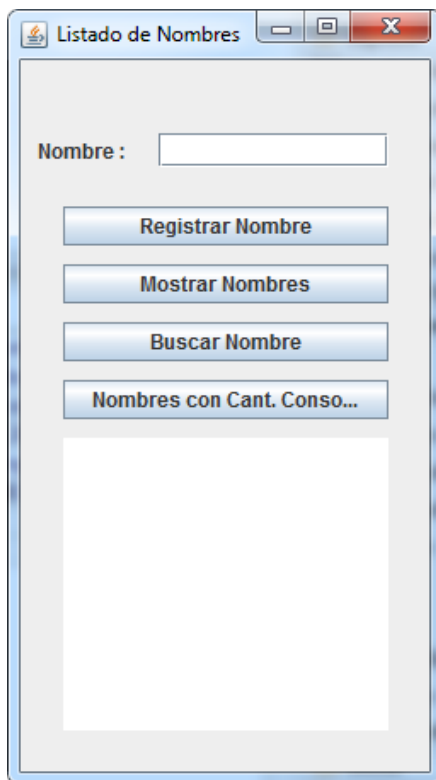


Algoritmo y Estructura de Datos I

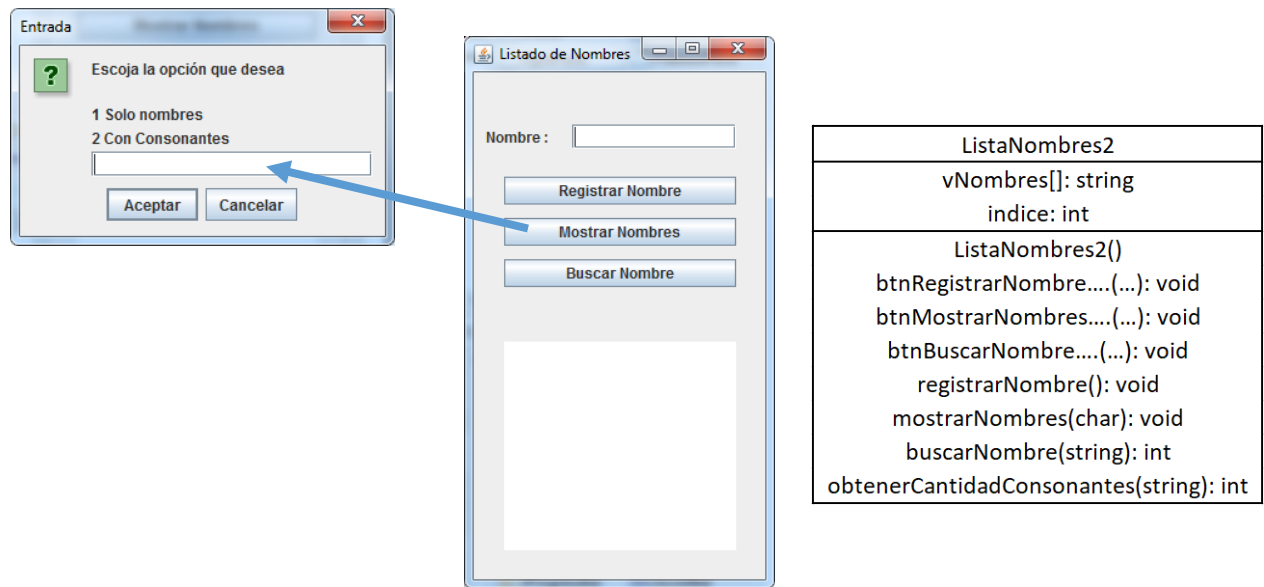
1. Crear un programa con un vector de tamaño 10, que permita registrar nombres.
 - Se debe validar que el vector no esté lleno.
 - Se debe validar que el nombre no sea una cadena vacía.
 - El botón *Mostrar Nombres*, mostrará una lista con los nombres ingresados.
 - El botón *Buscar Nombre*, solicitará mediante una caja de dialogo el nombre que desea buscar en el vector. Si encuentra el nombre mostrará la posición en que se encuentra el dato, caso contrario mostrará en una caja de dialogo el mensaje "Nombre no encontrado".
 - El botón *Nombres con Cant. Consonantes*, mostrará un listado de los nombres registrados con la correspondiente cantidad de consonantes. Ej. Luisa tiene 2 consonantes.
 - Tener presente los métodos de los botones.
 - Respetar el diagrama de clase.
 - Tener presente el diseño de pantalla.



