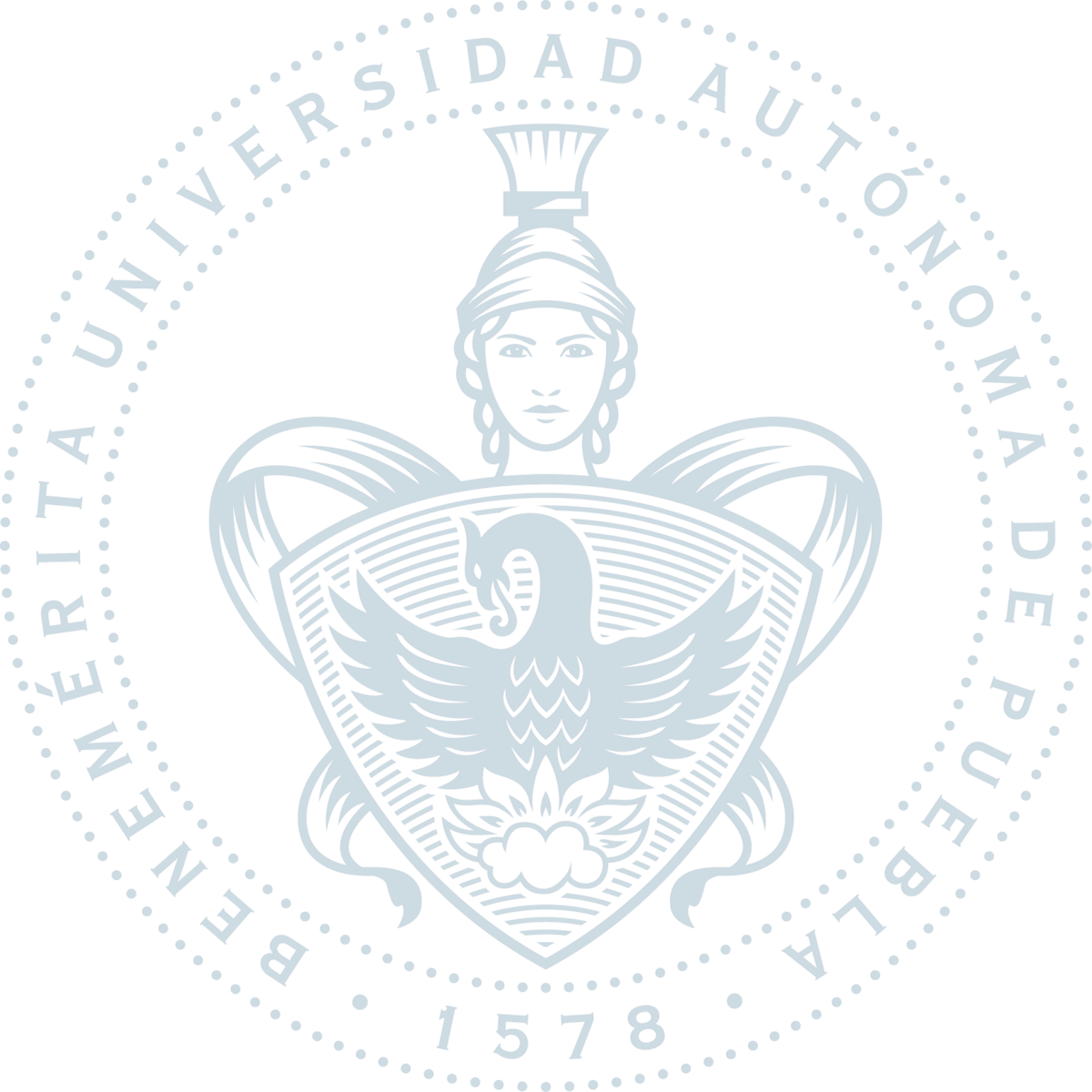
**BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA**

