

# Trabajo final Gestión de Alumnos



## Alumnos:

- Erica Nathalia Nasra
- Nelio Basan
- Bulich Canelo Agustina
- Mansilla Sara
- Fernando Iglesias
- Fernando Serrano
- Miguel Marchione
- Evelyn pini
- Sergio Peluffo
- Mauro Cardin

## Profesora:

Mabel Escobar Korajay







El código proporcionado muestra un programa en Java que realiza diferentes operaciones relacionadas con el registro de alumnos, materias, notas y profesores. A continuación, se detalla el paso a paso del programa:

- Se importa la clase "java.util." para utilizar las clases Scanner, HashSet, HashMap y ArrayList. Se define la clase Main.
- Se declara una variable de instancia llamada "nombre" de tipo String.
- Se define el método principal "main" con la declaración "throws InputMismatchException" para manejar excepciones de tipo
- Se crea un objeto Scanner llamado "sele" para leer la entrada del usuario.
- Se crea un objeto HashSet llamado "nombres" para almacenar los nombres de los alumnos.
- Se crea un objeto HashMap llamado "alumnosNotas" para almacenar las notas de los alumnos por materia.
- Se declara una variable de tipo String llamada "eleccionContinuar".
- Se crea un objeto ArrayList llamado "materias" y se agregan las materias a la lista.
- Se crea un objeto ArrayList llamado "materias Hora" y se agregan los horarios de las materias a la lista.
- Se crea un objeto ArrayList llamado "docentes" y se agregan los nombres de los profesores a sa la lista.
- Se imprime por pantalla el encabezado del programa.
- Se inicia un bucle "do-while" para mostrar el menú de opciones y realizar las operaciones correspondientes.
- Se muestra por pantalla el menú de opciones.
- Se lee la opción ingresada por el usuario y se guarda en la variable "opción".
- Se limpia el buffer del Scanner.
- Se utiliza una estructura de control "switch" para ejecutar el código correspondiente a la opción seleccionada.

```
© Main.java x
1  t java.util.*;
2
3  RSION 9
4
5  -----Alumnos-----
6  ica Nathalia Nasra // *Nelio Bazan // *Fernando Iglesias
7  nsilla Sara // *Fernando Serrano // *Miguel Marchione // *Evelyn Pini
8  rgio Peluffo // *Bulich Canelo Agustina // *Mauro Cardin //
9  -----
10
11  c class Main {
12  o usages
13  tring nombre;
14  ublic static void main(String[] args) throws InputMismatchException {
15      Scanner sele = new Scanner(System.in);
16      HashSet<String> nombres = new HashSet<String>();
17      HashMap<String, HashMap<String, Integer>> alumnosNotas = new HashMap<>();
18      String eleccionContinuar;
19
20      ArrayList<String> materias = new ArrayList<>();
21      materias.add("Teleinformatica"); //sacamos todo esto del caso 5 para que no rompa el caso 6
22      materias.add("Programación web");
23      materias.add("Arquitectura del computador");
24      materias.add("Diseño y administración de base de datos");
```



```
33 ArrayList<String> docentes = new ArrayList<String>();
34 docentes.add("Prof. Mabel Escobar Korajay"); //sacamos todo esto del caso 4
35 docentes.add("Dibu Martinez");
36 docentes.add("Papu Gomez");
37 docentes.add("Leandro Paredes");
38 docentes.add("Enzo Fernandez");
39
40
41 System.out.println("+++++");
42 System.out.println("      Instituto Herman Hollerith");
43 System.out.println("      Moron, Buenos Aires");
44 System.out.println("-----version 9.0 2023-11-12-----");
45 System.out.println("+++++");
46
47
```



```
48 do {
49     System.out.println("Elija una de las opciones:");
50     System.out.println("1- Ingresar alumnos ");
51     System.out.println("2- Modificación de alumno cargado");
52     System.out.println("3- Borrar alumno");
53     System.out.println("4- Listar profesores");
54     System.out.println("5- Listar materias y horarios");
55     System.out.println("6- Cargar las notas de las materias");
56     System.out.println("7- Lista de alumnos");
57     System.out.println("8- Lista de alumnos con materias, notas y estado de aprobación");
58
59     int opcion = sele.nextInt();
60     sele.nextLine(); // Limpiar el buffer del scanner

