



UNIVERSIDAD POLITÉCNICA DE MADRID

Facultad de Informática

PROYECTO FIN DE CARRERA

Integración de evaluación de condiciones simples en Google Spreadsheet

**AUTOR:** Sergio Romaniega Pñeiro

**TUTOR:** Julio Mariño Carballo

**FECHA DE PRESENTACIÓN:** 17-7-2018

## Agradecimientos

Quisiera aprovechar para escribir unas palabras de agradecimiento a todas las personas que me han apoyado y ayudado durante la realización de este proyecto. En primer lugar, quisiera dar las gracias a Julio Mariño por darme la oportunidad de llevar a cabo este proyecto. Por su plena disponibilidad y consejos que me han ayudado enormemente a la realización de este proyecto.

También a mis padres y a mis hermanas por su comprensión y apoyo total durante toda la carrera, tan necesario e importante en algunos momentos críticos.

A ese grupo de compañeros que fuimos durante la carrera, siempre los unos ayudando a los otros en las múltiples asignaturas y prácticas de cada una de ellas.

Y, por último, a todos mis amigos que me han brindado esos buenos momentos durante estos años inolvidables.

Muchas gracias, este proyecto nunca hubiera lo mismo sin la aportación de cada uno de vosotros.

Sergio Romaniega Piñeiro



*La inteligencia consiste no sólo en el conocimiento,  
sino en la destreza de aplicar los conocimientos en la práctica.*

Aristóteles.

*Si buscas resultados distintos  
no hagas siempre lo mismo.*

Albert Einstein

*No siempre los más altos llegan mas arriba,  
no siempre los más rápidos llegan antes,  
no siempre los más fuertes son los que ganan la batalla.*

*Pero los que más entrenan,  
más se esfuerzan y más creen en el equipo  
son los que más veces conseguirán la victoria*

Pau Gasol



# Resumen

La hoja de cálculo es un bien que se emplea en multitud de ámbitos y es fundamental en estos días para todas las empresas. Se trata de una herramienta que facilita y automatiza gran parte de las tareas que se realizan en departamentos de economía, finanzas, recursos humanos, ..., etc. Sirve para el manejo de gran cantidad de datos que, aplicando una serie de funciones ya predefinidas en la hoja de cálculo, permiten el rápido uso de éstas funciones en multitud de filas y columnas. Pero como sabemos, todo tiene un límite y es aquí donde surge la idea de este proyecto. Éste proyecto de investigación se encarga de cubrir funcionalidades hasta ahora no cubiertas por las hojas de cálculo, pasando a darle inteligencia para la resolución de problemas complementando las operaciones estáticas aritméticas que posee actualmente. Ésta inteligencia nos permite, además de todas las funcionalidades básicas de las hojas de cálculo, ser capaces de resolver problemas más complejos mediante imposición de condiciones como por ejemplo las resolución de un sudoku o la realización de un horario. Para la definición de dicho modelo ha sido necesario un profundo estudio de todas las plataformas y posibilidades actuales en el mercado. Finalmente se ha decidido tomar como base otro proyecto desarrollado por alumnos de la Universidad Politécnica de Madrid y se ha integrado con la hoja de cálculo de Google, ya que simplemente añadiendo una extensión podemos disponer de todas estas funcionalidades extra. Éstas mejoras son añadidas automáticamente durante la carga de la página por lo que el usuario dispone de ellas desde el primer momento y sin ningún requerimiento específico para su utilización. Así, de una manera sencilla y rápida, ésta integración nos permite la resolución de problemas hasta ahora ni siquiera planteables para su resolución en este tipo de entorno.



# Índice general

<b>1. INTRODUCCIÓN</b>	<b>1</b>
1.1. Motivación y objetivos . . . . .	1
1.2. Estructura del documento . . . . .	2
<b>2. ESTADO DEL ARTE</b>	<b>5</b>
2.1. Hoja de Cálculo . . . . .	5
2.1.1. Origen . . . . .	5
2.1.2. Excel . . . . .	6
2.1.3. Otros sistemas con hojas de cálculo . . . . .	9
2.2. Desarrollo . . . . .	12
2.2.1. Lenguajes de Programación . . . . .	12
2.2.2. Entornos de Desarrollo . . . . .	16
2.3. Otras herramientas utilizadas . . . . .	18
2.3.1. GIT . . . . .	18
2.3.2. Apache . . . . .	19
2.3.3. Latex . . . . .	20
2.3.4. AWS . . . . .	20
<b>3. PLANTEAMIENTO DEL PROBLEMA</b>	<b>25</b>
3.1. Problema a resolver . . . . .	25
3.1.1. Empresas . . . . .	25
3.1.2. Particulares . . . . .	25



3.1.3. Oportunidad . . . . .	26
<b>4. SOLUCIÓN PLANTEADA</b>	<b>27</b>
4.1. Visión general . . . . .	27
4.1.1. Toma de decisión . . . . .	27
4.1.2. Descripción básica del sistema . . . . .	28
4.2. Descripción detallada del sistema . . . . .	28
4.2.1. Código de Drive . . . . .	28
4.2.2. Servidor . . . . .	37
4.2.3. Código Java . . . . .	39
<b>5. RESULTADOS Y CONCLUSIONES</b>	<b>43</b>
5.1. Resultados . . . . .	43
5.2. Fortalezas y mejoras . . . . .	44
5.3. Futuras líneas de investigación . . . . .	44
5.4. Conclusiones . . . . .	45
<b>6. Manual de Usuario</b>	<b>47</b>
6.1. Introducción . . . . .	47
6.2. Requisitos . . . . .	47
6.3. Instalación . . . . .	47
6.4. Uso de la aplicación . . . . .	49
6.4.1. Arranque de la aplicación . . . . .	49
6.4.2. Funciones . . . . .	49
6.4.3. Resultados . . . . .	55
<b>7. Bibliografía</b>	<b>57</b>

# Índice de Figuras

2.1. <i>ApacheOpenOffice</i> . . . . .	10
2.2. <i>Libre Office</i> . . . . .	10
2.3. <i>Quattro Pro</i> . . . . .	11
2.4. <i>Lotus 1-2-3</i> . . . . .	11
2.5. <i>Visicalc</i> . . . . .	12
2.6. <i>Eclipse</i> . . . . .	17
2.7. <i>NetBeans</i> . . . . .	17
2.8. <i>Estado del servidor</i> . . . . .	21
2.9. <i>Puertos accesibles</i> . . . . .	22
2.10. <i>Clave para acceso al Servidor</i> . . . . .	22
2.11. <i>Copias del Servidor</i> . . . . .	23
4.1. <i>Diagrama de la solución</i> . . . . .	28
4.2. <i>Nuevo Editor</i> . . . . .	29
4.3. <i>Nueva Pestaña</i> . . . . .	30
4.4. <i>sendData</i> . . . . .	31
4.5. <i>getData</i> . . . . .	31
4.6. <i>Domain</i> . . . . .	32
4.7. <i>Alldiff</i> . . . . .	33
4.8. <i>Ntimes</i> . . . . .	34
4.9. <i>Sudoku</i> . . . . .	35
4.10. <i>Resolver parte 1</i> . . . . .	36

---

4.11. <i>Resolver parte 2</i> . . . . .	37
4.12. <i>Index.php</i> . . . . .	38
4.13. <i>Eclipse Neon</i> . . . . .	39
4.14. <i>Main</i> . . . . .	40
4.15. <i>Ejecución escenario.pl</i> . . . . .	41
6.1. <i>Buscar módulo</i> . . . . .	48
6.2. <i>Añadir módulo</i> . . . . .	48
6.3. <i>Módulo instalado</i> . . . . .	48
6.4. <i>Domain 1</i> . . . . .	49
6.5. <i>Domain 2</i> . . . . .	50
6.6. <i>Domain 3</i> . . . . .	50
6.7. <i>AllDiff 1</i> . . . . .	51
6.8. <i>AllDiff 2</i> . . . . .	52
6.9. <i>NTimes 1</i> . . . . .	53
6.10. <i>NTimes 2</i> . . . . .	53
6.11. <i>NTimes 3</i> . . . . .	54
6.12. <i>Resolver 1</i> . . . . .	55
6.13. <i>Resolver 2</i> . . . . .	55

# INTRODUCCIÓN

## 1.1. Motivación y objetivos

El objetivo de este proyecto consta de la integración de una aplicación llamada Spread-Solver, desarrollada en Facultad de Informática por varios alumnos, en la hoja de cálculo de Google (SpreadSheet). Ésta aplicación es la encargada de calcular, dependiendo de los datos de entrada, si existe una posible solución realizando evaluación de condiciones simples y en caso afirmativo mostrar cada uno de los valores que se deben presentar para que se cumplan todos los requisitos descritos como parámetros de entrada. Se ha elegido la integración con SpreadSheet ya que es de código abierto y se puede integrar fácilmente con la simple instalación de un Addon o extensión.

Para ésta integración, el modelo se ha diseñado con las siguientes características:

- *Movilidad:* Ésta solución está realizada para que en cualquier lugar con conexión a internet se pueda disponer de ella.
- *Adaptabilidad:* Haciendo posible que la aplicación se ejecute en cualquier dispositivo sin importar las características ya que la mayor carga de trabajo la soporta el lado servidor.
- *Funcional e intuitivo:* Anteponiendo la sencillez a la hora de utilizar la solución a realizar, con un diseño que sea estético y funcional. En otras palabras, una solución fácil de usar.

- *Código estructurado y mantenible*: Para que de forma fácil se puedan modificar funciones implementando código claro y conciso.
- *Solución extendible y escalable*: Por si en un futuro fuese necesario aumentar las funcionalidades de la misma o se realizasen mejoras.

### 1.2. Toma de decisión

Para la realización de este proyecto de investigación lo primero que se tuvo que tener cuenta fué el diseño del modelo que se iba a crear y cómo iban a intercambiar la información y datos cada una de las piezas que conforman el diseño a alto nivel. Surgieron varias plataformas para el Front-End que a priori contaban con los requisitos para la integración como fueron, Excel, LibreOffice, OpenOffice, Quattro, Apache calc, ..., etc. Partiendo como premisa que se quería como solución una plataforma gratis se comenzó con la implantación de la solución en Libreoffice. Se descartó en una fase temprana del desarrollo ya que el código y la API para su desarrollo no eran lo simple y claros que se pretendía. Después de la experiencia previa se opta por la solución de Google y su hoja de cálculo, ya que muestra una API completa y con múltiples ejemplos y contribuciones en la web, quedando como solución final para este proyecto. Con respecto al servicio que alojara nuestro Back-End se optó de primeras por la solución de Amazon ya que es muy modulable y da mucha flexibilidad tanto a los servidores que queramos ejecutar como al tamaño y precio de ellos.

### 1.3. Estructura del documento

En el primer capítulo, *"Introducción y Objetivos"*, se realiza una breve introducción y se explican los objetivos y la motivación para la realización del proyecto.

En el segundo capítulo, *"Estado del Arte"*, se ofrece una descripción completa de todo lo referente al proyecto, donde se puede comparar con entornos equivalentes para obtener una visión global.

En el tercer capítulo, *"Planteamiento del problema"*, se profundiza en la carencia que se quiere cubrir al desarrollar esta aplicación.

En el cuarto capítulo, *"Solución planteada"*, se lleva a cabo el proceso completo de desarrollo de la aplicación explicando cada componente utilizado.

### 1.3 Estructura del documento

---

En el quinto capítulo, "*Resultados y Conclusiones*", se analizan a fondo los resultados obtenidos tras la finalización del proyecto, comparándolos con los objetivos marcados inicialmente, los puntos fuertes y débiles y se trazan las posibles líneas de desarrollo futuras.

Por último, la parte final del documento contiene varios anexos que complementan esta memoria añadiendo información para mejorar la comprensión del conjunto. Entre estos anexos se encuentra la bibliografía utilizada además de los manuales de usuario y de instalación referentes a la aplicación.



