UNIVERSIDAD DE VALLADOLID

**Álgebra, Análisis Matemático, Geometría y Topología**

# Fomento de la creatividad y el pensamiento abstracto a través de las matemáticas

**Trabajo Final del Máster Universitario de Profesor en Educación Secundaria Obligatoria y Bachillerato, Formación Profesional y Enseñanza de Idiomas. Especialidad de Matemáticas.**

**Alumno: Bernardo Martínez Celda**

**Tutor: Philippe Thierry Gimenez**

**Valladolid, junio 2022**

# Índice general

# Introducción

La creatividad es una competencia clave a desarrollar en el siglo XXI ya que el desarrollo económico del mundo actual depende cada vez más de la innovación. El pensamiento creativo contribuye a ver la realidad desde otra perspectiva, a generar descubrimientos y a innovar en procesos productivos [Sanz de Acedo, 2004].

Hoy en día hay gran interés en el desarrollo de la creatividad, tanto es así que en la realización de las pruebas PISA de primavera del 2022 se ha incluido como competencia innovadora el pensamiento creativo. Según puede leerse en el Instituto Nacional de Evaluación Educativa [INEE]:

> En este estudio se mide la evolución del conocimiento y las habilidades de los estudiantes en un contexto de proliferación de las tecnologías de la información y la comunicación, y su capacidad para responder a las demandas de un mundo en constante cambio, donde la innovación y la creatividad son elementos clave para la adaptación. [INEE, , s.f.]

Por otro lado, aprender a programar en estos niveles académicos puede aportar aspectos beneficiosos. Desarrolla el pensamiento abstracto, algo muy conveniente en matemáticas, también se trata de una forma de aprender interactiva, y además pueden empezar a adquirir de forma temprana una habilidad que, probablemente, en muchos casos necesitarán en un futuro.

En relación a esto último, hoy en día ya existen academias como Space Techies [1] que ofrecen talleres y cursos sobre robótica y programación para todas las edades, incluso para docentes. No cabe duda que el auge del sector tecnológico,

[1] https://www.spacetechies.com/

debido a los incesantes avances en Big Data e Inteligencia Artificial, se está traduciendo en una mayor demanda de conocimientos de programación, y del mismo modo que la globalización impulsó la necesidad de aprender nuevos idiomas, ahora quizá sea un momento oportuno para introducir a los estudiantes en el mundo de la programación.

Sin embargo, la programación es una herramienta muy poco explotada en esta etapa académica. Posiblemente se deba a que en algunos casos requiere formación complementaria por parte del profesorado, o que los centros no tenían medios digitales suficientes, o que el proceso de instalación de los programas requeridos fuese complejo. No obstante, hoy en día gracias a Internet tenemos acceso a una gran cantidad de recursos gratuitos sobre programación que nos pueden enseñar desde los conceptos más básicos, hasta los más complejos. También existen herramientas gratuitas como Google Colab o Scratch que nos permiten programar desde un navegador web como Google Chrome, por lo que no es necesario instalar nada ni tampoco es relevante tener un ordenador de última generación. Por lo tanto, si se tiene la voluntad por parte del profesorado, y el centro cuenta con los medios necesario, actualmente es mucho más sencillo iniciar a los alumnos en la programación.

El presente trabajo se ha planteado del siguiente modo: en primer lugar se establece un marco teórico sobre la creatividad, en segundo lugar se justifica por qué puede trabajarse la creatividad en las asignaturas de matemáticas teniendo en cuenta el currículo académico, en tercer lugar se proponen diferentes actividades para fomentar la creatividad tomando como punto de partida lo expuesto en el marco teórico, y por último se comparte brevemente la experiencia del autor durante su periodo de prácticas.

# Capítulo 1

# Marco teórico

En este capítulo se pretende dar respuesta a preguntas tales como: ¿qué es la creatividad?, ¿qué determina que una persona sea creativa?, ¿el potencial creativo es exclusivo de las personas inteligentes?, ¿cómo se puede trabajar la creatividad?

## 1.1. Definición del término *creatividad*

A modo de introducción, una de las muchas frases que se atribuyen al premiado físico Albert Einstein trata sobre la creatividad y dice lo siguiente: "La creatividad es ver lo que todo el mundo ha visto y pensar en lo que nadie había pensado". Por otra parte, y de forma más sencilla, la Real Academia Española define el término creatividad como: "facultad de crear", y a su vez define el verbo crear como: "producir algo nuevo". No obstante, ¿qué entienden los sociólogos y psicólogos por *creatividad*? A continuación se muestran varias definiciones dadas por algunos de los autores más influyentes en este campo.

"La creatividad, en sentido limitado, se refiere a las aptitudes que son características de los individuos creadores, como la fluidez, la flexibilidad, la originalidad y el pensamiento divergente." [Guilford, 1950]

"La creatividad es un proceso que vuelve a alguien sensible a los problemas, deficiencias, grietas o lagunas en los conocimientos y lo lleva a identificar dificultades, buscar soluciones, hacer especulaciones o formular hipótesis, aprobar y comprobar estas hipótesis, a modificarlas si es necesario además de comunicar los resultados.” [Torrance, 1965]

“La creatividad consiste en la producción de soluciones originales potencialmente aplicables a problemas novedosos, débilmente definidos, con una complejidad relativamente alta” [Lubart, 2001]

De esta forma, se han presentado varias definiciones del concepto creatividad. Si bien estas son distintas, se pueden identificar elementos comunes entre ellas como por ejemplo: solucionar, originalidad y flexibilidad.

Para una mejor comprensión de este trabajo, es conveniente revisar algunas de las aportaciones más relevantes que los estudiosos del tema han realizado en las últimas décadas. En concreto, cobra especial interés la teoría de Guilford pues se trata de una de las que más influencia ha tenido.

## 1.2. Los rasgos de la creatividad según Guilford

Guilford [1950], al igual que Torrance [1962], considera que el pensamiento divergente es una característica fundamental de la creatividad. De sus investigaciones se deduce que el pensamiento divergente tiene diferentes dimensiones, también conocidas como rasgos de la creatividad.

Para desarrollar esta sección se tomará como referencia tanto el trabajo del propio Guilford [Guilford, 1950], como el artículo Santos [1986], donde la autora analiza cómo ha evolucionado el concepto de pensamiento divergente en la teoría de Guilford.

En primer lugar cabe señalar las dos hipótesis con las que este autor trabaja:

1. El talento creador se presenta en grados distintos por toda la población.

2. El talento creador no se encuentra a modo de apéndice asociado a la inteligencia, sino que se postulan una una serie de rasgos[1] para explicar las

[1]Un rasgo es una forma de comportamiento distinguible y relativamente perdurable en la cual un

variedades conductuales a que da lugar.

Los rasgos que postula Guilford son los siguientes:

1. Sensibilidad a los problemas: actitud perceptual general que capacita a los individuos a darse cuenta de los inusual, lo raro, de las inconsistencias aparentes. Tal disposición ofrece al individuo numerosos problemas a resolver.

2. La fluidez: capacidad de dar muchas respuestas en un área de información determinada y en un tiempo dado. Se especifican diferentes tipos:

   *a*) Fluidez verbal: producción divergente de unidades simbólicas. Una forma de medirse consiste, por ejemplo, en enumerar la mayor cantidad de palabras que contengan una letra dada.

   *b*) Fluidez asociativa: producción divergente de relaciones semánticas. Podría medirse, por ejemplo, proponiendo diferentes adjetivos para completar símiles.

   *c*) Fluidez ideacional: producción divergente de unidades semánticas. Existe un test muy famoso para medir esta capacidad, se conoce en inglés como *Brick Uses* y consiste en dar diferentes usos a un ladrillo.

   *d*) Fluidez de expresión: producción divergente de sistemas semánticos. Explicar de formas distintas oraciones que contienen símiles es un buen recurso para medir esta capacidad.

3. Originalidad: se concibe de, al menos, tres modos distintos, en términos estadísticos se entiende como la infrecuencia de una respuesta, también puede entenderse en términos de asociaciones remotas, y también puede relacionarse con la calidad de la respuesta ante un problema dado.

4. Flexibilidad: se distinguen dos tipos, la espontánea, capacidad de introducir diversidad en las ideas producidas en una situación relativamente inestructu-

individuo difiere de otro. [Guilford, 1950] rada; y la adaptativa, capacidad de cambiar el camino que se está siguiendo para cumplir requisitos impuestos por las condiciones cambiantes.

5. Redefinición: capacidad de ofrecer nuevas interpretaciones o significados ante objetos familiares para darles un nuevo uso o sentido.

6. Evaluación: crítica del producto elaborado.

De este modo, el pensamiento creativo se presenta como un potencial, Guilford defiende que esta puede favorecerse mediante el entrenamiento. Ahora bien, en el desarrollo de la creatividad el ambiente tiene un papel clave. En la siguiente sección se recogen diversos autores que en sus estudios no solo tienen en cuenta las características personales sino también los factores contextuales como influyentes en el desarrollo del pensamiento creativo.

## 1.3. Teorías contextuales de la creatividad

Si uno quiere trabajar la creatividad, primero debe ser conscientes de los factores que intervienen en su desarrollo. El propósito de esta sección es presentar diferentes autores que han investigado qué factores ambientales tienen repercusión en el ejercicio del pensamiento creativo.

El desarrollo de esta sección se nutre del trabajo de Melero [2018].

### 1.3.1. Teresa Amabile

Para Amabile [1983] la creatividad es la elaboración de un producto o respuesta que es valorado como creativo por jueces apropiados para ello. El modelo planteado por Amabile incluye tres componentes que se consideran necesarias para lograr una producción creativa. Estas componentes son:

1. Las *habilidades creativas* incluyen el conocimiento de heurísticos[2] sobre la generación de ideas novedosas. Estas habilidades dependen de rasgos de per-

[2] Un heurístico sería definido como algún principio o regla general que ayuda en la búsqueda de una solución a un problema o en la realización de una tarea.

sonalidad como la autodisciplina, la habilidad para demorar la gratificación, la perseverancia ante la frustración, independencia, y ausencia de conformismo de pensamiento con respecto a las ideas sociales imperantes. Estas habilidades también se ven influenciadas por el entrenamiento.

2. Las *habilidades relevantes de dominio* incluyen información sobre hechos, principios, opiniones sobre varias cuestiones en un dominio específico, conocimiento de paradigmas para resolver problemas dentro del dominio, habilidades técnicas necesarias para el dominio y talentos especiales, que contribuyan a la productividad creativa dentro del dominio (como la habilidad de un compositor para escuchar los sonidos de todos los instrumentos musicales, necesarios para una composición musical a nivel de imaginación). De manera que, es necesario tanto un conocimiento familiar como factual del dominio.

3. La *motivación por la tarea* tiene dos elementos: las actitudes individuales hacia la tarea y la percepción individual de las razones por las cuales el individuo se compromete con la tarea en una determinada circunstancia. Además, se trata de una variable muy influenciada por las condiciones sociales y ambientales.

### 1.3.2. Mihály Csikszentmihalyi

Csikszentmihalyi [1983], autor de la Teoría General de Sistemas Creativos, considera que la creatividad no sólo se encuentra en la persona, sino que la creatividad es producto de la interacción entre el pensamiento de la persona y su entorno sociocultural. Este autor entiende la creatividad como la interacción de tres factores:

1. La *persona*, cuyas características más definitorias son estar abierto a nuevas experiencias, la motivación, la persistencia, la fluidez de ideas o la flexibilidad de pensamiento.

2. El *dominio de los conocimientos* de una determinada área. Si no se tiene un conocimiento estructurado es muy difícil introducir cambios relevantes en el mismo, principalmente porque no sería posible identificar problemas e inconsistencias que solucionar introduciendo alternativas.

3. El *campo*, al igual que Amabile, Csikszentmihalyi defiende que quién determina si una idea es creativa debe ser un comité de expertos.

Si, como defienden estos autores, determinar si una idea o producto es creativo debe estar en manos de un comité de expertos, entonces la siguiente pregunta surge de modo natural: ¿una persona puede ser creativa en un campo concreto y no serlo en otro distinto? En la siguiente sección se aborda esta cuestión.

## 1.4. Dominio de la creatividad

Debido a la falta de consenso en la definición de creatividad, esta ha sido abordada desde perspectivas psicológicas muy distintas. Por una parte, existen teorías que hablan de la creatividad como una habilidad de dominio general, es decir, quien sea creativo en una determinada área de conocimiento, lo será en el resto. Y, por otro lado, también existen otros teóricos que opinan que la creatividad es una habilidad de dominio específico, en cuanto a que está ligada, en cada individuo, a una determinada área de conocimiento.

En este sentido, recuperando de la sección 1.1 la definición que daba Torrance, se deduce que si un individuo es creativo, lo será con independencia del área de conocimiento del que parta el problema.

En contraposición a Torrance se encuentra Baer, quien defiende que la creatividad es de dominio específico [Baer, 2010]. Baer es uno de los autores que más investigaciones ha generado al respecto, en sus estudios encuentra evidencias empíricas de la baja correlación en el rendimiento creativo entre tareas de diferentes dominios [Melero, 2018].

Baer [2015] identifica cuatro componentes necesarios para ser creativo: inteligencia, motivación, habilidades de pensamiento y experiencia. Propone como ejemplo la situación de un cocinero que tiene una estantería repleta de especias (habilidades) y además sabe cómo hacer uso de ellas (conocimiento y experiencia), cuando debe preparar un plato (cuando se enfrenta a un problema o reto) no siempre utilizará las mismas especias del mismo modo, sino que por medio de su experiencia, conocimiento y creatividad determinará qué especia le va mejor a cada plato. Es decir, es necesario conocer cuáles de nuestras habilidades funcionan mejor cuando nos enfrentamos a un determinado reto. En conclusión, un individuo puede ser creativo en varios dominios pero ello requiere que posea un gran bagaje de habilidades y conocimientos de diversos dominios.

En este debate sobre el dominio de la creatividad surge una tercera postura que propone un modelo híbrido. Plucker y Beghetto [2004] entienden que la creatividad tiene características tanto de dominio específico como de dominio general. Si bien mantienen la postura de Amabile y Csikszentmihalyi, vista en la sección 1.3, de que serán los jueces de una disciplina, quienes determinarán qué debe poseer un producto para que sea considerado como creativo; postura que invita a concebir la creatividad como dominio específico, también destacan que la motivación y el compromiso son necesarios para desarrollar una actividad creativa, y que estos rasgos se manifiestan en las personas creativas con independencia del dominio.

En el desarrollo de este trabajo se adoptará esta última postura, el modelo híbrido.

## 1.5. Fomento de la creatividad

La gran variedad de enfoques teóricos que se han expuesto hasta ahora, consecuencia de la falta de consenso en la definición de creatividad, se traduce en una enorme diversidad de programas o ejercicios de fomento de la creatividad, muy heterogéneos y no siempre contrastados o comparables.

Por este motivo, se ha decidido acudir a un metaanálisis realizado en 2004 [Scott et al., 2004] que intenta dilucidar si es posible fomentar la creatividad, y en qué condiciones. Sus principales conclusiones son que sí es posible, que no son necesarios unos rasgos de personalidad o habilidades de partida, que es posible entrenar la creatividad en una gran variedad de ambientes y edades, y que el programa de entrenamiento debe estar diseñado en base a los procesos cognitivos que pretende desarrollar, ya que no todos los programas son igual de efectivos.

El metaanálisis se basa en 70 estudios que superan un cribado según criterios de calidad científica (validez interna y externa). Se compararon sus enfoques metateóricos, sus contenidos y métodos de impartición, y se midió su efectividad según contribuyesen a la mejora de cuatro variables distintas, que intentan abarcar los principales enfoques teóricos:

1. Pensamiento divergente: generar muchas soluciones.

2. Resolución creativa de problemas: generar soluciones funcionales.

3. Desempeño creativo: diseño de productos creativos.

4. Actitudes y comportamientos hacia el proceso creativo.

Tras el análisis estadístico, los autores concluyen que un programa de entrenamiento efectivo debe exponer heurísticos y técnicas de manejo de la información. El programa debe incluir ejemplos, y sobre todo ejercicios prácticos, y debe ser sostenido en el tiempo. Este tipo de entrenamiento es efectivo para todo tipo de personas, porque no requiere un perfil específico, sino que da herramientas para que cualquiera encauce la información que tiene a generar y evaluar nuevas ideas. Es decir, es un entrenamiento de procesos cognitivos específicos.

A continuación se anuncian algunos ejemplos de actividades que trabajan procesos cognitivos específicos: la identificación de problemas, combinación conceptual, creación de analogías, pensamiento crítico, identificación de restricciones y concatenación de conceptos (pensamiento convergente). No serían efectivos, por ejemplo, los ejercicios abiertos de exploración.

Asimismo, se presentan dos ejemplos de programas efectivos: el *Purdue Creative Training* [Feldhusen et al. , 1970] basado únicamente en la Teoría del Pensamiento Divergente, o el *Creative Problem-Solving Program* de Osborn y Parmes [Isaksen y De Schryver, 2000] basado en la teoría de la Resolución Creativa de Problemas.

Una vez conocidas las estrategias que contribuyen a mejorar la capacidad creativa, y dada la influencia que tiene el contexto, como se vio en la sección 1.3; es conveniente estudiar la creatividad ubicándola en el contexto de interés para este trabajo, el aula. Éste será el tema a tratar en la siguiente sección.

## 1.6. La creatividad en las aulas

La creatividad no sólo viene determinada por el individuo, sino que depende de varios factores. Uno de ellos es el ambiental. Este, a su vez, se divide en dos: el físico y el social. Si hablamos del primero, un ambiente que ofrece gran abundancia de material y medios, provoca inspiración, pensamiento divergente y asociaciones creativas. Si hablamos del segundo, la sociedad y el entorno cercano forjan la personalidad del individuo, sus valores y puede ayudar en el proceso mismo de creatividad. Es en este último caso, es cuando el papel del docente juega un rol determinante [Sanz de Acedo, 2007].

Torrance estudia las causas del desarrollo de la creatividad en los niños, especialmente en los ambientes educativos, analizando las causas que reprimen la creatividad y buscando las características del niño creativo. De sus investigaciones concluye que los niños creativos son vistos como ‘atípicos’, tanto por sus profesores como por sus compañeros de clase, por lo que generalmente, tanto sus maestros como sus compañeros de una forma u otra, los reprimen. En este sentido, realizó un estudio de tipo longitudinal que duró 12 años, durante los cuales aplicó una prueba a 392 alumnos de nivel secundaria, logrando demostrar que los niños creativos son más exitosos profesionalmente y se desempeñan en mejores trabajos. [Melero, 2018].

En Tan et al. [2016] se analizan los resultados de una investigación llevada a cabo en Singapur donde se compara el desarrollo de la creatividad en dos grupos distintos. Uno de los grupos sigue un programa de estudios cuyo currículo está orientado a realizar un examen para obtener el certificado escolar, mientras que el otro grupo participa en un programa mucho más flexible que les permite saltarse dicho examen. El currículo del primer grupo es estándar para todos los centros educativos, mientras que el currículo del segundo grupo goza de un mayor grado de libertad, lo que permite optimizar el tiempo liberado, al no tener que presentarse los alumnos participantes al examen de certificación escolar, y permite profundizar más en diferentes apartados del currículo así como introducir metodologías menos frecuentes. El estudio, cuya duración fue de dos años, revela que en ambos grupos se apreciaba un aumento de la creatividad aunque en el segundo caso el aumento era mucho más notable. Las conclusión que se extrae de esta investigación es que el tiempo disponible parar nutrir a los alumnos con conocimientos más amplios, así como la flexibilidad a la hora de definir el currículo, son herramientas esenciales para establecer un clima que cultive creatividad.

### 1.6.1. Beneficios de cultivar la creatividad en el aula

Torrance [1976] resalta que la enseñanza creativa produce variados efectos beneficiosos a los niños y niñas. Mejora la composición creativa y el pensamiento creador, aumenta la predisposición a asistir a clase, a la participación en actividades creativas, y produce cambios en las metas académicas y profesionales.

La creatividad promueve la reflexión personal del propio acto de aprender creando nuevas formas de hacerlo. Por otro lado, el hecho de que el profesor cree nuevos escenarios de aprendizaje y que el alumno sea el principal actor creador de ellos genera buenos resultados, aumentando la participación y motivación [Sanz de Acedo, 2007].

En un estudio donde se trabajó con 72 niños cuya edad estaba comprendida entre los 9 y los 12 años, y apoyándose en la batería de pruebas del pensamiento creativo de Torrance TTCT [3] se demostró que los creativos eran más productivos en las tareas que implicaban frustración, a diferencia de los menos creativos [Esquivias, 2004].

Torrance realizó uno de los principales estudios predictivos a largo plazo, basándose en las pruebas de pensamiento creativo que el mismo diseñó (TTCT). Se utilizó la escala de Lorge-Thorndike, para medir el coeficiente intelectual. La batería comprendía las siguientes pruebas: plantear preguntas, adivinar las causas, adivinar las consecuencias, la mejora de un producto, los usos poco habituales de un producto mejorado, los usos poco habituales de un objeto ordinario y la de los círculos. Este estudio demostró que las pruebas de creatividad administradas durante la vida escolar, pueden predecir el éxito en cuanto a la creatividad en la edad adulta [Esquivias, 2004].

Dada la gran influencia que tiene el profesor, como persona adulta, sobre los estudiantes, adolescentes, conviene conocer qué indica la literatura sobre el papel del profesor en el aula. Este será el tema que se abordará en el siguiente apartado.

### 1.6.2. La figura del profesor

Torrance y Hansen [1965] llevaron a cabo una investigación sobre el comportamiento de los docentes, clasificándolos en más o menos creativos mediante el test de Torrance TTCT. Seleccionaron a los 6 docentes con menores habilidades creativas de un total de 29 participantes. Posteriormente se observó de forma minuciosa la forma de comportarse en clase de los profesores seleccionados, poniendo énfasis en las preguntas que estos emitían. Fueron observados durante cuatro meses a lo largo de cinco cursos diferentes. Las preguntas que los profesores emitían eran anotadas y se clasificaban según la puntuación dada por la escala de Burkhart-Bernheim de poder divergente, en divergentes-estimulantes o factuales-reproductivas. Por otro lado, tamvién fueron analizadas las preguntas de los pro-

[3] Por sus siglas en inglés: *Torrance Thinking Creative Test* [Torrance, 1974]

fesores considerados creativos, obeteniendo estos últimos una mayor puntuación en lo referido a la capacidad divergente. Posteriormente Hansen utilizó un procedimiento de observación adaptado de Flanders para estudiar el comportamiento de los dos grupos de profesores mencionados. Se encontró que los docentes creativos aceptan de buena forma las ideas de sus alumnos y suelen incorporar esas ideas en la estructura o secuencia del tema a tratar, asimismo, utilizan más ejemplos estimulantes para sus estudiantes, por lo que los profesores menos creativos eran más directos y toleraban mayor número de períodos de silencio y de confusión.

En este contexto, Sanz de Acedo [2007] propone diferentes aprendizajes que los docentes pueden llevar a cabo para desarrollar la creatividad en los centros educativos:

1. Mirar la realidad con nuevos ojos dentro y fuera del aula. Si los alumnos se esfuerzan en investigar ideas y fenómenos que les generen curiosidad, lograrán dejarse llevar por sus inclinaciones literarias, científicas o artísticas.

2. Experimentar con las ideas sugeridas por los demás. Los estudiantes han de procurar divagar libremente con ellas y generar diversas hipótesis antes de tomar una decisión, lo cual exige elaboración y reflexión sobre lo que se investiga y estudia.

3. Anotar las inspiraciones y los pensamientos personales. Esta técnica es ideal porque en todo proceso creativo aparecen una serie de ideas que podrán desarrollarse después en circunstancias más apropiadas.

4. Dedicar un tiempo suficiente a las tareas. La calidad de un producto está determinada por el tiempo y dedicación que se le han dedicado. Es importante escoger actividades que requieran reflexión y que permitan momentos de desconexión de las mismas.

# Capítulo 2

# Encaje de la creatividad en el currículo de secundaria

Puesto que el tiempo de docencia es limitado, uno puede preguntarse, de forma totalmente natural, si es posible trabajar el pensamiento creativo y, al mismo tiempo, cumplir con el currículo vigente. En este apartado veremos que no solo es posible, sino que puede ser un recurso muy apropiado para cumplir con el mismo.

