya se la haya aplicado dicho formateo, de nuevo se le indicará al usuario. Una vez cargado el archivo, se debe pulsar el botón de la izquierda, `Process`, el cual procederá a modificar el archivo y conservar el *backup* con la extensión `.2mdl`. Cuando haya finalizado el proceso, se notificará al usuario con un mensaje en color verde. En caso de haber seleccionado el archivo erróneo, se podrá hacer uso del botón de `Cancel` para volver a la interfaz inicial, figura A.1. Si queremos comprobar la ruta completa del fichero que hemos seleccionado, podremos pulsar en el botón de `Absolute path` para mostrar dicha ruta. Pulsando de nuevo recuperaremos el nombre del archivo sin ruta. Finalmente, y con el objetivo de hacer la experiencia de usuario más clara y concisa, se explicará en la parte inferior de la interfaz que efecto tiene procesar un fichero.

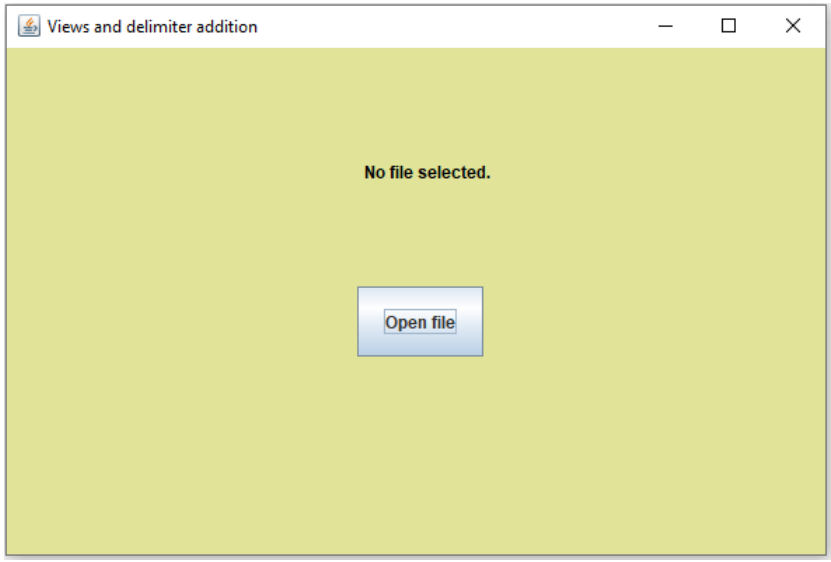

Figura A.1: Pantalla principal de la interfaz de adición de nombres de las vistas y delimitadores.

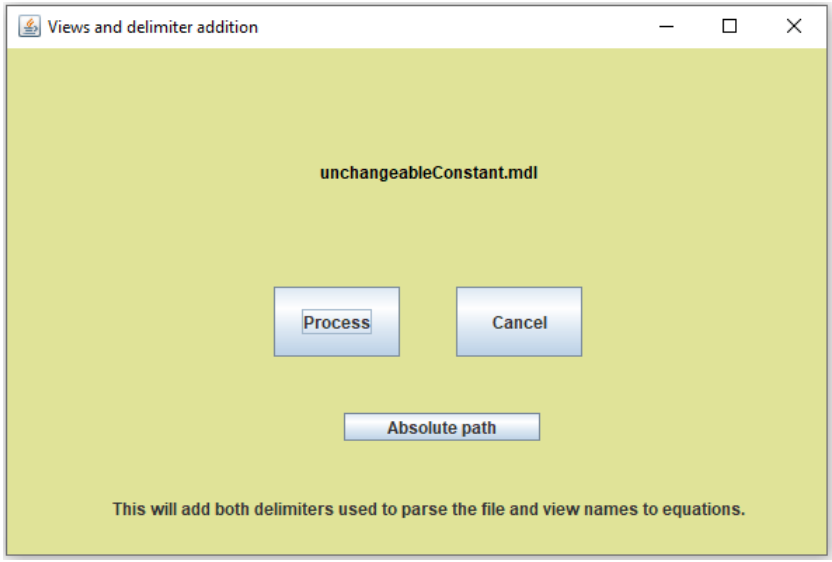

Figura A.2: Pantalla secundaria de la interfaz de adición de nombres de las vistas y delimitadores.

A.2.2. Plugin externo para gMaster.

