



UNIVERSIDAD DE BURGOS

**Máster Universitario en Profesor de Educación Secundaria Obligatoria y
Bachillerato, Formación Profesional y Enseñanza de Idiomas.**

TRABAJO FIN DE MÁSTER

**USO DE EXCEL PARA LA ENSEÑANZA DE MATEMÁTICAS EN
BACHILLERATO Y TALLERES DE INICIO A LA PROGRAMACIÓN.**

Curso 2018-2019

ALUMNO: Miguel Ángel Bárcena Pascual

ESPECIALIDAD: Matemáticas DIRECTOR:

Antonio Manzano Rodríguez

Contenido

1 - INTRODUCCIÓN	4
1.1 - PLANTEAMIENTO DEL PROBLEMA	4
1.2 - OBJETIVOS.....	9
1.3 - REVISIÓN BIBLIOGRÁFICA.....	10
2 - METODOLOGÍA	10
3 - PROFUNDIZACIÓN.....	12
3.1 - GEOGEBRA	12
3.2 - MAPLE.....	13
3.3 - DERIVE	14
3.4 - GEUP.....	15
3.5 - DESCARTES.....	16
3.6 - HOJAS DE CÁLCULO EXCEL.....	17
4 - MATERIAL PROPUESTO PARA LA DOCENCIA CON MICROSOFT EXCEL	18
4.1 - HOJAS DE CÁLCULO CON MACROS	19
4.2 - TALLERES DE PROGRAMACIÓN	20
5 - SESIÓN DE USO DE LA HOJA EXCEL PARA TRABAJAR CON CONTENIDOS DEL BLOQUE II DEL CURRÍCULO.....	22
5.1 - CÁLCULO DE LA INVERSA DE UNA MATRIZ.....	22
5.2 - REPASO DE RESOLUCIÓN DE SISTEMAS DE ECUACIONES	25
6 - SESIÓN DE USO DE LA HOJA DE EXCEL PARA TRABAJAR CON CONTENIDOS DEL BLOQUE IV DEL CURRÍCULO	30
6.1 - POSICIÓN RELATIVA EN EL ESPACIO DE DOS RECTAS, DE UNA RECTA Y UN PLANO, DE DOS Y DE TRES PLANOS.....	30
7 - SESIÓN DE USO DE LA HOJA EXCEL PARA TRABAJAR CON CONTENIDOS DEL BLOQUE III DEL CURRÍCULO.....	34
7.1 - CÁLCULO DE INTEGRALES DEFINIDAS	34
8 - TALLERES DE PROGRAMACIÓN.....	39
8.1 - TALLER Nº 1: INTRODUCCIÓN A LA PROGRAMACIÓN EN VBA	40
8.1.1 - Macros	40
8.1.2 - Variables	43
8.1.3 - Actividades.....	45
8.1.4 - Operadores	46
8.2 - TALLER Nº 2: COMANDOS HABITUALES EN VBA.....	47

8.3 - TALLER Nº 3: INSTRUCCIONES DE BUCLE Y CONDICIONALES EN PROGRAMACIÓN	53
8.3.1 - Instrucción IF	54
8.3.2 - Instrucción DO WHILE (bucle) y uso de variables contador.....	55
8.3.3 - Instrucción bucle FOR	57
8.3.4 - Operadores adicionales	58
8.4 - TALLER Nº 4: OBJETOS VBA EN EXCEL	62
8.4.1 - Botones	63
8.4.2 - Listas	64
8.4.3 - Checkboxes.....	69
8.4.4 - Botones de opción.....	71
8.5 - TALLER Nº 5: PRACTICUM.....	74
8.6 - TALLER Nº 6: REVISIÓN DE LOS PROGRAMAS EXCEL USADOS EN CLASES DE MATEMÁTICAS.....	79
9 - CONCLUSIONES Y LÍNEAS DE INVESTIGACIÓN FUTURAS	81
10 - Bibliografía.....	84
11 - ANEXO I: Encuesta a los docentes.....	86
11.1 - Test para los docentes	86
11.2 - Entrevista abierta	86
12 - ANEXO II: Test a los alumnos	87
12.1 - Test a los alumnos participantes en los talleres de programación	87
12.2 - Test para todos los alumnos	87

1 - INTRODUCCIÓN

Toda persona cercana al ámbito educativo actual es consciente del incremento que está teniendo la utilización de las TIC (Tecnologías de la Información y la Comunicación) en la docencia. Así, en nuestros días, es frecuente escuchar, cada vez en mayor medida, palabras como Geogebra, WxMaxima, Maple, Derive, etc. como software empleado en el aula. Otro ejemplo podría ser el uso de plataformas virtuales como Moodle, que permite gestionar una asignatura. También es cada vez más generalizado el uso de recursos como Powerpoint, para la presentación de trabajos, por parte de los alumnos.

Es indudable que la mayoría de las TIC proporcionan innumerables mejoras y ventajas en la enseñanza, sobre todo en lo que se refiere a facilitar el trabajo de profesores y alumnos. Sin embargo, una pregunta natural que podemos hacernos es: ¿Son estas herramientas un mero apoyo o ayuda al aprendizaje? ¿O, por el contrario, son herramientas de cuyos conocimientos el alumno se beneficiará una vez finalice su etapa educativa?

En este sentido, parece evidente que, de entre todas las herramientas relacionadas con las TIC que pueden usarse como recurso didáctico, serán especialmente útiles aquellas tales que, además de ayudar a asimilar conocimientos de una determinada materia, lo aprendido sobre la propia herramienta resulte de utilidad después de finalizar la etapa educativa.

En este Trabajo Fin de Máster (TFM) estamos interesados en esta cuestión, que será desarrollada en las secciones siguientes en el contexto de la enseñanza de las Matemáticas.

1.1 - PLANTEAMIENTO DEL PROBLEMA

Los estudiantes que podemos encontrar actualmente en nuestros centros de enseñanza tienen, desde su más temprana edad, un fácil acceso a las TIC. Lo hacen, por ejemplo, con su móvil al chatear con sus amigos, al subir una foto a cualquier red social para compartir su último viaje, o al compartir una noticia que les ha llamado la atención. Es lógico, entonces, esperar que el uso de las TIC termine extendiéndose y generalizándose también en la enseñanza.

Como todos sabemos este hecho ya está ocurriendo. A lo largo de mi paso por centros de enseñanza como alumno y como profesor en prácticas, he podido observar el gran uso de aplicaciones y plataformas que hay actualmente, sobre todo en el ámbito de la enseñanza de las matemáticas.

Este suceso tiende de manera natural a verse incrementado, ya que como afirma (Arias Cabezas & Maza Sáenz, 2006), al realizarse una correcta integración de las TIC en las clases de matemáticas se consigue una mejora en la docencia de esta materia, lo cual es valorado muy positivamente tanto por alumnos como por docentes, que consideran el empleo de las TIC como una metodología eficaz y satisfactoria en cualquier nivel educativo.

También podemos encontrar afirmaciones como la de (Aguilera Almaguer, Aguilera Borjas, & Peña Cruz, 2011), que mencionan que la inclusión de las TIC en la enseñanza, aparte de ser un tema de gran importancia, es algo que ha sido intensamente investigado, con gran número de trabajos que nos permiten conocer su impacto e influencia en la enseñanza.

En esta memoria se pretende ir más allá de dichas afirmaciones. Ahora que hemos dado por sentado no sólo la importancia de las TIC en educación, sino también su implantación actual en el aula, debemos preguntarnos si se está realizando con las herramientas más adecuadas en la asignatura de matemáticas.

Normalmente, la mayor parte de las herramientas tecnológicas empleadas para facilitar el aprendizaje en matemáticas son usadas únicamente durante la etapa educativa del alumno. Así, el docente suele utilizarlas para explicar y transmitir contenidos y el estudiante para entender y aprender éstos. Tras finalizar el proceso de enseñanza-aprendizaje, puede decirse que el alumno ha adquirido dos tipos de conocimientos. El primero, sobre el uso de dicha herramienta, ya que el alumno debe adquirir la destreza de manejarla para que este elemento le ayude en sus estudios. Y el segundo, el propio conocimiento matemático de los contenidos de la asignatura, al cual se accede más fácilmente gracias a la herramienta o aplicación usada.

No obstante, el conocimiento sobre el uso de dicha herramienta tiende a desaparecer en el futuro, debido a que el alumnado no dispone de ninguna razón para seguir empleando esa herramienta. Se podría afirmar que el alumno ha adquirido una

formación sobre algo que posteriormente no va a aprovechar. Esto no sucede con el conocimiento matemático adquirido, ya que éste suele ser usado en muchas situaciones de la vida diaria, por lo que es conservado una vez adquirido debido a su utilidad.

Es lógico plantearse buscar entonces una herramienta que no sólo ayude a adquirir conocimientos matemáticos al alumno, sino que le sea útil posteriormente. Con ello nos estamos refiriendo sobre todo al ámbito laboral, pero esto puede ser extensible a muchos otros ámbitos. Dicho de otro modo, pensamos que es interesante contar con una herramienta que no sólo le sirva para adquirir conocimientos, sino que le resulte útil también una vez realizado el aprendizaje.

Por todo lo anteriormente expuesto, proponemos el uso de una herramienta alternativa a las clásicas usadas para enseñar matemáticas mediante las TIC. En concreto, proponemos la utilización de Microsoft Excel, que forma parte del paquete de aplicaciones conocido como Microsoft Office. Dicho paquete es distribuido por la empresa Microsoft.

En relación con Microsoft Excel nos gustaría indicar que es de los programas más empleados debido a las siguientes razones:

-Es una herramienta que puede emplearse en multitud de ámbitos profesionales (además del educativo).

-Es increíblemente versátil, ya que permite la introducción sencilla de grandes cantidades de datos para, posteriormente, proceder a su manejo.

-La forma que tiene Excel de organizar los datos con una estructura de filas y columnas favorece el análisis de los datos del problema por parte del alumno. Esto es debido, entre otras muchas razones, a la necesidad del alumno de traducir adecuadamente los datos en información para que pueda ser gestionada por el programa (Pifarré Turmo & Sanuy Burgués, 2000).

-En la mayoría de computadoras, tanto a nivel usuario, como a nivel profesional, es posible instalar el paquete Office de Microsoft, por lo que el uso de la herramienta Excel se encuentra disponible en la mayoría de ocasiones.

Por otra parte, además de todo lo anterior, Microsoft Excel incorpora un espacio de trabajo basado en programación mediante lenguaje Visual Basic (VBA), lo cual lo convierte, si cabe, en una herramienta más potente, y con mayores aplicaciones a la docencia.

Sin embargo, a pesar de todas estas ventajas de Microsoft Excel es un hecho según (Iranzo & Fortuny, 2009) que la gran parte de los alumnos usan herramientas como Geogebra, porque creen que este software les ayuda de una manera más sencilla en un problema, permitiéndoles además visualizar claramente su solución.

Como se ha mencionado anteriormente, las TIC ``han venido para quedarse'', y esto es aplicable a todos los ámbitos. Debido al uso incesante y creciente que nuestra sociedad actual hace de la tecnología e Internet, cada vez más personas terminan desarrollándose laboralmente en un ámbito que, de una u otra manera, se encuentra íntimamente relacionado con la programación.

Asimismo, cada vez un mayor número de estudios posteriores a las etapas de E.S.O. o Bachillerato imparten dentro de su itinerario educativo conocimientos de programación informática en uno o varios lenguajes. Sin embargo, los conocimientos de programación impartidos antes de llegar a etapas educativas superiores son muy reducidos.

Poco a poco se está empezando a cambiar esta situación, ya que aprender programación trae consigo múltiples ventajas, lo cual es corroborado por (Cabrera Delgado, 2015) donde nos indica que no sólo la robótica sino también la programación están haciéndose un hueco en las enseñanzas básicas, tanto en Europa como en España, ya que estas materias ayudan a mejorar la adquisición de las competencias básicas. Por no hablar de que también potencian su creatividad, motivación, autonomía, además de ayudarles a desarrollar estrategias para la resolución de problemas.

No obstante, como se indica en la encuesta del INTEF (Instituto Nacional de Tecnologías Educativas y Formación del Profesorado), realizada desde el 13 de diciembre de 2017 hasta el 14 de enero de 2018, el número de horas dedicada a programación entre los niveles de Infantil, Primaria, E.S.O. y Bachillerato es aún bastante reducido, según se desprende de las respuestas dadas por los 351 docentes a los que se les realizó la encuesta.

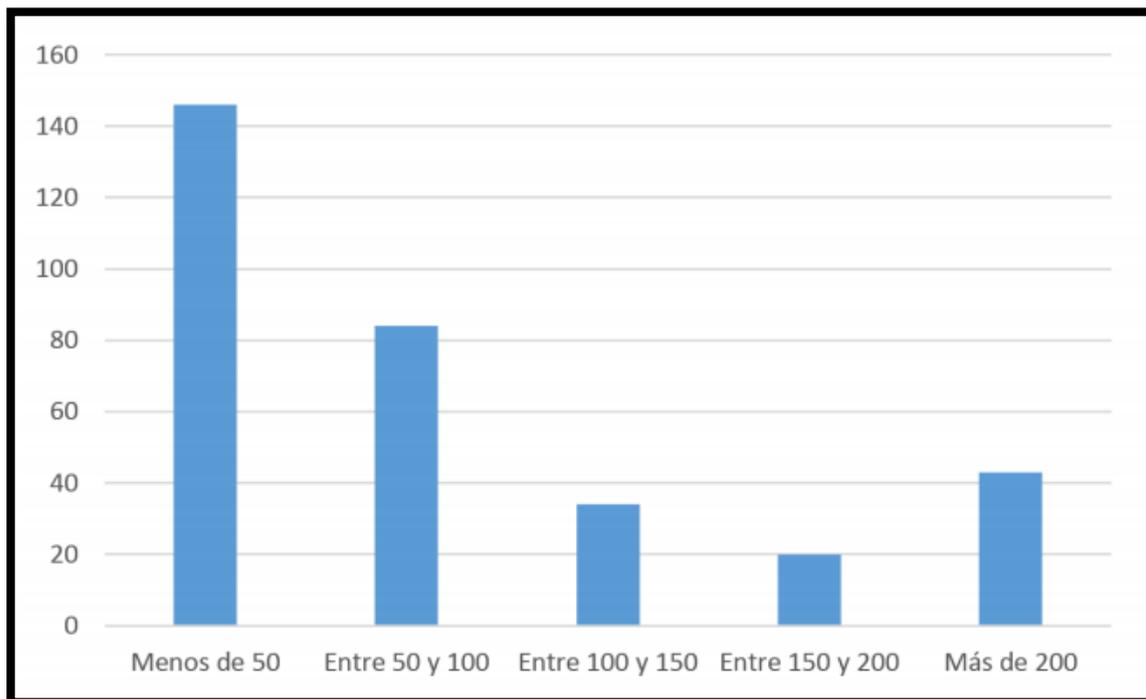


Ilustración 1 - Número de horas anuales dedicadas a la programación entre los niveles de Infantil, Primaria, ESO y Bachillerato (INTEF, 2018).

Este desconocimiento sobre programación proporciona una razón adicional para llevar Microsoft Excel junto con su lenguaje de programación basado en VBA a las aulas en estas etapas educativas. En este sentido, como se ha indicado anteriormente, Microsoft Excel posee un módulo dentro de él, que permite la creación y edición de pequeñas aplicaciones en lenguaje VBA llamadas ``macros``. Creemos que esto puede servir también para introducir a los alumnos interesados en el mundo de la programación.

Por otro lado, pensamos que Microsoft Excel es una de las herramientas más desconocidas en las etapas educativas de E.S.O. y Bachillerato, como pone de manifiesto el hecho de que el número de estudios relativos a las aplicaciones de la herramienta Microsoft Excel en la docencia de las matemáticas, y como herramienta de inicio a la programación, es mucho menor en comparación con los relativos, por ejemplo, a Geogebra.

Como ámbito de aplicación de este trabajo, hemos elegido el 2º curso de Bachillerato de la modalidad de Ciencias debido a las siguientes razones:

- La carga de la asignatura de matemáticas, ya que hay gran cantidad de contenidos a desarrollar en un periodo de tiempo relativamente pequeño.

- Dicho curso conduce hacia la mayor parte de estudios relacionados con la programación, ya que la mayoría de ellos o bien son universitarios o ciclos de grado superior. También es el curso académico después del que se incorporarán al mercado laboral algunos de los alumnos que deciden no continuar estudiando.
- En 2º de Bachillerato los alumnos no suelen estar interesados en aprender un software que dejarán de utilizar en un año, sino algo que les pueda servir en un futuro.

1.2 - OBJETIVOS

Este trabajo tiene marcados como principales objetivos aquellos descritos a continuación:

- Ilustrar de forma concreta la utilización de Microsoft Excel para la docencia de contenidos de la asignatura ``Matemáticas II`` en 2º de Bachillerato de la modalidad de Ciencias.
- Poner de manifiesto que Microsoft Excel resulta tan útil como otros programas matemáticos habitualmente usados en Bachillerato, con la ventaja de que el uso de Microsoft Excel está más extendido en otros ámbitos diferentes del educativo, o incluso en este último cuando se consideran etapas posteriores a 2º de Bachillerato.
- Demostrar, por medio de unas sesiones en las que se utilizará Microsoft Excel, que el empleo de hojas de cálculo diseñadas adecuadamente puede ayudar al profesor a explicar conceptos de la materia a impartir, además de facilitar la realización, corrección y comprensión de distintos tipos de ejercicios por parte del alumno de 2º de Bachillerato.
- Mostrar mediante ciertos talleres o sesiones adicionales que el uso de Microsoft Excel, junto con el lenguaje de programación VBA incorporado, resulta muy adecuado para realizar un primer acercamiento del alumno al mundo de la programación. Esto le ayudará a entender mejor el funcionamiento de otros lenguajes de programación diferentes de VBA en el futuro, y a adquirir un razonamiento basado en procesos lógicos.

1.3 - REVISIÓN BIBLIOGRÁFICA

Para la realización de este TFM se ha consultado información proveniente de diversos tipos de fuentes, entre las cuales destacan libros, publicaciones de revistas, normativa, etc. Todo ello ha sido realizado para desarrollar este trabajo de forma rigurosa y veraz.

2 - METODOLOGÍA

Con el fin de conseguir los objetivos anteriormente mencionados, se plantea utilizar Microsoft Excel en sesiones que se llevarán a cabo con estudiantes de 2º de Bachillerato de la modalidad de Ciencias. Dichas sesiones se desarrollarán de forma que no interfieran ni modifiquen en desarrollo temporal de la asignatura, ya que somos conscientes de la importancia de este curso académico para la mayoría de alumnos.

En concreto, proponemos usar algunas sesiones de clase para trabajar contenidos de la asignatura ``Matemáticas II`` con ayuda de unas hojas de Microsoft Excel previamente diseñadas, y recogidas en este TFM. Dichas hojas o archivos de Microsoft Excel estarán basadas, al menos en parte, en el lenguaje de programación VBA, e irán orientadas a reforzar contenidos anteriormente tratados en clase.

Una de características a destacar de las hojas de Excel que hemos elaborado es que, además de resolver el correspondiente ejercicio, permiten generar aleatoriamente ejercicios del mismo tipo, como se verá más adelante. Los alumnos dispondrán así de una herramienta útil que les permitirá practicar y ensayar todas las veces que sean necesarias.

Asimismo, de manera opcional para los alumnos, se propone un taller de programación impartido fuera de las horas lectivas habituales. Con dicho grupo de alumnos se tratará, de manera más específica, fundamentos de la programación con Visual Basic. Para ello se proponen una serie de ejercicios, recursos y guiones didácticos para que los alumnos que finalicen el taller terminen con conocimientos básicos de programación, siendo capaces de entender y crear código simple. Dicho taller estará constituido por una serie de sesiones descritas posteriormente en este mismo trabajo, así como los materiales, recursos y medios necesarios para su correcta realización.

En el siguiente cuadro están recogidos el calendario aproximado de las sesiones y talleres de programación. También los contenidos de la asignatura (junto con su bloque correspondiente) sobre los que se desea trabajar en las sesiones.

Tabla 1 – Calendario y tipo de ejercicio en las sesiones con Excel.

BLOQUE DE CONTENIDOS	SESIÓN CON EXCEL	FECHA
Cálculo del determinante de una matriz y obtención, cuando sea posible, de su inversa. Discusión y resolución de un sistema de tres ecuaciones con tres incógnitas.	BLOQUE 2: NÚMEROS Y ÁLGEBRA	Dos sesiones entre el 15 y el 25 de Noviembre.
Estudio de posiciones relativas en el espacio de rectas y planos.	BLOQUE 4: GEOMETRÍA	Una sesión entre el 10 y el 25 de Febrero.
Cálculo de integrales definidas y su aplicación al cálculo de áreas de regiones planas.	BLOQUE 3: ANÁLISIS	Una sesión entre el 5 y el 15 de Mayo.
Explicación, desde el punto de vista de la programación informática, del funcionamiento de las hojas usadas en clase.	TALLERES DE PROGRAMACIÓN OPCIONALES	Seis sesiones posteriores a las sesiones Excel.

Para poder contextualizar todo esto de manera correcta, un docente de la especialidad de matemáticas debe asumir no sólo la impartición de las sesiones con Excel relacionadas con los bloques de contenidos anteriormente aludidos, sino también la correcta gestión de los talleres de programación para los alumnos que deseen cursarlos. De acuerdo a nuestras estimaciones, sería deseable que el grupo de 2º de Bachillerato de Ciencias escogido para llevar a cabo dicha experiencia cuente al menos con unos veinte estudiantes, para asegurar que la parte de dicho alumnado que opta por los talleres de programación opcionales es lo suficientemente elevada como para ser representativa.

Como elemento adicional a la experiencia, se propone que los profesores y alumnos involucrados realicen unas encuestas. Por parte del equipo docente se propone la obtención de información doble, usando para ello un test de respuesta cerrada, con dos posibles opciones: afirmativa o negativa (investigación cuantitativa). Dicha

información se completará con una entrevista de carácter informal, basada en una serie de preguntas que permitan una respuesta abierta (investigación cualitativa).

En cuanto a los alumnos se debe tener en cuenta que existen dos grupos diferentes. Los que realizan el taller de programación, y aquéllos que sólo cursan sesiones basadas en Excel en sus clases de matemáticas. En el caso de que no asistan a los talleres de programación se les proporcionará un único test con preguntas con dos posibles opciones: respuesta afirmativa o negativa (investigación cuantitativa). Por otra parte a los participantes en los talleres de programación se les proporcionará, aparte de dicho test, otro más con cuestiones a las que deberán responder asignando un número natural entre 1 y 5, siendo el 1 "totalmente en desacuerdo" y el 5 "totalmente de acuerdo" (investigación cuantitativa).

No obstante, la obtención y análisis estadístico de estos datos, será algo que se plantea como línea de investigación futura.

3 - PROFUNDIZACIÓN

Como ya se ha mencionado en el apartado 1.1, a pesar de las grandes ventajas de aplicar la herramienta Microsoft Excel en la enseñanza, esta apenas es utilizada en comparación con otras en dicho ámbito. Seguidamente vamos a revisar brevemente algunos de los programas informáticos habituales orientados a la docencia de las matemáticas, y los principales usos y ventajas de éstos.

3.1 - GEOGEBRA

El uso de GeoGebra como herramienta didáctica, así como elemento complementario a la clase tradicional, ha demostrado ser práctico, útil y con un gran valor según las experiencias y estadísticas observadas (Giubergia, Graciela Socolovsky, & Ré, 2017).

Además según (Giubergia, Graciela Socolovsky, & Ré, 2017) GeoGebra presenta una serie de ventajas muy destacadas, las cuales son:

Tabla 2 - Datos resumen de GeoGebra.

