

**Unidad didáctica: Colecciones de datos**  
**Autora: Cristina Moreno Ruiz**

Concurso oposición de estabilización  
de Secundaria y RE (20/12/2022)

Especialidad: Informática





## ÍNDICE

|   |    |
|---|----|
| 1 INTRODUCCIÓN.....                                   | 2  |
| 2 NORMATIVA.....                                      | 2  |
| 3 CONTEXTUALIZACIÓN DE LA UNIDAD DIDÁCTICA.....       | 3  |
| 3.1 Relación con unidades previas.....                | 3  |
| 3.2 Relación con unidades didácticas posteriores..... | 4  |
| 3.3 Relación con otros módulos.....                   | 4  |
| 4 JUSTIFICACIÓN DE LA UNIDAD DIDÁCTICA.....           | 5  |
| 5 OBJETIVOS DE APRENDIZAJE.....                       | 6  |
| 6 CONTENIDOS.....                                     | 6  |
| 7 METODOLOGÍA.....                                    | 6  |
| 8 RECURSOS DIDÁCTICOS.....                            | 8  |
| 9 EVALUACIÓN Y CALIFICACIÓN.....                      | 9  |
| 9.1 De los alumnos.....                               | 9  |
| 9.2 De la práctica docente.....                       | 11 |
| 10 ATENCIÓN A LA DIVERSIDAD.....                      | 11 |
| 11 BIBLIOGRAFÍA.....                                  | 13 |
| 12 ÍNDICE DE IMÁGENES.....                            | 15 |
| 13 ÍNDICE DE TABLAS.....                              | 15 |
| 14 ANEXO I.....                                       | 16 |

## 1 INTRODUCCIÓN

El objetivo de este documento, es elaborar una guía para el proceso de enseñanza/aprendizaje de la Unidad Didáctica (UD) **Colecciones de datos**, del módulo Programación.

Este módulo forma parte del Ciclo Formativo de Grado Superior *Desarrollo de Aplicaciones Web (DAW)*, con código **IFCS03**, cuya duración es de 2000 horas, perteneciendo a la familia profesional de Informática y Comunicaciones, con referente europeo: **CINE-5b**.

El módulo profesional de Programación se imparte en el **primer curso**, con código **0485**, tiene una duración de 205 horas repartidas en 8 horas semanales y una equivalencia en 14 créditos ECTS.

He elegido esta UD para la oposición, porque es mi segundo curso impartiendo este módulo y en los materiales que utilizo del CEED (Centre Específic d'Educació a Distància de la Universidad de Valencia), considero que no están suficientemente desarrolladas las Colecciones de Datos.

## 2 NORMATIVA

El **R.D. 686/2010 de 20 de mayo** establece el título para el Ciclo y fija sus enseñanzas mínimas y el **Decreto 1/2011 de 13 de enero**, fija su currículo para la Comunidad de Madrid.

A nivel nacional, recientemente se ha publicado la **Ley Orgánica 3/2022, de 31 de marzo**, de ordenación e integración de la Formación Profesional y la **Ley Orgánica 3/2020, de 29 de diciembre**, Ley Orgánica de Modificación de la Ley Orgánica de Educación (LOMLOE), que aplicaremos en los evaluación y adaptación a la diversidad.

Puede consultarse el resto de normativa que regula la FP en <https://todofp.es/comunidad-docente/normativa.html>

### 3 CONTEXTUALIZACIÓN DE LA UNIDAD DIDÁCTICA

El lenguaje utilizado para introducirles en la programación es Java.

#### 3.1 Relación con unidades previas

Hemos dividido los contenidos del módulo en catorce unidades didácticas, siguiendo los contenidos indicados en la normativa autonómica y estatal.

