PROGRAMACIÓN I

GUÍA DE REVISIÓN

CONCEPTUAL

UNIDAD II

Estructuras Dinámicas I

**1. ¿Qué es un puntero?**

Un puntero es una variable que apunta hacia una dirección de memoria de un elemento

**2. ¿Qué es una función recursiva?**  
Es una función capaz de llamarse así misma

**3. ¿Para qué se usan las funciones recursivas?**  
Es utilizada para descomponer el problema complejo en unos más sencillos. A medida que se van resolviendo los problemas sencillos con la recursividad, el problema complejo se va resolviendo por partes

**4. ¿Qué es una lista?**

Una lista es una estructura de datos lineal que plantea establecer un tipo abstracto de datos, los cuales pueden ser representados como nodos entrelazados entre sí.

**5. ¿Cuántos tipos de listas existen y en qué se diferencian?**

Se diferencian en la forma en la que apuntan al próximo nodo.

Existen 4 tipos de listas

* Lista simplemente enlazada: plantea el primer nodo, el cual puede estar apunando a otro o a nulo
* Lista doblemente enlazada: tiene la capacidad de conocer al nodo anterior y al siguiente
* Lista simplemente enlazada circular: el último nodo apunta al primero
* Lista doblemente enlazada circular: El último nodo apunta a nulo pero el anterior al primero es el último.

**6. ¿Qué operaciones podemos realizar en una lista?**

* Agregar al final
* Agregar al principio
* Agregar en la posición N
* Borrar el último
* Borrar el primero
* Borrar el número de la posición N
* Swap entre el nodo N y el siguiente
* Swap entre el nodo N y el anterior
* Swap entre los nodos 1 y 2
* Cantidad de nodos
* Buscar un nodo con el contenido C
* Ordenar la lista ascendentemente
* Ordenar la lista descendentemente
* Borrar lista

**7. Explique paso a paso lo que debe realizar para agregar un elemento al final de una lista enlazada simple vacía.**

Como la lista está vacía, el elemento se inserta en la primera posición por lo que es el primer y último nodo. Pregunto si la lista está vacía, si es así, lo agrego en el nodo primero.

**8. Explique paso a paso lo que debe realizar para agregar un elemento al principio de una lista enlazada simple vacía.**

Como la lista está vacía, el elemento se inserta en la primera posición por lo que es él primer y último nodo. Pregunto si la lista está vacía, si es así, lo agrego en el nodo primero.

**9. Explique paso a paso lo que debe realizar para agregar un elemento al final de una lista enlazada simple con al menos un elemento.**

Si la lista no se encuentra vacía, el último elemento debe estar apuntado por un Nodo. El siguiente de ese nodo será el nuevo último nodo, por lo que el tamaño de la lista aumenta en 1

**10. Explique paso a paso lo que debe realizar para agregar un elemento al principio de una lista enlazada simple con al menos un elemento.**

Si la lista no se encuentra vacía, el primer elemento debe estar apuntado por un Nodo llamado primero. Creo un temporal para guardar el primer elemento y todo lo que se encuentra enlazado a él. El temporal apuntará al primer nodo, el primer nodo cambiará su valor por el nuevo ingresado. Y el siguiente del nuevo nodo será el temporal

**11. Explique paso a paso lo que debe realizar para borrar un elemento al final de una lista enlazada simple con un elemento.**

Se busca el último nodo ingresado, le pongo un valor null y saco un valor del tamaño de la lista y notifico que está vacía

**12. Explique paso a paso lo que debe realizar para borrar un elemento al principio de una lista enlazada simple con un elemento.**

Se busca el nodo que está apuntado por el centinela primero, lo pongo en null y bajo un valor del tamaño de la lista y notifico que está vacía

**13. Explique paso a paso lo que debe realizar para borrar un elemento al final de una lista enlazada simple con más de un elemento.**

Busco el elemento que está apuntado por el centinela último, lo pongo en null y bajo el valor.

**14. Explique paso a paso lo que debe realizar para borrar un elemento al principio de una lista enlazada simple con más un elemento.**

Busco el elemento que está apuntado por el centinela primero, creo un temporal para guardar todos los elementos que le siguen al primero.

