Nombre: Andrés Samuel Podadera González
Ciclo: CFGS Desarrollo de aplicaciones multiplataforma
Curso: 2024/2025

Acceso a datos

Tarea Unidad 1

Contenido

Nombre: Andrés Samuel Podadera González
Ciclo: CFGS Desarrollo de aplicaciones multiplataforma
Curso: 2024/2025

EJERCICIO 1.

En esta tarea te pediremos que investigues y profundices sobre las diferentes formas de acceso a los datos que puede usar un programa o aplicación, valorando que ventajas e inconvenientes presentan cada una. Para ello elabora un documento en .pdf en el que expliques de forma clara y resumida en que consiste cada una de las principales formas de acceso (puedes usar tablas, gráficos y todo lo que consideres necesario para que tu documento quede lo más claro y explicativo posible).

Acceso mediante Ficheros

En los primeros tiempos de la informática, los datos se almacenaban directamente en ficheros. Este método se usa aún en aplicaciones simples para guardar configuraciones o logs.

- **Ventajas**:
    - Simplicidad de implementación.
    - Ideal para almacenar pequeñas cantidades de datos.
- **Desventajas**:
    - Redundancia e inconsistencia de datos.
    - Requiere conocer la estructura exacta del fichero.
    - Escalabilidad limitada.

Bases de Datos Relacionales

Organizan los datos en tablas y utilizan SQL para realizar consultas. Son la opción más común en aplicaciones de negocio.

- **Ventajas**:
    - Soporte para transacciones y seguridad.
    - Centralización y eliminación de redundancias.
    - Independencia de datos: se puede modificar la estructura sin cambiar la aplicación.
- **Desventajas**:
    - Limitaciones en el modelado de datos complejos.
    - Menor rendimiento en operaciones con datos muy interrelacionados en comparación con bases de datos orientadas a objetos.

Bases de Datos Orientadas a Objetos

Estas bases de datos almacenan datos como objetos, siguiendo el modelo de programación orientado a objetos.

- **Ventajas**:
    - Integración directa con aplicaciones orientadas a objetos.
    - Mayor capacidad de modelado para datos complejos y multimedia.
- **Desventajas**:
    - Curva de aprendizaje y menor adopción en el mercado.
    - Falta de estándares universales, lo que puede afectar la interoperabilidad.

Bases de Datos Objeto-Relacionales

Combinan características de las bases de datos relacionales y orientadas a objetos, permitiendo flexibilidad en el diseño.

- **Ventajas**:
    - Permiten una transición desde sistemas relacionales sin grandes cambios.
    - Soporte para tipos de datos complejos y métodos personalizados.
- **Desventajas**:
    - Complejidad adicional y sobrecarga en la administración de datos.

Bases de Datos XML

Almacenan datos en formato XML, ideal para intercambio de datos entre diferentes plataformas.

- **Tipos**:
    - **Nativas XML**: almacenan datos en formato XML.
    - **XML-compatible**: permiten la entrada y salida de datos en formato XML.
- **Ventajas**:
    - Portabilidad y facilidad de intercambio entre sistemas.
    - Compatible con datos estructurados y semiestructurados.
- **Desventajas**:
    - Menor rendimiento en comparación con bases de datos relacionales en consultas complejas.

Mapeo Objeto-Relacional (ORM)

Técnica de programación que convierte datos entre sistemas orientados a objetos y bases de datos relacionales. Soluciones muy comprobadas que están integradas en los frameworks o marcos de trabajo.

- **Ventajas**:
    - Simplifica el desarrollo al evitar SQL explícito.

  - Gestiona automáticamente la persistencia de objetos en bases de datos.

- **Desventajas**:
  - Desfase objeto-relacional (dificultad de compatibilidad entre paradigmas).
  - Puede afectar el rendimiento en consultas muy complejas.

Conectores JDBC y ODBC

Estándares que facilitan la conexión entre aplicaciones y bases de datos.

  - **JDBC (Java Database Connectivity)**: específico para Java.
  - **ODBC (Open Database Connectivity)**: estándar que permite conectar aplicaciones a diversas bases de datos.

