

Introducción

Problemas, solución a problemas y algoritmos

Jonatan Gómez Perdomo, Ph.D.

jgomezpe@unal.edu.co

Arles Rodríguez, Ph.D.

aerodriguezp@unal.edu.co

Camilo Cubides, Ph.D.(c)

eccubidesg@unal.edu.co

Grupo de investigación en vida artificial – Research Group on Artificial Life – (Alife)

Departamento de Ingeniería de Sistemas e Industrial

Facultad de Ingeniería

Universidad Nacional de Colombia



Agenda

1 Problemas

- Análisis de problemas

2 Algoritmos

- Recursos de un algoritmo
- Ejemplos de algoritmos

3 Solución de problemas algorítmicos

- Metodología de programación



Definición de problema

Definición (Problema)

Se tiene un problema cuando se desea encontrar uno o varios objetos desconocidos (ya sean estos números, símbolos, diagramas, figuras u otras entidades), que cumplen condiciones o relaciones, previamente definidas, respecto a uno o varios objetos conocidos o desconocidos.



Solución de problemas

Solución de problemas

Solucionar un problema es encontrar los objetos desconocidos.

Para solucionar un problema se debe determinar de manera clara y concisa cuáles son los objetos conocidos, los desconocidos, el tipo de dichos objetos y, las condiciones y relaciones que hay entre ellos.



Tipos de problemas I

Bien condicionados: Un problema se dice bien condicionado si a partir de la entrada (no sobran ni faltan objetos conocidos) y a partir de la descripción de la relación (algunas veces llamada condición) entre la entrada y la salida se puede encontrar dicha salida.

En los problemas bien condicionados se cuenta con la información necesaria para resolver el problema y la solución no depende de una elección personal de índole emocional.



Tipos de problemas II

Ejemplo

Un granjero tiene cincuenta animales entre conejos y gallinas. Si la cantidad de patas de los animales es ciento cuarenta, ¿Cuántos conejos y cuántas gallinas tiene el granjero?.

Ejemplo

Si a es un número real positivo, entonces ¿Cuál es la solución a la ecuación?

$$x^2 - a = 0$$



Tipos de problemas II

Ejemplo

Un granjero tiene cincuenta animales entre conejos y gallinas. Si la cantidad de patas de los animales es ciento cuarenta, ¿Cuántos conejos y cuántas gallinas tiene el granjero?.

Ejemplo

Si a es un número real positivo, entonces ¿Cuál es la solución a la ecuación?

$$x^2 - a = 0$$



Tipos de problemas III

Mal condicionados: Son problemas en los cuales no se cuenta con la información necesaria para resolver el problema, o la respuesta está sujeta a una elección personal de cada ser humano (que depende de los sentimientos, emociones, cultura, religión, política, tradición, etc.).

Ejemplo

Un granjero tiene cincuenta animales entre conejos y gallinas. Si la cantidad de patas de los animales es ciento cuarenta, ¿Cuántos caballos y cuántos patos tiene el granjero?.

Ejemplo

Juanito tiene 5 manzanas, le da 2 a Pedro, calcule la distancia media del sol a la tierra.

Tipos de problemas III

Mal condicionados: Son problemas en los cuales no se cuenta con la información necesaria para resolver el problema, o la respuesta está sujeta a una elección personal de cada ser humano (que depende de los sentimientos, emociones, cultura, religión, política, tradición, etc.).

Ejemplo

Un granjero tiene cincuenta animales entre conejos y gallinas. Si la cantidad de patas de los animales es ciento cuarenta, ¿Cuántos caballos y cuántos patos tiene el granjero?.

Ejemplo

Juanito tiene 5 manzanas, le da 2 a Pedro, calcule la distancia media del sol a la tierra.

Tipos de problemas III

Mal condicionados: Son problemas en los cuales no se cuenta con la información necesaria para resolver el problema, o la respuesta está sujeta a una elección personal de cada ser humano (que depende de los sentimientos, emociones, cultura, religión, política, tradición, etc.).

Ejemplo

Un granjero tiene cincuenta animales entre conejos y gallinas. Si la cantidad de patas de los animales es ciento cuarenta, ¿Cuántos caballos y cuántos patos tiene el granjero?.

Ejemplo

Juanito tiene 5 manzanas, le da 2 a Pedro, calcule la distancia media del sol a la tierra.

Tipos de problemas IV

Computables: Son problemas para los cuales existe una secuencia finita de pasos “bien definidos” que permiten describir la relación entre entrada y salida, y que al ser aplicados a la entrada terminan en algún momento produciendo la salida.

