

LEA ESTE DOCUMENTO DETENIDAMENTE

Este es un proyecto **grupal**. Por tanto, todos los miembros del grupo deben involucrarse en el diseño e implementación de la solución que se solicita. La entrega del proyecto incluirá un reporte de auto y coevaluación, donde cada miembro del grupo calificará sus contribuciones al proyecto y las de sus compañeros.

Es su responsabilidad agruparse a tiempo para la entrega del proyecto. Usted debe indicar su grupo en el Aula Virtual hasta el **martes 19 de mayo**. Los estudiantes que no se hayan unido a ningún grupo para entonces, serán asignados aleatoriamente por el profesor. A partir de entonces, no se aceptarán cambios en la conformación de los grupos. **EN NINGUNA CIRCUNSTANCIA SE ACEPTARÁN ENTREGAS INDIVIDUALES.**

Si algún grupo experimenta el incumplimiento sistemático de alguno de sus miembros, esta situación debe ser reportada al profesor **TAN PRONTO COMO SE PRESENTE**. Cualquier estudiante cuyo incumplimiento sea comprobado, será retirado del grupo y **DEBERÁ IMPLEMENTAR EL PROYECTO DE MANERA INDIVIDUAL SOBRE EL 50% DEL PUNTAJE POSIBLE.**

Si algún estudiante siente que otro (u otros) toma(n) el control del grupo y no permite(n) a los demás miembros hacer aportes o colaborar en el proyecto, debe reportar esta situación al profesor **TAN PRONTO COMO SE PRESENTE**. Cualquier estudiante que pretenda monopolizar el trabajo del proyecto o impida a los demás hacer aportes será separado del grupo y **DEBERÁ IMPLEMENTAR EL PROYECTO DE MANERA INDIVIDUAL Y SOBRE EL 50% DEL PUNTAJE POSIBLE.**

Si algún estudiante reporta de manera extemporánea cualquiera de las dos situaciones detalladas en los dos párrafos precedentes, será penalizado con su 25% de la nota total obtenida en el proyecto. Reporte novedades al profesor **TAN PRONTO COMO ÉSTAS SE PRESENTEN.**

Considere las políticas de buena conducta académica que se explicaron en la primera clase del curso respecto al plagio y demás violaciones del código de Ética de la ESPOL. Los autores del proyecto deben ser usted y sus compañeros de grupo (no otras personas, **ni motores de Inteligencia Artificial**). Cualquier indicio de lo contrario, será reportado a las unidades correspondientes para el tratamiento pertinente.

Finalmente, **siga las instrucciones de este documento** para evitar inconvenientes. Si tiene dudas, consulte al profesor en lugar de asumir cosas que pueden ser incorrectas.

El incumplimiento de instrucciones explícitas indicadas en este documento o indicadas en nuestras sesiones teóricas derivará, de manera inapelable, en la penalización y rebaja de puntos sobre su entrega.

Estructuras de Datos

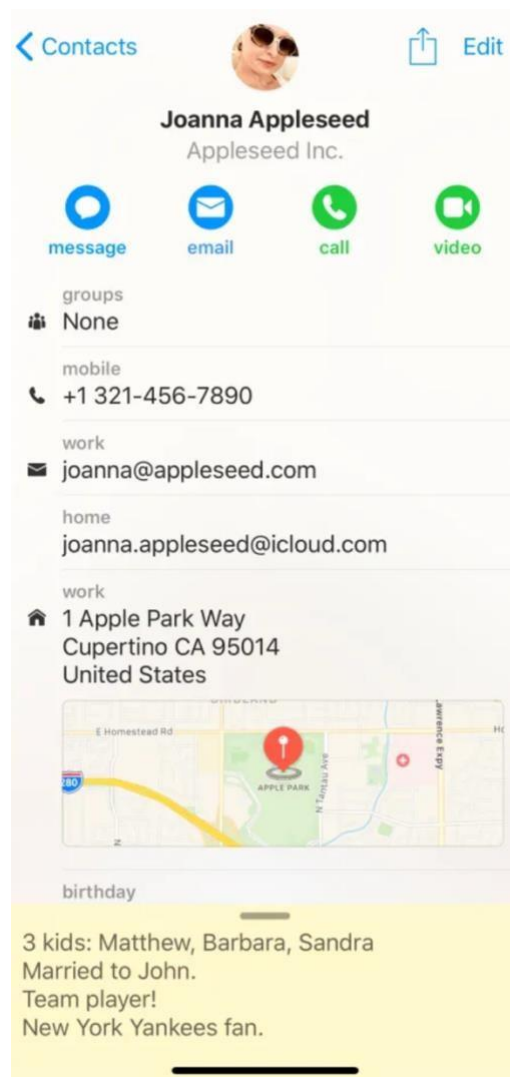
Proyecto: Primera Evaluación – PAO 1 2025