Una vez transformados los archivos deseados, se subirán al repositorio. Es importante modificar los archivos, puesto que si no, `gMaster` no será capaz de reconocer los archivos

semánticamente, sino sólo como texto plano. Tras ello, se procederá bien a crear otra rama o bien a modificar el archivo directamente en la misma rama para posteriormente realizar un nuevo *commit*. Cuando se haya modificado el archivo, se procederá a abrir la herramienta de `gMaster`. Esta detectará que la extensión del archivo es `.mdl` y utilizará el *parser* externo para realizar el análisis semántico. Tal y como se explicó en el capítulo de **Introducción**, la herramienta nos mostrará las diferencias entre los dos archivos, proporcionándonos información sobre qué significa cada línea modificada. Cuando hayamos fusionado los archivos correctamente, podremos subir los archivos o *pushearlos* al repositorio.

En la siguiente imagen, figura A.3, se pueden apreciar las diferentes partes de la interfaz gráfica de `gMaster`. En la primera sección señalada se pueden observar el recuento de cambios entre dos versiones de un mismo archivo. En dicha sección se puede observar que los cambios se distinguen en cinco apartados principales: adiciones, modificaciones, eliminaciones, movimientos y renombres. Dentro de cada una de estas secciones, a su vez podemos distinguir la categoría del elemento modificado, como puede ser una ecuación y una vista, así como su nombre. En la segunda sección se observan las dos versiones del archivo que se está analizando, mostrando en la parte izquierda la versión más antigua y en la derecha la actual. Existen franjas coloreadas que relacionan los cambios visualmente entre ambas versiones. Al lado de cada cambio aparecerá una inicial, la cual indicará el tipo de modificación de entre las cinco posibles explicadas anteriormente. El primer recuadro en color azul de esta sección, situado a la izquierda, sirve para cambiar entre el modo convencional de control de versiones con texto plano, y el modo semántico propio de `SemanticMerge`. El segundo recuadro situado a la derecha, permite ver los cambios de una forma más visual, utilizando únicamente los grafismos asociados a los cinco tipos posibles de modificaciones. Por último, en la tercera sección se aprecian los archivos modificados en la versión del proyecto o *commit* seleccionado.

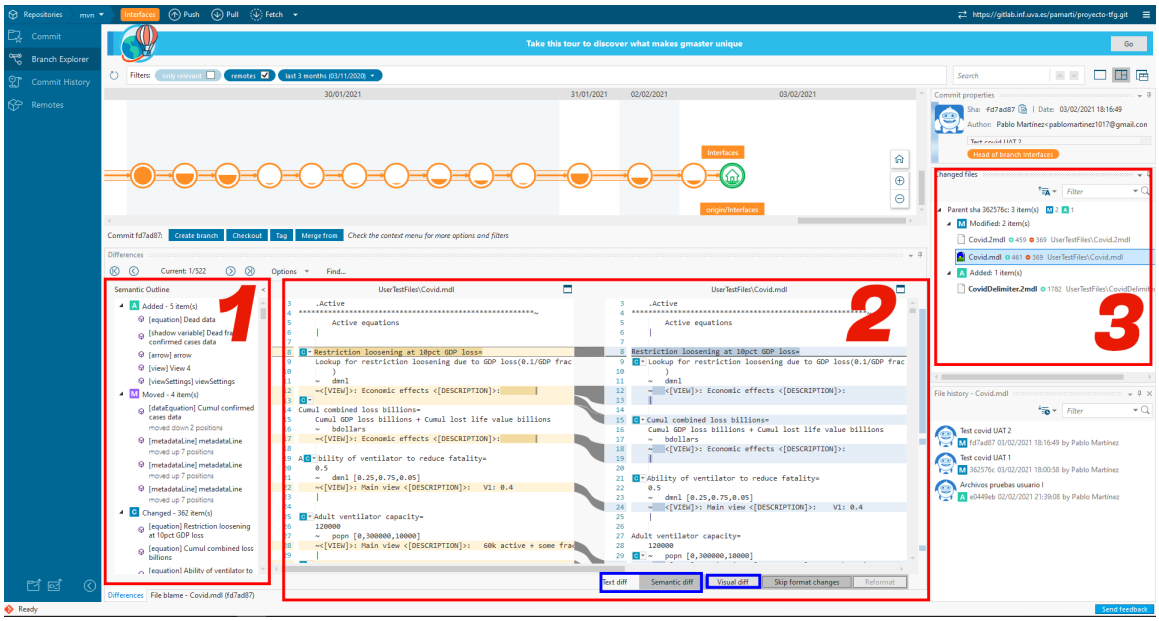

Figura A.3: Interfaz de gMaster y sus correspondientes partes.