- **Ventajas**:
  - Abstracción de la conexión de base de datos, independizando a la aplicación del tipo específico de base de datos.

- **Desventajas**:
  - Requiere configuración adecuada para optimizar el rendimiento.
  - Las conexiones pueden generar sobrecarga si no se gestionan correctamente.

Tabla resumen

Estrategia	Ventajas	Desventajas
Ficheros	Simple, económico, útil para pequeñas tareas	Redundancia, inconsistencia, no escalable
Bases de Datos Relacionales	Eficiencia, integridad, seguridad	Menos efectivas para datos complejos
Bases de Datos Orientadas a Objetos	Modelado de datos complejo, integración directa	Menor adopción, falta de estándares
Bases de Datos Objeto-Relacionales	Flexibilidad, permite transición gradual	Complejidad añadida, administración difícil
Bases de Datos XML	Ideal para intercambio de datos entre aplicaciones. Es un estándar.	Bajo rendimiento en consultas complejas
ORM	Simplificación del código, evita SQL explícito	Desfase objeto-relacional, rendimiento
Conectores JDBC/ODBC	Independencia de base de datos	Configuración y gestión de rendimiento

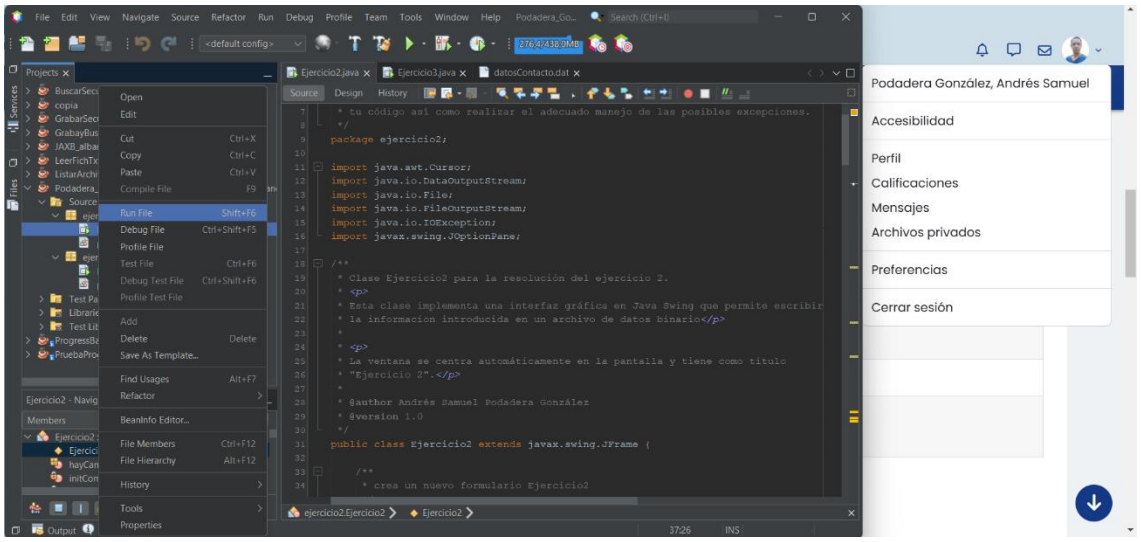
Nombre: Andrés Samuel Podadera González
Ciclo: CFGS Desarrollo de aplicaciones multiplataforma
Curso: 2024/2025

EJERCICIO 2.

