Algoritmo para hacer panqueques

1. Buscar los utensilios de cocina (bowl, tenedor, sartén y espátula)

2. Buscar los ingredientes de la receta (mantequilla, huevo, 30 g de mezcla para panqueques y 20 ml de agua)

3. Depositar todos los ingredientes en el bowl y comenzar a batir con el tenedor

4. Poner la estufa a fuego bajo y agregar un poco de mantequilla a la sartén

5. Una vez tengamos una mezcla homogénea la agregamos a la sartén y tapamos

6. Cuando el panqueque empiece a burbujear por encima damos vuelta y esperamos 1 minuto

7. Apagamos la estufa y servimos el panqueque

8. Acompañamos con nuestros toppings favoritos (miel, fruta, chocolate)

9. Disfrutar

Glosario de términos

Lenguaje de programación:

Un lenguaje de programación, en palabras simples, es el conjunto de instrucciones a través del cual los humanos interactúan con las computadoras. Un lenguaje de programación nos permite comunicarnos con las computadoras a través de algoritmos e instrucciones escritas en una sintaxis que la computadora entiende e interpreta en lenguaje de máquina.

Fuente: <https://openwebinars.net/blog/que-es-un-lenguaje-de-programacion/>

Algoritmo:

En [informática](https://concepto.de/informatica/), un algoritmo es una secuencia de instrucciones secuenciales, gracias al cual pueden llevarse a cabo ciertos [procesos](https://concepto.de/proceso/) y darse respuesta a determinadas necesidades o decisiones. Se trata de conjuntos ordenados y finitos de pasos, que nos permiten resolver un [problema](https://concepto.de/problema/) o [tomar una decisión](https://concepto.de/toma-de-decisiones/).

Fuente: <https://concepto.de/algoritmo-en-informatica/#ixzz89G9LA6sC>

Lenguaje de máquina:

También denominado código máquina. Es un conjunto de instrucciones entendibles directamente por el ordenador, puesto que se componen de unos y ceros. Generalmente, el programador utiliza un lenguaje de programación basado en el [lenguaje natural](https://www.glosarioit.com/Lenguaje_natural), y éste es traducido a código máquina posteriormente. Se trata del lenguaje de programación del más bajo nivel, es decir, se comunica con la computadora directamente a través del [código binario](https://www.glosarioit.com/C%C3%B3digo_binario).

Fuente: <https://www.glosarioit.com/Lenguaje_de_m%C3%A1quina>

Lenguaje de bajo nivel (ensamblador):

Lenguaje de programación que utiliza palabras clave e instrucciones que son un poco más complejas que los unos y ceros del lenguaje máquina. En sentido técnico, el lenguaje de bajo nivel incluye el lenguaje máquina, pero se refiere más comúnmente a un lenguaje ensamblador que emplea símbolos para crear instrucciones de máquina más fáciles de leer y entender por parte de los programadores.

Fuente:<https://support.esri.com/es-es/gis-dictionary/low-level-language#:~:text=En%20sentido%20t%C3%A9cnico%2C%20el%20lenguaje,por%20parte%20de%20los%20programadores>.

Lenguaje de alto nivel:

El Lenguaje de alto nivel es aquel que se aproxima más al lenguaje natural humano que al lenguaje [binario](https://www.ecured.cu/Sistema_binario) de las computadoras, el que se conoce como lenguaje de bajo nivel. Un lenguaje de alto nivel permite al programador escribir las instrucciones de un programa utilizando palabras o expresiones sintácticas muy similares al inglés. Por ejemplo, en [C](https://www.ecured.cu/C) se pueden usar palabras tales como: case, if, for, while, etc.

Fuente: <https://www.ecured.cu/Lenguaje_de_alto_nivel>

Algoritmos cualitativos:

Los algoritmos cualitativos son aquellos que se realizan por medio de las palabras, lo que quiere decir que las órdenes vienen dadas en forma verbal.