# Capítulo 2

## ESTADO DEL ARTE

### 2.1. Hoja de Cálculo

La hoja de Cálculo es una herramienta que nos permite realizar operaciones como sumas, restas, multiplicaciones o divisiones. Éstas operaciones se realizan en una tabla formada por celdas. Cada celda a su vez contendrá un valor fijo o una fórmula que será evaluada a partir de otras celdas.

#### 2.1.1. Origen

El inicio de la necesidad de un instrumento para la realización de cálculos se remonta hasta la era Mesopotámica alrededor del 2700-2300 a.C. con la aparición del comercio. Éste instrumento destinado a realizar operaciones aritméticas fue el ábaco.

Se trata de una tabla con columnas, que contiene unas fichas generalmente redondeadas insertadas en palos verticales de madera y ordenadas representando el sistema numérico sexagesimal. Se cree que su nacimiento fué en China, aunque otras teorías lo establecen en el norte de África.

Tiempo después en la antigua Grecia entre los años 150-100 a.C. se desarrolló el mecanismo de Anticitera, considerado el primer computador análogo mecánico. Su construcción está realizada a base de cobre y de estaño y sirvió para dar respuesta a teorías de matemáticas y astronomía. Fué descubierto a 45 metros bajo el agua frente a las costas de Grecia y solo parcialmente, posteriormente se han realizado varias expediciones por parte de Jacques Cousteau recuperándose más fragmentos del mecanismo, hasta un total de 82.



Fué mucho más tarde en 1972 cuando se creó el concepto de hoja de cálculo. El inventor fué un profesor que escribió una tabla en una pizarra y al equivocarse tuvo que borrar y rehacer múltiples celdas por lo que decidió crear un proceso que un ordenador fuera capaz de realizar los mismo pasos que él al realizar cualquier modificación en cualquier celda de la tabla. Esta idea fué la que dió lugar al VisiCalc, la primera hoja de cálculo, que sirvió como pionera para el establecimiento de las patentes realicionadas con elementos software.

Actualmente es mundialmente utilizada desde una vivienda hasta una multinacional por la riqueza y la variedad de funcionalidades que son posibles dentro de la hoja de cálculo, las cuales facilitan y simplifican las tareas diarias en cualquier organización.

### 2.1.2. Excel

Microsoft comercializó originalmente un programa de Hoja de cálculo llamado Multiplan en 1982, que fue muy popular en los sistemas CP/M , pero en los sistemas MS-DOS perdió popularidad frente al Lotus 1-2-3. Microsoft publicó la primera versión de Excel para Mac en 1985, y la primera versión de Windows (numeradas 2-05 en línea con el Mac y con un paquete de tiempo de ejecución de entorno de Windows) en noviembre de 1987. Lotus fue lenta al llevar 1-2-3 para Windows y esto ayudó a Microsoft a alcanzar la posición de los principales desarrolladores de software para hoja de cálculo de PC. Este logro solidificó a Microsoft como un competidor válido y mostró su futuro de desarrollo como desarrollador de software GUI. Microsoft empujó su ventaja competitiva lanzando al mercado nuevas versiones de Excel, por lo general cada dos años. La versión actual para la plataforma Windows es Excel 16.0, también denominada Microsoft Excel 2016. La versión actual para Mac OS X es Microsoft Excel 2011.

La lista de versiones de Microsoft Excel que han sido lanzadas al mercado para Microsoft Windows son:

- Versión 1 (1985). La primera versión de Excel fue creada para ser utilizada en la plataforma Macintosh de Apple.
- Versión 2 (1987). En este año se acababa de lanzar el sistema operativo Windows por lo tanto fue la primera versión de Excel para Windows sin embargo fue etiquetada como la versión 2 para dar continuidad con la versión creada previamente para Mac.

## 2.1 Hoja de Cálculo

---

- Versión 3 (1990). Para este año el sistema operativo Windows ya tenía una amplia aceptación en todo el mundo y Excel era el software estrella de Microsoft. En esta versión se introducen los gráficos 3-D.
- Versión 4 (1992). Durante más de dos años no hubo competencia para Excel en la plataforma Windows y se aprovechó para mejorar la herramienta.
- Versión 5 (1993). En esta versión finalmente Excel permite crear múltiples hojas dentro de un libro y agrega el soporte para el lenguaje de programación VBA.
- Versión 7 (1995). Se omite la versión 6 de Excel para empatar todos los productos de Microsoft Office [Word, Excel y PowerPoint] y a partir de esta versión se comienza a conocer a Excel por el año de su lanzamiento. Excel 95 es la primera versión de la hoja de cálculo en correr en un sistema operativo de 32-bits (como Windows 95).
- Versión 8 (1997). En Excel 97 se introduce el formato condicional y la validación de datos. Se realizan mejoras para los programadores de VBA al incluir un nuevo Editor de VBA e introducir los módulos de clase y los formularios de usuario.
- Versión 9 (1999). Se agrega el soporte para complementos COM y varias mejoras para las tablas dinámicas.
- Versión 10 (2001). Existen mejoras poco significativas pero se prepara esta versión para formar parte de Office XP y ser lanzado en conjunto con Windows XP.
- Versión 11 (2003). Esta versión ofrece un soporte mejorado para XML y correcciones en algunas funciones estadísticas.
- Versión 12 (2007). Un cambio significativo para la herramienta ya que se introduce un nuevo formato de archivo, una nueva interfaz de usuario que incluye la cinta de opciones. También se aumenta la capacidad de una hoja de cálculo al permitir muchas más filas y columnas.
- Versión 14 (2010). Un tanto supersticioso, Microsoft omite la versión 13 y sigue con la versión 14 también conocida como Excel 2010. En esta versión se introducen los minigráficos, la vista Backstage y la segmentación de datos.

- Versión 15 (2013). Esta versión de la herramienta introduce nuevas características como el relleno rápido de columnas que reconoce ciertos patrones en nuestros datos y hace sugerencias inmediatas. La nueva versión de Excel también analiza los datos para hacer sugerencias en cuanto a los gráficos a crear e inclusive nos hace sugerencias al momento de crear una tabla dinámica, además se introducen nuevas funciones en Excel 2013.
- Versión 16 (2015). Como parte de Office 2016, optimizado para Windows 10 y Windows 10 Mobile

A principios de 1993, Excel se convirtió en el objetivo de una demanda por otra empresa que ya tenía a la venta de un paquete de software llamado 'Excel'.<sup>en</sup> el sector financiero. Como resultado de la controversia Microsoft estaba obligada a hacer referencia al programa como "Microsoft Excel".<sup>en</sup> todos sus comunicados de prensa oficiales y documentos jurídicos. Sin embargo, con el tiempo esta práctica ha sido ignorada, y Microsoft aclaró definitivamente la cuestión cuando se adquirió la marca del otro programa. Microsoft también alentó el uso de las letras XL como abreviatura para el programa, el icono del programa en Windows todavía consiste en una estilizada combinación de las dos letras, y la extensión de archivo por defecto del formato Excel puede ser .xls en versiones anteriores o iguales a Excel 2003 (11.0), .xlsx para libros de Excel regulares en versiones posteriores o iguales a Excel 2007 (12.0), .xlsm para libros de Excel preparados para macros en versiones posteriores o iguales a Excel 2007 (12.0) o .xlsb para libros de Excel binarios en versiones posteriores o iguales a Excel 2007 (12.0).

Excel ofrece una interfaz de usuario ajustada a las principales características de las hojas de cálculo, en esencia manteniendo ciertas premisas que pueden encontrarse en la hoja de cálculo original, VisiCalc: el programa muestra las celdas organizadas en filas y columnas, y cada celda contiene datos o una fórmula, con referencias relativas ,absolutas o mixtas a otras celdas.

Excel fue la primera hoja de cálculo que permite al usuario definir la apariencia (las fuentes, atributos de carácter y celdas). También introdujo recomputación inteligente de celdas, donde celdas dependientes de otra celda que han sido modificadas, se actualizan al instante (programas de hoja de cálculo anterior recalculaban la totalidad de los datos todo el tiempo o esperaban para un comando específico del usuario). Excel tiene una amplia

## 2.1 Hoja de Cálculo

---

capacidad gráfica, y permite a los usuarios realizar, entre otras muchas aplicaciones, listados usados en combinación de correspondencia.

Cuando Microsoft primeramente empaquetó Microsoft Word y Microsoft PowerPoint en Microsoft Office en 1993, rediseñó las GUIs de las aplicaciones para mayor coherencia con Excel, producto insigne de Microsoft en el momento. Desde 1993, Excel ha incluido Visual Basic para Aplicaciones (VBA), un lenguaje de programación basado en Visual Basic, que añade la capacidad para automatizar tareas en Excel y para proporcionar funciones definidas por el usuario para su uso en las hojas de trabajo. VBA es una poderosa anexión a la aplicación que, en versiones posteriores, incluye un completo entorno de desarrollo integrado (IDE). La grabación de macros puede producir código (VBA) para repetir las acciones del usuario, lo que permite la automatización de simples tareas. (VBA) permite la creación de formularios y controles en la hoja de trabajo para comunicarse con el usuario. Admite el uso del lenguaje (pero no la creación) de las DLL de ActiveX (COM); versiones posteriores añadieron soporte para los módulos de clase permitiendo el uso de técnicas de programación básicas orientadas a objetos.

La funcionalidad de la automatización proporcionada por (VBA) originó que Excel se convirtiera en un objetivo para virus en macros. Este fue un grave problema en el mundo corporativo hasta que los productos antivirus comenzaron a detectar estos virus. Microsoft tomó medidas tardíamente para prevenir el uso indebido de Excel mediante la adición de la capacidad para deshabilitar la ejecución automática de las macros al abrir un archivo

### 2.1.3. Otros sistemas con hojas de cálculo

A continuación vamos a enumerar otros sistemas con hojas de cálculo de gran relevancia.

#### Apache OpenOffice Calc

Apache OpenOffice Calc es una hoja de cálculo libre y de código abierto que realizó su lanzamiento de la versión 1.1.0 en marzo de 2004 y se ha ido desarrollando hasta alcanzar la versión 4.1.3 actual. Está programado en C++ y Java y está creado para poder ser ejecutable en cualquier tipo de sistema operativo. También cuenta con licencia LGPL (Lesser General Public License), que pretende garantizar la libertad de compartir y modificar el software cubierto por ella, asegurando que el software es libre para todos sus

usuarios.



Figura 2.1: *ApacheOpenOffice*

### **Libre Office**

LibreOffice es un paquete de software libre y de código abierto desarrollado por The Document Foundation. Se creó como bifurcación de OpenOffice en 2010. El entorno está programado en los lenguajes informáticos Java, C++ y Python. Los sistemas operativos soportados son: GNU/Linux, OS X y Windows.



Figura 2.2: *Libre Office*

### **Quattro Pro**

Quattro Pro es un programa de hoja de cálculo desarrollado originalmente por la empresa Borland International, y desde 1996 perteneciente a la compañía desarrolladora canadiense Corel Corporation, como parte de su suite de oficina Corel WordPerfect Office.

Las distintas versiones por las que ha pasado Quattro Pro van desde su lanzamiento en 1989 con Quattro Pro 1.0 hasta la de 2012: Quattro Pro X6. Los sistemas operativos compatibles son Windows y MS-DOS. Está desarrollado en lenguaje ensamblador y Turbo C. Cabe destacar que a partir del año 1994 y tras la compra de la empresa por parte de Novell se empezó su distribución en las suites de WordPerfect Office. En 2006 con su versión 13 decidieron cambiar la nomenclatura a X3 para evitar éste número de mala suerte.

## 2.1 Hoja de Cálculo

---



Figura 2.3: *Quattro Pro*

### Lotus 1-2-3

Como se ha mencionado anteriormente Lotus 1-2-3 fué uno de los precursores de la hoja de cálculo, su lanzamiento tuvo origen en 1983 por la empresa Lotus Development Corporation, la cual fué comprada en 1996 por IBM. Fué creado únicamente compatible con el sistema operativo Microsoft Windows, ya que varios intentos posteriores de cambiar el core" fueron fallidos por las necesidades del mercado y de su poca adaptabilidad al estar programado en lenguaje ensamblador, han llevado a que ,a pesar de tener la versión 9 del Lotus alguna mejora comparandola con Excel, el propio IBM haya dejado su desarrollo en segundo plano.