Tomando como referencia el currículo de la Comunidad de Castilla y León (BOCYL n.º 86 8-mayo-2015 (jcyl.es)), se observa que en todos los niveles de Educación Secundaria Obligatoria y Bachillerato, las asignaturas de matemáticas tienen un bloque común y que a su vez es transversal al resto de bloques, *Bloque 1. Procesos, métodos y actitudes en matemáticas*. Según indica en el documento oficial recién citado:

> El primer bloque, «Procesos, métodos y actitudes en matemáticas», tiene un carácter transversal y vertebrador. Este bloque está constituido por cuatro grandes ejes: la resolución de problemas –más allá de la resolución de ejercicios de carácter rutinario y previsible-; el planteamiento y ejecución de investigaciones matemáticas relacionadas con los cuatro restantes bloques de números y álgebra, geometría, funciones y estadística y probabilidad; el enfoque modelizador e interpretativo que la matemática confiere a la realidad en

distintos entornos; el conocimiento de la propia capacidad y el desarrollo de una actitud positiva y responsable para enfrentarse a los retos que plantea el mundo, las ciencias y la matemática; y, finalmente, la capacitación para aplicar y utilizar los diferentes medios tecnológicos, especialmente informáticos.

Es interesante tener presentes los contenidos de este primer bloque ya que servirán para justificar la metodología y la forma de evaluar las actividades propuestas en el capítulo 3. Estos contenidos se detallan a continuación:

1. Planificación del proceso de resolución de problemas: análisis de la situación, selección y relación entre los datos, selección y aplicación de las estrategias de resolución adecuadas, análisis de las soluciones y, en su caso, ampliación del problema inicial.

2. Elección de las estrategias y procedimientos puestos en práctica: uso del lenguaje apropiado (gráfico, numérico, algebraico básico, etc.); construcción de una figura, un esquema o un diagrama; experimentación mediante el método ensayo-error; resolución de subproblemas dividendo el problema en partes; recuento exhaustivo, comienzo por casos particulares sencillos, búsqueda de regularidades; etc.

3. Reflexión sobre los resultados: revisión de las operaciones utilizadas, asignación de unidades a los resultados, comprobación e interpretación de las soluciones en el contexto de la situación, búsqueda de otras formas de resolución, etc.

4. Expresión verbal y escrita en Matemáticas.

5. Práctica de los procesos de matematización en contextos de la realidad y en contextos matemáticos.

6. Iniciación en el planteamiento de pequeñas investigaciones matemáticas escolares en contextos numéricos, geométricos, funcionales, estadísticos y probabilísticos, adecuados al nivel educativo y a la dificultad de la situación.

7. Confianza en las propias capacidades para desarrollar actitudes adecuadas y afrontar las dificultades propias del trabajo de la materia.

8. Utilización de medios tecnológicos en el proceso de aprendizaje para:

   - La recogida ordenada y la organización de datos mediante tablas.
   - La elaboración y creación de representaciones gráficas de datos numéricos, funcionales o estadísticos (gráficas de funciones, diagramas de sectores, barras,...).
   - Facilitar la comprensión de propiedades geométricas o funcionales y la realización de cálculos de tipo numérico, algebraico o estadístico.
   - La elaboración de predicciones sobre situaciones matemáticas diversas.
   - La elaboración de informes y documentos sobre los procesos llevados a cabo y los resultados y conclusiones obtenidos.
   - Comunicar y compartir, en entornos apropiados, la información y las ideas matemáticas.

A continuación, se comentará brevemente cómo se pueden integrar estos contenidos en la ejercitación del pensamiento creativo.

El primero de los contenidos encaja perfectamente en el esquema de los procesos cognitivos de Munford et al. [2012] para resolver problemas de forma creativa. Este esquema de Munford asume tres hipótesis. La primera es que la resolución de problemas debe tomar como base el conocimiento y la información, considerando el conocimiento como una herramienta para interpretar la información. La segunda es que un individuo no puede generar nuevas ideas contando únicamente con el conocimiento que ya posee. En su lugar, este conocimiento debe ser reorganizado y reestructurado para generar nuevo conocimiento que permita generar nuevas ideas. La tercera hipótesis es que las ideas generadas deben ser evaluadas y adecuarse para integrarse en planes de trabajo viables que de lugar a un proyecto creativo. El modelo de Munford asume que el pensamiento creativo empieza con la definición del problema. Esta definición da lugar a reunir la información y seleccionar los conceptos necesarios para entender dicha información. Luego estos conceptos forman la base para crear una combinación conceptual que, a su vez, permitirá la generación de ideas y su posterior evaluación. Después de la selección de ideas, se diseña el plan a seguir y posteriormente se ejecuta. Este flujo de procesos está ideado de tal forma que si alguno de los procesos no termina de forma exitosa, el individuo simplemente ha de dirigirse al proceso previo y reconsiderarlo para evitar el problema que se estaba encontrando. Ver figura 2.1.

En cuanto al segundo contenido del *Bloque 1. Procesos, métodos y actitudes en matemáticas*, se tiene que en la mayoría de las actividades planteadas en las secciones 3.5 y 3.6 una de las estrategias consistirá en la experimentación del método ensayo-error. Asimismo, las actividades a desarrollar en el Taller de la sección 3.6 comenzarán por casos particulares sencillos, luego se presentarán ejemplos más complejos y cómo estos pueden descomponerse en casos sencillos, en este tipo de actividades de programación será muy útil recurrir a diagramas, esquemas y figuras para poder resolver los diferentes retos propuestos.

Respecto al tercer contenido, en el transcurso de las actividades propuestas en la sección 3.6 se irán introduciendo modificaciones sobre el enunciado, por ejemplo limitar el uso de ciertos recursos, de modo que los alumnos tengan que revisar las operaciones que estaban haciendo y plantear una nueva solución. De este modo trabajan una de las dimensiones del pensamiento divergente, la flexibilidad.

Tratando los contenidos cuarto, quinto y sexto como un conjunto, cabe destacar que en el capítulo 3 se proponen actividades donde los alumnos deberán resolver un problema de la vida real, relacionado con las matemáticas, y deberán explicar cómo lo han resuelto. De este modo comparten su experiencia con otros compañeros y todos se benefician.

En relación al séptimo contenido, sobre la confianza en las propias capacidades, las actividades que se plantean son fácilmente adaptables a cualquier nivel. Todos partirán del mismo nivel, y aquellos que avancen más rápido se enfrenarán a retos de mayor dificultad. De este modo los alumnos irán adquiriendo confianza paso a paso.

Por último, el octavo contenido encaja de forma evidente en las actividades que se han planteado en la sección 3.6. Uno de los propósitos de este trabajo es promover el uso de las Tecnologías de la Información y la Comunicación.

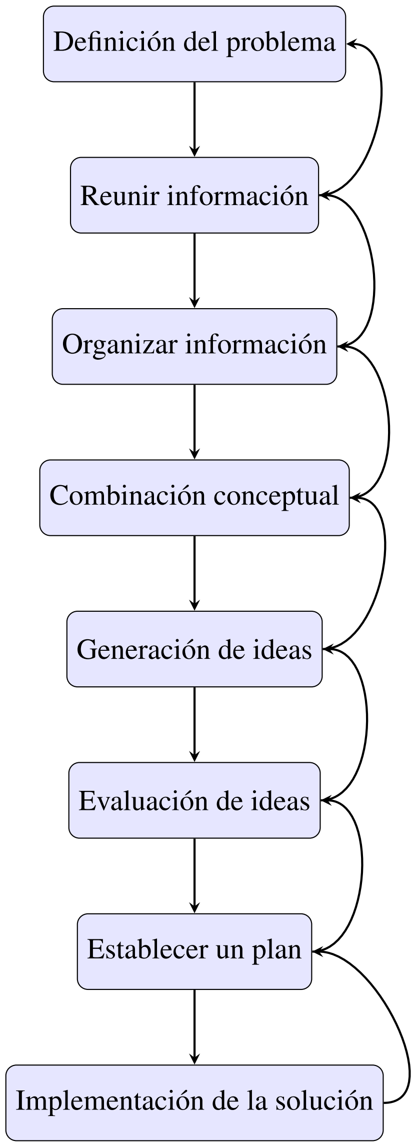



Figura 2.1: Esquema de los procesos cognitivos para la resolución creativa de problemas [Munford et al., 2012]. Este flujo de procesos está ideado de tal forma que si alguno de los procesos no termina de forma exitosa, el individuo simplemente ha de dirigirse al proceso previo y reconsiderarlo para evitar el problema que se estaba encontrando.

# Capítulo 3

# Actividades propuestas

Este capítulo se estructura del siguiente modo: en primer lugar se definirán la metodología, temporalización y evaluación de las actividades, en segundo y tercer lugar se presentarán, respectivamente, diferentes actividades de dominio general y específico, y por último se propondrá un taller para trabajar el pensamiento creativo.

Cabe destacar que las actividades se agrupan en tres categorías: desafíos, actividades (propiamente dichas) y proyectos. Los detalles que diferencian cada una de estas clases vienen en la sección 3.2. Asimismo, se cree conveniente aclarar la inclusión de actividades de dominio general en el presente trabajo. El motivo de incorporar este tipo de actividades se debe a que, muy probablemente, en algún momento de la trayectoria profesional del docente, este deba encargarse de tutorizar un grupo de alumnos, y quizá le resulte útil contar con este tipo de recursos. Más aún, podría ser que algún profesor se anime a proponer un proyecto en su centro educativo como, por ejemplo, una semana de la creatividad, donde los profesores se pongan de acuerdo y colaboren para que los alumnos realicen actividades que fomenten el pensamiento creativo.

Las actividades que se proponen seguirán el siguiente esquema, ver figura 3.1: título, categoría (desafío, actividad o proyecto), objetivos que se persiguen (competencias o dimensiones del pensamiento abstracto que se quieren trabajar), instrucciones detalladas para llevar a cabo la actividad, una distribución orientativa de los tiempos que requiere cada paso de la actividad, también se indicarán los materiales y recursos necesarios, y finalmente se propondrá una forma de evaluar la actividad.

**Título**

Categoría Duración

Objetivos

Instrucciones

Distribución de tiempos

Recursos

Evaluación

Figura 3.1: Esquema que seguirán las actividades propuestas.

## 3.1. Metodología

La metodología que se plantea para las actividades propuestas en este capítulo es una metodología activa. Consiste en seguir el esquema de Munford et al. [2012] para la resolución de problemas de forma creativa. Con esta metodología se busca dejar a un lado el aprendizaje pasivo y poner al alumno en el centro del aprendizaje. El objetivo que se persigue con esta metodología es, en última instancia, equipar al alumnado con una mochila llena de herramientas (habilidades) para aprender a aprender. Esta metodología se aplicará al aprendizaje basado en proyectos, aprendizaje en grupos colaborativos, y aprendizaje basado en la resolución de problemas.

Por lo tanto, el docente debe asumir el papel de guía, deberá explicar en voz altas las diferentes etapas de las actividades, que seguirán el esquema de Munford, y asesorar a los alumnos en cada una de ellas.

Asimismo, en determinadas ocasiones se recurrirá a la metodología de tipo clase magistral participativa, especialmente a la hora de introducir conceptos nuevos o establecer un marco teórico en el que los alumnos deberán trabajar.

A continuación se explicará con mayor detalle en qué consiste el método de Munford para la resolución creativa de problemas. Las hipótesis sobre las que se apoya el autor son:

- La resolución de problemas debe tomar como base el conocimiento y la información, considerando el conocimiento como una herramienta para interpretar la información.

- Un individuo no puede generar nuevas ideas contando únicamente con el conocimiento que ya posee. En su lugar, este conocimiento debe ser reorganizado y reestructurado para generar nuevo conocimiento que permita generar nuevas ideas.

- Las ideas generadas deben ser evaluadas y adecuarse para integrarse en planes de trabajo viables que de lugar a un proyecto creativo.

Las etapas del esquema de Munford son:

1. Definición del problema: entender el problema y los criterios que la solución debe cumplir.

2. Reunir la información: preguntarse qué conocimientos se tienen sobre el tema en cuestión.

3. Organizar la información: seleccionar los conocimientos que pueden contribuir a solucionar el problema.

4. Combinación conceptual: realizar un ejercicio de abstracción para relacionar y reorganizar los conocimientos seleccionados.

5. Generación de ideas: realizar una “lluvia de ideas”. Es decir, poner sobre la mesa todas las ideas que se ocurran para solucionar el problema.

6. Evaluación de ideas: filtrar las ideas generadas en base a determinados criterios como, por ejemplo, el coste de la solución, la eficiencia, dificultad, etc.

7. Establecer un plan: organizar las ideas filtradas y diseñar una ruta a seguir.

8. Implementar la solución: ejecutar el plan establecido.

Recuérdese que este esquema está pensado de tal forma que si una de las etapas no logra completarse con éxito, entonces hay que retroceder al paso anterior y reconsiderarlo para librarse del problema encontrado (figura 2.1).

Nótese que para que este método funcione, el profesor debe dominarlo muy bien y los alumnos deben estar motivados en su aplicación. Para ello es conveniente, como se verá en la sección 3.2, que se dedique parte de una sesión a explicar qué es la creatividad, por qué es conveniente trabajarla y cómo puede ejercitarse (esquema de Munford).

## 3.2. Temporalización

Es de sobra conocido que el tiempo es un recurso escaso en la docencia, por ello en este trabajo se proponen diferentes tipos de actividades:

- Desafíos: se trata de actividades muy dinámicas, en el sentido de que no requieren gran preparación y pueden completarse en un espacio de tiempo corto. Están pensadas para tener una duración de 10-15 minutos. Pueden ser recursos interesantes para motivar a los alumnos rompiendo la monotonía.

- Actividades (propiamente dichas): a diferencia de los desafíos, estas actividades requieren preparar material específico, o bien están pensadas para ocupar un intervalo de tiempo mayor (20-30 minutos). Pueden ser una herramienta idónea para introducir un nuevo tema o para reforzar algún concepto.

- Proyectos: se caracterizan por requerir un mayor tiempo y dedicación para su realización (varias sesiones). Están concebidos para ahondar en algún campo concreto.

Para evaluar los aprendizajes de los alumnos al trabajar siguiendo esta metodología, es fundamental que los alumnos tengan claros los objetivos de aprendizaje que se persiguen, por ello es importante establecer un marco teórico. Asimismo, el profesor debe indicar qué aspectos se tomarán en cuenta para la evaluación. Esto último se verá en la sección 3.3.

## 3.3. Evaluación

Con la finalidad de evitar mayores confusiones, debe hacerse una breve distinción. Por evaluación se puede entender tanto la evaluación sobre los alumnos, como la evaluación de la actividad. Ambas evaluaciones son importantes y deben hacerse.

En el caso de la evaluación sobre los alumnos, debe entenderse que tanto los desafíos, como las actividades, están pensadas para realizarse en momentos puntuales, por ello lo que se pretende es medir el grado de participación de los alumnos. No obstante, puesto que los proyectos requieren un mayor grado de esfuerzo, estos sí serían calificados y podrían contar, o bien como un examen parcial más, o bien computar dentro del porcentaje reservado para los trabajos (si lo hubiese).

En el segundo caso se trata de evaluar la actividad, con este fin pueden diseñarse rúbricas como la que aparece en la figura 3.3. Esta evaluación consiste en observar el desarrollo de la actividad para detectar aspectos mejorables de la misma: material de trabajo, indicaciones, enunciados, temporalización, etc; así como también poder medir la evolución de la clase si la actividad se repite alguna vez más.

## 3.4. Actividades de dominio general

### 3.4.1. *El dibujante*

La siguiente actividad ha sido concebida como un recurso a emplear con los alumnos después de establecer el marco teórico de la creatividad. Consiste en responder, mediante dibujos, a una cuestión que se plantea enmarcada en un contexto, posteriormente otras personas deberán deducir qué está ocurriendo en dichos dibujos y relacionarlos para reconstruir la problemática inicial o una similar.

Al tratarse de una actividad grupal se consigue que los estudiantes colaboren y se ayuden entre ellos para entender las diferentes etapas del pensamiento creativo. Además se trata de una actividad muy versátil pues puede llevarse a cabo con una perspectiva de dominio general para los más jóvenes, tal y como aquí se plantea, o también podría adaptarse para ser una actividad más específica de una área de conocimiento concreta enfocada a los estudiantes más mayores.

# El dibujante

---

Categoría: actividad Duración: 30-35’

## Objetivos

- Trabajar en equipo.
- Trabajar las siguientes dimensiones del pensamiento creativo que se vieron en la sección 1.2: sensibilidad, fluidez, flexibilidad, redefinición y evaluación.

## Instrucciones

Para el desarrollo de esta actividad ha de dividirse la clase en grupos de 3 o 4 personas. Idealmente, el profesor se encargará de designar los miembros de cada grupo. De este modo se puede cuidar que los grupos sean heterogéneos. Es muy conveniente que antes de revelar los miembros de cada grupo el docente explique la actividad y se asegure que todos lo han entendido.

Una vez se han formado los grupos, el profesor repartirá tres folios a cada grupo. Uno de ellos será un folio donde los alumnos podrán realizar anotaciones en sucio, otro se empleará para hacer dibujos y en el tercero deberán realizar una breve redacción.

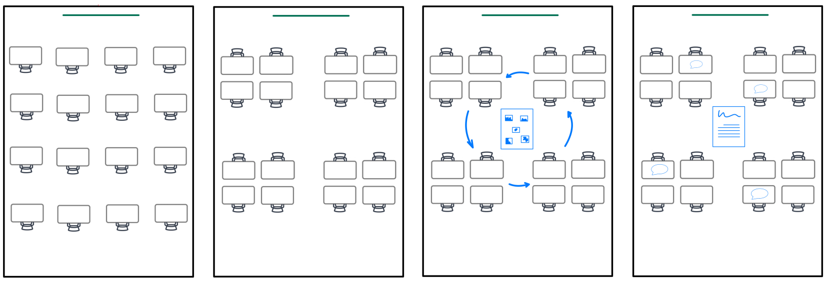
La actividad se divide en cuatro partes:

1. Cada grupo tiene asignada una problemática, esta consiste en un contexto y una pregunta relacionado con dicho contexto. En esta primera ronda el profesor se acercará a cada uno de los grupos y les contará en voz baja cuál es su problemática. Cada grupo debe conocer única y exclusivamente cuál es su cuestión a resolver. Cuando el profesor les haga saber dicho problema, estos deben escribir en el folio las anotaciones que consideren oportunas para poder recordarla. De este modo se trabaja su atención. Cuando todos los grupos conocen su problemática entonces se da paso a la siguiente parte. Para agilizar esta parte, en lugar de tener que ir por cada grupo indicando su problema, el profesor podría entregar bocabajo un trozo de papel con la problemática escrita en él.

2. Los alumnos disponen de 3 minutos para responder a la pregunta que viene en su problemática, la respuesta debe consistir en un conjunto de 5 dibujos (exclusivamente dibujos). Aquí además de fomentar el trabajo cooperativo, se está trabajando el pensamiento divergente (fluidez y evaluación de ideas). Al término de este tiempo, el profesor recoge el folio con los dibujos y los vuelve a repartir de modo que a ningún grupo le vuelva a tocar el folio con sus respuestas en forma de dibujo. Un consejo para agilizar la recogida y el reparto de folios, consiste en organizar bien los espacios de trabajo en grupo. Así los grupos pueden ser ordenados, por ejemplo siguiendo el sentido de las agujas del reloj, y tener identificado a quién tienen que pasar sus respuestas y de quién tienen que recibirlas.

3. Cada grupo tiene ahora un conjunto de 5 dibujos, pero desconocen de qué problemática proceden. Durante esta parte los alumnos tienen 5 minutos para redactar una breve narración tratando de adivinar cuál era la problemática que relaciona los dibujos, o dar una alternativa. Si algún grupo termina antes de tiempo entonces puede redactar nuevas alternativas. En este paso se trabaja la identificación de restricciones (flexibilidad y evaluación de ideas) al tener que buscar una relación entre los dibujos, y también el pensamiento divergente (originalidad y redefinición) al darles margen para que piensen diversas alternativas y añadan tantos detalles como deseen.

4. Al concluir los 5 minutos del apartado anterior, un portavoz de cada grupo podrá poner en común con el resto de la clase los dibujos que les han tocado y qué circunstancia sospechan ellos que conecta dichos dibujos. El grupo que originalmente había realizado dichos dibujos revelará entonces cuál era la problemática que les había tocado.



(a) Explicación. (b) Dibujos. (c) Intercambio de imágenes. (d) Puesta en común.

Figura 3.2: Esquema de las diferentes etapas de la actividad *El dibujante.*

## Distribución de tiempos

A continuación, se indica, de forma orientativa, cuánto tiempo puede llevar realizar los diferentes pasos de la actividad.

- Explicar en qué consiste la actividad, detallar los pasos y resolver las dudas que puedan surgir: 5’ - 10’ (en función de las dudas).

- Indicar a cada grupo su problemática: 5'.
- Dibujos: 3'.
- Repartir dibujos: 2'.
- Narración: 5'.
- Puesta en común: 10' - 15'.

En total son unos 30 - 35 minutos. No obstante, una vez aprendido el funcionamiento, las siguientes veces podría realizarse de forma más ágil. El tiempo mínimo requerido para realizar esta actividad es de 25 minutos, siempre y cuando los alumnos tengan un mínimo de experiencia y recuerden las reglas y los pasos.

## Materiales

Para realizar esta actividad solo se requieren 3 folios y una problemática por grupo. Puede hacerse uso de materiales de apoyo, como un proyector o una pizarra, para mostrar el esquema de las diferentes etapas de la actividad.

A continuación, se proponen algunos ejemplos de problemáticas que podrían emplearse en esta actividad:

- Es verano y has ido de vacaciones con tus padres a la playa, te has despistado y no localizas a tus padres. ¿Qué harías? Descríbelo en 5 dibujos.
- Vas a empezar una ruta de senderismo que dura varios días, durante la ruta no tendrás cobertura y en la mochila solo puedes llevar 5 objetos, ¿qué objetos te llevarías?
- Tienes síntomas de COVID y no puedes salir de casa, le pides a un amigo que vaya a comprar algunas cosas para ti, para no abusar decides pedirle 5 cosas, ¿qué le pedirías?
- Te has ido de vacaciones a la montaña, has salido a dar un paseo tú sólo y empieza a anochecer, dudas sobre cuál es el camino de vuelta. ¿Qué objetos te gustaría tener contigo?

- Tu primo, que tiene tu misma edad, ha ido a visitaros durante el puente de diciembre. Os han castigado sin pantallas (televisión, móvil, Tablet, etc.), ¿qué podéis hacer para entreteneros? Descríbelo en 5 dibujos.
- Has quedado con tus amigos en el centro, te bajas del autobús y cuando se marcha te das cuenta de que te has dejado el móvil. ¿Qué harías si no recuerdas dónde habéis quedado? Descríbelo en 5 dibujos.

## Evaluación

Un modo de evaluar esta actividad, de acuerdo con lo descrito en la sección 3.3, sería cumplimentar la rúbrica de la figura 3.3.

| Etapa de la actividad | Objetivo | Evaluación | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| Paso 1 | Identificación del problema | Entienden la problemática asignada | Nada | Ligeramente | Bastante | Completamente |
| | Generación de ideas | Número de ideas propuestas | Muy pocas | Pocas | Bastantes | Muy abundantes |
| | Evaluación de las ideas | Los dibujos se ajustan a la problemática | Nada | Ligeramente | Bastante | Totalmente |
| Paso 2 | Pensamiento convergente | Conectan el contenido de los dibujos de forma coherente | 2 dibujos | 3 dibujos | 4 dibujos | 5 dibujos |
| Transversal | Trabajo Colaborativo | Los integrantes del grupo colaboran | No hay colaboración | Una minoría | La mayoría | Todos |
| | Trabajo Colaborativo | Toma de decisiones | Impuestas por un individuo | Impuestas por una minoría | Una parte cede | Mediante consenso |

Figura 3.3: Ejemplo de rúbrica para evaluar el desarrollo de la actividad 3.4.1 *El dibujante*.