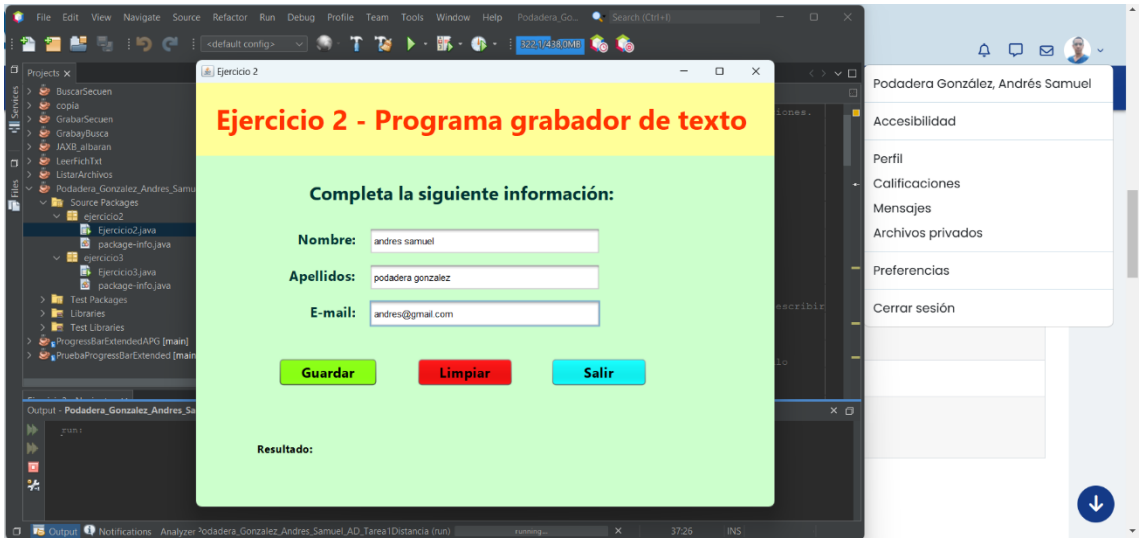
Esta tarea vamos a crear un programa en Java, que permite introducir el nombre, apellidos, y e-mail de un usuario mediante un menú que pida estos datos y los grabe de forma secuencial en un fichero llamado "datosContacto.dat" del directorio local del proyecto. Sería necesario introducir un carácter como por ejemplo $ para separar un campo de otro. Deberás documentar bien tu código, así como realizar el adecuado manejo de las posibles excepciones.

Guía de ejecución del Programa:

Ejecutamos nuestro programa haciendo clic derecho sobre el fichero y seleccionando “Run File”:

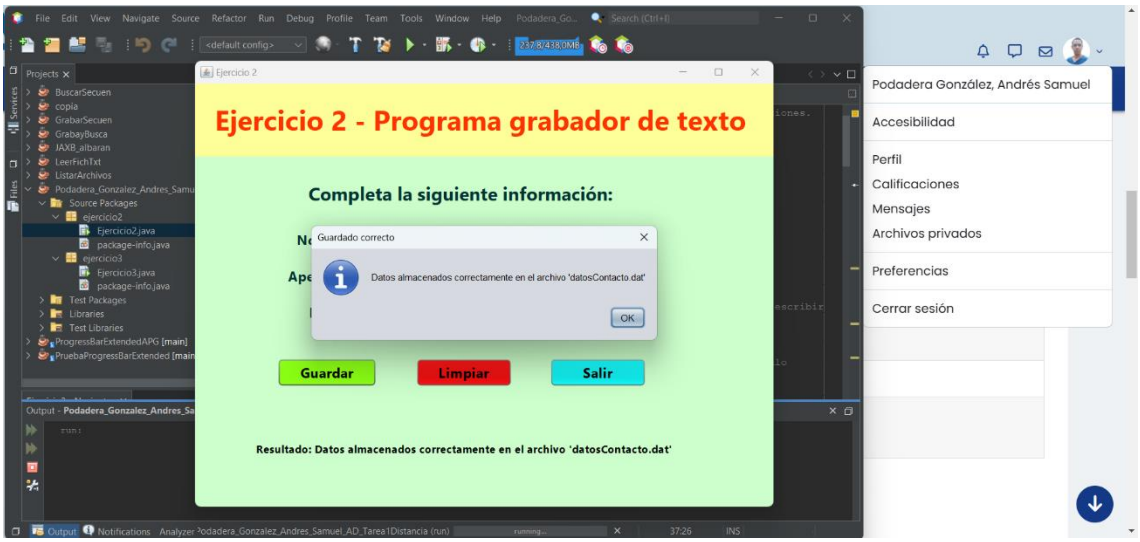


Una vez dentro de la interfaz rellenaremos la información solicitada y haremos clic en el botón Guardar:



Entonces el programa nos mostrará un aviso si la operación de guardado se realizó correctamente y un mensaje de resultado indicando don de guardó la información.

