

Resolución de Problemas y Algoritmos

Clase 1: algoritmos y computadoras



Dr. Diego R. García



Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur
Bahía Blanca - Argentina

BIENVENIDOS

Presentación

Resolución de Problemas y Algoritmos (RPA)

Profesor: Diego R. García
Asistente: Natalia Nill
Ayudantes: Leandro Volpe, Franco Morales y Fabio Gallo

Horarios:
miércoles 08.15 a 12.00 hs **Aula:** [8 Palihue](#)
viernes 08.15 a 12.00 hs **Aula:** [7 Palihue](#)

Resolución de Problemas y Algoritmos Dr. Diego R. García 3

¿De qué se trata esta materia?

Resolución de Problemas y Algoritmos

Resolución de Problemas

¿De dónde surgen los problemas para resolver?

Resolución de Problemas y Algoritmos Dr. Diego R. García 4

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente:
"Resolución de Problemas y Algoritmos. Notas de Clase". Diego R. García. Universidad Nacional del Sur. (c) 15/08/2019.

Introducción: resolver problemas con computadoras

En la actualidad, las computadoras resuelven una gran cantidad de problemas de nuestra sociedad.

Algunos ejemplos:

- ✓ *mantener la estación espacial en órbita,*
- ✓ *comunicarme con mis contactos en una red social,*
- ✓ *monitor de terapia intensiva de un hospital,*
- ✓ *buscar información en Internet,*
- ✓ *piloto automático de un avión.*



Desafío: encuentre alguna actividad en la cual no se usa una computadora. (*sugerencia antes de “cantar victoria” realice una búsqueda en Internet para ver si se usa o no*).

Introducción: resolver problemas con computadoras

- **En la actualidad,** para que las computadoras puedan resolver un problema, **alguien le debe indicar de manera muy precisa, y paso por paso cómo hacerlo.**
- **Como profesional en informática deberá poseer la capacidad de “programar” a una computadora para resolver problemas.** (*¿Qué es programar?*)
- **¿Alguien trajo una computadora?**
- **En estos días, cuando uno dice “computadora” la mayoría piensa en algo así como en la figura:**
- **Sin embargo, la forma externa no es lo que lo define que es una computadora 😊**



Algunas computadoras

Ecógrafo **Aibo** **Equipo de Cirugía Da Vinci**

smart TV

Resolución de Problemas y Algoritmos Dr. Diego R. García 8

Computadoras

Las computadoras pueden estar en dispositivos móviles y también empotradas en otros dispositivos (Embedded computers).
Por ejemplo en juguetes, lentes, audífonos, aviones, o robots industriales.

Si tiene más curiosidad vea los enlaces de las referencias al final de la presentación.

Pero entonces ¿Qué es lo que define a una computadora?

Resolución de Problemas y Algoritmos Dr. Diego R. García 9

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente:
"Resolución de Problemas y Algoritmos. Notas de Clase". Diego R. García. Universidad Nacional del Sur. (c) 15/08/2019.

Conceptos: Computadora

Una computadora es un sistema digital con tecnología microelectrónica compuesta por:

- 1- CPU (Unidad Central de Proceso)**
- 2- Memoria**
- 3- Dispositivos de Entrada y Salida**

Interconectados por un canal de comunicación (bus)

bus: canal de comunicación
(Ejemplo: USB es Universal Serial Bus)

Resolución de Problemas y Algoritmos Dr. Diego R. García 10

Conceptos: Computadora

Una computadora es un sistema digital con tecnología microelectrónica compuesta por:

- 1- CPU (Unidad Central de Proceso)**
- 2- Memoria**
- 3- Dispositivos de Entrada y Salida**

Interconectados por un canal de comunicación (bus)

Podemos distinguir:

- **computadoras de propósito general:** *PC – notebook – ultrabook – tablet PC – smartphone*
- **computadoras dedicadas (a veces empotradas):** *celular- rep. MP3 - impresora- consola de juegos - inyector de combustible de auto - placa de video – smart TV*

Resolución de Problemas y Algoritmos Dr. Diego R. García 11

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente:
 “Resolución de Problemas y Algoritmos. Notas de Clase”. Diego R. García. Universidad Nacional del Sur. (c) 15/08/2019.