Tal como puede verse en la siguiente tabla, las Colecciones de Datos se explican en la tercera evaluación, después de trabajar objetos, clases, conceptos avanzados de clases (herencia, súper y subclases, interfaces y clases abstractas) y excepciones:

| UNIDADES DIDÁCTICAS  | Competencia Profesional/ Personal/Social | Horas     | Evaluación |
|--|--|-----------|------------|
| U.D.1 Introducción a la programación                             | e  | 10        | 1ª         |
| U.D.2 Identificación de los elementos de un programa informático | e  | 10        | 1ª         |
| U.D.3 Uso de estructuras de control                              | e  | 30        | 1ª         |
| U.D.4 Introducción a la orientación a objetos                    | e  | 10        | 1ª         |
| U.D.5 Utilización de objetos                                     | e  | 15        | 1ª         |
| U.D.6 Desarrollo de clases                                       | e  | 30        | 2ª         |
| U.D.7 Aplicación de las estructuras de almacenamiento            | e  | 20        | 2ª         |
| U.D.8 Utilización avanzada de clases                             |  | 10        | 2ª         |
| U.D.9 Control y manejo de excepciones                            | m  | 10        | 2ª         |
| <b>U.D.10 Colecciones de datos</b>                               |  | <b>15</b> | <b>3ª</b>  |
| U.D.11 Lectura y escritura de información                        |  | 15        | 3ª         |

| UNIDADES DIDÁCTICAS                                    | Competencia Profesional/ Personal/Social | Horas      | Evaluación |
|--|--|------------|------------|
| U.D.12 Introducción a la programación con Swing        |  | 10         | 3ª         |
| U.D.13 Gestión de bases de datos relacionales          | d  | 10         | 3ª         |
| U.D.14 Mantenimiento de la persistencia de los objetos | d  | 10         | 3ª         |
| <b>** Total horas</b>                                  |  | <b>205</b> |            |

*Tabla 1: Temporalización Unidades Didácticas*

Es importante que previamente los alumnos tengan claro:

- cómo trabajar con objetos, para poder gestionar colecciones, generalmente de objetos.
- Qué son superclases, herencia e interfaces, ya que Java define sus colecciones en base a esos conceptos.
- Cómo gestionar las excepciones que se puedan producir al trabajar con colecciones.

### **3.2 Relación con unidades didácticas posteriores**

Conocer y saber cómo utilizar las colecciones, será algo imprescindible, para poder trabajar con Ficheros y Bases de Datos, ya que el manejo de la información accedida en ambos sistemas, suele mapearse o trabajarse en colecciones.

### **3.3 Relación con otros módulos**

Así como en las unidades de Ficheros y BBDD, encuentro relación con contenidos vistos en otros módulos, como Sistemas Informáticos y Bases de Datos, no encuentro relación entre las Colecciones de Datos y el resto de módulos de primero. Veo más relación con módulos de Desarrollo del segundo curso, donde tendrán que gestionar Colecciones de Datos.

#### 4 JUSTIFICACIÓN DE LA UNIDAD DIDÁCTICA

Aunque en los contenidos del currículo a nivel nacional, ni siquiera en los mínimos, no se incluyen las colecciones de datos, no tiene sentido ignorarlas, ya que son herramientas imprescindibles en Programación.

Tal como se ve en la Programación Didáctica (PD), el sexto Resultado de Aprendizaje que se relaciona con esta UD, que viene definido en el **Real Decreto 686/2010, de 20 de mayo** es: “6. *Escribe programas que manipulen información, seleccionando y utilizando tipos avanzados de datos.*” y los criterios a evaluar son :

- b) Se han reconocido las librerías de clases relacionadas con **tipos de datos avanzados**.*
- c) Se han utilizado **listas** para almacenar y procesar información.*
- d) Se han utilizado **iteradores** para recorrer los elementos de las listas.*
- e) Se han reconocido las características y ventajas de cada una de la **colecciones de datos** disponibles.”*

Es evidente que entre los tipos avanzados de Java, podríamos hablar de ArrayList, HashMap... y está claro que hay que evaluar el uso de listas, iteradores y colecciones.

Incluyo esta UD en mi PD, porque en el **DECRETO 1/2011, de 13 de enero**, en el que se establece el currículo para la comunidad de Madrid, sí que se detallan los contenidos relacionados con las Colecciones de Datos:

- Tipos de colecciones (listas, pilas, colas, tablas...).
- Jerarquías de colecciones.
- Operaciones con colecciones. Acceso a elementos y recorridos.
- Uso de clases y métodos genéricos.