Los problemas para los cuales no hay una forma de encontrar la solución para cada posible objeto conocido de entrada se denominan **indecidibles** o **no computables**.



Tipos de problemas V

Tratables: Son problemas para los cuales los recursos (tiempo y espacio) que se necesitan para su solución son finitos y no crecen exponencialmente a medida que aumenta el tamaño de la entrada.

Los problemas que son fácilmente solucionables para tamaños pequeños de entrada, pero no lo son para tamaños grandes de entrada debido a un crecimiento exponencial de alguno(s) de los recursos disponibles son llamados **intratables**.



Tipos de problemas VI

Solucionable: Son problemas para los cuales existe una salida valida (se puede encontrar) para la entrada dada. Los problemas **no solucionables** son aquellos para los que se tienen claro tanto la entrada como la relación entre la entrada y la salida pero no es posible encontrar dicha salida.

- **No solucionable \neq mal condicionado:** en los problemas mal condicionados es posible que la relación no este bien dada, no sea clara, sobren o falten objetos conocidos, mientras que en los no solucionables se tienen claro tanto la entrada como la relación entre entrada y salida.
- **No solucionable \neq intratable:** los problemas intratables son solucionables cuando el tamaño de la entrada es pequeño, pero resultan no solucionables para entradas medianas y/o grandes.



Agenda

1 Problemas

- Análisis de problemas

2 Algoritmos

- Recursos de un algoritmo
- Ejemplos de algoritmos

3 Solución de problemas algorítmicos

- Metodología de programación



Ejemplos de problemas I

Problema

¿Cuál es el número entero que es el cuadrado de un número entero dado?

- Este es un problema **bien condicionado**: la entrada es un número entero, tiene un objeto desconocido que es un número entero, la relación entre ellos es que el número entero desconocido es el cuadrado del número entero conocido.
- Es un problema **computable**: existe una secuencia finita de pasos bien definidos (multiplicar dígito por dígito y luego hacer sumas desplazadas) que permiten describir la relación entre la entrada y la salida.



Ejemplos de problemas II

- Es un problema **tratable**: si el tamaño de la entrada es n , siendo éste el número de dígitos del número entero conocido, la cantidad de dígitos de la salida a lo más es dos veces ($2n$), por lo que no crece exponencialmente en espacio. Requiere a lo mas n^2 operaciones de sumas y restas, por lo que no crece exponencialmente en tiempo.
- Es un problema **solucionable**: pues siempre es posible encontrar la salida dada la entrada.



Ejemplos de problemas III

Problema

¿A cuántas posiciones diferentes en un juego de ajedrez se puede llegar después de realizar n movimientos válidos (primero blancas, luego negras, luego blancas,...) a partir de la posición inicial?.

- Este es un problema **bien condicionado** pues la entrada es un número natural n que representa la cantidad de movimientos válidos a realizar. Se tiene un objeto desconocido que es un número natural que representa la cantidad de posiciones diferentes a las que se puede llegar desde la posición inicial. Se tiene clara la relación entre ellos, ya que el número natural desconocido es la cantidad de diferentes posiciones de un tablero de ajedrez a las que se puede llegar después de n movimientos válidos a partir de la posición inicial.



Ejemplos de problemas IV

- Es un problema **computable**, pues existe una secuencia finita de pasos bien definidos que permite relacionar la entrada con la salida:
 - 1 A partir de la posición dada genere todas las posibles posiciones a partir de los movimientos válidos y cuente cuantas posiciones se generaron e incremente la cuenta.
 - 2 Para cada posición generada realice el mismo procedimiento del paso 1. hasta alcanzar el máximo de movimientos dados.
- Es un problema **intratable** pues el número de diferentes posiciones crece exponencialmente a razón de 20^n de acuerdo al tamaño de la entrada n .
- Es un problema **solucionable** para valores pequeños de movimientos pero es **no solucionable** para valores medianos y/o grandes de la entrada.



Ejemplos de problemas V

Problema

De las siguientes imágenes, ¿Cuál es la más llamativa?



- Este es un problema **mal condicionado** pues a pesar de tener bien detallada la entrada (las imágenes de la figura) y tener una salida (es una de las imágenes de la entrada), no se tiene una clara relación entre la entrada y la salida: la noción de llamativo depende de quien vea las imágenes y no se puede describir.



