

Benemérita Universidad Autónoma de Puebla  
Facultad de Ciencias de la Computación  
**Programación Orientada a Objetos II**  
**Proyecto Final**

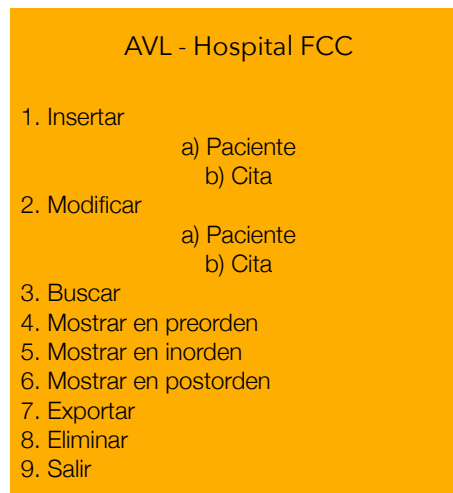
**Opción I: Árboles equilibrados (AVL) y listas ligadas simples.**

Objetivo: Realiza la implementación de un árbol equilibrado AVL que permita almacenar información acerca de los pacientes de un hospital.

Los datos que se requiere almacenar para cada paciente son:

- NSS: Número de seguridad social. (Este número define el orden para el AVL)
- Nombre
- Apellido paterno
- Apellido materno
- Sexo
- Edad
- Citas - Para la implementación de esta sección se deberá implementar una lista ligada ordenada por fecha, que contendrá la información de las citas del paciente. La información para cada cita es:
  - Fecha: Factor de ordenamiento de la lista
  - Hora
  - Nombre del médico
  - No. consultorio

Para verificar la implementación correcta de las clases se debe mostrar el siguiente menú (puedes modificar la distribución):



Descripción del funcionamiento de los métodos:

1. Insertar

- a) Paciente: Inserta la información básica de un nuevo paciente en el AVL, en esta opción es posible darlo de alta con o sin citas.

- b) Cita: Inserta una cita para un paciente ya existente en el AVL.
2. Modificar:
- a) Datos de un paciente: Permite modificar la información de un paciente ya existente en el AVL.
- b) Cita: Permite modificar una cita de algún paciente ya existente el el AVL.
3. Buscar: Dado el NSS de un paciente muestra su información personal junto con su lista de citas.
- 4, 5 y 6. Muestra toda la información contenida en el AVL (Información personal de cada paciente junto con sus citas), en el orden seleccionado.
7. Exportar: Guarda en un archivo .txt toda la información contenida en el AVL (Información personal de cada paciente junto con sus citas), en el orden que prefieras (preorden, inorden o postorden).
8. Eliminar: Dado el NSS de un paciente elimina su información del AVL.
9. Salir: Finaliza el programa.
- Antes de la revisión del programa, se deberá almacenar (de forma manual o automática) la siguiente información en el AVL:

NSS	99092013		
Nombre	Karla Ramírez Palacios	Sexo: F	Edad: 21
Citas			
20-10-2017	22-10-17		23-10-17
8:00 hrs Dr. Juárez Consultorio 1	10:00 hrs Dr. García Consultorio 11		16:00 hrs Dr. Cárdenas Consultorio 2

NSS	31051985			
Nombre	José Hernández Meza	Sexo: M	Edad: 41	
Citas				
20-11-2017	21-11-17	22-11-17	23-11-17	24-12-17
8:00 hrs Dr. Zarate Consultorio 20	9:00 hrs Dr. Corona Consultorio 6	8:00 hrs Dr. Zarate Consultorio 20	9:00 hrs Dr. Corona Consultorio 6	8:00 hrs Dr. Zarate Consultorio 20

NSS	15092017		
Nombre	Luis Robles Anzurez	Sexo: M	Edad: 56
Citas			
20-06-2017	21-07-17	22-08-17	23-09-17
10:00 hrs Dr. Pérez Consultorio 21	7:00 hrs Dr. Tzonpantzi Consultorio 10	9:00 hrs Dr. Pérez Consultorio 21	7:00 hrs Dr. Andrade Consultorio 3