A.2.3. Programa de borrado de los delimitadores.

Una vez tengamos los ficheros finales, deberemos utilizar el segundo programa para eliminar los delimitadores de secciones que se utilizan para hacer más fácil el análisis semántico del archivo. De no hacerlo, no se podrá abrir correctamente el archivo con la interfaz gráfica de `Vensim`. El proceso es el mismo que el del primer programa, y de nuevo, se debe prestar atención a utilizar aquella con el fondo gris predeterminado, puesto que las dos interfaces son prácticamente idénticas. De nuevo, al realizar la transformación de archivos, conservaremos a modo de *backup* el archivo antes de modificarse con formato `NombreDeArchivoDelimiter.2mdl` en el mismo directorio donde se encuentra el archivo a modificar. Dichas interfaces se muestran en la figuras A.4 y A.5.

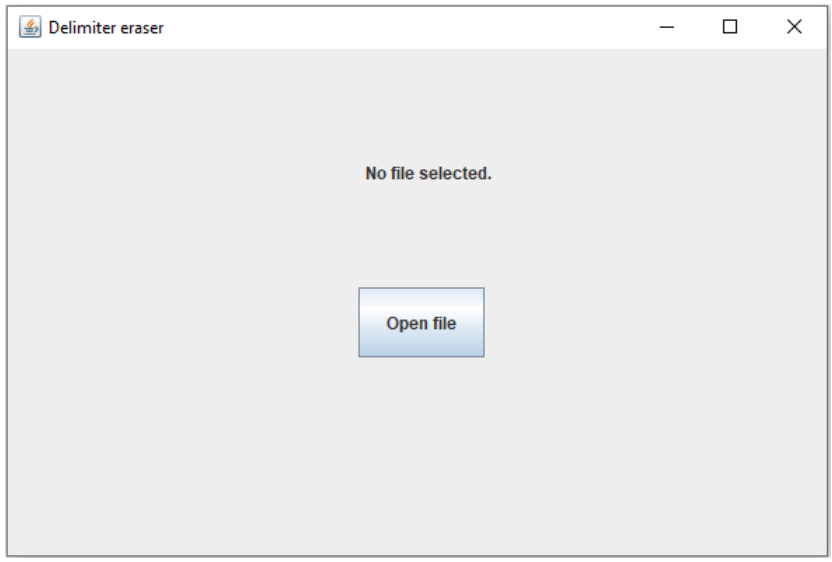

Figura A.4: Pantalla principal de la interfaz de eliminación de delimitadores.

A.2.4. Enlaces de interés.

En caso de querer conocer más acerca del funcionamiento específico de `SemanticMerge` y/o `gMaster` o cómo ambas herramientas implementan herramientas o *parsers* externos, se recomienda consultar [21], [31], [33] y [35] en la bibliografía del proyecto.

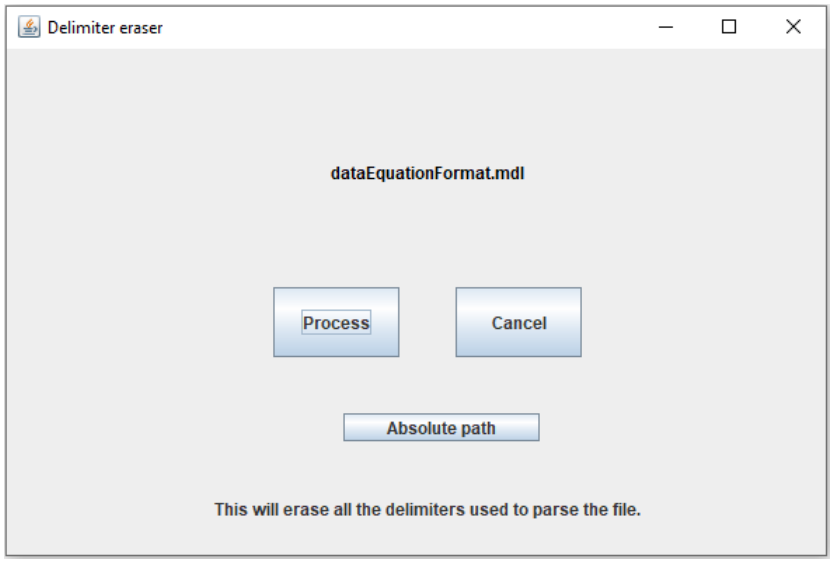

Figura A.5: Pantalla secundaria de la interfaz de eliminación de delimitadores.