Presentación

Resolución de Problemas y Algoritmos

Hablaremos ahora sobre algoritmos en informática

Algoritmos

Resolución de Problemas y Algoritmos Dr. Diego R. García 12

Resolución de Problemas y Algoritmos (RPA)

- En RPA veremos **técnicas para resolver problemas**.
- La solución encontrada para el problema será expresada en un **algoritmo**
- Este algoritmo **indicará cómo resolver** el problema.
- El algoritmo será **implementado** en un lenguaje de programación y se podrá usar en una **computadora**.

(De esta manera podremos programar una computadora para que resuelva problemas)

¿Algoritmo? ¿Lenguaje de programación?

¿Implementar?

Resolución de Problemas y Algoritmos Dr. Diego R. García 13

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente:
"Resolución de Problemas y Algoritmos. Notas de Clase". Diego R. García. Universidad Nacional del Sur. (c) 15/08/2019.

Concepto: Algoritmo

Un algoritmo es la especificación de una **secuencia de pasos u operaciones**, que al ser ejecutadas permiten resolver un problema.

Un algoritmo debe tener un único punto de inicio y al menos un punto final; y todos sus pasos deben estar expresados con operaciones comprensibles para quién las ejecutará (a las cuales llamamos **primitivas**).

Un algoritmo nos brinda una manera particular de expresar “cómo” resolver un problema.

¿Palabras nuevas?

Un algoritmo es la **especificación** de una **secuencia** de pasos u operaciones, que al ser **ejecutadas** permiten resolver un problema.

Un algoritmo debe tener un único punto de inicio y al menos un punto final; y todos sus pasos deben estar expresados con operaciones comprensibles para quién las ejecutará (a las cuales llamamos **primitivas**).

Una primitiva es una operación conocida, utilizada en un algoritmo y considerada como básica.

Concepto: Algoritmo

Un algoritmo es la **especificación** de una **secuencia** de pasos u operaciones, que al ser **ejecutadas** permiten resolver un problema.

Un algoritmo debe tener un **único punto de inicio** y al menos un punto final; y todos sus pasos deben estar expresados con operaciones comprensibles para quién las ejecutará (a las cuales llamamos **primitivas**).

Importante: Al construir un algoritmo debe tenerse en claro cuál es el problema que el algoritmo resolverá al ser ejecutado; y cuál es el conjunto de operaciones primitivas que pueden ser utilizadas.

Ejemplo: conectarse a una red Wi-Fi

Problema: Escriba un algoritmo que permita elegir a un usuario de un dispositivo una red Wi-Fi disponible para conectarse. Puede utilizar sólo las operaciones primitivas mostradas abajo.

Primitivas disponibles
(ordenadas alfabéticamente):

- bajar volumen
- buscar Wi-Fi accesibles
- conectar a Wi-Fi elegida
- esperar elección de usuario
- mostrar Wi-Fi detectadas
- silenciar parlantes
- subir volumen

Algoritmo: ??????
(veamos la definición.....)
... secuencia...
...operaciones...
...resuelve el problema...



Ejemplo: conectarse a una red Wi-Fi

Problema: Escriba un algoritmo que permita al usuario de un dispositivo elegir una red Wi-Fi disponible para conectarse. Puede utilizar sólo las operaciones primitivas mostradas abajo.

Primitivas disponibles (ordenadas alfabéticamente):

- bajar volumen
- buscar Wi-Fi accesibles
- conectar a Wi-Fi elegida
- esperar elección de usuario
- mostrar Wi-Fi detectadas
- silenciar parlantes
- subir volumen

Algoritmo:

- ↓ buscar Wi-Fi accesibles
- mostrar Wi-Fi detectadas
- esperar elección de usuario
- ↓ conectar a Wi-Fi elegida



Computadoras en la industria



El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente: "Resolución de Problemas y Algoritmos. Notas de Clase". Diego R. García. Universidad Nacional del Sur. (c) 15/08/2019.