BUAP

** FACULTAD DE CIENCIAS DE LA COMPUTACIÓN**

**INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN**

**PROGRAMACIÓN 2**

**“PRACTICA 17A. REPORTE RECURSIVIDAD”**

**PROFESOR:**

**DRA. CARMEN CERÓN GARNICA.**

**EQUIPO:  
BRENDA LIZETH COCA GARCIA**

**MATRICULA:201917154**

**BRENDA AILED RODRIGUEZ COLIS**

**MATRICULA:201910848**

**MIGUEL CARREON VAZQUEZ**

**MATRICULA: 201915389**

**IRVYN XICALE CARRERA**

**MATRICULA: 201963582**

**INTRODUCCIÓN**En java podemos crear y leer archivos ya sean archivos binarios, archivos de txt o archivos de objetos.

**PROGRAMA PROCESAR ARCHIVOS**

**DIAGRAMA UML**

**Diagrama

Descripción generada automáticamente**

**CLASE CONVERTIR**

import java.io.\*;

public class Convertir {

    String contenido = "";

    String convertido = "";

    public void leerArchivo(){

        // lee archivo

        try {

            BufferedReader bf = new BufferedReader(new FileReader("archivo.txt"));

            String bfRead;

            while((bfRead = bf.readLine()) != null){

                contenido = bfRead;

            }

        } catch (IOException er) {

            System.out.println("Error: " + er.getMessage());

        }

    }

    public void procesarArchivo(){

        int espacios = 0;

        for(int i=0; i<contenido.length(); i++){

            if(contenido.charAt(i) == ' '){

                espacios++;

                if(espacios < 2){

                    convertido+=contenido.charAt(i);

                }

            } else

            espacios = 0;

            if(contenido.charAt(i)>='a' && contenido.charAt(i)<='z' || contenido.charAt(i)>='A' && contenido.charAt(i)<='Z'){

                convertido+=contenido.charAt(i);

            }

        }

        System.out.println(convertido);

    }

    public void imprimirArchivo(){

        // imprime archivo

        try {

            FileWriter Registro = new FileWriter("archivo\_Final.txt");

            Registro.write(convertido);

            Registro.close();

        } catch (IOException e) {

            System.out.println("Error: " + e.getMessage());

        }

    }

}

**CLASE EJECUTA**

public class Ejecuta {

    public static void main(String[] args){

        Convertir c = new Convertir();

        c.leerArchivo();

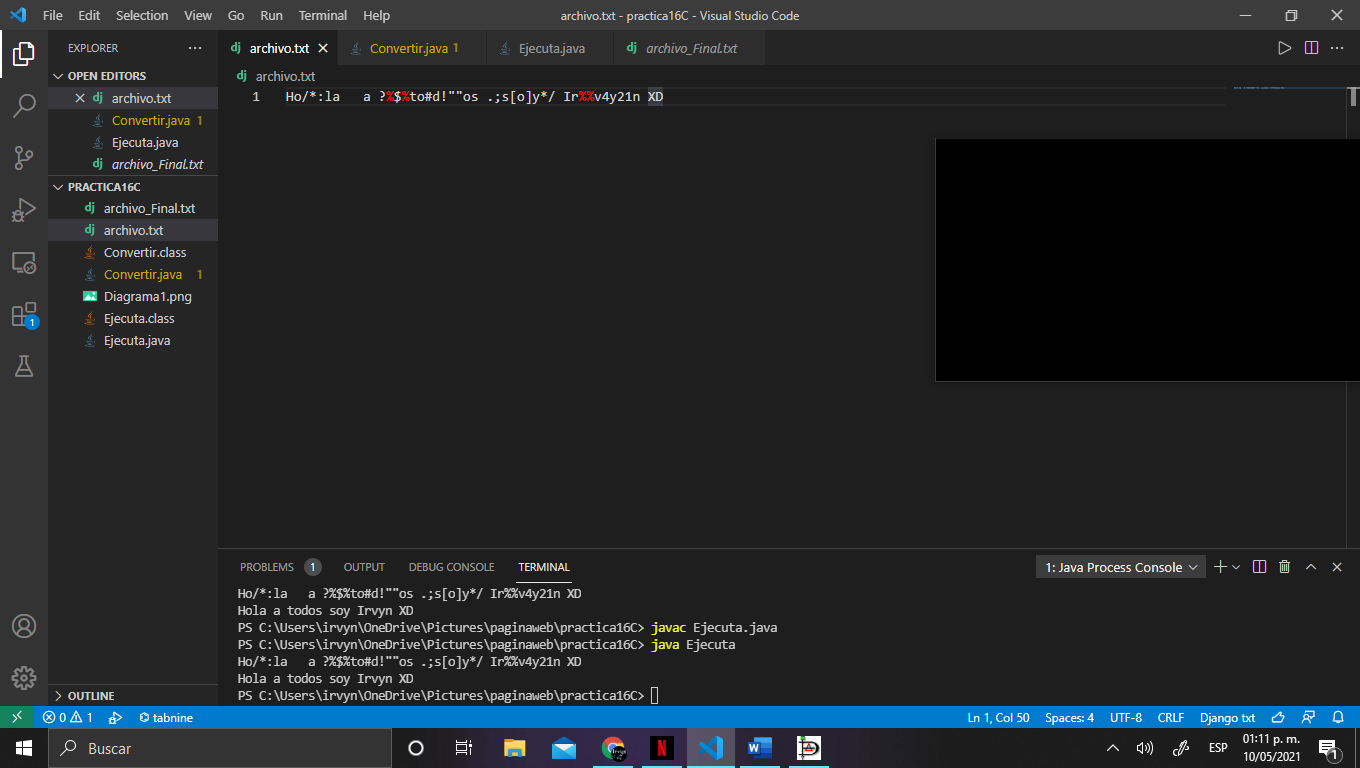
        c.procesarArchivo();