ListaNombres
vNombres[]: string indice: int
ListaNombres() btnRegistrarNombre....(...): void btnMostrarNombres....(...): void btnBuscarNombre....(...): void btnNomConsonantes....(...): void registrarNombre(): void mostrarNombres(): void buscarNombre(string): int mostrarNombresConsonantes(): void obtenerCantidadConsonantes(string): int

Algoritmo y Estructura de Datos I

2. Se pide lo mismo que el problema 1, pero teniendo en cuenta el nuevo diseño de pantalla y el nuevo diagrama de clase. Tener presente que el botón *Mostrar Nombres* debe mostrar una caja de dialogo que permita escoger una de las opciones para el listado de nombres.



3. Desarrollar una solución que permita ingresar y almacenar los promedios finales de los 20 alumnos matriculados en el curso de Algoritmo y Estructura de Datos I.

Posteriormente calcular y mostrar el promedio del aula, redondeado a 2 decimales.

Tener en cuenta:

- Al registrar los promedios de los alumnos, se debe validar si el vector está lleno o no. De estar lleno mostrar el mensaje "Vector Lleno" en una caja de dialogo.
- Se debe validar que los promedios de los alumnos estén en el rango de 0 a 20, caso contrario mostrar el mensaje "Promedio No Valido" en una caja de dialogo.
- Los promedios de los alumnos pueden contener dos decimales. Asumir que solo se ingresan números.
- El botón *Listar Promedios* debe mostrar un listado numerado con los promedios registrados en el vector. De no existir promedios en el vector mostrar el mensaje "No hay datos para listar" en una caja de dialogo.
- No se puede calcular el promedio del aula si no se han ingresado todos los promedios. Mostrar el mensaje correspondiente en una caja de dialogo.
- Tener presente el diagrama de clase y el diseño de pantalla.

Algoritmo y Estructura de Datos I

PromedioAula
vPromedios[]: int indice: int
PromedioAula() btnRegistrar....(): void btnListarNotas....(): void btnCalcularPromedioAula....(): void validarPromedioAlumno(double): char calcularPromedio(): double redondearDecimales(double): double

Promedio del Alumno: 18.17 Registrar

Listar Promedi... Calcular Promedio del Aula

1 13.25
2 8.9
3 11.63
4 18.17

4. Desarrollar una solución que permita ingresar y almacenar el sueldo de los 15 trabajadores de una empresa, calcular los nuevos sueldos y eliminar sueldos bajo un criterio, teniendo en cuenta lo siguiente:

- Validar si el vector está lleno.
- No puede registrarse sueldos menores a 600 soles.
- El botón *listar* debe mostrar los sueldos registrados en el vector. Puede usarse en cualquier momento.
- El botón *Aplicar Aumento*:
 - Debe aplicar el porcentaje de aumento correspondiente según la tabla y almacenar los nuevos sueldos.

Sueldo	Porcentaje de aumento
Menor o igual a 975	25%
Mayor a 975 y menor igual 2500	15%
Mayor a 2500	5%

- Puede realizar los cálculos para los sueldos registrados hasta un momento dado. Si se registran nuevos sueldos y se vuelve hacer clic al botón, el cálculo no debe realizarse para los sueldos anteriores ya calculados.
 - Debe mostrar el monto total de la planilla solo si el vector está lleno.
- El botón *eliminar sueldos* debe aplicar la eliminación de todos los sueldos mayores a 3500 soles (se puede presionar en cualquier momento)
- Los nuevos sueldos deben estar redondeados a 2 decimales.
- Mostrar los mensajes adecuados en las validaciones.
- Se debe respetar el diagrama de clase y el diseño de pantalla.

Algoritmo y Estructura de Datos I

AumentoSueldo
vSueldos[]: double
indice: int
procesados: int
AumentoSueldo()
btnRegistrar....(...): void
btnListar....(...): void
btnAplicarAumento....(...): void
btnEliminacion....(...): void
validarSueldo(double): int
obtenerPorcentaje(double): double

5. Se pide registrar los nombres y apellidos, y fecha de nacimiento de 20 ingresantes a la FIA. (validar que los vectores no estén llenos)

También, se debe generar el código de estudiante que será de 10 dígitos, el cual está conformado por el semestre de ingreso (20222) concatenado con un valor aleatorio en el rango de 50 a 4500. Este valor aleatorio debe ser completado con ceros a la izquierda para contar con la cantidad de 10 caracteres requeridos. Se debe validar que no se repita el código de estudiante.

El botón *Listar* mostrará los datos existentes en el vector, el botón *ordenar*, ordenará los datos teniendo en cuenta el código de alumno de manera descendente y el botón *eliminar* eliminará los datos de aquellos alumnos cuyos códigos tengan sus últimos tres dígitos con cero. Tener en cuenta que todos los botones se pueden presionar en cualquier momento.

Ingresante
vNombres[]: string
vFecNac[]: string
vCodigo[]: string
indice: int
Ingresante()
btnRegistrar....(...): void
btnListar....(...): void
btnAplicarOrdenar....(...): void
btnEliminar....(...): void
generarCodigo(): string
completarCeros(string): string