<u>GEOGEBRA</u>	
Versión más reciente.	6.0.503.0
Plataformas:	iOS, Android, Windows, Mac, Chromebook y Linux.
Licencia.	Gratuito (Freeware).
VENTAJAS	INCONVENIENTES
-Al ser gratuita, permite una mejor implementación en los centros.	Es una herramienta meramente didáctica, la cual es muy poco probable que el alumno termine usando al finalizar su etapa educativa.
-Es de las aplicaciones que está disponible en un mayor número de plataformas.	
-Tiene disponible una versión online para usar sin necesidad de realizar ningún tipo de descarga o instalación previa.	
-Es multitarea, pues dispone de funcionalidad dentro de las áreas de: trigonometría, álgebra, funciones, estadística, probabilidad, etc.	
-Se ha demostrado que motiva en gran medida al alumnado, ya que favorece la participación de éste dentro del aula haciéndole protagonista de su propio aprendizaje según (Hernández Gómez, Briones Peñalver, Serdeira Azevedo, & Medina Vidal, 2016).	
-El uso de Geogebra en la educación ha mejorado y favorecido considerablemente la situación de ésta, según (Hernández Gómez, Briones Peñalver, Serdeira Azevedo, & Medina Vidal, 2016).	

3.2 - MAPLE

Maple es una herramienta enfocada a resolver problemas matemáticos basándose en el algebra computacional. Este programa lleva siendo desarrollado desde el año 1981 y se encuentra en constante evolución, es más, su última versión data del año 2018.

Es de las herramientas matemáticas más potentes y veloces que nos puede ofrecer el mercado actual hoy en día.

Una de las principales características de Maple es que su funcionamiento y uso se basa en un lenguaje de programación interpretado, es decir, funciona a través de una consola de comandos.

Según (Delgado Pineda, 1998) las principales ventajas e inconvenientes que podemos obtener del uso de Maple en la Enseñanza Secundaria Obligatoria son las que siguen a continuación:

Tabla 3 - Datos resumen de Maple.

MAPLE	
Versión más reciente:	2018
Plataformas:	Windows, Mac y Linux.
Licencia:	De pago.
VENTAJAS	INCONVENIENTES
-Se puede generar texto matemático con la sintaxis clásica de un libro de texto tradicional, es por ello que texto e instrucción se integran fácilmente.	<p style="text-align: center;">-El programa no tiene un medio de ejecutar instrucciones de manera gráfica, toda interacción con el software es realizado mediante la consola de comandos. Esto puede generar dudas por parte del profesor a la hora de ejecutar el código.</p> <p style="text-align: center;">-Los resultados obtenidos, a veces no coinciden con los esperados exactamente por el docente.</p>
-Permite la manipulación sencilla y rápida de gráficos a través de un menú de opciones, en el que hay gran variedad de posibilidades.	
-Ciertas instrucciones devuelven una animación que puede ser en 2D o incluso en 3D.	
-Dentro de una misma hoja de trabajo se pueden establecer grupos de instrucciones, ejecutando éstas por separado cuando sea necesario. Además, estas instrucciones pueden minimizarse en caso de que así se requiera.	
-Se puede exportar toda la hoja como un fichero HTML y abrirla con cualquier tipo de navegador Web.	

3.3 - DERIVE

Derive es un software informático que dispone de grandes posibilidades y de sencillo manejo por parte del usuario. Además puede ser ejecutado en computadoras de características muy reducidas (Quintana Perera, Estudio de límites y la continuidad de funciones usando DERIVE, 1996).

Algunas de las ventajas e inconvenientes que podemos destacar del empleo de Derive de cara a la docencia con respecto a otros programas según (Quintana Perera, Derivadas con DERIVE, 1996) son:

Tabla 4 - Datos resumen de Derive.

DERIVE	
Versión más reciente:	6.1
Plataformas:	Windows.
Licencia:	De pago.
VENTAJAS	INCONVENIENTES
-Los requisitos mínimos de la computadora donde va a ser ejecutado Derive son mínimos, por lo que puede ser ejecutado en cualquier ordenador actual.	-Su línea de edición es bastante pobre, en general es una herramienta que sacrifica eficiencia por sencillez en el manejo. -Las opciones presentes en la generación de gráficas en tres dimensiones son increíblemente reducidas.
-Su manejo es muy sencillo, de hecho, la mayoría de alumnos son capaces de utilizarlo de una manera suficientemente fluida en tan sólo dos o tres sesiones de trabajo.	

3.4 - GEUP

Geup dispone de dos versiones: la clásica y la 3D (orientada a visualizaciones gráficas). Este programa presenta una gran capacidad tanto de cálculo como de representación gráfica, además de disponer de una interfaz sencilla y manejable, lo que lo hace increíblemente fácil de usar y comprender. Todas estas características lo convierten en un programa con una aplicación muy orientada a la docencia de las matemáticas en cualquier nivel.

Destacamos ciertas ventajas en el uso de la herramienta GEUP aplicada a la docencia de las matemáticas:

Tabla 5 - Datos resumen de Geup.

GEUP	
Versión más reciente:	8.0.1.9
Plataformas:	Windows.
Licencia:	Gratuito (freeware).
VENTAJAS	INCONVENIENTES

<p>-Esta herramienta facilita increíblemente la posibilidad de aplicación del método matemático a la par que supone en recurso muy motivador para los alumnos (Galván Fernández, 2003).</p>	<p>Como principal desventaja podemos mencionar el hecho de que es un software con un uso muy poco extendido en comparación con los anteriormente mencionados. Esto hace que exista menos material disponible para el usuario acerca del programa.</p>
<p>-Dispone de capacidad para interpretar y visualizar transformaciones geométricas: simetría central, giros, simetría axial, traslaciones, etc.</p>	

3.5 - DESCARTES

El proyecto Descartes ha sido creado por el Centro Nacional de Información y Comunicación Educativa del Ministerio de Educación de España. El objetivo de dicho proyecto es incluir las TIC en la docencia actual, mostrando una nueva forma tanto de aprender como de enseñar las matemáticas.

Dicho proyecto según (Madrigal Muga, 2007) tiene un total de 5 líneas de actuación: La herramienta (software que se va actualizando), los materiales didácticos (entre ellos unidades didácticas a disposición de los profesores), la propia página web (donde se ofrece información completa del proyecto), formación para los docentes (a distancia mediante tutoría telemática, cursos) y la experimentación en el aula (experiencias de profesores).

Tabla 6 - Datos resumen de proyecto Descartes

DESCARTES	
Versión más reciente:	En constante actualización.
Plataformas:	Recursos para Windows.
Licencia:	Gratuito (freeware).
VENTAJAS	INCONVENIENTES
<p>-Todos los recursos presentes en dicho proyecto son totalmente gratuitos.</p>	<p>Es un proyecto muy amplio, con muchos materiales, lo cual hace que sea difícil establecer su principal grupo objetivo. Además, al depender de una gran comunidad, su desarrollo es irregular.</p>
<p>-Todos los recursos son creados por, y para, docentes.</p>	
<p>-Existe gran cantidad de material: recursos electrónicos, guías didácticas, consejos, etc.</p>	

3.6 - HOJAS DE CÁLCULO EXCEL

Como paso previo a analizar el software Microsoft Excel y el potencial de éste como herramienta didáctica, debemos preguntarnos... ¿Qué es una hoja de cálculo?

Una hoja de cálculo es cualquier programa que al ejecutarse nos muestra una tabla con celdas, a modo de matriz con filas y columnas, identificables por un par de datos. Dichas celdas pueden contener números, texto, fórmulas, etc. Estas hojas también permiten visualizar y tratar la información de manera gráfica, siendo actualmente Microsoft Excel la aplicación más difundida en este aspecto (Pifarré Turmo & Sanuy Burgués, 2000).

Con las hojas de cálculo Excel se pueden hacer infinidad de operaciones y cálculos y manejar cantidades ingentes de datos a un mismo tiempo. Tienen además la ventaja de convertir estos procesos en tareas automatizadas, ahorrando tiempo y esfuerzo.

Existen diferentes estudios que vienen a corroborar las ventajas de las hojas de cálculo Excel. Por ejemplo, se ha demostrado que el uso de hojas de Excel puede ser una alternativa extremadamente útil para los alumnos de matemáticas de la rama de Ciencias Sociales (Escribano Benito & Martínez García, 2007).

Por otro lado, somos consciente de que Excel es una herramienta menos ``clara'', pues su principal fin no es la enseñanza. A diferencia de otras, Excel tiene una interfaz menos sencilla y visual. Así, la hoja de Excel añade un pequeño reto, ya que el tratamiento o gestión de los datos resulta algo más tosco. Sin embargo, esto dependerá en gran medida de cómo haya sido preparada previamente dicha hoja de cálculo. Una persona con experiencia puede crear un archivo que sea increíblemente intuitivo, incluso más que otros programas con aplicaciones en la docencia, pero para esto se requiere cierta experiencia y conocimiento.

Las principales ventajas de las hojas Excel con programación VBA son:

- Es un software que se puede aprender y utilizar de manera sencilla, aún a pesar de que existe cierto nivel de dificultad en relación con el apartado de programación.

- Las hojas de cálculo pueden ser personalizadas por el docente hasta el último de los detalles, en lo que aspecto y funcionamiento se refiere. Un docente experimentado será capaz de crear un archivo Excel muy intuitivo y útil para el alumno.
- Siempre tienen alguna manera de representar gráficamente todo tipo de datos, y disponen de una amplia galería con diversos gráficos.
- Es posible exportar los datos a otros formatos o programas.
- Al automatizar las funciones y las operaciones se obtiene un inestimable ahorro de tiempo en los tiempos de ejecución de las tareas, pudiendo realizar más ejercicios en periodos menores de tiempo.
- Una operación puede ser realizada ``de golpe'', por muy compleja que sea, o ser dividida en múltiples pasos intermedios, siempre a gusto del usuario.

Sin embargo, es cierto que las hojas de cálculo Excel presentan una serie de desventajas importantes respecto de otros tipos de software educativos:

- Para que cumplan con su finalidad, previamente el docente debe elaborar una hoja de cálculo que realice las funciones buscadas, lo cual conlleva un trabajo anterior.
- A pesar de disponer de múltiples funcionalidades, existen carencias que se ponen de manifiesto en algunas situaciones que pueden aparecer en la docencia. Por ejemplo, en relación con la representación gráfica en tres dimensiones.

4 - MATERIAL PROPUESTO PARA LA DOCENCIA CON MICROSOFT EXCEL

Tal como se ha indicado anteriormente, vamos a distinguir dos tipos de sesiones formativas: las basadas en hojas de cálculo con macros, que servirán para tratar contenidos de la asignatura de matemáticas, y aquellas relacionadas con los talleres de programación.

4.1 - HOJAS DE CÁLCULO CON MACROS

Como ya se ha indicado, la hoja de cálculo no puede ser usada como recurso en bruto. Es decir, se requiere de una edición previa por parte del docente que haga adecuado el funcionamiento de la hoja de cálculo a los objetivos que se persiguen.

En este sentido, proponemos el uso de su módulo de programación en lenguaje VBA (modo programador) que nos permitirá dotar a nuestro archivo de funcionalidades personalizadas hasta el más mínimo detalle.

Podemos entender dicho módulo como un ``entorno paralelo`` al clásico de la hoja de cálculo. Es decir, a veces, podemos tener dos caminos para conseguir un mismo objetivo. Si deseamos que nuestro programa realice una acción determinada, esto se puede conseguir utilizando el entorno tradicional, o a través de macros, siendo este último el modo que más opciones presenta.

Usando dicho lenguaje de programación podemos incluir dentro de nuestra hoja de Excel elementos tales como: botones, mensajes en pantalla, control de opciones, etc. Así, conseguimos que la macro o archivo diseñado tenga apariencia de un programa independiente. De esta forma logramos que el entorno básico de una hoja de cálculo, el cual tal vez es un poco tosco para las etapas educativas mencionadas, se convierta en algo más interactivo y visual.

En cualquier caso, es importante insistir en que el entorno de trabajo basado en VBA no necesariamente es el único método para configurar nuestra hoja de Excel. Las hojas de cálculo generadas pueden basar también parte de su funcionamiento en fórmulas y procedimientos establecidos dentro de la propia hoja.

En este trabajo hemos elaborado una serie de hojas Excel que utilizan macros (Visual Basic), para trabajar algunos contenidos que se imparten en el curso de 2º de Bachillerato, modalidad de Ciencias. A continuación, se adjunta una tabla en la que se menciona qué contenidos se trabajan en cada uno de los archivos de Excel que se han preparado, y el bloque de temario en el que se incluyen dichos contenidos.

Tabla 7 – Contenidos trabajados en las sesiones de matemáticas en Excel.

CONTENIDOS	BLOQUE DE CONTENIDOS
Determinantes. Propiedades elementales. Rango de una matriz. Matriz inversa. Representación matricial de un sistema. Discusión y resolución de sistemas de ecuaciones lineales. Teorema de Rouché Frobenius. Método de Gauss.	BLOQUE 2: NÚMEROS Y ÁLGEBRA
La integral definida. Regla de Barrow. Aplicación al cálculo de áreas de regiones planas.	BLOQUE 3: ANÁLISIS
Posiciones relativas entre rectas y planos en el espacio.	BLOQUE 4: GEOMETRÍA
Explicación, desde el punto de vista de la programación informática, del funcionamiento de las hojas usadas en clases.	TALLERES DE PROGRAMACIÓN OPCIONALES

El funcionamiento del código de VBA debe ser conocido y comprendido por el docente para poder desarrollar de manera adecuada el taller de programación. Esto supone algo de complicación porque pese a que la dificultad de dicho código no es extrema, es posible que algunos docentes presenten ciertas dificultades no tanto en su comprensión, sino en adaptarse para realizar las explicaciones requeridas a los alumnos que decidan cursar el taller.

4.2 - TALLERES DE PROGRAMACIÓN

Por talleres de programación nos referimos a una serie de sesiones formativas opcionales que serán llevadas a cabo fuera de las horas lectivas habituales, donde se pretende impartir contenidos básicos sobre programación, más concretamente, en lenguaje VBA.

Esta vez no se buscará una adquisición de competencias matemáticas, ya que esos conocimientos han sido, o deberían haber sido, correctamente afianzados en las clases destinadas a ellos. En este caso, se busca que al finalizar los talleres, los alumnos sean capaces de interpretar y escribir código simple, además de que hayan podido tener una primera experiencia en programación.

Será el propio alumno el que decide si acudir o no a los talleres, en función, del interés futuro que pueda tener en campos relacionados con la programación, la tecnología o las TIC. En caso de que asistan a los talleres pensamos que esto les ayudará a incorporarse con mayor seguridad a estudios orientados a ese ámbito o sector laboral.

Para la correcta realización de cada taller el profesor se deberá disponer de un aula con al menos un ordenador conectado a un proyector. Por parte de los alumnos se requiere que éstos dispongan de portátil propio. En caso contrario el centro debería de ser capaz de aportar un aula con un mínimo de un ordenador por cada dos alumnos, y preferiblemente con conexión a Internet.

Este tipo de talleres o sesiones, además de ayudar a los alumnos a iniciarse a la programación en un futuro, también les sirve para el presente, ya que mejora y amplía las competencias digitales presentes en el currículo de matemáticas en 2º de Bachillerato de Ciencias.

En dichos talleres se comenzará desde lo más básico de la programación, usando como lenguaje VBA. Dicho lenguaje puede ser usado con Microsoft Excel, en caso de que sea habilitado para ello. Se considera un lenguaje idóneo para comenzar, pues es relativamente sencillo, intuitivo y comparte similitudes con otros lenguajes de programación muy usados como pueden ser C y JAVA.

A continuación se muestra un tabla en la cual se recogen el número de talleres de programación previstos, así como los conocimientos de programación que van a ser impartidos en cada uno de ellos.

Tabla 8 - Sesiones de los talleres de programación y contenidos a impartir.

TALLER	CONTENIDOS
TALLER Nº 1: INTRODUCCIÓN A LA PROGRAMACIÓN EN VBA	Introducción a la programación – Variables y tipos – Operadores básicos.
TALLER Nº 2: COMANDOS HABITUALES EN VBA	Comandos básicos de VBA – Uso de las variables en funciones.
TALLER Nº 3: INSTRUCCIONES DE BUCLE Y CONDICIONALES EN PROGRAMACIÓN	Instrucción IF y ELSEIF – Instrucción DO WHILE – Instrucción FOR – Operadores enfocados a condicionales.

TALLER Nº 4: OBJETOS VBA EN EXCEL	Objeto lista – objeto checkbox – Objeto botón – Objeto botón de opción – Modos de asociar código a acciones – Rangos de las variables.
TALLER Nº 5: PRACTICUM	Repaso mediante ejemplos y actividades de todo lo anterior.
TALLER Nº 6: REVISIÓN DE LOS PROGRAMAS USADOS EN CLASES DE MATEMÁTICAS	Interpretación de un código VBA previamente elaborado. – Realización de diagramas de procesos asociados a un código.

5 - SESIÓN DE USO DE LA HOJA EXCEL PARA TRABAJAR CON CONTENIDOS DEL BLOQUE II DEL CURRÍCULO

5.1 - CÁLCULO DE LA INVERSA DE UNA MATRIZ

Para realizar con éxito dicha sesión el alumno debe disponer de los conocimientos necesarios para calcular matrices inversas por sí mismo. Es decir, debe de ser capaz de obtener la inversa de una matriz con lápiz y papel.

A continuación se presenta un cuadro donde recogemos todos los contenidos previos que pensamos que el alumno debe conocer, antes de poder hacer uso de la hoja de Excel preparada para dicha actividad.

Tabla 9 - Contenidos exigidos para poder realizar la sesión de Excel sobre el cálculo de la inversa.

CONTENIDOS QUE DEBEN SER CONOCIDOS ANTES DE LA SESIÓN DE CÁLCULO DE LA INVERSA DE UNA MATRIZ.
Nomenclatura y definición de una matriz.
Operaciones básicas con matrices. Suma y multiplicación de matrices. Multiplicación por un escalar. Transposición de matrices.
Operaciones elementales en una matriz. Intercambio de filas. Suma y resta de filas. Multiplicación de filas por un escalar.
Determinantes. Cálculo y propiedades.

Objetivos:

-Repasar y profundizar en los contenidos relacionados con la obtención de la matriz inversa.

-Asimilar, desde un punto de vista práctico, el cálculo de la matriz inversa.

-Fomentar la colaboración y la convivencia en clase, mediante el trabajo en equipo de los alumnos.

-Buscar la motivación del alumno en la asignatura, a partir del uso de herramientas que le resulten atractivas e interesantes, como las TIC.

-Proporcionar un recurso didáctico que permita calcular, de forma inmediata, la inversa de una matriz dada o generada aleatoriamente.

Espacio físico y agrupamiento:

Se requiere que dicha sesión sea realizada en el aula de informática, y de modo que los estudiantes puedan ser divididos en parejas.

También será conveniente que el docente proporcione hojas con algún tipo de marca especial, con el objetivo de evitar que los estudiantes puedan copiar.

Evaluación:

Dicha actividad contará como una entrega, es decir, en caso de ser completada con éxito (más adelante se describe en qué consiste la superación de la prueba) contribuirá a la mejora de la nota del alumno en dicha unidad, según el docente considere apropiado.

Descripción:

La actividad propuesta tendrá el mecanismo de un concurso. Así los alumnos tendrán que resolver un número de ejercicios consistentes en la obtención de la matriz inversa de una dada, la cuál será de dimensiones 3×3 . El grupo de alumnos que obtenga el mayor número de matrices inversas correctas, gana. Para el equipo ganador, el docente establecerá una especie de premio, a elegir a criterio del mismo. Sin embargo, con haber realizado el cálculo correcto de al menos tres matrices inversas, la actividad se considerará superada por dicho grupo.

El profesor proporcionará a los alumnos las matrices sobre las que trabajarán. Para cada una de estas matrices, los estudiantes deberán, primero, ver si dispone de inversa y, después, en caso de que exista, calcularla. Las operaciones deberán ser

realizadas en papel, para poder ser comprobadas por el profesor y los alumnos posteriormente si fuera necesario.

	A	B	C	D	E	F	G	H
1								
2								
3		Columna 1	Columna 2	Columna 3		GENERAR MATRIZ		
4	Fila 1	2	1	-2		CALCULAR INVERSA		
5	Fila 2	1	1	-2				
6	Fila 3	-1	0	1				
7								
8								
9		Inversa de la matriz						
10						Det A		
11								
12								
13								
14								
15								
16								
17								

Ilustración 2- Inicio del ejercicio sesión Excel de cálculo de la inversa (elaboración propia).

Con cada una de las matrices, y una vez que el alumno ha obtenido la inversa “a mano” que según él es la correcta, deberá introducir los datos de la matriz original en la hoja de Excel, y pulsar el botón “CALCULAR INVERSA” de la hoja para obtener la solución real. Si se corresponde con la calculada por el alumno, se considerará que ha resuelto con éxito el ejercicio y procederá a iniciar uno nuevo. En caso contrario, debe revisar los pasos, para intentar encontrar su fallo antes de continuar con otro ejercicio.

	A	B	C	D	E	F	G	H
1								
2								
3		Columna 1	Columna 2	Columna 3		GENERAR MATRIZ		
4	Fila 1	2	7	3		CALCULAR INVERSA		
5	Fila 2	8	4	0				
6	Fila 3	1	9	6				
7								
8								
9		Inversa de la matriz						
10						Det A		
11								
12								
13								
14								

Ilustración 3- Datos mostrados al pulsar introducir o generar una matriz (elaboración propia).

En alguna ocasión se utilizará la opción de la hoja ``GENERAR MATRIZ`` pero siempre de forma limitada, ya que el cálculo de la inversa puede resultar tedioso en este caso, al estar la matriz de partida generada de manera aleatoria.

Como método para evitar que el alumno copie, o realice la actividad de alguna manera fraudulenta, se propone la siguiente medida preventiva: el alumno realizará los pasos de la resolución a bolígrafo sobre unas hojas marcadas por el docente.

Una vez finalizada esta actividad, el alumno estará completamente familiarizado con el funcionamiento del archivo Excel, por lo que éste será subido a la plataforma virtual del centro para que pueda practicar indefinidamente con él.

	A	B	C	D	E	F	G	H
1								
2								
3		Columna 1	Columna 2	Columna 3		GENERAR MATRIZ		
4	Fila 1	2	1	-2				
5	Fila 2	1	1	-2				
6	Fila 3	-1	0	1		CALCULAR INVERSA		
7								
8								
9		Inversa de la matriz						
10		1,00	-1,00	0,00		Det A		
11		1,00	0,00	2,00		1		
12		1,00	-1,00	1,00				
13								
14								
15								

Ilustración 4- Resultado obtenido al pulsar calcular inversa (elaboración propia).

5.2 - REPASO DE RESOLUCIÓN DE SISTEMAS DE ECUACIONES

En 2º curso de Bachillerato se aplican los conceptos aprendidos sobre matrices para la resolución sencilla de sistemas de tres ecuaciones con tres incógnitas. Es por ello que planteamos una hoja Excel dentro del mismo archivo Excel de ``Matrices`` para repasar y practicar dichos conceptos.

A continuación se recogen los conocimientos previos que deben de haber sido trabajados para la realización de esta actividad:

Tabla 10 - Contenidos exigidos para poder realizar la sesión de Excel sobre resolución sistemas de ecuaciones

CONTENIDOS QUE DEBERÍAN SER CONOCIDOS ANTES DE LA SESIÓN EXCEL RELACIONADA CON LA RESOLUCIÓN DE SISTEMAS DE ECUACIONES
Nomenclatura y definición de una matriz.
Operaciones básicas con matrices. Suma y multiplicación de matrices. Multiplicación por un escalar. Transposición de matrices.
Operaciones elementales en una matriz. Intercambio de filas. Suma y resta de filas. Multiplicación de filas por un escalar.
Determinantes. Cálculo y propiedades. Regla de Sarrus.
Rango de una matriz. Teorema de Rouché-Frobenius.
Resolución de sistemas de ecuaciones. Método de Gauss y de la inversa. Regla de Cramer

Objetivos:

-Repasar y profundizar en los contenidos relacionados con resolución de sistemas de ecuaciones.

-Asimilar, desde un punto de vista práctico, la resolución de sistemas de ecuaciones.

-Fomentar la colaboración y la convivencia en clase, mediante el trabajo en equipo de los alumnos.

-Buscar la motivación del alumno en la asignatura, a partir del uso de herramientas que le resulten atractivas e interesantes, como las TIC.

-Proporcionar un recurso didáctico que permita calcular, de forma inmediata, la solución de un sistema de ecuaciones, y generar uno de manera aleatoria.

Espacio físico y agrupamiento:

Se requiere que dicha sesión sea realizada en el aula de informática, en parejas.