Temporal = primero. Siguiente;

Primero= temporal;

Tamaño--;

**15. Explique paso a paso lo que debe realizar para insertar un elemento en una**

**posición “n” de la una lista enlazada simple.**

Busco el elemento de la posición N. Busco el elemento que está antes de ese número. Creo un temporal para que apunte a elemento que actualmente está en la posición N

Temporal = NodoN;

NodoAnd.Siguiente=NuevoNodo;

NuevoNodo.Siguiente= temporal;

**16. Explique paso a paso lo que debe realizar para realizar un intercambio de un**

**elemento de la lista enlazada simple con el que está a su derecha.**

Busco el nodo anterior al elemento que quiero cambiar.

NodoAnt.siguiente= B;

B.siguiente=A;

**17. Explique paso a paso lo que debe realizar para realizar un intercambio de un**

**elemento de la lista enlazada simple con el que está a su izquierda.**

Busco el nodo dos posiciones atrás del que quiero cambiar

NodoAntAnt.siguiente = B;

B.siguiente = A;

**18. Explique paso a paso lo que debe realizar para realizar conocer el número de elementos que posee una lista enlazada simple.**

La lista tendrá un contador que va a indicar el número de elementos que posee.

**19. Explique paso a paso lo que debe realizar para agregar un elemento al final de una lista doblemente enlazada vacía.**

Pregunto si la lista está vacía, si es así, lo agrego en el nodo primero. La lista está vacía, entonces inserto el elemento en la primera posición por lo que es el primer y último nodo, por este motivo no tiene nodo siguiente ni nodo anterior.

**20. Explique paso a paso lo que debe realizar para agregar un elemento al principio de una lista doblemente enlazada vacía.**

Pregunto si la lista está vacía, si es así, lo agrego en el nodo primero. La lista está vacía, entonces inserto el elemento en la primera posición por lo que es el primer y último nodo, por este motivo no tiene nodo siguiente ni nodo anterior.

**21. Explique paso a paso lo que debe realizar para agregar un elemento al final de una lista doblemente enlazada con al menos un elemento.**

Pregunto si la lista está vacía, como esto no es así, señalo al nuevo nodo como siguiente del primer nodo y señalo al primer nodo como anterior del nuevo nodo, por último, pongo null como nodo siguiente del nuevo nodo.

**22. Explique paso a paso lo que debe realizar para agregar un elemento al principio de una lista doblemente enlazada con al menos un elemento.**

Después de revisar que la no está vacía. Creo un temporal para guardar el primer elemento y todo lo que se encuentra enlazado a él. El temporal apuntará al primer nodo, el primer nodo cambiará su valor por el nuevo ingresado. Y el siguiente del nuevo nodo será el temporal

**23. Explique paso a paso lo que debe realizar para borrar un elemento al final de una lista doblemente enlazada con un elemento.**

Primero se busca el ante último nodo ingresado, y luego le pongo un valor null al siguiente de ese nodo, para finalmente disminuir en 1 el tamaño de la lista.

**24. Explique paso a paso lo que debe realizar para borrar un elemento al principio de una lista doblemente enlazada con un elemento.**

Se busca el nodo que está apuntado por el centinela primero, le pongo un valor null, y disminuyo el tamaño de la lista para avisar que esta está vacía.

**25. Explique paso a paso lo que debe realizar para borrar un elemento al final de una lista doblemente enlazada con más de un elemento.**

Se busca al nodo anterior al último y se cambia el valor de su siguiente por null, para finalmente disminuir el tamaño de la lista.

**26. Explique paso a paso lo que debe realizar para borrar un elemento al principio de una lista doblemente enlazada con más un elemento.**

Primero se busca el nodo que está apuntado por el centinela primero, después le pongo el valor del segundo nodo y al siguiente de primero le pongo el siguiente del segundo nodo, eliminando así el primer nodo, para finalmente disminuir el tamaño de la lista en 1.

**27. Explique paso a paso lo que debe realizar para insertar un elemento en una**

**posición “n” de la una lista doblemente enlazada.**

Busco el elemento de la posición N. Busco el elemento que está antes y después de ese número. Creo un temporal para que apunte a elemento que actualmente está en la posición N y otro temporal a su anterior.

Temporal = NodoN;

NodoAnd.Siguiente=NuevoNodo;

NodoAnd.Anterior= Temporal.Anterior

NuevoNodo.Siguiente= Temporal;

**28. Explique paso a paso lo que debe realizar para realizar un intercambio de un**

**elemento de la lista doblemente enlazada con el que está a su derecha.**

Busco el nodo anterior al elemento que quiero cambiar.