## 5 OBJETIVOS DE APRENDIZAJE

Para trabajar de una manera más lógica dichos contenidos, el orden de los objetivos sería el siguiente:

- comprender la jerarquía de colecciones,
- tipos de colecciones,
- operaciones con colecciones y
- uso de métodos comunes a colecciones.

## 6 CONTENIDOS

La relación entre objetivos de aprendizaje y contenidos es:

| Objetivos                               | Contenidos                                |
|---|---|
| Comprender la jerarquía de colecciones. | Collections Framework.                    |
| Tipos de colecciones.                   | ArrayList, LinkedList, HashMap y HashSet. |
| Operaciones con colecciones.            | Añadir elementos, buscar, eliminar...     |
| Uso de métodos comunes a colecciones.   | Entre otros Comparable y Comparator.      |

*Tabla 2: Relación entre objetivos de aprendizaje y contenidos*

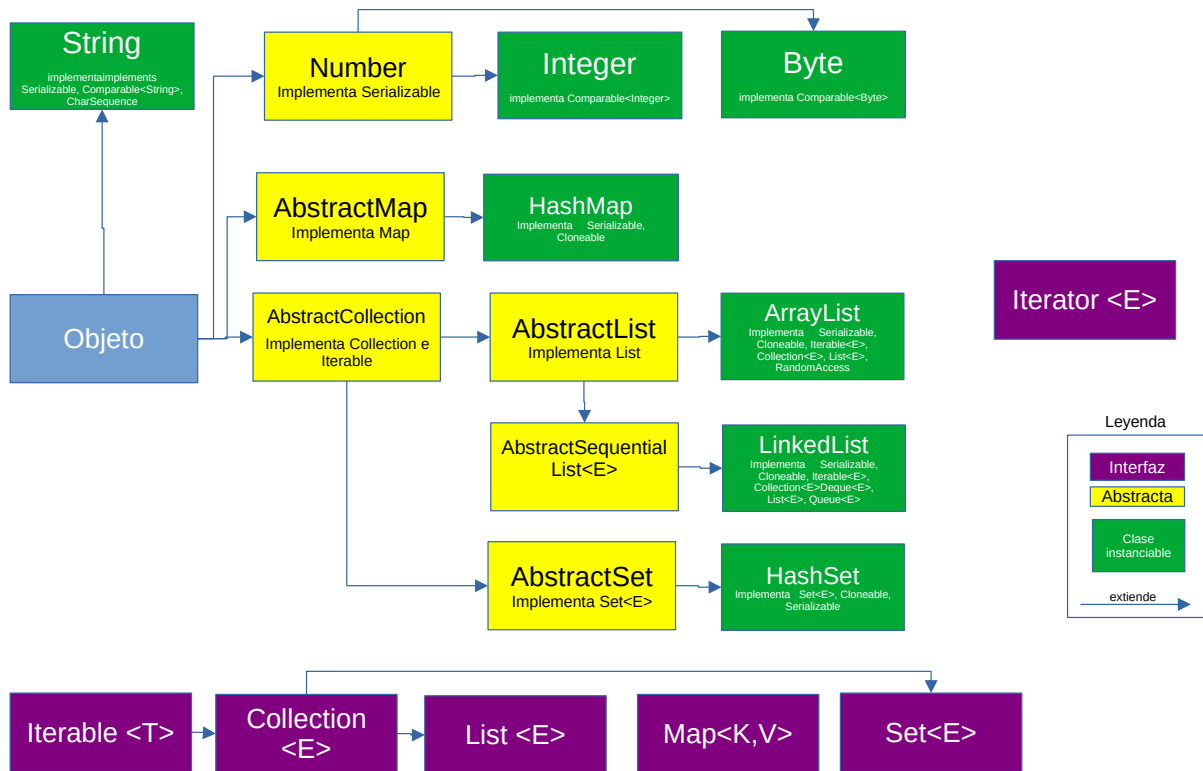
## 7 METODOLOGÍA

En la temporalización de la PD, se le ha asignado a esta UD 15 sesiones.