Fuente: <https://conogasi.org/articulos/algoritmo/>

Algoritmos cuantitativos:

Los algoritmos cuantitativos son aquellos algoritmos que utilizan operaciones algebraicas y cálculos numéricos específicos para definir un proceso, obteniendo valores concretos.

Fuente: <https://www.lifeder.com/algoritmos-cuantitativos/>

Lenguajes algorítmicos gráficos:

Es la representación gráfica de las operaciones que realiza un algoritmo también llamado diagrama de flujo. El diagrama de flujo se caracteriza por la forma detallada como representa de forma gráfica los pasos a seguir para encontrar la solución del problema propuesto.

Fuente:<https://sites.google.com/a/misena.edu.co/wiki-adsi/home/lenguajes-algoritmicos-graficos>

Lenguajes no gráficos:

Representa de forma descriptiva las operativas que debe realizar un algoritmo (pseudocódigo).

Fuente:<https://sites.google.com/a/misena.edu.co/wiki-adsi/home/lenguajes-no-graficos>

Pseudocódigo:

El pseudocódigo es una forma de expresar los distintos pasos que va a realizar un programa, de la forma más parecida a un lenguaje de programación. Su principal función es la de representar por pasos la solución a un problema o algoritmo, de la forma más detallada posible, utilizando un lenguaje cercano al de programación.

Fuente: <https://openwebinars.net/blog/que-es-pseudocodigo/>

Definición del problema:

Es el primer paso en un proyecto de programación, consiste en preparar, en la terminología del cliente, un enunciado breve del problema que se solucionará y de las restricciones que existen en su resolución. El enunciado definitivo del problema debe incluir una descripción de la situación actual y de las metas que debe lograr el nuevo sistema.

Fuente: <https://prezi.com/w63pfo98mkhj/programacion-descripcion-del-problema/>

Análisis del problema:

consiste en el estudio detallado del problema. Se debe identificar los datos de entrada, de salida y la descripción del problema.

Fuente:<https://plataforma.josedomingo.org/pledin/cursos/curso_cpp1/curso/u01/#:~:text=Definici%C3%B3n%20o%20an%C3%A1lisis%20del%20problema,de%20un%20problema%20dado%3A%20algoritmo>.

Diseño del algoritmo:

Describe la secuencia ordenada de pasos que conduce a la solución de un problema dado: algoritmo.

Fuente:<https://plataforma.josedomingo.org/pledin/cursos/curso_cpp1/curso/u01/#:~:text=Definici%C3%B3n%20o%20an%C3%A1lisis%20del%20problema,de%20un%20problema%20dado%3A%20algoritmo>.

Codificación:

La codificación es el proceso de usar lenguajes de programación para dar instrucciones a una computadora. Estas instrucciones impulsan los sitios web, el software y las aplicaciones que la gente usa todos los días.

Fuente:<https://blog.hubspot.es/website/como-empezar-a-programar#:~:text=los%20conceptos%20b%C3%A1sicos.-,%C2%BFQu%C3%A9%20es%20codificar%3F,gente%20usa%20todos%20los%20d%C3%ADas>.

Compilación y ejecución:

Compilar es el proceso de transformar un programa informático escrito en un [lenguaje](https://developer.mozilla.org/es/docs/Glossary/Computer_Programming) en un conjunto de instrucciones en otro formato o lenguaje. Un compilador es un programa de computadora que realiza dicha tarea.

La ejecución en ingeniería informática y de software es el proceso mediante el cual una computadora o máquina virtual lee y actúa según las instrucciones de un programa informático. Cada instrucción de un programa es una descripción de una acción particular que debe llevarse a cabo para resolver un problema específico.

Fuentes: https://developer.mozilla.org/es/docs/Glossary/Compile

<https://academia-lab.com/enciclopedia/ejecucion-informatica/#:~:text=La%20ejecuci%C3%B3n%20en%20ingenier%C3%ADa%20inform%C3%A1tica,para%20resolver%20un%20problema%20espec%C3%ADfico>.