También será conveniente que el docente proporcione hojas con algún tipo de marca especial, con el objetivo de evitar que los estudiantes puedan copiar.

Evaluación:

Dicha actividad contará como una entrega, es decir, en caso de ser completada con éxito (más adelante se describe en qué consiste la superación de la prueba) contribuirá a la mejora de la nota del alumno en dicha unidad, según el docente considere apropiado.

Descripción:

Debido a que dicha actividad requiere del uso de una serie de conceptos que marcan el fin de un bloque de contenidos, se recomienda que el docente repase muy brevemente los conceptos relacionados con: soluciones de un sistema en función del rango de la matriz de coeficientes y de la ampliada, cálculo de la inversa, y propiedades de una matriz con determinante distinto de cero.

The screenshot shows an Excel spreadsheet with the following layout:

	A	B	C	D	E	F	G	H	I	J
1										
2										
3		X	Y	Z	Ind.					
4		1	1	1	-1					
5		2	1	0	0					
6		1	0	1	-1					
7										
8										
9										
10		Inversa de la matriz				Det A				
11										
12										
13										
14										
15										
16										
17		Resultado								
18		X								
19		Y								
20		Z								
21										
22										

Buttons on the right side of the spreadsheet:

- Generar
- RESOLVER
- Abrir Calculadora

Ilustración 5 - Inicio del ejercicio sesión Excel de resolución de sistemas de ecuaciones (elaboración propia).

La forma de proceder en esta actividad es muy parecida a la seguida en la sesión Excel dedicada al cálculo de la inversa de una matriz. Así, el profesor facilitará a los estudiantes los sistemas de ecuaciones lineales que deberán resolver ``a mano'' por parejas. Todos los pasos y operaciones realizadas con cada sistema serán detallados por los alumnos en las hojas proporcionadas por el docente. Hecho esto, los estudiantes introducirán los datos del sistema en la hoja Excel y verificarán la validez de lo que han obtenido ``a mano''.

Antes de que los alumnos comiencen a estudiar los sistemas, se les indicará que existen dos formas en las que se puede obtener la solución con la hoja de Excel. Pueden usar el método de la inversa, en cuyo caso realizarán todos los pasos en las hojas marcadas, para finalmente comprobar si la solución es correcta, pulsando el botón "RESOLVER". Es entonces cuando el programa les mostrará la solución del correspondiente sistema de ecuaciones, pudiendo los alumnos comprobar si la resolución realizada con bolígrafo y papel era la correcta, y corrigiendo lo necesario en caso contrario.

Los estudiantes también pueden utilizar el método de reducción de Gauss, mediante el botón "Abrir Calculadora" disponible en la hoja de Excel. Esto hace aparecer una pequeña ventana que le permite al alumno introducir los datos de las filas que quiere sumar entre sí, y los múltiplos de éstas, para así crear un sistema equivalente pero más sencillo. Todos los cambios que realice serán instantáneamente actualizados en la correspondiente matriz. En este caso, todas las operaciones elementales con filas que realice el alumno, serán apuntadas de modo esquemático en las hojas marcadas para su posterior corrección.

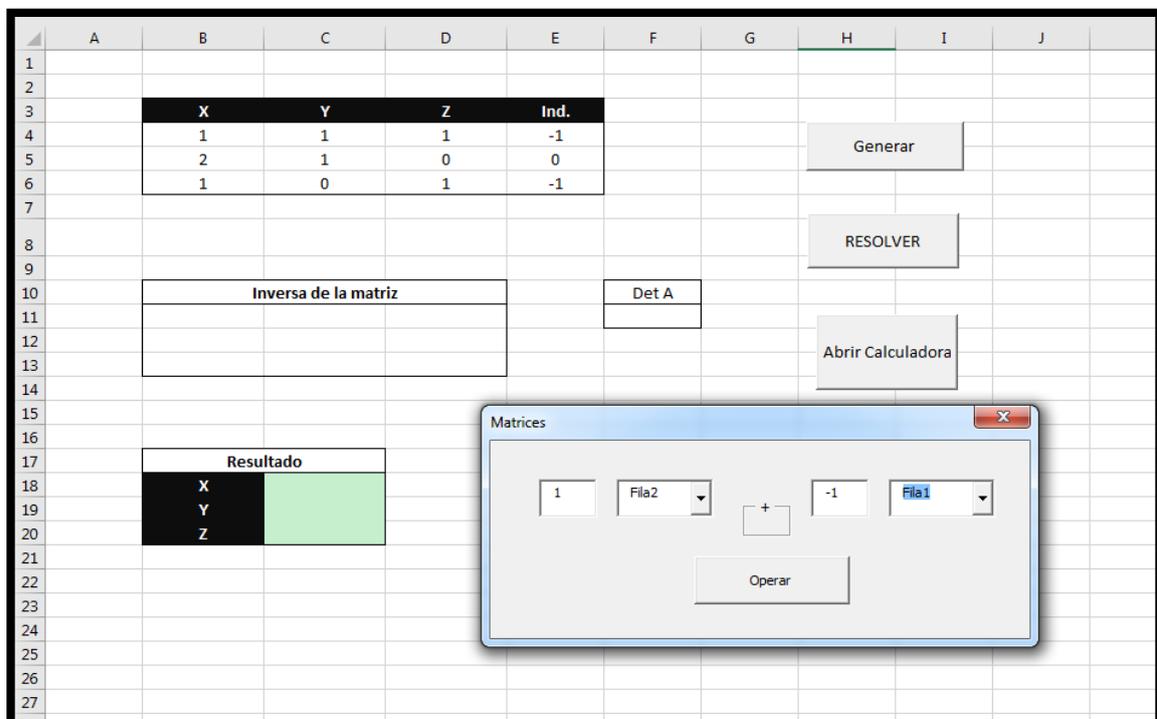


Ilustración 6- Calculadora de operaciones elementales de matrices (elaboración propia).

Como podemos ver en la ilustración anterior, para simplificar el sistema, se puede restar una vez la fila 1 a la fila 2. Si introducimos esas operaciones en la calculadora, el sistema las realizará y mostrará la nueva matriz resultante. Una vez dicha matriz esté correctamente escalonada, el alumno solo tendrá que ir obteniendo el valor de las incógnitas x , y , z .

	A	B	C	D	E	F	G	H	I
1									
2									
3		X	Y	Z	Ind.				
4		1	1	1	-1			Generar	
5		2	1	0	0				
6		1	0	1	-1				
7									
8								RESOLVER	
9									
10		Inversa de la matriz				Det A			
11		-0,50	0,50	0,50		-2			
12		1,00	0,00	-1,00					
13		0,50	-0,50	0,50				Abrir Calculadora	
14									
15									
16									
17		Resultado							
18		X	0						
19		Y	0						
20		Z	-1						
21									
22									

Ilustración 7 - Resultado mostrado por el programa (elaboración propia).

Es importante señalar que, en caso de que el sistema considerado no sea compatible determinado, la hoja de Excel diseñada nos informará sobre ello mediante un mensaje, dejando las casillas de matriz inversa y resultado como conjuntos vacíos. También se puede observar que, para facilitar la tarea de corrección al alumno cuando quiera repasar, el programa muestra el determinante de la matriz de coeficientes.

Pensamos que resulta útil poner a disposición de los estudiantes, en la plataforma virtual del centro, el archivo Excel, con el fin de que los alumnos puedan continuar practicando indefinidamente hasta que consideren que ya dominan esta parte de la materia.

6 - SESIÓN DE USO DE LA HOJA DE EXCEL PARA TRABAJAR CON CONTENIDOS DEL BLOQUE IV DEL CURRÍCULO

Aprovecharemos esta sesión con Excel para afianzar y practicar todo lo explicado en clase sobre posiciones relativas entre rectas y planos en el espacio.

6.1 - POSICIÓN RELATIVA EN EL ESPACIO DE DOS RECTAS, DE UNA RECTA Y UN PLANO, DE DOS Y DE TRES PLANOS.

En la asignatura ``Matemáticas II'', de 2º de Bachillerato de la modalidad de Ciencias, se estudian diferentes cuestiones relacionadas con geometría analítica en el espacio tridimensional. Especialmente importante resulta, en este sentido, el estudio de las posiciones relativas de dos rectas, de una recta y un plano y de dos y tres planos. Para tratar cada una de estas situaciones se ha elaborado una hoja Excel, que han sido recogidas por comodidad dentro del mismo archivo Excel.

Del mismo modo que sucedía con las sesiones relativas a contenidos del bloque II, los alumnos tienen que tener unos conocimientos previos para el correcto desarrollo de la actividad, de modo que sea realizada en condiciones aceptables y pueda ser de utilidad. Estos conocimientos previos están recogidos en la siguiente tabla:

Tabla 11 - Contenidos exigidos para poder realizar la sesión de Excel sobre posiciones relativas en el espacio.

CONTENIDOS QUE DEBERÍAN SER CONOCIDOS ANTES DE LA SESIÓN EXCEL SOBRE POSICIONES RELATIVAS EN EL ESPACIO.
Obtención del rango de una matriz.
Cálculo del determinante de una matriz.
Ecuaciones implícitas de una recta y de un plano en el espacio.
Posibles situaciones que pueden darse en el espacio entre rectas y planos.

Objetivos:

-Revisar de forma práctica los conceptos y resultados fundamentales impartidos sobre posiciones relativas de rectas y planos en el espacio.

-Afianzar y repasar nociones estudiadas con anterioridad, como rango de una matriz, determinante de una matriz e inversa de una matriz.

-Fomentar la colaboración y la convivencia en el aula, mediante el trabajo por equipos de los alumnos.

-Buscar la motivación del alumno en la asignatura, a partir del uso de herramientas que le resulten atractivas e interesantes, como las TIC.

-Proporcionar un recurso didáctico que permita analizar la posición relativa de rectas y planos en el espacio (mediante el estudio de los rangos de las correspondientes matrices), con la posibilidad, además, de generar ejercicios similares de manera aleatoria, para que el alumno pueda practicar de forma individual.

Espacio físico y agrupamiento:

Se requiere que la sesión sea realizada en el aula de informática, en parejas de alumnos.

También será conveniente que el docente proporcione hojas con algún tipo de marca especial, con el objetivo de evitar que los estudiantes puedan copiar.

Evaluación:

Pensamos que para que la actividad pueda ser satisfactoriamente realizada, cada grupo de estudiantes deberá entregar al menos un ejercicio correctamente resuelto de cada uno de los cuatro tipos que se van a plantear (con un mínimo de ocho por grupo). En ese caso, la actividad contribuirá positivamente a la mejora de la nota del alumno en esta unidad, según el profesor considere apropiado.

Descripción:

De manera previa al comienzo de la actividad, el docente escribirá en la pizarra la siguiente información para refrescar los contenidos impartidos, y ayudar al alumnado en la realización de los ejercicios.

POSICIÓN RELATIVA DE DOS RECTAS EN EL ESPACIO

$Rg(A) = 2 = Rg(A^*)$ // *Las rectas son coincidentes.*

$Rg(A) = 2 \neq Rg(A^*) = 3$ // *Las rectas son paralelas.*

$Rg(A) = 3 = Rg(A^*)$ // *Las rectas se cortan en un punto.*

$Rg(A) = 3 \neq Rg(A^*) = 4$ // *La rectas se cruzan en el espacio.*

POSICIÓN RELATIVA DE UNA RECTA Y UN PLANO EN EL ESPACIO

$Rg(A) = 2 = Rg(A^*)$ // *La recta está contenida en el plano.*

$Rg(A) = 2 \neq Rg(A^*) = 3$ // *La recta y el plano son paralelos.*

$Rg(A) = 3 = Rg(A^*)$ // *La recta y el plano se cortan en un punto.*

POSICIÓN RELATIVA DE DOS PLANOS EN EL ESPACIO

$Rg(A) = 1 = Rg(A^*)$ // *Los planos son coincidentes.*

$Rg(A) = 1 \neq Rg(A^*) = 2$ // *Los planos son paralelos.*

$Rg(A) = 2 = Rg(A^*)$ // *Los planos se cortan en una recta.*

POSICIÓN RELATIVA DE TRES PLANOS EN EL ESPACIO

$Rg(A) = 1 = Rg(A^*)$ // *Los planos son coincidentes.*

$Rg(A) = 1 \neq Rg(A^*) = 2$ // *2 planos paralelos y 1 coincidente, o 3 paralelos.*

$Rg(A) = 2 = Rg(A^*)$ // *Los planos se cortan en una recta.*

$Rg(A) = 2 \neq Rg(A^*) = 3$ // *Planos sectantes 2 a 2, o 2 paralelos y 1 secante.*

$Rg(A) = 3 = Rg(A^*) = 3$ // *Los planos se cortan en un punto.*

De modo análogo a las sesiones Excel anteriores, el profesor proporcionará los enunciados de los ejercicios a los estudiantes. Cada grupo deberá analizar la posición relativa entre rectas y planos del ejercicio correspondiente, realizando primero los cálculos ``a mano``. Posteriormente, a partir de las ecuaciones implícitas de las rectas y planos considerados, introducirá correctamente la información necesaria en la hoja Excel con el fin de comprobar, pulsando el botón ``Analizar``, si las conclusiones obtenidas en papel han sido acertadas. En caso de que no sea así, repasarán los cálculos ``a mano`` para intentar averiguar dónde está el error. Si por el contrario, la respuesta es correcta, continuarán con otro ejercicio o problema.

Durante la sesión se puede utilizar la opción ``Generar`` de la hoja Excel correspondiente, que generará aleatoriamente una matriz ampliada que será utilizada

para el estudio de la correspondiente posición relativa. Sin embargo, como el cálculo en papel del rango de una matriz generada aleatoriamente puede resultar laborioso para los alumnos, esta posibilidad será usada de manera ocasional, recomendando su uso sólo para practicar en casa.

Una vez concluida la actividad en clase, el archivo Excel será puesto a disposición de los estudiantes, de modo que pueden seguir sirviéndose de su ayuda durante el estudio de esta parte de la asignatura.

MATRIZ DADA POR LAS ECUACIONES DE LAS DOS RECTAS				
	Término X	Término Y	Término Z	Término independiente
RECTA 1	9	8	1	11
	12	11	3	0
RECTA 2	9	10	9	7
	0	3	1	0

Rango A	Rango A*	Resultado
3	4	Se cruzan en el espacio

Microsoft Excel
Rango A = 3 y Rango A+ = 4
Aceptar

Ilustración 8 - Ejercicio de posición relativa de dos rectas. Botón Analizar (elaboración propia).

MATRIZ DADA POR LAS ECUACIONES DE LA RECTA Y DEL PLANO				
	Término X	Término Y	Término Z	Término independiente
RECTA 1	1	1	1	2
	2	3	1	3
PLANO 1	7	10	4	11

Rango A	Rango A*	Resultado
2	2	La recta está contenida en el plano

Ilustración 9 - Ejercicio de posición relativa de una recta y un plano. Botón Analizar (elaboración propia).

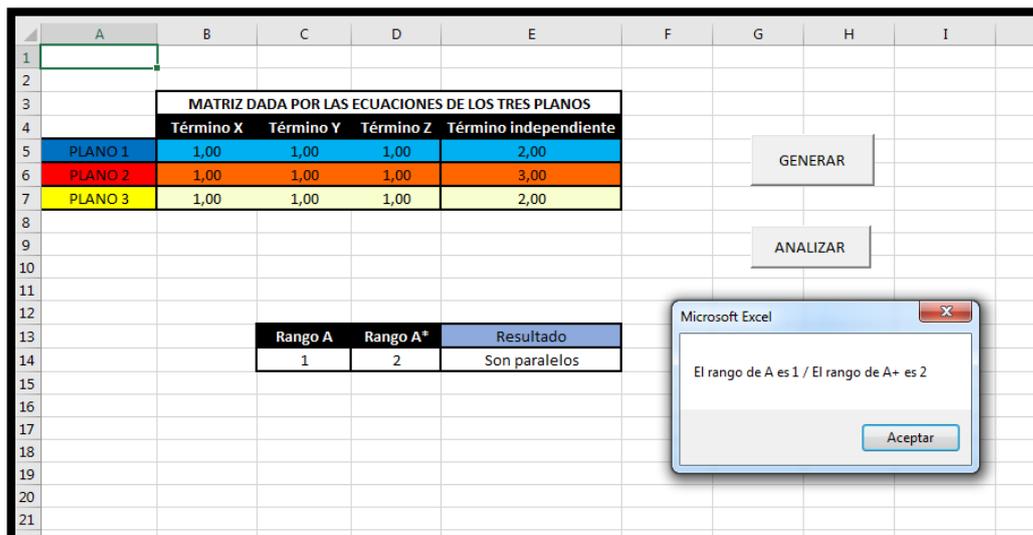


Ilustración 10 - Ejercicio de posición relativa de tres planos. Botón Analizar (elaboración propia).

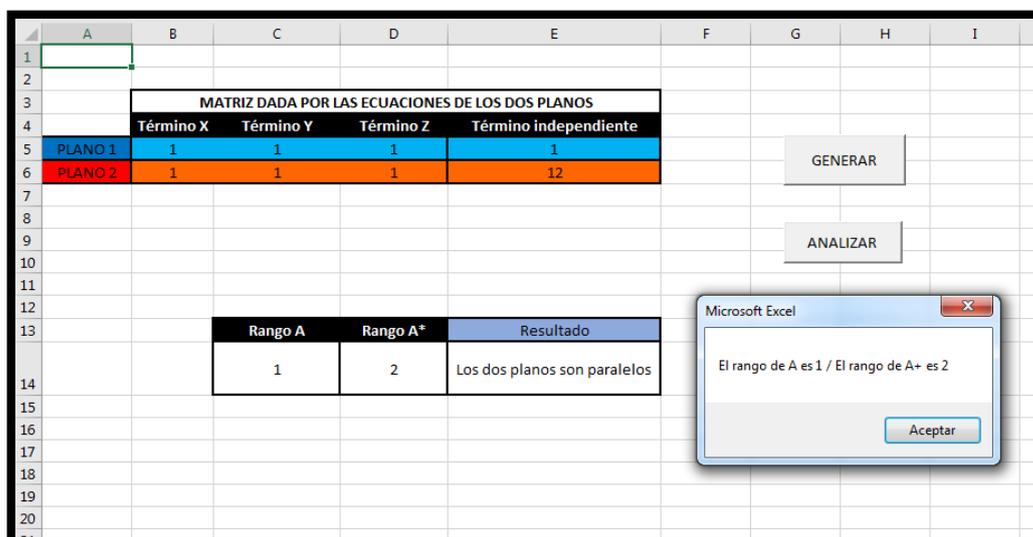


Ilustración 11 - Ejercicio de posición relativa de dos planos. Botón Analizar (elaboración propia).

7 - SESIÓN DE USO DE LA HOJA EXCEL PARA TRABAJAR CON CONTENIDOS DEL BLOQUE III DEL CURRÍCULO

7.1 - CÁLCULO DE INTEGRALES DEFINIDAS

En 2º curso de Bachillerato de Ciencias, se estudian diversos conceptos y resultados matemáticos relativos a funciones reales, entre ellos los relacionados con la integral definida. En concreto, los alumnos comienzan con el cálculo y obtención de las integrales indefinidas para, posteriormente, proceder al estudio de las integrales definidas. Como aplicación de estas últimas resulta de especial interés el cálculo de área de regiones planas limitadas por curvas dadas por funciones.

Para que el estudiante pueda seguir y realizar adecuadamente esta sesión, nos parece conveniente que conozca y esté familiarizado con los contenidos que se mencionan en la tabla siguiente:

Tabla 12 - Contenidos exigidos para poder realizar la sesión Excel sobre integrales definidas.

CONTENIDOS QUE DEBERÍAN SER CONOCIDOS ANTES DE LA SESIÓN EXCEL RELACIONADA CON LA INTEGRAL DEFINIDA.
Nociones básicas sobre funciones. Dominio y recorrido. Límite y continuidad. Simetría, etc.
Derivada. Definición, reglas de derivación y derivadas de funciones elementales.
Integral indefinida. Definición, cálculo de primitivas inmediatas y propiedades.
Métodos de integración. Integración por partes. Integración de funciones racionales. Integración por cambio de variable.
Regla de Barrow.

Objetivos:

-Repasar y afianzar lo estudiado en clase sobre los conceptos de integral indefinida e integral definida.

-Propiciar que el alumnado adquiriera agilidad en el cálculo de integrales tanto indefinidas como definidas.

-Revisar desde un punto de vista práctico la interpretación geométrica de integral definida como área de una región plana.

-Buscar la motivación de los alumnos por la asignatura mediante el uso de herramientas que resulten atractivas e interesantes, como las TIC.

-Proporcionar un recurso didáctico que permita obtener la integral definida de una función (continua) en un intervalo (cerrado y acotado).

Espacio físico y agrupamiento:

Se requiere que dicha sesión sea realizada en el aula de informática, en parejas.

También será conveniente que el docente proporcione hojas con algún tipo de marca especial, con el objetivo de evitar que los estudiantes puedan copiar.

Evaluación:

Durante dicha actividad los alumnos deberán calcular, utilizando bolígrafo y papel, una serie de integrales definidas, haciendo el correspondiente desarrollo previo que conduce a la obtención de éstas. El número de integrales propuestas será cinco o seis. Una vez finalizada la sesión, el docente recogerá los ejercicios para evaluarlos, y asignará a éstos el porcentaje que estime oportuno como calificación positiva dentro de la evaluación correspondiente al tema de integración.

Descripción:

En esta actividad se animará a los alumnos a proponer el cálculo de una integral preferiblemente polinómica, ya sea definida o indefinida, que en algún momento les supuso problemas al resolverla ``a mano''. En caso de que la integral propuesta por el alumno sea indefinida, será el propio docente el que establezca unos intervalos de integración apropiados, en función de la dificultad del ejercicio.

Una vez mostrada la integral, se dará a los alumnos un plazo de tiempo de unos diez minutos máximo para su resolución. Dicho periodo de tiempo debe estar impuesto por el criterio del propio docente, que valorará entre otras cosas la dificultad de dicha integral.

Transcurrido el periodo de tiempo establecido, el docente mostrará mediante el fichero Excel ``Integrales definidas'' la solución de dicha integral y además la gráfica de la función considerada. Los alumnos deberán calcular la integral definida también con su propio archivo de Excel. Cuando se haya concluido se procederá a proponer otra integral, repitiendo el proceso.

Para mostrar su resolución con Excel, solamente es necesario introducir en la casilla correspondiente la función que se desea integrar, además del intervalo de integración correspondiente, como se muestra en la figura adjunta.

Uso de Excel para la enseñanza de matemáticas en Bachillerato y talleres de inicio a la programación.

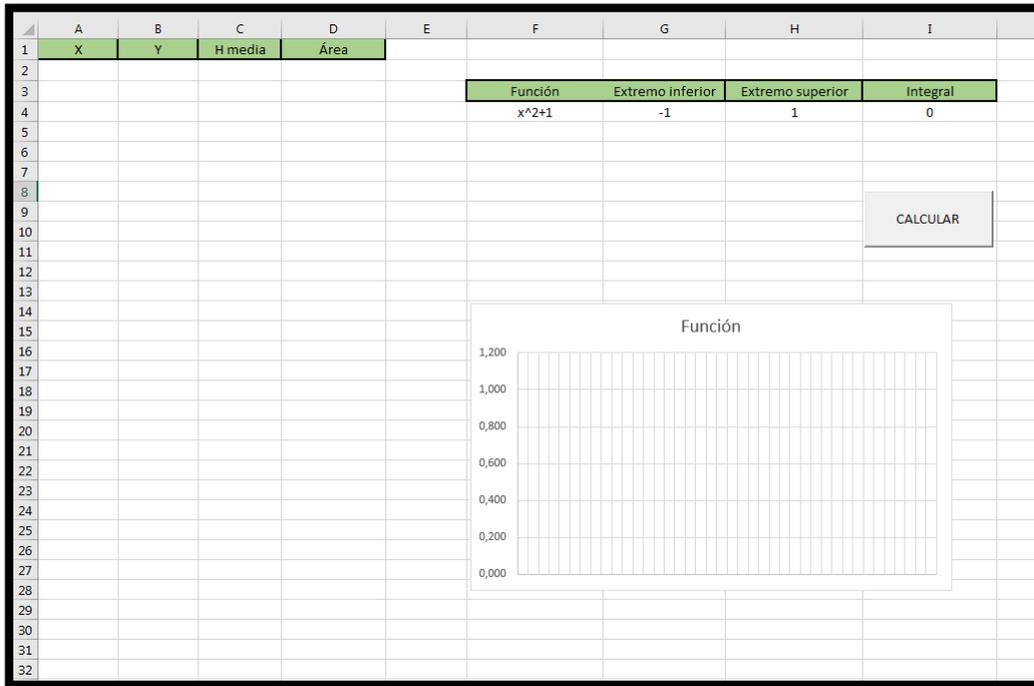


Ilustración 12 - Ejemplo previo de integral definida con la función $f(x)=x^2+1$ (elaboración propia).