Nombre: Andrés Samuel Podadera González
Ciclo: CFGS Desarrollo de aplicaciones multiplataforma
Curso: 2024/2025



Ya podremos hacer clic en OK e introducir la información de otro usuario o por otro lado, hacer clic en el botón Salir si deseamos salir del programa.

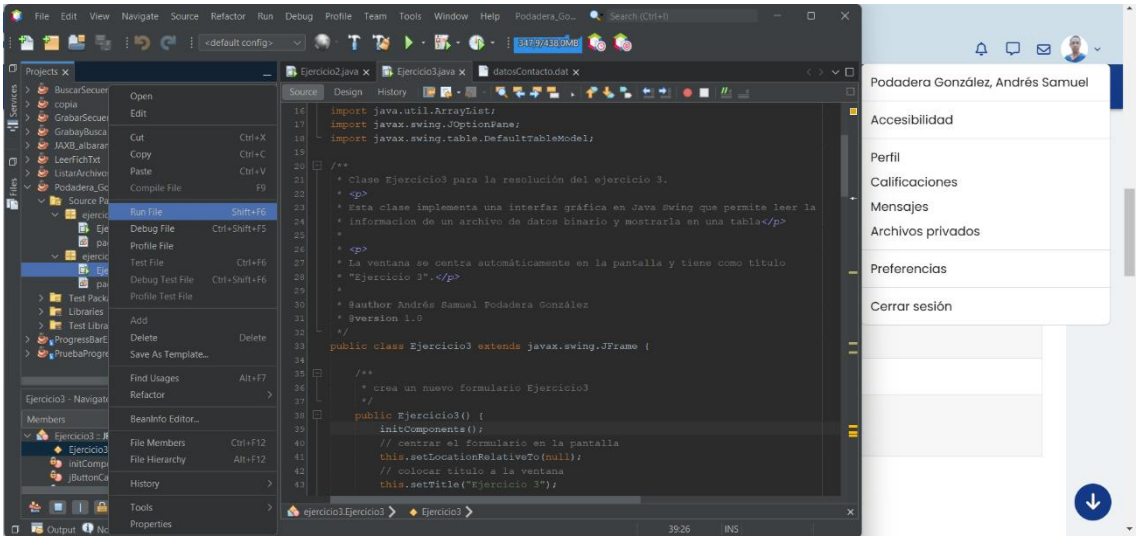
Nota: el botón Limpiar, nos sirve para limpiar el formulario.

EJERCICIO 3.

Crea un programa en Java, que permita leer el fichero llamado "datosContacto.dat" del ejercicio anterior del directorio datos y permita mostrarlos por pantalla de forma secuencial, también habría que contar el número de registros que existen en el fichero. Se entiende por un registro cada línea creada por cada uno de los campos. Deberás documentar bien tu código, así como realizar el adecuado manejo de las posibles excepciones.

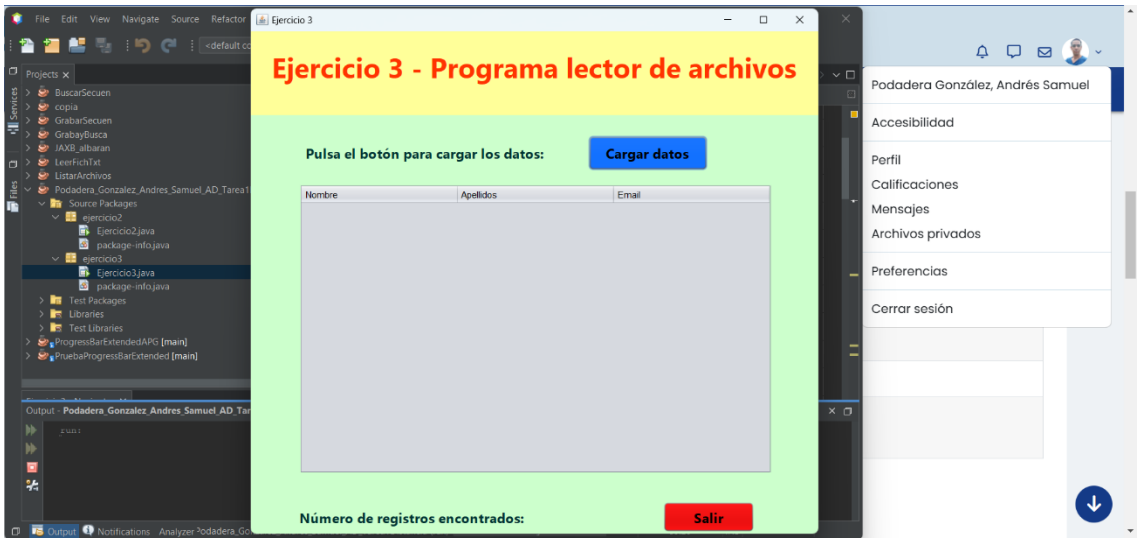
Guía de ejecución del Programa:

Siguiendo el mismo procedimiento anterior, ejecutamos el programa haciendo clic derecho sobre el fichero y seleccionando la opción “Run File”:

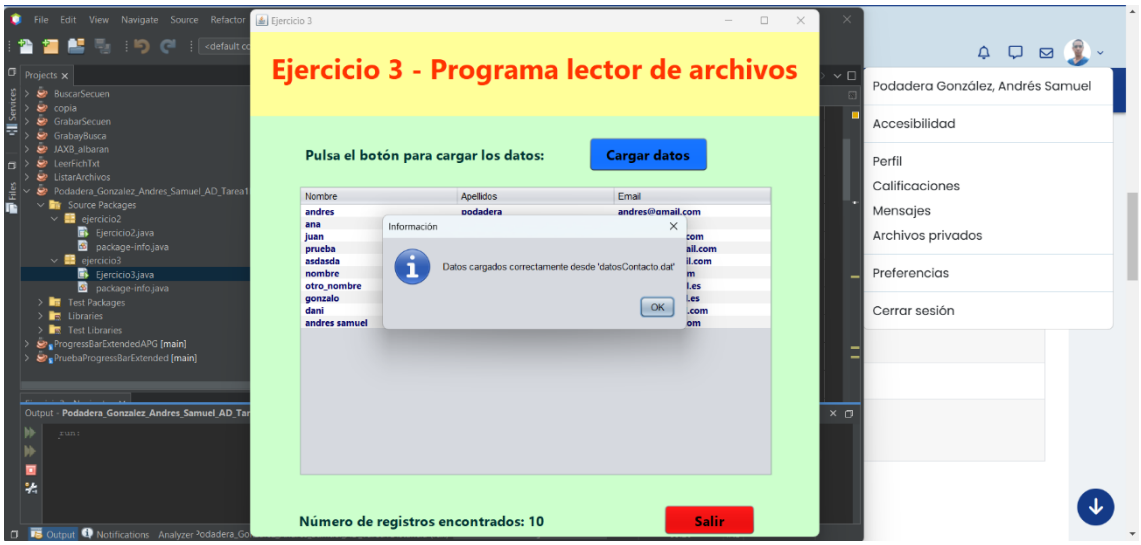


Nombre: Andrés Samuel Podadera González
Ciclo: CFGS Desarrollo de aplicaciones multiplataforma
Curso: 2024/2025

Seguidamente nos aparecerá la interfaz del programa, en la cuál deberemos hacer clic en el botón “Cargar datos” para obtener la información del archivo anteriormente almacenado.



Una vez obtenida la información(en caso de no encontrar el archivo mostrará un mensaje de error) mostrará un mensaje de información al usuario indicando que la operación se realizó correctamente.



Podremos hacer clic en OK y podremos visualizar los registros del archivo en forma de tabla, sin elementos duplicados y el número de registros obtenidos en la etiqueta correspondiente.

Nombre: Andrés Samuel Podadera González
Ciclo: CFGS Desarrollo de aplicaciones multiplataforma
Curso: 2024/2025

Si el usuario hiciera clic de nuevo en el botón “Cargar datos” sólo se cargarán los datos que no estén ya incluidos en la tabla evitando comportamientos erróneos de programa y duplicidad en registros de la tabla.

Cuando el usuario desee salir del programa, tan sólo tendrá que hacer clic en el botón Salir.

FIN