Ejemplo: algoritmo para un problema simplificado

Escribir un algoritmo para que un brazo robot coloque un producto en su envase y lo cierre.

Primitivas disponibles (ordenadas alfabéticamente):

- cerrar envase
- esperar por envase vacío
- poner producto en envase
- tomar producto



Algoritmo:

- tomar producto
- esperar por envase vacío
- poner producto en envase
- cerrar envase

Resolución de Problemas y Algoritmos

Dr. Diego R. García

20

Búsqueda de errores

No cualquier secuencia de operaciones es correcta para resolver el problema. Vea la que está a continuación ¿qué problema encuentra?



- tomar producto
- esperar por envase vacío
- cerrar envase X
- poner producto en envase

Es importante asegurarse que una solución sea correcta antes de que sea utilizada.

Si hay errores, su solución puede afectar considerablemente a otras personas o usted mismo.

Resolución de Problemas y Algoritmos

Dr. Diego R. García

21

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente:
 “Resolución de Problemas y Algoritmos. Notas de Clase”. Diego R. García. Universidad Nacional del Sur. (c) 15/08/2019.

Concepto: Traza de un algoritmo

Una traza es una simulación de la ejecución real de los pasos de un algoritmo, en la cual se lleva cuenta de los movimientos realizados y los cambios que se producen en los elementos o datos involucrados.

- La **traza** es una herramienta muy simple que ayuda a verificar si un algoritmo es correcto (*verá más adelante que no es la única herramienta*).
- Si la **traza** se realiza correctamente (sin hacer trampa), y **no se obtiene** el resultado esperado, entonces **hay un error** en el algoritmo.
- (*¿Por qué los pilotos practican primero en un simulador?*)

Realizar trazas para estas posibles soluciones

Una traza es una simulación de la ejecución real de los pasos de un algoritmo, en la cual se lleva cuenta de los movimientos realizados y los cambios que se producen en los elementos o datos involucrados.

Problema: Escribir un algoritmo para disponer de un **termo lleno con agua caliente**.

Operaciones disponibles
(ordenadas alfabéticamente):

- calentar agua en pava
- completar termo con pava
- poner tapón del termo
- sacar tapón del termo
- vaciar termo

Tarea: Realice una traza de la propuesta 1.

¿Resuelve el problema?

Propuesta 1:

- sacar tapón del termo
- vaciar termo
- calentar agua en pava
- poner tapón del termo

Realizar trazas para estas posibles soluciones

Problema: Tener un termo lleno con agua caliente.

Operaciones disponibles
(ordenadas alfabéticamente):

- calentar agua en pava
- completar termo con pava
- poner tapón del termo
- sacar tapón del termo
- vaciar termo

Propuesta 2:

- sacar tapón del termo
- calentar agua en pava
- completar termo con pava
- vaciar termo
- poner tapón del termo

Tarea: utilice una traza para detectar los errores en las propuestas, luego trate de encontrar una solución correcta y muestre su solución en la clase práctica. ¿cómo puede evaluar que su nueva propuesta no tiene errores?

Realizar trazas para estas posibles soluciones

Problema: Tener un termo lleno con agua caliente.

Operaciones disponibles
(ordenadas alfabéticamente):

- calentar agua en pava
- completar termo con pava
- poner tapón del termo
- sacar tapón del termo
- vaciar termo

Propuesta 3:

- sacar tapón del termo
- calentar agua en pava
- completar termo con pava
- poner tapón del termo

- ¿ Produce el resultado esperado? (no se apure a responder....)
- ¿ Qué ocurre si el termo ya tenía agua fría?

Para especificar un algoritmo podemos utilizar:

- (1) **Secuencia:** de primitivas (cómo en los ejemplos vistos)
- (2) **Condiciones:** permiten especificar alternativas. Requieren de condiciones o de primitivas de "testeo" Ejemplo:

Si hay Wi-Fi disponible
entonces conecto red Wi-Fi
de lo contrario conecto red telefonía móvil

- (3) **Repeticiones:**
 Permiten especificar de una manera abreviada una secuencia repetida de operaciones.
 Puede ser una repetición basada en una condición o una repetición incondicional (se repite un número fijo de veces).