```



## Caso 1: Ingresar alumnos.

- Se solicita al usuario que ingrese el nombre del alumno. b. Se agrega el nombre del alumno al HashSet "nombres". c. Se pregunta al usuario si quiere seguir cargando nombres y se guarda la respuesta en la variable "b". d. Se repite el paso a y b mientras "b" sea igual a "s" o "S". e. Se muestra por pantalla la lista de alumnos almacenados en el HashSet "nombres". f. Se rompe el caso 1.

## Caso 2: Modificación de alumno cargado.

- Se muestra por pantalla la lista de alumnos almacenados en el HashSet "nombres". b. Se solicita al usuario que ingrese el nombre del alumno que desea modificar y se guarda en la variable "nombreModificar". c. Si el nombre ingresado se encuentra en el HashSet "nombres": i. Se remueve el nombre antiguo del alumno del HashSet "nombres". ii. Se solicita al usuario que ingrese el nuevo nombre para el alumno y se guarda en la variable "nuevoNombre". iii. Se agrega el nuevo nombre del alumno al HashSet "nombres". iv. Se muestra por pantalla la lista de alumnos actualizada. d. Si el nombre ingresado no se encuentra en el HashSet "nombres", se muestra un mensaje de error. e. Se rompe el caso 2.

```
© Main.java x
61
62 switch (opcion) {
63     case 1:
64         System.out.println("Escriba el nombre del alumno: ");
65         String a = sele.nextLine();
66         nombres.add(a);
67         String b;
68         System.out.println("¿Quiere seguir cargando? s/n");
69         do {
70             b = sele.nextLine().toLowerCase();
71             if (!b.equals("s") && !b.equals("n")) {
72                 System.out.println("Ingrese una opción válida (s/n): ");
73             }
74         } while (!b.equals("s") && !b.equals("n"));
75         while ("s".equalsIgnoreCase(b)) {
76             System.out.println("Escriba el nombre del alumno: ");
77             a = sele.nextLine();
78             nombres.add(a);
79             System.out.println("¿Quiere seguir cargando? s/n");
80             do {
81                 b = sele.nextLine().toLowerCase();
82                 if (!b.equals("s") && !b.equals("n")) {
83                     System.out.println("Ingrese una opción válida (s/n): ");
84                 }
85             } while (!b.equals("s") && !b.equals("n"));
86         }
87     }
```

```
© Main.java x
91 case 2:
92     System.out.println("La lista de alumnos a modificar es : " + nombres);
93     System.out.println("Ingrese el nombre del alumno que desea modificar: ");
94     String nombreModificar = sele.nextLine();
95
96     if (nombres.contains(nombreModificar)) {
97         nombres.remove(nombreModificar);
98
99         System.out.println("Ingrese el nuevo nombre para el alumno: ");
100        String nuevoNombre = sele.nextLine();
101        nombres.add(nuevoNombre);
102
103        System.out.println("Alumno modificado. La lista de alumnos es: " + nombres);
104    } else {
105        System.out.println("El alumno no se encuentra en la lista.");
106    }
107
108    break;
```



### Caso 3: Borrar alumno.

- Se muestra por pantalla la lista de alumnos almacenados en el HashSet "nombres". b. Se solicita al usuario que ingrese el nombre del alumno que desea eliminar y se guarda en la variable "nombreAEliminar". c. Si el nombre ingresado se encuentra en el HashSet "nombres": i. Se remueve el nombre del alumno del HashSet "nombres". ii. Se muestra un mensaje indicando que el alumno ha sido eliminado de la lista. d. Si el nombre ingresado no se encuentra en el HashSet "nombres", se muestra un mensaje de error. e. Se muestra por pantalla la lista de estudiantes actualizada. f. Se rompe el caso 3.

### Caso 4: Listar profesores.

- Se muestra por pantalla la lista de profesores almacenada en el ArrayList "docentes". b. Se rompe el caso 4.