NSS	16091989		
Nombre	Tania Romero Cuamatzi	Sexo: F	Edad: 23

Citas		
7-12-2017	22-12-17	23-1-18
8:00 hrs Dr. Juárez Consultorio 1	9:00 hrs Dr. Tzonpantzi Consultorio 10	8:00 hrs Dr. Juárez Consultorio 1

NSS	16062017		
Nombre	Diana Gómez Arroyo	Sexo: F	Edad: 18
Citas			
20-12-2017	21-01-18	22-02-18	23-03-18
7:00 hrs Dr. Zarate Consultorio 20	17:00 hrs Dr. Corona Consultorio 6	14:00 hrs Dr. Corona Consultorio 6	9:00 hrs Dr. Zarate Consultorio 20

El programa se evaluará mediante los siguientes criterios:

Característica del programa	Puntuación	Puntuación obtenida
1. Compila sin ningún error.	0.5	
2. Contiene una clase Nodo para el AVL	0.5	
3. Contiene una clase Nodo para la Lista Ligada	0.5	
4. El método insertar funciona de forma correcta, manteniendo el orden de los nodos y verificando si un elemento está duplicado.	3	
5.El programa permite modificar la información de los pacientes y sus citas.	2	
6. El programa permite realizar la búsqueda de información de un paciente, mostrando la información en pantalla.	0.5	
7. El programa exporta en un archivo la información del AVL en algún orden (preorden, inorden o postorden).	1	
9. El programa elimina de forma correcta la información de un paciente.	2	

## Opción II: Interfaz gráfica de usuario - Listas ligadas simples

Objetivo: Realizar una simulación del juego del 15 (Taken)

El **juego del 15 o taken** es una cajita formada por 16 casillas de las cuáles sólo quince están ocupadas. Todas las fichas están colocadas en orden numérico, excepto la 14 y la 15, que tienen sus posiciones intercambiadas. El juego consiste en maniobrar todas las fichas para corregir el error que hay en la fila inferior de la cajita, de manera que todas las fichas queden en orden consecutivo.

Fuente: [https://es.wikipedia.org/wiki/Juego\\_del\\_15](https://es.wikipedia.org/wiki/Juego_del_15)

Se sugiere realizar clase Taken que represente la parte lógica del juego:

Taken
+ lista[] : String
+ movimientos: int
+Taken()
+ verificarTablero()
+ moverTecla()
+ obtenerMovimientos()
+ reiniciar()

Eres totalmente libre de agregar o modificar atributos y métodos de los mostrados en el diagrama anterior.

Cuando finalices el diseño y programación de la lógica del juego, inicia la vinculación con los componentes de la interfaz.

El juego debe contener o permitir:

- **Título del juego.**
- **Desactivar tablero:** Los botones del tablero se desactivan cuando el usuario ha logrado alguna de las disposiciones permitidas para ganar el juego.
- **Alerta:** El programa debe detectar cuando el usuario ha ganado. De ser así, muestra un mensaje en pantalla y desactiva el tablero.
- **Reiniciar:** Devuelve la disposición inicial de todos los botones para iniciar un nuevo juego, el número de movimientos regresa a cero. **La disposición inicial siempre es en Horizontal.**

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

Horizontal

- **Mostrar disposiciones:** Muestra las diferentes disposiciones de los botones para ganar un juego, éstas son:

7	8	9	10
6	1	2	11
5	4	3	12
	15	14	13

**Espiral**

15	14	13	12
11	10	9	8
7	6	5	4
3	2	1	

**Imposible**

1	2	3	4
12	13	14	5
11		15	6
10	9	8	7

**Periferico**

1	5	9	13
2	6	10	14
3	7	11	15
4	8	12	

**Vertical**

El programa se evaluará bajo los siguientes criterios:

Característica del programa	Puntuación	Puntuación obtenida
1. La aplicación compila sin ningún error.	1	
2. Se muestran las diferentes disposiciones para ganar el juego.	1	
3. Se muestra un mensaje cuando el usuario ha ganado.	1	
4. Se permite reiniciar el juego.	1	
5. El diseño y colores del juego son estéticos.	2	
6. Se usa una lista ligada para el almacenamiento de los números del tablero.	2	
7. Contiene una clase Taken con la parte lógica del juego	2	