Para especificar un algoritmo podemos utilizar:

- (1) **Secuencia**
 Se asume de manera implícita que el **orden** en que deben ejecutarse es de **arriba hacia abajo; y de izquierda a derecha** si están en la misma línea.

Algoritmo:

↓
 buscar Wi-Fi accesibles
 mostrar Wi-Fi detectadas
 esperar elección de usuario
 ↓
 conectar a Wi-Fi elegida

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente:
 "Resolución de Problemas y Algoritmos. Notas de Clase". Diego R. García. Universidad Nacional del Sur. (c) 15/08/2019.

Para especificar un algoritmo podemos utilizar:

(2) **Condiciones:** permiten indicar opciones alternativas dentro de un algoritmo.

Si “condición”
entonces “secuencia 1”
de lo contrario “secuencia 2”

- Si al momento de ejecutarse la “condición” da como resultado **verdadero** se ejecutará solamente “secuencia 1” (y no se ejecutará “secuencia 2”).
- En cambio si la “condición” da resultado **falso**, se ejecutará solamente “secuencia 2” (y no se ejecutará “secuencia 1”).

Es decir, la condición permite expresar dos secuencias de acciones alternativas y excluyentes (solo una se ejecutará, no ambas).

Ejemplo:

(2) **Condiciones:** permiten indicar opciones alternativas dentro de un algoritmo.

Si hay Wi-Fi disponible
entonces conecto red Wi-Fi
de lo contrario conecto red telefonía móvil

En este ejemplo, la primitiva “**hay Wi-Fi disponible**” es una operación de detección que retorna verdadero o falso.

- Si al momento de ejecutarse, “hay Wi-Fi disponible” retorna **verdadero** se ejecutará solamente “conecto red Wi-Fi”.
- En cambio “hay Wi-Fi disponible” retorna falso, entonces se ejecutará solamente “conecto red telefonía móvil”

Como la condición permite expresar **dos secuencias de acciones alternativas y excluyentes**, entonces nunca se conectará simultáneamente a las dos redes.

Ejemplo: conectarse a una red Wi-Fi

Primitivas de “acción”:

- buscar Wi-Fi accesibles
- conectar a Wi-Fi elegida
- elegir una conocida
- esperar elección de usuario
- mostrar texto “no Wi-Fi”
- mostrar Wi-Fi detectadas
- silenciar parlantes
- subir volumen

Primitivas de “sensado”

- hay Wi-Fi accesibles
- hay Wi-Fi conocida

Algoritmo:

buscar Wi-Fi accesibles

SI hay Wi-Fi accesibles

ENTONCES

mostrar Wi-Fi detectadas

esperar elección de usuario

conectar a Wi-Fi elegida

DE LO CONTRARIO

mostrar texto “no Wi-Fi”

Resolución de Problemas y Algoritmos
Dr. Diego R. García
30

Para especificar un algoritmo podemos utilizar:

- (1) Secuencia
- (2) Condiciones:
 - Observación: la opción “de lo contrario” puede omitirse y en ese caso si la condición es falsa simplemente no se ejecuta la secuencia 1.

Si “condición”

entonces “secuencia 1”

Ejemplo:

Si tiene agua el termo

entonces vaciar termo

Llenar termo con agua caliente

Resolución de Problemas y Algoritmos
Dr. Diego R. García
31

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente:

“Resolución de Problemas y Algoritmos. Notas de Clase”. Diego R. García. Universidad Nacional del Sur. (c) 15/08/2019.

Problema propuesto

Considere un celular en el cual los contactos tienen un número de teléfono, nombre, y de manera opcional una foto.

El celular tiene un tono de llamada predeterminado, pero algunos contactos pueden tener configurado un tono de llamada particular.