Una vez introducida la función y los intervalos de integración, sólo resta pulsar el botón "Calcular" para que el programa Excel nos devuelva el valor aproximado de dicha integral definida.

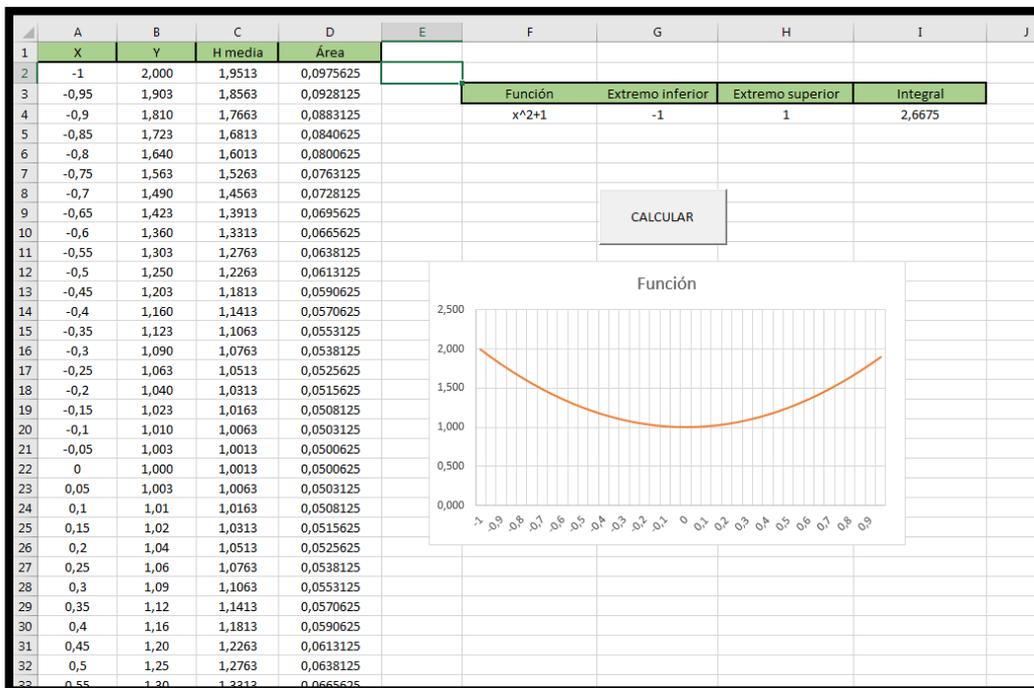


Ilustración 13 - Resultado de Excel tras resolver la integral de $f(x)=x^2+1$ (elaboración propia).

En caso de que quede tiempo suficiente, el profesor puede explicar cómo obtener un valor aproximado de la integral definida de una función sencilla en un intervalo, a partir de la suma de las áreas de unos rectángulos cuya base y altura están determinados, respectivamente, por el intervalo y la función correspondientes. Este tipo de procedimiento es precisamente el que se emplea en la hoja Excel diseñada para esta sesión.

Para ello, se puede considerar un ejemplo concreto y sencillo de integral definida, como es:

$$\int_0^1 x dx.$$

Después de dibujar en la pizarra la gráfica de $f(x) = x$, $x \in [0,1]$, el profesor representará los rectángulos azules que se muestran en la imagen que sigue a continuación.

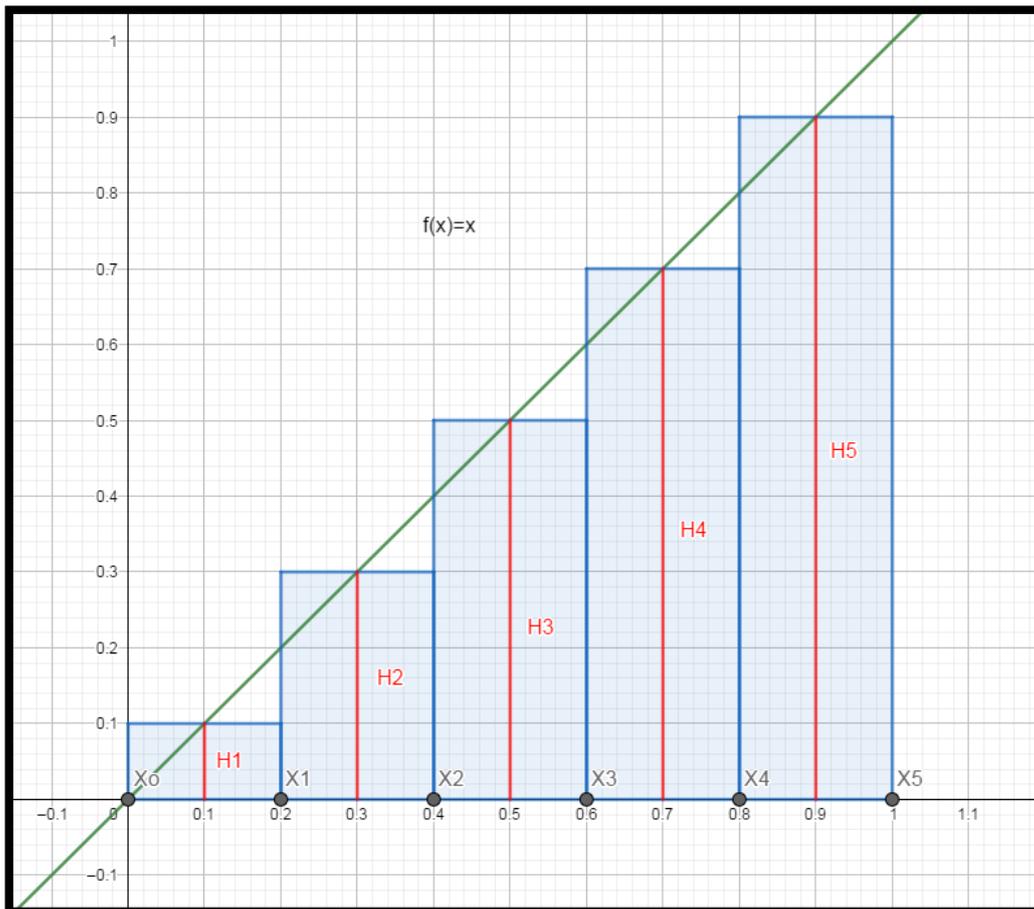


Ilustración 14 - Gráfica auxiliar para explicar cómo aproximamos la integral definida (elaboración propia).

El docente debe tratar de hacer ver a los alumnos que para aproximarse al valor del área de la región ubicada debajo de la función dada, en el intervalo $[0,1]$, se puede calcular el área de cada uno de los rectángulos en azul y sumar posteriormente las áreas de dichos rectángulos.

Debe hacerse especial hincapié en que dicho método trae consigo un error más que evidente, ya que el área obtenida no será exactamente la real, sino aproximada. Dicho error puede ser notablemente reducido disminuyendo el tamaño de la base de cada uno de los rectángulos. Muchos programas informáticos realizan los cálculos de esta manera, suponiendo rectángulos muy pequeños, con el fin de obtener una aproximación razonable o adecuada.

Se planteará la siguiente pregunta a los alumnos. ¿sabemos qué área tiene cada uno de los rectángulos? Tras un momento de reflexión, el docente puede indicar que, a partir de la ilustración anterior, es fácil observar que en un rectángulo cuya base esté determinada por un intervalo $[x_{n-1}, x_n]$, la altura de dicho rectángulo será

$$h_n = \frac{f(x_{n-1}) + f(x_n)}{2} \quad (n \geq 1).$$

Así, el área de este rectángulo vendrá dado por el valor:

$$Base \cdot Altura = (x_n - x_{n-1}) \cdot h_n = \frac{(x_n - x_{n-1}) \cdot (f(x_{n-1}) + f(x_n))}{2} \quad (n \geq 1).$$

Como ya se ha mencionado, cuanto mayor sea el número n de subintervalos que se consideren (es decir, cuanto más pequeña sea la longitud de los intervalos que constituyen la base de los rectángulos), mejor será la aproximación obtenida.

8 - TALLERES DE PROGRAMACIÓN

A continuación, se va a proceder a describir las sesiones educativas planteadas para iniciar a los alumnos a la programación informática. Como ya se ha mencionado, todas estas actividades deberán tener carácter opcional y libre ya que, en función de las inquietudes académicas o aspiraciones laborales de los alumnos, es muy posible que una parte de ellos no encuentren interés en dicho tipo de actividad, mientras que otros la encontrarán sumamente enriquecedora.

8.1 - TALLER Nº 1: INTRODUCCIÓN A LA PROGRAMACIÓN EN VBA

En este primer taller se va a iniciar a los alumnos en la programación. En particular, se va a plantear una serie de conceptos teóricos acerca de los tipos de variables existentes en lenguaje VBA y su uso. Posteriormente, se propondrán ejercicios prácticos para afianzar los conceptos previamente mencionados.

Objetivos:

- Comprender cómo funciona un lenguaje de programación de manera básica.
- Ser capaz de diferenciar los distintos tipos de variables existentes en el lenguaje de programación VBA.
- Asignar correctamente un tipo de variable adecuado para el almacenaje de datos en función de su tipo.
- Comprender las operaciones básicas que pueden llevarse a cabo entre variables de tipo numérico.

Espacio físico y agrupamiento:

Se debe disponer de un aula con un ordenador por cada alumno a ser posible. Si esto no fuera posible se propone establecer grupos de dos alumnos. Dichas computadoras deberán estar dotadas del paquete Microsoft Office, para que los alumnos puedan seguir las instrucciones del docente mediante el apartado Visual Basic de Excel, y tomar apuntes usando Microsoft Word.

Descripción:

El docente debe explicar que la programación consiste en una serie de órdenes lógicas y estructuradas, que se le envían a la máquina y que son realizadas siempre con un orden de prioridad. Es importante recalcar que, en programación, sólo puede realizarse una acción u operación al mismo tiempo. Al final de la clase, se mostrará un ejemplo de por qué esto siempre ocurre así.

8.1.1 - Macros

Macro, que deriva de la palabra macroinstrucción, no es más que un conjunto de instrucciones más simples, especificadas por un código informático en concreto. Según

(ebriik.com, 2010) una macro consiste en un conjunto de líneas de código entendidas como una sola instrucción, ejecutadas por nosotros a voluntad al pulsar un determinado botón, combinación de teclas, etc. Es decir, una macro nos ayuda a automatizar y gestionar tareas más fácilmente. Podemos verlo en la siguiente imagen.



Ilustración 15 - Herramienta "Macros" dentro del panel de Programador (elaboración propia).

Pulsando en el icono "Macros" (ver imagen anterior), los alumnos pueden acceder a la ventana de gestión de las Macros, permitiéndoles crear una nueva, modificar las existentes, borrarlas, etc.

Una de las primeras explicaciones que el docente debe realizar, es que nosotros podemos asociar código a acciones sobre distintos elementos. Por ejemplo, se puede crear un botón de forma que cada vez que hacemos clic en él, o pasamos el botón por encima de él, etc. se ejecuta el código.

El docente debe indicar a los alumnos que como todavía no saben escribir código para crear una por ellos mismos, se va a usar a modo de introducción la herramienta "Grabadora de Macros".

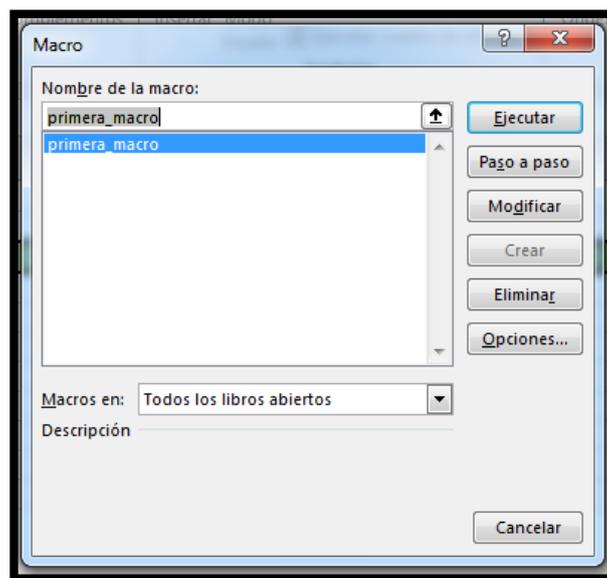


Ilustración 16 - Ventana de Macros (elaboración propia).

A continuación, se solicitará a todos los alumnos que realicen los siguientes pasos de manera ordenada para familiarizarles con el funcionamiento de la grabadora de macros.

- 1) Pulsar en el botón “Grabar Macro”, que aparece justo al lado del de “Macros”.

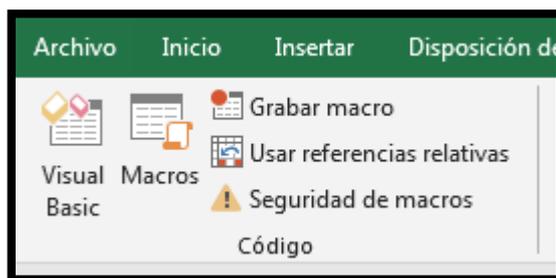


Ilustración 17 - Botón “Grabadora de Macros” (elaboración propia).

- 2) Dar nombre a su primera macro, por ejemplo “Primera_Macro”. Además, deberán asignarle una letra del teclado, para poder ejecutarla automáticamente mediante la combinación de Ctrl y dicha tecla.

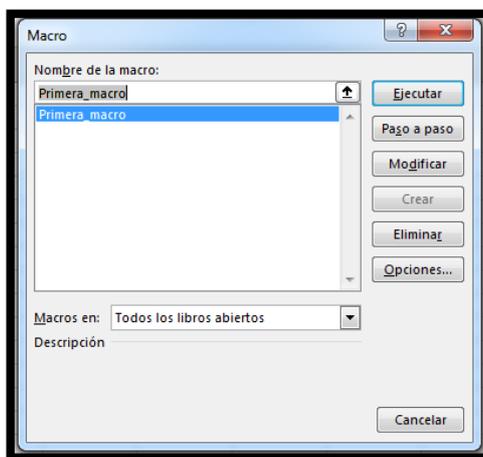


Ilustración 18 - Una macro creada llamada “Primera_Macro” (elaboración propia).

- 3) A continuación, deberán ir a la celda C4 y escribir “Texto en celda C4 y copiado a la celda B5”.
- 4) Tras escribir esto, deben copiar el contenido de dicha celda a la celda B5.
- 5) Después, acudirán al botón que antes era “Grabar Macro”, ahora convertido en “Detener grabación” y lo pulsarán. Ya tienen su primera Macro creada con dicho nombre, que pueden ejecutar siempre que lo deseen, desde la ventana “Macros” o pulsando Ctrl más la tecla asignada.

- 6) Los alumnos deberán acudir a la ventana Macro, seleccionar la suya, y pulsar en modificar. Es ahora momento del docente, de explicar que ese es el código de programación que hace la acción que ellos realizaron manualmente, y es objetivo de dichos talleres de programación el aprender a crearlo, y analizarlo.

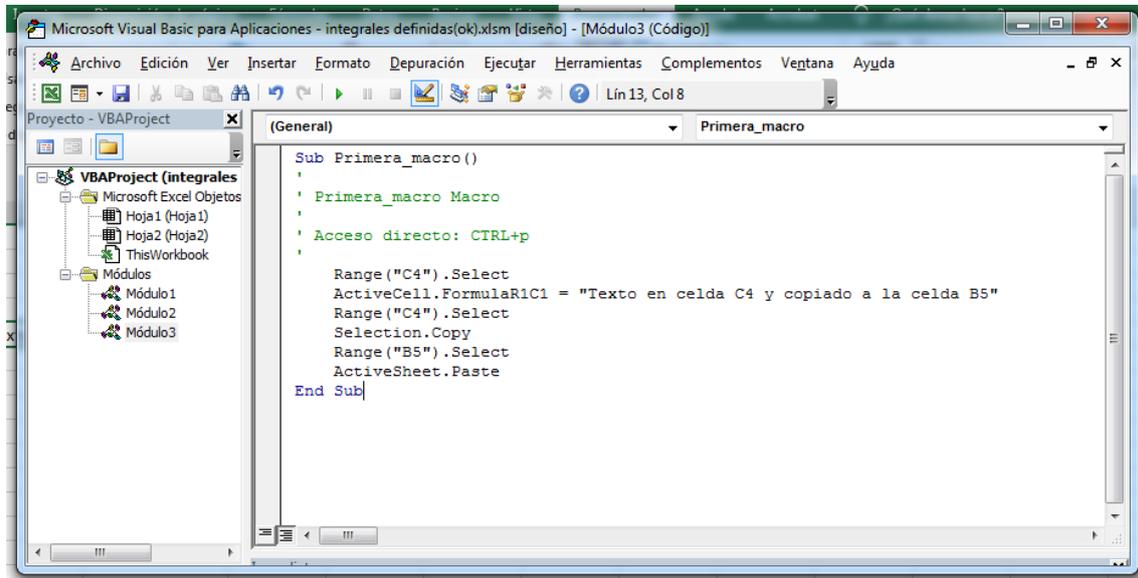


Ilustración 19 - Visualización del código asociado a "Primera_Macro" (elaboración propia).

8.1.2 - Variables

Una **variable** es un elemento que el lenguaje de programación crea para almacenar un tipo de dato o información concreta según (Lannoy, 2011). Si nosotros queremos darle a la máquina una información, por ejemplo, el número 5, tenemos que crear algún método para que la máquina pueda almacenar el dato introducido. Para ello se usan las variables:

number = 5

Con esta instrucción el programa entiende que tiene una variable llamada **number** y en el espacio de almacenaje de esa variable, almacenamos el número cinco. De tal manera, siempre que el programa o nosotros recurramos o solicitemos la variable **number** eso deberá ser entendido como "5". Esto será así siempre y cuando no variemos el contenido de dicha variable.

Por otro lado, señalaremos que para almacenar números no nos vale cualquier tipo de variable, ya que existen distintas clases de éstas. Debemos de escoger aquélla que permita almacenar números en su interior.

A continuación, se proporciona una tabla, donde se muestran algunos de los diferentes tipos de variables en el lenguaje de VBA (Duigou, 2016), así como los tipos de datos que son capaces de albergar.

Tabla 13 - Tipos de variables en VBA.

TIPO DE VARIABLE	TIPO DE DATOS	OBSERVACIONES
<i>Integer</i>	Números enteros.	De poca longitud.
<i>Long</i>	Números enteros más largos.	De más longitud.
<i>Single</i>	Número decimal.	Hasta 6 decimales.
<i>Double</i>	Número decimal más largo.	Más de 6 decimales.
<i>Boolean</i>	Verdadero o Falso.	Sí o No.
<i>String</i>	Texto.	Almacena cualquier texto.

Para finalizar con la explicación teórica, se indicará cómo abrir el gestor de código dentro del propio Microsoft Excel (con la pestaña ``Programador´´) para así poder declarar nuestras variables. Declarar una variable consiste en ejecutar la línea de código necesaria para que el programa entienda que existe una variable con el nombre y tipo que nosotros deseamos. La línea de código a enseñar para la declaración de variables sería la siguiente.

Ejemplo 1 – taller nº 1:

*Dim **nombredvariable** as tipodevariable*

***Nombredvariable** = dato*

En la primera línea del código anterior, lo único que hemos hecho ha sido crear la variable **nombredvariable**, la cual es del tipo especificado. Una vez creada la variable, se puede operar con ella, por ejemplo almacenando un dato en su interior, pero es muy importante tener bien claro que primero es necesario declararla correctamente.

```
Dim numerouno as Integer
```

```
Dim numerodos as Long
```

```
Numerouno = 2
```

```
Numerodos = 2.34
```

```
Dim resultado as Long
```

*Resultado = numerouno * numerodos*

¿Qué podemos observar en el ejemplo anterior? Vemos que hemos declarado tres variables: una de número entero (*integer*) y dos de decimal tipo largo (*long*).

Posteriormente, se ha asignado a la variable **numerouno** el valor 2 y a la variable **numerodos** el número 2,34. Sin embargo, a la variable **resultado** se le ha asignado el producto de las dos variables anteriores. Esto es posible ya que, recordando lo mencionado anteriormente, cuando se escribe en el código el nombre de una variable que tiene un dato almacenado, el programa entiende el valor de dicho dato. De esta forma en la variable **resultado** el valor almacenado es 4,68.

Hay que tener mucho cuidado con el tipo de variable asignado, porque si la variable resultado fuese de tipo *string*, es decir, texto, el programa devolvería error, ya que no entiende que la multiplicación de dos valores numéricos sea almacenado en una variable tipo texto.

Otra curiosidad más es que si la variable resultado fuese de tipo *Integer*, devolvería el valor de 5, ya que no es capaz de almacenar números decimales, y los redondea en caso de obligarle a almacenar uno.

8.1.3 - Actividades

Para asegurarnos de la correcta asimilación de los contenidos teóricos anteriormente mencionados por parte de los alumnos, se proponen las siguientes actividades para que sean desarrolladas después de la correspondiente explicación por parte del docente:

Actividad 1 - taller nº 1:

Se proporcionará una serie de datos que se quieren introducir al programa, para poder posteriormente acceder a ellos a través del código. Sin embargo, los datos son de orígenes muy diversos. Se pide al alumno que asigne el tipo de variable más recomendable para su almacenamiento.

Tabla 14 - Actividad 1 - taller nº 1: tipos de variables (elaboración propia).

DATO	TIPO DE VARIABLE
5	Integer
gato	String
5,2354	Single
Verdadero	Boolean
casa	String
-4	Integer
6,4325342934	Double
Falso	Boolean
43984329329049032	Long
Variable integer * variable integer	Long
Variable single * variable single	Double

8.1.4 - Operadores

Aquello que conocemos como operadores no son más que elementos que nos permiten combinar el valor de variables, constantes, etc. entre sí, dando como resultado de dicha operación otro valor (Blanco, 2002).

Para que sea más fácil de entender, en matemáticas, por ejemplo, son operadores, la suma, la resta, la multiplicación, la división, etc.

A continuación, se presenta una tabla con operadores básicos que los alumnos deben conocer para comenzar a manejarse en VBA.

Tabla 15 - Operadores básicos en VBA (Blanco, 2002).

OPERADOR	NOMBRE	DESCRIPCIÓN
*	Multiplicación	Multiplica dos valores numéricos.
+	Suma	Suma dos valores numéricos.
-	Resta	Resta dos valores numéricos.
/	División	Divide dos valores numéricos.
&	Concatenación	Permite unir variables de tipo texto.
\	División entera	Divide dos cifras, y devuelve el resultado entero.
^	Potencia	Eleva un número a otro.
a Mod b	Resto	Devuelve el resto de a entre b.

8.2 - TALLER Nº 2: COMANDOS HABITUALES EN VBA

En este segundo taller se profundizará en el uso de variables en la programación, ahora que los alumnos comprenden los distintos tipos existentes. Además se indicará cómo pueden relacionarse entre ellas.

Para ello se presentan distintos comandos o funciones propios del lenguaje VBA aplicado en Excel, y cómo pueden interactuar éstos con las variables, para poder crear un programa simple.

Objetivos:

- Conocer comandos básicos de VBA aplicado a Excel.
- Conocer el editor de código, y ser capaz de interactuar con él para generar programas sencillos.
- Familiarizarse con la interfaz de usuario de un compilador.

Espacio físico y agrupamiento:

El taller será realizado de manera individual, pudiendo los alumnos compartir por parejas un ordenador en caso de ser necesario. Dichos ordenadores deberán estar dotados del paquete Microsoft Office, para que los alumnos puedan seguir las instrucciones del docente mediante el apartado Visual Basic de Excel, y tomar apuntes usando Microsoft Word.