NodoAnt.Siguiente= B;

B.Siguiente = A;

B.Anterior = C

**29. Explique paso a paso lo que debe realizar para realizar un intercambio de un**

**elemento de la lista doblemente enlazada con el que está a su izquierda.**

Busco el nodo dos posiciones atrás del que quiero cambiar

NodoAntAnt.Siguiente= B;

B.Siguiente = A;

B.Anterior = C;

**30. Explique paso a paso lo que debe realizar para realizar conocer el número de elementos que posee una lista doblemente enlazada.**

La lista tendrá un contador que va a indicar el número de elementos que posee.

Aumentando cuando se agrega un valor y disminuyendo cuando se elimina.

**31. Explique paso a paso lo que debe realizar para agregar un elemento al final de una lista enlazada simple circular vacía.**

Cuando está vacía, el primer elemento ingresado también es el último, por lo tanto, su siguiente es null.

**32. Explique paso a paso lo que debe realizar para agregar un elemento al principio de una lista enlazada simple circular vacía.**

Cuando está vacía, el primer elemento ingresado también es el último, por lo tanto, su siguiente es null.

**33. Explique paso a paso lo que debe realizar para agregar un elemento al final de una lista enlazada simple circular con al menos un elemento.**

El último que apunta a null, apuntará al nuevo nodo ingresado. Su siguiente será null

nodoultimo.siguiente = nuevonodo;

Ultimo= nuevonodo;

Nuevonodo.siguiente=nodoprimero;

**34. Explique paso a paso lo que debe realizar para agregar un elemento al principio de una lista enlazada simple circular con al menos un elemento.**

Para agregar al principio debo crear un temporal que almacene el valor del primero.

Temp= primero;

Primero=nuevonodo;

Primero.siguiente= temp;

**35. Explique paso a paso lo que debe realizar para borrar un elemento al final de una**

**lista enlazada simple circular con un elemento.**

Se debe eliminar el valor del último elemento e indicar que la lista se encuentras vacía.

ultimo = null;

size--;

**36. Explique paso a paso lo que debe realizar para borrar un elemento al principio de**

**una lista enlazada simple circular con un elemento.**

Se pone el primer elemento en null y se indica que la lista está vacía

primero=null;

size--;

**37. Explique paso a paso lo que debe realizar para borrar un elemento al final de una**

**lista enlazada simple circular con más de un elemento.**

anteultimo.siguiente=primero;

size--;

**38. Explique paso a paso lo que debe realizar para borrar un elemento al principio de**

**una lista enlazada simple circular con más un elemento.**

primero=primero.siguiente;

size--;

**39. Explique paso a paso lo que debe realizar para insertar un elemento en una**

**posición “n” de la una lista enlazada simple circular.**

Busco la posición N y su anterior.

Nanterior.siguiente=nuevonodo;

Nuevonodo.siguiente= viejonodo;

**40. Explique paso a paso lo que debe realizar para realizar un intercambio de un**

**elemento de la lista enlazada simple circular con el que está a su derecha.**

Busco el elemento que quiero cambiar y su anterior, al igual que el posterior del nodo a su derecha.

Temp= nododerecha.siguiente;

Nanterior.siguiente= nododerecha;

Nododerecha.siguiente= nodocambio;

Nodocambio.siguiente=temp;

**41. Explique paso a paso lo que debe realizar para realizar un intercambio de un**

**elemento de la lista enlazada simple circular con el que está a su izquierda.**

**Busco el elemento que quiero cambiar y su posterior, al igual que el anterior del nodo a su izquierda.**

Temp= nodocambio.siguiente;

Nizqant.siguiente= nodocambio;

Nodocambio.siguiente= Nizq;

Nizq.siguiente=temp;

**42. Explique paso a paso lo que debe realizar para realizar conocer el número de**

**elementos que posee una lista enlazada simple circular.**

La lista tendrá un contador que va a indicar el número de elementos que posee.

Aumentando cuando se agrega un valor y disminuyendo cuando se elimina.

