**Objetivo**

Utilizar bibliotecas propias del lenguaje para realizar algunas tareas comunes y recurrentes.

**Previo**

**¿Qué es el API de Java?:**

Una API incluye clases, interfaces, paquetes y también sus métodos, campos y constructores.

Todas estas clases integradas dan beneficios al programador. Solo los programadores entienden cómo aplicar esa clase. Una interfaz de usuario ofrece la interacción básica del usuario entre el usuario y la computadora, de la misma manera, la API funciona como una interfaz de programa de aplicación que proporciona conexión entre el software y el consumidor. API incluye clases y paquetes que generalmente ayudan a un programador a minimizar las líneas de un programa.

**¿Qué es una utilería?:**

En Java varias clases e interfaces utilitarias vienen en el paquete java.util.

Los paquetes son el mecanismo que usa Java para facilitar la modularidad del código. Un paquete puede contener una o más definiciones de interfaces y clases, distribuyéndose habitualmente como un archivo. Para utilizar los elementos de un paquete es necesario importar este en el módulo de código en curso, usando para ello la sentencia import.

**¿Qué es una colección?**

Las colecciones en Java se utilizan en casi todas las aplicaciones. Java Collections Framework es una de las partes centrales del lenguaje de programación Java. El tutorial de Java Collections presentará en detalleJavaCollections Framework.

**¿Qué es un wrapper?**

La clase Wrapper en Java convierte o envuelve tipos de datos primitivos como objetos. Esto significa que podemos convertir valores primitivos en objetos y viceversa. Hay 8 tipos de datos primitivos que tienen una clase contenedora equivalente. Estas clases contenedoras amplían la clase Number, que es la clase principal.

Introducción:

trabajar en un problema de programación, normalmente se debe verificar si hay clases pre-construidas que satisfagan las necesidades del programa. Si existen esas clases, entonces hay que utilizarlas: “no tratar de reinventar la rueda”. Hay dos ventajas principales de usar clases preconstruidas: se puede ahorrar tiempo ya que no es necesario escribir nuevas; y el uso de clases preconstruidas también puede mejorar la calidad de los programas ya que han sido probadas completamente, depuradas y sometidas a un proceso de escrutinio para asegurar su eficiencia.

**Actividades:**

**Conocer el API del lenguaje.**

El api de java provee las herramientas necesarias para quien va a generar un código dentro del lenguaje, al investigar dentro de la documentación que contiene estás herramientas (“API”), nos puede otorgar el flujo con el cual deben ser utilizadas y su funcionalidad.

La documentación API:

<https://docs.oracle.com/en/java/javase/17/>

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Donde encontraremos toda la documentación.

**Reconocer las bibliotecas de uso común**

a clase System que pertenece a la librería java.lang (ésta es la librería estándar de Java y no es necesario importarla).

Commons Collections, amplía el framework de colecciones de Java con nuevas interfaces, implementaciones y utilities.

Apache Commons CSV, para la lectura y escritura de archivos en formato CSV.

Commons Math, librería de componentes matemáticos y estadísticos para abordar los problemas más comunes no disponibles en Java.

Commons IO, librería de utilities para ayudar en el desarrollo de funcionalidad Input/ Output, facilita la ejecución de operaciones Input/ Output .

**Implementar colecciones.**

Se implementaron las colecciones dentro de la práctica, en la imagen se Una colección representa un grupo de objetos. Esto objetos son conocidos como elementos. Cuando queremos trabajar con un conjunto de elementos, necesitamos un almacén donde poder guardarlos. En Java, se emplea la interfaz genérica Collection para este propósito.

HashSet: esta implementación almacena los elementos en una tabla hash. Es la implementación con mejor rendimiento de todas pero no garantiza ningún orden a la hora de realizar iteraciones.

TreeSet: esta implementación almacena los elementos ordenándolos en función de sus valores. Es bastante más lento que HashSet. Los elementos almacenados deben implementar la interfaz Comparable.

LinkedHashSet: esta implementación almacena los elementos en función del orden de inserción. Es, simplemente, un poco más costosa que HashSet.

En el ejemplo del código impartido en clase se puede apreciar su funcionamiento

Captura de pantalla de computadora

Descripción generada automáticamente

Aquí se aprecia como se utilizan las colecciones mencionadas anteriormente

**Actividades Extra**

**Realizar un diccionario con 5 palabras usando Hashtable e imprimir todos los elementos**

Captura de pantalla de computadora

Descripción generada automáticamente

Se anexa el código para que se aprecie el funcionamiento

**Realizar una agenda con 5 registros guardando nombre de persona y su cumpleaños usando Hashtable y calendar e imprimir todos los elementos**

Texto

Descripción generada automáticamente

Se anexa el código para que se aprecie el funcionamiento

**Conclusiones:**

La práctica nos sirvió para conocer las diferentes herramientas que maneja java para poder codificar óptimamente y se pueda agilizar la creación del código.