

TAREA #4
“CARRERA BUAP 2025”
PROGRAMACION AVANZADA (POO) 2025
FACULTAD DE CIENCIAS DE LA COMPUTACION
M. C. GRACIANO CRUZ ALMANZA

Hacer un programa en Java para el registro de asistentes a la Carrera BUAP 2025. El uso del sistema está protegido y permite el ingreso a varios usuarios: el usuario “Lola” con clave “1234”; “Emma” con clave “123456”; “Laura” con clave “a1b2c3”; pueden ser más a futuro. La ventana inicial es:

<h2>CARRERA BUAP 2025</h2>		
<div style="border: 1px solid black; width: 100px; height: 100px; margin: 0 auto; display: flex; flex-direction: column; align-items: center; justify-content: center;"><div>ESCUDO</div><div>BUAP</div></div>	Usuario:	<input style="width: 150px;" type="text"/>
	Clave:	<input style="width: 150px;" type="password"/>
<div style="display: flex; justify-content: space-around; margin-top: 20px;"><div style="border: 1px solid black; padding: 5px 20px;">INGRESAR</div><div style="border: 1px solid black; padding: 5px 20px;">SALIR</div></div>		

La carrera es un evento de tres competencias al que asisten muchas personas (hombres, mujeres y niños), se debe registrar y administrar la información. Se requiere un menú general para las tres competencias con este formato:

<h2>CARRERA BUAP 2025</h2> <h3>MENU GENERAL</h3>		
<div style="border: 1px solid black; width: 150px; height: 100px; margin: 0 auto; display: flex; flex-direction: column; align-items: center; justify-content: center;"><div>FOTO</div><div>DE CORREDORES</div></div>	<div style="display: flex; flex-direction: column; align-items: center; margin: 0 auto;"><div style="border: 1px solid black; padding: 5px 20px; margin-bottom: 5px;">HOMBRES</div><div style="border: 1px solid black; padding: 5px 20px; margin-bottom: 5px;">MUJERES</div><div style="border: 1px solid black; padding: 5px 20px;">NIÑOS</div></div>	<div style="border: 1px solid black; width: 150px; height: 100px; margin: 0 auto; display: flex; flex-direction: column; align-items: center; justify-content: center;"><div>FOTO</div><div>BUAP</div></div>
<div style="display: flex; justify-content: space-around; margin-top: 20px;"><div style="border: 1px solid black; padding: 5px 20px;">AUTOR</div><div style="border: 1px solid black; padding: 5px 20px;">SALIR</div></div>		

Al seleccionar el botón de hombres debe aparecer la siguiente ventana y ocultar la anterior. Esta ventana registra los asistentes hombres con el nombre, el apellido paterno, apellido materno, edad y facultad de procedencia:

CARRERA BUAP 2025		
REGISTRO DE ASISTENTES HOMBRES		
FOTO DE CORREDOR HOMBRE	Nombre:	<input type="text"/>
	Paterno:	<input type="text"/>
	Materno:	<input type="text"/>
	Edad:	<input type="text"/>
	Facultad:	<input type="text"/>
<div>REGISTRARLISTAR 1LISTAR 2BUSCARREGRESAR</div>		

Al seleccionar el botón de mujeres debe aparecer la siguiente ventana y ocultar la anterior. Esta ventana registra los asistentes mujeres con el nombre, el apellido paterno, apellido materno, edad y facultad de procedencia:

CARRERA BUAP 2025		
REGISTRO DE ASISTENTES MUJERES		
FOTO DE CORREDOR MUJER	Nombre:	<input type="text"/>
	Paterno:	<input type="text"/>
	Materno:	<input type="text"/>
	Edad:	<input type="text"/>
	Facultad:	<input type="text"/>
<div>REGISTRARLISTAR 1LISTAR 2BUSCARREGRESAR</div>		

Al seleccionar el botón de niños debe aparecer la siguiente ventana y ocultar la anterior. Esta ventana registra los asistentes niños con el nombre, el apellido paterno, apellido materno, edad, nombre del tutor, teléfono de contacto y escuela del niño:

CARRERA BUAP 2025		
REGISTRO DE ASISTENTES NIÑOS		
FOTO DE CORREDOR NIÑO	Nombre:	<input type="text"/>
	Paterno:	<input type="text"/>
	Materno:	<input type="text"/>
	Edad:	<input type="text"/>
	Tutor:	<input type="text"/>
	Teléfono:	<input type="text"/>
	Escuela:	<input type="text"/>
<div>REGISTRAR</div> <div>LISTAR 1</div> <div>LISTAR 2</div> <div>BUSCAR</div> <div>REGRESAR</div>		

COMENTARIOS SOBRE EL DISEÑO:

- Las ventanas deben ser del tamaño de toda la pantalla.
- La ventana inicial debe tener como fondo una foto de CU2.
- El tipo de letra, tamaño y los colores se deja a decisión del diseñador, no usar tipos básicos.
- Como se puede apreciar hay tres secciones de las ventanas y deben usarse dos colores que combinen. El texto de los botones debe ser negro.
- El tipo de letra de los encabezados debe ser legible y estética.
- El tamaño de las fotos debe ser adecuado a los espacios.
- No debe mostrar ninguna información al inicio e informar al usuario “NO HAY DATOS” en caso de querer un listado.
- Se solicita que cuando el cursor pase sobre los botones muestre un cursor diferente para ayudar al capturista (mano).

FUNCIONALIDAD:

- El botón registrar debe almacenar la información del asistente para posibles búsquedas y listados. Antes de almacenar debe preguntar (verificar) si quiere registrar con una ventana de dialogo. Esto evita errores de registro. Borre la basura.
- Al registrar un asistente, se le asigna un numero consecutivo iniciando con (H-0001, M-0001, N-0001) es el identificador en la carrera.
- El botón LISTAR 1 debe mostrar todos los datos de asistentes, uno por uno, en ventanas informativas (incluido el número de corredor).
- El botón LISTAR 2 debe mostrar (compacto) todos los datos de asistentes en la consola (simula la impresora). Pero antes debe preguntar si los quiere como están registrados, ordenados por apellido paterno u ordenados por nombre. El ordenar para imprimir no debe modificar el registro original.
- El botón buscar, pregunta al usuario si la búsqueda es por nombre o por apellido, y lo busca. Como resultado puede informar que SI ESTÁ REGISTRADO con una ventana informativa e imprimir los datos en la misma ventana o NO ESTÁ REGISTRADO.
- El botón regresar, regresa a la ventana anterior (menú general).
- El botón Autor debe mostrar: nombre completo, curso, matrícula y número de lista. Debe usar una ventana informativa.
- Todos los botones salir deben ser verificados para evitar accidentes.

CARACTERISTICAS GENERALES:

- Hacerse individualmente, se hará una verificación del código.
- Programarse en Java.
- Usar la clase Persona previamente hecha por su antecesor que se anexa.
- Las clases deben estar en archivos diferentes.
- Usar Relaciones de Herencia y Asociación.
- Para ordenar use la función SORT de ArrayList.
- Usar encapsulado (datos privados y funciones públicas) y constructores.
- Entregar el diagrama de clases impreso con los datos del autor. Debe contener Nombre, Atributos y Métodos usados en la aplicación (sin getters / setters), así como la accesibilidad. No olvidar dibujar las relaciones con toda la información (dirección, cardinalidad y oración conectora).

NOTA: Se debe tener la autorización previa del profesor sobre el diagrama de clases en un periodo de cinco días (obligatorio).