Prueba y depuración:

La depuración es el proceso de encontrar y solucionar errores en el código fuente de cualquier software. Cuando un software no funciona tal y como se espera, los programadores de computadoras estudian el código para determinar por qué ocurren algunos errores. Con este objetivo, los programadores utilizan herramientas de depuración para ejecutar el software en un entorno controlado, comprobar el código paso a paso y analizar y solucionar el error.

La depuración y las [pruebas](https://aws.amazon.com/blogs/devops/tag/testing/) son procesos complementarios que garantizan que los programas de software funcionen como deben. Después de escribir una sección completa o parte de un código, los programadores realizan pruebas para identificar fallos y errores.

Fuente:<https://aws.amazon.com/es/what-is/debugging/#:~:text=La%20depuraci%C3%B3n%20y%20las%20pruebas%20son%20procesos%20complementarios%20que%20garantizan,para%20identificar%20fallos%20y%20errores>.

Documentación:

La documentación de software se define como la información enfocada en la descripción del sistema o producto para quienes se encargan de desarrollarlo, implementarlo y utilizarlo. Por lo que también puede determinarse como una documentación de procesos. Además, esta documentación incorpora aspectos de los manuales, como las funciones de ayuda, sus versiones en línea y demás.

Fuente:<https://keepcoding.io/blog/que-es-la-documentacion-de-software/#:~:text=La%20documentaci%C3%B3n%20de%20software%20se,como%20una%20documentacion%20de%20procesos>.

Tipos de datos:

En programación un tipo de dato es la clasificación de un dato según sus características. Puede tratarse de una cadena, de un texto, de números, etc., y un valor es cualquier clase de dato que se halle dentro de un tipo de dato.

Fuentes:<https://edu.gcfglobal.org/es/conceptos-basicos-de-programacion/valor-y-tipos-de-datos/1/>

Expresiones:

Las expresiones son combinaciones de constantes, variables y operadores de diferentes tipos que nos permiten manipular datos para obtener información nueva. Los lenguajes de programación usan los mismos conceptos de las operaciones matemáticas tradicionales, tales como el agrupamiento y la precedencia de operadores.

Fuente:<https://es.wikiversity.org/wiki/Fundamentos_de_programaci%C3%B3n/Expresiones#:~:text=Las%20expresiones%20son%20combinaciones%20de,y%20la%20precedencia%20de%20operadores>.

Operadores y operandos:

Los operadores son elementos que relacionan de forma diferente, los valores con los que trabajamos en los lenguajes de programación. En los lenguajes de programación usamos los operadores para manipular los valores y transformarlos, con el objetivo de realizar los objetivos de los programas.

Los operandos son los valores que se utilizan para alimentar los operadores. Por ejemplo, en el operador suma necesitamos dos operandos para sumar ambos valores.

Fuentes: <https://desarrolloweb.com/articulos/operadores-operandos-programacion>

Identificadores (variables y constantes):

Un identificador es un conjunto de caracteres alfanuméricos de cualquier longitud que sirve para identificar las entidades del programa (clases, funciones, variables, tipos compuestos) Los identificadores pueden ser combinaciones de letras y números. Cada lenguaje tiene sus propias reglas que definen como pueden estar construidos.

Un identificador es un conjunto de caracteres alfanuméricos de cualquier longitud que sirve para identificar las entidades del programa (clases, funciones, variables, tipos compuestos) Los identificadores pueden ser combinaciones de letras y números. Cada lenguaje tiene sus propias reglas que definen como pueden estar construidos.

Fuentes:<https://andresmtzg.wordpress.com/2012/09/27/identificadores-variables-y-constantes/>

Dispositivos de entrada:

En [informática](https://concepto.de/informatica/), se conoce como dispositivos de entrada o dispositivos de alimentación (input) a los aparatos que permiten ingresar información al sistema informático, ya sea proveniente del usuario o de otra [computadora](https://concepto.de/computadora/). Por ejemplo: el [teclado](https://concepto.de/teclado-informatica/), el [mouse](https://concepto.de/mouse/) o el micrófono.