**43. Explique paso a paso lo que debe realizar para agregar un elemento al final de una lista doblemente enlazada circular vacía.**

Cuando está vacía, el primer elemento ingresado también es el último, no tiene un anterior y su siguiente es null

**44. Explique paso a paso lo que debe realizar para agregar un elemento al principio de una lista doblemente enlazada circular vacía.**

Cuando está vacía, el primer elemento ingresado también es el último, no tiene un anterior y su siguiente es null

**45. Explique paso a paso lo que debe realizar para agregar un elemento al final de una lista doblemente enlazada circular con al menos un elemento.**

El último que apunta a null, apuntará al nuevo nodo ingresado y éste pasará a ser el nuevo último. Su siguiente será null

nodoultimo.siguiente = nuevonodo;

Ultimo= nuevonodo;

Nuevonodo.siguiente=null;

**46. Explique paso a paso lo que debe realizar para agregar un elemento al principio de una lista doblemente enlazada circular con al menos un elemento.**

Para agregar al principio debo crear un temporal que almacene el valor del primero.

Temp= primero;

Primero=nuevonodo;

Primero.siguiente= temp;

Primero.anterior= ultimo;

**47. Explique paso a paso lo que debe realizar para borrar un elemento al final de una**

**lista doblemente enlazada circular con un elemento.**

Se debe eliminar el valor del último elemento e indicar que la lista se encuentras vacía.

ultimo = null;

size--;

**48. Explique paso a paso lo que debe realizar para borrar un elemento al principio de**

**una lista doblemente enlazada circular con un elemento.**

Se pone el primer elemento en null y se indica que la lista está vacía

primero=null;

size--;

**49. Explique paso a paso lo que debe realizar para borrar un elemento al final de una**

**lista doblemente enlazada circular con más de un elemento.**

primero.anterior=ultimo.anterior;

ultimo=null;

size--;

**50. Explique paso a paso lo que debe realizar para borrar un elemento al principio de**

**una lista doblemente enlazada circular con más un elemento**.

primero=primero.siguiente;

primero.anterior=ultimo;

size--;

**51. Explique paso a paso lo que debe realizar para insertar un elemento en una**

**posición “n” de la una lista doblemente enlazada circular.**

Busco la posición N y su anterior.

Nanterior.siguiente=nuevonodo;

Nuevonodo.siguiente= viejonodo;

**52. Explique paso a paso lo que debe realizar para realizar un intercambio de un**

**elemento de la lista doblemente enlazada circular con el que está a su derecha.**

Busco el elemento que quiero cambiar y su anterior, al igual que el posterior del nodo a su derecha.

Temp= nododerecha.siguiente;

Nanterior.siguiente= nododerecha;

Nododerecha.siguiente= nodocambio;

Nodocambio.siguiente=temp;

**53. Explique paso a paso lo que debe realizar para realizar un intercambio de un**

**elemento de la lista doblemente enlazada circular con el que está a su izquierda.**

Busco el elemento que quiero cambiar y su posterior, al igual que el anterior del nodo a su izquierda.

Temp= nodocambio.siguiente;

Nizqant.siguiente= nodocambio;

Nodocambio.siguiente= Nizq;

Nizq.siguiente=temp;

**54. Explique paso a paso lo que debe realizar para realizar conocer el número de**

**elementos que posee una lista doblemente enlazada circular.**

Se imprimirá el contador que leva el recuento de todos los elementos de la lista

**55. ¿Qué es una pila?**

Colección ordenada de elementos en la cual los datos se insertan o se retiran por el mismo extremo superior.

**56. ¿Qué operaciones debe tener una pila?**

Las operaciones básicas son Push(agregar) y Pop(sacar)

**57. Explique paso a paso como se realiza la operación de apilar cuando la pila está**

**vacía y cuando no lo está.**

Primero se debe preguntar si la pila se encuentra vacía. En caso de que sea así, el valor de la pila se ingresará en la primera posición y el puntero cima apuntará a él. En caso de que la pila no esté vacía, el dato se ingresará en la posición de la cima+1 y ahora la cima apuntará a él.

**58. Explique paso a paso como se realiza la operación de desapilar cuando la pila está vacía y cuando no lo está.**

Cuando la pila se encuentra vacía, no se puede desapilar ningún elemento, por lo que debe aparecer un cartel que indique esto. Cuando la pila posee elementos, el elemento que está en la cima se debe ir y cima desciende una posición.