### 3.4.2. *Puzle narrativo*

Esta actividad consiste en construir una breve historia, a partir de palabras que han sido filtradas mediante un proceso de *brainstorming* [1] ajustándose a una condición determinada.

Al igual que en la actividad 3.4.1, esta puede adaptarse a diferentes edades y niveles educativos con simples cambios. En esta ocasión se ha decidido orientarla a un alumnado de los primeros cursos de Educación Secundaria Obligatoria.

# Puzle narrativo

---

Categoría: actividad Duración: 35-40’

## Objetivos

- Trabajar en equipo.
- Trabajar las siguientes dimensiones del pensamiento creativo que se vieron en la sección 1.2: fluidez, originalidad, flexibilidad, y evaluación.

## Instrucciones

Para el desarrollo de esta actividad deberán formarse grupos de 4 personas. Idealmente, el profesor se encargará de designar los miembros de cada grupo. De este modo se puede cuidar que los grupos sean heterogéneos. Es muy conveniente que antes de revelar los miembros de cada grupo el docente explique la actividad y se asegure que todos lo han entendido.

Inicialmente no es necesario que los alumnos se sienten juntos por grupos ya que los primeros pasos de la actividad se trabajan de forma individual.

El profesor dispone de tarjetas (cartulinas A5) de 4 colores diferentes. A cada miembro del grupo le corresponde una tarjeta de un color distinto, cada una tiene su

[1] Lluvia de palabras.

propio significado. Es conveniente anotar o proyectar en la pizarra el significado de cada color. Por otro lado, el profesor también cuenta con folios que tienen escrito un contexto (un momento histórico, una situación cotidiana, una escena de película, etc.), y cada contexto va asociado a un grupo. Es decir, hay tantos contextos como grupos, y además cada uno aparece repetido 4 veces, de modo que el profesor podrá repartir a cada miembro un folio indicante el que le corresponde a su grupo.

La actividad se divide en varias etapas:

1. El profesor reparte a cada alumno una tarjeta y un folio con el contexto que le ha tocado a su grupo. El folio puede ir metido en un sobre, doblado o simplemente dejarse bocabajo en la mesa de los estudiantes. En este punto los estudiantes todavía no saben qué significa el color de las tarjetas. Cuando el profesor termina de repartir los materiales entonces revela el significado de los colores. Cada color está asociado a un campo distinto, nombres de personajes (reales o ficticios), verbos, sentimientos, y objetos. Ahora pueden leer cuál el contexto que aparece en el folio.

2. Lluvia de palabras: los alumnos disponen de 2 minutos y medio para escribir, de forma individual, en una de las caras de la tarjeta, tantas palabras de su campo y relacionadas su contexto como se les ocurra. Debe mantenerse silencio en esta parte, el trabajo es individual. Se trata de ejercitar la fluidez verbal.

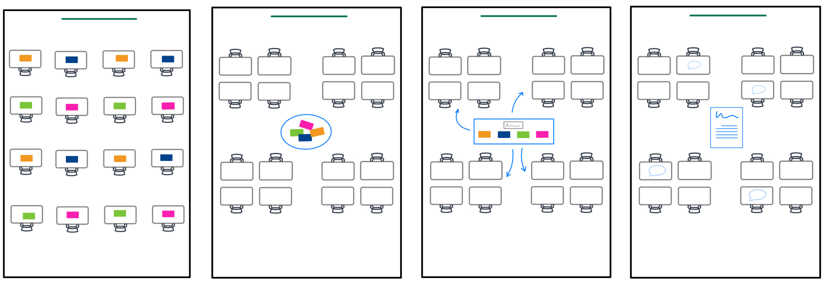
3. A continuación, se introduce una condición sobre cada campo, por ejemplo:

   - Personaje: ficticio (personaje de película, dibujos animados, cómics, etc.).
   - Verbo: que tengan al menos tres sílabas.
   - Sentimiento: que contenga la p.
   - Objeto: palabras llanas.

Los alumnos disponen de 1 minuto para escribir en la otra cara de la tarjeta una, y solo una, palabra de su campo que cumpla la condición impuesta. Para ello pueden apoyarse en la lluvia de palabras que han realizado en la cara opuesta de la tarjeta. Este paso es un ejercicio de flexibilidad.

4. Se recogen las tarjetas. El profesor agrupa las tarjetas por colores y baraja cada uno de los montones. Mientras los alumnos se sientan en sus grupos correspondientes. A continuación, reparte una tarjeta de cada color a los grupos. Desde entonces, estos tienen 5 minutos para redactar una o varias narraciones donde expliquen, atendiendo a las palabras que les han tocado, por qué ese personaje está realizando la acción del verbo en ese contexto, para qué quiere el objeto que le ha tocado, y por qué tiene ese sentimiento. En esta parte de la actividad se trabaja tanto la originalidad, como la flexibilidad y la evaluación de ideas.

5. Al concluir los 5 minutos del apartado anterior, un portavoz de cada grupo podrá poner en común con el resto de la clase su narración y las palabras que les había tocado en el reparto de tarjetas.



(a) Generación de palabras. (b) Intercambio de tarjetas. (c) Conexión de palabras. (d) Puesta en común.

Figura 3.4: Esquema de las diferentes etapas de la actividad *El dibujante*.

## Distribución de tiempos

A continuación, se indica, de forma orientativa, cuánto tiempo puede llevar realizar los diferentes pasos de la actividad. Para agilizar el proceso de repartir y recoger las tarjetas, puede ser conveniente contar con un o dos ayudantes, bien sean alumnos o compañeros docentes.

- Explicar en qué consiste la actividad, detallar los pasos y resolver las dudas que puedan surgir: 5’ - 10’ (en función de las dudas).
- Repartir las tarjetas: 2’.
- Lluvia de palabras: 3’.
- Indicar la restricción sobre cada campo: 2’.
- Recoger las tarjetas, barajarlas y sentarse por grupos: 4’.
- Repartir tarjetas a cada grupo: 2’.
- Elaboración de la narración: 5’.
- Puesta en común: 10’ - 15’.

En total son unos 35 - 45 minutos. No obstante, una vez aprendido el funcionamiento, las siguientes veces podría realizarse de forma más ágil. El tiempo mínimo requerido para realizar esta actividad es de 25 - 30 minutos, siempre y cuando los alumnos tengan un mínimo de experiencia y recuerden las reglas y los pasos.

## Recursos

Para realizar la actividad los materiales necesarios son los siguientes:

- Fichas de 4 colores, cartulinas tamaño A5. El número de cartulinas de cada color debe ser igual al número de grupos.
- Folios con los diferentes contextos, debe haber tantos contextos como grupos y cada uno debe aparecer 4 veces (uno por integrante del grupo).

Puede usarse un proyector o una pizarra, para mostrar el esquema de las diferentes etapas de la actividad, así como el código de colores de las tarjetas y las condiciones impuestas en la etapa 3.

A continuación, se proponen algunos ejemplos de contextos que podrían emplearse en esta actividad:

- Fiesta de una familia patricia (adinerada) en la época del Imperio Romano.
- Hundimiento del Titanic (crucero que se hundió al chocar con un iceberg).
- Construcción de las pirámides de Egipto.
- Persecución policial a unos ladrones de bancos.

Nótese que los contextos empleados deben adaptarse a la edad de los alumnos que van a participar en la actividad. En este caso los contextos están enfocados a trabajar la actividad con alumnos de los primeros cursos de Educación Secundaria Obligatoria.

## Evaluación

Un modo de evaluar esta actividad, de acuerdo con lo descrito en la sección 3.3, sería cumplimentar la rúbrica de la figura 3.5.

| **Etapa de la actividad** | **Objetivo** | **Evaluación** | **1** | **2** | **3** | **4** |
|---|---|---|---|---|---|---|
| Paso 1 | Fluidez | Número de ideas propuestas | Muy pocas | Pocas | Bastantes | Muy abundantes |
| | Originalidad | Las palabras se repiten | Muchas | Bastante | Algo | Poco |
| Paso 2 | Identificación de restricciones | Las palabras seleccionadas se ajusta a la condición | Menos de la mitad | Más de la mitad | La mayoría | Todas |
| Paso 3 | Pensamiento convergente | Conectan el contenido de las tarjetas de forma coherente | No hay coherencia | 2 palabras | 3 palabras | 4 palabras |
| Transversal | Trabajo Colaborativo | Los integrantes del grupo colaboran | No hay colaboración | Una minoría | La mayoría | Todos |
| | Trabajo Colaborativo | Toma de decisiones | Impuestas por un individuo | Impuestas por una minoría | Una parte cede | Mediante consenso |

Figura 3.5: Ejemplo de rúbrica para evaluar el desarrollo de la actividad 3.4.2 *Puzle narrativo*.

### 3.4.3. *Wordle*

*Wordle* es un juego rápido y sencillo de aprender. Esto hace que sea un candidato excelente para llevar al aula ya que no requiere mucha preparación. Este juego, llevado al idioma español por el Colombiano Daniel Rodriguez [Wardle, s.f], consiste en adivinar una palabra oculta mediante una serie de pistas.

El motivo de seleccionar este juego es doble, por un lado la naturaleza de este juego invita al usuario a trabajar la fluidez verbal y también la flexibilidad, y por otro lado, también es una actividad que permite introducir de forma natural otras actividades como el *Ooodle*, de contenido matemático pero algo más complejo, que se propone en el apartado 3.5.1.

# Wordle

---

Categoría: desafío Duración: 10-15'

## Objetivos

- Trabajar de forma individual.
- Trabajar las siguientes dimensiones del pensamiento creativo que se vieron en la sección 1.2: fluidez y flexibilidad.

## Instrucciones

Para empezar la actividad los alumnos deben tener a su alcance un dispositivo con conexión a internet, un ordenador, una tableta o un teléfono móvil.

Una vez tengan listo el dispositivo desde el que van a trabajar, deben dirigirse al siguiente enlace web: `https://wordle.danielfrg.com/`, o bien introducir en su navegador los siguientes términos: "wordle daniel frg". Ver figura 3.6. El usuario puede elegir jugar el modo científico si así lo desea. La diferencia frente al modo normal es que las palabras pueden tener entre 5 y 7 letras (frente a las 5 del modo normal), y que dichas palabras están relacionadas con la ciencia.

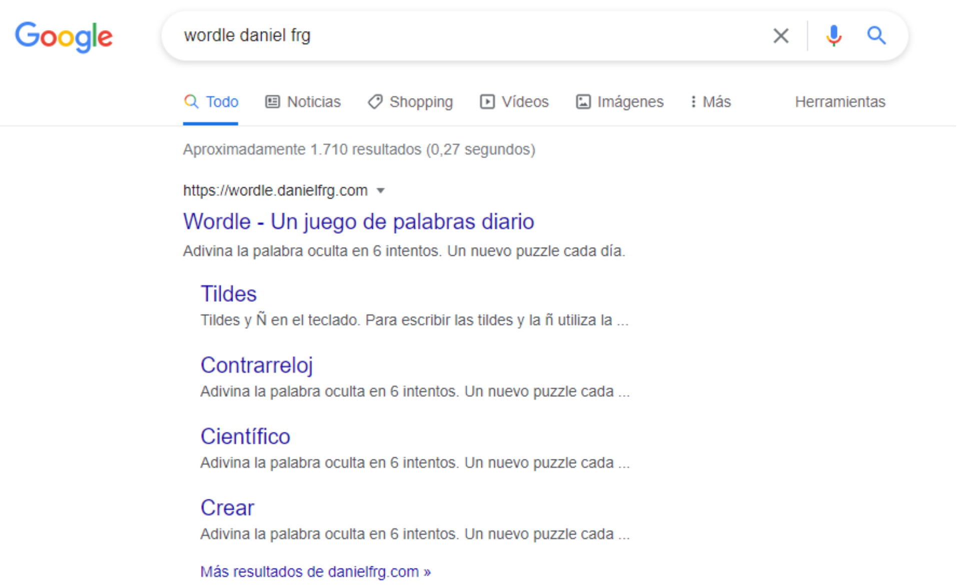



Figura 3.6: Búsqueda de la web de *Wordle* desde el navegador.

El juego consiste en adivinar una palabra que se encuentra oculta, con un máximo de 6 intentos. Cada intento debe ser una palabra válida que contenga tantas letras como espacios admite la palabra oculta. Después de cada intento el color de las letras cambia ofreciendo una pista sobre lo acertado que ha sido el intento. Los colores posibles son:

- Verde: la letra coloreada está en la palabra oculta ocupando esa misma posición.
- Amarillo: la letra coloreada está en la palabra oculta ocupando ocupando una posición distinta.
- Gris: la letra coloreada no aparece en la palabra oculta.

En cada intento el participante debe realizar una búsqueda de candidatos válidos, ejercitando así la fluidez, y también ha de ser flexible para poder utilizar aquello que conoces de los intentos anteriores, las letras con sus posiciones y color.

Ver la figura 3.7 para una descripción detallada del trascurso de una partida.

Aquellos participantes que resuelvan la palabra oculta pueden empezar un nuevo reto con una palabra distinta.

## Distribución de tiempos

A continuación, se indica, de forma orientativa, cuánto tiempo puede llevar realizar los diferentes pasos de la actividad.

1. Explicar en la pizarra en qué consiste la actividad: 5'.
2. Adivinar la palabra oculta: 5-10'.

## Recursos

Para realizar la actividad los materiales necesarios son los siguientes:

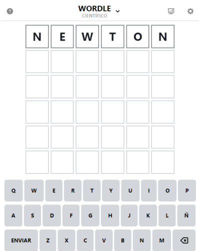
- Dispositivo con conexión a internet.
- Conexión a internet.

Además, es conveniente disponer de una pizarra para explicar cómo funciona el *Wordle* y resolver alguna duda que pueda surgir.

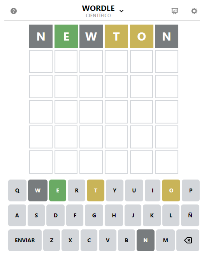
A mayores, si el docente así lo desea, puede crearse una palabra oculta personalizada y compartir el enlace a los alumnos. De este modo pueden emplearse palabras vistas en clase para que los alumnos refuercen sus conocimientos.

## Evaluación

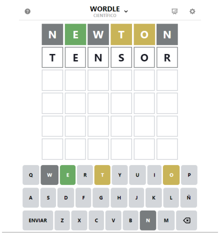
Al tratarse de un desafío, se supone que esta debe ser una actividad fluida y de corta duración. Por ello, más que una rúbrica como se ha planteado en las actividades anteriores, aquí se sugiere que el profesor pase una encuesta a los alumnos para que estos indiquen el tiempo y el número de intentos que han necesitado para resolver la palabra oculta. De este modo el profesor puede hacer un seguimiento sencillo de la evolución de los alumnos, y estos también podrán ver como van mejorando.



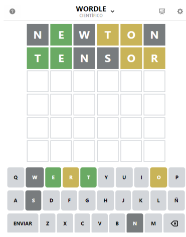

(a) Inicia la partida, se tienen seis intentos por delante y solo se conoce el número de letras que tiene la palabra oculta, 6 en este caso. Se prueba con una palabra cualquiera de 6 letras.



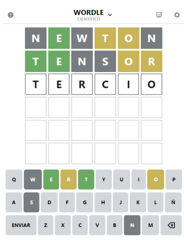

(b) Ahora se sabe que la palabra oculta tiene una “E” en la segunda posición, tiene una “T” fuera de la cuarta posición, una “O” fuera de la quinta posición, y ni la “N” ni la “W” aparecen.



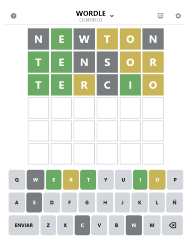

(c) En el siguiente intento se prueba con una palabra nueva teniendo en cuenta los resultados del intento anterior. En este paso no se busca resolver el reto sino dar un paso tratando de averiguar donde deben ir las letras “T” y “O”.



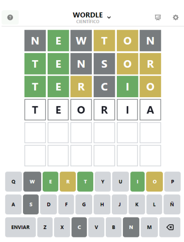

(d) “TENSOR” no podía ser solución pues tenía una “N”, pero ha permitido conocer la posición de la “T” así como nueva información de las letras “S”, “O” y “R”.



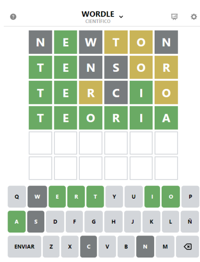

(e) Se prueba un nuevo intento haciendo uso de la información conocida. Esta vez tratando de afinar más.




(f) Ahora se conocen 5/6 letras, y la posición de 3 de ellas.




(g) Ahora ya se cuenta con mucha información, es más fácil intentar resolver el juego.




(h) La palabra oculta ha sido descubierta.

Figura 3.7: Descripción del sistema de intentos en el juego *Wordle*.

## 3.5. Actividades de dominio específico: matemáticas

### 3.5.1. *Ooodle*

Esta actividad es similar al *Wordle*, propuesta en el apartado 3.4.3, pero con un enfoque matemático. Se trata de una actividad más compleja que el *Wordle*, por lo que se recomienda introducirlo después. El *Ooodle* consiste en descubrir los cuatro operandos de una determinada operación, de tal modo que se cumpla cierta igualdad [Maths Zone, s.f].

Este juego requiere que el usuario se abstraiga y razone, que proponga diferentes caminos y que evalúe las diferentes opciones. Es un juego que estimula la creatividad y al mismo tiempo ejercita el cálculo mental.

Nótese que esta actividad requiere soltura en el cálculo mental y capacidad de abstracción, por lo que podría ser demasiado exigente en los primeros niveles de Educación Secundaria Obligatoria. Se recomienda su uso en los cursos de tercero y cuarto de la E.S.O. así como también en Bachillerato y Formación Profesional. Asimismo, se recomienda para motivar y estimular a los alumnos de altas capacidad o que sean muy aventajados de cualquier curso.

Pueden encontrarse más juegos de este estilo en: `https://nerdlegame.com/`. Una web con diversos recursos para ejercitar las matemáticas y el pensamiento creativo. Disponen de una aplicción propia tanto en la tienda de Android, Play Store, como en la de iOS, App Store.

# Ooodle

---

Categoría: desafío Duración: 10-15’

## Objetivos

- Trabajar de forma individual.

- Trabajar las siguientes dimensiones del pensamiento creativo que se vieron en la sección 1.2: fluidez y flexibilidad.

## Instrucciones

Para empezar la actividad los alumnos deben tener a su alcance un dispositivo con conexión a internet, un ordenador, una tableta o un teléfono móvil.

Una vez tengan listo el dispositivo, deben dirigirse al siguiente enlace web: `https://mathszone.co.uk/resources/grid/ooodle/`, o bien introducir en su navegador: "ooodle". Ver figura 3.8.

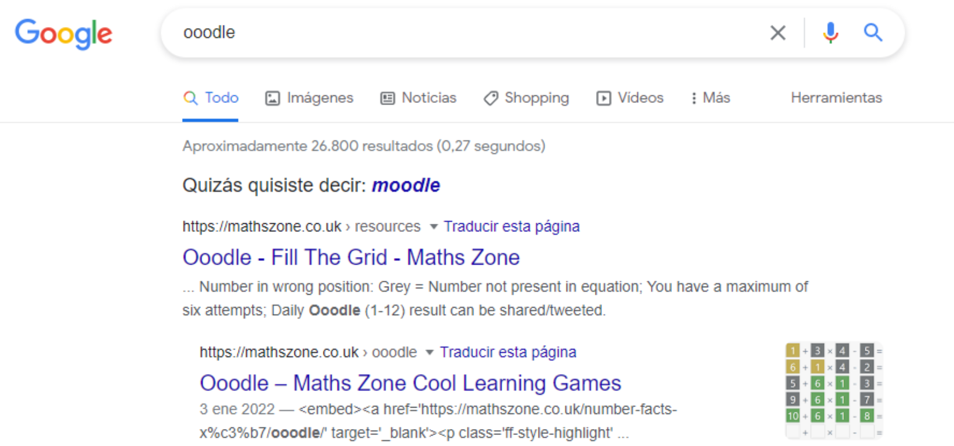



Figura 3.8: Búsqueda de la web de *Ooodle* desde el navegador.

El juego consiste en rellenar cuatro celdas con dígitos del 1 al 9 sin repetición. Hay un límite de seis intentos, donde cada intento consiste en una combinación de dígitos que se colocan en diferentes celdas constituyendo los operandos de una determinada operación. Cada intento debe ser una combinación de dígitos que cumpla una igualdad dada. Después de cada intento el color de los dígitos cambia ofreciendo una pista sobre lo acertado que ha sido el intento. Los colores posibles son:

- Verde: el dígito coloreada está en la lista de operandos ocupando esa misma posición.

- Amarillo: el dígito coloreada está en la lista de operandos ocupando una posición distinta.
- Gris: el dígito coloreada no está en la lista de operandos.

En cada intento el participante debe realizar una búsqueda de candidatos válidos, ejercitando así la fluidez, y también ha de ser flexible para poder utilizar aquello que conoces de los intentos anteriores, los dígitos con sus posiciones y color.

Ver la figura 3.9 para una descripción detallada del trascurso de una partida.

Aquellos participantes que resuelvan el problema pueden empezar un nuevo reto.

## Distribución de tiempos

A continuación, se indica, de forma orientativa, cuánto tiempo puede llevar realizar los diferentes pasos de la actividad.

1. Explicar en la pizarra en qué consiste la actividad: 5'.
2. Adivinar la operación oculta: 5-10'.

## Recursos

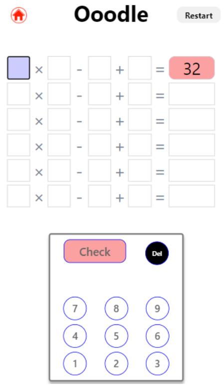
Para realizar la actividad los materiales necesarios son los siguientes:

- Dispositivo con conexión a internet.
- Conexión a internet.

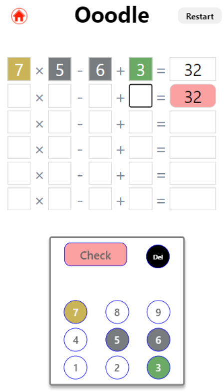
Además, es conveniente disponer de una pizarra para explicar cómo funciona el *Ooodle*, poner un ejemplo y resolver alguna duda que pueda surgir.

## Evaluación

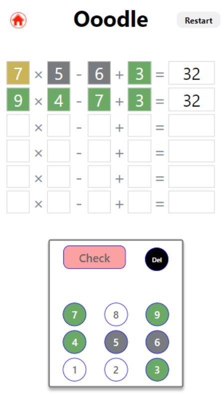
Al tratarse de un desafío, se supone que esta debe ser una actividad fluida y de corta duración. Por ello, más que una rúbrica como se ha planteado en actividades anteriores, aquí se sugiere que el profesor pase una encuesta a los alumnos para que estos indiquen el tiempo y el número de intentos que han necesitado para resolver el desafío. De este modo el profesor puede hacer un seguimiento sencillo de la evolución de los alumnos, y estos también podrán ver como van mejorando.




(a) Inicia la partida, se tienen seis intentos por delante y solo se conoce la operación y la igualdad que debe cumplirse. La operación debe ser igual a 32, dos números del 1 al 9 cuyo producto se acerca al 32 son el 7 y el 5, sobrarían 3 unidades que pueden obtenerse con el 3 y el 6.




(b) Del intento anterior se tiene que el 3 aparece en última posición, el 7 aparece en una posición distinta y ni el 5 ni el 6 estás presentes. El 4 y el 9 siguen abiertos, su producto está próximo a 32. Pueden ser buenos candidatos.




(c) Se ha encontrado la solución.

Figura 3.9: Descripción del sistema de intentos en el juego *Ooodle*.

### 3.5.2. *¿Cuál de ellas no encaja?*

*¿Cuál de ellas no encaja?* es un juego donde los participantes se encuentran frente a un conjunto de 4 imágenes y deben seleccionar, justificando su respuesta, aquella que menos encaja, la diferente.