Escribir un algoritmo que, para una llamada entrante, el dispositivo use el tono de llamada adecuado y luego muestre la información disponible del contacto.

Primitivas disponibles

- mostrar foto de número entrante
- mostrar nombre de número entrante
- reproducir tono de llamada número entrante
- reproducir tono predeterminado

Primitivas que retornan verdadero o falso:

- número entrante pertenece a contactos
- número entrante tiene foto asociada
- número entrante tiene tono de llamada particular

Resolución de Problemas y Algoritmos Dr. Diego R. García 32

Problema propuesto

Escribir un algoritmo que, para una llamada entrante, el dispositivo use el tono de llamada adecuado y luego muestre la información disponible del contacto.

Primitivas disponibles

- mostrar foto de número entrante
- mostrar nombre de número entrante
- reproducir tono de llamada número entrante
- reproducir tono predeterminado
- número entrante pertenece a contactos
- número entrante tiene foto asociada
- número entrante tiene tono de llamada particular

Algoritmo Llamada entrante

- SI número entrante tiene tono de llamada particular
ENTONCES reproducir tono de llamada número entrante
DE LO CONTRARIO reproducir tono predeterminado
- SI número entrante pertenece a contactos
ENTONCES mostrar nombre número entrante
- SI número entrante tiene foto asociada
ENTONCES mostrar foto número entrante

Resolución de Problemas y Algoritmos Dr. Diego R. García 33

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente:

“Resolución de Problemas y Algoritmos. Notas de Clase”. Diego R. García. Universidad Nacional del Sur. (c) 15/08/2019.

Concepto: Traza de un algoritmo

Una traza es una simulación de la ejecución real de los pasos de un algoritmo, en la cual se lleva cuenta de los movimientos realizados y los cambios que se producen en los elementos o datos involucrados.

- Para verificar una algoritmo hay que elegir cuidadosamente los **casos de prueba** y realizar una traza para cada uno de los caso de prueba elegidos.

Los casos de prueba son situaciones particulares en los cuales se aplicaría el algoritmo desarrollado, y se utilizan para verificar que el algoritmo se comporta de la manera esperada. Según sea la naturaleza del problema en cual se usará el algoritmo, los casos de prueba pueden incluir valores particulares para los datos u otros elementos que están involucrados en la solución del problema.

Verificación del algoritmo

Algoritmo Llamada entrante

- SI número entrante tiene tono de llamada particular
ENTONCES reproducir tono de llamada número entrante
DE LO CONTRARIO reproducir tono predeterminado
- SI número entrante pertenece a contactos
ENTONCES mostrar nombre número entrante
- SI número entrante tiene foto asociada
ENTONCES mostrar foto número entrante

¿Hay verificar con todos los contactos de un teléfono? ¿y si hay uno solo? ¿y si son 300? ¿Cuántos casos diferentes realmente hay?

Casos de prueba:

1. el número entrante no pertenece a contactos
2. nro. pertenece a contactos, tiene foto, tiene tono particular
3. nro. pertenece a contactos, no tiene foto, tiene tono particular
4. nro. pertenece, tiene foto, no tiene tono particular
5. nro. pertenece, no tiene foto, no tiene tono particular

Verificación del algoritmo

Propuesta B:

- SI número entrante tiene tono de llamada particular ENTONCES reproducir tono de llamada número entrante
- SI número entrante pertenece a contactos ENTONCES mostrar nombre número entrante
- SI número entrante tiene foto asociada ENTONCES mostrar foto número entrante

Casos de prueba:

1. el número entrante no pertenece a contactos ←
2. nro. pertenece a contactos, tiene foto, tiene tono particular
3. nro. pertenece a contactos, no tiene foto, tiene tono particular
4. nro. pertenece, tiene foto, no tiene tono particular
5. nro. pertenece, no tiene foto, no tiene tono particular

El caso de prueba 1 muestra que **la propuesta B falla** porque no reproduce ningún tono de llamada (ni predeterminado ni particular)