Apéndice B

Resumen de enlaces adicionales

Los enlaces útiles de interés en este Trabajo Fin de Grado son:

- Repositorio del código: `https://gitlab.inf.uva.es/pamarti/proyecto-tfg`.

Bibliografía

[1] andrew1234. Java swing jfilechooser. https://www.geeksforgeeks.org/java-swing-jfilechooser/, 2018. Accesed: 2021-19-01.

[2] Apache. Maven. https://maven.apache.org/, 2020. Accesed: 2020-9-23.

[3] Baeldung. Java with antlr. https://www.baeldung.com/java-antlr, 2020. Accesed: 2020-9-15.

[4] Daniel Bazaco. Final vensim grammar in antlr 4, developed for this project. https://gitlab.inf.uva.es/danbaza/tfg-sonarvensim/-/blob/master/src/main/antlr/Model.g4, 2019. Accesed: 2020-9-15.

[5] Rocket Chat. Rocket chat. https://rocket.chat/, 2020. Accesed: 2020-9-23.

[6] CoronA. Antlr 4.5 - mismatched input 'x' expecting 'x'. https://stackoverflow.com/questions/29777778/antlr-4-5-mismatched-input-x-expecting-x, 2015. Accesed: 2020-9-18.

[7] Departamento de Informática. Archivos cloud vensim. https://cloud.infor.uva.es/index.php/s/3r85R4oavbe3ivk, 2020. Accesed: 2020-9-30.

[8] Universidad de Valladolid. Guía docente de la signatura. trabado de fin de grado (mención ingeniería de software). https://www.inf.uva.es/wp-content/uploads/2016/06/G46976.pdf, 2020. Accesed: 2020-10-2.

[9] Universidad de Valladolid. Preguntas frecuentes. https://www.uva.es/export/sites/uva/2.docencia/2.02.mastersoficiales/2.02.13.preguntasfrecuentes/index.html, 2020. Accesed: 2020-10-2.

[10] eclass. ¿cuáles son los eventos de scrum? https://blog.eclass.com/cuales-son-los-eventos-de-scrum-conocelos-aqui-0, 2020. Accesed: 2021-22-01.

[11] Norberto Figuerola. Riesgos: Plan de mitigación vs plan de contingencia vs fallback plan. https://articulospm.files.wordpress.com/2015/06/riesgos-plan-mitigacion-vs-plan-contingencia-vs-fallback-plan.pdf, 2015. Accesed: 2020-9-26.

[12] Tobi G. Java special characters in regex. https://stackoverflow.com/questions/14134558/list-of-all-special-characters-that-need-to-be-escaped-in-a-regex, 2019. Accesed: 2020-11-11.

[13] Research Gate. Locomotion h2020. https://www.locomotion-h2020.eu/, 2020. Accesed: 2020-9-16.

[14] Research Gate. Medeas. https://www.medeas.eu/#home, 2020. Accesed: 2020-9-16.

[15] GitLab.org. Gitlab. https://gitlab.com/gitlab-org, 2020. Accesed: 2020-9-23.

[16] James P. Houghton. test-models. https://github.com/SDXorg/test-models, 2020. Accesed: 2020-9-30.

[17] Joel Francia Huambachano. ¿qué es scrum? https://www.scrum.org/resources/blog/que-es-scrum, 2017. Accesed: 2021-22-01.

[18] La información. ¿cuál es el coste real de tener un trabajador para una empresa? https://www.lainformacion.com/practicopedia/cual-es-el-coste-real-de-un-trabajador-para-una-empresa/6491464/#:~:text=Cotizaciones%20a%20la%20Seguridad%20Social,26.300%20euros%20a%20la%20empresa., 2019. Accesed: 2020-9-26.

[19] Jitsi.org. Jitsi meet. https://meet.jit.si/, 2020. Accesed: 2020-9-23.

[20] Bart Kiers. If/else statements in antlr using listeners. https://stackoverflow.com/questions/15610183/if-else-statements-in-antlr-using-listeners, 2013. Accesed: 2020-9-15.

[21] Sergio L. Using external parsers with gmaster. http://blog.gmaster.io/2018/03/using-external-parsers-with-gmaster.html, 2018. Accesed: 2020-14-11.

[22] Pablo Santos Luaces. charposition. https://github.com/PlasticSCM/charposition, 2016. Accesed: 2020-31-10.

[23] Pablo Martínez López. Gramática en antlr4 para vensim. https://gitlab.inf.uva.es/pamarti/proyecto-tfg/-/blob/master/src/main/antlr4/es/uva/inf/grammar/Grammar.g4, 2020. Accesed: 2020-9-29.