Ejemplos de problemas VI

- Este problema es **no solucionable** por no tener una buena definición de la relación entre la entrada y la salida.
- Al no ser solucionable es **intratable** (no importa el tamaño de la entrada no se puede encontrar una solución válida para todos los posibles casos).
- Es **no computable** ya que no existe una serie finita de pasos bien definidos que permitan definir que es ser más llamativo.



Ejemplos de problemas VII

Problema

¿La secuencia de dígitos $d = d_1 d_2 d_3 \dots d_n$ se encuentra en la expansión decimal de π ?

- Este es un problema **bien condicionado**: la entrada es una secuencia de dígitos, la salida es un valor de verdad (verdadero o falso), y es clara la relación entre entradas y salidas (es verdadero si la secuencia de dígitos se encuentra en la expansión decimal de π , falso en otro caso).



Ejemplos de problemas VIII

- Para este problema existe una secuencia finita de pasos bien definidos que relaciona la entrada con la salida: ir desde la primera posición mirando si la secuencia buscada inicia en dicha posición, si no es de dicha manera avanzar a la siguiente y mirar y si no pasar a la tercera y de esa manera hasta encontrarla. A pesar de existir esta secuencia finita de acciones bien definidas, este problema es **no computable**, pues la aplicación de dicha secuencia de pasos puede que no pare para cada una de las posibles entradas, es decir, no termine en algún momento para algún caso.
- Es un problema **intratable** pues puede no parar (posiblemente se requiere tiempo infinito).
- En general es **no solucionable**.



Agenda

1 Problemas

- Análisis de problemas

2 Algoritmos

- Recursos de un algoritmo
- Ejemplos de algoritmos

3 Solución de problemas algorítmicos

- Metodología de programación



Algoritmos

Introducción

Durante toda la historia de la humanidad, se han construido diferentes máquinas y herramientas que le permiten al ser humano solucionar, de una mejor manera, diferentes tipos de problemas. El computador es una máquina que le permite al ser humano solucionar diferentes tipos de problemas que son computables (de allí el nombre de computador), es decir, problemas en los que la relación entre la entrada y la salida puede ser descrita por una secuencia finita de tareas bien definidas que al ser comunicadas al computador, para ser aplicadas a la entrada terminan en algún momento generando la salida.



Algoritmos y programación

Definición (Algoritmo)

Secuencia finita de tareas bien definidas, cada una de las cuales se puede realizar con una cantidad finita pero no limitada de recursos

Pero para que un computador, sea capaz de solucionar un problema dado, este debe ser programado.

Definición (Programar un computador)

Es la actividad mediante la cual se le describe (usualmente mediante un lenguaje de programación) la secuencia finita de pasos bien definidos (un algoritmo) con la que el computador puede solucionar un problema computable.



Características de un algoritmo

Un algoritmo debe cumplir con:

Precisión: Hay un orden preciso en el cual deben ejecutarse las tareas que conforman el algoritmo.

Determinismo: Cada vez que se ejecuten las tareas o pasos de un algoritmo, con las mismas condiciones iniciales, se deben obtener los mismos resultados.

Finitud: Un algoritmo debe usar una cantidad finita de recursos.



Estructura de un algoritmo

Un algoritmo está conformado por:

Datos: Donde se almacena la información de entrada, de salida o intermedia.

Instrucciones: Las acciones, procesos u operaciones que el algoritmo realiza sobre los datos.

Estructuras de control: Las que determinan el orden en que se ejecutarán las instrucciones del algoritmo.



Agenda

1 Problemas

- Análisis de problemas

2 Algoritmos

- Recursos de un algoritmo
- Ejemplos de algoritmos

3 Solución de problemas algorítmicos

- Metodología de programación



Recursos de un algoritmo I

Un algoritmo cuenta con dos recursos para su ejecución: tiempo y espacio

Tiempo definido en el número de tareas primitivas y/o básicas que debe ejecutar (por ejemplo sumas de enteros y operaciones lógicas).

Los computadores tienen un tiempo t definido para ejecutar una operación básica (definido por la velocidad, los Hertz del procesador), por lo tanto el tiempo requerido por un algoritmo será $T = t * o$ donde o es el número de operaciones básicas del algoritmo.



Recursos de un algoritmo II

Espacio definido en cantidad de memoria (variables) que debe mantener.

Esto ya que las variables son almacenadas en un número fijo s de bytes (Un byte es la medida mínima de almacenamiento actual en los computadores), por lo tanto el espacio requerido por un algoritmo será $S = s * v$, donde v es el número de variables del algoritmo.



