



Práctica 1

Archivos

Prof. Carmen Lucia Bustillo Hernández

carmenbustillohernandez@gmail.com

Facultad de Ciencias
Universidad Nacional Autónoma de México

Fecha de entrega: 26 de Agosto de 2025

Datos del Alumno

Nombre:
Matrícula:

Objetivo General

Las computadoras utilizan archivos para la retención a largo plazo de grandes cantidades de datos, incluso hasta después de que terminan los programas que crean esos datos. Los datos que se mantienen en archivos como datos persistentes, ya que existen más allá de la duración de la ejecución del programa. Las computadoras almacenan archivos en dispositivos de almacenamiento secundario como discos duros, discos ópticos y cintas magnéticas. El objetivo de esta practica es la implementación de archivos para un caso de estudio de una Agenda Electrónica e implementar las funciones de creación, eliminación, modificación y consulta de registros en lenguaje Java.

Objetivos Específicos

1. Creación de archivos
2. Apertura y cierre de archivos
3. Lectura y escritura de archivos
4. Creación de registros en agenda electrónica
5. Consultar de registros en agenda electrónica
6. Modificación de registros en agenda electrónica
7. Eliminación de registros en agenda electrónica

Material

1. Equipo de cómputo con sistema operativo Windows. Mac o Linux y acceso a Internet.
2. Equipo de cómputo con JRE o JSDK y una IDE de desarrollo (Eclipse o NetBeans) o
3. Equipo de cómputo con compilador en línea https://www.onlinegdb.com/online_c_compiler

Bibliografía

1. López Gaona Amparo, Introducción al desarrollo con Java. Las prensas de Ciencias, Segunda Edición, 2011
2. Deitel, H. M., Deitel, P. J., & García, G. A. C. (2008). Java: Cómo programar.
3. Viso E. y Peláez C., Introducción a las Ciencias de la Computación con Java. Las Prensas de Ciencias

1. Instrucciones

1. Escriba el programa que implemente cada uno de los ejercicios propuestos.
2. Escriba el programa que implemente una agenda electrónica.
3. Implemente la función para la creación de registros agenda electrónica.
4. Implemente la función para la consulta de registros agenda electrónica.
5. Implemente la función para la eliminación de registros agenda electrónica.
6. Implemente la función para la modificación de registros agenda electrónica.
7. Agregue en su carpeta correspondiente de Google Classroom su código y su ejecutable desarrollado de cada problema, de acuerdo a las siguientes etiqueta:
 - AgendaElectronica.java
 - LibroDirecciones.java
 - Citas.java
 - Notas.java
 - Calendario.java

2. Reglas Importantes

- Cumplir con los lineamientos de entrega.
- Todos los archivos deberán contener nombre y número de cuenta, como comentario de autor en el código.
- El código debe estar comentado. Esto es, clases, atributos, métodos y comentarios extra.
- Para cada actividad solicitada, crear un nuevo archivo.
- Utilizar correctamente las convenciones para nombrar variables, constantes, clases y métodos.
- El archivo debe estar comprimido en un ZIP, como se mostró el primer día de clase.
- Queda prohibido copiar código, dado el caso la calificación será nula, si incurre nuevamente anula práctica y proyecto.
- En caso de no cumplirse alguna de las reglas anteriores, se restará 0.5pts. en la calificación obtenida.

3. Ejercicios

3.1. Creación de un archivo

```
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;

public class CrearArchivo {
    public static void main(String[] args) {
        try {
            // Nombre del archivo
            File archivo = new File("ejemplo.txt");

            // Crear el archivo si no existe
            if (archivo.createNewFile()) {
                System.out.println("Archivo creado: " + archivo.getName());
            } else {
                System.out.println("El archivo ya existe.");
            }

            // Escribir en el archivo
            FileWriter escritor = new FileWriter(archivo);
            escritor.write("Hola, este es un archivo de ejemplo en Java.");
            escritor.close();
            System.out.println("Se escribió en el archivo correctamente.");

        } catch (IOException e) {
            System.out.println("Ocurrió un error.");
            e.printStackTrace();
        }
    }
}
```

3.2. Lectura de un archivo

```
import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;

public class LeerArchivo {
    public static void main(String[] args) {
        try {
            // Nombre del archivo a leer
            File archivo = new File("ejemplo.txt");

            // Usamos Scanner para leer línea por línea
            Scanner lector = new Scanner(archivo);

            while (lector.hasNextLine()) {
                String linea = lector.nextLine();
            }
        }
    }
}
```

```

        System.out.println(linea);
    }

    lector.close();

} catch (FileNotFoundException e) {
    System.out.println("No se encontró el archivo.");
    e.printStackTrace();
}
}
}

```

3.3. Escritura de un archivo

```

import java.io.FileWriter;
import java.io.BufferedWriter;
import java.io.IOException;

public class EscrituraArchivo {
    public static void main(String[] args) {
        try {
            // Creamos un FileWriter (true = agregar al final, false = sobrescribir)
            FileWriter fw = new FileWriter("ejemplo.txt", false);

            // Usamos BufferedWriter para mayor eficiencia
            BufferedWriter bw = new BufferedWriter(fw);

            // Escribir líneas en el archivo
            bw.write("Primera línea en el archivo.");
            bw.newLine(); // Salto de línea
            bw.write("Segunda línea escrita en Java.");
            bw.newLine();
            bw.write(" Archivo escrito con éxito!");

            // Cerrar el flujo
            bw.close();

            System.out.println("Se escribió correctamente en el archivo.");

        } catch (IOException e) {
            System.out.println("Ocurrió un error al escribir en el archivo.");
            e.printStackTrace();
        }
    }
}

```

3.4. Caso de Estudio: Agenda Electrónica

Como un caso de estudio se va a desarrollar un programa de consola de una agenda electrónica. La lectura de los datos sera por medio de un archivo con extensión .data, en cual deberas leer, buscar, eliminar y escribir la información pertinente. La funcionalidad de la agenda incluye lo siguiente: **(100 pts)**.

1. Libro de Direcciones. Incluye los siguientes campos de información de las personas como son:
 - (a) Nombre(s)
 - (b) Apellido Paterno
 - (c) Apellido Materno
 - (d) Dirección(es)
 - (e) Teléfono
 - (f) Móvil
 - (g) Correo electrónico
 - (h) Compañia
 - (i) Puesto
 - (j) URL
 - (k) Fbk
 - (l) IG
 - (m) Cita 1
 - Título
 - Fecha
 - Hora
 - (n) Cita 2
 - Título
 - Fecha
 - Hora
 - (ñ) Nota 1
 - (o) Nota 2
2. Citas. Utilidades para gestionar las citas y reuniones del usuario, insertar citas, borrar citas, buscar citas por fecha.
 - (a) Titulo
 - (b) Fecha
 - (c) Hora
3. Notas. Utilidades para gestionar una lista de notas, insertar notas , borrar notas o listar notas.
 - (a) Descripcion

4. Calendario. Permite la visualización de calendarios anuales.

Nota: Puede utilizar un diseño modular de top-down para la implementación