[24] Pablo Martínez López. Simplejavaparser-semanticmerge. https://github.com/HylianPablo/SimpleJavaParser-SemanticMerge, 2020. Accesed: 2020-9-29.

[25] McGraw-Hill. Business dynamics. http://www.mhhe.com/business/opsci/sterman/models.mhtml, 2001. Accesed: 2020-9-30.

[26] Microsoft. Visual studio code. https://code.visualstudio.com/, 2020. Accesed: 2020-9-23.

[27] migraciones y seguridad social Ministerio de trabajo. Boletín oficial del estado. iii. otras disposiciones. https://www.boe.es/boe/dias/2019/06/03/pdfs/BOE-A-2019-8222.pdf, 2019. Accesed: 2020-9-26.

[28] MiryamGSM. External parser sample. https://github.com/PlasticSCM/external-parser-sample/tree/master-SCM20098, 2017. Accesed: 2020-8-28.

[29] picodotdev. Cómo ejecutar un proceso del sistema con java. https://picodotdev.github.io/blog-bitix/2016/03/como-ejecutar-un-proceso-del-sistema-con-java/, 2016. Accesed: 2020-16-12.

[30] PlasticSCM. Semanticmerge pricing. https://users.semanticmerge.com/Checkout, 2020. Accesed: 2020-9-26.

[31] Plastic SCM. gmaster - git gui. https://gmaster.io/, 2021. Accesed: 2021-21-01.

[32] Shodor. Vensim models. http://shodor.org/talks/ncsi/vensim/, 2005. Accesed: 2020-9-19.

[33] Códice Software. External parsers. https://semanticmerge.com/documentation/external-parsers/external-parsers-guide, 2016. Accesed: 2020-9-19.

[34] Códice Software. Advanced version control - pocket guide. https://www.plasticscm.com/documentation/advanced-version-control-guide#two-way-merge, 2020. Accesed: 2020-9-17.

[35] Códice Software. Semanticmerge 2.0. https://semanticmerge.com/, 2020. Accesed: 2020-7-11.

[36] Códice Software. Semanticmerge features. http://www.semanticmerge.com/features, 2020. Accesed: 2020-9-17.

[37] Códice Software. Semanticmerge intro guide. https://semanticmerge.com/documentation/intro-guide/semanticmerge-intro-guide, 2020. Accesed: 2020-9-17.

[38] Códice Software. Xdiff and xmerge. https://www.plasticscm.com/features/xmerge, 2020. Accesed: 2020-9-17.

[39] Saumitra Srivastav. Antlr4 - visitor vs listener pattern. https://saumitra.me/blog/antlr4-visitor-vs-listener-pattern/, 2017. Accesed: 2020-9-18.

[40] Johannes Link Matthias Merdes Marc Philipp Juliette de Rancourt Christian Stein Stefan Bechtold, Sam Brannen. Junit 5 user guide. https://junit.org/junit5/docs/current/user-guide/, 2020. Accesed: 2020-23-12.

[41] Ventana Systems. Defined and shadow variables. https://www.vensim.com/documentation/22890.htm, 2020. Accesed: 2020-10-6.

[42] Ventana Systems. Sketch information. https://www.vensim.com/documentation/ref_sketch_format.htm, 2020. Accesed: 2020-9-30.

[43] Ventana Systems. Sketch objects. https://www.vensim.com/documentation/index.html?23275.htm, 2020. Accesed: 2020-10-3.

[44] Ventana Systems. Variable types. https://www.vensim.com/documentation/ref_variable_types.htm, 2020. Accesed: 2020-9-16.

[45] Ventana Systems. Vensim. https://vensim.com/, 2020. Accesed: 2020-23-12.

[46] Linus Torvalds. Git. https://git-scm.com/, 2020. Accesed: 2020-9-23.

[47] TutorialsPoint. Compiler design - symbol table. https://www.tutorialspoint.com/compiler_design/compiler_design_symbol_table.htm#:~:text=Symbol%20table%20is%20an%20important,synthesis%20parts%20of%20a%20compiler., 2020. Accesed: 2020-10-7.

[48] Wangdq. How to display antlr tree gui. https://stackoverflow.com/questions/23809005/how-to-display-antlr-tree-gui, 2014. Accesed: 2020-9-24.

[49] Wikipedia. Swing (biblioteca gráfica). https://es.wikipedia.org/wiki/Swing_(biblioteca_gr%C3%A1fica), 2021. Accesed: 2021-20-01.