**59. Explique paso a pasa como se realiza la operación de mirar el elemento a desapilar cuando la pila está vacía y cuando no lo está.**

Cuando la pila está vacía, se debe indicar que la pila no posee elementos. Cuando posee elementos, se debe mostrar al elemento que apunta la cima

**60. ¿Qué es una cola?**

La cola es una estructura de datos que posee dos extremos, uno es la parte delantera por donde se retirarán los elementos, y otro es la parte trasera que es por donde se ingresarán los elementos

**61. ¿Qué operaciones debe tener una cola?**

Las operaciones que debe tener una cola es Enqueue (agregar elemento) y Dequeue

(eliminar el primer elemento de la cola)

**62. Explique paso a paso como se realiza la operación de encolar cuando la pila está**

**vacía y cuando no lo está.**

Cuando la cola está vacía, el primer elemento será almacenado en la primera posición de la cola y a la vez será el último elemento.

Cuando la cola no está vacía:

Ultimo.siguiente= nuevonodo;

Ultimo= nuevonodo;

Cola++;

**63. Explique paso a paso como se realiza la operación de desencolar cuando la pila**

**está vacía y cuando no lo está.**

El elemento que se encuentra en la primera posición será el primero en salir.

Primero=primero.siguiente;

Cola--;

**64. Explique paso a pasa como se realiza la operación de mirar el elemento a**

**desencolar cuando la pila está vacía y cuando no lo está.**

Cuando la cola está vacía, se notificará que se encuentra vacía

Cuando no está vacía, se muestra el primer elemento de la cola

**65. Explique las diferencias entre una pila y una cola.**

La principal diferencia que existe entre ambas es la forma de ingreso y salida.

En las pilas, los primeros elementos ingresados son los últimos en salir, en cambio en las colas, lo primero que entra es lo primero en salir

**66. Enumere y explique las propiedades, métodos y eventos más importantes del tipo Stack.**

|  |  |
| --- | --- |
| **Propiedades** | |
| Count | Obtiene el número de elementos contenidos en la pila |
| IsSynchronized | Obtiene un valor que indica si el acceso a la pila está sincronizado |
| SyncRoot | Obtiene un objeto que se puede usar para sincronizar el acceso a la pila |
| **Métodos** | |
| Clear() | Elimina todos los objetos de la pila |
| Clone() | Crea una copia superficial de la pila |
| Peek() | Devuelve el objeto en la parte superior de la pila sin quitarlo |
| Pop() | Elimina y devuelve el objeto en la parte superior de la pila |
| Push(Object) | Inserta un objeto en la parte superior de la pila |
| Synchronized(Stack) | Devuelve un contenedor sincronizado para la pila |
| ToArray() | Copia la pila a una nueva matriz. |
| ToString() | Devuelve una cadena que representa el objeto actual. |

**67. Enumere y explique las propiedades, métodos y eventos más importantes del tipo Queue.**

|  |  |
| --- | --- |
| **Propiedades** | |
| Count | Devuelve la cantidad total de elementos de la Cola |
| **Métodos** | |
| Enqueue() | Añade los elementos a la Cola |
| Dequeue() | Remueve y retorna un elemento del principio del enqueue |
| Peek() | Devuelve el primer elemento de la Cola |
| Clear() | Remueve todos los elementos de la Cola |

GUÍA DE TRABAJOS PRÁCTICOS

UNIDAD II

Estructuras Dinámicas I

1. Ingresar un número base 10 y mostrar su equivalente en binario usando una

función recursiva. (pista: usar divisiones sucesivas)

2. Ingresar un número en cualquier base y mostrar su equivalente en binario usando

una función recursiva.

3. Determinar la potencia de un número con funciones recursivas.

4. Determinar la suma de los primeros "n" números naturales usando una función

recursiva.

5. Determinar los n primeros números de la serie de Fibonacci usando una función

recursiva.

6. Determinar los números de la serie de Fibonacci menores que el valor n usando una

función recursiva.

7. Desarrolle un programa que permita construir una lista simplemente enlazada

sobre la cual se puedan realizar las siguientes operaciones: agregar un elemento al

final de la lista, agregar un elemento al principio de la lista, borrar un elemento al

final de la lista, borrar un elemento al principio de la lista, agregar un elemento en

la posición “n”, borrar un elemento de la posición “n”, intercambiar un elemento

con el de su derecha, intercambiar un elemento con el de su izquierda, conocer

cuantos elementos posee la lista.

8. Ídem al anterior, pero para una lista doblemente enlazada.

9. Ídem al anterior, pero para una lista simple circular.

10. Ídem al anterior, pero para una lista doblemente enlazada circular.

