

|  |  |
| --- | --- |
| Objetivo | Aplicar orientación a Objetos. (primer alcance) Excepciones, fechas, arraylist |
| Publico | Estudiantes de técnicas de programación, Grupo 10:00 am |
| Valor | 15% |
| Fecha de entrega | Hasta el 6 de octubre |
| Entregables y observaciones | 1. Carpeta de proyecto Netbean, incluye imágenes, sonidos, archivos fuente .java.  2. Diseño conceptual  3. Diagrama de clases  4. Código  4. “Pantallazo” de Resultado, prueba de que la aplicación funciona, con datos  5. Incluya en un archivo .doc todos los supuestos que realizo Tener en cuenta que en la plataforma solo se puede subir un solo archivo, se sugiere realizar comprimido con toda la información a enviar. Cuando encuentre la palabra XXX esto significa que usted deberá dar valor a esta palabra. Por ejemplo Manejo de puntos, cada uno podrá colocar el nombre de taller que desee. |
| Descripción | **EJERCICIO 1**  Usando la clase JOptionPane para la captura y despliegue de la información, realice una aplicación que permita el registro de n personas en una arraylist y realizar:   1. Colocar formato a la fecha tanto en el ingreso como en la salida, este formato será elegido por el usuario y/o validado a su ingreso. 2. Hallar la cantidad de días que la persona ha vivido 3. Proyectar la cantidad de días que le faltan por vivir a esa persona 4. Calcular el costo de seguro de vida que hoy pagaría cada persona teniendo en cuenta la tabla <https://www.sura.com/documentos/canales/Tabla_Valores_V3_2013.pdf>, y que el aumento anual es del 3% (actualice la tabla a 2017). 5. Realice una implementación reusable que controle las excepciones que puedan presentarse en la captura de la información, en la gestión de fechas y del tiempo. 6. Informe a cada persona cuánto cuesta su seguro de vida. En caso de tomarlo registre la poliza, teniendo en cuenta la hora y fecha a partir de la cual empieza a regir y hasta cuando tiene vigencia.   Observaciones NO USE SWING. No use Vectores |
| Rubrica | |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | **NIVEL** | | | | | **Criterio** | **Excelente**  **(5)** | **Satisfactorio**  **(4)** | **Aceptable**  **(3)** | **Inadecuado**  **(2)** | | Diseño conceptual de la Solución propuesta | Considera todos los aspectos dados en el enunciado y adiciona nuevos detalles que mejoran el ejercicio y su desarrollo. | Considera todos los aspectos dados en el enunciado | Considera los elementos fundamentales del enunciado | Considera algunos elementos del enunciado, pero olvida elementos fundamentales | | Interfaz | Implementa una interfaz con alto sentido de la usabilidad, intuitiva y que responde ampliamente a las necesidades de interacción planteadas en el problema, adicionando elementos que investigo. | Implementa una interfaz con alto sentido de la usabilidad, intuitiva y que responde ampliamente a las necesidades de interacción planteadas en el problema | Implementa una interfaz usable, que responde las necesidades básicas de interacción planteadas en el problema. | No implementa una interfaz. | | La implementación soluciona el problema. (gestión de fechas, tiempo y excepciones) | La implementación soluciona totalmente los requerimientos expresados en el enunciado y realiza aportes | La implementación soluciona totalmente los requerimientos expresados en el enunciado. | La implementación soluciona los requerimientos fundamentales expresados en el enunciado. | La implementación soluciona algunos de los requerimientos expresados en el enunciado, o lo hace parcialmente. | | Documentación del código. | Realiza la documentación de cada componente, de manera clara y coherente. Aplica código Limpio | Realiza la documentación de cada componente. | Realiza la documentación de los componentes fundamentales como las clases y algunos métodos y variables | Realiza la documentación de algunos elementos, pero olvida elementos fundamentales en el entendimiento del código | |