

Objetivos de unidad

Este ejercicio busca trabajar los objetivos del **Capítulo 3: Estructuras lineales enlazadas** y **Capítulo 4: Mecanismos de reutilización y desacoplamiento**

Para la entrega del ejercicio, revise el documento publicado en moodle: [Normativas de MiniProyectos](#).

Enunciado del problema

Nombre del proyecto: Energía para la vida

La conocida vidente y mentalista Anina Montrev tiene tantas ideas de mejoramiento de vida, mensajes y motivaciones para cada momento del día que siente que se están perdiendo porque al no guardarlos, con el paso del tiempo, los ha ido olvidando y no quiere que esto pase más. Además, muchas veces percibe en las energías cuáles son las palabras adecuadas para decir en dos días, en una semana, puede sentir la necesidad energética de quienes le rodean a futuro y ya le ha pasado muchas veces que cuando llega el día indicado no percibe nuevamente las palabras necesarias. Su primera estrategia fue escribir sus percepciones en una libreta, pero pronto la llenó, se desorganizó y posteriormente no fue muy útil.

Ahora, Anina lo ha contactado y desea que su nueva herramienta sea una aplicación que le permita administrar todas sus ideas, así se asegurará de que no se perderán y las podrá tener a la mano siempre, sin importar cuántas sean.

Anina está muy emocionada con el proyecto, pero quiere hacer las cosas paso por paso para asegurarse de que el resultado final sea realmente el que tiene en mente. Ella ha dispuesto que por ahora sólo se creará la sección para mensajes y motivaciones.

La sección de mensajes y motivaciones debe permitir crear días, en ellos se debe especificar la fecha (actual o a futuro por las visiones de Anina) y el color recomendado para ese día. Dentro de cada día, la aplicación debe permitir la creación de mensajes, donde cada uno representará una motivación para el día. Cada mensaje debe tener su mensaje motivación y el autor (muchas veces Anina utiliza proverbios y sabiduría popular para comunicar sus ideas).

En la aplicación los días deben estar uno seguido de otro organizados por la fecha para la que fueron creados, en la aplicación sólo se mostrará la información del día con la fecha actual o la pasada más cercana a la actual. El avance en los días debe ser sólo hacia adelante, de las antiguas a las futuras. En cuanto a los mensajes, se encuentran conectados de tal forma que pueden avanzar hacia el mensaje siguiente o el mensaje anterior y cuando llega al último mensaje, éste se conecta con el primero.

En la interfaz gráfica Anina espera encontrar la pantalla dividida en dos, una sección para la creación de los días y sus mensajes, en donde se encontrarán las opciones para la creación de día, creación de mensajes y eliminación de mensajes. Una segunda sección deberá ser para visualizar información, donde se pueda indicar el día por la fecha y ver uno a uno los mensajes (si los hay) creados para esa fecha. Se deberá mostrar los datos del día requerido junto con el contenido de su primer mensaje, debajo del espacio de visualización, debe tener botones de izquierda o derecha para que al presionar vaya cambiando la información del mensaje en la vista por el siguiente mensaje que se desea ver, si el anterior o el siguiente respectivamente.

RESTRICCIONES DEL PROBLEMA

1. Esta unidad evalúa el uso de estructuras enlazadas definidas por el programador. Por lo tanto se espera que usted NO haga uso de las clases en Java que ya tienen implementada esta estructura.
2. Esta unidad introduce elementos gráficos que deben ser incluidos dentro de su solución.

Publicación: 17 de Febrero de 2014

Entrega 1: 35% - 24 de Febrero 2014

Usted debe entregar los siguientes ítems:

- **Planning - Planeación del proyecto:**

1. Cronograma de trabajo y los formatos PSP -Opcional-
Significa la asignación de responsabilidades específicas a cada uno de los integrantes del grupo, detalle de las tareas a realizar y fechas de revisión entre el grupo de estudiantes. Además de los formatos donde se registre su avance real en el desarrollo del proyecto.
2. Documento de requerimientos funcionales y no funcionales con formato Cupi2.

- **Design - Diseño:**

1. Diagrama de clases del mundo y de la interfaz.

- **Code - Código (Implementación):**

1. Implementación de la interfaz gráfica.

Rubrica del ejercicio (entrega 1)

Criterio	Puntos
Planning - Planeación del proyecto	Subtotal 40
<i>(Opc)Cronograma de trabajo y Formatos PSP 5 puntos por cada formato.</i>	20
<i>Documento de requerimientos funcionales y no funcionales con formato Cupi2.</i>	20
Design - Diseño	Subtotal 55
<i>Estándar de la documentación</i>	5
<i>Compleitud de los requerimientos</i>	10
<i>Diagrama de clases del mundo del problema</i>	25
<i>Diagrama de clases de la interfaz</i>	15
Code - Código (Implementación)	Subtotal 25
<i>Implementación de la interfaz gráfica</i>	25
TOTAL	120

Entrega 2: 65% - 8 de Marzo 2014

Usted debe entregar: los mismos ítems de la entrega 1 (si les ha realizado modificaciones o no) **más** los siguientes:

- **Code - Código (Implementación):**

1. Paquetes Java con la implementación de la solución (incluye el paquete test asociado a las pruebas).

- **Design - Diseño:**

1. Diagrama de clases del mundo, de la interfaz y de las pruebas.

- **Test - Pruebas:**

1. Diseño de los casos de prueba formato Cupi2 (escenarios y casos de prueba en formato Cupi2)

Rubrica del ejercicio (entrega 2)

Recuerde que en esta segunda entrega debe enviar su proyecto completo, con todas las partes detalladas a continuación. Si usted no modificó ninguna de las partes enviadas en la primera entrega, de igual modo debe enviarlas con esta segunda parte y serán calificadas nuevamente aunque esta nueva ponderación solamente será tomada en cuenta en la nota de la segunda entrega. La nota de la Primera Entrega no cambia. Por tanto, si usted actualizó alguna de estas partes y desea que le sean tomadas en cuenta estas mejoras, no olvide incluir los documentos actualizados (no los anteriores).

Criterio	Puntos
Planning - Planeación del proyecto:	Subtotal 10
<i>(Opc) Cronograma de trabajo y Formatos PSP</i>	5
<i>Documento de requerimientos funcionales y no funcionales con formato Cupi2.</i>	5
Design - Diseño	Subtotal 21
<i>Estándar de la documentación</i>	2
<i>Complejidad de los requerimientos</i>	2
<i>Diagrama de clases del mundo del problema</i>	6
<i>Diagrama de clases de la interfaz</i>	3
<i>Casos de uso y Diagrama de clases de las pruebas</i>	8
Code - Código (Implementación)	Subtotal 59
<i>Cumplimiento de los estándares de programación</i>	4
<i>Cumplimiento de la documentación</i>	4
<i>Cumplimiento de las restricciones del enunciado en cuanto a los temas de la unidad</i>	10
<i>Funcionalidad en general de la aplicación</i>	25
<i>Funcionalidad en general de las pruebas</i>	16
Test - Pruebas:	Subtotal 15
<i>Estándar de la documentación</i>	5
<i>Calidad de los escenarios y casos de prueba</i>	10
TOTAL	105