Figura 2.4: *Lotus 1-2-3*

### Google Spreadsheet

Esta facilidad creada por Google salió al mercado en forma de beta para un limitado número de usuarios en Junio del 2006. Su programación está realizada en Ajax y es de libre disposición al público siempre que se tenga una cuenta de correo perteneciente a Google, en este caso basta con tener una cuenta de gmail para poder hacer uso de este servicio. La novedad que tiene este método es que las hojas de cálculo que se creen pueden

ser accedidas desde cualquier parte o dispositivo que disponga de internet ya que son creadas en la nube. Este proyecto se ha realizado utilizando ésta facilidad debido a la gran versatilidad que ofrece tener alojados documentos en la nube, control de versiones, recuperación de documentos borrados por equivocación, fácil accesibilidad, ..etc.

### VisiCalc

Y por último y no menos importante tenemos VisiCalc, que como hemos denominado, ha sido la primera hoja de cálculo disponible para ordenadores personales. Fué creado en 1978 por Dan Brinklin en la compañía Software Arts. Fué diseñado para el sistema operativo de Apple SOS y DOS. Al estar orientado a un sistema operativo diferente de todos los anteriores, VisiCalc se convirtió en el mayor exponencial de Apple a pesar de tener únicamente 5 columnas y 20 filas.



Figura 2.5: *Visicalc*

## 2.2. Desarrollo

Para el desarrollo del proyecto han sido necesarias la aplicación y utilización de varios lenguajes de programación así como distintas herramientas. Esta variedad que vamos a enumerar a continuación ha sido necesaria debido a la cantidad de entornos distintos necesarios, y sin embargo indispensables, para poder realizar la integración del módulo desarrollado en Google Spreadsheet.

### 2.2.1. Lenguajes de Programación

#### Java

Java es un lenguaje de programación orientado a objetos lanzado en 1995 por Sun Microsystems (posteriormente adquirido por Oracle) con su versión JDK 1.0. Desde entonces, ha experimentado múltiples cambios, así como un gran incremento en el número de clases y paquetes que componen la biblioteca estándar. Ha ido ganado popularidad hasta el punto de ser uno de los lenguajes más difundidos y utilizados del mundo. Surgió a

## 2.2 Desarrollo

---

partir de los lenguajes C y C++, pero elimina ciertos aspectos de los mismos como por ejemplo el manejo de memoria con punteros y se convierte en un lenguaje completamente orientado a objetos. Como características principales del lenguaje, podemos destacar que es multiplataforma, seguro, compilado y fácilmente ampliable.

Para desarrollar en Java es imprescindible la instalación de la herramienta JDK (Java Development Kit). Una vez compilados los ficheros, pueden ser portados a otra máquina sin necesidad de recompilarlos. Lo único necesario para ejecutarlos es tener instalada también la máquina virtual de Java (JVM). El lenguaje en sí mismo está especificado en la Java Language Specification (JLS) y regulado por el Java Community Process (JCP).

La parte en la cual ha sido necesario el conocimiento de Java, es el módulo encargado del parser de los datos que le llegan al servidor, y que prepara los datos para la generación del fichero de entrada, que posteriormente, se ejecutará en prolog.

### Php

Lenguaje de programación orientado a objetos, inspirado en la sintaxis de C y escrito inicialmente por Rasmus Lerdorf en 1995. Es considerado software libre bajo la licencia PHP License. Fue diseñado desde sus inicios para ejecutarse en el lado del servidor y orientado al desarrollo web permitiendo generar código dinámico. PHP es el acrónimo de Personal Home Page Tools , sin embargo su nombre oficial es PHP Hypertext Procesor desde 1997. PHP es capaz de ser ejecutado en casi todos los sistemas operativos y como he comentado anteriormente, me he decantado hacia la distribución de Linux CentOS, sobre el que se ejecuta el servidor web Apache. PHP es uno de los lenguajes de programación más extendidos para el desarrollo web y al ser software libre, es el lenguaje elegido para realizar herramientas, gestor de contenidos, librerías, plugins de software libre. Un claro ejemplo es la gran cantidad de gestores de contenidos que existen desarrollados en este lenguaje como por ejemplo:

- Drupal.
- Wordpress.
- Joomla.
- ATutor.



- Moodle.
- Magento.
- Prestashop.

PHP es un lenguaje muy útil y extendido, sin embargo en mi opinión no ha evolucionado con las nuevas tecnologías de desarrollo ágil como son Node.js o Swift.

Es el lenguaje en que está desarrollada nuestra página estándar del servidor o también denominada index.html Encargada de recibir los datos de entrada y pasarlos como parámetros al .jar

### Javascript

Javascript es un lenguaje de programación interpretado y orientado a objetos lanzado en 1995 por Netscape. Se trata de un lenguaje que normalmente se usa en el lado cliente, por eso su denominación como lenguaje de scripts para páginas web. Sin embargo también posee implementaciones significativas en la parte de servidor como Node.js que es usado en proyectos importantes. Debido a su gran utilización en los navegadores, y para evitar incompatibilidades, forma parte del estándar ECMAScript que poco después se convirtió en un estándar ISO.

De las grandes ventajas que ofrece JavaScript es que se ejecuta en el navegador del cliente, por lo que las respuestas son más rápidas y pueden dar respuesta a comportamientos o acciones del usuario que simplemente con XML no se es capaz de realizar. Esta tendencia es cada vez mayor, Ajax hace uso de esta técnica de manera similar.

La programación del script en Google Spreadsheet está basado en este lenguaje de programación, aunque no es posible utilizar todas las funciones que posee el lenguaje en sí, ya que sólo es posible realizar operaciones con los objetos y clases definidos en la API. Todos ellos están orientados a operaciones con celdas y hojas de cálculo por lo que solo se ha tenido alguna dificultad para el envío y la recepción de los datos hacia o desde el exterior.

## 2.2 Desarrollo

---

### Xml

XML (eXtensible Markup Language) es un Lenguaje de Etiquetado Extensible estandarizado y proviene de SGML (Standard Generalized Markup Language). SGML proviene a su vez de GML (Generalized Markup Language), lenguaje creado por IBM en los años setenta, que surgió de la necesidad de la misma de almacenar grandes cantidades de información y compartirla con otros sistemas operativos y plataformas. La ISO se interesó por el mismo y trabajaron para normalizarlo creando SGML en 1986.

Es de código abierto, muy simple pero estricto, y fundamental en el intercambio de una gran variedad de datos. Está desarrollado por el World Wide Web Consortium (W3C) y es utilizado para estructurar, almacenar e intercambiar datos de forma legible. Es muy similar a HTML, pero su función principal es describir datos y no mostrarlos como es el caso de HTML. XML es una tecnología sencilla que tiene a su alrededor otras que la complementan y le dan grandes posibilidades. Tiene un papel muy importante en la actualidad, puesto que se puede usar en editores de texto, bases de datos, hojas de cálculo...

Un documento XML se compone de prólogo y cuerpo, conteniendo el primero las declaraciones necesarias respecto a xml y al documento, y conteniendo el segundo la parte más importante que son los datos a los que se les ha añadido el marcado estructurados en forma de árbol.

### Prolog

Se trata de un lenguaje de programación lógica que se dió a conocer en Francia en los años 70, sus autores son Phillipe Rousset y Alain Colmerauer. Desde sus inicios se trató como un lenguaje interpretado hasta que David Warren desarrolló en 1983 un compilador capaz de generar una máquina abstracta a partir de un conjunto de líneas de código escritas en Prolog. Fué a partir de entonces que Prolog se trata de un lenguaje semiinterpretado, esta característica perdura en la actualidad.

La principal característica en que está basado Prolog es en la ruptura de la clásica ejecución de código secuencial y lineal, sólomente alterado por instrucciones como bucles o condiciones de salto. En Prolog, sin embargo, su ejecución está formada por la unificación y el backtracking.

Los programas están compuestos por cláusulas lógicas llamadas cláusulas de Horn, que

determinan que si es verdad la primera parte de la cláusula entonces será verdad la segunda parte.

En la parte del proyecto que nos atañe, cabe decir que el programa que se ejecuta en la parte servidor, consta de una parte desarrollada en Prolog, en este caso en concreto, con la versión en desarrollo CiaoDE 1.15 desarrollada por el departamento del IMDEA-SW. Ha sido necesaria la utilización de esta versión para poder tener disponible la librería CLP(FD), capaz de realizar evaluación de condiciones simples. Ésta versión se ha tenido que instalar en un servidor Linux de Amazon.

### **2.2.2. Entornos de Desarrollo**

Del inglés Integrated Development Environment (IDE) es un programa que proporciona servicios integrales para facilitar el desarrollo de software a un programador, como por ejemplo la creación, modificación, compilación, implementación y depuración de software. "Maestro I", producto de Softlab Múnich creado en 1975, fue el primer entorno de desarrollo integrado IDE para software. Fue instalado por 22.000 programadores en todo el mundo.

Suele contener un editor de código fuente, un compilador, un intérprete y un depurador entre otros. Permiten la agregación automática de fragmentos de código y el auto-completado, puede permitir la búsqueda de objetos y la jerarquía de clases. Suele distinguir las palabras clave del lenguaje de programación del resto de palabras y puede servir para un único lenguaje o para varios a través de la instalación de plugins para los mismos. También puede incluir simuladores y la posibilidad de elegir versiones y suele tener un entorno gráfico. La mayor ventaja de un Entorno de Desarrollo es que aumenta en gran medida la productividad del programador. En algunos lenguajes, el entorno de desarrollo puede funcionar como un sistema en tiempo de ejecución. Actualmente, además, suelen contener menú de ayuda, reconocimiento de sintaxis, diccionarios, control de versiones y soporte a multilenguaje.

#### **Eclipse**

Eclipse es una plataforma software multilenguaje para desarrollar "Aplicaciones de Cliente Enriquecido" compuesto por un conjunto de herramientas de programación de código abierto multiplataforma. Eclipse surgió como sucesor de VisualAge y fue desarro-

## 2.2 Desarrollo

---

lado por IBM con su primera versión estable en 2014. Actualmente, es mantenido por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios. Está programado mayoritariamente en el lenguaje Java, aunque también utiliza ANSI C y algunos más. Originalmente fue liberado bajo la Common Public License, pero después se cambió a Eclipse Public License. Ambas licencias son de software libre, pero incompatibles con GNU GPL. La última versión estable es la 4.5.2 de febrero de 2016.



Figura 2.6: *Eclipse*

### NetBeans

NetBeans es una plataforma multilenguaje y multiplataforma para el desarrollo de aplicaciones de escritorio usando Java y un IDE desarrollado usando la Plataforma NetBeans. Comenzó como un proyecto de estudiantes en Republica Checa (llamado Xelfi) en 1996 en la Facultad de Matemáticas y Física de la Universidad de Charles en Praga. Actualmente es un proyecto de código abierto fundado por Sun Microsystems junio de 2000 tras su compra un año antes. Ese mismo año pasó a ser de código abierto. La última versión estable es la 8.1 de noviembre de 2015. Desde julio de 2006, NetBeans IDE tiene licencia Common Development and Distribution License (CDDL), una licencia basada en la Mozilla Public License (MPL). En octubre de 2007, pasó a tener licencia CDDL y GPL versión 2.



Figura 2.7: *NetBeans*

### Otros Servicios Linux

Para terminar con el lado del backend, veremos otros servicios Linux que han sido necesarios.

- SSH: Secure Shell, permite conectarse al servidor a distancia y ejecutar su línea de comandos.
- VI: Editor de ficheros propio de la distribución UNIX.
- SFTP: Basado sobre SSH permite la transmisión de ficheros entre dos ordenadores.
- Crontab: Los sistemas Linux tienen la posibilidad de ejecutar ciertos comandos configurados previamente de forma periódica. Lo utilizamos para controlar las versiones y hacernos una copia por si hubiera algún problema en el servidor.
- Git: Permite gestionar y enviar los cambios al sistema de control de versiones.
- NTP: Sincroniza la hora entre servidores Linux.