Fuente: <https://concepto.de/dispositivos-de-entrada/#ixzz89HD0cwnF>

Dispositivos de salida:

En [informática](https://concepto.de/informatica/), se conoce como dispositivos de entrada o dispositivos de alimentación (input) a los aparatos que permiten ingresar información al sistema informático, ya sea proveniente del usuario o de otra [computadora](https://concepto.de/computadora/). Por ejemplo: el [teclado](https://concepto.de/teclado-informatica/), el [mouse](https://concepto.de/mouse/) o el micrófono.

Fuente: <https://concepto.de/dispositivos-de-entrada/#ixzz89HD0cwnF>

Unidad central de procesamiento (c.p.u):

La unidad central de procesamiento o CPU (por el acrónimo en inglés de central processing unit), o simplemente el procesador o microprocesador, es el componente del computador y otros dispositivos programables, que interpreta las instrucciones contenidas en los programas y procesa los datos. Los CPU proporcionan la característica fundamental de la computadora digital (la programabilidad) y son uno de los componentes necesarios encontrados en las computadoras de cualquier tiempo, junto con el almacenamiento primario y los dispositivos de entrada/salida. Se conoce como microprocesador el CPU que es manufacturado con circuitos integrados. Desde mediados de los años 1970, los microprocesadores de un solo chip han reemplazado casi totalmente todos los tipos de CPU, y hoy en día, el término "CPU" es aplicado usualmente a todos los microprocesadores.

Fuente: <https://www.lawebdelprogramador.com/diccionario/CPU/>

Unidad de control:

Es el "cerebro del microprocesador". Es la encargada de activar o desactivar los diversos componentes del microprocesador en función de la instrucción que el microprocesador esté ejecutando y en función también de la etapa de dicha instrucción que se esté ejecutando.

La unidad de control (UC) interpreta y ejecuta las instrucciones almacenadas en la memoria principal y genera las señales de control necesarias para ejecutarlas.

Fuente: <https://www.lawebdelprogramador.com/diccionario/Unidad-de-Control/>

Unidad Aritmético-Lógica:

La Unidad aritmético-lógica (ALU por sus siglas en inglés: Arithmetic Logic Unit) es un contador digital capaz de realizar las operaciones aritméticas y lógicas entre los datos de un circuito; suma, resta, multiplica y divide, así como establece comparaciones lógicas a través de los condicionales lógicos “si”, “no”, y, “o”. Desde los circuitos más simples, como relojes y calculadoras, hasta complejos circuitos, como los microchips actuales, todos incluyen al menos una Unidad aritmético-lógica, que varía su poder y complejidad según su finalidad.

Fuente: <https://www.ecured.cu/Unidad_Aritm%C3%A9tico_L%C3%B3gica>

Unidad central (interna):

La memoria central o simplemente memoria (interna o principal) se utiliza para almacenar información (RAM, Random Access Memory). En general, la información almacenada puede ser de dos tipos: Instrucciones de un programa y datos con los que operan las instrucciones

Fuente: <https://edukativos.com/apuntes/archives/3768>

Memoria RAM:

Es la memoria de corto plazo del computador. Su función principal es recordar la información que tienes en cada una de las aplicaciones abiertas en el computador, mientras este se encuentre encendido.

Memoria ROM:

La ROM es la memoria no volátil lo que significa que la información se almacena de forma permanente en el chip. La memoria no depende de la corriente eléctrica para guardar los datos. Al contrario, los datos se escriben en celdas individuales usando el código binario. La memoria no volátil se utiliza en las partes de la computadora que no cambian, como la porción de arranque inicial del software o las instrucciones del firmware que hacen funcionar a la impresora. Apagar la computadora no tiene ningún efecto sobre la ROM. Los usuarios no pueden cambiar la memoria no volátil.

Fuente:<https://www.crucial.mx/articles/about-memory/what-is-the-difference-between-ram-and-rom>

Memoria auxiliar (externa):

La memoria secundaria o auxiliar es en un ordenador el conjunto de dispositivos que permiten almacenar datos complementarios a aquellos que se archivan en la memoria principal.

Fuente: https://www.definicionabc.com/tecnologia/memoria-auxiliar.php