Sistema para gestión de contactos

Introducción

Una aplicación de gestión de contactos suele incluir atributos básicos para cada contacto: nombre, número de teléfono, foto, y dirección. No obstante, actualmente, los perfiles de contactos se han vuelto más complejos y suelen incluir múltiples atributos adicionales, como los listados a continuación, algunos de los cuales se muestran en la figura:

- Dirección de domicilio o trabajo, descriptiva y/o con ubicación geográfica (link de Google Maps, por ejemplo).
- Varios emails: de uso personal, de trabajo, etc.
- Una lista de números telefónicos,
- Identificadores de distintas redes sociales.
- Una colección de fotos.
- Varias fechas de Interés: cumpleaños, aniversarios, renovación de suscripción o de contrato, pagos pendientes, etc.
- Contactos relacionados:
 - Por ejemplo, el contacto de una empresa podría tener asociado el contacto de su director o de alguien que trabaja en un área en particular (ventas, coordinación, etc).
 - El contacto de nuestro médico podría tener asociado también el de su asistente (para agendar citas).
 - El contacto de nuestro mejor amigo podría tener asociado el de su pareja y los de algunos familiares.



Algunas de estas características están incluidas en aplicaciones de gestión de contactos existentes. Les sugerimos investigar cómo estas funcionalidades se presentan típicamente **para inspirar vuestro diseño**.

En este proyecto se implementarán y utilizarán estructuras de datos, específicamente colecciones lineales, para desarrollar una aplicación de gestión de contactos con características similares a las descritas anteriormente. Además, se deberá implementar una interfaz gráfica en JavaFX que permita visualizar un listado de contactos (personas y empresas). Para ello, también será necesario diseñar e implementar clases que representen a los contactos y sus atributos, así como a las demás entidades involucradas en el problema.

Requerimientos mínimos

El proyecto deberá incluir una interfaz gráfica de usuario (GUI) funcional, desarrollada con JavaFX, que permita al usuario realizar, al menos, las siguientes operaciones sobre los contactos:

- Visualizar la lista completa de contactos registrados.
- Crear nuevos contactos.
- Editar la información de cualquier contacto existente.
- Eliminar contactos registrados en el sistema.

El diseño de la interfaz gráfica puede inspirarse en aplicaciones existentes de gestión de contactos (**las cuales deberán ser citadas en el informe final del proyecto**). No obstante, se espera que el equipo de desarrollo incorpore elementos de diseño propios, demostrando criterio estético, usabilidad y creatividad.

Consideraciones:

Cada contacto está compuesto por múltiples atributos, y cada atributo incluye un identificador o descripción (**por ejemplo**, “teléfono del trabajo”) y un valor asociado (**por ejemplo**, “0912345678”). Es importante tener en cuenta que el conjunto de atributos puede variar entre tipos de contacto; por ejemplo, una empresa no posee apellido. No obstante, para garantizar la coherencia y estandarización en la gestión de datos, la aplicación deberá mantener un “listado maestro” de atributos que contenga los campos comunes a todos los contactos. Este listado servirá como base estructural para asegurar uniformidad en la visualización, edición y almacenamiento de la información, independientemente del tipo de contacto.