### 2.3. Otras herramientas utilizadas

A parte de la herramienta principal utilizada para la implementación de la aplicación, ha sido necesario el uso de otras herramientas para el adecuado desarrollo y mantenimiento del proyecto. Todas estas son muy útiles en proyectos de mediana y gran escala, los cuales tienen un periodo largo de duración o intervienen numerosos ingenieros. No tienen tanto sentido aquí, pero son sin duda una buena práctica a seguir. A continuación pasamos a describir cada una de ellas de forma breve.

#### 2.3.1. GIT

Git es un software de control de versiones de código abierto diseñado por Linus Torvalds, pensando en la gestión y mantenimiento eficiente, rápida y fiable de las diferentes versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente y trabajan en ellas diferentes personas. Al principio, Git se pensó como un motor de bajo nivel sobre el cual otros pudieran escribir la interfaz de usuario o front end como Cogito o StGIT. Sin embargo, Git se ha convertido desde entonces en un sistema de control de versiones

## 2.3 Otras herramientas utilizadas

---

con funcionalidad plena. Hay algunos proyectos de mucha relevancia que ya usan Git, en particular, el grupo de programación del núcleo Linux.

El mantenimiento del software Git está actualmente (2009) supervisado por Junio Hamano, quien recibe contribuciones al código de alrededor de 280 programadores.

### GitHub

GitHub es una plataforma de desarrollo colaborativo de software para alojar proyectos utilizando el sistema de control de versiones Git. Utiliza el framework Ruby on Rails por GitHub, Inc. (anteriormente conocida como Logical Awesome). Desde enero de 2010, GitHub opera bajo el nombre de GitHub, Inc. El código se almacena de forma pública, aunque también se puede hacer de forma privada, creando una cuenta de pago.

En la actualidad, GitHub es mucho más que un servicio de alojamiento de código. Además de éste, se ofrecen varias herramientas útiles para el trabajo en equipo. Entre ellas, caben destacar:

- Una wiki para el mantenimiento de las distintas versiones de las páginas.
- Un sistema de seguimiento de problemas que permite a los miembros de tu equipo detallar un problema con tu software o una sugerencia que deseen hacer.
- Una herramienta de revisión de código, donde se pueden añadir anotaciones en cualquier punto de un fichero y debatir sobre determinados cambios realizados en un commit específico.
- Un visor de ramas donde se pueden comparar los progresos realizados en las distintas ramas de nuestro repositorio.

### 2.3.2. Apache

Apache es un servidor web, muy configurable gracias a la cantidad de módulos que se pueden integrar, surgió en 1995 y su código está bajo la licencia Apache 2.0. Se calcula que el 70% de los servidores web son Apache. Cabe mencionar que Apache es el servidor web por defecto de todos la mayoría de las distribuciones linux. También es interesante saber que las nuevas tecnologías que están emergiendo actualmente eliminan cualquier servidor

web de por medio, siendo totalmente innecesario sobrecargar y relentizar una web con un software tan grande y pesado como Apache. Con esto me refiero a Node.js que su propio código es capaz de servir y enrutar páginas web como si se tratase de un servidor HTTP. En el proyecto se ha configurado en el servidor alojado en AWS (Amazon Web Services), que se encarga de recibir las peticiones de nuestra spreadsheet de Google.

### 2.3.3. Latex

L<sup>A</sup>T<sub>E</sub>X es un sistema de composición de textos, orientado a la creación de documentos escritos que presenten una alta calidad tipográfica. Por sus características y posibilidades, es usado de forma especialmente intensa en la generación de artículos y libros científicos que incluyen, entre otros elementos, expresiones matemáticas.

L<sup>A</sup>T<sub>E</sub>X está formado por un gran conjunto de macros de TeX, escrito por Leslie Lamport en 1984, con la intención de facilitar el uso del lenguaje de composición tipográfica, TeX, creado por Donald Knuth. Es muy utilizado para la composición de artículos académicos, tesis y libros técnicos, dado que la calidad tipográfica de los documentos realizados con LaTeX es comparable a la de una editorial científica de primera línea.

Además, L<sup>A</sup>T<sub>E</sub>X es software libre bajo licencia LPPL.

### TeXstudio

TeXstudio es un editor de LaTeX de código abierto y multiplataforma con una interfaz similar a Texmaker. Es un IDE de LaTeX que proporciona soporte moderno de escritura, como la corrección ortográfica interactiva, el plegado de código y el resaltado de sintaxis.

Originalmente se llamó TexMakerX, y se inició como un Fork de Texmaker tratando de extenderlo con características adicionales, pero manteniendo su apariencia. Se ejecuta en Windows, Unix / Linux, BSD, y sistemas Mac OS X.

### 2.3.4. AWS

AWS son las siglas de Amazon Web Services y es la plataforma donde se ha alojado la solución planteada y dónde ha sido necesaria la configuración de la mayor parte de las piezas que conforman todo el proyecto.

Las partes más destacadas que consta el servidor son:

## 2.3 Otras herramientas utilizadas

---

- Apache
- Index.html
- Spreadsolver.jar
- CiaoDE 1.15

Hablaremos de todas ellas en detalle más adelante.

Para la configuración de este servidor se ha contado con la plataforma que proporciona Amazon llamada EC2.



Figura 2.8: *Estado del servidor*

En la imagen podemos ver en todo momento, y en tiempo real, el estado de nuestro servidor. También se han realizado distintas tareas en este apartado como por ejemplo:

1. *Seguridad* Se ha configurado únicamente que un cierto tipo de puertos estén accesibles desde el exterior para evitar ataques o la ejecución de cualquier tipo de software malicioso. Se han dejado accesibles los puertos 80, 443 y 22 a través de la configuración de un grupo de seguridad.



Type <i>i</i>	Protocol <i>i</i>	Port Range <i>i</i>	Source <i>i</i>
HTTP	TCP	80	0.0.0.0/0
SSH	TCP	22	0.0.0.0/0
HTTPS	TCP	443	0.0.0.0/0

Figura 2.9: Puertos accesibles

2. *Clave* Para certificar que con la seguridad anterior no se pudiera acceder al servidor, se ha generado una clave servidor-cliente identificando desde el origen desde donde es posible la conexión al servidor por SSH.

Key pair name	Fingerprint
serverkeypar	9c:8a:a9:d8:3f:cf:41:b6:cd:2c:88:75:de:c6:6f:e7:92:8b:1b:82

Figura 2.10: Clave para acceso al Servidor

3. *Copia de seguridad* Para asegurarnos que si sufrimos algún problema con el servidor, estado inconsistente, borrado accidental de ficheros, intrusión de terceros, etc, se han creado varias copias del estado de la máquina asegurándonos de la fiabilidad de los datos y configuraciones realizadas en el servidor.

## 2.3 Otras herramientas utilizadas

---

Create Volume   Actions ▾

🔍 Filter by tags and attributes or search by keyword

<input type="checkbox"/>	Name ▾	Volume ID ▾	Size ▾	Volume Type ▾	IOPS ▾	Snapshot ▾
<input type="checkbox"/>		vol-daecb8f	10 GiB	standard	-	snap-24cb800c
<input type="checkbox"/>		vol-dbecb8e	8 GiB	standard	-	snap-135ce239

Figura 2.11: Copias del Servidor



## PLANTEAMIENTO DEL PROBLEMA

### 3.1. Problema a resolver

Como hemos visto y una vez ubicados en el marco tecnológico, somos capaces de ver la evolución y el continuo desarrollo de las herramientas anteriormente descritas y por ello somos capaces de entender la necesidad de realizar este proyecto.

Para dar más sentido y permitir mayor comprensión sobre las necesidades y los beneficios del desarrollo de esta herramienta, a continuación veremos de una manera explícita los problemas planteados que nuestra herramienta es capaz de resolver en el mundo real.

#### 3.1.1. Empresas

Con el avance tecnológico, todas las empresas ya sean PYMES o grandes multinacionales, cuentan con un departamento de finanzas. Es clara la tendencia que se está tomando por la mayoría de ellas el tener ciertos servicios en la nube. Un claro ejemplo de esto es el Office365, que permite a los usuarios tener todas las herramientas de Microsoft accediendo via web o descargando sus productos en hasta 5 dispositivos distintos, portátil, sobremesa e incluso teléfono móvil.

#### 3.1.2. Particulares

También es importante destacar que las nuevas generaciones están cada vez más conectadas a las nuevas tecnologías y no es raro ver personas que para su uso privado utilizan el Excel diariamente, para el control de gastos mensual, control de nóminas, etc.

### 3.1.3. Oportunidad

Por todo ello, consideramos interesante desarrollar un módulo integrado en Google Drive capaz de dar respuesta a muchas necesidades diarias que afectan a multitud de usuarios.

Por otro lado, intentaremos que la aplicación sea lo más accesible a los usuarios. Por ello se va a realizar un manual de usuario con alto nivel de detalle para que cualquier duda que pudiera surgir al utilizarse sea fácilmente resuelto buscando rápidamente la función que se está queriendo utilizar. De momento, no está planteado el subir la aplicación a la tienda de aplicaciones, sino que su desarrollo ha sido más bien desde el punto de vista docente. No obstante, las funcionalidades que incluye cubren un hueco actualmente existente en el mercado y puede llegar a ser muy interesante con las adecuadas mejoras.

En el capítulo siguiente explicaremos todo el proceso y cada componente de manera individual para que sirva tanto para explicar el trabajo realizado como para posibles líneas de trabajo futuras.

# Capítulo 4

## SOLUCIÓN PLANTEADA

### 4.1. Visión general

Por todo lo anterior, se pueden plantear varios escenarios posibles para el problema, viendo todas las posibilidades descritas de plataformas existentes y que pueden servir como base para la implementación de nuestro módulo de evaluación de dominios finitos.

#### 4.1.1. Toma de decisión

Primeramente se optó por la implementación basada en “LibreOffice” que ofrecía facilidad y versatilidad en cuanto a la adquisición, ya que se trata de una de las herramientas descritas anteriormente sin coste de adquisición o necesidad de licencia. Cuando se fué profundizando más en el desarrollo se pudo comprobar que las facilidades y el entorno de desarrollo, lenguaje utilizado, APIs, etc., no eran los más apropiados por carecer de gran documentación y ser capaz de realizar algunas de las tareas fundamentales que se requerían para el desarrollo.

Posteriormente y tras descartar definitivamente a “LibreOffice”, se sopesaron varias opciones. Ya con un mayor conocimiento de lo necesario se decidió realizar el desarrollo para Google Drive y su Spreadsheet o hoja de cálculo. Los factores fundamentales para esta decisión fueron sobre todo la muy buena documentación de la API y el lenguaje de programación utilizado “javascript”, ya conocido por haber sido necesario en alguna de las asignaturas cursadas durante la carrera.

### 4.1.2. Descripción básica del sistema

Para este proyecto, ha sido necesario el código proporcionado por uno de los alumnos y compañero de la Facultad de Informática, desarrollado en “Java“. Éste código ha pasado de poder ejecutarse localmente en un pc a poder ejecutarse dentro de la hoja de cálculo de Google. Para ello ha sido necesario el desarrollo de los siguientes elementos que intervienen en todo el ciclo de vida de los datos.

- *Código de Drive* Necesario para la definición de las funciones y del envío y recepción de los datos del servidor externo.
- *Servidor* Sistema donde se va alojar el código Java modificado y va a resolverse, si hay solución, el problema planteado.
- *Código Java* El “.Jar“ modificado para la utilización de la interfaz del Spreadsheet de Google.