En la UD anterior, aprendieron a utilizar ArrayList, para poder trabajar con colecciones de objetos, sin que profundizáramos en el origen de la clase ArrayList, lo tratamos como otra clase a su servicio, igual que el Scanner.

Para iniciar la UD repasaremos los tipos primitivos que conocen y los no primitivos como String, Integer y ArrayList.





Se dedicarán sesiones de tres horas a cada una de las colecciones, explicando:

- declaración y creación,
- sus métodos de llenado, acceso, manipulación
- y recorrido.

La secuencia y temporalización de las sesiones sería:

| Contenidos   | Sesiones |
|--|----------|
| Tipos primitivos y no primitivos. Collections Framework. Iterator. | 3        |
| LinkedList   | 3        |
| HashMap  | 3        |
| HashSet  | 3        |
| Comparable y Comparator.   | 3        |

Tabla 3: Temporalización de la UD

## 8 RECURSOS DIDÁCTICOS

Los materiales utilizados en clase están subidos al aula virtual del centro, organizados en una sección por cada UD.

Incluyen teoría, presentaciones, enunciados de ejercicios y tareas para que los alumnos realicen sus entregas o calificarles la corrección en clase.

El aula virtual es <https://aulavirtual32.educa.madrid.org/ies.lopedevega.madrid/course/view.php?id=1182> y la contraseña de invitado “oposición”.

Los materiales también están dentro de mi repositorio en GitHub:

<https://github.com/cmorenoruiz/ApuntesProgramacion>, es un fork de los materiales utilizados por el CEED de la Universidad de Valencia. Desde hace dos años los estoy modificando y mejorando, bajo licencia CC-BY-SA. En concreto el enlace a los contenidos de esta UD es:

<https://github.com/cmorenoruiz/ApuntesProgramacion/tree/master/Unidad%20transversal%20Colecciones>

## 9 EVALUACIÓN Y CALIFICACIÓN

### 9.1 De los alumnos

Como se comenta de manera más detallada en la PD del módulo, la evaluación es continua, que el alumno perderá con 31 faltas de asistencia, al tener el módulo 205 horas. El contenido de esta unidad se evalúa tanto en la tercera evaluación, como en las convocatorias ordinaria y extraordinaria. Siendo los criterios de cada evaluación los siguientes:

| Criterio  | Peso |
|---|------|
| Pruebas específicas de evaluación                       | 70%  |
| Ejercicios prácticos entregados y/o corregidos en clase | 30%  |

*Tabla 4: Relación entre objetivos de aprendizaje y contenidos*

Los ejercicios deberán ser entregados dentro del plazo establecido por la profesora. Cada ejercicio será evaluado de 0 a 10 y será necesario tener, en cada una de ellos, una calificación superior o igual a 5 para que el alumno pueda aprobar la evaluación. Al final de la evaluación, se calculará la nota media de todas estos ejercicio. En este apartado se pueden incluir trabajos de investigación. Las actividades que haya que corregir en clase serán evaluadas por la profesora con una nota de 0 a 10, no siendo en este caso necesario haberlas aprobado para poder aprobar la evaluación.

Todas las actividades/ejercicios no tienen porque tener el mismo peso, en el cálculo de la nota media de actividades de una evaluación.

En los exámenes, que son eminentemente prácticos y siempre necesitan definir y manejar varias colecciones de datos, incluyo una rúbrica que tiene en cuenta la completitud de los requisitos solicitados. Por ejemplo, si tienen que mostrarme el numero de Autoras por país

de residencia, dada una colección de Autoras que habrán creados ellos, les evaluaré los siguientes criterios:

- La definición y creación de estructura de almacenamiento datos.
- Cálculo y muestra de los datos solicitados.