Funcionalidad mínima

La aplicación deberá implementar, como mínimo, las siguientes funcionalidades esenciales:

- **Navegación de contactos:** El usuario debe poder visualizar y recorrer la lista de contactos de manera bidireccional (hacia adelante y hacia atrás). Esta navegación debe ser **circular**: al avanzar desde el último contacto, debe mostrarse el primero; al retroceder desde el primero, debe visualizarse el último.
- **Navegación de fotos asociadas:** Cada contacto debe contar con múltiples fotos asociadas. La aplicación deberá permitir al usuario navegar entre dichas imágenes también de forma bidireccional y circular.
- **Gestión de contactos:** El sistema debe permitir crear, editar y eliminar contactos. Durante la creación, el usuario podrá seleccionar entre al menos **dos plantillas prediseñadas**: *persona natural* o *empresa*, con atributos relevantes predefinidos para cada caso. La creación también deberá incluir la asociación de una o más fotos al nuevo contacto. El grupo puede extender esta funcionalidad añadiendo más plantillas, si lo considera pertinente.
- **Gestión de atributos:** El usuario podrá añadir, modificar o eliminar atributos individuales de un contacto existente. Ejemplos: añadir una dirección, editar el nombre de usuario en redes sociales o eliminar una foto.

- **Ordenamiento de contactos:** La aplicación deberá permitir ordenar la lista de contactos según al menos **tres** de los siguientes criterios:
 - Apellido y nombre
 - Cantidad de atributos registrados
 - Fecha de cumpleaños más próxima (cuando esté disponible)
 - Nombre de la empresa
 - País de residencia
 - Tipo de contacto (persona o empresa)
 - Presencia de un atributo específico (por ejemplo, contactos con usuario de Instagram)
 - Cualquier otro criterio adicional que el equipo considere relevante
- **Filtrado de contactos:** El usuario podrá aplicar filtros sobre la lista de contactos utilizando, como mínimo, tres de los criterios de ordenamiento previamente listados. **Ejemplo:** mostrar únicamente los contactos que residen en Quito, o aquellos con fecha de nacimiento en febrero.
- **Visualización de contactos asociados:** La aplicación debe permitir visualizar los contactos relacionados con un contacto específico. Además, al seleccionar uno de los contactos asociados, el sistema deberá mostrar en detalle su información.

Funcionalidades opcionales

Las funcionalidades descritas arriba son obligatorias y constituyen los requerimientos mínimos para que su proyecto sea aceptado y admitido a ser calificado hasta el 80% de la nota máxima. Este proyecto, sin embargo, es también una oportunidad para ser creativo. Así, usted y sus compañeros de grupo pueden implementar funcionalidad extra que contribuya a una mejor versión de la aplicación solicitada.

A continuación, se listan algunas funcionalidades opcionales que usted y sus compañeros de grupo podrían implementar para aplicar a puntos extra:

- **Búsqueda avanzada:** Permitir la búsqueda de contactos basada en múltiples criterios a la vez, como apellido y ubicación (**p. ej.**, *Listar todos los contactos de apellido Miranda nacidos en abril, o todas las empresas con dirección en Guayaquil*).
- **Etiquetado de contactos:** Ofrecer la opción de etiquetar contactos con categorías personalizadas (amigos, trabajo, familia, etc.), y luego filtrar o buscar basado en esas etiquetas.
- **Exportar/Importar contactos:** Permitir la exportación e importación de contactos a/desde formato vCard.
- **Favoritos y acceso rápido:** Permitir marcar contactos como favoritos para un acceso rápido.
- **Grupos de contactos:** Ofrecer la posibilidad de crear y gestionar grupos de contactos (por ejemplo, el grupo “familia”, o el grupo “amigos ESPOL”).
- **Inicio de sesión:** Manejar distintos usuarios, cada uno de los cuales tiene un directorio de contactos propios. Al iniciar sesión, la aplicación le permitiría al usuario correspondiente explorar únicamente su lista de contactos asociados. También podría existir un contacto administrador, que pueda ver y modificar la lista de contactos de todos los contactos registrados en el sistema.