La idea de este juego proviene del libro Danielson [2016], donde su autor, Christopher Danielson, incluye en cada una de las páginas un conjunto de 4 imágenes con diferentes formas geométricas. Estas figuras han sido seleccionadas cuidadosamente de tal modo que cualquiera de ellas puede ser la respuesta a la pregunta "¿Cuál de ellas no encaja?" (del inglés, *Which one doesn't belong*).

La clave de este juego es que no hay una única respuesta correcta, cualquiera de las 4 imágenes podría ser elegida. Por lo tanto, como todas las imágenes son válidas, los alumnos ponen el foco en razonar por qué motivo una u otra imagen debería ser excluida. Se trata de una actividad donde los alumnos trabajan la fluidez en la generación de ideas, la sensibilidad al buscar patrones en las imágenes, y también la originalidad al tratar de realizar asociaciones remotas.

# ¿Cuál de ellas no encaja?

---

Categoría: actividad Duración: 20-25'

## Objetivos

- Trabajar en equipo.
- Trabajar las siguientes dimensiones del pensamiento creativo que se vieron en la sección 1.2: sensibilidad, fluidez, originalidad, y evaluación.

## Instrucciones

Para el desarrollo de esta actividad ha de dividirse la clase en grupos de 3 o 4 personas. Idealmente, el profesor se encargará de designar los miembros de cada grupo. De este modo se puede cuidar que los grupos sean heterogéneos. Es muy conveniente que antes de revelar los miembros de cada grupo el docente explique la actividad y se asegure que todos lo han entendido.

No es necesario que los alumnos se agrupen desde el inicio de la actividad ya que la primera parte consiste en trabajar de forma individual.

Una vez el profesor ha explicado en qué consiste el juego, observar el conjunto de las 4 imágenes y dar diferentes motivos por las que una de ellas no encaja en ese conjunto; el profesor proyectará en la pizarra el conjunto de 4 imágenes con el que los alumnos trabajarán, o en su defecto repartirá una fotocopia por persona (o grupo).

A partir de aquí la actividad se divide en tres etapas:

1. Trabajo individual: cada alumno debe pensar 2 o 3 respuestas diferentes a la respuesta "¿cuál de ellas no pertenece?".

2. Puesta en común con los compañeros del grupo: se reúnen los miembros de cada grupo y ponen en común sus respuestas. A continuación deben debatir parar seleccionar hasta un máximo de 5 respuestas, las que sean a su juicio más originales.

3. Puestas en común con el resto de la clase: siguiendo un determinado orden, por medio de un representante, cada grupo debe dar una de sus respuestas al profesor. Si convencen al profesor obtendrán un punto. Las respuestas no pueden repetirse, por ello conviene que en el debate anterior cada grupo seleccione las más originales. Cuando todos los grupos hayan dado una respuesta, se iniciará una nueva ronda donde deberán dar una nueva respuesta. Así hasta un total de 5 rondas.

Nótese que puede haber más de un ganador, incluso podría darse la situación de que todos los grupos obtengan 5 puntos (la máxima puntuación posible). Si se desea, el profesor puede desempatar la situación pidiendo una nueva respuesta y dando la victoria al grupo que aporte la solución más original.

## Distribución de tiempos

A continuación, se indica, de forma orientativa, cuánto tiempo puede llevar realizar los diferentes pasos de la actividad.

- Explicar en qué consiste la actividad, detallar los pasos y resolver las dudas que puedan surgir: 5'.
- Repartir los conjuntos de imágenes: 2'.
- Trabajo individual: 3'.
- Debate grupal: 5'.
- Puesta en común: 15'.

En total suman 30' minutos, no obstante una vez los alumnos han aprendido la dinámica de la actividad este tiempo se reduciría a unos 20-25 minutos.

Asimismo, si se desea agilizar la actividad podría reducirse el número de rondas de 5 a 3 por ejemplo. Con ello se reduciría el tiempo de debate y puesta en común. Más aún, podría eliminarse la parte de puesta en común y el profesor pedir que cada grupo entregue, de forma anónima, la respuesta más original, leer las respuestas y destacar las que él considere más interesantes.

## Recursos

Para realizar la actividad los materiales necesarios son los siguientes:

- Conjunto de 4 imágenes elegidas a conciencia para dar juego con la pregunta "¿Cuál de ellas no encaja?" (ver figura 3.11).
- Proyector o una fotocopia por persona (o grupo) para que los alumnos puedan ver las cuatro imágenes de esta actividad.

Un recurso muy útil para preparar esta actividad es la página web `https://wodb.ca/`, donde pueden encontrarse multitud de conjuntos de imágenes idóneos.

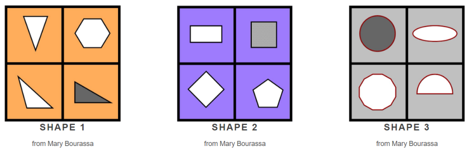
Algunos de ellos consisten en cuatro números, cuatro gráficas de funciones, cuatro figuras geométricas, incluso cuatro objetos distintos.

## Evaluación

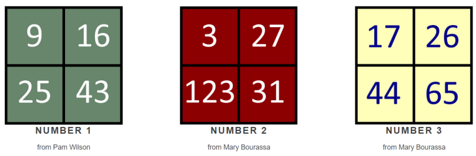
Un modo de evaluar esta actividad sería cumplimentar la rúbrica de la figura 3.10.

| Etapa de la actividad | Objetivo | Evaluación | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| Paso 1 | Generación de ideas | Número de respuestas presentadas | 0 | 1 | Entre 2 y 3 | Más de 3 |
| | Evaluación de las ideas | Las respuestas son coherentes | Nada | Ligeramente | Bastante | Totalmente |
| Paso 2 | Trabajo colaborativo | Toma de decisiones | No hay colaboración | Una minoría | La mayoría | Todos |
| Paso 3 | Originalidad | Los grupos tienen ideas originales | Generalmente no | 1 respuesta original | 2-3 respuestas originales | Todas son originales |

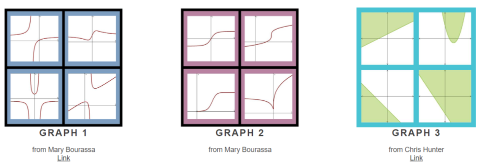
Figura 3.10: Ejemplo de rúbrica para evaluar el desarrollo de la actividad 3.5.2 *¿Cuál de ellas no encaja?*




(a) Ejemplos cuya temática son las figuras geométricas.




(b) Ejemplos cuya temática son los números.




(c) Ejemplos cuya temática son las gráficas.

Figura 3.11: Ejemplos de conjuntos de imágenes para la actividad 3.5.2 *¿Cuál de ellas no pertenece?* Imágenes tomadas de la página web `https://wodb.ca/`

### 3.5.3. *El ahorcado matemático*

Esta actividad no es más que una sencilla adaptación al contexto matemático del famoso juego de *El ahorcado*. En dicho juego los participantes competían por adivinar una palabra oculta, para ello contaban únicamente con el número de letras que esta tiene y un contexto temático de la misma. En cada turno un participante debía dar una letra que él pensase que aparecería en la palabra oculta. Si fallaba, si le letra no se encontraba en la palabra, entonces se dibuja un nuevo miembro del ahorcado (cabeza, torso, brazos, piernas, ...). El juego terminaba al adivinar la palabra oculta o al completarse la figura del ahorcado.

En esta nueva versión del juego, los participantes competirán por adivinar un número oculto, sabiendo únicamente el número de dígitos que tiene. En esta ocasión, los participantes no deben tratar de adivinar los dígitos sino que han de formular preguntas dicotómicas, que solo acepten como respuesta "SI" o "NO", adoptando la mecánica del conocido juego *¿Quién es quién?*.

Dada la simpleza del juego y sus mecánicas, esta actividad está pensada para los más jóvenes de la E.S.O. Se trata de un recurso estupendo para practicar el concepto de divisibilidad.

# El ahorcado matemático

---

Categoría: desafío Duración: 5-10'

## Objetivos

- Fomentar un clima de compañerismo.
- Trabajar las siguientes dimensiones del pensamiento creativo que se vieron en la sección 1.2: originalidad, y evaluación.

## Instrucciones

Para realizar la actividad el profesor explicará en qué consiste y luego dividirá la clase en dos o cuatro grupos.

En primer lugar, el profesor debe pensar un número de entre 3 y 6 cifras. En segundo lugar, el profesor ha de dibujar la horca en la pizarra (ver figura 3.12) y pintar tantos guiones como cifras tiene el número.

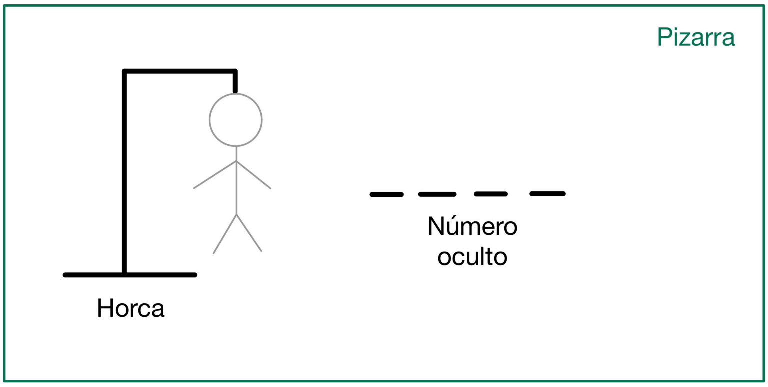



Figura 3.12: Dibujo del ahorcado para un número oculto de 4 cifras. De color gris aparece pintada la silueta del ahorcado, que irá completándose por partes durante la actividad.

A continuación empiezan las rondas donde cada grupo, de forma ordenada, puede tratar de adivinar el número oculto o formular al profesor una pregunta dicotómica sobre dicho número. Si la respuesta a la pregunta es positiva (por ejemplo, “SI” o “Verdadero”) entonces el turno de palabra pasa al siguiente grupo. Por el contrario, si la respuesta es negativa (por ejemplo, “NO” o “Falso”) entonces el profesor dibujará un nuevo miembro del ahorcado. Después de cada pregunta, y antes de pasar el turno al siguiente grupo, el profesor ha de apuntar en la pizarra la pregunta que se acaba de formular y su repuesta. Así los alumnos tienen la información recogida en un lugar concreto y además se les da espacio para pensar su siguiente cuestión.

Ejemplos de preguntas que los alumnos pueden pensar en formular son:

- ¿El número oculto es par?
- ¿El número oculto tiene algún cero?

- ¿El número oculto es divisible por tres?
- ¿El número oculto empieza por uno?

El ahorcado tiene 6 miembros: una cabeza, un torso, dos brazos y dos piernas. El juego termina cuando un grupo se proclama vencedor al adivinar el número, o cuando se completa la figura del ahorcado y quien gana es el profesor.

## Distribución de tiempos

Se trata de una actividad muy rápida y dinámica, según el número de dígitos que tenga el número oculto puede tardarse más o menos. No obstante, una vez los alumnos saben cómo jugar, la actividad debería completarse en un máximo de 10 minutos. Si se concluye muy rápido este desafío el profesor puede pensar un nuevo número y empezar de nuevo.

## Recursos

Para realizar esta actividad tan solo es necesario una pizarra y rotuladores o tiza para escribir en ella.

## Evaluación

Al tratarse de un desafío, se supone que esta debe ser una actividad fluida y de corta duración. Por ello, más que una rúbrica como se ha planteado en actividades, aquí se propone que el profesor preste atención a las preguntas de los alumnos y observe si están atentos y formulan preguntas coherentes, teniendo en cuenta la información que ya conocen. De este modo el profesor puede comprobar si los alumnos dominan el concepto de divisibilidad, por ejemplo si saben que el número oculto es par entonces no tiene sentido que pregunten si termina en 0, etc.

### 3.5.4. *Dibujando con matemáticas*

La actividad que se propone a continuación consiste en realizar dibujos a partir de funciones reales de una variable en el programa gratuito Desmos.

Desmos es una calculadora gráfica *online* gratuita que, entre otras muchas cosas, permite representar funciones.

Esta actividad tiene la intención de motivar al alumnado para que haga uso de recursos *online* gratuitos, como este mismo, que, si son bien empleados, pueden ser muy beneficiosos para su formación. Estos recursos tienen sobre todo un enorme potencial para que los alumnos se sientan cómodos estudiando de forma autónoma.

Esta actividad requiere un nivel de abstracción mayor que en las anteriores y está pensada para los más mayores de la E.S.O. (tercero y cuarto), Bachillerato y Formación Profesional. Sin embargo, puede simplificarse para que los más jóvenes puedan disfrutar de ella, por ejemplo limitando las funciones disponibles a rectas verticales y horizontales.

En el apartado "Instrucciones" que viene en la ficha de esta actividad no solo se explica las diferentes etapas que la componen sino que incluye una breve guía para iniciar al lector en el uso de Desmos. Con esta finalidad, dicho apartado se ha dividido en dos, uno correspondiente a una primera sesión introductoria (la guía de iniciación) y otro destinado a explicar la actividad como tal (realizar un dibujo representando dibujos en Desmos).

# Dibujando con matemáticas

Categoría: actividad Duración: 50-60'

## Objetivos

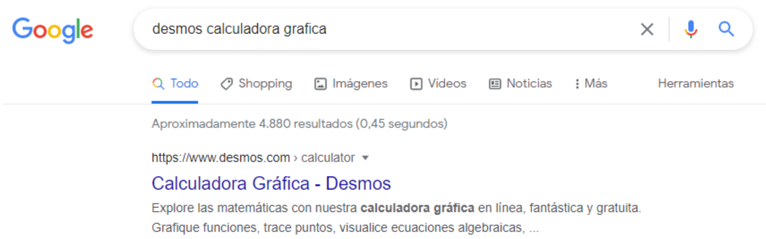
- Trabajar en equipo.
- Aprender a utilizar software para representar funciones.

- Trabajar las siguientes dimensiones del pensamiento creativo que se vieron en la sección 1.2: sensibilidad, fluidez, originalidad, y flexibilidad.

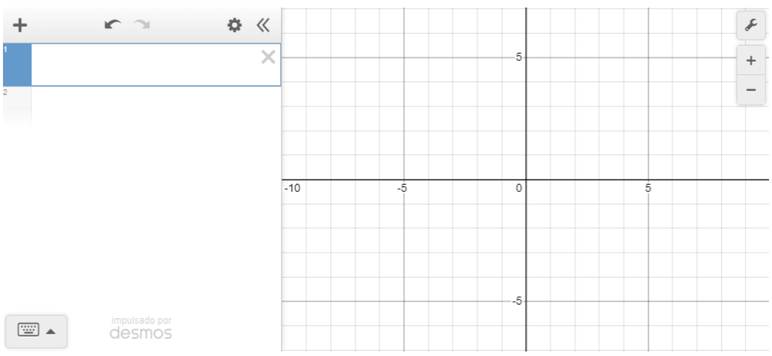
# Instrucciones

## Sesión introductoria

Lo primero será acceder a la página web de Desmos, se puede acceder bien mediante el siguiente enlace `https://www.desmos.com/calculator`, o bien buscando en el navegador los términos: "desmos calculadora gráfica". Una vez dentro de la página web debería verse una gráfica vacía (ver figura 3.13).



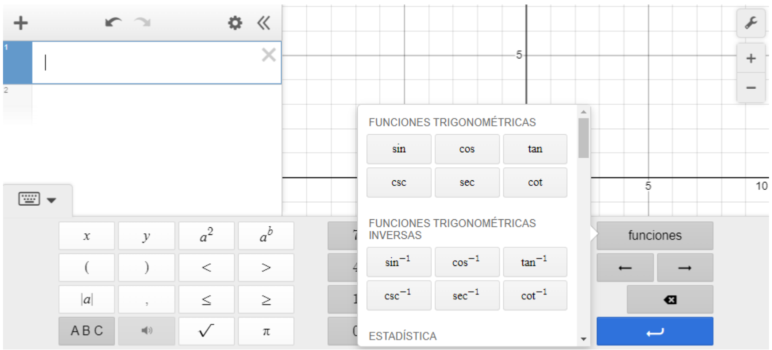

(a) Búsqueda de la web de Desmos desde el navegador.



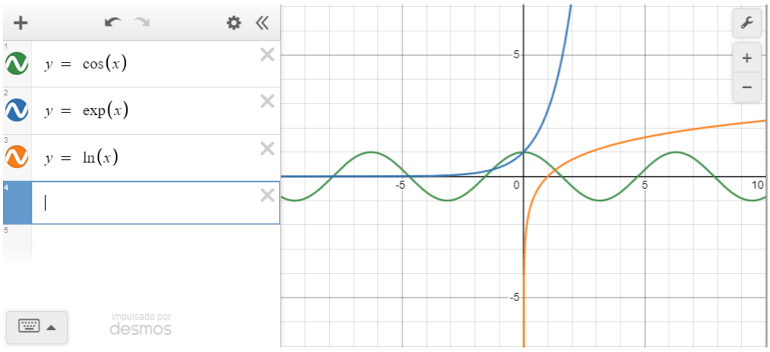

(b) Pantalla principal de la calculadora gráfica de Desmos.

Figura 3.13: Acceso a la calculadora gráfica de Desmos `https://www.desmos.com/calculator`

En el panel izquierdo hay una columna de celdas que admiten texto. Es en dichas celdas donde pueden escribirse las funciones que se deseen representar. Las expresiones pueden escribirse desde el teclado de forma natural, o también puede hacerse uso del teclado auxiliar en la esquina inferior izquierda que contiene las funciones ya escritas (ver figura 3.14a). Pueden escribirse varias funciones simultáneamente como se aprecia en la figura 3.14b.




(a) Teclado auxiliar de Desmos con las funciones ya escritas.



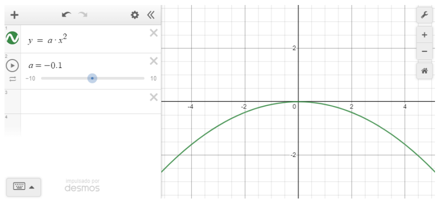

(b) Representación simultánea de varias funciones.

Figura 3.14: Representación de funciones con Desmos

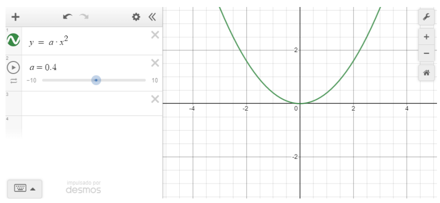
Aunque parezca muy simple, esta calculadora gráfica ofrece multitud de opciones. Por ejemplo, como se muestra en la figura 3.15 la inserción de parámetros, o la definición del dominio de la función (valores que puede tomar la variable independiente $x$).

Además de funciones, esta calculadora gráfica también permite dibujar curvas en el plano bidimensional como se puede apreciar en la figura 3.16.

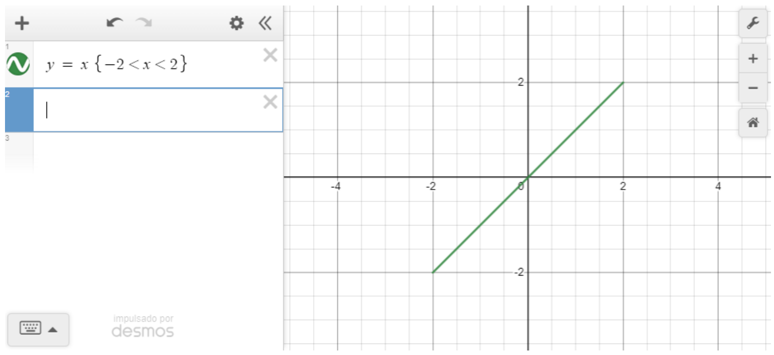
En esta primera sesión introductoria se propone que los alumnos tengan una primera toma de contacto con la herramienta, para ello se les explicará como funciona y luego se les pedirá que de forma individual dibujen lo siguiente (ver figura 3.17):




(a) Función $y = a \cdot x^2$ con $a = -0{,}1$.




(b) Función $y = a \cdot x^2$ con $a = 0{,}4$.




(c) Representación de la recta $y = x$ en el dominio $(-2, 2)$.

Figura 3.15: Opciones de la calculadora gráfica de Desmos

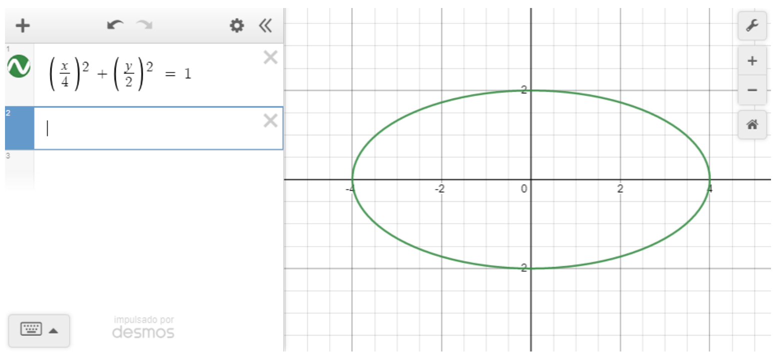



Figura 3.16: Representación de curvas en Desmos

1. La recta que pasa por los puntos $(0, 2)$ y $(2, 0)$.

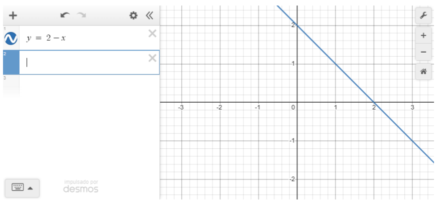
2. La función seno.

3. Una función cuadrática cualquiera que pase por el punto $(0, -1)$.

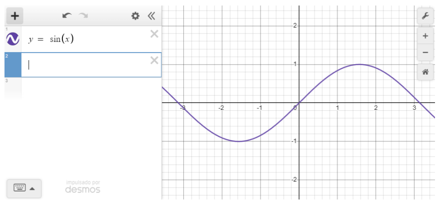
4. La función tangente definida en $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$.

5. La función definida a trozos: $f(x) = \exp +x$ en $[-\infty, 0]$, y $f(x) = \exp -x$ en $[0, \infty]$.

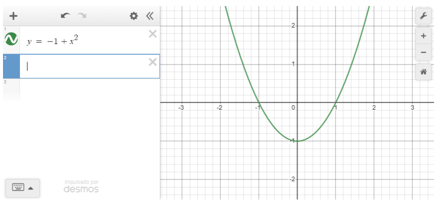
6. El círculo de radio unidad y centro en el punto $(0, 1)$.



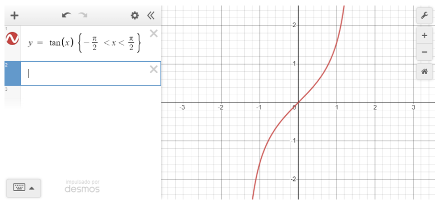
(a) La recta que pasa por los puntos $(0, 2)$ y $(2, 0)$.



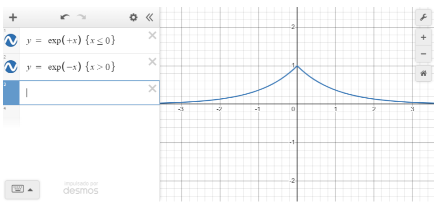
(b) La función seno. Una función cuadrática cualquiera que pase por el punto $(0, -1)$.



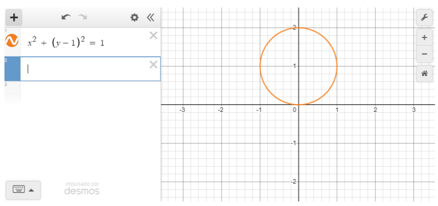
(c) Una función cuadrática cualquiera que pase por el punto $(0, -1)$.



(d) La función tangente definida en $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$.



(e) La función definida a trozos: $f(x) = \exp +x$ en $[-\infty, 0]$, y $f(x) = \exp -x$ en $[0, \infty]$.



(f) El círculo de radio unidad y centro en el punto $(0, 1)$

Figura 3.17: Ejercicios para introducirse en el uso de la calculadora gráfica de Desmos

### Actividad principal

En primer lugar cabe destacar que esta actividad puede realizarse tanto de forma individual como por parejas. En este segundo caso puede ser conveniente que el profesor escoja los miembros de las parejas para que así estas estén bien niveladas.