Descripción:

Al comienzo del taller, el docente deberá mencionar que existen una serie de instrucciones, comandos, etc. que cada lenguaje de programación reconoce de manera automática. Esto quiere decir que si nosotros escribimos cierto conjunto de caracteres, el programa lo entenderá y realizará siempre cierta acción. En la siguiente tabla se recogen algunos ejemplos de este tipo de macros.

Tabla 16 - Listado de comandos o funciones básicas de inicio a VBA.

COMANDO / FUNCIONES BÁSICAS	FUNCIONAMIENTO
MsgBox "Texto "& resultado &" texto "	Muestra por pantalla: Texto (datos en variable resultado) texto.
Variable = Range("A1")	Almacena en la variable lo que haya en la celda A1 (compatibilidad de datos!).
Variable = Cells(i, j)	Almacena en la variable lo que haya en la celda (i,j)(compatibilidad de datos). Además, i j deben ser variables Integer o números enteros.
elemento.Value	Convierte, si es posible, el contenido de elemento (puede ser celda, variable, etc.) en números.
Elemento.Text	Convierte si es posible, el contenido de elemento (puede ser celda, variable, etc.) en texto.
variable = Application.InputBox(prompt:="El segundo número es:", Type:=2)	Pide al usuario un valor por teclado, y lo almacena en la variable. Alterando el parámetro TYPE podremos hacer que varíe la ventana saliente por pantalla.
Range("F6:G7") = Application.WorksheetFunction.MInverse(Range("B1:C2"))	Hace la inversa de la matriz B1:C2 y la almacena en las celdas F6:G7.
Deter= Application.WorksheetFunction.MDeterm(Range("B1:C2"))	Hace el determinante de la matriz B1:C2 y lo almacena en la variable deter.
Cells(i, j) = InputBox("Por favor inserte un valor")	Pide al usuario un valor por teclado, y lo almacena en la celda (i,j).
Range("B1:C2").Clear	Borra y limpia las celdas desde B1 hasta C2.

El docente debe explicar, aunque sea de manera breve, el funcionamiento de cada uno de estos comandos, para posteriormente proceder con la exposición de un ejercicio práctico.

Ejemplo 1 - taller nº 2:

Se va a proceder a crear un programa, que pedirá un total de dos números al usuario, tras lo cual, los multiplicará para mostrar el resultado en un mensaje, y almacenarlo en una celda en concreto.

```
Dim numerouno As Double
Dim numerodos As Double
Dim resultado As Double
numerouno = Application.InputBox(prompt:="El primer número es:", Type:=2)
numerodos = Application.InputBox(prompt:="El segundo número es:", Type:=2)
resultado = numerouno * numerodos
MsgBox "La multiplicación da: " & resultado & ""
Cells(3, 3) = resultado
```

El docente explicará punto por punto el proceso que el programa sigue:

- Se definen las variables con su tipo correspondiente.
- Se asignan valores a las variables, las dos primeras con valores introducidos mediante el teclado.
- Se muestra la variable **resultado** por pantalla.
- Se almacena la variable **resultado** en la celda C3.

A continuación, se introduce el concepto de **diagrama de procesos**. Esto no es más que una especie de esquema, que ayuda a comprender los pasos que un programa sigue en su ejecución, mediante bocadillos y flechas, incluyendo si es necesario algunas partes del código. Es de gran utilidad debido a su componente visual, por lo que a partir de ahora, y hasta el final del taller, todos los ejemplos del docente irán acompañados de su correspondiente diagrama de procesos. También se espera que los alumnos vayan siendo capaces de dibujar los suyos propios al mismo tiempo que comprenden el funcionamiento del código.

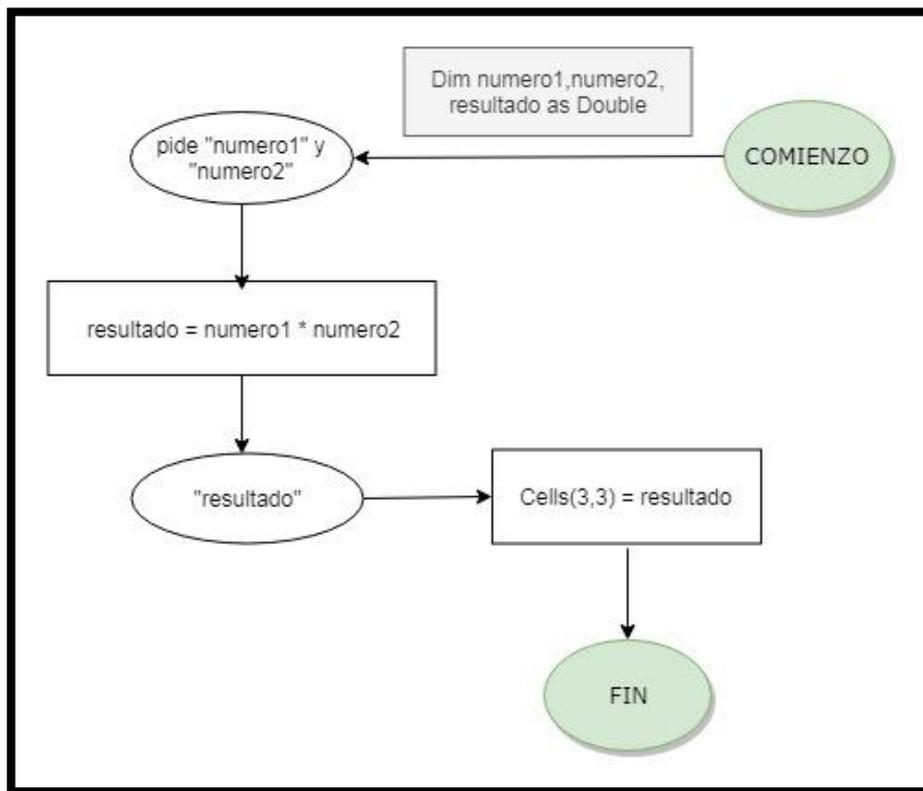


Ilustración 20 - Diagrama del ejemplo 1 - taller nº 2 (elaboración propia).

Es necesario destacar, como se hizo con el primer taller, que las órdenes o instrucciones en programación siempre tienen un orden y nunca pueden suceder al mismo tiempo. Esto puede apreciarse en el diagrama y en el propio código, donde cada instrucción aparece en una línea, por lo que tiene asignada un orden relativo con respecto a otras líneas de código.

En el caso anterior si, por ejemplo, si la variable **resultado** recibiera un valor antes que las otras dos, al estar una de ellas vacías, multiplicaría por cero, dándonos un fallo en el funcionamiento del programa. O si, por ejemplo, asignamos un valor a una variable antes de definirla, al no entender qué es, el programa devolverá un error. Éstos son sólo algunos de los muchos ejemplos que existen, donde se ve que el orden de las líneas de código es muy importante para el correcto funcionamiento del programa.

Antes de proponer el siguiente ejercicio en el taller, es necesario que el profesor realice el siguiente apunte: existe un procedimiento llamado **depuración**, que permite ubicar la línea de código en donde se ha producido un error. Así, al ejecutar el programa o macro, aparece un mensaje de aviso a modo de ventana y el programa se detiene. En

dicha ventana se muestra la opción de acceder al elemento llamado depurador (Amelot, 2016).

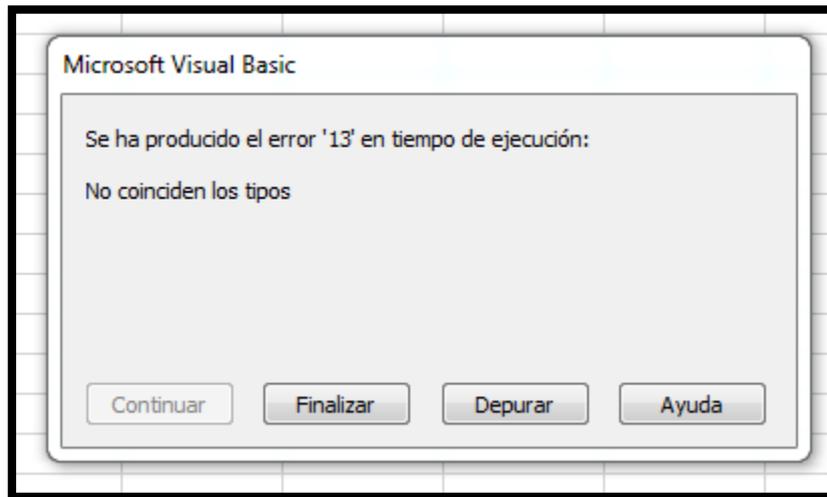


Ilustración 21 - Aparición de un error, y opción de depurar el código (elaboración propia).

Una vez pulsado el botón depurar, el programa nos llevará al código de la macro donde se ha producido el problema, resaltando en color amarillo la línea concreta. Debemos editar dicha línea adecuadamente, de forma que al ejecutar de nuevo el programa o macro no se produzca error.

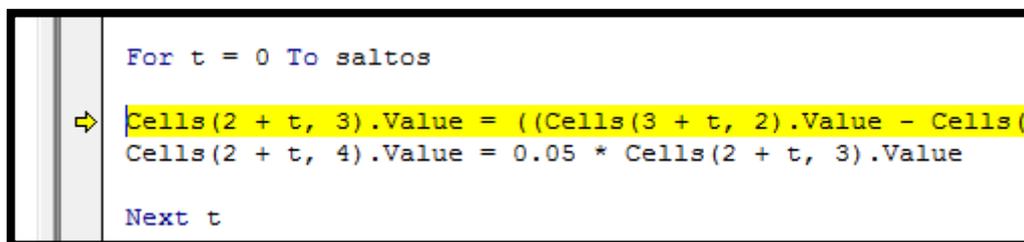


Ilustración 22 - Excel resaltando la línea donde está presente el error de código (elaboración propia).

Actividad 1 – taller nº 2:

A continuación, se propone a los alumnos que hagan uso de los comandos y tipos de variables conocidos para desarrollar un programa que realice la siguiente tarea:

Dada una matriz cuadrada, cuyos términos se encuentran entre las celdas A1:C3, el programa pedirá dos números. El primero de ellos deberá ser sumando al determinante de la matriz y el segundo multiplicará a dicha cifra resultante. Después de realizar estas dos operaciones, el resultado obtenido deberá de ser almacenado en la celda G4. Por último, se le sumará una unidad, y se mostrará por pantalla.

Dim numerouno As Double

Dim numerodos As Double

Dim deter As Double

Dim suma As Double

Dim multi As Double

numerouno = Application.InputBox(prompt:="El primer número es:", Type:=2)

numerodos = Application.InputBox(prompt:="El segundo número es:", Type:=2)

deter = Application.WorksheetFunction.MDeterm(Range("A1:C3"))

suma = deter + numerouno

*multi = suma * numerodos*

MsgBox "El número buscado es:" & multi + 1 & ""

Cells(4, 7) = multi

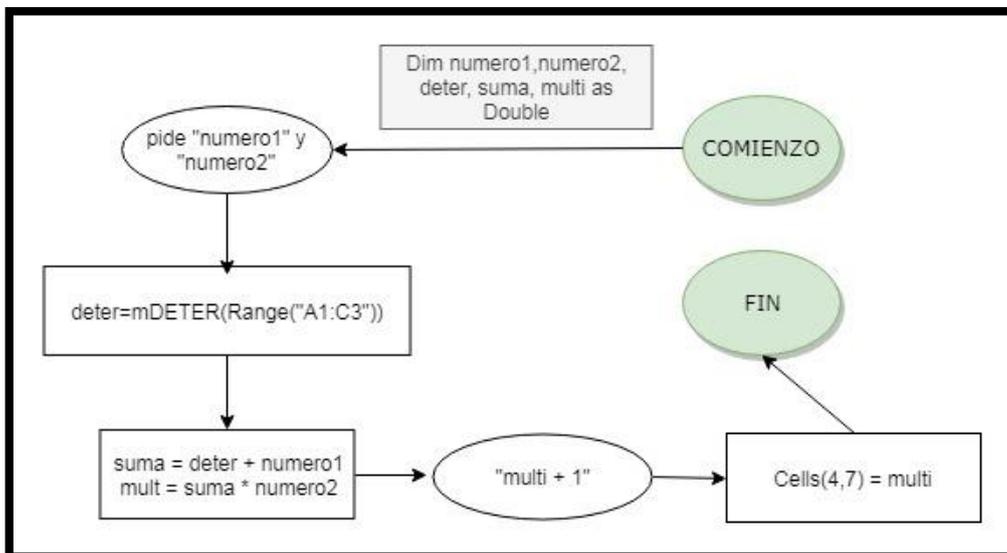


Ilustración 23 - Diagrama de la actividad 1 - taller nº 2 (elaboración propia).

Se debe destacar la importancia de crear una variable para cada resultado, a pesar de poder hacer todas las operaciones de golpe en una sola variable, ya que esto nos ayuda a tener un mayor control sobre la ejecución del programa. También merece la pena mencionar que podría crearse una tercera variable para mostrar el resultado por pantalla, pero se puede hacer de manera directa como vemos en el diagrama.

8.3 - TALLER N° 3: INSTRUCCIONES DE BUCLE Y CONDICIONALES EN PROGRAMACIÓN

En esta sesión de los talleres de programación se va a introducir al alumno en el manejo de instrucciones básicas de programación como son: **IF, DO WHILE, FOR**.

Objetivos:

-Comprender tanto el uso como el funcionamiento de las instrucciones **IF, DO WHILE** y **DO FOR** en programación.

-Ser capaz de analizar los requisitos que se piden en un programa y elegir el modo de emplear estas instrucciones de bucle y condicionales para dar solución a la situación planteada.

-Comprender la importancia de las variables de tipo contador y sus usos principales en condicionales y bucles.

-Ser capaz de establecer las condiciones adecuadas mediante el uso de elementos o variables.

Espacio físico y agrupamiento:

Se requiere que la sesión sea llevada a cabo en el aula de informática. Será realizada de manera individual, pudiendo los alumnos compartir por parejas un ordenador en caso de ser necesario. Dichos ordenadores deberán estar dotados con el paquete Microsoft Office, para que los alumnos puedan seguir las instrucciones del docente mediante el apartado Visual Basic de Excel, y tomar apuntes usando Microsoft Word.

Descripción:

Pensamos que este taller puede ser organizado mejor dividiéndolo en varias partes: una por cada una de las tres instrucciones anteriormente mencionadas, otra para un ejemplo, y una última dedicada a la realización de una actividad propuesta por el docente.

8.3.1 - Instrucción IF

La instrucción **IF** en programación es comúnmente conocida como un condicional.

Cuando usamos **IF** en programación debe ir unido a una condición. Si se cumple dicha condición (TRUE) se ejecuta el código dentro del **IF**. Si no se cumple dicha condición (FALSE), no se ejecuta el código del **IF** (Le Guen, 2013). Además, **IF** permite de manera opcional, añadir condiciones adicionales mediante el uso de **ELSEIF**. Todo lo anterior se entiende mejor con la siguiente imagen:

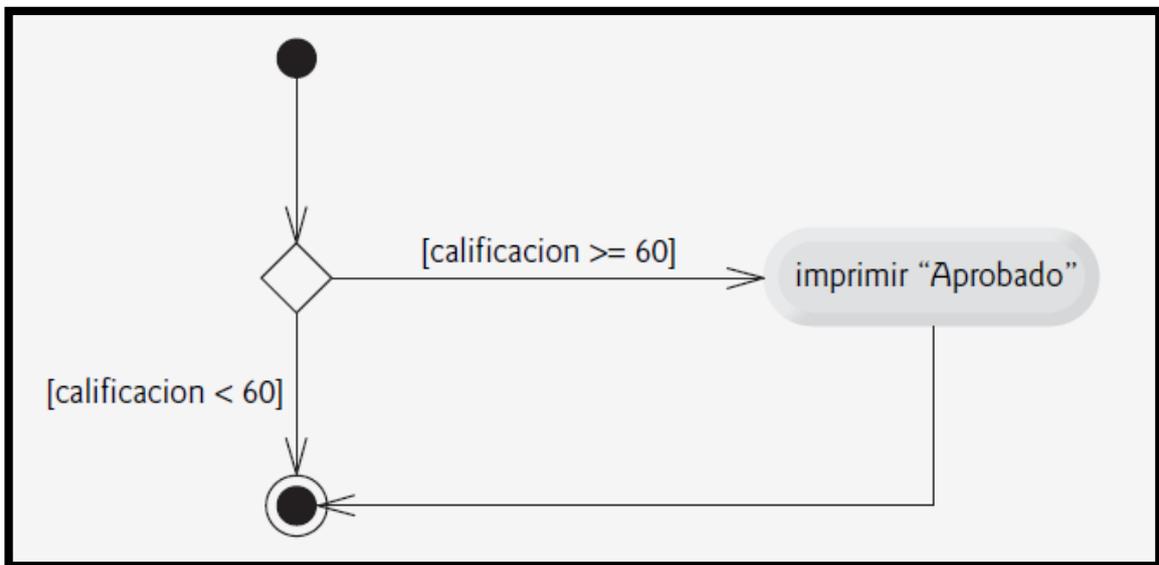


Ilustración 24 - Funcionamiento de instrucción IF sin ELSE (Deitel & Deitel, 2008).

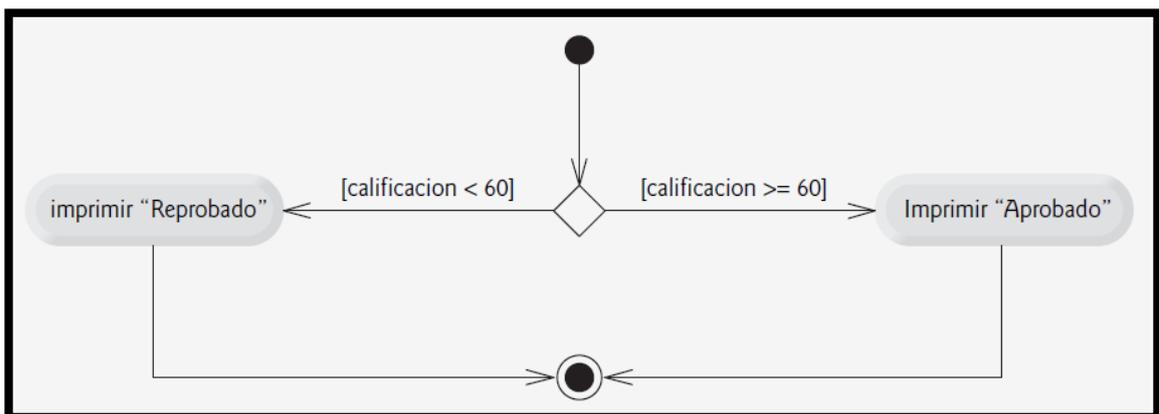


Ilustración 25 - Funcionamiento de instrucción IF - ELSE (Deitel & Deitel, 2008).

El código para ejecutar dicha instrucción es el siguiente, y debe ser mostrado al alumno.

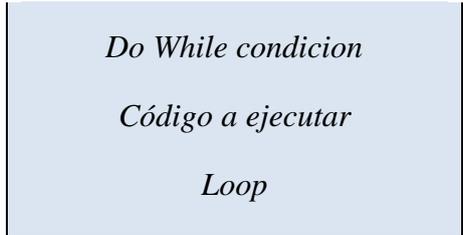
```
If condicion1 Then  
(Codigo a ejecutar si se cumple la condición 1)  
-----Código opcional-----  
ElseIf condicion2 Then  
(Codigo a ejecutar si se cumple la condición 2)  
ElseIf condicion3 Then  
(Codigo a ejecutar si se cumple la condición 3)  
-----Código opcional-----  
Else  
(Código a ejecutar si no se cumple ninguna condición anterior)  
End If
```

Si **condición1** se cumple, se ejecuta el código dentro de **IF**. En caso de que se cumpla **condición2**, se ejecuta el código asociado al segundo **ELSEIF**, lo mismo para **condición3**, y por último, en caso de no cumplirse ninguna de las condiciones anteriores, se ejecuta el código asociado a **ELSE**. Si nuestro **IF** carece de instrucción **ELSE** al final del mismo, al no cumplirse ninguna condición de las anteriores, el programa no haría nada. Pasa algo parecido con los apartados **ELSEIF**, los cuales son prescindibles.

No hay que olvidar que se pueden incluir condiciones **IF** dentro de otros **IF**, es lo que en programación se conoce como condicionales anidados.

8.3.2 - Instrucción DO WHILE (bucle) y uso de variables contador.

La instrucción **DO WHILE** es también una de las más usadas en programación, independientemente del lenguaje usado. Es una instrucción que ejecuta ciertas líneas de código, siempre y cuando se repita una condición (Ladron de Guevara, 2014). Es decir, si cierta condición se cumple, ejecuta un código; al terminar de ejecutarlo, comprueba de nuevo si se cumple dicha condición y en caso afirmativo repite el código anterior. A continuación se muestra la estructura general de dicha instrucción.



Como dato destacable, hay que señalar que, dentro del código de la instrucción **DO WHILE**, es necesario imponer algo que sea capaz de alterar la condición que hace que dicha instrucción se repita, ya que si esto no sucede, la instrucción se repetirá de manera infinita, generando un bucle sin fin que va a impedir al programa salir de él, obligándonos a cerrar de golpe el proceso, y perdiendo los datos no guardados anteriormente.

Esta condición para controlar la entrada y salida en **DO WHILE** se realiza la mayoría de ocasiones con una variable, la cual llamaremos a partir de ahora **variable contador**. Podemos apreciar en el siguiente diagrama de procesos que dentro de **DO WHILE** alteramos el valor de dicha variable, para evitar el bucle infinito antes mencionado.

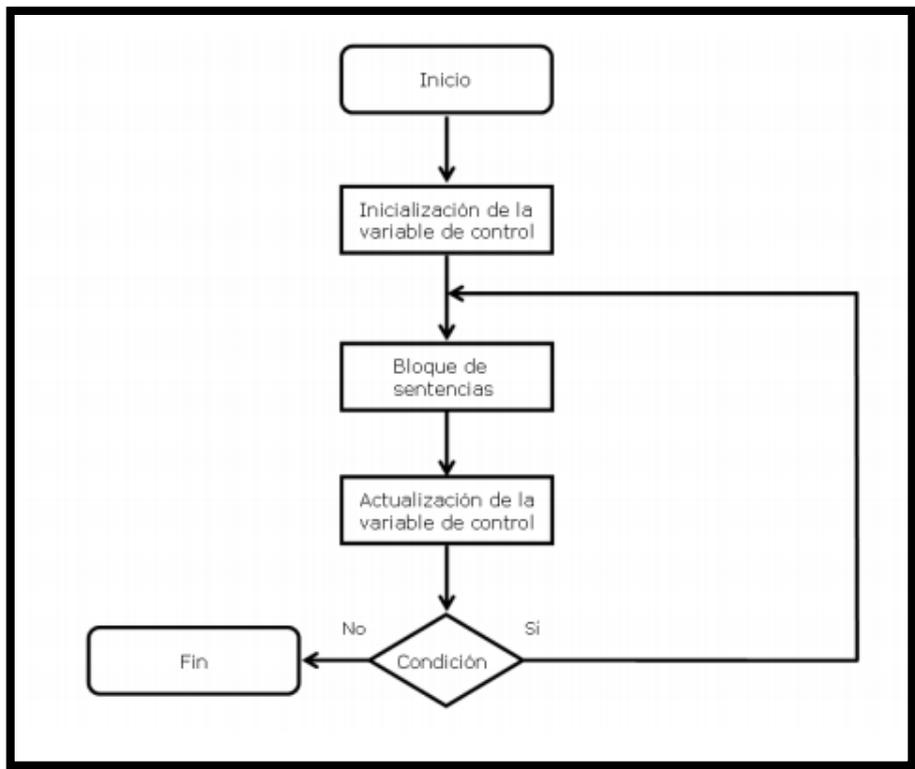
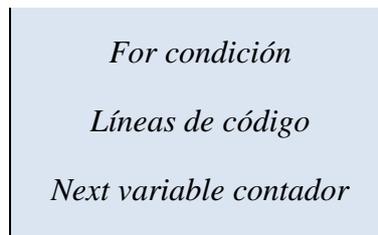


Ilustración 26 - Funcionamiento de la instrucción bucle DO WHILE (Ladron de Guevara, 2014).