El grupo tiene la libertad de implementar cualquier funcionalidad que consideren relevante y útil para este proyecto, siempre que dicha funcionalidad haga un uso pertinente y apropiado de las estructuras de datos revisadas en el curso durante el primer parcial.

Modelamiento del problema y consideraciones técnicas

- La funcionalidad solicitada en este proyecto puede ser lograda con listas, los contactos manejados en la aplicación pueden ser almacenados en una lista circular doblemente enlazada (al igual que las fotos asociadas a cada contacto). Esto permitiría al usuario iterar sobre los contactos hacia adelante hacia atrás. La aplicación debe consistir en una interfaz gráfica de JavaFX que permita al usuario realizar las operaciones indicadas arriba y todas las demás que sean pertinentes para lograr los requerimientos mínimos.
- Dados los objetivos que se persiguen en este curso, usted y sus compañeros de grupo serán responsables de implementar las estructuras que consideren pertinentes para la solución de este problema. **El uso de las clases ArrayList y LinkedList de Java está explícitamente prohibido. Violar esta prohibición invalidaría su proyecto y le otorgaría una nota automática de cero (0).** Las demás estructuras de datos de Java Collection Framework (pilas, colas, conjuntos, mapas) pueden ser utilizadas sin problemas, así como las clases que permiten la comparación (Comparable, Comparator) e iteración de elementos (Iterator, ListIterator).
- Toda la información que deba ser “recordada” por la aplicación (por ejemplo, los contactos existentes, las fotos, los registrados en el sistema, etc.) se debe almacenar en archivos. De esta forma, los datos de la aplicación serán persistentes y estarán disponibles cada vez que ésta se ejecute, **sin necesidad de que alguien tenga que ingresar toda la información desde cero.** El grupo es responsable de decidir la estrategia mediante la cual logrará la persistencia de los datos. Es decir, queda a discreción utilizar archivos de texto plano o serializar la información en archivos binarios. **No utilice almacenamiento que requiera motores de bases de datos.**

Entregables

Entrega parcial (40%): **domingo 08 de junio 8:00pm**

Su proyecto de NetBeans (enviado como un archivo .zip) debe incluir, por lo menos:

- Implementación básica de la interfaz gráfica de usuario (GUI) en JavaFX.
- Diseño e implementación de las clases para representar a los contactos y sus atributos.
- Funcionalidad para visualizar la lista de contactos.
- Funcionalidad para crear nuevos contactos.

Entrega final (60%): **lunes 23 de junio 8:00pm**

- Esta entrega corresponde al proyecto de NetBeans que implementa correctamente la interfaz gráfica final de su proyecto, con –al menos– las funcionalidades mínimas detalladas anteriormente.
- Su entrega deberá incluir además un archivo **.docx**, que es la plantilla que se adjunta a este documento el cual debe de contener:
 - ❖ Screenshots de su interfaz explicando su funcionalidad,
 - ❖ Una imagen de su clase principal, y
 - ❖ Tabla de auto- y co-evaluación de los miembros del grupo
- Su proyecto de Netbeans y el archivo .docx deben ser entregados a través del Aula Virtual en un único archivo comprimido .zip.

No se aceptarán:

- Entregas atrasadas.
- Entregas sin reporte.
- Entregas que no implementen los requerimientos mínimos.

Recuerde que este **es un proyecto grupal**. Esto significa que usted debe trabajar **con** sus compañeros de grupo para sacar el proyecto adelante. Su grupo deberá reunirse periódicamente para coordinar y discutir acciones relacionadas al proyecto. Dividir funcionalidad y repartir responsabilidades podría no ser la mejor estrategia, sobre todo en las fases tempranas del proyecto.