### Caso 5: Listar materias y horarios.

- Se muestra por pantalla la lista de materias y horarios almacenada en el ArrayList "materiasHora". b. Se rompe el caso 5.

```
case 3:
    System.out.println("Lista de alumnos: " + nombres);

    System.out.print("Ingresa el nombre del alumno que desea eliminar: ");
    String nombreAEliminar = sele.nextLine();

    if (nombres.contains(nombreAEliminar)) {
        nombres.remove(nombreAEliminar);
        System.out.println(nombreAEliminar + " ha sido eliminado de la lista.");
    } else {
        System.out.println("Error: el nombre no existe en la lista.");
    }

    System.out.println("Lista de estudiantes actualizada: " + nombres);
    break;

case 4:

    System.out.println("Lista de profesores: " + docentes);
    break;

case 5:

    System.out.println("Lista de materias y horarios: " + materiasHora);

    break;
```





### Caso 6: Cargar las notas de las materias.

a. Se muestra por pantalla la lista de materias almacenada en el ArrayList "materias". b. Se muestra por pantalla la lista de estudiantes actualizada. c. Se definen las variables "notas" y "nota" de tipo int. d. Se crea un HashMap llamado "notasMaterias". e. Se solicita al usuario que ingrese el nombre del alumno y se guarda en la variable "nombreAlumno". f. Se inicia un bucle "for" para iterar sobre las materias. i. Se solicita al usuario que ingrese la nota de la materia y se guarda en la variable "notas". ii. Se utiliza un bloque "try-catch" para capturar una excepción de tipo InputMismatchException en caso de que se ingrese una nota no válida. iii. Se asigna la nota ingresada a la posición correspondiente del array "nota". iv. Se valida que la nota ingresada esté entre 0 y 10. v. Se repite el paso f-i hasta que se ingrese una nota válida. vi. Se agrega la materia y la nota al HashMap "notasMaterias". g. Se agrega el HashMap "notasMaterias" al HashMap "alumnosNotas" utilizando el nombre del alumno como clave. h. Se muestra por pantalla las notas registradas para el alumno. i. Se muestra por pantalla el HashMap "notasMaterias". j. Se pregunta al usuario si quiere seguir cargando notas y se guarda la respuesta en la variable "b". k. Se rompe el caso 6.

```
© Main.java x
138
139 case 6:
140     System.out.println("La lista de materias es " + materias);
141     System.out.println("Lista de estudiantes actualizada: " + nombres);
142     int notas;
143     int[] nota = new int[5];
144     nota = new int[5];
145
146     System.out.println("Ingrese Nombre del Alumno: ");
147     String nombreAlumno = sele.nextLine();
148
149     HashMap<String, Integer> notasMaterias = new HashMap<>();
150
151     for (int i = 0; i < 5; i++) {
152         do {
153             try {
154                 System.out.println("Ingrese la nota de " + materias.get(i) + ": ");
155                 notas = sele.nextInt();
156                 if (notas < 0 || notas > 10) {
157                     throw new InputMismatchException("La nota debe estar entre 0 y 10.");
158                 }
159             } catch (InputMismatchException e) {
160                 System.out.println("Error: Ingrese una nota válida entre 0 y 10.");
161                 sele.nextLine(); // limpiar el buffer del scanner
162             }
163         } while (nota[i] < 0 || nota[i] > 10); // Repetir hasta que se ingrese una nota válida
164
165         notasMaterias.put(materias.get(i), nota[i]);
166     }
167
168     alumnosNotas.put(nombreAlumno, notasMaterias);
169
170     System.out.println("Notas registradas para el alumno: " + nombreAlumno);
171     System.out.println(notasMaterias);
172
173     b = sele.nextLine();
174     break;
```

```

175     }
176     nota[i] = notas;
177 } catch (InputMismatchException e) {
178     System.out.println("Error: Ingrese una nota válida entre 0 y 10.");
179     sele.nextLine(); // limpiar el buffer del scanner
180 }
181 } while (nota[i] < 0 || nota[i] > 10); // Repetir hasta que se ingrese una nota válida
182
183 notasMaterias.put(materias.get(i), nota[i]);
184 }
185
186 alumnosNotas.put(nombreAlumno, notasMaterias);
187
188 System.out.println("Notas registradas para el alumno: " + nombreAlumno);
189 System.out.println(notasMaterias);
190
191 b = sele.nextLine();
192 break;
```



### Caso 7: Lista de alumnos.

- a. Se muestra por pantalla la lista de alumnos almacenada en el HashSet "nombres". b. Se rompe el caso 7.