En la siguiente imagen podemos ver cómo se interrelacionan cada una de las partes enumeradas anteriores:

Figura 4.1: *Diagrama de la solución*

## 4.2. Descripción detallada del sistema

### 4.2.1. Código de Drive

Para que la hoja de cálculo sea capaz de tener nuevas funciones, ha sido necesaria la incorporación de una nueva pestaña llamada “Finite Domains“, en la cual, hemos definido todas las funciones necesarias para poder sustituir la interfaz escrita en Java. Para escribir el código es necesario definirse un nuevo “Sript editor“ de la forma:

## 4.2 Descripción detallada del sistema

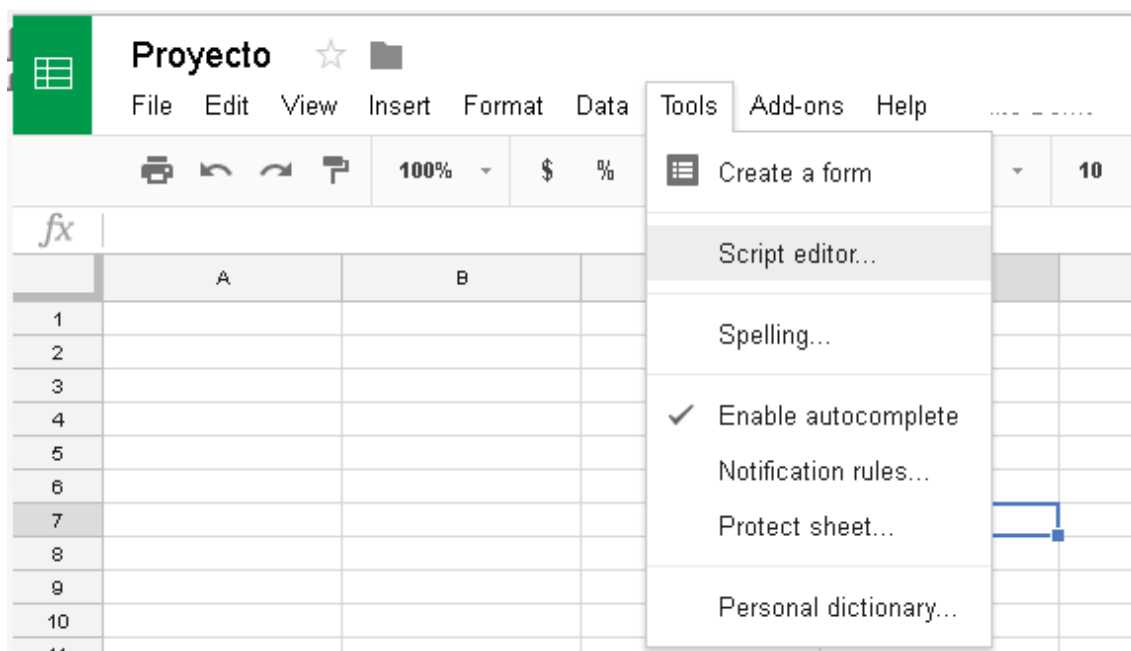


Figura 4.2: *Nuevo Editor*

La página que aparece en este caso le hemos dejado el nombre que nos da por defecto “Code.gs” y, es donde se va a empezar a codificar las distintas funciones definidas en Javascript y que están contempladas en el API de Google.

### onOpen

Para la creación de la nueva pestaña es necesaria la utilización de la función “onOpen()” que, nos asegura que siempre que se abra una hoja de cálculo, se añada la pestaña definida como:



```
function onOpen() {
  var ss = SpreadsheetApp.getActiveSpreadsheet();
  var menuEntries = [{name: "Domain", functionName: "Domain"},
                    {name: "AllDiff", functionName: "AllDiff"},
                    {name: "NTimes", functionName: "NTimes"},
                    {name: "Sudoku", functionName: "Sudoku"},
                    {name: "Resolver", functionName: "resolver"}];
  ss.addMenu("Finite Domains", menuEntries);
  PropertiesService.getScriptProperties().setProperty('numero', "1");
};
```

Figura 4.3: Nueva Pestaña

En éste código se puede ver como están definidas todas las funciones, que van a formar parte de este nuevo menú. Además se ha definido una variable local “numero“, que servirá para la definición de los parámetros del Dominio y, fundamental si se va a realizar la definición de varios dominios. Veamos como funciona:

Se determina el valor inicial de la variable número a “1“, a partir de ahí este valor se va a ir incrementando según se vayan definiendo nuevos valores en la función Dominio, por ejemplo: Si se define una celda nueva de dominio con los valores rojo, verde, azul. éste valor se va a ir incrementando para la representación interna de los posibles valores a asociar para las soluciones. En este caso serán (rojo:1, verde:2, azul:3), con lo que nuestra variable local número pasará a valer 4 por haberse definido 3 valores en el dominio y así poder asignarle el valor 4 al primer elemento del siguiente dominio de se defina.

### sendData

Se trata de la parte encargada de realizar las peticiones al servidor externo de Amazon. Como se puede ver se ha utilizado el método “post“ para el envío de datos y encapsulados en formato JSON.

## 4.2 Descripción detallada del sistema

---

```
function sendDataNoReturn(urlTarget, data){
    var options = {
        'method' : 'post',
        'contentType': 'application/json',
        'payload' : JSON.stringify(data)
    };
    UrlFetchApp.fetch(urlTarget, options);
    return null;
}
```

Figura 4.4: *sendData*

### getData

Muy parecida a la función anterior, pero en vez de mandar los datos hacia el servidor, se encarga de la recepción de ellos.

```
function getData(urlTarget){
    var fetchURL = urlTarget; //petición de URL
    var response= UrlFetchApp.fetch(fetchURL);
    //DEBUG//Browser.msgBox(response.getContentText());
    return response.getContentText();
}
```

Figura 4.5: *getData*

### Domain

Es la función encargada de definir los posibles valores de las celdas que hemos decidido con anterioridad. Como su nombre indica, es el dominio o rango de valores posibles para las celdas seleccionadas. Para facilitar su utilización, se han configurado distintas ventanas emergentes para guiarnos con el proceso de definición de un dominio y, así minimizar los errores en la escritura manual. Es en ésta función donde se utiliza la variable “numero” como se puede ver en la primera línea de código.

```

function Domain(){
    var first = PropertiesService.getScriptProperties().getProperty('numero');
    first = (first*1)
    var ss = SpreadsheetApp.getActiveSpreadsheet();
    var actual;
    var celdas= ss.getActiveSelection();
    var Domain = "DOMAIN(";
    for (j=1;j<=celdas.getNumRows();j++){
        for (i=1;i<=celdas.getNumColumns();i++){
            actual=celdas.getCell(j,i);
            if (i!=celdas.getNumColumns() || j!=celdas.getNumRows()){
                Domain = Domain + actual.getA1Notation() + ",";
            } else{
                Domain = Domain + actual.getA1Notation() + ")";
            }
        }
    }
    var content = "(";
    var browser = Browser.inputBox("Domain","Introduzca los valores del dominio
    separados por comas", Browser.Buttons.OK_CANCEL);
    content = content + browser + ")";
    var array = browser.split(",");
    var last = (first*1) + (array.length - 1);
    var destino = Browser.inputBox("Destino","Seleccione la celda destino en formato A1",
    Browser.Buttons.OK_CANCEL);
    SpreadsheetApp.getActiveSpreadsheet().setActiveSelection(destino);
    var rango = "(" + first + "," + last + ")";
    SpreadsheetApp.getActiveRange().setValue(Domain + content + rango);
    PropertiesService.getScriptProperties().setProperty("numero", (last*1)+(1*1));
}

```

Figura 4.6: *Domain*

## Alldiff

Es la función que fija que las celdas seleccionadas deben tener valores distintos entre ellas, es decir que todas las celdas deben tener valores distintos de los definidos en el dominio, si esto no se puede cumplir, la aplicación mostrará una ventana indicando que el problema no tiene solución. Para facilitar su utilización, se han configurado distintas ventanas emergentes para guiarnos con el proceso de definición de Alldiff y, así minimizar los errores en la escritura manual.

## 4.2 Descripción detallada del sistema

---

```
function AllDiff(){
  var ss = SpreadsheetApp.getActiveSpreadsheet();
  var celdas= ss.getActiveSelection();
  var actual;
  var AllDiff = 'ALLDIFF(';
  for (j=1;j<=celdas.getNumRows();j++){
    for (i=1;i<=celdas.getNumColumns();i++){
      actual=celdas.getCell(j,i);
      if (i!=celdas.getNumColumns() || j!=celdas.getNumRows()){
        AllDiff = AllDiff + actual.getA1Notation() + ",";
      } else{
        AllDiff = AllDiff + actual.getA1Notation() + ')';
      }
    }
  }
  var destino = Browser.inputBox("AllDiff",
    "Seleccione la celda destino en formato A1", Browser.Buttons.OK_CANCEL);
  SpreadsheetApp.getActiveSpreadsheet().setActiveSelection(destino);
  SpreadsheetApp.getActiveRange().setValue(AllDiff);
}
```

Figura 4.7: *Alldiff*

### Ntimes

El propósito de ésta función es la de fijar un valor del dominio, para que aparezca en la posible solución las veces que se definan. Por ello se nos indicará que indiquemos un valor del dominio y las repeticiones que queremos en la solución. Para facilitar su utilización, se han configurado distintas ventanas emergentes para guiarnos con el proceso de definición de Ntimes y, así minimizar los errores en la escritura manual.

```

function NTimes(){
  var ss = SpreadsheetApp.getActiveSpreadsheet();
  var celdas= ss.getActiveSelection();
  var actual;
  var NTimes = "NTIMES(";
  for (j=1;j<=celdas.getNumRows();j++){
    for (i=1;i<=celdas.getNumColumns();i++){
      actual=celdas.getCell(j,i);
      if (i!=celdas.getNumColumns() || j!=celdas.getNumRows()){
        NTimes = NTimes + actual.getA1Notation() + ",";
      } else{
        NTimes = NTimes + actual.getA1Notation() + ")";
      }
    }
  }
  var content = "(";
  var browser = Browser.inputBox("NTimes","Introduzca el elemento del dominio
  y el numero de repeticiones separados por una coma", Browser.Buttons.OK_CANCEL);
  content = content + browser + ")";
  var destino = Browser.inputBox("Destino","Seleccione la celda destino en formato A1",
  Browser.Buttons.OK_CANCEL);
  SpreadsheetApp.getActiveSpreadsheet().setActiveSelection(destino);
  SpreadsheetApp.getActiveRange().setValue(NTimes + content);
}

```

Figura 4.8: *Ntimes*

## Sudoku

No se trata en sí de una función necesaria para la resolución de algún problema, pero se ha definido para hacer pruebas y así definir de una manera más sencilla uno de los problemas típicos y que, se pueden resolver utilizando el módulo desarrollado en este proyecto. Lo que realiza fundamentalmente, es pintar el esqueleto de un tablero de sudoku en una nueva Spreadsheet llamada “Sudoku”.