Una vez se han formado las parejas, en el caso de que las haya, el profesor empezará explicando en qué consiste la actividad. Es recomendable que durante este tiempo los ordenadores vayan encendiéndose si no lo estaban. Posteriormente el profesor entregará a cada pareja una fotocopia con una imagen. También puede colgarse en internet, por ejemplo en el campus virtual (si lo hay), un archivo PDF con las imágenes asignadas a cada carpeta. La actividad consiste en dibujar dicha imagen mediante funciones en Desmos.

La actividad se divide en tres etapas:

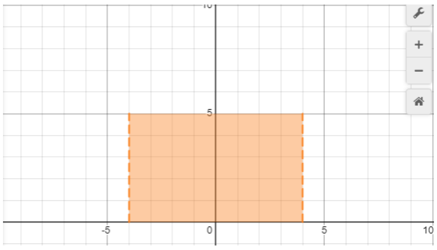
1. Primeramente los alumnos deben pintar sobre la imagen, bien en papel con un bolígrafo, o bien con un editor de imágenes como Paint; los contornos más importantes de la imagen. Se trata de hacer un boceto con figuras geométricas, que serán las que representarán en Desmos. De este modo en los siguientes pasos lograrán centrarse en lo importante y no perderse con detalles menores.

2. En segundo lugar, los alumnos deben realizar un ejercicio de abstracción para separar los contornos de la imagen en diferentes contornos que puedan dibujar en Desmos. En este punto trabajan la fluidez en la generación de ideas y también su posterior evaluación. En ocasiones habrá figuras que puedan dibujarse de diferentes modos y deberán escoger el que más les convenga. En este apartado también se trabaja la originalidad, ya que las soluciones más simples pueden requerir un mayor trabajo (más funciones a dibujar) mientras que otras menos evidentes pueden ahorrar mucho tiempo.

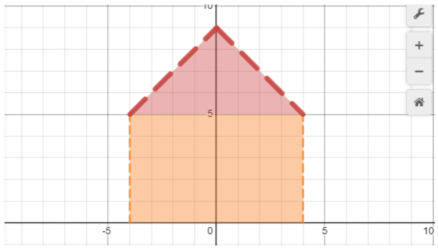
3. Por último, si todavía queda tiempo disponible, los alumnos pueden darle un toque personal al dibujo. Añadiendo nuevos detalles que no aparecían en la

imagen original.

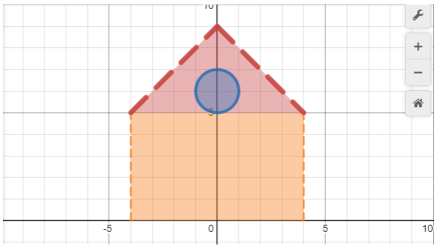
A continuación, a modo de ejemplo, en la figura 3.18 se muestra un sencillo dibujo de una casa paso a paso:



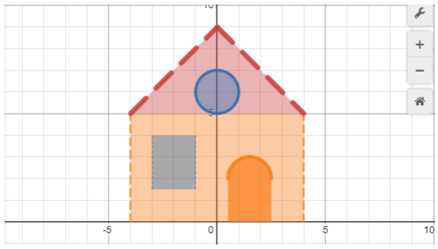
(a) La base de la casa viene dada por sencillas inecuaciones $-4 \leq x \leq 4$ y $0 \leq y \leq 5$.



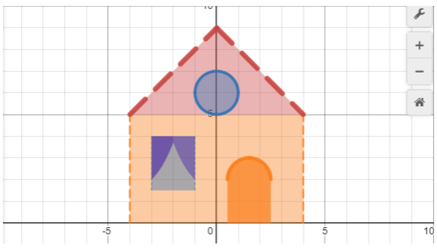
(b) El tejado viene dado por la norma $|x| + |y - 5| \leq 4$ sujeto a la condición $5 \leq y$.



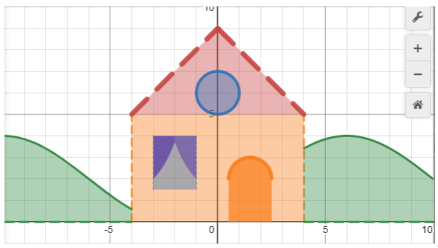
(c) La ventana del tejado se dibuja recurriendo a la ecuación del círculo, desplazándolo sobre el eje vertical: $x^2+(y-6)^2 \leq 1$



(d) La ventana y la puerta se han construido de forma similar a la base de la casa, además a la puerta se le ha añadido un semicírculo en la parte superior.



(e) Para dibujar la ventana se ha recurrido a funciones exponenciales.



(f) Para las montañas se ha utilizado la función coseno: $0 < y \leq 2 \cdot \cos\left(\frac{x}{8} \cdot \pi + 5\frac{\pi}{4}\right) + 2$

Figura 3.18: Ejemplo de dibujo con funciones en la calculadora gráfica de Desmos.

# Distribución de tiempos

A continuación, se indica, de forma orientativa, cuánto tiempo puede llevar realizar los diferentes pasos de los dos tipos de actividades.

### Sesión introductoria

- Explicar en qué consiste la actividad, encender ordenadores y acceder a Desmos: 5-10’.
- Introducción a Desmos: 15-20’.
- Realizar los ejercicios propuestos: 20’.

### Actividad principal

- Explicar en qué consiste la actividad, encender ordenadores y acceder a Desmos: 5’.
- Agruparse por parejas: 5’.
- Detectar las figuras más relevantes de la imagen: 10’.
- Realizar el dibujo: 30’.

# Recursos

Para realizar la actividad los materiales necesarios son los siguientes:

- Dispositivo con conexión a internet.
- Conexión a internet.

Además, es conveniente disponer de una pizarra o proyector para explicar cómo funciona la calculadora gráfica de Desmos.

# Evaluación

Un modo de evaluar esta actividad sería cumplimentar la rúbrica de la figura 3.19.

| Etapa de la actividad | Objetivo | Evaluación | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| Paso 1 | Sensibilidad | Los contornos seleccionados... | No son un reflejo fiel de la imagen | Se pierden con los detalles | Correctos pero demasiados | Correctos y necesarios |
| Paso 2 | Fluidez y evaluación de ideas | Deducen las funciones para representar figuras... | Reciclando la misma función simple una y otra vez | A base de prueba y error | Buscando por internet ecuaciones de diferentes funciones | Rápidamente ya que conocen bien las funciones |
| | Originalidad | Las figuras que se dibujan... | No reflejan aquello que quieren representar | Se entiende el dibujo pero emplean demasiadas funciones | Recurren a funciones ingenuas que permiten ahorrar tiempo | Además de recurrir a funciones ingenuas, tienen un toque personal |
| Transversal | Trabajo Colaborativo | Los integrantes del grupo colaboran | No hay colaboración | Hay escasa colaboración | Hay colaboración pero la mayor parte del trabajo la realiza la misma persona | Hay reparto de tareas y ambos trabajan por igual |
| | Trabajo Colaborativo | Toma de decisiones | Impuestas por un integrante | Un integrante cede sin ser convencido | Un integrante cede habiendo sido convencido | Hay consenso integrando ideas de ambos |

Figura 3.19: Ejemplo de rúbrica para evaluar el desarrollo de la actividad 3.5.4 *Dibujando con matemáticas*.

A continuación se muestra una imagen con dibujos generados mediante la calculadora gráfica de Desmos en una clase de Matemáticas [Valencia, 2022], figura 3.20. Cabe destacar que fue al ver estas imágenes cuando surgió la idea de proponer esta actividad.

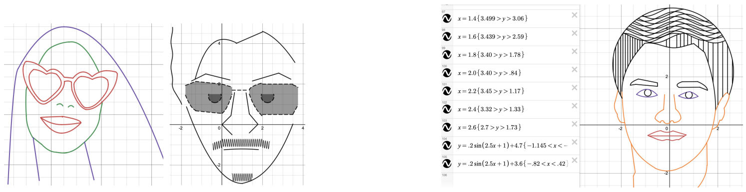


Figura 3.20: Representación de retratos faciales con Desmos, imágenes tomadas de Valencia [2022].

## 3.6. Taller: iniciación a la programación

En esta sección se propone una serie de actividades para, de forma simultánea, introducir a los alumnos en el mundo de la programación y trabajar la creatividad.

Se ha optado englobar todas estas actividades en un mismo bloque, dándole una estructura de taller, ya que tiene un elemento en común, la programación. Este taller podría llevarse a cabo en horas extraescolares, o también durante el periodo que se encuentra entre el final de la tercera evaluación y la última fecha del calendario académico.

La presente sección se ha dividido en tres partes, en primer lugar se proporcionará una base teórica, en segundo lugar se realiza una introducción al lenguaje de programación Python, y por último se propone un proyecto.

Se ha escogido Python por ser un lenguaje de código abierto respaldado por una gran comunidad de usuarios. Esto es, se trata de un recurso gratuito y popular. En consecuencia se han desarrollado multitud de recursos a disposición del público general. Uno de esos recursos se llama Google Colab, una herramienta desarrollada por Google que permite programar en Python desde el navegador sin necesidad de instalar ningún programa adicional.

Las actividades que aquí se proponen son aptas para todos los cursos, aunque, naturalmente, por lo general los más jóvenes necesitarán más tiempo para completarlas con éxito ya que están empezando a desarrollar el pensamiento abstracto.

**Observación.** Todos los ejemplos, ejercicios y problemas que se plantean en esta sección aparecen en el apéndice A con el mismo formato que se vería en Google Colab y con los resultados de las celdas ejecutadas.

### 3.6.1. Base teórica

De forma simple, programar consiste en definir una serie de instrucciones (código) que el ordenador deberá seguir para completar una acción.

El ser humano tiene la facultad de comunicarse con los demás gracias al lenguaje, de forma análoga, los humanos pueden comunicarse con el ordenador a través de lenguajes de programación. Para comunicarse con el ordenador y darle las instrucciones hay que escoger un lenguaje de programación. No obstante, la base teórica es similar en cualquier lenguaje de programación. A continuación de detallan los elementos más básicos de la programación.

Una **variable** es un espacio de la memoria interna del ordenador que se utiliza para almacenar valores con la intención de recuperarlos más tarde. Las variables pueden ser de varios tipos, entre ellos:

1. Boleanas, *bool* del inglés *boolean*, son estados lógicos (verdadero o falso).
2. Enteras, *int* del inglés *integer*, son números enteros.
3. De punto flotante, *float* del inglés *floating point*, esencialmente son números decimales.
4. Caracteres, *char* del inglés *character*, se trata de letras sueltas.
5. Palabras, *str* del inglés *string*, son cadenas de caracteres.
6. Listas, *list* en inglés, son variables que sirven para almacenar otras variables.

Entre variables del mismo tipo pueden realizarse operaciones, por ejemplo pueden sumarse letras para formar palabras, en el apartado 3.6.2.1 se definen los **operadores** más comunes.

Los **condicionales** en programación son estructuras que determinan si una parte de las instrucciones debe ejecutarse o no, o si deben ejecutarse otras en su lugar.

Los **bucles** son estructuras que leen de forma repetida un conjunto de instrucciones hasta que se cumple una condición determinada.

Las **funciones** son estructuras que permiten dividir el trabajo de un programa (el total de instrucciones), en trabajos más pequeños (en subgrupos de instrucciones) separados de la parte principal.

Un **algoritmo** es un conjunto de instrucciones ordenadas que permiten resolver un problema concreto.

### 3.6.2. Introducción a Python

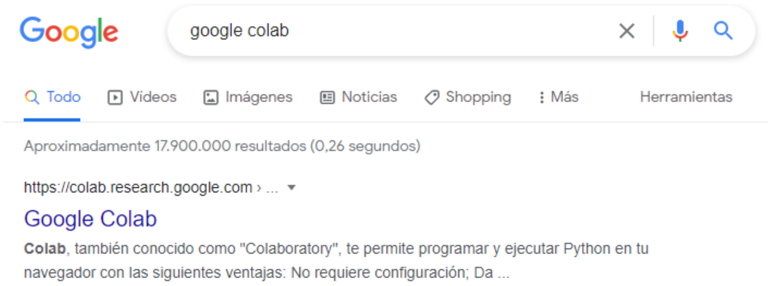
Python es un lenguaje de programación, pero para comunicarse con el ordenador es necesario un entorno de programación. Esto es, un medio que permita la comunicación del individuo con la máquina. Para entenderlo mejor, los humanos nos comunicamos en un idioma, por ejemplo el español, pero para que se produzca esa comunicación es necesario que exista un medio, por ejemplo el aire o una carta.

Aquí se propone hacer uso de Google Colab dado que es un recurso gratuito, de fácil acceso y solo requiere disponer de un navegador web como Google Chrome (no hay que instalar nada de forma adicional). Es un recurso magnífico para educación.

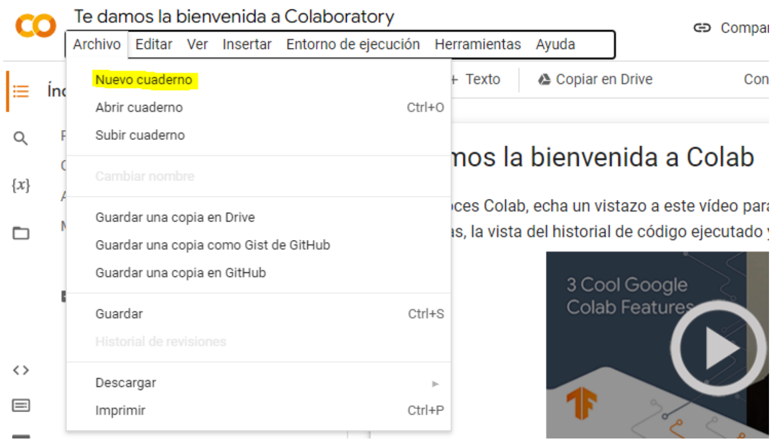
Para acceder a Google Colab hay que acceder a la siguiente dirección url: `https://colab.research.google.com` o buscar en el navegador "Google Colab", ver figura 3.21a. Para empezar a escribir código hay que seleccionar en el menú superior "Archivo", y luego pulsar sobre "Nuevo cuaderno" (ver figura 3.21b). Pedirá acceso a la cuenta de Google, iniciar sesión permitirá guardar el trabajo realizado.

Los cuadernos de Google Colab están compuestos por celdas de dos tipos, unas en las que puede insertarse código (instrucciones para el ordenador), y otras en las que puede insertarse un comentario de texto. Los comentarios de texto no son leídos por el ordenador como las instrucciones, la máquina los omite, y permiten explicar las instrucciones que se están proporcionando al ordenador para que cualquier individuo pueda entender lo que se está haciendo. Las celdas van seguidas unas de otras y su ejecución es independiente. Asimismo, es posible añadir comentarios de texto dentro de las celdas de código, para ello debemos añadir el símbolo # delante del texto que queremos comentar. Con la almohadilla # se está indicando al ordenador que esa línea de código es un comentario y por tanto debe saltársela (no interpretarla como una instrucción).

En la figura 3.22 se muestra un esquema sencillo de las celdas de Google Colab. Se distinguen los dos tipos de celdas y en la celda de tipo código aparece un




(a) Búsqueda de la web de Google Colab desde el navegador.




(b) Pantalla de introducción a Google Colab y selección de “Nuevo cuaderno”.

Figura 3.21: Acceso a Google Colab, `https://colab.research.google.com`

comentario y una instrucción (cálculo de $5 \cdot 7 - 2$). Para ejecutar una celda basta con pulsar el botón *play* (triángulo blanco sobre fondo gris) que se encuentra en la esquina superior izquierda de la celda. Nótese que la primera ejecución suele tardar un poco más de tiempo ya que nuestro ordenador debe conectarse a los servidores de Google. El resultado de la ejecución aparece debajo de la celda ejecutada. Para añadir una nueva celda basta con pasar el cursor del ratón sobre la parte inferior (o superior) de una celda e indicar qué tipo de celda queremos insertar abajo (o arriba) de la misma.

Para empezar, puede escribirse lo siguiente en una celda de tipo código:

```
print(''Hola mundo.'')
```

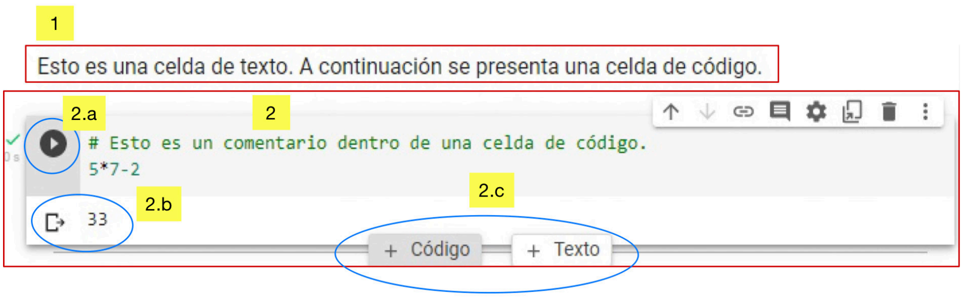



Figura 3.22: Esquema de las celdas de Google Colab: 1 celda de tipo comentario, 2 celda de tipo código con una primera línea comentada y la segunda linea contiene una instrucción, 2.a botón para ejecutar la celda, 2.b resultado de ejecutar la celda, 2.c opción para insertar una nueva celda debajo.

Con este código se está llamando a la función **`print()`** que se encarga de revelar al usuario el contenido que hay en su parámetro de entrada. En este caso se trata de un ejemplo trivial, sin embargo esta función es muy útil cuando quiere conocerse el valor que almacena una variable. Se aconseja no perderla de vista.

A continuación se describen los operadores más básicos que existen en Python, y cuya comprensión es esencial para realizar las actividades de forma correcta.

#### 3.6.2.1. Operadores en Python

Un **operador aritmético** toma dos operandos como entrada, realiza un cálculo con ellos y devuelve el resultado. Por ejemplo, en la sentencia `1+2`, `1` y `2` son los operandos y el símbolo + es el operador aritmético. En la figura 3.23 se encuentran los diferentes operadores aritméticos.

| **Nombre** | **Sintaxis** | **Definición** | **Ejemplo de sentencia** | **Resultado de la sentencia** |
|---|---|---|---|---|
| Suma | `+` | Adición de los dos operandos | `3+4` | `7` |
| Resta | `-` | Substracción entre dos operandos | `2-3` | `-1` |
| Producto | `*` | Producto entre dos operandos | `2*3` | `6` |
| División | `/` | División entre dos operandos | `2/3` | `1.5` |
| Módulo | `%` | Devuelve el resto de la división entre dos operandos enteros | `22%3` | `1` |
| Potencia | `**` | Calcula la potencia entre dos operandos | `2**3` | `8` |

Figura 3.23: Operadores aritméticos en Python

Un **operador relacional** toma dos operandos como entrada, los compara y devuelve un resultado de tipo de tipo estado, *bool*, verdadero o falso. Por ejemplo, en la sentencia `1<2`, `1` y `2` son los operandos y el símbolo < es el operador relacional. En la figura 3.24 se encuentran los diferentes operadores relacionales.

| Nombre | Sintaxis | Definición | Ejemplo de sentencia | Resultado de la sentencia |
|---|---|---|---|---|
| Mayor que | `>` | Comprueba si el primer operando es mayor que el segundo | `3>3` | `False` |
| Menor que | `<` | Comprueba si el primer operando es menor que el segundo | `2<3` | `True` |
| Igual que | `==` | Comprueba si ambos operandos son iguales | `2==3` | `False` |
| Mayor o igual que | `>=` | Comprueba si el primer operando es mayor o igual que el segundo | `2>=3` | `False` |
| Menor o igual que | `<=` | Comprueba si el primer operando es menor o igual que el segundo | `2<=2` | `True` |
| Distinto | `!=` | Comprueba si ambos operandos son distintos | `2!=3` | `True` |

Figura 3.24: Operadores relacionales en Python

Un **operador de asignación** toma una variable y un valor como entrada, dicho valor se almacena en la variable o modifica su contenido. Por ejemplo, en la sentencia `a = 2`, `a` es la variable, `2` el valor, y el símbolo = es el operador de asignación. En la figura 3.25 se encuentran los diferentes operadores de asignación.

| Nombre | Sintaxis | Definición | Ejemplo de sentencia | Resultado de la sentencia |
|---|---|---|---|---|
| Asignación | `=` | Almacena un valor en una variable | `a = 2*6` | Almacena el valor `12` en la variable `a` |
| Asignación aditiva | `+=` | Suma el valor a la variable y sobrescribe el resultado | `a += 1` | `a` pasa a tener asignado el valor `13`. |

Figura 3.25: Operadores de asignación en Python

Un **operador lógico** se utiliza en las estructuras condicionales para evaluar varias condiciones simultáneamente. Este operador toma como entrada un o dos operandos de tipo estado (verdadero o falso) y devuelve un resultado de tipo estado basado en estos operandos. En la figura 3.26 se encuentran los diferentes operadores lógicos. Por ejemplo, `2<3 and 4 == 6` tiene dos operandos de tipo estado, que vienen dados a su vez por operadores relacionales, `2<3` (verdadero) y `4 == 6` (falso); y el operador lógico es `and`. En la figura 3.26 se encuentran los diferentes operadores lógicos.

El **operador de pertenencia** `in` se utiliza para comprobar si un elemento se encuentra dentro de lista. En la figura 3.27 se encuentra un ejemplo de este opera-

| Nombre | Sintaxis | Definición | Ejemplo de sentencia | Resultado de la sentencia |
|---|---|---|---|---|
| Conjunción | `and` | Comprueba si los dos estados son verdaderos | `2 <3 and 4 == 6` | `False` |
| Disyunción | `or` | Comprueba si alguno de los dos estados es verdadero | `2 <3 or 4 == 6` | `True` |
| Negación | `not` | Alterna (niega) el estado recibido | `not 4 == 6` | `True` |

Figura 3.26: Operadores lógicos en Python

dor.

| Nombre | Sintaxis | Definición | Ejemplo de sentencia | Resultado de la sentencia |
|---|---|---|---|---|
| Pertenencia | `in` | Comprueba si un elemento está en una lista | `5 in [ 7, 23, 5, 2, 2, 5, 6]` | `True` |

Figura 3.27: Operador de pertenencia en Python

A continuación se presentan una serie de ejercicios resueltos para aprender a utilizar los operadores básicos de Python.

**Ejemplo 1** (Operadores aritméticos)**.**

**1.** Sumar $545 + 65 + 6$.

```
545 + 65 + 6
```

**2.** Restar $5 - 7$.

```
5 - 7
```

**3.** Multiplicar $2 \cdot 2 \cdot 2$.

```
2*2*2
```

**4.** Dividir $115/7$.

```
115/7
```

**5.** Obtener el resto de dividir $115/7$.

```
115%7
```

**6.** Calcular $2^3$.

```
2**3
```

**7.** Sumar $2 + 9$, multiplicarlo por $3$ y elevar el resultado a la quinta potencia.

```
((2 + 9)*3)**5
```

**8.** Restar a $115$ el resto de dividir $115/7$ y dividir el resultado entre $7$.

```
(115 - (115%7))/7
```

**Ejemplo 2** (Operadores relacionales)**.**

**9.** Comprobar si $\frac{2}{5}$ es mayor que $\frac{3}{7}$.

```
2/5 > 3/7
```

**10.** Comprobar si $2^{10}$ es menor que $1000$.

```
2**10 < 1000
```

**11.** Comprobar si $\frac{3}{7}$ es igual que $\frac{39}{91}$.

```
3/7 == 39/91
```

**12.** Comprobar si $913$ es múltiplo de $11$.

```
913//11 == 0
```

**Ejemplo 3** (Operadores asignación)**.**

**13.** Guarda el resultado de multiplicar $81 \cdot 13$ en la variable `a`. Luego guarda en la variable `b` el resultado de multiplicar `a` por $10$. Copia el valor de `b` en la variable `a`. Muestra el valor de `a` con la función `print()`.

```
a = 81*13
b = a*10
a = b
print(a)
```

**14.** Repite el ejemplo anterior sin recurrir a una segunda variable. Puedes sobrescribir el valor directamente sobre la variable `a`.

```
a = 81*13
a = a*10
print(a)
```

**15.** Utiliza el operador asignación para hallar la parte entera de la división $115/7$. Para ello, guarda el resto de dividir $115/7$. Réstaselo a $115$, guarda el valor y divídelo entre $7$. Muestra el resultado con la función `print()`, debería ser un número entero.

```
a = 115%7
a = 115 - a
a = a/7
print(a)
```

**Ejemplo 4** (Operadores lógico)**.**

**16.** Comprueba si 3 y 7 son divisores de 273. Puedes hacer uso del operador asignación para almacenar valores intermedios.

```
a = 273%3
b = 273%7
(a == 0) and (b==0)
```

**17.** Comprueba si 3 o 13 son divisores de 279. Puedes hacer uso del operador asignación para almacenar valores intermedios.

```
a = 279%3
b = 279%13
(a == 0) or (b==0)
```

**18.** Comprueba que 13 no es divisor de 279.

```
not (279%13 == 0)
```

**Ejemplo 5** (Operador de pertenencia)**.**

**19.** Comprueba si 68 está en la lista de los 5 primeros múltiplos de 17.

```
68 in [17, 2*17, 3*17, 4*17, 5*17]
```

#### 3.6.2.2. Condicionales en Python

La estructura condicional más simple en Python se conoce como condicional `if`, evalúa un estado lógico y si es verdadero entonces permite al ordenador leer una serie de instrucciones para que las ejecute, mientras que si el estado evaluado es negativo (falso) entonces esas instrucciones son omitidas.