Agenda

1 Problemas

- Análisis de problemas

2 Algoritmos

- Recursos de un algoritmo
- Ejemplos de algoritmos

3 Solución de problemas algorítmicos

- Metodología de programación



Ejemplos de algoritmos I

Problema

Un estudiante se encuentra en su casa (durmiendo) y debe ir a la universidad (¡¡a tomar la clase de programación!!), ¿qué debe hacer el estudiante?.

Solución

- 1 Dormir.
- 2 Hacer 1. hasta que suene el despertador (o lo llame la mamá).
- 3 Mirar la hora.



Ejemplos de algoritmos II

Solución (continuación)

- 4 ¿Hay tiempo suficiente?
 - 4.1 Si hay, entonces
 - 4.1.1 Bañarse.
 - 4.1.2 Vestirse.
 - 4.1.3 Desayunar.
 - 4.2 Si no,
 - 4.2.1 Vestirse.
- 5 Lavarse la boca.
- 6 Despedirse de la mamá y el papá.
- 7 ¿Hay tiempo suficiente?
 - 7.1 Si, caminar al paradero.
 - 7.2 Si no, correr al paradero.



Ejemplos de algoritmos III

Solución (continuación)

- 8 Hasta que pase un bus para la universidad hacer:
 - 8.1 Esperar el bus.
 - 8.2 Ver a las demás personas que esperan un bus.
- 9 Tomar el bus.
- 10 Mientras que no llegue a la universidad hacer:
 - 10.1 Seguir en el bus.
 - 10.2 Pelear mentalmente con el conductor.
- 11 Timbrar.
- 12 Bajarse.
- 13 Entrar a la universidad.



Ejemplos de algoritmos IV

Problema

Cambiar la rueda pinchada de un automóvil teniendo un gato mecánico en buen estado, una rueda de reemplazo y una llave inglesa.

Solución

- 1 Aflojar los tornillos de la rueda pinchada con la llave inglesa.
- 2 Ubicar el gato mecánico en su sitio.
- 3 Levantar el gato hasta que la rueda pinchada pueda girar libremente.
- 4 Quitar los tornillos con la llave inglesa.
- 5 Quitar la rueda pinchada.
- 6 Poner la rueda de repuesto y los tornillos.
- 7 Bajar el gato hasta que se pueda liberar.
- 8 Sacar el gato de su sitio.
- 9 Apretar los tornillos con la llave inglesa.

Ejemplos de algoritmos V

Problema

Sean los puntos $P = (a, b)$ y $Q = (c, d)$ que definen una recta, encontrar un segmento de recta perpendicular a la anterior que pasa por el punto medio de los puntos dados.

Solución

- 1 Trazar un círculo con centro en el punto P que pase por el punto Q .
- 2 Trazar un círculo con centro en el punto Q que pase por el punto P .
- 3 Trazar un segmento de recta entre los puntos de intersección de las circunferencias trazadas.
- 4 El segmento de recta trazada es el buscado.



Ejemplos de algoritmos VI

Ejecución

Sean P y Q los dos siguientes puntos

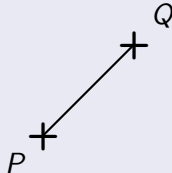


Figure: Paso 0.



Ejemplos de algoritmos VII

Ejecución (continuación)

- 1 Trazar una circunferencia con centro en el punto P que pase por el punto Q .

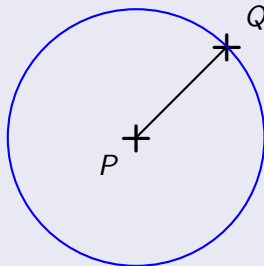


Figure: Paso 1.



Ejemplos de algoritmos VIII

Ejecución (continuación)

- 2 Trazar una circunferencia con centro en el punto Q que pase por el punto P .

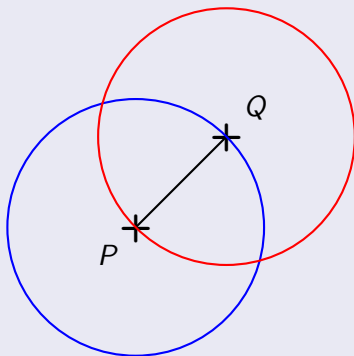


Figure: Paso 2.

Ejemplos de algoritmos IX

Ejecución (continuación)

- 3 Trazar un segmento de recta entre los puntos de intersección de las circunferencias trazadas.