## 4.2 Descripción detallada del sistema

---

```
function Sudoku(){
  var ss = SpreadsheetApp.getActiveSpreadsheet();
  var actual;
  var sresult;
  //Creamos la nueva pestaña para la plantilla de sudoku
  if (ss.getSheetByName("Sudoku") == null) {
    sresult=ss.insertSheet("Sudoku");
  } else {
    sresult=ss.getSheetByName("Sudoku");
  }
  var celdas = ss.setActiveSelection("C6:K14");
  Domain();
  for (j=1;j<=celdas.getNumRows();j++){
    for (i=1;i<=celdas.getNumColumns();i++){
      actual=celdas.getCell(j,i);
      actual.setHorizontalAlignment("center");
      actual.setVerticalAlignment("middle");
      //Browser.msgBox(actual.getA1Notation());
      if (i==1){
        actual.setBorder(null, true, null, null, false, false);
      }
      if (i==3 || i==6 || i==9 ){
        actual.setBorder(null, null, null, true, false, false);
      }
      if (j==1){
        actual.setBorder(true, null, null, null, false, false);
      }
      if (j==3 || j==6 || j==9){
        actual.setBorder(null, null, true, null, false, false);
      }
    }
  }
}
```

Figura 4.9: *Sudoku*

### Resolver

Se trata de la función clave ya que, una vez definidas todas las celdas, es la encargada de una vez seleccionadas las celdas que queremos evaluar para la solución, enviarlas con formato JSON al servidor. También se encarga de la recepción de los datos devueltos por el servidor y posterior parseo para que los resultados aparezcan en las celdas correspondientes.

```

function resolver(){
  var ss = SpreadsheetApp.getActiveSpreadsheet();
  var celdas= ss.getActiveSelection();
  var sresult;
  var response;
  if (ss.getSheetByName("Solucion") == null) {
    sresult=ss.insertSheet("Solucion");
  } else {
    ss.getSheetByName("Solucion").clear();
    sresult=ss.getSheetByName("Solucion");
  }
  var resultado; // = SpreadsheetApp.getActiveSheet().setActiveSelection('A1');
  var dos=(celdas.getNumColumns());
  var a= (celdas.getNumRows());
  var yeison;
  var datos="{";
  var actual;
  for (j=1;j<=celdas.getNumRows();j++){
    for (i=1;i<=celdas.getNumColumns();i++){
      actual=celdas.getCell(j,i);
      if (actual.getValue() != ""){
        if (i!=celdas.getNumColumns() || j!=celdas.getNumRows()){
          datos =datos + actual.getA1Notation()+":"+actual.getValue()+"\\";
        } else {
          datos =datos +actual.getA1Notation()+":"+actual.getValue()+"}";
        }
      }
    }
  }
  //Necesario borrar el ultimo caracter de separacion por si la ultima celda es vacia
  if (datos.substring(datos.length-1) == ";"){
    datos = datos.substring(0, datos.length-2);
    datos = datos + "}";
  }
}

```

Figura 4.10: Resolver parte 1

## 4.2 Descripción detallada del sistema

---

```
if (datos.length > 0){
    response =sendData('http://54.194.94.136/', {n:datos});
    response = response.toString();
    response = response.split("\n");
    if (response[0] != 'no sol'){
        //response = response.split("\n");
        for (var z=0 ; z<response.length-1 ; z+=2){
            resultado =ss.setActiveSheet(sresult).setActiveSelection(response[z]);
            var sinigual = response[z+1].split("=");
            resultado.setValue(sinigual[1]);
        }
    }else{
        Browser.msgBox("El problema no tiene solucion");
    }
}else {
    var nada=sendDataNoReturn('http://54.194.94.136/', {n:datos});
    Browser.msgBox("Su solucion se esta calculando");
    simple();
}
```

Figura 4.11: Resolver parte 2

### 4.2.2. Servidor

Para la realización del proyecto se ha contratado un servidor en AWS “Amazon Web Services“. Se ha realizado todo el proyecto con el servidor ofrecido más pequeño ya que era gratis durante el primer año. Se trata de un servidor Linux Centos de 64 bits en el que se han realizado las siguientes configuraciones:

- *Apache*. Se ha configurado para la recepción de los datos.
- *Java* Necesario para la ejecución del código empaquetado en un “.Jar“.
- *Index.php*. Punto de entrada al servidor, donde se gestiona el flujo y traspaso de los datos dentro del servidor hasta la devolución de la salida.
- *CiaoDE*. Lenguaje necesario para la ejecución del código que prepara el .Jar con formato .pl y que sirve de programa prolog a ejecutar. Se ha instalado la versión 1.15 para poder incluir la librería “CLP(FD)“.
- *Seguridad*. Para tener control sobre los usuarios que pueden acceder al servidor, se ha generado una clave cliente-servidor para limitar los equipos que pueden acceder por



SSH.

### Flujo de datos

Conviene explicar el flujo de datos del servidor ya que, al pasar de unos lenguajes a otros dificulta realizar un seguimiento sencillo de los datos dentro del servidor.

1. *Entrada.* Los datos llegan al servidor encapsulados en formato JSON, éstos son recibidos en el fichero `index.php`, que se encarga de recogerlos en una variable llamada “`n`” y se los pasa como parámetro al `.Jar`.
2. *Escenario.* Una vez los datos son procesados por el `.Jar` (este proceso se definirá en más profundidad en el siguiente capítulo), se genera un fichero denominado “`escenario.pl`” el cual está ya preparado para ser ejecutado por `CiaoDE`.
3. *Salida.* Una vez evaluado el fichero `escenario.pl` se generará otro fichero llamado “`Salida.txt`” en el que tendremos las celdas y sus valores correspondientes, o en su defecto, la frase “`no sol`” indicando que el fichero generado `escenario.pl` no tiene solución.

Todo lo anterior lo podemos ver, aparte de otras configuraciones como salidas de logs, viendo el contenido del fichero `index.php`, que se muestra a continuación:

```
?php
    $json=json_decode(file_get_contents('php://input'),true);
    error_log("json ok");
    error_log(var_export($json["n"],true) );

    $comando = "java -jar Spreadsolver_V5.jar " . $json["n"];
    error_log("comando ok: ".$comando);
    header('Content-Type: application/json');
    $salida = shell_exec($comando);
    echo shell_exec('cat salida.txt');
    error_log($salida);
```

Figura 4.12: *Index.php*

## 4.2 Descripción detallada del sistema

---

### 4.2.3. Código Java

Esta parte del proyecto ha sido desarrollada íntegramente en una máquina virtual creada en “VMware Workstation“, concretamente en un Ubuntu 15.02 para poder recrear la ejecución del código java en un entorno Linux. La herramienta de desarrollo utilizada ha sido eclipse neon:



Figura 4.13: *Eclipse Neon*

La parte más compleja ha sido la de “recortar“ la parte de la interfaz que estaba definida en el código recibido, para lo cual han sido necesarios varios cambios dentro de distintas clases, aparte de por supuesto en el main, que simplemente iniciaba el interfaz de la aplicación.

#### Clases Modificadas

- *Main*. Se ha tenido que definir completamente desde cero para realizar el parseo de los datos recibidos por index.php, ya que en el modelo anterior, los datos de entrada los iba insertando el usuario en la interfaz generada por la misma aplicación.

```

import interfaz.*;

public class Main {

    public static void main(String[] args) {
        MyTableModel TableModel = new MyTableModel(100,100);
        String yeison = args[0];
        String igual = "=";
        yeison = yeison.substring(1,yeison.length()-1);
        String[] yeisonarray = yeison.split(";");

        for (int i =0; i < yeisonarray.length ; i++) {
            String[] array = yeisonarray[i].split(":");
            Celda casilla = new Celda (array[0]);
            if ((array[1].startsWith("ALLDIFF")) || (array[1].startsWith("DOMAIN"))
                || (array[1].startsWith("NTIMES")) ){
                String contenido = array[1];
                TableModel.setValueAt(contenido, casilla.getColumna(),
                    casilla.getFila() );
            }else {
                String contenido = array[1];
                TableModel.setValueAt(igual + contenido, casilla.getColumna(),
                    casilla.getFila() );
            }
        }
        TableModel.resolverEscenario();
    }
}

```

Figura 4.14: *Main*

- *Celda*. Se ha tenido que modificar esta clase para incluir un constructor más, ya que el predefinido como:

```
public Celda(int fila, int columna)
```

No sirve para los nuevos datos que nos llegan en el JSON, por eso se ha definido el método:

```
public Celda (String A1)
```

Capaz de crear las celdas en el formato “A1” que es como vienen recogidos de la Spreadsheet de Google.

- *MyTablemodel*. En esta clase los principales cambios han sido la declaración de varias funciones que, originalmente estaban definidas como “private” y, al tener que ser

## 4.2 Descripción detallada del sistema

---

llamadas por la clase Main, han tenido que ser redefinidas como “public”.

```
public void setValueAt(Object value, int row, int col)
public void resolverEscenario()
```

### Ejecución

Como ya se ha podido ver en la imagen anterior del contenido del fichero index.php, la manera de ejecutar el código .Jar generado con todas las clases originales, incluyendo las modificaciones descritas, es simplemente:

```
java -jar Spreadsolver.jar
```

Para la ejecución del fichero escenario.pl por parte de CiaoDE, está definida en la clase “Mytablemodel” la función resolverEscenario que, como código relevante en la ejecución del fichero, tiene las líneas:

```
Process p = Runtime.getRuntime().exec ("ciao escenario");
p.waitFor();
p = Runtime.getRuntime().exec ("./escenario");
p.waitFor();
```

Figura 4.15: *Ejecución escenario.pl*



## RESULTADOS Y CONCLUSIONES

Por último, en este capítulo vamos a realizar un análisis sobre la concordancia entre los resultados obtenidos y los objetivos establecidos al principio del proyecto, las fortalezas y carencias del mismo, un resumen de las futuras líneas de trabajo y la presentación de las conclusiones que han derivado de la realización del proyecto.

### 5.1. Resultados

El objetivo principal era integrar un módulo para la Spreadsheet de Google autocontenido que, añadiera diversas funciones relacionadas con operaciones basadas en dominios finitos. El cual se ha cumplido a la perfección, aunque, como ya hemos mencionado, se puede ampliar incorporando diversas funcionalidades o mejoras.

Respecto a los objetivos personales, destacar brevemente el aumento de conocimientos sobre la creación y funcionalidad desde el punto de vista del servidor y la ampliación de conocimientos sobre el envío o recepción de parámetros encapsulados en JSON.

Respecto a los objetivos del proyecto, se ha conseguido la comunicación desde la Spreadsheet, ubicada en el Drive de Google, hacia el servidor alojado en internet, por lo que el objetivo de movilidad se ha conseguido. También se ha logrado, gracias a la estructura de la plataforma, que esta funcionalidad esté disponible desde cualquier tipo de dispositivo, con el único requisito de la necesidad de conectividad con internet. Para la definición y usabilidad de las funciones integradas, se han puesto ventanas emergentes para facilitar la experiencia del usuario, haciendo que aunque no se conozca el funciona-

miento se pueda utilizar de una manera sencilla e intuitiva. El código está estructurado y es mantenible gracias a que está adecuadamente comentado y dividido en diferentes clases y funciones. Por los mismos motivos, la aplicación es extendible y escalable tal y como se esperaba.

### 5.2. Fortalezas y mejoras

Las fortalezas del módulo integrado son las siguientes:

- Envío y recepción de datos de forma segura definidos en formato JSON.
- Cada una de las piezas que componen el proyecto son independientes por lo que es sencillo identificar algún problema.
- Tanto en el front end como en el back end se han definido logs para tener una mayor trazabilidad del funcionamiento del conjunto del módulo.

En cambio, las posibles mejoras son las siguientes:

- Ampliar la definición de funciones en CiaoDE para tener una mayor versatilidad para la resolución de problemas.
- Módulo integrado no portable a otras plataformas, aunque el back end al ser independiente sí podría ser reutilizado.

### 5.3. Futuras líneas de investigación

No siendo parte de los objetivos principales, una línea clara de evolución del proyecto sería la creación de un add-on para poder añadirlo al listado de aplicaciones disponibles para la Spreadsheet de Google. Éste desarrollo podría tener también carácter comercial por la posibilidad de adquisición del módulo por parte de Google o, si se ve una clara necesidad por parte de los usuarios, la creación de una plataforma dedicada a dar servicio al módulo desarrollado.

### 5.4. Conclusiones

Después de terminar la integración y de analizar los resultados obtenidos, se obtienen las conclusiones siguientes:

En primer lugar, destacar que, como ya hemos visto, todos los objetivos propuestos han sido cumplidos. Tanto los referentes al proyecto como los personales.

Aunque el desarrollo del proyecto y, en concreto, la unificación de los parámetros en las distintas plataformas ha sido lento, el resultado ha sido muy bueno. Se ha aprendido y profundizado sobre la programación en diversos sistemas, la implementación de funciones y la integración de un módulo para el Spreadsheet de Google.

Desarrollando este proyecto he investigado, disfrutado, sufrido y aprendido mucho sobre un campo apasionante y muy extenso, el cual era desconocido a este nivel de profundidad para mí.





# Manual de Usuario

## 6.1. Introducción

El módulo de dominios finitos permite añadir unas funcionalidades a la Spreadsheet de Google relacionadas con la evaluación de condiciones simples. El acceso a éste módulo es totalmente gratuito, aunque todavía no está disponible como add-on para incorporar a nuestra hoja de cálculo.

## 6.2. Requisitos

Los requerimientos mínimos para que podamos utilizar este módulo son los siguientes:

- Tener vigente una cuenta de Google (gmail).
- Conexión a internet para el cálculo de las posibles soluciones.
- Abrir una nueva hoja de cálculo e importar el módulo para poder utilizar sus funciones.