Las **variables contador** son variables de tipo *Integer* que sirven para generar la condición de la instrucción **DO WHILE**. Esta variable aumentará en cierto valor cada vez que se repita el bucle (normalmente una unidad), haciendo que éste se repita justo el número de veces deseado. Posteriormente se plantea un ejemplo para que sea más sencillo de comprender.

8.3.3 - Instrucción bucle FOR

La instrucción **FOR** tiene el mismo objetivo o función que la instrucción **DO WHILE**. Nos permite repetir una serie de líneas de código siempre que se cumpla una determinada condición, que será comprobada cada vez que dicho código se ejecuta (Ladron de Guevara, 2014). A continuación se muestra su estructura de manera genérica:



```
For condición
Líneas de código
Next variable contador
```

En este caso, podemos observar que el fin del bucle **FOR** lo marca la línea **next variable**. Dicha variable es recomendable que sea la misma que usamos para la condición (**variable de contador**), de tal manera que podemos controlar fácilmente el número de veces que el bucle se repite. Al ejecutar **next** la variable aumenta en una unidad.

Como comentario final, queremos señalar que las **variables contador** usadas para medir el comienzo y final de los bucles, deben de ser de tipo *Integer*.

En la siguiente imagen se puede apreciar un diagrama de procesos estándar en lo que al funcionamiento de la instrucción de bucle **FOR** se refiere.

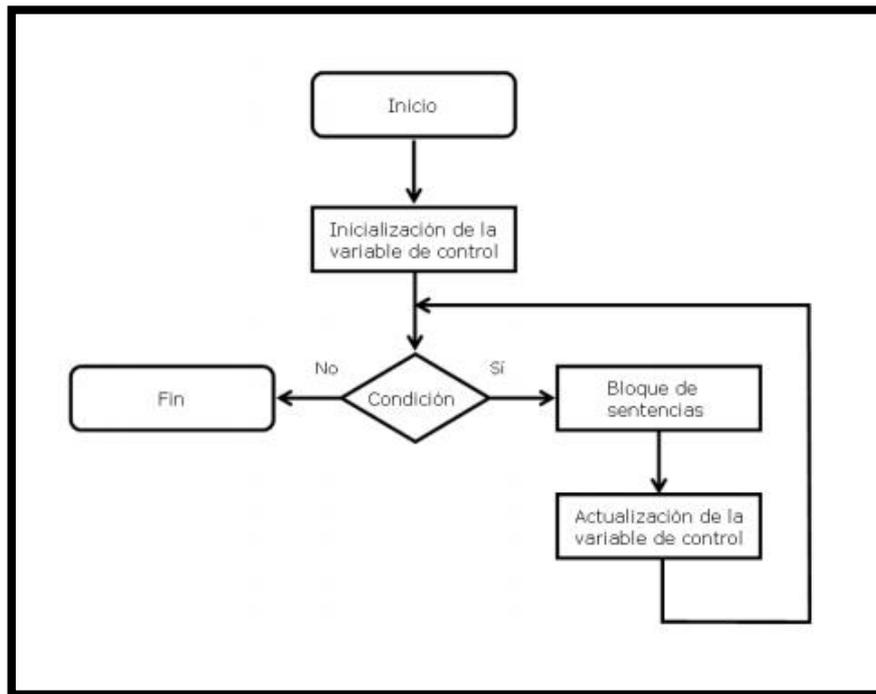


Ilustración 27 - Funcionamiento de la instrucción FOR (Ladron de Guevara, 2014).

8.3.4 - Operadores adicionales

A continuación se incluye una nueva lista de operadores especiales, cuya utilización es muy frecuente para establecer condiciones más complejas, tanto para su uso con **IF**, como con **DO WHILE** o **FOR**:

Tabla 17 - Operadores especiales para su uso en condicionales (Blanco, 2002).

OPERADOR	NOMBRE	DESCRIPCIÓN
And	Conjunción	A And B // Devuelve TRUE cuando se cumplan a la vez A y B.
Not	Negación	Devuelve el valor contrario de una expresión.
Or	Disyunción	A or B // Devuelve TRUE cuando se cumpla A o B o ambas.
AndAlso	And cortocircuito	A AndAlso B // Devuelve TRUE si ambas se cumplen. Si la primera es FALSE, no evalúa la segunda.
OrElse	Or cortocircuito	A OrElse B // Devuelve TRUE si una se cumple. Si la primera es TRUE, no evalúa la segunda.

Con esta ampliación sobre los conceptos de operadores ya se ha preparado a los alumnos para comenzar con la resolución de ejercicios prácticos.

Pensamos que el siguiente programa de ejemplo, explicado por el docente, puede ayudar a entender mejor el funcionamiento de alguna de estas instrucciones:

Ejemplo 1 – taller nº 3:

Se quiere hacer un programa que pregunte al usuario si desea salir o continuar. Si escribe SÍ o ``SI'', el programa mostrará una serie de cuatro mensajes, visualizando en ellos los números 1, 2, 3 y 4 respectivamente. En caso de escribir NO el programa no hará nada más. Por último, si no se da ninguno de los anteriores escenarios, el programa emitirá el mensaje ``No se ha introducido ni sí ni no''.

```
Dim siono As String
Dim contador As Integer
contador = 1
siono = Application.InputBox(prompt:="Continuar: SI - Salir: NO", Type:=2)
If siono = "SI" or siono = "SÍ" Then
    Do While contador < 5
        MsgBox "" & contador & ""
        contador = contador + 1
    Loop
ElseIf siono = "NO" Then
    MsgBox "Has salido del programa"
Else
    MsgBox "No se ha introducido ni sí ni no"
End If
```

En relación con el ejemplo anterior, el docente puede comentar que el bucle también podría ser conseguido mediante la instrucción **FOR**.

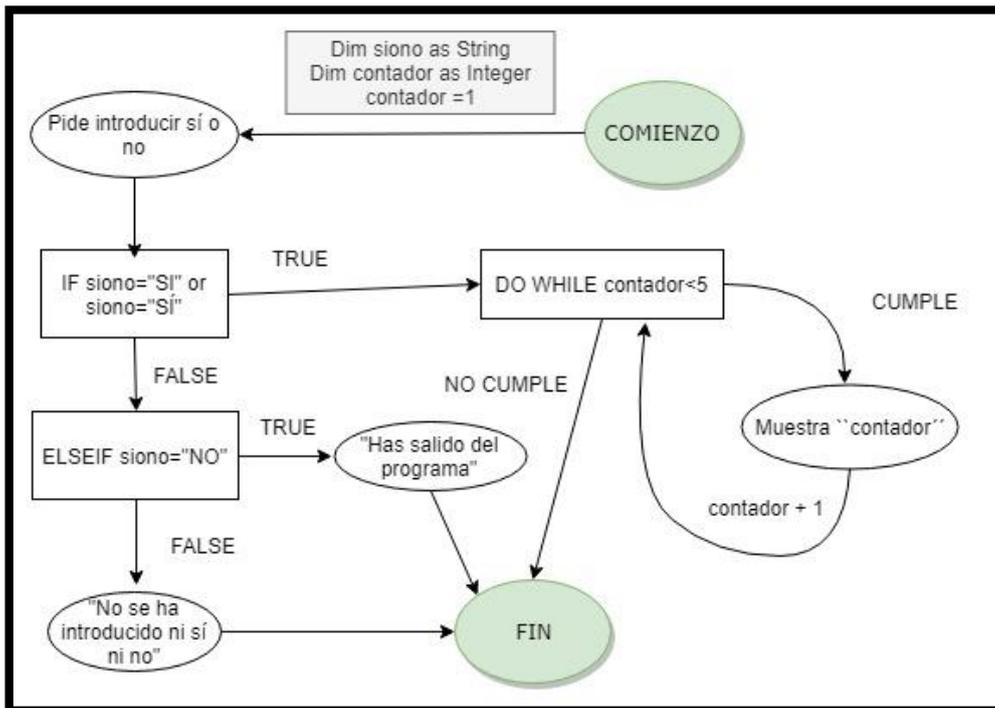


Ilustración 28 - Diagrama del ejemplo 1 - taller nº 3 (elaboración propia).

El proceso que sigue el programa es:

- 1- Declaración de las variables contador y de texto.
- 2- Entrada en la instrucción **IF**.
 - a. Si elige SÍ o SI entra en el bucle, tras cuatro veces, sale.
 - b. Si elige NO, muestra un mensaje donde anuncia la salida del programa.
 - c. Si elige alguna otra opción, aparece *``No se ha introducido ni sí ni no''*.

Para finalizar dicho taller, se propone un ejercicio o actividad con el fin de que sea realizado por los alumnos.

Actividad 1 – taller nº 3:

Se pide crear un programa que solicita un número al usuario, hasta que introduzca un número que sea positivo. En tal caso, si este número es mayor de 100, mostrará sólo la parte entera truncada. Si por el contrario es menor que 100 mostrará también la parte decimal. Finalmente escribirá el número exacto introducido en la celda B2.

```
Dim numero As Double
Dim entero As Integer
numero = 0
Do While numero <= 0
numero = Application.InputBox(prompt:="Introduce un número mayor que 0",
Type:=2)
Loop
If numero < 100 Then
MsgBox "El número introducido es:" & numero & ""
Cells(2, 2) = numero
Else
entero = numero
MsgBox "El número introducido es:" & entero & ""
Cells(2, 2) = entero
End If
```

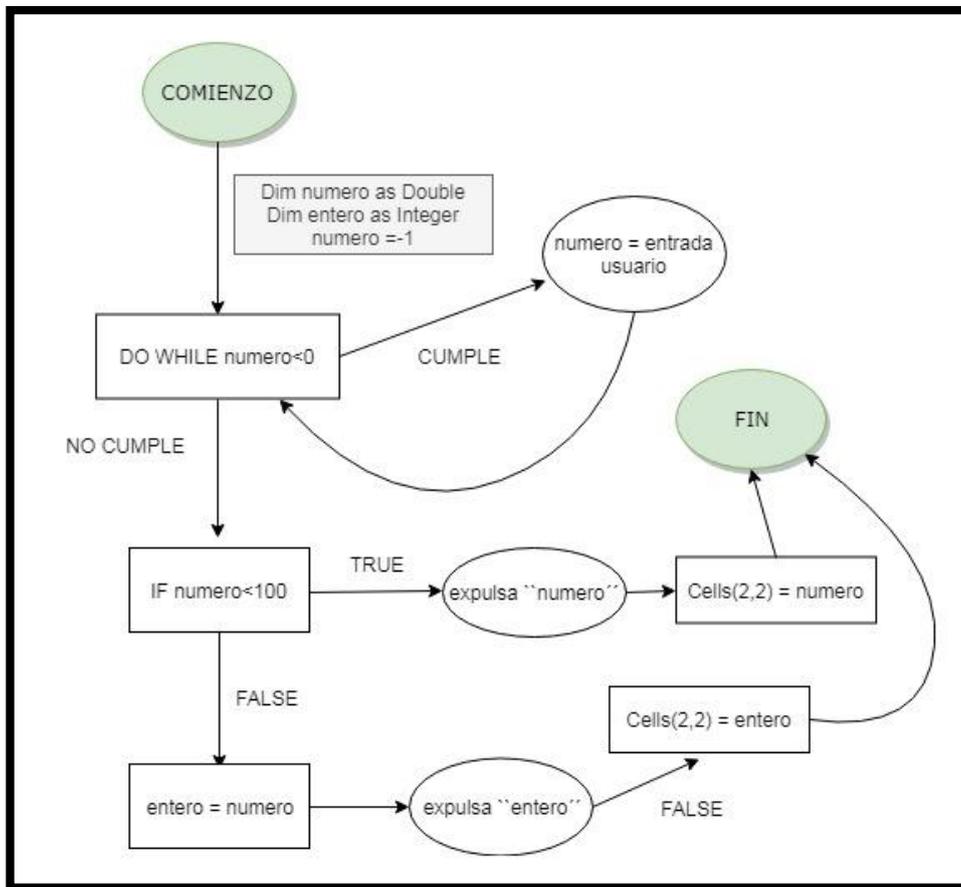


Ilustración 29 - Diagrama de la actividad 1 - taller nº 3 (elaboración propia).

El funcionamiento correcto del programa es:

- 1- Declaramos las variables **numero** como *Double* y **entero** como *Integer*.
- 2- Asignamos un valor negativo a la variable **numero**, para entrar en el siguiente **DO WHILE** al menos una vez.
- 3- El bucle **DO WHILE** se repite cada vez que la variable introducida tenga valor cero o negativo.
- 4- El programa solicita una variable, si es negativa, se repite por el bucle.
- 5- El programa entra en **IF** evaluando la variable **numero**.
 - a. Si **numero** < 100 lo expulsa por pantalla y almacena en la celda B2.
 - b. Si el **numero** >100 lo almacena en una variable tipo *Integer* y lo muestra por pantalla.

8.4 - TALLER N° 4: OBJETOS VBA EN EXCEL

En este taller se presentarán **objetos** de Excel que están íntimamente ligados a la programación en VBA. Son **objetos** que pueden incorporarse a la hoja de Excel habitual, y que nos ayudan a realizar algunas acciones, o bien ejecutan cierta parte de un código al realizar determinada acción sobre ellos, como clicar, o seleccionar otro objeto distinto.

Además, en el taller se mostrará un total de cuatro ejemplos, sin embargo, no se realizarán actividades. Esto es debido a la gran cantidad de conceptos nuevos que son presentados. El taller que viene a continuación de éste (que hemos denominado practicum) será usado exclusivamente para realizar actividades (un total de dos) englobando conceptos del actual taller, y de los anteriores.

Objetivos:

-Conocer parte de los objetos orientados a programación presentes en la herramienta Microsoft Excel.

-Ser capaz de elegir el objeto programable más adecuado para la solución de cada problema en particular.

-Hacer un uso inteligente y eficiente de las ordenes de programación asociadas a dichos objetivos.

Espacio físico y agrupamiento:

Se requiere que la sesión sea llevada a cabo en el aula de informática. Será realizada de manera individual, pudiendo los alumnos compartir por parejas un ordenador en caso de ser necesario. Dichos ordenadores deberán estar dotados con el paquete Microsoft Office, para que los alumnos puedan seguir las instrucciones del docente mediante el apartado Visual Basic de Excel, y tomar apuntes usando Microsoft Word.

8.4.1 - Botones

Un **botón** en Excel es un elemento que tiene una serie de líneas de códigos asociados a él, las cuales se ejecutarán, normalmente, cuando el usuario haga clic en el botón, aunque otras opciones son también posibles.

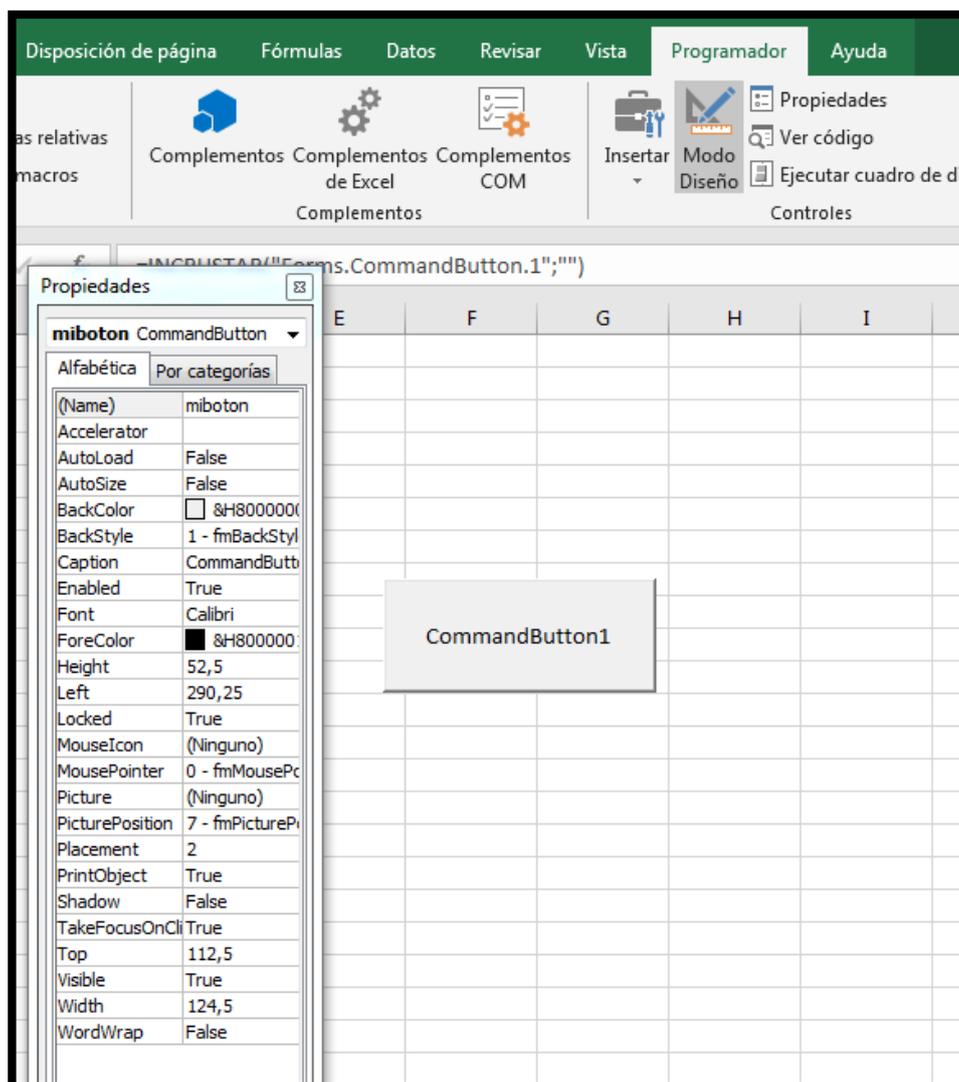


Ilustración 30 - Objeto botón "miboton" en Excel (elaboración propia).

Para añadir un botón, debemos buscar dentro de la pestaña programador de Excel, en el menú ``Insertar`` – ``Controles ActiveX`` (Salvador Ullauri & Sandoval Zambrano, 2015). Escogiendo el elemento botón, lo ubicamos en algún lugar de nuestra hoja. Una vez hecho esto, veremos que el modo diseño de Excel se encuentra activado. Mientras estemos en modo diseño (botón de modo diseño pulsado en la barra de menú de la pestaña programador) se puede realizar la edición del código asignado a cualquier elemento, haciendo doble clic en él. Además pulsando con el botón derecho, también se puede acceder a las distintas propiedades del objeto (nombre, forma, color, etc.).

Como ejemplo, se dispone del botón de nombre **miboton** presente en la imagen anterior.

8.4.2 - Listas

El objeto **lista** consiste en un desplegable que al ser seleccionado muestra una lista de elementos de entre los cuáles el usuario puede elegir uno de ellos, quedando éste seleccionado.

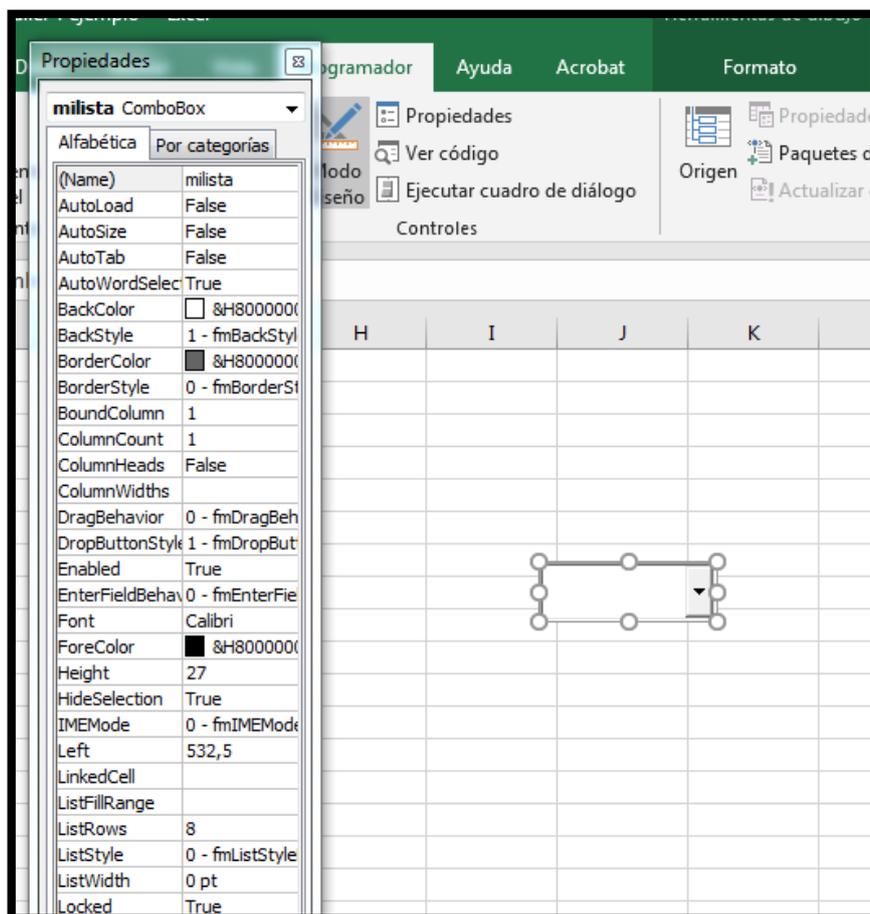


Ilustración 31- Objeto lista ``milista`` (elaboración propia).

Cabe mencionar que este elemento seleccionado se convertirá en el valor del objeto lista. Así, VBA reconocerá el objeto **lista** con el nombre proporcionado y su valor será igual al elemento seleccionado en dicha lista.

Como podemos ver en el ejemplo de la imagen adjunta, hemos llamado a nuestra lista **milista**. Esto quiere decir que aludiendo a dicha palabra a lo largo del código será igual que si escribiéramos el valor almacenado en **milista**. A continuación clarificaremos la última frase con una serie de ejemplos.

Tabla 18 - Comandos o funciones para trabajar con el objeto lista.

COMANDOS PARA ELEMENTO MILISTA	OPERACIÓN REALIZADA
<code>milista.AddItem "Prueba"</code>	Añade el elemento ``Prueba`` a la lista. Puede realizarse de manera continuada con más elementos.
<code>milista.AddItem Range("A1")</code>	Añade a la lista el elemento presente en la celda ``A1``.
<code>milista.AddItem Val(Range("A1")) * 5</code>	Añade a la lista el valor de la celda ``A1`` multiplicado por 5.
<code>milista.Clear</code>	Limpia totalmente los contenidos de la lista.
<code>Range("B1").Value = milista.Text</code>	Añadimos a la celda B1 como valor, el contenido de la lista previamente convertido a texto.

Ejemplo 1 - taller nº 4:

Se propone el siguiente ejercicio para combinar el uso de **botones** y **listas**. El docente va a proceder a crear un programa que contenga dos botones y una lista. El primer botón, añadirá el elemento de la celda ``A1`` a la lista, pero si es pulsado de nuevo, añadirá el elemento de la celda A2, si se pulsa otra vez, el de la celda ``A3``, y así sucesivamente. El segundo botón, simplemente reiniciará el proceso, y además limpiará la lista de cualquier elemento.

Tabla 19 - Código asociado a cada objeto en el ejemplo 1 - taller 4 (elaboración propia).

ELEMENTO	CÓDIGO ASIGNADO
Botón añadir	<i>milista.AddItem Range("A" & i + 1).Value i = i + 1</i>
Botón limpiar	<i>i = 0 milista.Clear</i>
Declaración de variable de proyecto	<i>Public i As Integer</i>

Este ejemplo es perfecto para explicar que en relación con una variable resulta importante no sólo el tipo, sino también el rango. En este caso, nuestra variable *i* tiene el prefijo *public* ¿Por qué?

La respuesta es que si la variable fuera declarada como parte del código asignado al botón, cada vez que éste fuera pulsado, ésta sería reiniciada, y empezaría a contar desde la celda uno de nuevo.

Podría también declararse como instrucción, nada más el libro de Excel sea abierto, sin embargo, tal y como se han explicado las variables hasta ahora, dicha declaración terminaría una vez el bloque de código fuese ejecutado correctamente. Para evitar eso se puede utilizar lo que se denomina **variable de proyecto**.