Resolución de Problemas y Algoritmos Dr. Diego R. García 36

Verificación del algoritmo

Propuesta C:

- SI número entrante tiene tono de llamada particular ENTONCES reproducir tono de llamada número entrante DE LO CONTRARIO reproducir tono predeterminado
- SI número entrante pertenece a contactos ENTONCES mostrar nombre número entrante DE LO CONTRARIO mostrar foto número entrante

Casos de prueba:

1. el número entrante no pertenece a contactos ←
2. nro. pertenece a contactos, tiene foto, tiene tono particular ←
3. nro. pertenece a contactos, no tiene foto, tiene tono particular
4. nro. pertenece, tiene foto, no tiene tono particular ←
5. nro. pertenece, no tiene foto, no tiene tono particular

En el caso 1 la **propuesta C falla** porque intentará mostrar una foto de un contacto que no existe. Además, en los casos 2 y 4, **también falla** porque no mostrará la foto aunque esta existe.

Resolución de Problemas y Algoritmos Dr. Diego R. García 37

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente:
 "Resolución de Problemas y Algoritmos. Notas de Clase". Diego R. García. Universidad Nacional del Sur. (c) 15/08/2019.

Algoritmos en la vida cotidiana

- Los algoritmos no son solamente propios de la Informática.
- También están en nuestra vida cotidiana, a veces con el nombre de “Instrucciones”, “Recetas” o “Pasos”.
- Y también están presentes en otra disciplinas profesionales y científicas: matemática, medicina, etc.

Problema propuesto

Se quiere preparar correctamente líquido refrigerante para un automóvil y en la etiqueta de la botella de 1 litro dice:

“Antes de usar, diluir el contenido de este envase en 2 litros de agua destilada.”

Observe que no dice “cómo” hacerlo. Queremos escribir una solución que indique “cómo” resolver el problema siguiendo una secuencia de pasos (un algoritmo) y para esto debemos saber que operaciones (**primitivas**) podemos usar y que elementos disponemos.



Enunciado de problema propuesto

Escribir un algoritmo que indique cómo preparar correctamente líquido refrigerante. Se sabe que: **“Antes de usar, diluir todo el contenido de este envase en 2 litros de agua destilada.”**

Se dispone de: 1 botella con un litro de líquido refrigerante, 5 botellas vacías de 1,5 litros, un bidón de 5 litros con algo de agua y 1 botellita vacía de 500ml. Un recipiente con más de 3 litros de agua destilada.

Se pueden utilizar las siguientes **primitivas**:



- **vaciar** recipiente1 (el cual quedará vacío)
- **llenar** recipiente1 con recipiente2 (recipiente2 debe tener suficiente líquido y recipiente1 quedará lleno)
- **trasvasar** recipiente1 al recipiente2 (recipiente2 deberá tener lugar suficiente y recipiente1 quedará vacío)



Propuesta 2 ¿es correcta?

- Vaciar el bidón de 5 litros
- Llenar la botella de 1,5 litros con bidón de agua destilada
- Trasvasar la botellita de medio litro al bidón de 5 litros
- Llenar la botellita de medio litro con bidón de agua destilada
- Trasvasar la botella de 1,5 litros al bidón de 5 litros
- Trasvasar el líquido refrigerante al bidón de 5 litros

Realizar la traza 😊

- Vaciar el bidón de 5 litros
- Llenar la botella de 1,5 litros con bidón de agua destilada
- Trasvasar la botellita de medio litro al bidón de 5 litros
- Llenar la botellita de medio litro con bidón de agua destilada
- Trasvasar la botella de 1,5 litros al bidón de 5 litros
- Trasvasar el líquido refrigerante al bidón de 5 litros

Propuesta 2
¿es correcta?