En ambos casos aplicando los siguientes niveles de calificación a cada criterio:

| Nivel  | Peso |
|--|------|
| La tarea está incompleta, no se ha realizado con arreglo al guion establecido o contiene errores graves. | 0    |
| La tarea no se puede compilar o ejecutar. La tarea se ha copiado total o parcialmente.                   | 0    |
| La tarea no obtiene los resultados esperados o está resuelta de forma incorrecta.                        | 3    |
| Tarea contiene errores leves que pueden causar resultados erróneos en ciertas ocasiones                  | 6,5  |
| Tarea ejecuta, pero no cumple los requisitos solicitados. O no maneja excepciones.                       | 9    |
| Tarea ejecuta pero realiza tareas innecesarias o faltan algunos requisitos.                              | 10,5 |
| Algún fallo mínimo, que no impide la ejecución.  | 13   |
| Tarea resuelta correctamente con arreglo al guion establecido y a los requisitos solicitados.            | 15   |

*Tabla 5: Niveles de calificación para un criterio o requisito de la rúbrica*

Se puede consultar en la PD que, tras lo expuesto en los puntos anteriores, solo cabe destacar que la calificación final del módulo será el resultado de calcular la media ponderada de las calificaciones obtenidas en las tres evaluaciones, valoradas la primera en un 30%, la segunda en un 30% y la tercera en un 40%. Para poder realizar esta media es condición indispensable que la tercera evaluación esté aprobada. En caso contrario será la menor entre dicha media y 4.

Al estar esta UD dentro de la tercera evaluación, la única manera de recuperarla es presentarse a la convocatoria ordinaria. Los criterios de las convocatorias ordinarias y extraordinarias se encuentran en el PD.

## **9.2 De la práctica docente**

A finales de curso incluyo una encuesta anónima en el aula virtual, que enlaza a un formulario en <https://formularios.educa.madrid.org/>, en el que recojo sus opiniones, incluyendo una pregunta abierta en la que pueden sugerirme otras preguntas a hacer en la encuesta o cualquier comentario.

También completo el cuestionario que rellenamos cada profesor en el Departamento.

## **10 ATENCIÓN A LA DIVERSIDAD**

Mi experiencia en clase, me ha hecho darme cuenta de que hay tantas maneras de acercarse a los nuevos conceptos y a practicar, como alumnos hay en clase. Suelo tener al menos uno e incluso dos alumnos con dislexia o problemas de lectura, para los que mi material escrito no aporta nada. Suelen preferir explicaciones en clase con imágenes o vídeos.

Todos los alumnos disponen en el aula virtual de los materiales teóricos completos, en formato ODT y PDF, así como de las presentaciones que utilizo en clase para facilitar la explicación. En ambos documentos incluyo enlaces a vídeos y recursos alternativos. Pueden verlos en sus pantallas, a la vez que yo los utilizo en clase.

A principios de curso les explico las diferentes condiciones que pueden dificultar su aprendizaje: dislexia, daltonismo, pérdida auditiva, problemas de visión... les cuento las que yo tengo y cómo suplo estas dificultades. Les pido que si tienen alguna condición que les dificulte el aprendizaje, me informen, para facilitarles material adaptado.

Por ahora no he tenido alumnos daltónicos en clase, aunque los diagramas que yo creo, como el que he incluido en esta UD, se basan en colores de alto contraste, para facilitar la identificación de sus componentes. Entre mis propuestas futuras, está tener un material pensado para imprimir en tonos grises o texturas y otro en color, para ver en pantalla, porque he observado que hay alumnos que imprimen los apuntes teóricos y los diagramas quedan muy oscuros.

A los alumnos con dislexia les enseño

- a: utilizar la fuente OpenDyslexic, que pueden usar para formatear mis documentos e integrar en Eclipse,
- así como a instalar un plugin en el navegador, que les permite leer los recursos textuales con dicha fuente.

El examen lo imprimo con ambas fuentes.

Este es un texto de ejemplo, en el que se utiliza OpenDyslexic 3, puede verse que la asimetría vertical y horizontal, facilita que los alumnos detecten cuando están girando las letras y no duden en qué están leyendo. Se diferencian sin problemas la b, d, q y p.