Tabla 20 - Rangos de variables en VBA.

Rango de variable	CÓDIGO	EXPLICACIÓN
Variable de procedimiento	<i>Dim variable as Tipo</i>	Se elimina al terminar el bloque de código que la contiene.
Variable módulo	<i>Dim/private variable as Tipo</i>	Solo disponible para los procedimientos del módulo.
Variable de proyecto	<i>Public variable as Tipo</i>	Disponible en todo el proyecto (archivo).

Se debe explicar la utilidad de disponer de una variable siempre declarada, pero también dejar claro que puede que en ocasiones, necesitemos que una variable sea ejecutada cada vez que arranca un determinado procedimiento y eliminada al finalizar éste.

El ejemplo del programa anterior es útil para explicar todo lo anteriormente mencionado. Es importante dar la explicación, después de dejar pensar a los alumnos sobre el problema antes mencionado.

Continuando con la explicación del código anterior, procederemos a profundizar sobre el código asociado a acciones sobre objetos. Dentro del editor de código de VBA existe una pestaña que nos permite seleccionar los distintos elementos, u objetos disponibles, a los que asociar determinadas líneas de código. Además de asociar líneas de código, se debe indicar la acción que debe ser realizada sobre dicho objeto para que el código asociado sea ejecutado. Este código es automáticamente escrito cuando se selecciona la opción en el desplegable.

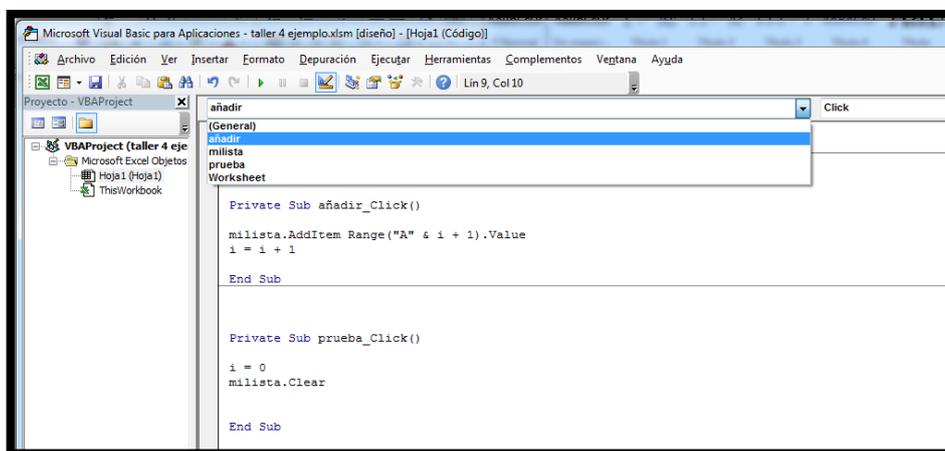


Ilustración 32- Selector de objetos del editor de código en Excel (elaboración propia).

En la imagen anterior se puede ver claramente que tenemos un total de 4 objetos. Los tres que hemos creado anteriormente, con código asociado en función de si hacemos clic en el objeto, deslizamos el cursor sobre el objeto, etc. Pero también se puede apreciar que el cuarto objeto es uno que no ha sido creado por el usuario, llamado "Worksheet". Dicho objeto es el propio libro de Excel o archivo, y también puede asociarse código al realizar una determinada acción sobre él, por ejemplo, abrirlo. Por último, todo el código asociado a (General) sucederá siempre que el archivo Excel esté abierto, pero es un tema sobre el que no entraremos en materia en esta ocasión.

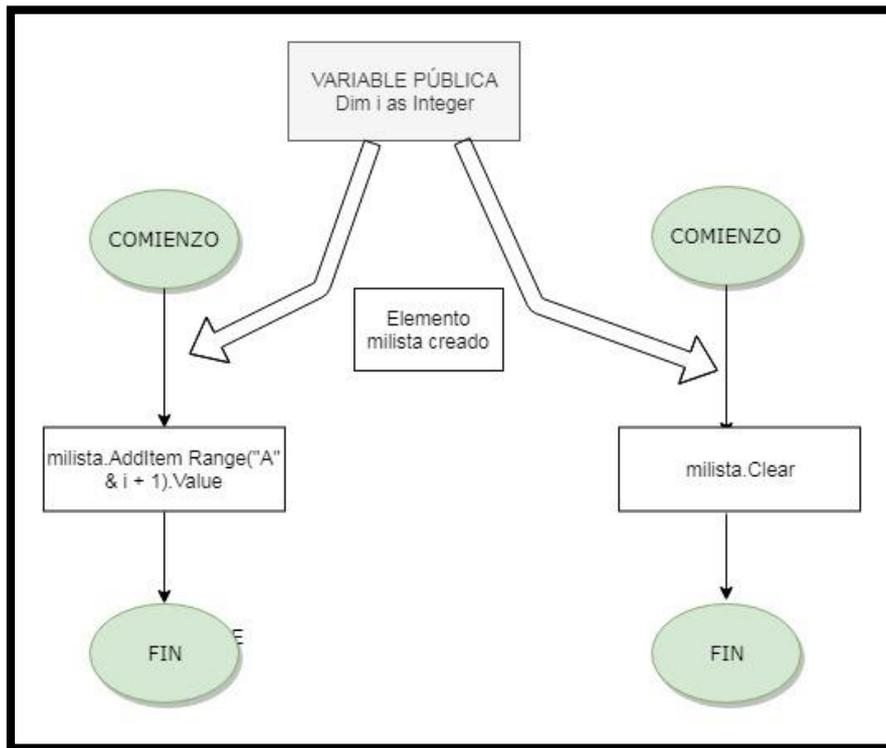


Ilustración 33 - Diagrama del ejemplo 1 - taller nº 4 (elaboración propia).

Ejemplo 2 – taller nº 4:

A continuación se muestra un ejemplo de código muy sencillo para comprender el uso del objeto **Worksheet**. Se va a generar un programa de modo que tan pronto como se ejecute el fichero Excel nos pida ingresar una secuencia de números, y se repita siempre que ese conjunto de números no coincida con la contraseña: 12345.

```
Private Sub Workbook_Open()  
    Dim mensaje As String  
    Dim contraseña As String  
    contraseña = "12345"  
    Do While mensaje <> contraseña  
        mensaje = Application.InputBox(prompt:="Por favor, inserte la contraseña:",  
                                        Type:=2)  
    Loop  
End Sub
```

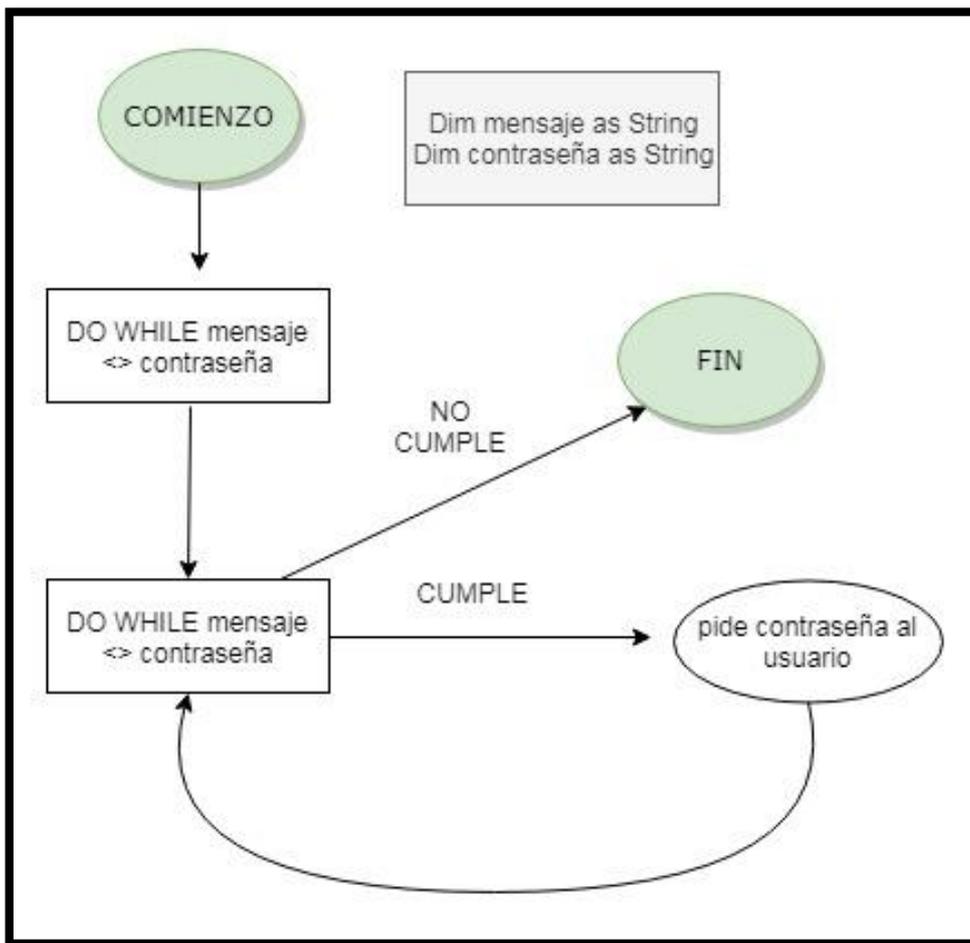


Ilustración 34 - Diagrama del ejemplo 2 – taller nº 4 (elaboración propia).

En la primera línea se establece que el código está dentro de una función llamada ``Workbook_Open``, lo cual es entendido por Excel como algo que sucede siempre que se abra el fichero. En este caso, merece la pena destacar que se puede volver a usar **variables procedimiento**, ya que una vez finalizado dicho bloque de código asociado a la instrucción ``Workbook_Open`` no nos interesa que sigan existiendo.

8.4.3 - Checkboxes.

Los **checkboxes** son muy interesantes debido a sus múltiples posibilidades de uso en las instrucciones **IF**. Son un tipo de objeto cuya única función es permanecer activados o desactivados. En el primer caso, el valor de dicho objeto tendrá asociado por defecto ``True``. En caso de permanecer desactivado, el valor será ``False``.

Esto recuerda a las **variables** de tipo *Boolean* mencionadas en el primer taller.

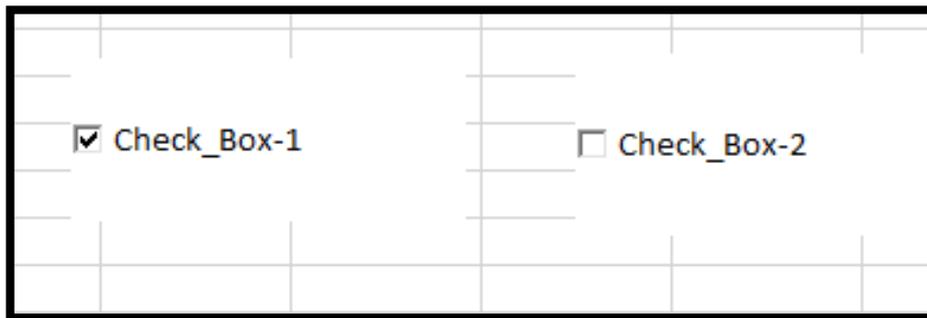


Ilustración 35 - Ejemplo gráfico de checkbox (elaboración propia).

Seguidamente se propone un ejemplo sencillo que ayuda a entender el funcionamiento de este nuevo objeto.

Ejemplo 3 – taller nº 4:

Se debe crear un programa con dos objetos tipo **checkbox** y un **botón**. Cuando se pulse el botón, el programa deberá mostrar por pantalla un mensaje indicando cuántos de esos dos **checkboxes** se encuentran activados.

```
Private Sub comprobar_Click()  
  
If CheckBox1 = True And CheckBox2 = True Then  
  
MsgBox "Ambos check_box están activados"  
  
ElseIf CheckBox1 = False And CheckBox2 = False Then  
  
MsgBox "Ambos check_box están desactivados"  
  
ElseIf CheckBox1 = True And CheckBox2 = False Then  
  
MsgBox "El Check_box 1 está activado, y el Check_box 2 está desactivado"  
  
ElseIf CheckBox1 = False And CheckBox2 = True Then  
  
MsgBox "El Check_box 1 está desactivado, y el Check_box 2 está activado"  
  
End If  
  
End Sub
```

Con este ejercicio, se pretende también repasar la instrucción condicional **IF**.

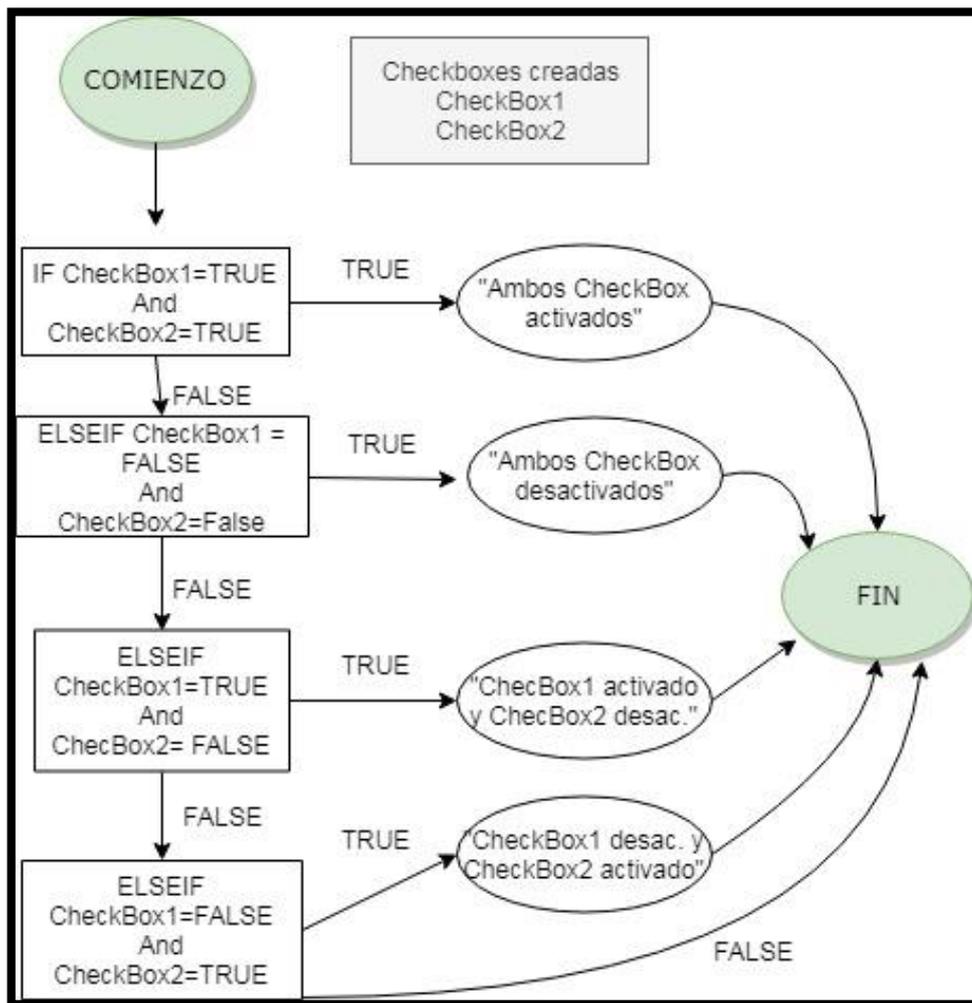


Ilustración 36 - Diagrama del ejemplo 3 - taller nº 4 (elaboración propia).

8.4.4 - Botones de opción.

Este tipo de objeto es muy parecido, en cuanto al funcionamiento, a **checkbox**. Sin embargo, existe una diferencia importante entre ellos. Los **botones de opción** pueden estar activados o desactivados, pero sólo uno de todos los usados en el documento Excel podrá ser el activo.

Por ejemplo, si disponemos de 5 **botones de opción**, siempre uno de ellos estará activado (TRUE), debido a las características de dicho objeto, mientras que los otros cuatro estarán desactivados (FALSE). Si activamos otro **botón de opción** distinto, automáticamente, el que presentaba la condición de activo anteriormente, pasará a estar desactivado. Esto puede apreciarse mejor en la imagen siguiente.

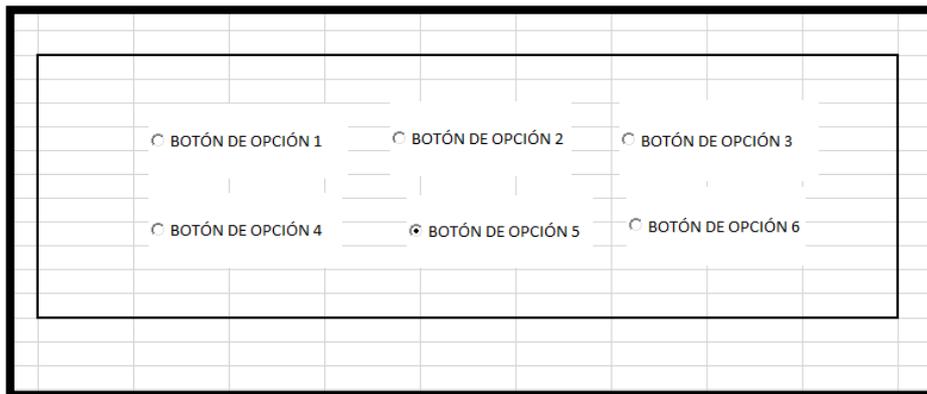


Ilustración 37 - Ejemplo de botones de opción (elaboración propia).

Para comprender mejor las diferencias en el uso de **botón de opción** y **checkbox** se planteará el siguiente ejercicio a modo de ejemplo:

Ejemplo 4 – taller nº 4:

Se pide crear una hoja de Excel que nos permita introducir por pantalla dos cifras. Dichas cifras serán sumadas, restadas, multiplicadas o divididas, en función de la elección del usuario. Posteriormente, tanto el resultado como las cifras introducidas deben de ser visualizadas por pantalla.

```
Private Sub empezar_Click()  
  
Dim numerouno As Double  
  
Dim numerodos As Double  
  
If sumar = True Then  
  
numerouno = Application.InputBox(prompt:="El primer número es:", Type:=2)  
numerodos = Application.InputBox(prompt:="El segundo número es:", Type:=2)  
  
MsgBox "Texto " & numerouno + numerodos & " texto "  
  
ElseIf restar = True Then  
  
numerouno = Application.InputBox(prompt:="El primer número es:", Type:=2)  
numerodos = Application.InputBox(prompt:="El segundo número es:", Type:=2)
```

```
MsgBox "Texto " & numerouno - numerodos & " texto "  
  
ElseIf multiplicar = True Then  
  
numerouno = Application.InputBox(prompt:="El primer número es:", Type:=2)  
numerodos = Application.InputBox(prompt:="El segundo número es:", Type:=2)  
  
MsgBox "Texto " & numerouno * numerodos & " texto "  
  
ElseIf dividir = True Then  
  
numerouno = Application.InputBox(prompt:="El primer número es:", Type:=2)  
numerodos = Application.InputBox(prompt:="El segundo número es:", Type:=2)  
  
MsgBox "Texto " & numerouno / numerodos & " texto "  
  
Else  
  
MsgBox "No ha marcado ninguna de las opciones"  
  
End If
```

Hay que destacar una serie de detalles en dicho código:

- Dentro de **IF** se ha tenido en cuenta la posibilidad de que aún no se haya seleccionado ningún **botón de opción**. En ese caso, se mostrará un mensaje por pantalla indicándonoslo.
- Es importante que las **variables** usadas permitan el uso de decimales en ellas, por ejemplo las de tipo *double*, para mejorar los resultados mostrados por el programa, ya que las de tipo *integer* truncarían los resultados.
- Las **variables** pueden ser de **procedimiento**, ya que una vez ejecutado el código asociado al botón "empezar" no es necesario que éstas sigan declaradas en la memoria (al pulsar el botón de nuevo podrían ser inicializadas sin problema).

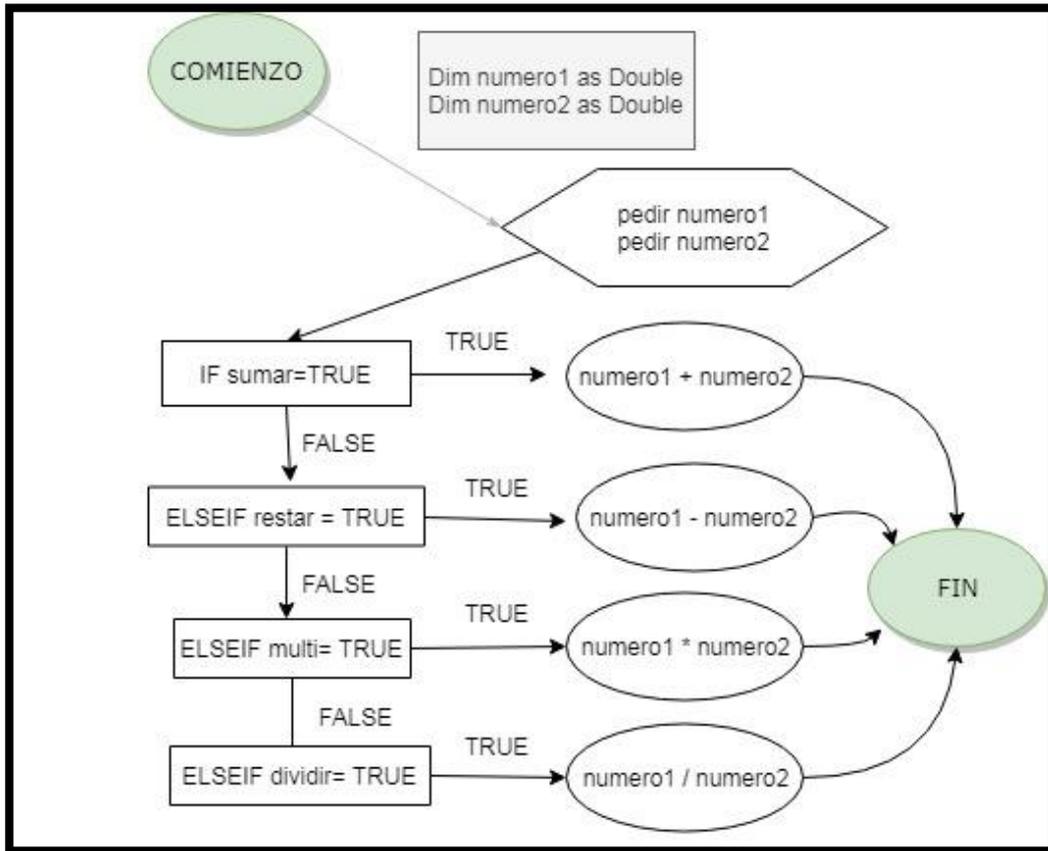


Ilustración 38 - Diagrama del ejemplo 4 – taller nº 4 (elaboración propia).

8.5 - TALLER Nº 5: PRACTICUM

Como ya se ha mencionado anteriormente, este taller de programación tendrá un carácter fundamentalmente práctico. Así, se propondrá la realización de ejercicios que requieran el diseño y elaboración de código VBA por parte de los alumnos, adecuando a cada uno de los problemas planteados.

El taller servirá, además, para atender posibles dudas y cuestiones que planteen los estudiantes sobre lo tratado en todos los talleres de programación anteriores.

Objetivos:

-Asentar los conocimientos tanto teóricos como prácticos presentados en los talleres previos: inicio a la programación de VBA, variables (tipos y rangos), instrucciones comunes en programación, tipos de objetos programables en Excel, etc.

-Practicar y adquirir agilidad en la resolución de problemas que requieren de la creación de código VBA.

-Propiciar la utilización del razonamiento lógico por parte del alumno, contribuyendo, en particular, a mejorar sus capacidades en relación con este tipo de razonamiento.

-Solventar todas las dudas que pudieran haber sido generadas en talleres anteriores.

Espacio físico y agrupamiento:

Se requiere que dicha sesión sea llevada a cabo en el aula de informática. Será realizada de manera individual, pudiendo los alumnos compartir por parejas un ordenador en caso de ser necesario. Dichos ordenadores deberán estar dotados con el paquete Microsoft Office, para que los alumnos puedan seguir las instrucciones del docente mediante el apartado Visual Basic de Excel, y tomar apuntes usando Microsoft Word.