### Caso 8: Lista de alumnos con materias, notas y estado de aprobación.

- a. Se muestra por pantalla la lista de alumnos con sus respectivas materias, notas y estado de aprobación. b. Se rompe el caso 8.

Default: Si la opción ingresada no es válida, se muestra un mensaje de error.

- Se pregunta al usuario si desea continuar en el programa y se guarda la respuesta en la variable "eleccionContinuar".
- Se repite el bucle "do-while" mientras "eleccionContinuar" sea igual a "s" o "S".
- Se muestra por pantalla un mensaje indicando que el programa ha sido cerrado. Fin del programa.

```
© Main.java x
177 case 7:
178     System.out.println("Lista de alumnos:");
179     for (String alumno : nombres) {
180         System.out.println(alumno);
181     }
182     break;
183
184
```



```
84 case 8:
85     System.out.println("Lista de alumnos con materias, notas y estado de aprobación:");
86     for (Map.Entry<String, HashMap<String, Integer>> entry : alumnosNotas.entrySet()) {
87         System.out.println("Alumno: " + entry.getKey());
88         HashMap<String, Integer> notasMateriasEntry = entry.getValue();
89
90         for (Map.Entry<String, Integer> notaEntry : notasMateriasEntry.entrySet()) {
91             if (notaEntry.getValue() >= 7)
92                 System.out.println("Materia: " + notaEntry.getKey() + ". Nota " + notaEntry.getValue() + ". APROBADO.");
93             else
94                 System.out.println("Materia: " + notaEntry.getKey() + ". Nota " + notaEntry.getValue() + (notaEntry.getValue() < 4 ? ". REPROBADO" : ". A FINAL"));
95         }
96     }
97     break;
```





# Fin del programa.

Agradecemos a todos por utilizar esta aplicación.  
Esperamos que les haya sido útil y que lo hayan disfrutado. ¡

```
202     }
203
204     System.out.println("¿Desea continuar en el programa? s/n");
205     eleccionContinuar = sele.nextLine().toLowerCase();
206     while (!eleccionContinuar.equals("s") && !eleccionContinuar.equals("n")) {
207         System.out.println("Ingrese una opción válida (s/n): ");
208         eleccionContinuar = sele.nextLine().toLowerCase();
209     }
210     while ("s".equals(eleccionContinuar));
211     System.out.println("El programa ha sido cerrado");
212 }
213 }
```

Link Código: [Trabajo final Gestión de Alumnos](#)



Erica Nathalia Nasra  
Nelio Basan  
Bulich Canelo Agustina  
Mansilla Sara  
Fernando Iglesias  
Fernando Serrano  
Miguel Marchione  
Evelyn Pini  
Sergio Peluffo  
Mauro Cardin

<https://www.linkedin.com/in/nathy-nasra-933a7928/>  
<https://www.linkedin.com/in/nelio-bazan-1a742764/>  
<https://www.linkedin.com/in/agustina-bulich-canelo-48a66129b/>  
<https://www.linkedin.com/in/sara-mansilla-909371222/>  
<https://github.com/Fernando-I-005>  
[serranofernando.979@gmail.com](mailto:serranofernando.979@gmail.com)  
[miguelmarchione@gmail.com](mailto:miguelmarchione@gmail.com)  
<https://www.linkedin.com/in/evelyn-pini/>  
<https://www.linkedin.com/in/sergio-peluffo-8a4324142/>  
<https://www.linkedin.com/in/mauro-ezequiel-cardin-860330172/>