## 6.3. Instalación

Para poder utilizar el módulo de dominios finitos basta con realizar los siguientes pasos:

1. Empezamos con hacer click en el menú “Add-ons” y pinchar en “Get add-ons...”.

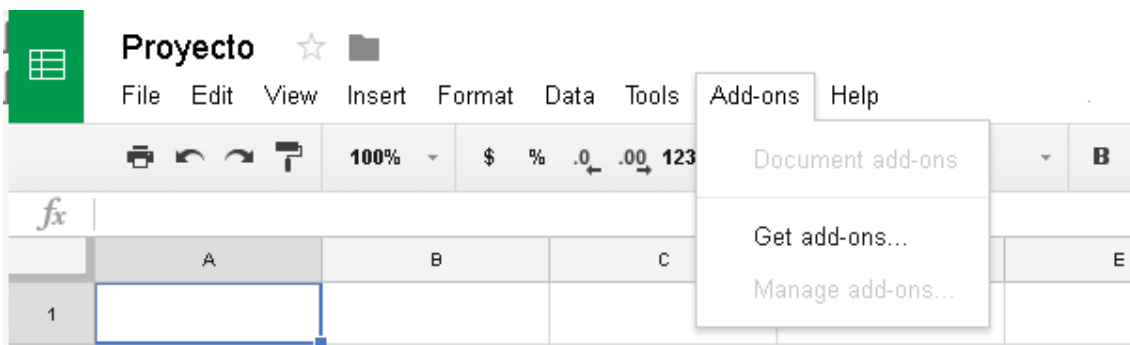


Figura 6.1: *Buscar módulo*

2. Después en el buscador ponemos “finite domains” y pulsamos el botón “free”.

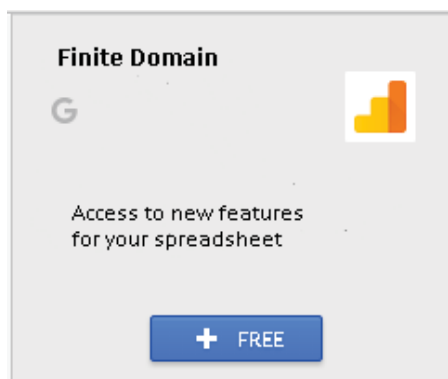


Figura 6.2: *Añadir módulo*

3. Finalmente recargamos la página pulsando F5 y nos deberá salir una nueva pestaña. Esto indica que el módulo se ha importado correctamente.

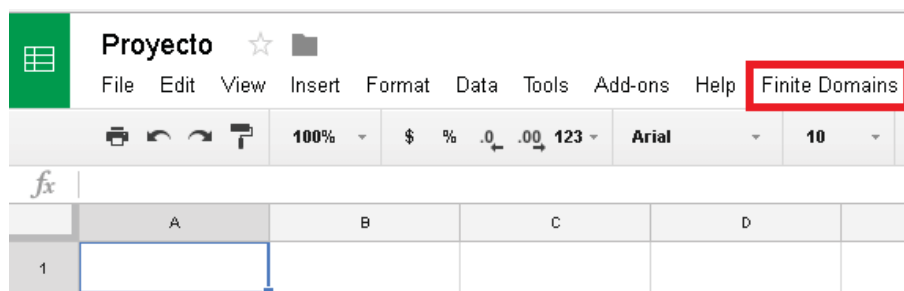


Figura 6.3: *Módulo instalado*

### 6.4. Uso de la aplicación

#### 6.4.1. Arranque de la aplicación

Cada vez que abramos una hoja de cálculo se cargará por defecto el módulo instalado de dominios finitos.

#### 6.4.2. Funciones

En esta sección vamos a explicar las distintas funciones y el uso detallado de cada una de ellas. Se han configurado cada una de las opciones para que además de ser intuitivas nos vayan guiando con mensajes por pantalla para un mejor entendimiento y fácil configuración de las reglas que vamos a utilizar.

#### Domain

La función “Domain“ nos permite definir el rango de valores que van a poder tomar un determinado número de celdas que definamos. La definición de dominios puede ser múltiple siempre y cuando no haya celdas comunes definidas en ambos dominios. Para utilizar esta función seguiremos los siguientes pasos:

1. Seleccionar las celdas que van a formar parte del dominio a definir.
2. Hacer click sobre la pestaña “Finite Domains“ y seleccionar la función Domain.

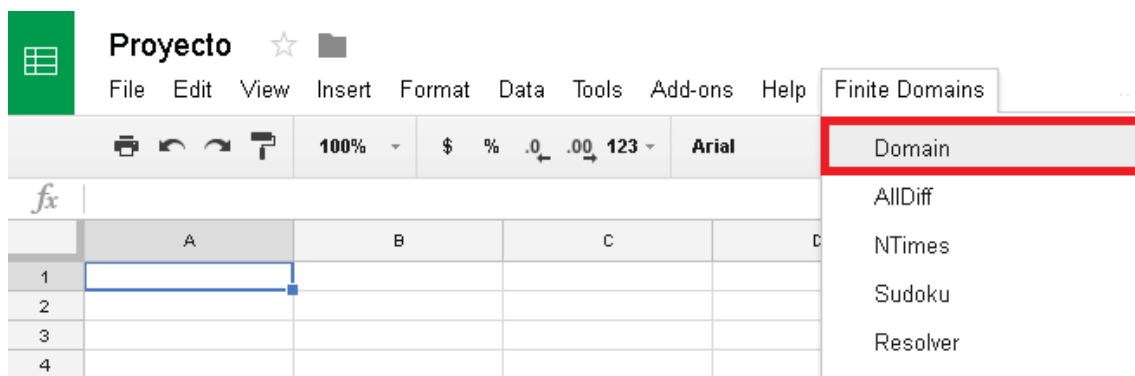


Figura 6.4: *Domain 1*

3. Saldrá una ventana emergente indicando que se introduzcan los valores del dominio deseados. En el ejemplo vemos que hemos definido un dominio con los valores “rojo, verde, azul, amarillo“. Para finalizar pulsar el botón OK.

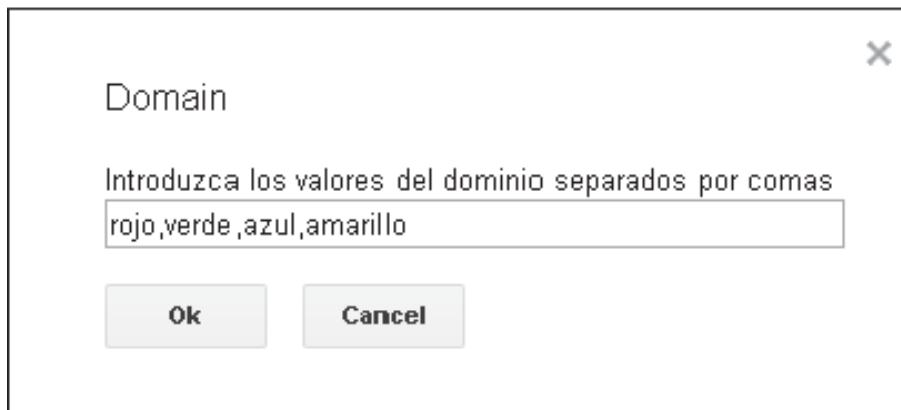


Figura 6.5: *Domain 2*

4. Saldrá una segunda ventana emergente preguntando en qué celda hay que insertar la condición. És necesario insertar la celda como se indica en formato A1 para un correcto funcionamiento y aplicación de la condición antes indicada. En el ejemplo se ha decidido poner como celda de destino la “A1“. Para finalizar pulsar el botón OK.

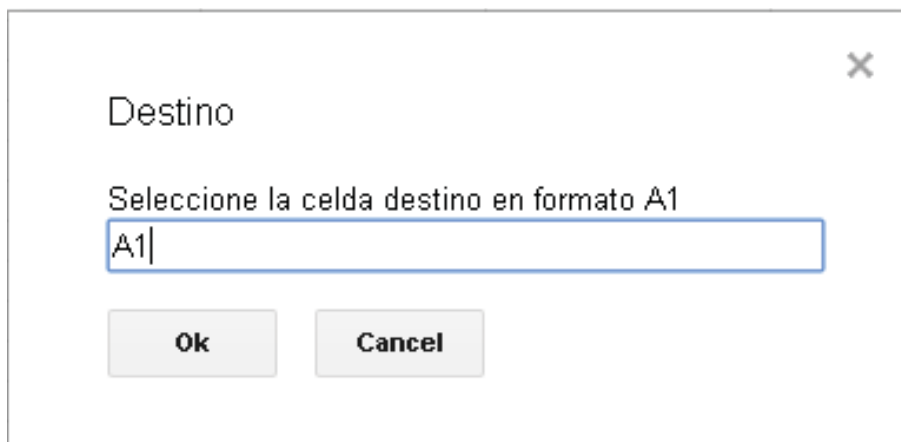


Figura 6.6: *Domain 3*

## 6.4 Uso de la aplicación

5. Finalmente aparecerá en la celda especificada anteriormente la condición completa creada según los parámetros que se han definido. En el ejemplo especificado según los pasos anteriores será:

DOMAIN(Lista de celdas seleccionadas)(rojo,verde,azul,amarillo)(1,4)

El último rango que aparece entre paréntesis corresponde con el número de elementos del dominio que se han definido. Esto es para que internamente se pueda asignar un valor a la celda para el cálculo de la solución. En el caso especificado será:

- Rojo(1)
- Verde(2)
- Azul(3)
- Amarillo(4)

### AllDiff

La función “AllDiff” se usa para definir un rango de celdas, pertenecientes a un dominio, en las cuales todos los valores deben de ser diferentes. Para utilizar esta función seguiremos los siguientes pasos:

1. Seleccionar las celdas que van cumplir la condición que se va a definir.
2. Hacer click sobre la pestaña “Finite Domains” y seleccionar la función AllDiff.

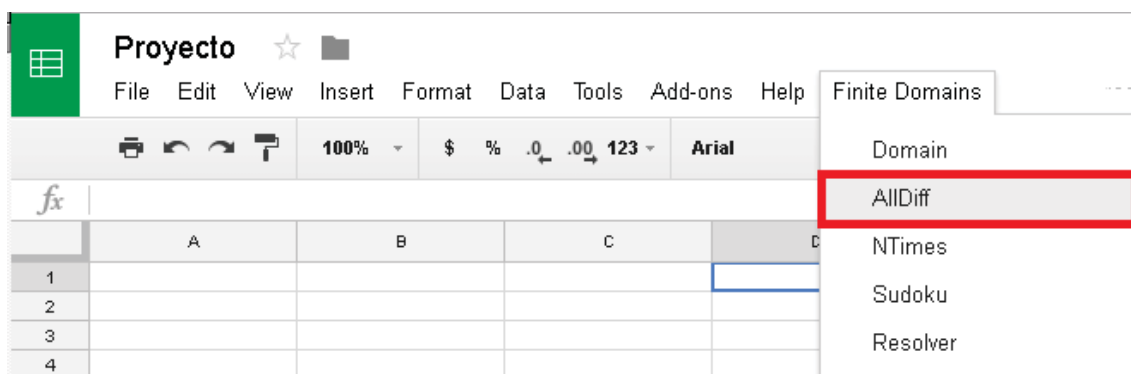


Figura 6.7: AllDiff 1

3. Saldrá una ventana emergente preguntando en qué celda hay que insertar la condición. Es necesario insertar la celda como se indica en formato A1 para un correcto funcionamiento y aplicación de la condición antes indicada. En el ejemplo se ha decidido poner como celda de destino la “A1”. Para finalizar pulsar el botón OK.