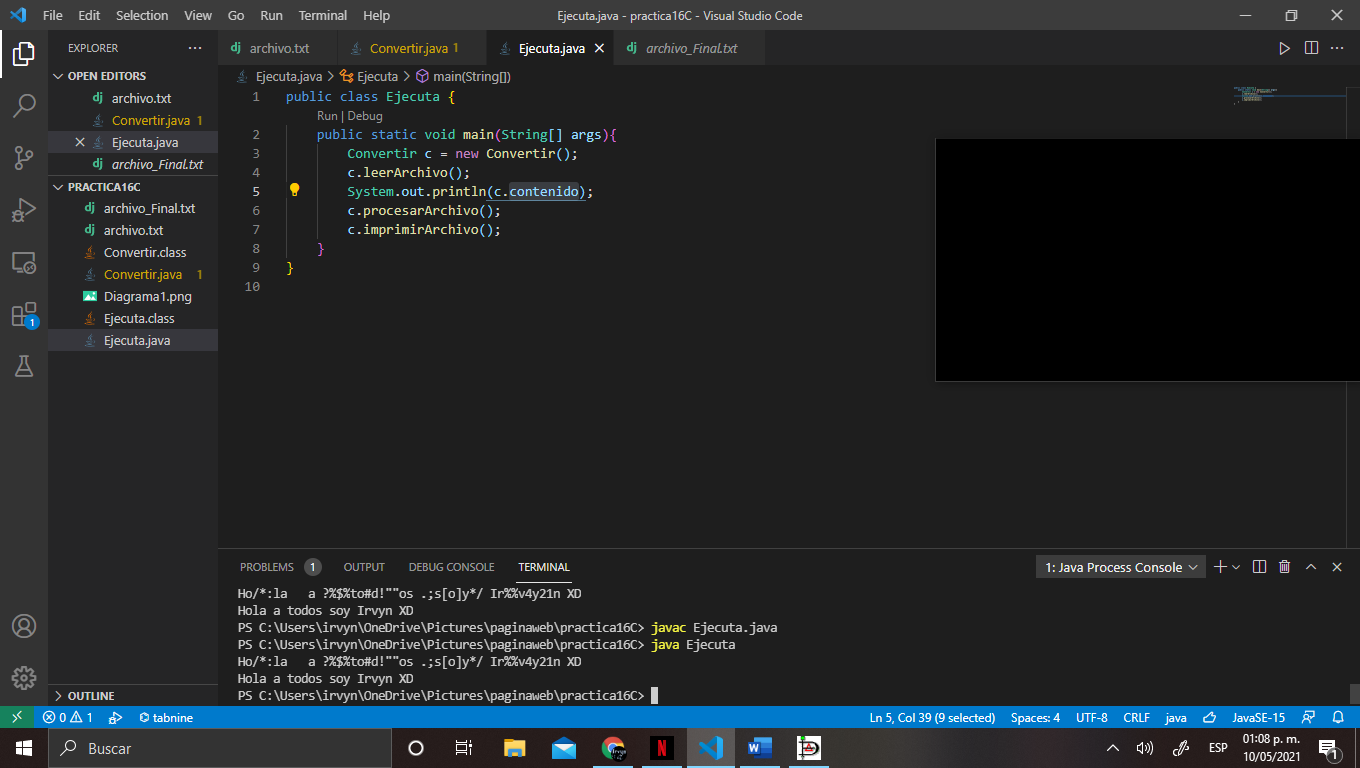
        c.imprimirArchivo();