11. Desarrollar un programa que permita administrar una lista de personas. Cada

persona posee nombre, apellido, edad, peso y sexo. Por cada persona que ingrese

o egrese de la lista se desea conocer: promedio de edad de todas las personas de la

lista, promedio de peso de todas las personas de la lista, la edad de la persona más

anciana, la edad de la persona más joven, el peso de la persona más delgada, el

peso de la persona más obesa, cuantas personas hay en la lista, cuantas personas

poseen más de 50 años y cuantas personas pesan más de 60 Kilogramos. Los datos

de cada persona de la lista, así como los resultados requeridos se deben visualizar

en controles seleccionados por Ud.

12. Un banco tiene guardados los registros de los movimientos en una lista. Los

movimientos están ordenados primero por fecha y luego por número de cuenta.

El tipo del elemento de la lista es: TElemLista

La estructura es la siguiente:

Fecha (Clave ordenamiento 1)

Nro\_Cta (Clave ordenamiento 2)

Monto

Tipo (Deposito/Extracción)

Realizar procedimientos para este TDA que:

Permita Calcular la cantidad de depósitos realizados entre el 01/01/20014 y el

31/07/2016.

Permita Calcular el total depositado y extraído en el año 2015.

Permita Calcular el saldo de la cuenta 8894 del año 2013.

13. Tomando como base el ejercicio anterior permitir ingresar los datos desordenados

y generar los algoritmos que permiten ordenar la lista.

14. Una empresa de servicios guarda en una lista las tareas que debe realizar cada

empleado.

El tipo del elemento de la lista es: TelemLista

La estructura de la lista es:

Código Empleado (Clave ordenamiento)

Cantidad de Tareas

Realizar procedimientos para este TDA que:

Permita ingresar una nueva tarea en el empleado que tenga menos tareas.

Permita reducir una tarea al empleado de un código determinado.

15. Realizar un programa que constituya una pila e implemente las acciones de apilar,

desapilar y mirar el elemento a desapilar.

16. Realizar un programa que utilizando pilas simule el juego de las torres de hanoi.

17. Realizar un programa que utilizando pilas logre invertir un número ingresado por

el usuario.

18. Realizar un programa que utilizando pilas logre invertir una palabra ingresada por

el usuario.

19. Desarrollar los siguientes algoritmos, utilizando sólo las operaciones primitivas de

pila:

a. Imprimir el contenido de una pila de enteros sin cambiar su contenido.

b. Colocar en el fondo de una pila un nuevo elemento.

c. Calcular el número de elementos de una pila sin modificar su contenido.

d. Eliminar de una pila todas las ocurrencias de un elemento dado.

e. Intercambiar los valores del tope y el fondo de una pila.

f. Duplicar el contenido de una pila.

g. Verificar si el contenido de una pila de caracteres es un palíndromo.

h. Calcular la suma de una pila de enteros sin modificar su contenido.

i. Calcular el máximo de una pila de números reales.

20. Escribir una función que reciba una pila de enteros y retorne dos pilas una con los

números pares y otra con los impares.

21. Realizar un programa que constituya una cola e implemente las acciones de encolar, desencolar y mirar el elemento a desencolar.

22. Realizar un programa que simule la cola de clientes por cobrar en una caja de

supermercado. Los clientes poseen Id y el importe que han comprado. Se desea

poder encolar clientes y desencolar los clientes a los que se les va cobrando. En todo

momento se desea saber cuánto dinero hay por cobrar, cuánto dinero se cobró,

cuantos clientes aún no se les ha cobrado, cuantos clientes ya se les ha cobrado.

23. Desarrollar un programa que simule la creación de trabajos a ser impresos. Los

trabajos a ser impresos poseen id y una cantidad de páginas. La cantidad de páginas

debe generarse aleatoriamente. La generación de trabajos debe realizarse con un

temporizador. Si los trabajos poseen un número de páginas pares deben encolarse

en la impresora1 y si el número de páginas es impar deben encolarse en la

impresora2. En todo momento se desea conocer cuántos trabajos aún no han sido

impresos y se encuentran en cada cola impresora y cuantos han sido impresos por

cada impresora. También cuanto tiempo ha estado imprimiendo cada impresora y

cuanto tiempo le falta por imprimir considerando los trabajos que están encolados.

Un temporizador en cada impresora es el encargado de simular que una vez

transcurrido el tiempo el trabajo ha sido impreso.