Una segunda estructura condicional muy recurrente es la conocida como `if ... else`. De forma similar a la anterior, comprueba un estado lógico y si es verdadero entonces se ejecutan las instrucciones que vienen después del `if`, mientras que si el estado es falso entonces se ejecutan las instrucciones que vienen después del `else`.

La tercera estructura condicional se conoce como `if ... elif`. De forma análoga a la estructura anterior, comprueba un estado lógico y si es verdadero entonces se ejecutan las instrucciones que vienen después del `if`, mientras que si el estado es falso entonces se evalúa otro estado lógico.

En Python debe escribirse dos puntos **:** después de la condición (el estado lógico que va a evaluarse) y las instrucciones que se deben ejecutar, si se cumple la condición, deben ir en las siguientes líneas con sangría. El código que no tenga sangría después de un condicional se entiende que es independiente de la estructura condicional.

```
# El texto despues de la almohadilla (texto en verde), como este,
    no se interpreta como instrucciones y sirve para insertar
    comentarios dentro del codigo.
# Comprobamos un estado logico dado por el operador relacional
    "<". Despues de la condicion evaluada debe ir ":".
if 2 < 3:
    # Si la condicion se cumple entonces enviamos un mensaje. Las
    instrucciones deben ir con sangria.
    print("2 es menor que 3")
# Cuando deja de haber sangria se entiende que se ha terminado la
    estructura condicional, y lo siguiente no depende de la misma.
print("Ya se ha ejecutado la estructura condicional")
```

A continuación se presentan una serie de ejercicios resueltos para aprender a utilizar los condicionales en Python.

**Ejemplo 6** (Condicional `if`)**.**

**20.** Comprueba si 187 es divisible por 11, en caso afirmativo imprime por pantalla con el comando `print()` un mensaje indicándolo.

```
if 187%11 == 0:
    print("187 es multiplo de 11")
```

**Ejemplo 7** (Condicional `if ... else`)**.**

**21.** Calcula $2^{10}$, si es menor que 1000 entonces súmale 1, en caso contrario réstale 10. Imprime el resultado final con el comando `print()`.

```
a = 2**10
if a < 1000:
    a = a + 1
else:
    a = a - 10
print(a)
```

**Ejemplo 8** (Condicional `if ... elif`)**.**

**22.** Comprueba si 391 es múltiplo de 7, si es así, emite un mensaje indicándolo; en caso contrario comprueba si es múltiplo de 13, si lo es, lanza un mensaje indicándolo. Si ninguna de las dos condiciones se cumple, emite indícalo con la función `print()`.

```
if 391%7 == 0:
    print("391 es multiplo de 7")
elif 391%13 == 0:
    print("391 es multiplo de 13")
else:
    print("391 no es multiplo de 7 ni de 13")
```

#### 3.6.2.3. Bucles en Python

Recordemos que un bucle es una estructura que permite ejecutar un mismo código de forma repetida hasta que se da una condición de parada. Cada una de esas ejecuciones se conoce como iteración. El código que se ejecuta en cada iteración se conoce como cuerpo del bucle.

En Python existen dos tipos de bucles, los bucles `for` y los bucles `while`. Los primeros tienen como parámetro una lista y una variable va recorriendo los diferentes elementos de la lista ejecutando el código una vez para cada uno de esos elementos. En cambio, los segundos tienen como parámetro una variable de tipo estado lógico (*bool*) y el cuerpo del bucle se mantiene en ejecución mientras la variable tenga un valor positivo (verdadero). Debe prestarse especial atención a este tipo de bucles ya que si el estado lógico no cambia en el cuerpo del bucle, entonces se entra en un bucle infinito, el programa se mantendrá en ejecución por tiempo indefinido.

Asimismo, cabe destacar que los bucles pueden anidarse. Esto es, en el cuerpo de un bucle puede haber otro nuevo bucle.

De forma similar a la estructura de los condicionales, en Python debe escribirse dos puntos `:` después de declarar el bucle. Seguidamente deben ir las instrucciones que desean ejecutarse en cada iteración, estas deben ir sangradas. El cuerpo del bucle termina cuando una línea de código deja de estar sangrada.

```
# En cada iteracion la variable x toma el valor de un elemento de
    lista.
for x in lista:
    # El codigo que se encuentre sangrado aqui se ejecutara en
    cada iteracion.
    print(x)
# Cuando deja de haber sangria se entiende que se ha terminado el
    cuerpo del bucle y esas instrucciones no se ejecutan hasta que
     el bucle ha terminado todas las iteraciones.
print("El bucle ha terminado")
```

A continuación se presentan una serie de ejercicios resueltos para aprender a utilizar los condicionales en Python.

**Ejemplo 9** (Bucle `for`)**.**

**23.** Imprime por pantalla con el comando `print()` los números pares del 1 al 9.

```
# Iniciamos una lista con los numeros enteros del 1 al 9.
candidatos = [1,2,3,4,5,6,7,8,9]
# En cada iteracion la variable i recorre los elementos de la
    lista candidatos.
for i in candidatos:
    # Se comprueba si la variable es multiplo de 2.
    if(i%2 == 0):
        # Si es multiplo de 2, se imprime por pantalla el valor de
     la variable.
        print(i)
```

**24.** Imprime por pantalla con el comando `print()` los resultados de multiplicar los elementos de la lista `[2,4,6]` y los de la lista `[3,5,7]` mediante todas las combinaciones posibles.

```
# Creamos las dos listas con sus respectivos elementos
lista_1 = [2,4,6]
lista_2 = [3,5,7]
# Recorremos los elementos de la primera lista con la variable i.
for i in lista_1:
```

```
    # Para cada elemento del primera lista, recorremos todos los elementos de la segunda lista con la variable j.
    for j in lista_2:
        # Imprimimos el resultado de multiplicar i por j.
        print(i*j)
```

**Ejemplo 10** (Bucle `while`)**.**

**25.** Imprime por pantalla con el comando `print()` los múltiplos de 6 que hay entre el 1 y el 30.

```
# Iniciamos una variable con el menor de los valores candidatos, en cada iteracion del bucle el valor de la variable ira incrementandose en una unidad.
a = 1
# En cada iteracion del bucle se comprueba la variable, si es menor que 30 entonces se ejecuta el cuerpo del bucle.
while(a <= 30):
    # Se comprueba si la variable es multiplo de 6.
    if(a%6 == 0):
        # Si es multiplo de 6, se imprime por pantalla el valor de la variable.
        print(a)
    # Incrementamos el valor de la variable en una unidad.
    a = a + 1
```

#### 3.6.2.4. Listas en Python

Las listas son estructuras que pueden almacenar simultáneamente valores de diferentes tipos (*bool*, *int*, *float*, *char*, *str*). Las listas en Python pueden asignarse a variables con el comando de asignación =. Las listas pueden declararse de formas muy distintas.

El método más simple de crear una lista es definiéndola elemento a elemento, para ello hay que poner sus elementos entre corchetes `[]` separados por comas. Si la lista declarada no tiene elementos se dice que está vacía y tiene 0 elementos.

```
# Aqui se esta asignando a la variable l_vacia una lista vacia.
l_vacia = []
# Aqui se esta asignando a la variable l_pares una lista con algunos numeros pares.
l_pares = [2,4,6,8]
```

Cada elemento de la lista se comporta como una variable, los elementos pueden ser de diferentes tipos. Para acceder a los elementos de una lista basta llamar a la lista y especificar el índice del elemento entre corchetes. Si lista declarada no tiene elementos se dice que está vacía y tiene 0 elementos. Nótese que en Python, el primer elemento de la lista está asociado al índice 0. Para modificar un elemento de la lista, basta acceder a él y asignarle con = el nuevo valor.

```
# Creamos una lista y la asignamos a la variable l.
l = ["perro", "gato", "pez", 8, -0.5, False, True, "Maria"]
# Accedemos al primer elemento de la lista y lo revelamos con el comando print().
print(l[0])
# Accedemos al tercer elemento de la lista y lo revelamos con el comando print().
print(l[2])
# Cambiamos el segundo elemento de la lista, "gato", por el numero 1001.
l[1] = 1001
# Imprimimos por pantalla la lista completa con el comando print().
print(l)
# Recorremos con un bucle los elementos de la lista y los imprimimos de uno en uno.
for elemento in l:
    print(elemento)
```

Podemos crear listas periódicas fácilmente.

```
# Creamos una lista que repite tres veces la lista [1,4,5].
l = [1,4,5]*3
# Creamos una lista con 101 ceros.
l_0 = [0]*101
# Imprimimos ambas listas
print(l)
print(l_0)
```

También es posible rellenar listas empleando bucles y condicionales. Este método es muy útil cuando se quiere crear una lista a partir de otra. Para añadir un elemento `x` a una lista `l` basta con llamar a la lista seguido de `.append(x)`.

```
# Declaramos una lista inicial.
l_inicial = [2,4,6,8,10,12,14,16,18,20]
# Declaramos una lista vacia que iremos rellenando con los multiplos de 3 de l_inicial.
l_filtrada = []
# Recorremos los elementos de la primera lista con un bucle for.
for x in l_inicial:
    # Dentro del bucle comprobamos si el elemento de la lista es multiplo de 3.
    if(x%3 == 0):
        # En caso afirmativo, guardamos ese elemento.
        l_filtrada.append(x)
```

```
# Cuando termina el bucle imprimimos la lista filtrada.
print(l_filtrada)
```

Un recurso muy útil para rellenar listas es el uso del bucle `for` junto con la función `range(inicio, fin)`. La función `range(inicio, fin)` crea una lista con números enteros desde `inicio` (incluyéndolo) hasta `fin` (sin incluirlo).

```
# Declaramos una lista vacia que vamos a ir llenando.
l = []
# Vamos a llenar la lista con los numeros del 13 al 131.
for i in range(13, 132):
    l.append(i)
# Imprimimos la lista.
print(l)
```

Otro recurso muy útil al trabajar con listas es la función `len()`, esta función admite como parámetro de entrada una lista y devuelve el número de elementos que contiene la misma.

```
# Creamos una lista y la asignamos a la variable l.
l = ["perro", "gato", "pez", 8, -0.5, False, True, "Maria"]
# Imprimimos la  longitud de la lista.
print(len(l))
```

A continuación se presentan una serie de ejercicios resueltos para aprender a utilizar las listas en Python.

**Ejemplo 11** (Listas)**.**

**26.** Crea una lista con los números del 1 al 100 y a partir de ella obtén los elementos que sean pares.

```
# Declaramos vacia la lista inicial.
l_100 = []
# Declaramos vacia la lista final.
l_fin = []
# Iniciamos una lista con los numeros enteros del 1 al 100.
for i in range(1,101):
    l_100.append(i)
# Con un bucle buscamos los elementos pares.
for n in l_100:
    # Se comprueba si la variable es multiplo de 2.
    if(n%2 == 0):
        # Si es multiplo de 2, se guarda el valor en la lista final.
        l_fin.append(n)
# Al terminar el bucle imprimimos la lista final por pantalla.
print(l_fin)
```

**27.** Crea una lista con los números del 1 al 10 y a partir de ella obtén los primeros 10 múltiplos de 13.

```
# Declaramos vacia la lista inicial.
l_10 = []
# Iniciamos una lista con los numeros enteros del 1 al 10.
for i in range(1,11):
    l_10.append(i)
# Creamos la lista final con tantos ceros como elementos tiene
   l_10.
l_fin = [0]*len(l_10)
# Ejecutamos un bucle con tantas iteraciones como elementos tiene
   la lista inicial.
for i in range(0, len(l_10)):
    # La lista l_fin ya esta creada, podemos acceder a sus
   elementos uno a uno y modificarlos.
    l_fin[i] = 13*l_10[i]
# Al terminar el bucle imprimimos la lista final por pantalla.
print(l_fin)
```

#### 3.6.2.5. Problemas a resolver en Python

A continuación se proponen varios problemas que pueden resolverse de forma sencilla mediante programación. En esta ocasión se ha escogido la temática de modelos matemáticos para los problemas. Esto se debe a que en su resolución se ven involucradas relaciones recursivas que pueden llevarse al código de un programa mediante bucles y condicionales.

**Problema 1** (Modelo de Malthus)**.** Supongamos que tenemos una población de plantas que se reproducen de manera anual produciendo tres nuevos ejemplares cada planta, y que las plantas no sobreviven de un año a otro.

Sabemos que inicialmente hay 30 plantas. ¿Cuántas plantas habrá pasados cuatro años? ¿En qué momento la población superará los 10000 ejemplares?

Solución propuesta para el número de plantas superados los cuatro años.

```
# Plantas iniciales.
plantas  = 30
# Numero de periodos totales.
periodos = 4
# En cada iteracion del bucle, cada periodo, el numero de plantas
   se multiplica.
for i in range(1, periodos + 1):
  # Cada planta da lugar a 3 nuevas plantas y no sobrevive al
   siguiente periodo.
```

```
  plantas = plantas * 3
# Imprimimos el numero de plantas que habra al final.
print(plantas)
```

Solución propuesta para el número de años que han de pasar hasta que la población supere los 10000 ejemplares.

```
# Plantas iniciales.
plantas  = 30
# Establecemos el limite de plantas.
limite   = 10000
# Declaramos un contador que ira sumando una unidad cada ejecucion
    del siguiente bucle.
contador = 0
# Definimos un bucle que se mantendra en ejecucion hasta el que el
    tamano de la poblacion supere el limite.
while(plantas < limite):
  # Cada planta da lugar a 3 nuevas plantas y no sobrevive al
    siguiente periodo.
  plantas = plantas * 3
  # Aumentamos en una unidad el contador de periodos.
  contador = contador + 1
# Imprimimos el numero de periodos que hay en el contador.
print(contador)
```

**Problema 2** (Modelo de Malthus)**.** La construcción de una fábrica de celulosa en la orilla de un río ha afectado a viabilidad de los huevos de una población de truchas, de tal manera que cada año la población desciende un 11 %. Sabemos que inicialmente había 1000 ejemplares. Supongamos que todos los años se repuebla el río introduciendo 100 ejemplares de la misma especie de trucha. ¿Qué le ocurrirá al tamaño de la población de truchas en el futuro?

Solución propuesta para mostrar una lista con el tamaño de la población a lo largo de 30 periodos (años).

```
# Truchas iniciales.
iniciales  = 1000
# Mortalidad anual.
mortalidad = 0.11
# Repuesto anual.
repuesto   = 100
# Periodos de observacion.
periodos   = 30
# Declaramos una lista donde almacenaremos el numero de truchas
   cada periodo.
truchas    = [0]*(periodos + 1)
# Fijamos el numero de truchas iniciales .
truchas[0] = iniciales
```

```
# Ejecutamos un bucle con tantas iteraciones como periodos de observacion.
for i in range(1, periodos + 1):
  # Cada periodo sobreviven (1-mortalidad)% de las truchas del periodo anterior y se anaden otras de repuesto.
  truchas[i] = truchas[i-1] * (1 - mortalidad) + repuesto
  # Redondeamos a un valor entero con la funcion int().
  truchas[i] = int(truchas[i])
# Imprimimos la lista con el numero de truchas cada periodo.
print(truchas)
```

**Problema 3** (Inversiones a interés compuesto)**.** Tus padres quieren que aprendas a ahorrar y te ofrecen un pacto. De forma semanal, además de darte una paga de 5€, te pagarán un 10 % de lo que haya en tu hucha. Si inicialmente tienes 7€, y decides ahorrar la paga de la semana durante 3 meses (12 semanas), ¿cuánto dinero tendrás ahorrado después de esos 3 meses? ¿Y si no te hubiesen ofrecido el pacto?

Solución propuesta para el dinero ahorrado con el pacto.

```
# Establecemos los ahorros iniciales.
ahorros = 7
# Fijamos la paga que se obtiene de forma semanal.
paga = 5
# Establecemos el interes.
interes = 0.1
# Fijamos el numero de periodos (semanas) de interes.
periodos = 12
# Ejecutamos un bucle con tantas iteraciones como periodos de interes.
for i in range(1, periodos + 1):
  # En cada periodo anadimos a la hucha un interes sobre lo que habia y la paga del periodo actual.
  ahorros = ahorros + ahorros*interes + paga
# Redondeamos a un valor entero con la funcion int().
ahorros = int(ahorros)
# Imprimimos el valor por pantalla.
print(ahorros)
```

Para obtener el dinero que se habría ahorrado sin el pacto basta con establecer `interes = 0` en el código anterior.

### 3.6.3. Proyecto final

El enunciado del proyecto que se plantea es sencillo, se pide elaborar un código en Python que permita calcular el mínimo común múltiplo y el máximo común divisor de dos números enteros, de hasta cuatro cifras. Para resolver el problema deben emplearse los conocimientos adquiridos en la sección 3.6.2. Comprobar el funcionamiento del código tomando como ejemplo los números $1464$ y $2684$.

Los alumnos disponen de 7 días desde el momento en que conocen el problema para realizar el proyecto. Se aconseja dividir el proyecto en diferentes etapas y seguir los procesos cognitivos de Munford (figura 2.1). En este sentido la figura del docente cobra un papel clave ya que debe guiar a los alumnos entre las diferentes etapas. Para ello el profesor debe plantear preguntas que estimulen a sus alumnos, como por ejemplo:

1. ¿Qué métodos conoces para calcular el mínimo común múltiplo y el máximo común divisor de dos números enteros?
2. ¿Qué es un número primo? ¿Cómo puedo saber si un número es primo?
3. ¿En qué afecta que los enteros tengan como máximo cuatro cifras?
4. ¿Cómo puedo enseñarle al ordenador cuáles los números primos?

De forma sintetizada, podemos dividir el proyecto en las siguientes partes:

1. Resolver el problema de forma manual.
2. Listar las instrucciones que necesita conocer el ordenador para resolver el problema.
3. Escribir el pseudocódigo de esas instrucciones. Esto es, un esquema donde se indiquen las estructuras básicas del código como el uso de determinados operadores, bucles, condicionales, etc.
4. Escribir cada una de las instrucciones en Google Colab. Evaluar su funcionamiento con diferentes ejemplos. Si no se obtiene el resultado esperado, plantearse si puede ser un error de código (alguna instrucción mal indicada o con el formato incorrecto) o si es un error de concepto (la instrucción está bien indicada pero no cumple el objetivo esperado).

5. Organizar las diferentes partes del código, ejecutarlas en su conjunto y comprobar los resultados.

Nótese que los pasos descritos en el párrafo anterior, donde deben realizarse comprobaciones, permiten retroceder al paso previo para volver a considerar las ideas que se tenían y plantear alternativas.

A continuación se propone una solución al problema. Nótese que esta solución no es única.

## Solución propuesta

Lo primero es pensar sobre el papel como operar para resolver el problema. Por ejemplo, una solución podría ser descomponer cada número en factores primos. ¿Qué necesitamos para ello? Saber cuáles qué números son primos. ¿Cuántos primos hacen falta? Como máximo son necesarios los primos de hasta 4 cifras.

**Listado de números primos**

¿Cómo saber si un número es primo? Probando que ninguno de los primos, obviamente menores que él, es divisor suyo. El siguiente código sirve para comprobar si el número 13 es primo.

```
# Definimos el numero con el que vamos a trabajar y los primos
    menores que este.
numero = 13
primos = [2,3,5,7,11]
# Inicialmente suponemos que es un numero primo.
es_primo = True
# Recorremos la lista de numeros primos con un bucle for.
for p in primos:
  # Comprobamos si algun primo divide al numero de interes.
  if numero%p == 0:
    # En caso afirmativo, el numero no puede ser primo.
    es_primo = False
# Imprimimos el estado logico resultante.
print(es_primo)
```

Para crear una lista con los números primos de hasta 4 cifras, habrá que recorrer los números del 1 al 9999 comprobando si son primos y añadiéndolos a dicha lista.

```
# Creamos una lista con los numeros primos menores que 10 y donde
    iremos guardando el resto de primos.
primos = [2, 3, 5, 7]
```

```
# Recorremos los numeros del 10 al 9999 con un bucle for.
for n in range(10, 10000):
  # De partida suponemos  que el numero en la variable n es primo.
  es_primo = True
  # Comprobamos si cada numero es primo diviendolo entre todos los primos conocidos menores que este.
  for p in primos:
    if(n%p == 0):
      es_primo = False
  # Si no tiene divisores primos, entonces ha de ser un nuevo primo, lo anadimos a la lista.
  if (es_primo):
    primos.append(n)
# Imprimimos la lista con los numeros primos encontrados.
print(primos)
```

#### Descomposición en factores primos

Una vez ya se tiene la lista con los números primos que hacen falta, podemos escribir un código que descomponga un número entero con menos de 5 cifras en factores primos.

```
# Concretamos los dos numeros enteros a estudiar.
a_original = 1464
b_original = 2684
# Iniciamos las variables con los numeros a estudiar.
a = a_original
b = b_original
# Creamos una lista vacia para cada uno de ellos donde almacenaremos sus factores primos.
factores_a = []
factores_b = []
# Tiene sentido buscar candidatos mientras que el numero sea mayor que uno.
while(a > 1):
  # Recorremos con un bucle los primos de interes.
   for p in primos:
     # Comprobamos si es divisor.
     if(a%p == 0):
       # Si es divisor, dividimos y guardamos el primo como factor.
       a = a/p
       factores_a.append(p)
# Imprimimos los factores primos del primer numero.
print(factores_a)
# Reciclamos el codigo para el segundo numero.
while(b > 1):
   for p in primos:
     if(b%p == 0):
       b = b/p
       factores_b.append(p)
print(factores_b)
```

### Cálculo del mínimo común múltiplo

Hay que juntar los factores primos de ambos números sin que aparezcan repetidos, y luego multiplicarlos.

```
# Creamos una lista vacia donde almacenaremos sus factores primos unicos.
factores_unicos = []
# Creamos una variable donde iremos multiplicando los factores unicos para formar el mcm.
mcm = 1
# Llenamos la lista con los factores del primer numero.
for x in factores_a:
  # Comprobamos si el elemento seleccionado todavia no esta en la lista.
  if not (x in factores_unicos):
    # Si todavia no esta, lo anadimos.
    factores_unicos.append(x)
# Llenamos la lista con los factores del segundo numero.
for x in factores_b:
  # Comprobamos si el elemento seleccionado todavia no esta en la lista
  if not (x in factores_unicos):
    # Si todavia no esta, lo anadimos.
    factores_unicos.append(x)
# Imprimimos por pantalla la lista de factores unicos.
print(factores_unicos)
# Recorremos los factores unicos y los vamos multiplicando al mcm.
for x in factores_unicos:
  mcm = mcm*x
# Imprimimos por pantalla el valor resultante.
print(mcm)
```

### Cálculo del máximo común divisor

Basta con obtener los factores comunes y multiplicarlos entre sí, teniendo en cuenta la multiplicidad de estos factores. Esto es, que pueden aparecer más de una vez en los factores primos de ambos números.

```
# Recuperamos los valores originales de los numeros a estudiar.
a = a_original
b = b_original
# Creamos una lista vacia donde almacenaremos los factores comunes unicos.
factores_comunes = []
# Creamos una variable donde iremos multiplicando los factores unicos para formar el mcm.
mcd = 1
# Recorremos todos los factores primos del primer y segundo numero con un bucle anidado.
for i in factores_a:
  for j in factores_b:
```

```
    # Si los elementos i y j coinciden, si es un factor comun, lo anadimos a la lista.
    if (i == j):
      # Comprobamos que este factor comun no estuviese ya en la lista.
      if not i in factores_comunes:
        # Si no lo estaba, lo anadimos.
        factores_comunes.append(i)
# Imprimimos por pantalla los factores comunes.
print(factores_comunes)
# Recorremos en un bucle los factores comunes para construir el mcd.
for x in factores_comunes:
  # Tenemos en cuenta la multiplicidad de los factores para obtener el MAXIMO comun divisor.
  # Cada factor se utilizara hasta que deje de dividir a uno de los dos terminos.
  while((a%x == 0) and (b%x == 0)):
    mcd = mcd*x
    # Dividimos a y b por el factor para actualizar la condicion del bucle while().
    a = a/x
    b = b/x
# Imprimimos por pantalla el valor resultante
print(mcd)
```

# Capítulo 4

# Experiencia en el Prácticum

El Prácticum forma parte del Máster Universitario de Profesor en Educación Secundaria Obligatoria y Bachillerato, Formación Profesional y Enseñanza de Idiomas. Consiste en un periodo de prácticas que tiene una duración de ocho semanas, durante este tiempo el estudiante ha de de asistir a un centro educativo que le habrá sido asignado previamente. Allí podrá vivir en primera persona el día a día de un profesor de educación secundaria obligatoria.