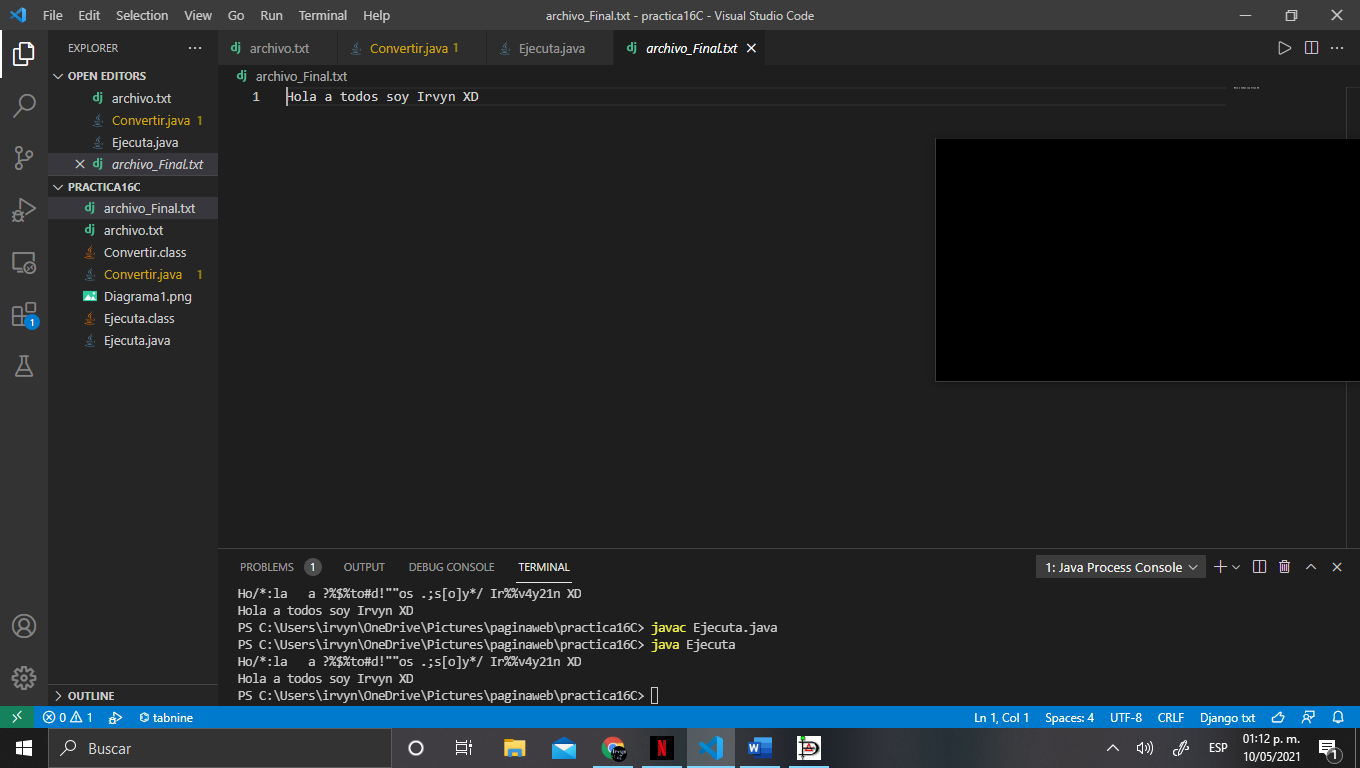
    }

}

**PRUEBA DE ESCRITORIO**







**CONCLUSIÓN**

Los archivos son muy importantes y un gran recurso que podemos usar en java y si logramos manejar los archivos ampliaremos nuestro conocimientos en java.