Presentaciones

El proyecto ya tiene que haber sido entregado en el Aula Virtual según las indicaciones dadas. El **martes 24 de junio de 13:00 a 15:00** en el laboratorio donde se desarrollan las clases el grupo tendrá **15 MINUTOS** para efectuar la respectiva presentación, las observaciones sobre la entrega serán dadas de manera verbal.

Solo los grupos que hayan entregado un proyecto funcional podrán presentarse a esta reunión.

La reunión iniciará con uno de los miembros del grupo mostrando su proyecto en funcionamiento. Esta demostración debe evidenciar que el grupo ha cumplido con los requerimientos mínimos del proyecto y debe mencionar explícitamente toda funcionalidad extra que hayan implementado. **Su grupo debe definir con anticipación qué miembro del grupo estará a cargo de esta presentación inicial.**

Recomendaciones y condiciones para la presentación del proyecto

- La presentación deberá centrarse exclusivamente en **lo que su grupo ha desarrollado**. No repitan ni describan los requerimientos del proyecto, ya que estos son conocidos por el docente.
- Se espera una **demostración funcional del sistema** en ejecución. Muestren con claridad qué partes han implementado, qué problemas enfrentaron y cómo los resolvieron. El foco debe estar en **lo logrado, no en lo planeado**.
- Cada integrante del grupo **debe responder al menos una pregunta técnica** durante la exposición. Estas preguntas evaluarán su comprensión del código, su participación en el desarrollo y su dominio del proyecto. Las respuestas serán calificadas individualmente y representarán uno de los dos **factores multiplicativos** que compondrán la nota final del proyecto.
- Cada estudiante deberá tener abierto el código fuente **en su estación de trabajo**, en la computadora del laboratorio donde se realice la presentación. No se aceptará que un solo miembro manipule el proyecto durante toda la demostración.
- Está **terminantemente prohibido** presentar código que no haya sido desarrollado por el equipo. Se realizarán preguntas orientadas a identificar posibles casos de copia o uso indebido de material externo. Las inconsistencias en las respuestas técnicas o el desconocimiento del funcionamiento del sistema serán tomadas como **evidencia de no autoría** y afectarán gravemente la calificación.

Restricciones adicionales y criterios de evaluación rigurosos

- Los estudiantes deben demostrar **conocimiento profundo** del código que presentan. Se penalizará el desconocimiento de funciones implementadas, estructuras utilizadas o decisiones de diseño tomadas durante el desarrollo.
- El proyecto debe reflejar **criterios mínimos de diseño**, como separación de responsabilidades, claridad en la lógica de control y uso adecuado de estructuras de datos.
- Se evaluará negativamente el uso excesivo o injustificado de código genérico de internet (plantillas, fragmentos no comprendidos, soluciones “copiadas y pegadas” sin adaptación ni entendimiento).
- Se valorará la **capacidad de defender decisiones técnicas** (por qué eligieron determinada estructura de datos, cómo gestionaron la persistencia, cómo validaron sus pruebas, etc.).

Instrucciones Finales

- Recuerde que **esto es un proyecto grupal.**
- **Siga las instrucciones de este documento** para evitar inconvenientes, en caso de tener dudas, consulte al profesor (en lugar de asumir cosas que pueden ser incorrectas). Los ayudantes de la materia podrían no ser la fuente de información más confiable cuando se trata de responder preguntas sobre el proyecto.
- Sea oportuno(a) en cuanto a sus preguntas: **No se atenderán consultas sobre el proyecto después del 20 de junio de 2023.** Tampoco se responderán preguntas que consulten sobre instrucciones indicadas explícitamente en este documento. Antes de hacer preguntas, **LEA ESTE DOCUMENTO DETENIDAMENTE.**

¡Muchos éxitos a todos!