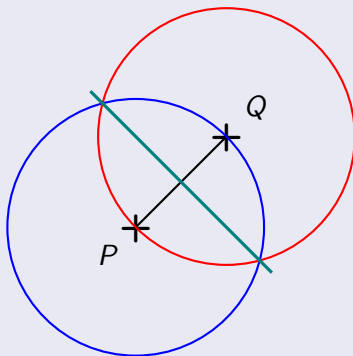


Figure: Paso 3.

Ejemplos de algoritmos X

Ejecución (continuación)

- 4 El segmento de recta trazada es el buscado.

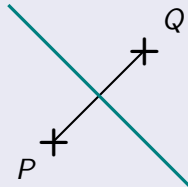


Figure: Paso 4.



Ejemplos de algoritmos XI

Problema

Encontrar los números primos entre 0 y 50.

Solución

- 1 Escribir los números de 0 al 50.
- 2 Tachar los números 0 y 1 ya que no son números primos.
- 3 Para i entre 2 y la raíz cuadrada de 50, si el número i no está tachado, tachar los múltiplos del número i .
- 4 Los números que no se tacharon son los números primos entre 0 y 50.



Ejemplos de algoritmos XII

Problema

Realizar la suma de los números 2448 y 5746.

Solución

- 1 Colocar los números el primero encima del segundo, de tal manera que las unidades, decenas, centenas, etc., de los números queden alineadas. Trazar una línea debajo del segundo número.
- 2 Empezar por la columna más a la derecha.
- 3 Sumar los dígitos de dicha columna.
- 4 Si la suma es mayor a 9 anotar un 1 encima de la siguiente columna a la izquierda y anotar debajo de la línea las unidades de la suma. Si no es mayor anotar la suma debajo de la línea.
- 5 Si hay más columnas a la izquierda, pasar a la siguiente columna a la izquierda y volver al numeral 3.
- 6 El número debajo de la línea es la solución.

Ejemplos de algoritmos XIII

Solución

(El símbolo ↓ indica la columna que se está operando)

| | | |
|----|--|---|
| 1. | Colocar los números el primero encima del segundo, de tal manera que las unidades, decenas, centenas, etc., de los números queden alineadas. Trazar una línea debajo del segundo número. | $\begin{array}{r} 2448 \\ + 5746 \\ \hline \end{array}$ |
| 2. | Empezar por la columna más a la derecha. | $\begin{array}{r} 2448 \\ + 5746 \\ \hline \end{array}$ <div style="text-align: right; margin-right: 20px;">↓</div> |
| 3. | Sumar los dígitos de dicha columna. | $8 + 6 = 14$ |



Ejemplos de algoritmos XIV

Solución

| | | |
|----|---|---|
| 4. | Si la suma es mayor a 9 anotar un 1 encima de la siguiente columna a la izquierda y anotar abajo de la línea las unidades de la suma. Si no es mayor anotar la suma debajo de la línea. | $ \begin{array}{r} 1 \downarrow \\ 2448 \\ + 5746 \\ \hline 4 \end{array} $ |
| 5. | Si hay más columnas a la izquierda, pasar a la siguiente columna a la izquierda y volver al paso 3. del algoritmo. | $ \begin{array}{r} \downarrow \\ 1 \\ 2448 \\ + 5746 \\ \hline 4 \end{array} $ |



Ejemplos de algoritmos XV

Solución

| | | |
|----|---|--|
| 3. | Sumar los dígitos de dicha columna. | $1 + 4 + 4 = 9$ |
| 4. | Si la suma es mayor a 9 anotar un 1 encima de la siguiente columna a la izquierda y anotar abajo de la línea las unidades de la suma. Si no es mayor anotar la suma debajo de la línea. | $ \begin{array}{r} \downarrow \\ 1 \\ 2448 \\ + 5746 \\ \hline 94 \end{array} $ |
| 5. | Si hay más columnas a la izquierda, pasar a la siguiente columna a la izquierda y volver al paso 3. del algoritmo. | $ \begin{array}{r} \downarrow \\ 1 \\ 2448 \\ + 5746 \\ \hline 94 \end{array} $ |