**REFERENCIAS BIBLIOGRÁFICAS**

* Fagg, A.H.(2017). GUI Programming: Components. Consultado el día 12 de agosto de 2020 en: <https://www.cs.ou.edu/~fagg/classes/cs2334/lecture/GUI-Components.pdf>
* Horstmann, C. S., & Cornell, G. (2013). Core Java Volumen I Fundamentals. New Jersey: Prentice Hall.
* Joyanes Aguilar, L., & Zahonero Martínez, I. (2011). Programación en Java 6. Algoritmos y programación orientada a objetos. Cd. de México: Mc Graw Hill.
* Poo, D., Kiong, D., & Ashok, S. (2008). Object-Oriented Programming and Java. Singapore: Springer.
* Sznajdleder, P. A. (2016). Java a fondo. Alfaomega.
* Liang, Y. D. ( 2014 ). Introduction to Java Programming, Comprehensive Version. Pearson
* Malik D. S. (2012 ). Java Programming: From Problem Analysis to Program Design, Fifth Edition. Cengage Learning
* GeeksforGeeks. (Última edición: 20 de Julio de 2020) , GeeksforGeeks. https://www.geeksforgeeks.org/types-of-recursions .Consultado de 30 de julio del 2020

**PROGRAMACION II**

**RUBRICA HOJA DE EVALUACIÓN DE PRACTICAS**

**MATRICULA: 201963582 FECHA: 9-MAYO-2021**

**NOMBRE: Irvyn Xicale Cabrera NO. PRACTICA: 16C INDIVIDUAL ( ) COLABORATIVA(x)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Deficiente** | **Suficiente** | **Bueno** | **Excelente** | **Calificación Obtenida** | |
| **CRITERIOS** | **1-5.9** | **6‐7.9** | **8‐9** | **9.1‐10** |
| **%** | **Puntos** |
| **CONOCIMIENTO TEORICO**  **20%** | Conocimiento deficiente de los fundamentos teóricos de POO y no puede aplicarlos en el diseño de clases. | Conocimiento confuso de los fundamentos teóricos de POO y el diseño de las clases y relaciones es incompleto. | Conocimiento claro de los fundamentos teóricos POO, pero requiere mejorar el modelado de las clases, sus métodos  y sus relaciones entre clases. | Dominio del Conocimiento de los fundamentos teóricos POO y puede aplicarlos de forma completa en el modelado de las clases, métodos y todas las relaciones entre clase. |  |  |
| **EJECUCCIÓN DE LA PRACTICA**  **30%** | No puede realizar la práctica ya que desconoce el entorno de trabajo y desarrollo de la práctica en lenguaje UML y Java. | Realiza la práctica de forma incompleta ya que desconoce el entorno de trabajo del lenguaje UML y Java | Realiza la práctica, pero requiere mejorar en el manejo del entorno de trabajo del lenguaje de programación (sintaxis y semántica) | Realiza la práctica de forma correcta y completa, demuestra dominio del entorno de trabajo del lenguaje de programación (sintaxis y semántica). |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **SOLUCIÓN DE LA PRACTICA**  **40%** | No puede generar las soluciones o programas a los problemas planteados ya que no posee el dominio teórico y práctico del modelado y el lenguaje de programación. | Propone soluciones confusas o programas incompletos a los problemas planteados, ya que carece del dominio del modelado y lenguaje de programación | Genera soluciones con poca profundidad y los programas no están orientados de acuerdo a los problemas solicitados, por lo cual no tiene un dominio profundo de la temática y del Lenguaje. | Genera soluciones con profundidad y los programas son correctos de acuerdo a los problemas planteados, por lo cual demuestra un dominio de la temática y del Lenguaje de Programación. |  |  |
| **ACTITUD DE APRENDER Y COLABORAR EN EQUIPO DE TRABAJO**  **10%** | No posee una actitud proactiva para un aprendizaje autónomo y no le gusta participar y trabajar en equipo. | Posee una actitud propositiva para un aprendizaje autónomo, participa, pero no le gusta trabajar en equipo. | Posee una actitud propositiva logrando un aprendizaje autónomo, colaborativo, le gusta trabajar en equipo, pero requiere mejorar su participación y portaciones de forma profunda. | Posee una actitud proactiva logrando un aprendizaje autónomo, participa con propuestas concretas y profundas, le gusta trabajar en equipo y asume su responsabilidad dentro para lograr éxito del equipo |  |  |
| **Total** | | | | |  |  |  |