Una traza ayuda a encontrar errores



0	?	vacía	vacía	llena	>3 lit
1	vacío	vacía	vacía	llena	>3 lit
2	vacío	vacía	1.5 lit.	llena	> 1.5
3	vacío	vacía	1.5 lit.	llena	> 1.5
4	vacío	0.5 lit	1.5 lit.	llena	> 1 lit
5	1.5 lit.	0.5 lit	vacía	llena	> 1 lit
6	1.5lit+L.R.	0.5 lit	vacía	vacía	> 1 lit

Para especificar un algoritmo utilizamos:

- (1) Secuencia
- (2) Condiciones
- (3) Repeticiones: permiten especificar de una manera abreviada una secuencia repetida de operaciones.

Llenar botella
 Pasar a bidón
 Llenar botella
 Pasar a bidón
 Llenar botella
 Pasar a bidón
 Guardar bidón

Se puede indicar que se repita un número **fijo** de veces.

Repetir 3 veces:

Llenar botella
 Pasar a bidón
 Guardar bidón

Problema propuesto

Utilizando las primitivas “tiene agua”, “vaciar”, “llenar” y “trasvasar”, escribir un algoritmo para obtener exactamente 10 litros de agua. Se dispone de un bidón de 12 litros y 1 botella de medio litro.

Algoritmo 1:

Si tiene agua el bidón **entonces** vaciar el bidón

Si tiene agua la botella **entonces** vaciar la botella

Repetir 20 veces lo que sigue:

- Llenar botella
- Trasvasar botella a bidón

Problema propuesto

Utilizando las primitivas “tiene agua”, “vaciar”, “llenar” y “trasvasar”, escribir un algoritmo para obtener exactamente 10 litros de agua. Se dispone de un bidón de 12 litros y 1 botella de medio litro.

Algoritmo 2:

Si tiene agua la botella **entonces** vaciar la botella

Llenar el bidón

Repetir 4 veces lo que sigue:

- Trasvasar bidón a botella
- Vaciar botella

Para especificar un algoritmo utilizamos:

- (1) Secuencia
- (2) Condiciones
- (3) Repeticiones: permiten especificar de una manera abreviada una secuencia repetida de operaciones.

Se puede indicar que se repita **hasta** que se cumpla una condición.

Repetir hasta bidón lleno:

```

Llenar botella
Pasar a bidón
Guardar bidón
    
```

Para especificar un algoritmo utilizamos:

- (1) Secuencia
- (2) Condiciones
- (3) Repeticiones: permiten especificar de una manera abreviada una secuencia repetida de operaciones.

Se puede indicar que se repita **mientras** que se cumpla una condición.

Repetir mientras bidón NO esté lleno:

```

Llenar botella
Pasar a bidón
Guardar bidón
    
```

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente: "Resolución de Problemas y Algoritmos. Notas de Clase". Diego R. García. Universidad Nacional del Sur. (c) 15/08/2019.

Ejemplo: algoritmo para un problema simplificado

Si se dispone de una operación que indica que hay productos, entonces podemos indicar que se repita la secuencia mientras que se cumpla la condición.



Primitivas disponibles (ordenadas alfabéticamente):

- cerrar envase
- esperar por envase vacío
- **hay productos**
- poner producto en envase
- tomar producto

Algoritmo:

Repetir mientras hay productos

- tomar producto
- esperar por envase vacío
- poner producto en envase
- cerrar envase

Resolución de Problemas y Algoritmos
Dr. Diego R. García
50

Repetición condicional

- **Escribir un algoritmo para averiguar la capacidad de un bidón, que sabemos es un número entero de litros. Se dispone únicamente de una botella de medio litro.**

Solución:

- **Vacíó el bidón, y luego voy llenando de a medio litro y contando cuantos “medios litros” puse en el bidón, luego divido esa cantidad por 2 y tengo el resultado.**

Resolución de Problemas y Algoritmos
Dr. Diego R. García
51

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente:
 “Resolución de Problemas y Algoritmos. Notas de Clase”. Diego R. García. Universidad Nacional del Sur. (c) 15/08/2019.

Repetición condicional

- Escribir un algoritmo para averiguar la capacidad de un bidón, que sabemos es un número entero de litros. Se dispone únicamente de una botella de medio litro.