Ejemplos de algoritmos XVI

Solución

| | | |
|----|---|---|
| 3. | Sumar los dígitos de dicha columna. | $4 + 7 = 11$ |
| 4. | Si la suma es mayor a 9 anotar un 1 encima de la siguiente columna a la izquierda y anotar abajo de la línea las unidades de la suma. Si no es mayor anotar la suma debajo de la línea. | $ \begin{array}{r} \downarrow \\ 1 \quad 1 \\ 2448 \\ + 5746 \\ \hline 194 \end{array} $ |
| 5. | Si hay más columnas a la izquierda, pasar a la siguiente columna a la izquierda y volver al paso 3. del algoritmo. | $ \begin{array}{r} \downarrow \\ 1 \quad 1 \\ 2448 \\ + 5746 \\ \hline 194 \end{array} $ |
| 3. | Sumar los dígitos de dicha columna. | $1 + 2 + 5 = 8$ |



Ejemplos de algoritmos XVII

Solución

| | | |
|----|---|--|
| 4. | Si la suma es mayor a 9 anotar un 1 encima de la siguiente columna a la izquierda y anotar abajo de la línea las unidades de la suma. Si no es mayor anotar la suma debajo de la línea. | $ \begin{array}{r} \downarrow \\ 1 \quad 1 \\ 2448 \\ + 5746 \\ \hline 8194 \end{array} $ |
| 5. | Si hay más columnas a la izquierda, pasar a la siguiente columna a la izquierda y volver al paso 3. del algoritmo. | |
| 6. | El número debajo de la línea es la solución. | 8194 |



Agenda

1 Problemas

- Análisis de problemas

2 Algoritmos

- Recursos de un algoritmo
- Ejemplos de algoritmos

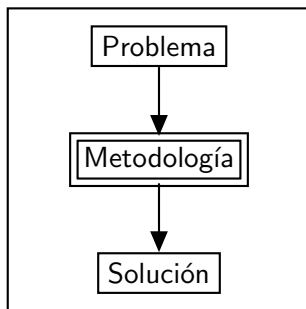
3 Solución de problemas algorítmicos

- Metodología de programación



Solución de problemas algorítmicos

Dado un problema, para encontrar la solución a dicho problema se puede pensar en utilizar una metodología general que se aplique sistemáticamente de tal manera que como resultado se obtenga la solución del problema, así como se muestra en el siguiente diagrama



Agenda

1 Problemas

- Análisis de problemas

2 Algoritmos

- Recursos de un algoritmo
- Ejemplos de algoritmos

3 Solución de problemas algorítmicos

- Metodología de programación



Metodología de programación I

Una metodológica usualmente utilizada para resolver problemas algorítmicos está compuesta por la siguiente serie de etapas:

- 1 Análisis del problema
- 2 Especificación del problema
- 3 Diseño del algoritmo
- 4 Prueba del algoritmo y refinamiento
- 5 Codificación
- 6 Prueba y verificación



Metodología de programación II

Análisis del problema: Entender de manera clara el problema que se está resolviendo. Esto involucra el proceso de identificación de objetos conocidos, objetos desconocidos y condiciones del problema.

Especificación del problema: Descripción clara y precisa de

- Las entradas del problema.
- Las salidas del problema.
- Las condiciones, es decir la dependencia que mantendrán las salidas obtenidas con las entradas recibidas.



Metodología de programación III

Diseño del algoritmo: Es la fase en la que se construye el algoritmo que permitirá encontrar la solución al problema. Es una buena practica de programación subdividir el problema en problemas más sencillos de resolver.

Prueba del algoritmo y refinamiento: Proceso de seguimiento del algoritmo para verificar que cumple con la especificación. Si no se cumple con la especificación se va refinando hasta lograr el objetivo.



Metodología de programación IV

Codificación: Proceso en el cual se escribe el algoritmo en un lenguaje de programación, utilizando alguna herramienta de programación.

Prueba y verificación: Proceso en el cual se corrigen los errores de sintaxis y de lógica del programa, hasta lograr que el programa resuelva el problema de forma exitosa.



Metodología de programación V

Ejercicio

Utilizar la metodología de programación presentada previamente para resolver el siguiente problema bien condicionado presentado previamente.

Ejemplo

Un granjero tiene cincuenta animales entre conejos y gallinas. Si la cantidad de patas de los animales es ciento cuarenta, ¿Cuántos conejos y cuántas gallinas tiene el granjero?.



Ejercicio

Problemas para estudio

- Los Simpson: acertijo del zorro, el pato y el maíz
url: <https://www.youtube.com/watch?v=Lp-nuB0k5Sg>
- Cuento de Ajedrez Nueva Versión
url: https://www.youtube.com/watch?v=FR7_XUwA63g
- El Hotel Infinito
url: <https://www.youtube.com/watch?v=iAF37vVeV-Y>