Otros alumnos necesitan ver la solución completa, para tener una idea general del problema, así que incluyo enlaces a repositorios en GitHub con las soluciones a los ejercicios y exámenes.

La solución a los exámenes, la grabo en vídeo, para que los que necesitan más tiempo, puedan reproducirla las veces que necesiten y al ritmo que prefieran.

Además, al poner ejemplos de soluciones en clase, tanto si soy yo, como un alumno/a el que muestre su código, proyectamos nuestra pantalla en videoconferencia, para que los demás puedan verla de cerca y no tengan dificultades por la calidad de la imagen del proyector del aula.

## **11 BIBLIOGRAFÍA**

Real Decreto 686/2010, de 20 de mayo, por el que se establece el título de Técnico Superior en Desarrollo de Aplicaciones Web y se fijan sus enseñanzas mínimas. Boletín Oficial del Estado, 143, de 12 de junio de 2010, 50009-50073.

[https://www.boe.es/diario\\_boe/txt.php?id=BOE-A-2010-9269](https://www.boe.es/diario_boe/txt.php?id=BOE-A-2010-9269)

Orden EDU/2887/2010, de 2 de noviembre, por la que se establece el currículo del ciclo formativo de Grado Superior correspondiente al título de Técnico Superior en Desarrollo de Aplicaciones Web. Boletín Oficial del Estado, 273, de 11 de noviembre de 2010, páginas 94471 a 94500.

[https://www.boe.es/diario\\_boe/txt.php?id=BOE-A-2010-17329](https://www.boe.es/diario_boe/txt.php?id=BOE-A-2010-17329)

DECRETO 1/2011, de 13 de enero, del Consejo de Gobierno, por el que se establece para la Comunidad de Madrid el currículo del ciclo formativo de grado superior correspondiente al título de Técnico Superior en Desarrollo de Aplicaciones Web. Boletín Oficial de la Comunidad de Madrid, 25, de 31 de enero de 2011, 9-34.

<https://www.bocm.es/2011-01-31-25200120110007>

Ley Orgánica 3/2020, de 29 de diciembre, por la que se modifica la Ley Orgánica 2/2006, de 3 de mayo, de Educación. Boletín Oficial del Estado, 340, de 30 de diciembre de 2020, 122868-122953. <https://www.boe.es/boe/dias/2020/12/30/pdfs/BOE-A-2020-17264.pdf>

Ley Orgánica 3/2022, de 31 de marzo, de ordenación e integración de la Formación Profesional. Boletín Oficial del Estado, 78, de 1 de abril de 2022.

<https://www.boe.es/eli/es/lo/2022/03/31/3/con>

Java:

- <https://docs.oracle.com/javase/8/docs/api/>
- <https://www.w3schools.com/java>

Tipos primitivos:

- <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html>
- [https://www.w3schools.com/java/java\\_data\\_types.asp](https://www.w3schools.com/java/java_data_types.asp)

Byte <https://docs.oracle.com/javase/8/docs/api/java/lang/Byte.html>

Integer: <https://docs.oracle.com/javase/8/docs/api/java/lang/Integer.html>

String <https://docs.oracle.com/javase/7/docs/api/java/lang/String.html>

Enum: [https://www.w3schools.com/java/java\\_enums.asp](https://www.w3schools.com/java/java_enums.asp)

Collections Framework:

<https://docs.oracle.com/javase/8/docs/technotes/guides/collections/index.html>

LinkedList:

- <https://docs.oracle.com/javase/8/docs/api/java/util/LinkedList.html>
- [https://www.w3schools.com/java/java\\_linkedlist.asp](https://www.w3schools.com/java/java_linkedlist.asp)

HashMap:

- [https://www.w3schools.com/java/java\\_hashmap.asp](https://www.w3schools.com/java/java_hashmap.asp)



- <https://docs.oracle.com/javase/8/docs/api/java/util/HashMap.html>

HashSet:

- [https://www.w3schools.com/java/java\\_hashset.asp](https://www.w3schools.com/java/java_hashset.asp)
- <https://docs.oracle.com/en/java/javase/12/docs/api/java.base/java/util/HashSet.html>