El periodo de prácticas se divide en dos fases, una primera fase de observación donde el alumno, acompañando a su tutor de prácticas, toma nota del transcurso habitual de las clases, de las metodologías que se siguen en el aula, y también aprende cómo lidiar con situaciones del día a día en la docencia.

La segunda fase del periodo de prácticas es la intervención, en este tiempo el alumno del máster será quien impartirá las clases. De forma previa a esta fase, el tutor indicará al alumno el curso académico en el que realizará la intervención, la Unidad didáctica que deberá impartir, y el número de sesiones que dispondrá. El estudiante del máster deberá organizarse debidamente y preparar el material necesario para el desarrollo de la Unidad didáctica y su evaluación.

El periodo de prácticas para el curso 2021-2022 tuvo lugar entre el 16 de febrero y el 6 de abril de 2022, el centro educativo donde realicé las prácticas es el IES JOSÉ JIMÉNEZ LOZANO, bajo la tutela de Beatriz Manso Castellot. Beatriz es profesora de Matemáticas en dos grupos de primero y en un grupo de cuarto de Educación Secundaria Obligatoria, también imparte clases en segundo de Bachillerato de Matemáticas Académicas y tutoriza un grupo de primero de la ESO.

En estas prácticas tuve la ocasión de poder realizar actividades alternativas con los alumnos de primero de la ESO. Una de ellas fue el *El ahorcado matemático* 3.5.3, la planteaba cuando tenía margen en alguna de las sesiones de clase. Aunque la actividad puede tener una apariencia simple, lo cierto es que tuvo una acogida muy buena. Los alumnos disfrutaron haciendo matemáticas, además practicaron cálculo mental y otros conceptos como la divisibilidad.

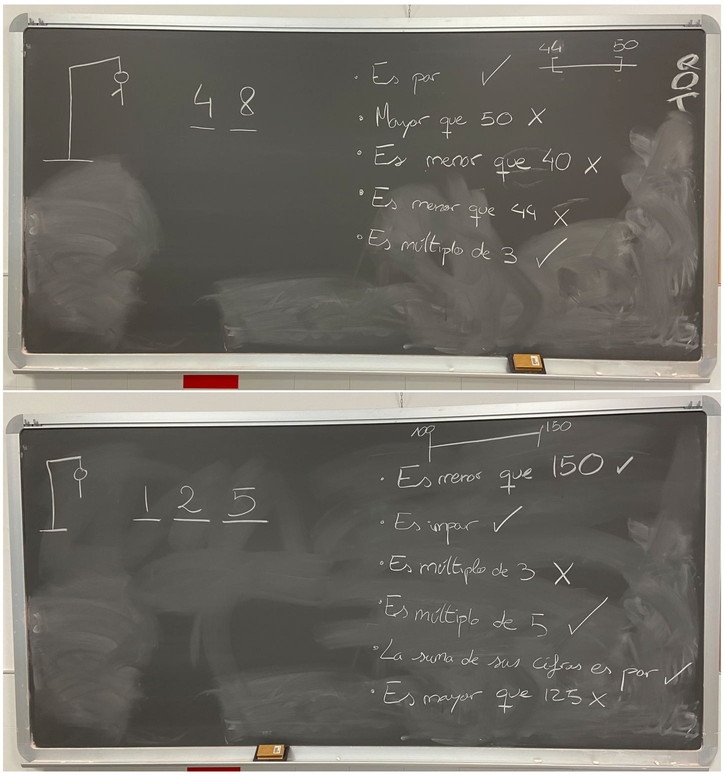
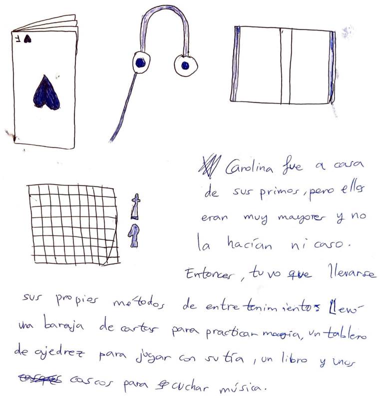
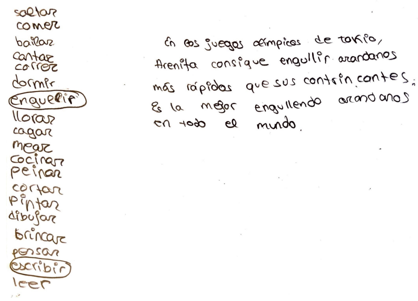



Figura 4.1: Experiencia en el Prácticum: *Ahorcado matemático*

Durante este período de prácticas, además de impartir la asignatura de matemáticas, tuve la oportunidad de dirigir una clase de tutoría con un grupo de primero de la ESO. De mutuo acuerdo con mi tutora, decidimos dedicar esta sesión a explicar qué es la creatividad y trabajarla mediante dos actividades. Escogí las actividades de *El dibujante* 3.4.1 y el *El puzle narrativo* 3.4.2 por ser actividades de fácil preparación, donde tienen que trabajar en equipo, y además se adecuaban al tiempo disponible. La realización de estas actividades nos permitió conocer más a los alumnos, aprendimos cuáles eran sus temas de interés y sus gustos.




(a) Actividad: *El dibujante*.




(b) Actividad: *Puzle narrativo*.

Figura 4.2: Experiencia en el Prácticum.

Para concluir, uno de los aprendizajes que me ha aportado realizar el Prácticum es que detrás de los alumnos hay personas, con sus gustos y aficiones, con muchas ganas de aprender, y con mucho potencial. Los docentes tienen en su mano decidir qué tipo de estudiantes quieren formar y cómo quieren hacerlo. Es cierto que el temario es extenso y el tiempo pasa rápido, pero los alumnos agradecen realizar otro tipo de actividades, diferentes a las repetitivas actividades de los libros de texto, y esto ofrece una oportunidad a los docentes para plantear nuevas propuestas como la que se ha presentado en este Trabajo de Fin de Máster.

# Bibliografía


Amabile, T. M. (1983). *The social psychology of creativity: A componential conceptualization.* Journal of personality and social psychology, 45(2), 357-376.

Baer, J. (2010). Is creativity domain specific?. In J. C. Kaufman R. J. Sternberg (Eds.), *The Cambridge handbook of creativity* (pp. 321–341). Cambridge University Press. `https://doi.org/10.1017/CBO9780511763205.021`

Baer, J. (2015). *The Importance of Domain-Specific Expertise in Creativity.* Roeper Review, 37(3), 165-178.

ORDEN EDU/362/2015, de 4 de mayo, por la que se establece el currículo y se regula la implantación, evaluación y desarrollo de la educación secundaria obligatoria en la Comunidad de Castilla y León. *Boletín Oficial de Castilla y León, 86*, de 8 de Mayo de 2015. `https://bocyl.jcyl.es/boletines/2015/05/08/pdf/BOCYL-D-08052015-4.pdf`

Csikszentmihalyi, M. (1990). The domain of creativity. In M. A. Runco R. S. Albert (Eds.). *Theories of Creativity* (pp. 190-212). Newbury Park, C. A: Sage.

Danielson C. (2016). *Which One Doesn't Belong? A Shapes Book.* Stenhouse.

Esquivias Serrano, M. T. (2004). *Creatividad: definiciones, antecedentes y aportaciones.* Revista Digital de la Universidad Nacional Autónoma de México, 5(1), 2-17. `https://ru.tic.unam.mx/handle/123456789/693`

Feldhusen, J. F., Treffinger, D. J., Bahlke, S. J. (1970). *Developing creative thinking: The Purdue creativity program.* The Journal of Creative Behavior, 4(2), 85-90.

Guildford, J.P. (1950). *Creativity.* American Psychologist, 5, 444-454.

Instituto Nacional de Evaluación Educativa. (s.f.). *PISA 2022* https://www.educacionyfp.gob.es/inee/evaluaciones-internacionales/pisa/pisa-2022.html

Isaksen, Scott De Schryver, Luc. (2000). *Making a difference with CPS: A summary of the evidence.* Facilitative Leadership: Making A Difference with Creative Problem Solving. 187-248.

Kienitz, E., Quintin, E., Saggar, M., Bott, N. T., Royalty, A., Hong, D. W., Liu, N., Chien, Y., Hawthorne, G., y Reiss, A. L. (2014) *Targeted intervention to increase creative capacity and performance: A randomized controlled pilot study.* Thinking Skills and Creativity, 13, 57-66.

Lubart, T. I. (2001). *Models of the creative process: Past, present and future.* Creativity research journal, 13(3-4), 295-308. https://doi.org/10.1207/S15326934CRJ1334_07

Maths Zone (s. f.). *Ooodle*. Recuperado 26 de junio de 2022, de https://mathszone.co.uk/resources/grid/ooodle/

Ruiz Melero, M. J. (2018). *Estudiar los perfiles creativos de los estudiantes en los distintos ámbitos escolares de la Educación Secundaria.* [Tesis de doctorado, Universidad de Murcia] https://dialnet.unirioja.es/servlet/tesis?codigo=155013

Mayseless, N., Saggar, M., Hawthorne, G., Reiss, A. (2018). Creativity in the Twenty-first Century: The Added Benefit of Training and Cooperation. In:

Plattner, H., Meinel, C., Leifer, L. (eds) *Design Thinking Research. Understanding Innovation.* Springer, Cham. `https://doi.org/10.1007/978-3-319-60967-6_12`

Mumford, M.D., Medeiros, K.E., Partlow, P.J. (2012). *Creative Thinking: Processes, Strategies, and Knowledge.* Journal of Creative Behavior, 46, 30-47.

Plucker, J., Beghetto, R. (2004). Why creativity is domain general, why it looks domain specific, and why the distinction does not matter. In R. J. Sternberg, E. L. Grigorenko J. L. Singer (Eds.), *Creativity: From potential to realization* (pp. 153–167). Washington, DC: American Psychological Association.

Santos, M. R. (1986). *Treinta y cinco años del pensamiento divergente: teoría de la creatividad de Guilford.* Estudios de psicología, 7(27-28), 175-192.

Sanz de Acedo Lizarraga, M. L., y Sanz de Acedo Baquedano, M. T. (2004). *La creatividad: un fenómeno cognitivo complejo con implicaciones educativas y empresariales* Huarte de San Juan. Psicología y pedagogía, (11), 65-85.

Sanz de Acedo Lizarraga, M. L., y Sanz de Acedo Baquedano, M. T. (2007). *Creatividad individual y grupal en la educación.* Madrid: Eiunsa.

Scott, G., Leritz, L. E., y Mumford, M. D. (2004). *The effectiveness of creativity training: A quantitative review.* Creativity research journal, 16(4), 361-388.

Tan, L.S., Lee, S.S., Ponnusamy, L.D., Koh, E.R., Tan, K.C.K. (2016). *Fostering Creativity in the Classroom for High Ability Students: Context Does Matter.* Education Sciences 6(4):36. `https://doi.org/10.3390/educsci6040036`

Torrance, E. P. (1962). *Building Creative Talent.* Englewood Cliffs, NJ: Prentice-Hall.

Torrance, E. P. (1965). *Scientific views of creativity and factors affecting its growth.* Daedalus, 94(3), 663-681.

Torrance, E. P., y Hansen, E. (1965). *The Question-Asking Behavior of Highly Creative and Less Creative Basic Business Teachers Identified by a Paper-and-Pencil Test.* Psychological Reports, 17(3), 815\OT1\textendash818. https://doi.org/10.2466/pr0.1965.17.3.815

Torrance, E. P. (1974). *The Torrance Tests of Creative Thinking - Norms- Technical Manual Research Edition - Verbal Tests, Forms A and B - Figural Tests, Forms A and B.* Princeton NJ: Personnel Press.

Torrance, E. P. (1976). Respetad sus derechos al "diablo". En J. Curtis, G. Demos, E. P. Torrance (Eds.), *Implicaciones educativas de la creatividad* (pp. 145-157). Ediciones Anaya.

Valencia L. [@MrValencia24]. (5 de junio de 2022). *I wholeheartedly agree with @Desmos guiding principle "We believe that students are creative, knowledgeable, and brilliant" Here is proof from* [Tweet], [Imagen adjunta]. Twitter. https://twitter.com/MrValencia24/status/1533466485695336451?s=20&t=w39_lr8z7KDWW6Wa61SbfQ

Wardle, John. (s. f.). *Wordle*. Recuperado 26 de junio de 2022, de https://wordle.danielfrg.com/

# Apéndice A

# Cógido python

## A.1. Ejemplos de operadores

### A.1.1. Operadores aritméticos

Sumar $545 + 65 + 6$.

[1]:
```
545 + 65 + 6
```

[1]: 616

Restar $5 - 7$.

[2]:
```
5 - 7
```

[2]: -2

Multiplicar $2 \cdot 2 \cdot 2$.

[3]:
```
2*2*2
```

[3]: 8

Dividir $115/7$.

[4]:
```
115/7
```

[4]: 16.428571428571427

Obtener el resto de dividir $115/7$.

[5]:
```
115%7
```

[5]: 3

Calcular $2^3$.

[6]:
```
2**3
```

[6]: 8

Sumar $2 + 9$, multiplicarlo por 3 y elevar el resultado a la quinta potencia.

[7]:
```
((2 + 9)*3)**5
```

[7]: `39135393`

Restar a 115 el resto de dividir 115/7 y dividir el resultado entre 7.

[8]:
```
(115 - (115%7))/7
```

[8]: `16.0`

### A.1.2. Operadores relacionales

Comprobar si $\frac{2}{5}$ es mayor que $\frac{3}{7}$.

[9]:
```
2/5 > 3/7
```

[9]: `False`

Comprobar si $2^{10}$ es menor que 1000.

[10]:
```
2**10 < 1000
```

[10]: `False`

Comprobar si $\frac{3}{7}$ es igual que $\frac{39}{91}$.

[11]:
```
3/7 == 39/91
```

[11]: `True`

Comprobar si 913 es múltiplo de 11.

[12]:
```
913//11 == 0
```

[12]: `False`

### A.1.3. Operadores de asignación

Guarda el resultado de multiplicar $81 \cdot 13$ en la variable $a$. Luego guarda en la variable $b$ el resultado de multiplicar $a$ por 10. Copia el valor de $b$ en la variable $a$. Muestra el valor de $a$ con la función *print()*.

[13]:
```
a = 81*13
b = a*10
a = b
print(a)
```

```
10530
```

Repite el ejemplo anterior sin recurrir a una segunda variable. Puedes sobrescribir el valor directamente sobre la variable $a$.

[14]:
```
a = 81*13
a = a*10
print(a)
```

```
10530
```

Utiliza el operador asignación para hallar la parte entera de la división 115/7. Para ello, guarda el resto de dividir 115/7. Réstaselo a 115, guarda el valor y divídelo entre 7. Muestra el resultado con la función *print()*, debería ser un número entero.

[15]:
```
a = 115%7
a = 115 - a
a = a/7
print(a)
```

```
16.0
```

### A.1.4. Operadores lógicos

Comprueba si 3 y 7 son divisores de 273. Puedes hacer uso del operador asignación para almacenar valores intermedios.

[16]:
```
a = 273%3
b = 273%7
(a == 0) and (b==0)
```

[16]:
```
True
```

Comprueba si 3 o 13 son divisores de 279. Puedes hacer uso del operador asignación para almacenar valores intermedios.

[17]:
```
a = 279%3
b = 279%13
(a == 0) or (b==0)
```

[17]:
```
True
```

Comprueba que 13 no es divisor de 279.

[18]:
```
not (279%13 == 0)
```

[18]:
```
True
```

### A.1.5. Operdador de pertenecia

Comprueba si 68 está en la lista de los 5 primeros múltiplos de 17.

[19]:
```
68 in [17, 2*17, 3*17, 4*17, 5*17]
```

[19]:
```
True
```

## A.2. Ejemplos de condicionales

### A.2.1. Condicional *if*

Comprueba si 187 es divisible por 11, en caso afirmativo imprime por pantalla con el comando print() un mensaje indicándolo.

[20]:
```
if 187%11 == 0:
    print("187 es multiplo de 11")
```

```
187 es multiplo de 11
```

### A.2.2. Condicional *if... else*

Calcula $2^{10}$, si es menor que 1000 entonces súmale 1, en caso contrario réstale 10. Imprime el resultado final con el comando *print()*.

[21]:
```
a = 2**10
if a < 1000:
    a = a + 1
else:
    a = a - 10
print(a)
```

```
1014
```

### A.2.3. Condicional *if... elif*

Comprueba si 391 es múltiplo de 7, si es así, emite un mensaje indicándolo; en caso contrario comprueba si es múltiplo de 13, si lo es, lanza un mensaje indicándolo. Si ninguna de las dos condiciones se cumple, emite indícalo con la función *print()*.

[22]:
```
if 391%7 == 0:
    print("391 es multiplo de 7")
elif 391%13 == 0:
    print("391 es multiplo de 13")
else:
    print("391 no es multiplo de 7 ni de 13")
```

```
391 no es multiplo de 7 ni de 13
```

## A.3. Ejemplos de bucles

### A.3.1. Bucle *for*

Imprime por pantalla con el comando *print()* los números pares del 1 al 9.

[23]:
```
# Iniciamos una lista con los numeros enteros del 1 al 9.
candidatos = [1,2,3,4,5,6,7,8,9]
# En cada iteracion la variable i recorre los elementos de la lista␣
 ↪candidatos.
for i in candidatos:
    # Se comprueba si la variable es multiplo de 2.
    if(i%2 == 0):
        # Si es multiplo de 2, se imprime por pantalla el valor de la␣
 ↪variable.
        print(i)
```

```
2
4
```

```
6
8
```

### A.3.2. Bucle *for* anidado

Imprime por pantalla con el comando *print()* los resultados de multiplicar los elementos de la lista [2, 4, 6] y los de la lista [3, 5, 7] mediante todas las combinaciones posibles.

[24]:
```
# Creamos las dos listas con sus respectivos elementos
lista_1 = [2,4,6]
lista_2 = [3,5,7]
# Recorremos los elementos de la primera lista con la variable i.
for i in lista_1:
    # Para cada elemento del primera lista, recorremos todos los elementos de␣
 ↪la segunda lista con la variable j.
    for j in lista_2:
        # Imprimimos el resultado de multiplicar i por j.
        print(i*j)
```

```
6
10
14
12
20
28
18
30
42
```

### A.3.3. Bucle *while*

Imprime por pantalla con el comando *print()* los múltiplos de 6 que hay entre el 1 y el 30.

[25]:
```
# Iniciamos una variable con el menor de los valores candidatos, en cada␣
 ↪iteracion del bucle el valor de la variable ira incrementandose en una␣
 ↪unidad.
a = 1
# En cada iteracion del bucle se comprueba la variable, si es menor que 30␣
 ↪entonces se ejecuta el cuerpo del bucle.
while(a <= 30):
    # Se comprueba si la variable es multiplo de 6.
    if(a%6 == 0):
        # Si es multiplo de 6, se imprime por pantalla el valor de la␣
 ↪variable.
        print(a)
    # Incrementamos el valor de la variable en una unidad.
    a = a + 1
```

```
6
12
18
24
30
```

## A.4. Ejemplos de listas

Para acceder a los elementos de una lista basta llamar a la lista y especificar el índice del elemento entre corchetes. Si lista declarada no tiene elementos se dice que está vacía y tiene 0 elementos. Nótese que en Python, el primer elemento de la lista está asociado al índice 0. Para modificar un elemento de la lista, basta acceder a él y asignarle con = el nuevo valor.

[26]:
```
# Creamos una lista y la asignamos a la variable l.
l = ["perro", "gato", "pez", 8, -0.5, False, True, "Maria"]
# Accedemos al primer elemento de la lista y lo revelamos con el comando␣
 ↪print().
print(l[0])
# Accedemos al tercer elemento de la lista y lo revelamos con el comando␣
 ↪print().
print(l[2])
# Cambiamos el segundo elemento de la lista, "gato", por el numero 1001.
l[1] = 1001
# Imprimimos por pantalla la lista completa con el comando print().
print(l)
# Recorremos con un bucle los elementos de la lista y los imprimimos de uno␣
 ↪en uno.
for elemento in l:
    print(elemento)
```

```
perro
pez
['perro', 1001, 'pez', 8, -0.5, False, True, 'Maria']
perro
1001
pez
8
-0.5
False
True
Maria
```

Podemos crear listas periódicas fácilmente.