Algoritmo: Capacidad bidón

Si tiene agua el bidón **entonces** vaciar bidón
 Contador de botellas es cero

Repetir hasta que el bidón esté lleno:

Llenar botella de 0.5 litros con canilla

Trasvasar botella 0.5 a bidón

Incrementar contador de botellas en uno

Capacidad es contador de botellas dividido 2

Repetición condicional

- Escribir un algoritmo para averiguar la capacidad de un bidón, que sabemos es un número entero de litros. Se dispone únicamente de una botella de medio litro.

Algoritmo: Capacidad bidón 2

Si tiene agua el bidón **entonces** vaciar bidón
 Contador de litros es cero

Repetir hasta que el bidón esté lleno:

Llenar botella 0.5 con canilla

Trasvasar botella 0.5 a bidón

Incrementar contador de litros en 0.5

Capacidad es contador de litros

Información adicional

Origen del término “algoritmo” (Wikipedia)

Abu Abdallah Muḥammad ibn Mūsā al-Jwārizmī

(أبو عبد الله محمد بن موسى الخوارزمي ابو جعفر)

Conocido como **al-Juarismi**, (al-Khwārizmī)

Persa musulmán chií, que nació en la región de Khwārizmī y vivió en Bagdad entre los años 780 y 850.

Fue matemático, astrónomo y geógrafo.

El término “algoritmo” proviene de la última parte de su nombre.



Algoritmo y álgebra: su origen (Wikipedia)

Su libro “**Kitab al-jabr wa'l muqabala**”:
 “*Compendio de cálculo por restauración y comparación*”,
 pretende enseñar un álgebra aplicada a la resolución de problemas de la vida cotidiana.
 Su libro es un manual, destinado a **«hacer más claro lo que era oscuro y [...] facilitar lo que era difícil»** con el objeto de resolver problemas concretos de cómputo (cómo herencias, medidas de tierra o problemas de comercio).



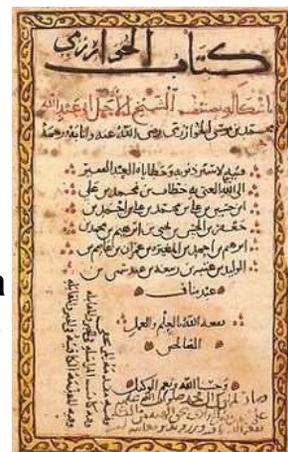
1ra. página del libro

Algoritmo y álgebra: su origen (Wikipedia)

La palabra "**Algebra**" viene de **al-jabr**, una de las dos operaciones usadas en su obra: “**Kitab al-jabr wa'l muqabala**”.

Por ello, **al-Juarismi** es considerado el padre del álgebra y el introductor del sistema de numeración decimal a Persia (luego introducido en Europa en el siglo XII por **Fibonacci**).

...pero otro día hablaremos de **Fibonacci** y del sistema decimal 😊



1ra. página del libro

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente:
 “Resolución de Problemas y Algoritmos. Notas de Clase”. Diego R. García. Universidad Nacional del Sur. (c) 15/08/2019.

Sobre las presentaciones en clase

- El **objetivo** de estas presentaciones es simplemente **agilizar** la clase y permitir al alumno concentrarse en el tema y no estar pendiente de “copiar del pizarrón”
- Estas presentaciones serán impresas y estarán **disponibles** para los alumnos.
- **No reemplazan a la clase.** Son sólo una guía dentro del desarrollo de la clase.
- No deben tomarse como libro o un apunte, y ni siquiera como un resumen de donde estudiar.
- **Hay muchas cosas de la clase que no están aquí: y eso es lo que el alumno debe tomar nota**

Referencias y enlaces

<http://en.wikipedia.org/wiki/Computer>

<http://es.wikipedia.org/wiki/Al-Juarismi>

<http://www.google.com/glass/start/what-it-does/>

<http://www.technologyreview.com/news/515666/contact-lens-computer-like-google-glass-without-the-glasses/>

http://www.nasa.gov/mission_pages/station/main/

<http://store.irobot.com/cleaning-robots/shop.jsp?categoryId=2804605>