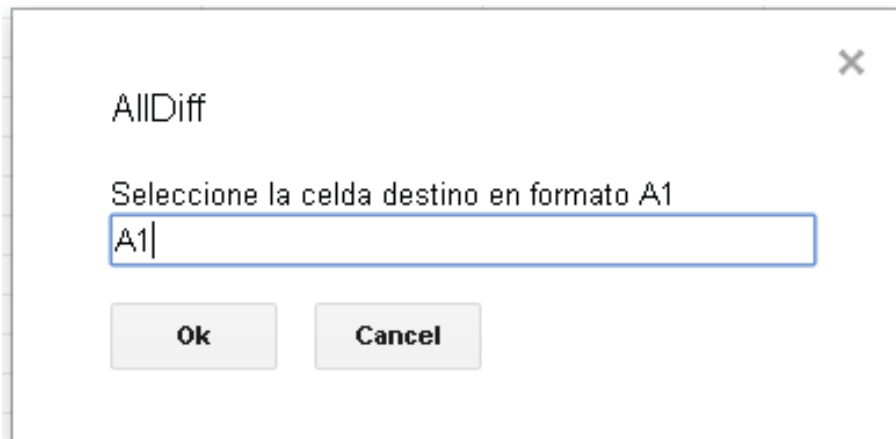


Figura 6.8: *AllDiff 2*

4. Finalmente aparecerá en la celda especificada anteriormente la condición completa creada según los parámetros que se han definido. En el ejemplo especificado según los pasos anteriores será:  
ALLDIFF(Lista de celdas seleccionadas)

### Ntimes

La función “Ntimes” sirve para definir una condición, en la cual un valor perteneciente a un dominio definido, tiene que aparecer “X” veces en la posible solución del problema. Para utilizar esta función seguiremos los siguientes pasos:

1. Seleccionar las celdas que van a cumplir la condición que se va a definir.
2. Hacer click sobre la pestaña “Finite Domains” y seleccionar la función NTimes.

## 6.4 Uso de la aplicación

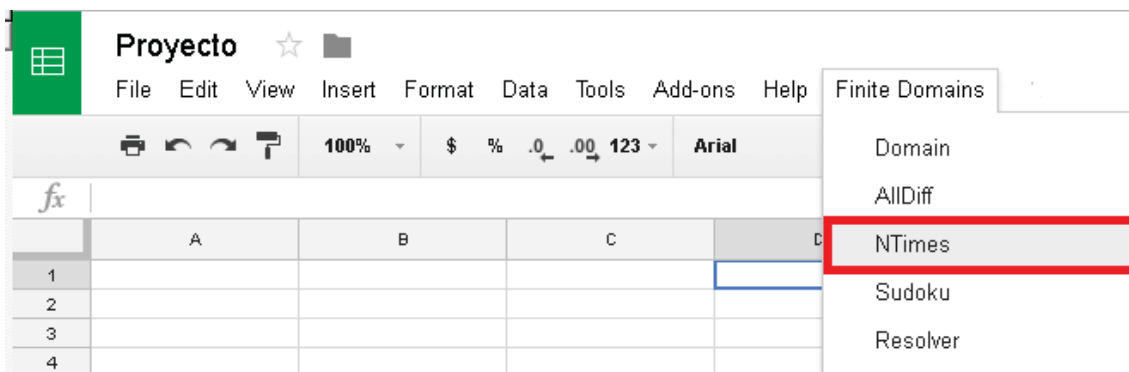


Figura 6.9: *NTimes 1*

3. Saldrá una ventana emergente indicando que se introduzca el valor del dominio y el número de repeticiones deseadas. En el ejemplo vemos que hemos definido que el valor rojo debe aparecer 3 veces en la solución. Para finalizar pulsar el botón OK.

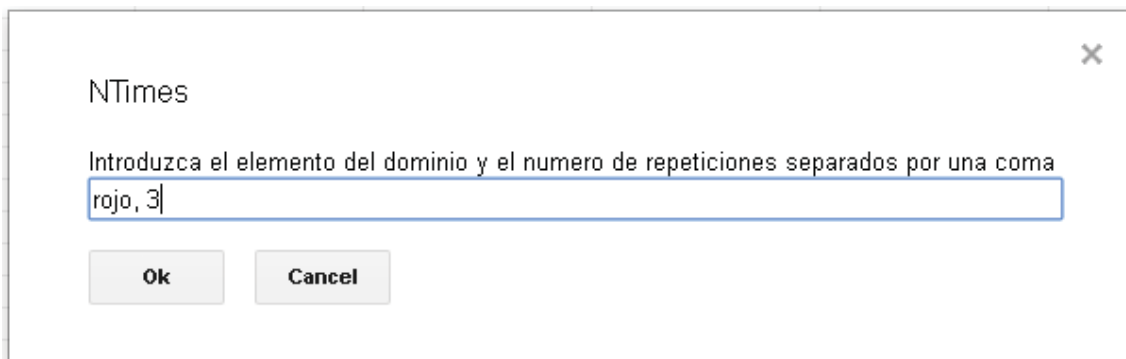


Figura 6.10: *NTimes 2*

4. Saldrá una segunda ventana emergente preguntando en qué celda hay que insertar la condición. És necesario insertar la celda como se indica en formato A1 para un correcto funcionamiento y aplicación de la condición antes indicada. En el ejemplo se ha decidido poner como celda de destino la "A1". Para finalizar pulsar el botón OK.



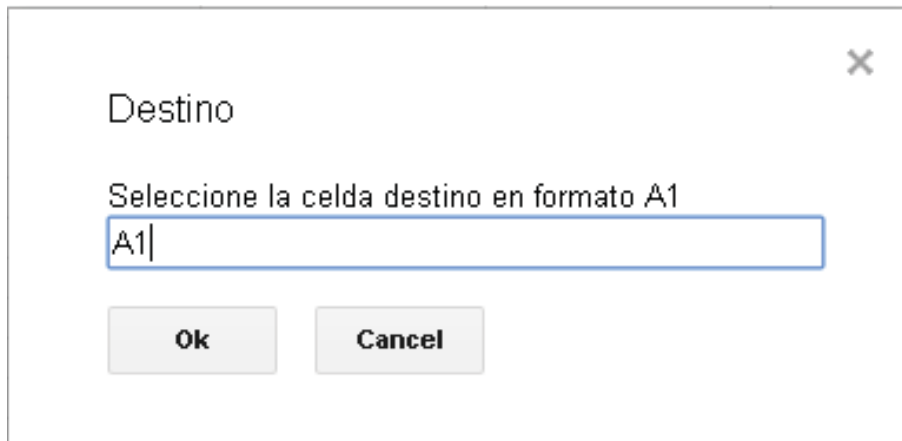


Figura 6.11: *NTimes 3*

5. Finalmente aparecerá en la celda especificada anteriormente la condición completa creada según los parámetros que se han definido. En el ejemplo especificado según los pasos anteriores será:

`NTIMES(Lista de celdas seleccionadas)(rojo, 3)`

### Resolver

La función “Resolver” es la encargada de realizar los cálculos necesarios para encontrar una posible solución tomando diversas condiciones como parámetros de entrada. El resultado se mostrará en una pestaña nueva llamada “Solucion” o, si por el contrario, no existe solución se mostrará una ventana emergente mostrando que el problema con las condiciones seleccionadas no se puede resolver. Para utilizar esta función seguiremos los siguientes pasos:

1. Seleccionar las celdas que se van a tener en cuenta como parámetros de entrada, ya sean condiciones o celdas con elementos fijos.
2. Hacer click sobre la pestaña “Finite Domains” y seleccionar la función Resolver.

## 6.4 Uso de la aplicación

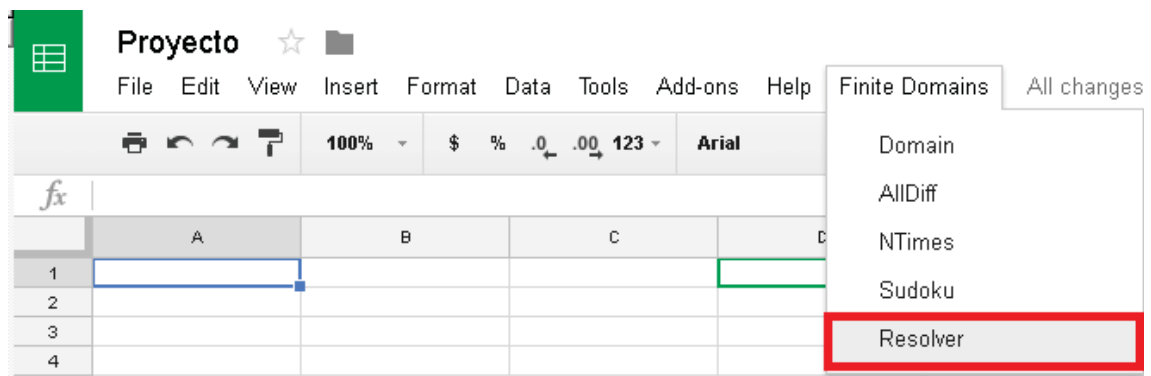


Figura 6.12: *Resolver 1*

3. Esperar a que el programa sea ejecutado, aparecerá un mensaje en la parte superior indicando que el programa está en ejecución.



Figura 6.13: *Resolver 2*

### Sudoku

No se trata de una función en sí, se trata de una plantilla que al ejecutarse crea una nueva hoja llamada "Sudoku" con el clásico tablero de 9x9 dibujado, se ha realizado para pruebas mas que para cualquier utilidad definitiva asociada al proyecto.

### 6.4.3. Resultados

Aquí se explican la forma y manera en la que se muestran los resultados generados por la función "Resolver", explicada en la sección anterior de Funciones.

### Destino

Para facilitar y poder realizar múltiples pruebas con los datos de entrada, se ha configurado para que los datos resultantes se escriban en una hoja de cálculo nueva llamada

"Solucion". De esta manera se escribirán, tanto las condiciones que hemos seleccionado para la resolución como el resto de celdas, que servirán para cálculo de la posible solución a nuestro problema planteado. La solución se escribe en esta nueva hoja tanto si existe como si no, es decir, si no se ha realizado ninguna operación previa se creará automáticamente para así poder escribir la solución.

- *PRECAUCIÓN* Hay que confirmar antes de llamar a la función "Resolver" que ninguna de nuestras hojas se llama "Solucion", ya que en ese caso se sobrescribirán las celdas que se vayan a devolver como solución.

### Datos

Los datos, que se van a escribir en la hoja solución, como se ha explicado anteriormente constan de:

- *Condiciones* Las funciones definidas por el usuario como: Domain, ntimes o alldiff.
- *Celdas* Las celdas con valores fijos en nuestro sistema. P.j:  $A1 = 5$ .

Los datos que se escribirán lo harán en la misma celda que lo harían, según hemos definido el dominio y las condiciones en nuestra hoja, donde hemos seleccionado las celdas para la función "Resolver". Ésto se ha realizado para no alterar los datos de entrada y, así poder tener más versatilidad para el cálculo de distintas soluciones con variables de entrada distintas.

## Bibliografía

- <http://angelicathaliasalinasdiaz.blogspot.com.es/2011/10/versiones-de-la-hoja-de-calculo.html>
- <https://es.wikipedia.org/wiki/MicrosoftExcel>
- <http://minisconlatex.blogspot.com.es/>
- <https://es.wikipedia.org/wiki/QuattroProOr.C3.ADgenesehistoria>
- <https://developer.mozilla.org/es/docs/Web/JavaScript>
- <https://es.wikipedia.org/wiki/LibreOfficeCalc>
- <http://www.sciencedirect.com/science/article/pii/S1571066111001939>
- <https://es.wikipedia.org/wiki/ApacheOpenOfficeCalc>
- M. Kassoff, A. Valente An introduction to logical spreadsheets Knowledge Eng. Review (2007), pp. 213-219
- <https://es.wikipedia.org/wiki/Hojadec>
- <https://ciao-lang.org/>
- <https://es.wikipedia.org/wiki/MecanismodeAnticitera>
- <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/SSL-on-an-instance.html>
- <https://es.wikipedia.org/wiki/>

- <https://es.sharelatex.com/learn/MainPage>
- <https://es.wikipedia.org/wiki/VisiCalc>
- <https://developers.google.com/apps-script/reference/spreadsheet/>
- <https://es.wikipedia.org/wiki/GoogleSpreadsheets>
- <https://es.wikipedia.org/wiki/JavaScript>