Descripción:

Con el fin de que el alumno afiance y practique lo visto sobre comandos y funciones de VBA, el profesor puede proponer las siguientes actividades.

Actividad 1 – taller nº 5:

Desarrollar un programa en Excel que, al pulsar un botón, seleccione el número escrito en la celda C3 del documento y, mediante un mensaje en la pantalla, muestre si dicho número es o no es primo.

El código resultante que plantea el docente debe ser el siguiente:

```
Private Sub comprobar_Click()  
  
Dim n As Double  
  
Dim numeroencelda As Integer  
  
Dim comprobarentero As Double  
  
Dim contador As Integer
```

Dim primo As Boolean

primo = True

numeroencelda = Range("C3").Value

comprobarentero = numeroencelda - Range("C3").Value

contador = 2

If comprobarentero <> 0 Then

"MsgBox "no es entero "

Else

Do While contador < numeroencelda

If numerocelda Mod contador <> 0 Then

contador = contador + 1

ElseIf numeroencelda Mod contador = 0 Then

*"MsgBox "El número " & numeroencelda Mod contador & " no
es primo "*

primo = False

End If

contador = contador + 1

Loop

If primo = False Then

MsgBox "El número " & numeroencelda & " no es primo "

Else

```
MsgBox "El número " & numeroencelda & " es primo "  
  
End If  
  
End If  
  
End Sub
```

Se debe valorar y analizar todas las soluciones planteadas por los alumnos, y se debe dejar claro ante toda la clase la existencia de diferentes soluciones válidas dentro del mundo de la programación.

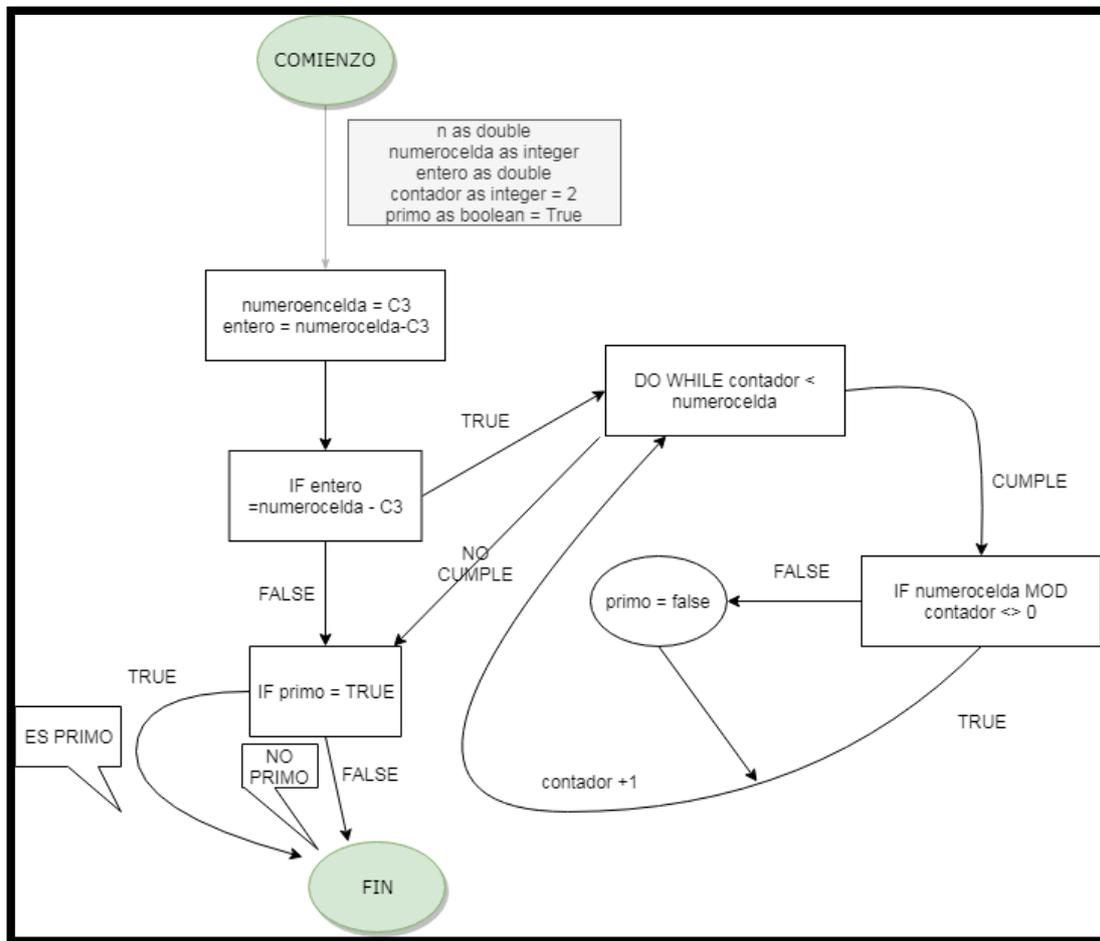


Ilustración 39 - Diagrama de la actividad 1 - taller nº 5 (elaboración propia).

Actividad 2 – taller nº 5:

Crear el código de un programa que al introducir un número natural por medio del teclado, lo almacene en una celda (B2), para después buscar todos los divisores de dicho número. Una vez encontrados, el programa deberá colocar en la columna B los divisores uno a uno, y mostrarlos por pantalla.

```
Private Sub cmd_divisores_Click()

    Dim numero As Integer

    Dim j As Integer

    j = 1

    Dim i As Integer

    i = 1

    Dim resto As Double

    numero = Application.InputBox(prompt:="El número para el que quiero ver los
    divisores es...", Type:=2)

    Range("B3") = numero

    Do While i < numero

    If numero Mod i = 0 Then

        Cells(5 + j, 2) = i

        MsgBox "Texto " & i & " texto "

        j = j + 1

    End If

    i = i + 1

    Loop
```

Como medio explicativo, el docente usará el siguiente diagrama de procesos:

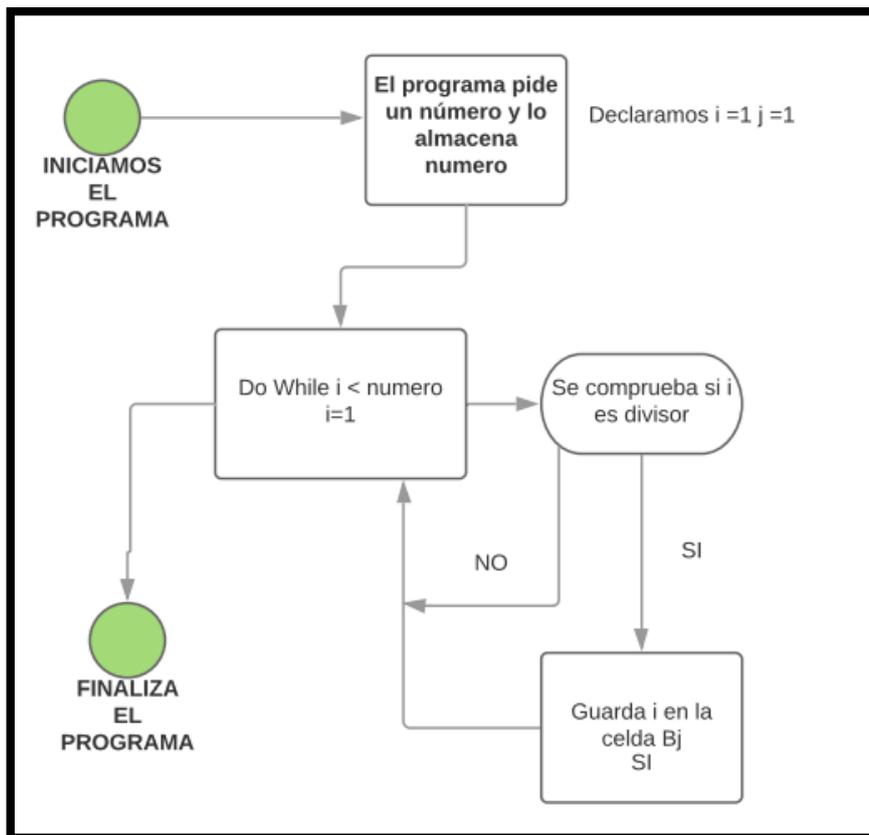


Ilustración 40 - Diagrama de la actividad 2 – taller nº 5 (elaboración propia).

8.6 - TALLER Nº 6: REVISIÓN DE LOS PROGRAMAS EXCEL USADOS EN CLASES DE MATEMÁTICAS

Hasta ahora se han utilizado archivos u hojas Excel para trabajar con contenidos de una temática (de 2º curso de Bachillerato de Ciencias) y para explicar nociones e ideas sobre programación. Este tipo de archivos han servido también para ilustrar con ejemplos y actividades lo anterior.

En este taller se pretende trabajar de una manera distinta. Así, el propósito no va a ser presentar nuevos conceptos, ni la elaboración de código por parte del alumnado. Ahora lo que se propone es que los alumnos tengan que interpretar o “descifrar” qué realiza un determinado código y que, además, hagan el correspondiente diagrama de procesos para explicar su funcionamiento.

Pensamos que puede ser útil para ello usar precisamente los archivos Excel que se han presentado en las sesiones Excel que hemos descrito anteriormente en este Trabajo

Fin de Máster. Así los estudiantes estarán familiarizados con el código de los programas, pues ya han sido usados y con conocidos.

Objetivos:

-Facilitar que el alumno pueda llegar a entender el funcionamiento de un código de programación más complejo incluso que el que es capaz de diseñar.

-Propiciar que los estudiantes puedan, después de haber entendido el funcionamiento de un código de programación, representar dicho funcionamiento en un diagrama de procesos.

-Poner de manifiesto un buen número de usos y aplicaciones de la programación, y todo lo que podemos lograr con ella.

-Intentar que el alumno sea consciente de todos los avances realizados durante el transcurso de los talleres.

-Propiciar que los estudiantes adquieran soltura en el manejo de la herramienta de grabación de macros.

Espacio físico y agrupamiento:

Se requiere que la sesión sea llevada a cabo en el aula de informática. Será realizada de manera individual, pudiendo los alumnos compartir por parejas un ordenador en caso de ser necesario. Dichos ordenadores deberán estar dotados con el paquete Microsoft Office, para que los alumnos puedan seguir las instrucciones del docente mediante el apartado Visual Basic de Excel, y tomar apuntes usando Microsoft Word.

Descripción:

Para las sesiones Excel realizadas en horas lectivas de la asignatura de matemáticas de 2º de Bachillerato de la modalidad de Ciencias, hemos preparado tres archivos de Excel, cada uno compuesto con una o varias hojas. Una vez escogido uno de esos archivos, el alumno deberá abrir el editor de código para poder visualizar éste en su totalidad, tras lo cual comenzará a leerlo detenidamente, con los apuntes que estime oportuno sobre su mesa. Después de leerlo las veces que sean necesarias, el alumno deberá comenzar a comprender el código. Cabe mencionar que no hay impedimento

alguno en que los alumnos ejecuten los comandos, o código de los archivos, para comprobar y recordar su funcionamiento, y probar con diferentes datos y resultados.

Tras este tiempo de reflexión se pide a los alumnos las siguientes tareas:

- 1- Escribir en un folio el procedimiento que creen que realiza el programa, para obtener el resultado a partir de los datos iniciales introducidos.
- 2- Realización de un esquema o gráfico similar a los presentados a lo largo del taller, donde se pueda percibir de manera esquemática, breve y simple el funcionamiento del programa (diagrama de procesos).

Estos documentos permitirán al docente tener información concreta acerca de si los alumnos han adquirido los conocimientos previstos sobre programación y VBA. Por otro lado, este último taller permitirá que los estudiantes tomen conciencia de sus propios avances en una rama de conocimientos tan interesante como la programación, motivándolos y animándolos a seguir formándose en ella.

9 - CONCLUSIONES Y LÍNEAS DE INVESTIGACIÓN FUTURAS

Tras la elaboración de esta memoria, para la que se han utilizado un buen número de fuentes y referencias bibliográficas, pensamos que se puede concluir que:

- Hoy en día la inclusión de las TIC tanto en el ámbito educativo, como fuera de él, resultan un hecho incontestable y forman parte de la vida cotidiana de los estudiantes de E.S.O. y Bachillerato.
- El uso de programas informáticos en el aula, con fines docentes, aumenta el interés y la motivación del alumno por las asignaturas de matemáticas, y favorece el trabajo en equipo y la convivencia en clase.
- Existen herramientas informáticas cuyo potencial docente suele ser desconocido para profesores y alumnos, por lo que, generalmente, no son usadas en el aula. Este es el caso de Microsoft Excel.
- Añadir Microsoft Excel a los diferentes recursos ya empleados habitualmente en la enseñanza proporciona ventajas adicionales para el alumno, que compensan el hecho de que Microsoft Excel pueda tener una ``apariencia`` o manejo menos sencillo y visual que otras herramientas comúnmente usadas. Estas ventajas resultan evidentes si se piensa en etapas

educativas posteriores a Bachillerato, o en ciertos ámbitos profesionales, donde el conocimiento del funcionamiento de Microsoft Excel resulta muy interesante, por lo extendido de su uso.

- Puesto que no abundan los trabajos y estudios sobre la aplicación de Microsoft Excel como recurso docente en asignaturas de matemáticas en Bachillerato, pensamos que este TFM tiene interés y aporta elementos novedosos en este sentido.
- En relación también con el punto anterior, el diseño de sesiones específicas (con ejemplos y actividades) sobre la utilización de Microsoft Excel, junto con la elaboración de material concreto sobre este programa (como hojas de cálculo y macros), para la docencia de contenidos de la asignatura ``Matemáticas II'' de 2º curso de Bachillerato (de la modalidad de Ciencias), nos parece que puede resultar de gran utilidad.
- La descripción de los talleres de programación realizada en este TFM, si se lleva a la práctica, puede servir para introducir a estudiantes de 2º de Bachillerato en el mundo de la programación, que está cada vez más presente en titulaciones universitarias y ámbitos laborales.

Como líneas de investigación futuras nos parecen de interés las siguientes:

- En primer lugar, creemos que resultaría muy interesante llevar a la práctica todo lo mencionado en las sesiones de Microsoft Excel para trabajar con contenidos de matemáticas de 2º de Bachillerato (de la modalidad de Ciencias), así como lo descrito sobre los talleres de programación.
- Con respecto al punto anterior, y una vez puestos en práctica en el aula las sesiones y talleres de programación mencionados, sería positivo realizar un estudio estadístico que permitiese valorar y extraer conclusiones de dichas experiencias. En este sentido, nos parecen adecuadas las encuestas que hemos preparado en los Anexos I y II para profesores y alumnos, respectivamente.
- Otra cuestión que consideramos de interés es la planificación y elaboración de sesiones con Microsoft Excel para la docencia en otras asignaturas de matemáticas de E.S.O. y Bachillerato.

- Finalmente, pensamos que sería adecuado plantear el diseño de talleres de programación similares para otros lenguajes de programación (distintos de VBA) que resulten asequibles e interesantes para estudiantes de Bachillerato. Esto podría realizarse de forma coordinada y conjunta con la materia ``Tecnologías de la Información y de la Comunicación`` (existente en Bachillerato en las modalidades de Ciencias, y de Humanidades y Ciencias Sociales), donde se estudian diferentes aspectos relacionados con la programación informática.

10 - Bibliografía

- Aguilera Almaguer, O., Aguilera Borjas, M., & Peña Cruz, E. (2011). Integración de la enseñanza de las TIC en la educación. *Cuadernos de educación y desarrollo*, 3(29), 1-15. doi:1989-4155
- Amelot, M. (2016). *VBA Excel 2016: Programación en Excel: Macros y lenguaje VBA*. Barcelona: Ediciones ENI.
- Arias Cabezas, J. M., & Maza Sáenz, I. (2006). Uso de las Tecnologías de la Información y la Comunicación en Matemáticas para la ESO y los Bachilleratos. *La gaceta de la RSME*, 223-243.
- Bernardo Carrasco, J., & Calderero Hernández, J. F. (2000). *Aprendo a investigar en educación*. Madrid: Rialp.
- Blanco, L. M. (2002). *Programación en Visual Basic.Net*. Madrid: Grupo EIDOS. Obtenido de <ftp://april.frm.utn.edu.ar/Libros%20y%20Material/VISUAL%20BASIC/04.%20Programaci%F3n%20en%20Visual%20Basic%20.NET/vbnet.pdf>
- Cabrera Delgado, J. M. (01 de Diciembre de 2015). Programación informática y robótica en la enseñanza básica. *Avances en supervisión educativa: Revista de la Asociación de Inspectores de Educación de España*(24), 1-26. doi:10.23824/ase.v0i24.17
- Deitel, P. J., & Deitel, H. M. (2008). *Como programar en Java*. México: Pearson Educación.
- Delgado Pineda, M. (1998). MAPLE en la Enseñanza Secundaria. *Gaceta de la Real Sociedad Matematica Española*, 1(1), 114-120.
- Duigou, C. (2016). *VBA Excel 2016. Cree aplicaciones profesionales. Ejercicios y correcciones*. Madrid: Ediciones ENI.
- ebriik.com. (2010). *Microsoft Office Excel 2010*. Obtenido de <https://www.uv.mx/personal/llopez/files/2013/03/Manual-Microsoft-Office-Excel-2010.pdf>
- Escribano Benito, J. J., & Martínez García, M. Á. (2007). Excel: una eficaz herramienta matemática para los alumnos de Ciencias Sociales. *Unión: revista iberoamericana de educación matemática*(9), 9-16.
- Galván Fernández, C. (Junio de 2003). Algunas actividades con GEUP. *Números: Revista de didáctica de las matemáticas*(54), 31-40. Obtenido de <http://www.sinewton.org/numeros/numeros/54/Articulo03.pdf>
- Giubergia, M. F., Graciela Socolovsky, S., & Ré, M. Á. (Junio de 2017). Incorporación de TICs a las clases de Análisis Matemático. *TE & ET*(19), 16-23.

- Hernández Gómez, E., Briones Peñalver, A. J., Serdeira Azevedo, P., & Medina Vidal, F. (2016). Geogebra y TIC en Matemáticas de enseñanza secundaria. *Anuario de Jóvenes Investigadores*, 9, 212-215.
- INTEF. (12 de Febrero de 2018). *Programación, robótica y pensamiento computacional en el aula*. Madrid: Ministerio de Educación, Cultura y Deporte.
- Iranzo, N., & Fortuny, J. M. (2009). La influencia conjunta del uso de geogebra y lápiz y papel en la adquisición de competencias del alumnado. *Enseñanza de las ciencias: revista de investigación y experiencias didácticas*, 27(3), 433-446. Obtenido de <https://www.raco.cat/index.php/Ensenanza/article/view/142075/332857>
- Ladron de Guevara, J. M. (2014). *Fundamentos de programación en Java*. Madrid: EME. Obtenido de <https://docs.google.com/file/d/0Byy7aUI9u4fBdnZjVnZOampmTjA/edit>
- Lannoy, L. (2011). *VBA Excel 2010: Desarrolle mediante un ejemplo una aplicación profesional*. Barcelona: Ediciones ENI.
- Le Guen, F. (2013). *Macros y lenguaje VBA: aprender a programar con Excel*. Barcelona: Ediciones ENI.
- Madrigal Muga, J. (2007). Descartes: un proyecto para ver y hacer matemáticas. *Matematicalia: revista digital de divulgación matemática de la Real Sociedad Matemática Española*, 3(1), Educación. Obtenido de http://www.matematicalia.net/index.php?option=com_content&task=view&id=354&Itemid=214
- Pifarré Turmo, M., & Sanuy Burgués, J. (Noviembre de 2000). El aprendizaje de estrategias de resolución de problemas con una hoja de cálculo. *SUMA*(35), 35-43.
- Quintana Perera, M. (1996). Derivadas con DERIVE. *Números: Revista de didáctica de las matemáticas*(28), 45-64. Obtenido de <http://www.sinewton.org/numeros/numeros/28/Articulo06.pdf>
- Quintana Perera, M. (1996). Estudio de límites y la continuidad de funciones usando DERIVE. *Números: Revista de didáctica de las matemáticas*(27), 59-80. Obtenido de <http://www.sinewton.org/numeros/numeros/27/Articulo05.pdf>
- Raviolo, A., Álvarez, M., & Aguilar, A. (2011). La hoja de cálculo en la enseñanza de la Física: recreando simulaciones. *Revista de enseñanza de la física*, 24(1), 97-107. Obtenido de <https://dialnet.unirioja.es/ejemplar/297370>
- Salvador Ullauri, L. A., & Sandoval Zambrano, F. (2015). *Introducción a la programación en Excel con Visual Basic Application*. Obtenido de <http://bibdigital.epn.edu.ec/bitstream/15000/11934/1/Introducci%C3%B3n%20en%20Excel%20con%20VBA.pdf>

11 - ANEXO I: Encuesta a los docentes

11.1 - Test para los docentes

El test proporcionado a los docentes será el siguiente:

PREGUNTA	SÍ	NO
¿Ha visto a los alumnos motivados con este planteamiento en las clases de matemáticas?		
¿Cree que Excel, aplicando la metodología descrita, es más útil que otros programas informáticos similares usados en la enseñanza?		
¿Considera que la participación en el taller de programación ha sido elevada?		
¿Piensa que la metodología empleada en las sesiones Excel para trabajar con contenidos de matemáticas ha fomentado el trabajo en equipo?		
¿Le ha resultado fácil adaptarse a este nuevo método en las sesiones de matemáticas?		
¿Le han resultado difíciles los conceptos trabajados en el taller de programación?		
¿Ha visto a los alumnos motivados en el taller de programación?		
¿Cree que las sesiones de matemáticas han ayudado a los alumnos a interiorizar los conceptos?		
¿Una vez subidas las hojas Excel a la plataforma, han sido usadas por los alumnos?		
¿Piensa que los conocimientos de programación eran asimilados de manera correcta por los alumnos?		
¿Considera un éxito las sesiones de Excel aplicado a matemáticas?		
¿Considera un éxito las sesiones de los talleres de programación?		
¿Cree que sería recomendable repetir la experiencia al año que viene?		

11.2 - Entrevista abierta

En esta entrevista se busca, más que realizar preguntas, permitir que el docente se exprese libremente sobre todo aquello, relativo a las actividades desarrolladas, que considere oportuno. Esto le permitirá también comentar y explicar la razón de su respuesta en aquellas preguntas del test en que considere que sólo "sí" o "no" no proporcionan una respuesta satisfactoria. En función de lo expresado por el docente, se le podrán plantear (de hecho, esto será conveniente) preguntas más concretas.

12 - ANEXO II: Test a los alumnos

12.1 - Test a los alumnos participantes en los talleres de programación

Dicha encuesta, únicamente será realizada por los alumnos participantes en los talleres de programación. Deberán asignar un valor del 1 (totalmente en desacuerdo) al 5 (totalmente de acuerdo) a cada uno de los enunciados.

PREGUNTA (taller de programación)	PUNTUACIÓN
¿Consideras que has aprendido mucho?	
¿Consideras útil lo que has aprendido?	
¿Tienes pensado trabajar en un ámbito laboral que tiene relación con lo que has aprendido?	
¿Consideras que se han explicado los contenidos de manera correcta?	
¿Te ha parecido interesante?	
¿Ha cambiado tu percepción sobre qué es la programación?	
Si pudieras volver al pasado... ¿volverías a apuntarte?	

12.2 - Test para todos los alumnos

Dicho test será realizado tanto por los alumnos que asistan a los talleres de programación, como por los que no, y su fin es obtener información sobre las sesiones de Excel aplicado a las matemáticas.

PREGUNTA (Excel aplicado a matemáticas)	SÍ	NO
¿Habías usado anteriormente Microsoft Excel?		
¿Consideras que has aprendido mucho?		
¿Consideras que te ha sido útil para repasar y afianzar conceptos?		
¿Has aprendido a usar mejor Microsoft Excel después de estas sesiones?		
¿Lo has usado para repasar ejercicios cuyos datos han sido generados aleatoriamente por el programa?		
¿Te ha parecido interesante el uso de Microsoft Excel en la clase de matemáticas?		
¿Piensas que este tipo de actividades contribuyen a aumentar la motivación y el interés del alumnado por la asignatura?		
Si pudieras elegir... ¿preferirías no haber tenido estas sesiones?		