Comparable:

<https://docs.oracle.com/en/java/javase/20/docs/api/java.base/java/lang/Comparable.html>

Comparator:

<https://docs.oracle.com/en/java/javase/20/docs/api/java.base/java/util/Comparator.html>

OpenDyslexic:

- Web del autor: <https://abbiecod.es/>
- Fuentes y recursos: <https://opendyslexic.org/>
- Plugin Chrome:  
<https://chrome.google.com/webstore/detail/opendyslexic-for-chrome/cdnapgjopgaggbmfgbiinmmbdcglnam>
- Plugin Firefox: <https://addons.mozilla.org/en-GB/firefox/addon/opendyslexic-for-firefox/>

Repositorio público de esta UD: [https://github.com/cmorenorui/UD\\_y\\_PD\\_a\\_entregar](https://github.com/cmorenorui/UD_y_PD_a_entregar)

## 12 ÍNDICE DE IMÁGENES

Figura 1: Jerarquía de colecciones (Collections Framework) Fuente: Cristina Moreno Ruiz

..... 7

## 13 ÍNDICE DE TABLAS

Tabla 1: Temporalización Unidades Didácticas..... 4

Tabla 2: Relación entre objetivos de aprendizaje y contenidos..... 6

Tabla 3: Temporalización de la UD..... 8

|  |    |
|--|----|
| Tabla 4: Relación entre objetivos de aprendizaje y contenidos.....               | 9  |
| Tabla 5: Niveles de calificación para un criterio o requisito de la rúbrica..... | 10 |

## 14 ANEXO I

Posibles ejercicios a realizar:

HashMap:

- Crea un programa que solicite al usuario que ingrese una lista de nombres y sus respectivos números de teléfono. Utiliza un HashMap para almacenar los nombres como claves y los números de teléfono como valores. Luego, permite al usuario buscar un nombre y mostrar el número de teléfono correspondiente.
- Implementa un programa que cuente la frecuencia de cada palabra en una oración. Utiliza un HashMap para almacenar las palabras como claves y sus frecuencias como valores. Muestra el resultado al final.

HashSet:

- Crea una clase "Persona" con atributos como nombre, edad y ciudad. Crea un conjunto HashSet de objetos de tipo Persona y agrega varias instancias a él. Luego, recorre el conjunto y muestra los nombres de las personas mayores de 30 años.
- Implementa un programa que encuentre los elementos duplicados en un arreglo de enteros. Utiliza un HashSet para almacenar los elementos duplicados y muéstralos al final.

LinkedList:

- Crea un programa que simule una lista de reproducción de música. Utiliza una LinkedList para almacenar las canciones en orden. Implementa funciones para agregar canciones al inicio y al final de la lista, reproducir la siguiente canción y mostrar la lista de reproducción completa.

- Implementa una cola utilizando una LinkedList. Crea funciones para agregar elementos a la cola y para sacar elementos de la cola en orden.

Comparable:

- Crea una clase "Producto" con atributos como nombre, precio y cantidad. Implementa la interfaz Comparable en la clase Producto para que los productos se puedan ordenar por precio de forma ascendente. Crea una lista de objetos Producto y ordénala utilizando el método sort() de Collections.
- Crea una clase "Estudiante" con atributos como nombre, edad y promedio. Implementa la interfaz Comparable en la clase Estudiante para que los estudiantes se puedan ordenar por promedio de forma descendente. Crea una lista de objetos Estudiante y ordénala utilizando el método sort() de Collections.

Comparator:

- Crea una clase "Libro" con atributos como título, autor y año de publicación. Implementa un comparador personalizado llamado "ComparadorLibros" que ordene los libros por año de publicación de forma descendente. Crea una lista de objetos Libro y ordénala utilizando el comparador personalizado.
- Crea una clase "Pelicula" con atributos como título, director y duración. Implementa un comparador personalizado llamado "ComparadorPeliculas" que ordene las películas por duración de forma ascendente. Crea una lista de objetos Pelicula y ordénala utilizando el comparador personalizado.