[27]:
```
# Creamos una lista que repite tres veces la lista [1,4,5].
l = [1,4,5]*3
# Creamos una lista con 101 ceros.
l_0 = [0]*101
# Imprimimos ambas listas
print(l)
print(l_0)
```

```
[1, 4, 5, 1, 4, 5, 1, 4, 5]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

También es posible rellenar listas empleando bucles y condicionales. Este método es muy útil cuando se quiere crear una lista a partir de otra. Para añadir un elemento $x$ a una lista $l$ basta con llamar a la lista seguido de *.append(x)*.

[28]:
```
# Declaramos una lista inicial.
l_inicial = [2,4,6,8,10,12,14,16,18,20]
# Declaramos una lista vacia que iremos rellenando con los multiplos de 3 de␣
↪l_inicial.
l_filtrada = []
# Recorremos los elementos de la primera lista con un bucle for.
for x in l_inicial:
    # Dentro del bucle comprobamos si el elemento de la lista es multiplo de␣
↪3.
    if(x%3 == 0):
        # En caso afirmativo, guardamos ese elemento.
        l_filtrada.append(x)
# Cuando termina el bucle imprimimos la lista filtrada.
print(l_filtrada)
```

```
[6, 12, 18]
```

Un recurso muy útil para rellenar listas es el uso del bucle *for* junto con la función *range(inicio, fin)*. La función *range(inicio, fin)* crea una lista con números enteros desde *inicio* (incluyéndolo) hasta *fin* (sin incluirlo).

[29]:
```
# Declaramos una lista vacía que vamos a ir llenando.
l = []
# Vamos a llenar la lista con los números del 13 al 131.
for i in range(13, 132):
    l.append(i)
# Imprimimos la lista.
print(l)
```

```
[13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32,
33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52,
53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72,
73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92,
93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109,
110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125,
126, 127, 128, 129, 130, 131]
```

Otro recurso muy útil al trabajar con listas es la función *len()*, esta función admite como parámetro de entrada una lista y devuelve el número de elementos que contiene la misma.

[30]:
```
# Creamos una lista y la asignamos a la variable l.
l = ["perro", "gato", "pez", 8, -0.5, False, True, "Maria"]
# Imprimimos la  longitud de la lista.
print(len(l))
```

```
8
```

Crea una lista con los números del 1 al 100 y a partir de ella obtén los elementos que sean pares.

[31]:
```
# Declaramos vacia la lista inicial.
l_100 = []
```

```
# Declaramos vacia la lista final.
l_fin = []
# Iniciamos una lista con los numeros enteros del 1 al 100.
for i in range(1,101):
    l_100.append(i)
# Con un bucle buscamos los elementos pares.
for n in l_100:
    # Se comprueba si la variable es multiplo de 2.
    if(n%2 == 0):
        # Si es multiplo de 2, se guarda el valor en la lista final.
        l_fin.append(n)
# Al terminar el bucle imprimimos la lista final por pantalla.
print(l_fin)
```

```
[2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42,
44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66, 68, 70, 72, 74, 76, 78, 80, 82,
84, 86, 88, 90, 92, 94, 96, 98, 100]
```

Crea una lista con los números del 1 al 10 y a partir de ella obtén los primeros 10 múltiplos de 13.

[32]:
```
# Declaramos vacia la lista inicial.
l_10 = []
# Iniciamos una lista con los numeros enteros del 1 al 10.
for i in range(1,11):
    l_10.append(i)
# Creamos la lista final con tantos ceros como elementos tiene l_10.
l_fin = [0]*len(l_10)
# Ejecutamos un bucle con tantas iteraciones como elementos tiene la lista␣
 ↪inicial.
for i in range(0, len(l_10)):
    # La lista l_fin ya esta creada, podemos acceder a sus elementos uno a␣
 ↪uno y modificarlos.
    l_fin[i] = 13*l_10[i]
# Al terminar el bucle imprimimos la lista final por pantalla.
print(l_fin)
```

```
[13, 26, 39, 52, 65, 78, 91, 104, 117, 130]
```

## A.5. Problemas propuestos

### A.5.1. Problema 1 (Modelo Malthus)

Supongamos que tenemos una población de plantas que se reproducen de manera anual produciendo tres nuevos ejemplares cada planta, y que las plantas no sobreviven de un año a otro.

Sabemos que inicialmente hay 30 plantas. ¿Cuántas plantas habrá pasados cuatro años? ¿En qué momento la población superará los 10000 ejemplares?

Número de plantas después de 4 años:

[33]:
```python
# Plantas iniciales
plantas  = 30
# Número de años totales
periodos = 4
# En cada iteración del bucle, cada año, el número de plantas se multiplica.
for i in range(1, periodos + 1):
  # Cada planta da lugar a 3 nuevas plantas y no sobrevive al siguiente año.
  plantas = plantas * 3
# Imprimimos el número de plantas que habrá al final
print(plantas)
```

2430

Años necesarios para que la población supere las 10000 plantas:

[34]:
```python
# Plantas iniciales
plantas  = 30
# Establecemos el límte de plantas
limite   = 10000
# Declaramos un contador que irá sumando 1 unidad cada ejecución del␣
 ↪siguiente bucle
contador = 0
# Definimos un bucle que se mantendrá en ejecución hasta el que el tamaño de␣
 ↪la población supere el límite
while(plantas < limite):
  # Cada planta da lugar a 3 nuevas plantas y no sobrevive al siguiente año.
  plantas = plantas * 3
  # Aumentamos en una unidad el contador de años
  contador = contador + 1
# Imprimimos el número de años que hay en el contador
print(contador)
```

6

### A.5.2. Problema 2 (Modelo Malthus)

La construcción de una fábrica de celulosa en la orilla de un río ha afectado a viabilidad de los huevos de una población de truchas, de tal manera que cada año la población desciende un 11 %. Sabemos que inicialmente había 1000 ejemplares. Supongamos que todos los años se repuebla el río introduciendo 100 ejemplares de la misma especie de trucha. ¿Qué le ocurrirá al tamaño de la población de truchas en el futuro?

Mostramos una lista con el tamaño de la población a lo largo de 30 periodos (años).

[35]:
```python
# Truchas iniciales
iniciales   = 1000
# Mortalidad anual
mortalidad = 0.11
# Repuesto anual
repuesto   = 100
# Periodos de observación
periodos   = 30
# Declaramos una lista donde almacenaremos el número de truchas cada periodo
```

```
truchas     = [0]*(periodos + 1)
# Inicialmente hay
truchas[0] = iniciales
# Ejecutamos un bucle con tantas iteraciones como periodos de observación
for i in range(1, periodos + 1):
  # Cada periodo sobreviven (1-mortalidad)% de las truchas y se añaden otras␣
 ↪de repuesto
  truchas[i] = truchas[i-1] * (1 - mortalidad) + repuesto
  # Redondeamos a un valor entero con la función int()
  truchas[i] = int(truchas[i])
# Imprimimos la lista con el número de truchas cada periodo
print(truchas)
```

```
[1000, 990, 981, 973, 965, 958, 952, 947, 942, 938, 934, 931, 928, 925, 923,
921, 919, 917, 916, 915, 914, 913, 912, 911, 910, 909, 909, 909, 909, 909, 909]
```

### A.5.3. Problema 3 (Inversiones a interés compuesto)

Dinero ahorrado con el pacto.

[36]:
```
# Establecemos los ahorros iniciales
ahorros = 7
# Fijamos la paga que se obtiene de forma semanal
paga = 5
# Establecemos el interés
interes = 0.1
# Fijamos el número de periodos (semanas) de interés
periodos = 12
# # Ejecutamos un bucle con tantas iteraciones como periodos de interés
for i in range(1, periodos + 1):
  # En cada periodo añadimos a la hucha un interés sobre lo que había y la␣
 ↪paga del periodo actual
  ahorros = ahorros + ahorros*interes + paga
# Redondeamos a un valor entero con la función int()
ahorros = int(ahorros)
# Imprimimos el valor por pantalla
print(ahorros)
```

```
128
```

## A.6. Proyecto final

Crear un programa que calcule el mínimo común múltiplo y el máximo común divisor de dos números de hasta 4 cifras.

Lo primero es pensar sobre el papel como operar. Por ejemplo, una solución podría ser descomponer cada número en factores primos. ¿Qué necesitamos para ello? Saber cuáles qué números son primos. ¿Cuántos primos hacen falta? Como máximo son necesarios los primos de hasta 4 cifras.

**Listado de primos necesario** ¿Cómo saber si un número es primo? Probando que ninguno de los primos, obviamente menores que él, es divisor suyo.

[37]:
```python
numero = 13
primos = [2,3,5,7,11]

es_primo = True

for p in primos:
  if numero%p == 0:
    es_primo = False

print(es_primo)
```

```
True
```

Para crear una lista con los números primos de hasta 4 cifras, recorremos los números del 1 al 9999 comprobando si son primos y añadiéndolos a una lista.

[38]:
```python
# Creamos una lista con los números primos menores que 10 y donde iremos␣
↪guardando el resto de primos.
primos = [2, 3, 5, 7]
# Recorremos los números del 10 al 9999 con un bucle for.
for n in range(10, 10000):
  # De partida suponemos  que el número en la variable n es primo.
  es_primo = True
  # Comprobamos si cada número es primo diviéndolo entre todos los primos␣
↪conocidos menores que él.
  for p in primos:
    if(n%p == 0):
      es_primo = False
  # Si no tiene divisores primos, entonces ha de ser un nuevo primo, lo␣
↪añadimos a la lista.
  if (es_primo):
    primos.append(n)
# Imprimimos la lista con los números primos encontrados.
print(primos)
```

```
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73,
79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163,
167, 173, 179, 181, 191, 193, 197, 199, 211, 223, 227, 229, 233, 239, 241, 251,
257, 263, 269, 271, 277, 281, 283, 293, 307, 311, 313, 317, 331, 337, 347, 349,
353, 359, 367, 373, 379, 383, 389, 397, 401, 409, 419, 421, 431, 433, 439, 443,
449, 457, 461, 463, 467, 479, 487, 491, 499, 503, 509, 521, 523, 541, 547, 557,
563, 569, 571, 577, 587, 593, 599, 601, 607, 613, 617, 619, 631, 641, 643, 647,
653, 659, 661, 673, 677, 683, 691, 701, 709, 719, 727, 733, 739, 743, 751, 757,
761, 769, 773, 787, 797, 809, 811, 821, 823, 827, 829, 839, 853, 857, 859, 863,
877, 881, 883, 887, 907, 911, 919, 929, 937, 941, 947, 953, 967, 971, 977, 983,
991, 997, 1009, 1013, 1019, 1021, 1031, 1033, 1039, 1049, 1051, 1061, 1063,
1069, 1087, 1091, 1093, 1097, 1103, 1109, 1117, 1123, 1129, 1151, 1153, 1163,
1171, 1181, 1187, 1193, 1201, 1213, 1217, 1223, 1229, 1231, 1237, 1249, 1259,
1277, 1279, 1283, 1289, 1291, 1297, 1301, 1303, 1307, 1319, 1321, 1327, 1361,
1367, 1373, 1381, 1399, 1409, 1423, 1427, 1429, 1433, 1439, 1447, 1451, 1453,
1459, 1471, 1481, 1483, 1487, 1489, 1493, 1499, 1511, 1523, 1531, 1543, 1549,
1553, 1559, 1567, 1571, 1579, 1583, 1597, 1601, 1607, 1609, 1613, 1619, 1621,
```

```
1627, 1637, 1657, 1663, 1667, 1669, 1693, 1697, 1699, 1709, 1721, 1723, 1733,
1741, 1747, 1753, 1759, 1777, 1783, 1787, 1789, 1801, 1811, 1823, 1831, 1847,
1861, 1867, 1871, 1873, 1877, 1879, 1889, 1901, 1907, 1913, 1931, 1933, 1949,
1951, 1973, 1979, 1987, 1993, 1997, 1999, 2003, 2011, 2017, 2027, 2029, 2039,
2053, 2063, 2069, 2081, 2083, 2087, 2089, 2099, 2111, 2113, 2129, 2131, 2137,
2141, 2143, 2153, 2161, 2179, 2203, 2207, 2213, 2221, 2237, 2239, 2243, 2251,
2267, 2269, 2273, 2281, 2287, 2293, 2297, 2309, 2311, 2333, 2339, 2341, 2347,
2351, 2357, 2371, 2377, 2381, 2383, 2389, 2393, 2399, 2411, 2417, 2423, 2437,
2441, 2447, 2459, 2467, 2473, 2477, 2503, 2521, 2531, 2539, 2543, 2549, 2551,
2557, 2579, 2591, 2593, 2609, 2617, 2621, 2633, 2647, 2657, 2659, 2663, 2671,
2677, 2683, 2687, 2689, 2693, 2699, 2707, 2711, 2713, 2719, 2729, 2731, 2741,
2749, 2753, 2767, 2777, 2789, 2791, 2797, 2801, 2803, 2819, 2833, 2837, 2843,
2851, 2857, 2861, 2879, 2887, 2897, 2903, 2909, 2917, 2927, 2939, 2953, 2957,
2963, 2969, 2971, 2999, 3001, 3011, 3019, 3023, 3037, 3041, 3049, 3061, 3067,
3079, 3083, 3089, 3109, 3119, 3121, 3137, 3163, 3167, 3169, 3181, 3187, 3191,
3203, 3209, 3217, 3221, 3229, 3251, 3253, 3257, 3259, 3271, 3299, 3301, 3307,
3313, 3319, 3323, 3329, 3331, 3343, 3347, 3359, 3361, 3371, 3373, 3389, 3391,
3407, 3413, 3433, 3449, 3457, 3461, 3463, 3467, 3469, 3491, 3499, 3511, 3517,
3527, 3529, 3533, 3539, 3541, 3547, 3557, 3559, 3571, 3581, 3583, 3593, 3607,
3613, 3617, 3623, 3631, 3637, 3643, 3659, 3671, 3673, 3677, 3691, 3697, 3701,
3709, 3719, 3727, 3733, 3739, 3761, 3767, 3769, 3779, 3793, 3797, 3803, 3821,
3823, 3833, 3847, 3851, 3853, 3863, 3877, 3881, 3889, 3907, 3911, 3917, 3919,
3923, 3929, 3931, 3943, 3947, 3967, 3989, 4001, 4003, 4007, 4013, 4019, 4021,
4027, 4049, 4051, 4057, 4073, 4079, 4091, 4093, 4099, 4111, 4127, 4129, 4133,
4139, 4153, 4157, 4159, 4177, 4201, 4211, 4217, 4219, 4229, 4231, 4241, 4243,
4253, 4259, 4261, 4271, 4273, 4283, 4289, 4297, 4327, 4337, 4339, 4349, 4357,
4363, 4373, 4391, 4397, 4409, 4421, 4423, 4441, 4447, 4451, 4457, 4463, 4481,
4483, 4493, 4507, 4513, 4517, 4519, 4523, 4547, 4549, 4561, 4567, 4583, 4591,
4597, 4603, 4621, 4637, 4639, 4643, 4649, 4651, 4657, 4663, 4673, 4679, 4691,
4703, 4721, 4723, 4729, 4733, 4751, 4759, 4783, 4787, 4789, 4793, 4799, 4801,
4813, 4817, 4831, 4861, 4871, 4877, 4889, 4903, 4909, 4919, 4931, 4933, 4937,
4943, 4951, 4957, 4967, 4969, 4973, 4987, 4993, 4999, 5003, 5009, 5011, 5021,
5023, 5039, 5051, 5059, 5077, 5081, 5087, 5099, 5101, 5107, 5113, 5119, 5147,
5153, 5167, 5171, 5179, 5189, 5197, 5209, 5227, 5231, 5233, 5237, 5261, 5273,
5279, 5281, 5297, 5303, 5309, 5323, 5333, 5347, 5351, 5381, 5387, 5393, 5399,
5407, 5413, 5417, 5419, 5431, 5437, 5441, 5443, 5449, 5471, 5477, 5479, 5483,
5501, 5503, 5507, 5519, 5521, 5527, 5531, 5557, 5563, 5569, 5573, 5581, 5591,
5623, 5639, 5641, 5647, 5651, 5653, 5657, 5659, 5669, 5683, 5689, 5693, 5701,
5711, 5717, 5737, 5741, 5743, 5749, 5779, 5783, 5791, 5801, 5807, 5813, 5821,
5827, 5839, 5843, 5849, 5851, 5857, 5861, 5867, 5869, 5879, 5881, 5897, 5903,
5923, 5927, 5939, 5953, 5981, 5987, 6007, 6011, 6029, 6037, 6043, 6047, 6053,
6067, 6073, 6079, 6089, 6091, 6101, 6113, 6121, 6131, 6133, 6143, 6151, 6163,
6173, 6197, 6199, 6203, 6211, 6217, 6221, 6229, 6247, 6257, 6263, 6269, 6271,
6277, 6287, 6299, 6301, 6311, 6317, 6323, 6329, 6337, 6343, 6353, 6359, 6361,
6367, 6373, 6379, 6389, 6397, 6421, 6427, 6449, 6451, 6469, 6473, 6481, 6491,
6521, 6529, 6547, 6551, 6553, 6563, 6569, 6571, 6577, 6581, 6599, 6607, 6619,
6637, 6653, 6659, 6661, 6673, 6679, 6689, 6691, 6701, 6703, 6709, 6719, 6733,
6737, 6761, 6763, 6779, 6781, 6791, 6793, 6803, 6823, 6827, 6829, 6833, 6841,
6857, 6863, 6869, 6871, 6883, 6899, 6907, 6911, 6917, 6947, 6949, 6959, 6961,
6967, 6971, 6977, 6983, 6991, 6997, 7001, 7013, 7019, 7027, 7039, 7043, 7057,
7069, 7079, 7103, 7109, 7121, 7127, 7129, 7151, 7159, 7177, 7187, 7193, 7207,
```

```
7211, 7213, 7219, 7229, 7237, 7243, 7247, 7253, 7283, 7297, 7307, 7309, 7321,
7331, 7333, 7349, 7351, 7369, 7393, 7411, 7417, 7433, 7451, 7457, 7459, 7477,
7481, 7487, 7489, 7499, 7507, 7517, 7523, 7529, 7537, 7541, 7547, 7549, 7559,
7561, 7573, 7577, 7583, 7589, 7591, 7603, 7607, 7621, 7639, 7643, 7649, 7669,
7673, 7681, 7687, 7691, 7699, 7703, 7717, 7723, 7727, 7741, 7753, 7757, 7759,
7789, 7793, 7817, 7823, 7829, 7841, 7853, 7867, 7873, 7877, 7879, 7883, 7901,
7907, 7919, 7927, 7933, 7937, 7949, 7951, 7963, 7993, 8009, 8011, 8017, 8039,
8053, 8059, 8069, 8081, 8087, 8089, 8093, 8101, 8111, 8117, 8123, 8147, 8161,
8167, 8171, 8179, 8191, 8209, 8219, 8221, 8231, 8233, 8237, 8243, 8263, 8269,
8273, 8287, 8291, 8293, 8297, 8311, 8317, 8329, 8353, 8363, 8369, 8377, 8387,
8389, 8419, 8423, 8429, 8431, 8443, 8447, 8461, 8467, 8501, 8513, 8521, 8527,
8537, 8539, 8543, 8563, 8573, 8581, 8597, 8599, 8609, 8623, 8627, 8629, 8641,
8647, 8663, 8669, 8677, 8681, 8689, 8693, 8699, 8707, 8713, 8719, 8731, 8737,
8741, 8747, 8753, 8761, 8779, 8783, 8803, 8807, 8819, 8821, 8831, 8837, 8839,
8849, 8861, 8863, 8867, 8887, 8893, 8923, 8929, 8933, 8941, 8951, 8963, 8969,
8971, 8999, 9001, 9007, 9011, 9013, 9029, 9041, 9043, 9049, 9059, 9067, 9091,
9103, 9109, 9127, 9133, 9137, 9151, 9157, 9161, 9173, 9181, 9187, 9199, 9203,
9209, 9221, 9227, 9239, 9241, 9257, 9277, 9281, 9283, 9293, 9311, 9319, 9323,
9337, 9341, 9343, 9349, 9371, 9377, 9391, 9397, 9403, 9413, 9419, 9421, 9431,
9433, 9437, 9439, 9461, 9463, 9467, 9473, 9479, 9491, 9497, 9511, 9521, 9533,
9539, 9547, 9551, 9587, 9601, 9613, 9619, 9623, 9629, 9631, 9643, 9649, 9661,
9677, 9679, 9689, 9697, 9719, 9721, 9733, 9739, 9743, 9749, 9767, 9769, 9781,
9787, 9791, 9803, 9811, 9817, 9829, 9833, 9839, 9851, 9857, 9859, 9871, 9883,
9887, 9901, 9907, 9923, 9929, 9931, 9941, 9949, 9967, 9973]
```

**Descomposición en factores primos** Una vez ya tenemos la lista con los números primos que nos hacen falta, podemos escribir un código que descomponga un número entero con menos de 5 cifras en factores primos.

[39]:
```python
# Concretamos los dos números enteros a estudiar.
a_original = 1464
b_original = 2684
```

[40]:
```python
# Iniciamos las variables con los números a estudiar.
a = a_original
b = b_original
# Creamos una lista vacía para cada uno de ellos donde almacenaremos sus␣
 ↪factores primos.
factores_a = []
factores_b = []
# Tiene sentido buscar candidatos mientras que el número sea mayor que uno.
while(a > 1):
  # Recorremos con un bucle los primos de interés.
   for p in primos:
     # Comprobamos si es divisor.
     if(a%p == 0):
       # Si es divisor, dividimos y guardamos el primo como factor.
       a = a/p
       factores_a.append(p)
# Imprimimos los factores primos del primer número
print(factores_a)
```

```
# Reciclamos el código para el segundo número
while(b > 1):
   for p in primos:
     if(b%p == 0):
       b = b/p
       factores_b.append(p)
print(factores_b)
```

```
[2, 3, 61, 2, 2]
[2, 11, 61, 2]
```

**Cálculo de mcm** Debemos juntar los factores primos de ambos números sin que aparezcan repetidos, y luego multiplicarlos.

[41]:
```
# Creamos una lista vacía donde almacenaremos sus factores primos únicos.
factores_unicos = []
# Creamos una variable donde iremos multiplicando los factores unicos para␣
 ↪formar el mcm.
mcm = 1
# Llenamos la lista con los factores del primer número.
for x in factores_a:
  # Comprobamos si el elemento seleccionado todavía no está en la lista
  if not (x in factores_unicos):
    # Si todavía no está, lo añadimos
    factores_unicos.append(x)
# Llenamos la lista con los factores del segundo número.
for x in factores_b:
  # Comprobamos si el elemento seleccionado todavía no está en la lista
  if not (x in factores_unicos):
    # Si todavía no está, lo añadimos
    factores_unicos.append(x)
# Imprimimos por pantalla la lista de factores únicos
print(factores_unicos)
# Recorremos los factores únicos y los vamos multiplicando al mcm
for x in factores_unicos:
  mcm = mcm*x
# Imprimimos por pantalla el valor resultante
print(mcm)
```

```
[2, 3, 61, 11]
4026
```

**Cálculo de mcd** Debemos quedarnos con los factores comunes y multiplicarlos entre sí, teniendo en cuenta la multiplicidad de estos factores. Esto es, que pueden aparecer más de una vez en los factores primos de ambos números.

[42]:
```
# Recuperamos los valores originales de los números a estudiar.
a = a_original
b = b_original
# Creamos una lista vacía donde almacenaremos los factores comunes únicos.
factores_comunes = []
```

```
# Creamos una variable donde iremos multiplicando los factores unicos para␣
↪formar el mcm.
mcd = 1
# Recorremos todos los factores primos del primer y segundo número con un␣
↪bucle anidado.
for i in factores_a:
  for j in factores_b:
    # Si los elementos de i y j coinciden, si es un factor común, lo añadimos␣
↪a la lista.
    if (i == j):
      # Comprobamos que este factor común no estuviese ya en la lista.
      if not i in factores_comunes:
        # Si no lo estaba, lo añadimos
        factores_comunes.append(i)
# Imprimimos por pantalla los factores comunes.
print(factores_comunes)
# Recorremos en un bucle los factores comunes para construir el mcd.
for x in factores_comunes:
  # Tenemos en cuenta la multiplicidad de los factores para obtener el MÁXIMO␣
↪común divisor.
  # Cada factor se utilizará hasta que deje de dividir a uno de los dos␣
↪términos.
  while((a%x == 0) and (b%x == 0)):
    mcd = mcd*x
    # Dividimos a y b por el factor para actualizar la condición del bucle␣
↪while().
    a = a/x
    b = b/x
# Imprimimos por pantalla el valor resultante
print(mcd)
```

```
[2, 61]
244
```

**Ejemplo de idea no válida aunque lo parecía inicialmente** El siguiente código es bueno para detectar factores comunes y no comunes, pero **no es capaz de tener en cuenta la multiplicidad de los factores** (las veces que aparecen).

[43]:
```
# Concretamos los dos números enteros a estudiar.
a = a_original
b = b_original
# Creamos una lista donde almacenaremos los factores comunes y los no comunes.
comunes    = []
no_comunes = []
# Recorremos con un bucle los primos de interés.
for p in primos:
  # Utilizamos el operador and para comprobar si el primo es divisor común a␣
↪ambos.
  if((a%p == 0) and (b%p == 0)):
```

```
        # Si es común a ambos, realizamos la división y guardamos el primo como␣
↪factor común.
        a = a/p
        b = b/p
        comunes.append(p)
    # Si no es divisor común, comprobamos si es dividor del primero.
    elif(a%p == 0):
        # Si es divisor del primero realizamos la división y guardamos el primo␣
↪como factor no común.
        a = a/p
        no_comunes.append(p)
    # Si no es divisor común, ni divisor del primero, comprobamos si lo es del␣
↪segundo
    elif(b%p == 0):
        # Si es divisor del segundo realizamos la división y guardamos el primo␣
↪como factor no común.
        b = b/p
        no_comunes.append(p)
# Imprimimos los factores comunes
print(comunes)
# Imprimimos los factores no comunes
print(no_comunes)
```

```
[2, 61]
[3, 11]
```