

## Análisis Laboratorio #3

1. Los requisitos funcionales del programa que debe de considerar.
  - a. Publicar posts
    - i. Pedir que tipo de post quieren
    - ii. Pedir los datos del post
    - iii. Agregar post a lista
  - b. Comentar posts
    - i. Pedir por cual método quiere buscar el post
    - ii. Pedir que post quiere agregar el comentario
    - iii. Pedir el comentario
    - iv. Agregar el comentario
  - c. Dar like a los posts
    - i. Pedir por cual método quiere buscar el post
    - ii. Pedir que post quiere agregar el like
    - iii. Agregar el like
2. Clases: El nombre de cada clase y para qué se usará en el programa, El nombre de cada una de las propiedades de cada clase y su propósito, El nombre de cada uno de los métodos de cada clase y su propósito.
  - a. Vista (interactúa con el usuario)
    - i. Propiedades:
      1. Scan: Scanner (Importamos la clase scanner)
    - ii. Métodos:
      1. Bienvenida(): void (Damos la bienvenida)
      2. menu(): int (Mostramos el menu)
      3. tipoPost(): String (Le pedimos el método para buscar un post)
      4. crearPostTexto(): String[] (Pedimos los datos necesarios para crear un post de tipo texto)
      5. crearPostImagen(): String[] (Pedimos los datos necesarios para crear un post de tipo imagen)
      6. crearPostAudio(): String[] (Pedimos los datos necesarios para crear un post de tipo audio)
      7. crearPostVideo(): String[] (Pedimos los datos necesarios para crear un post de tipo Video)
      8. crearPostEmoticon(): String[] (Pedimos los datos necesarios para crear un post de tipo emoticon)
      9. mostrarPosts(ArrayList posts):void (Mostramos todos los posts realizados)
      10. mostrarPost(Post posts): void (Mostramos un post en específico)
      11. ObtenerFecha(): LocalDate (Obtenemos por cual fecha se van a buscar los posts)
      12. ObtenerHashtags(): String (Obtenemos el hashtag que se va a buscar)

13. accion(): int (Pedimos que acción quiere realizar sobre la publicacion)
  14. comentario():String (Pedimos el comentario en el que se va a la publicacion)
  15. seleccionPost(): int (Pedimos sobre que post va a realizar la accion)
- b. Controlador (Es el programa principal)
- i. Propiedades:
    1. sistem: Sistema (Importamos la clase sistema)
    2. vista:Vista (Importamos la clase vista)
    3. opcion:int (guardamos el valor de la opción elegida)
  - ii. Métodos:
    1. main(args:String[]) : void (es donde todo ocurre)
- c. Sistema (Donde se analiza la información)
- i. Propiedades:
    1. Posts: ArrayList<Post> (Donde guardamos los posts que van publicando)
    2. numPost: int
  - ii. Métodos:
    1. crearPostTexto(String[] datos): void (Se usa para crear posts de tipo texto)
    2. crearPostImagen(String[] datos): void (Se usa para crear posts de tipo imagen)
    3. crearPostAudio(String[] datos): void (Se usa para crear posts de tipo audio)
    4. crearPostVideo(String[] datos): void (Se usa para crear posts de tipo video)
    5. crearPostEmoticon(String[] datos): void (Se usa para crear posts de tipo emoticon)
    6. getPosts():ArrayList (Se usa para obtener los posts publicados)
    7. agregarComentario(int num, String post): void (Se usa para agregar un comentario)
    8. agregarLike(int num): void (Se usa para agregar un like)
    9. acción(int num): void
- d. Post (Se usa como default para los posts)
- i. Propiedades:
    1. autor: String (Guardamos el autor)
    2. fecha: LocalDate (Guardamos la fecha)
    3. hora: LocalTime (Guardamos la hora)
    4. likes: int (Guardamos la cantidad de likes)
    5. Hashtags: String (Guardamos los hashtags)
    6. comentarios: ArrayList<String> (Guardamos los comentarios)
    7. tipo: String

8. ID: int
- ii. Métodos:
  1. Post(int ID, String autor, String Hashtags): Constructor (Es el constructor de la clase)
  2. agregarLike(): void (Se usar para agregar un like a la clase)
  3. agregarComentario(): void (Se usa para agregar un comentario)
  4. *toString()*: void (Se usa para desplegar un mensaje)
  5. getAutor(): String
  6. getFecha(): LocalDate
  7. getHora(): LocalTime
  8. getLikes(): int
  9. getHashtags(): String
  10. getComentarios(): ArrayList
  11. setTipo(): void
  12. getTipo(): String
  13. getID(): int
- e. Texto (Se usa como default para los posts de texto)
  - i. Propiedades:
    1. texto: String (Es donde almacenamos el texto)
  - ii. Métodos:
    1. Texto(int ID, String autor, String Hashtags, String texto): Constructor (Es el constructor de la clase)
    2. *toString()*: void (Sirve para desplegar un mensaje)
    3. getTexto(): String
- f. Multimedia (Se usa como default para los posts de multimedia)
  - i. Propiedades:
    1. url: String (Se almacena la url del mensaje)
    2. kilobytes: int (Se almacena cuanto pesa el archivo)
  - ii. Métodos:
    1. Multimedia(int ID, String autor, String Hashtags, String url, int kilobytes): Constructor (Constructor de la clase)
    2. *toString()*: void (Sirve para desplegar un mensaje)
    3. getUrl(): String
    4. getKilobytes(): int
- g. Emoticon (Se usa como default para los posts de Emoticon)
  - i. Propiedades:
    1. Emoticon: String (Se almacena el emoticon)
  - ii. Métodos:
    1. Emoticon(int ID, String autor, String Hashtags, String Emoticon): Constructor (Constructor de la clase)
    2. *toString()*: void (Sirve para desplegar un mensaje)
    3. getEmoticon(): String
- h. Imagen (Default para los posts multimedia con imagen)
  - i. Propiedades:

1. formato: String (Almacena el formato del video)
  2. Resolucion: int (Almacena la resolución del video)
- ii. Métodos:
  1. Imagen(int ID, String autor, String Hashtags, String url, int kilobytes, String formato, int resolucion): Constructor (Es el constructor de la clase)
  2. toString(): void (Despliega un mensaje)
  3. getFormato(): String
  4. getResolucion(): int
- i. Audio (Default para los posts multimedia con audio)
  - i. Propiedades:
    1. sampleRate: int (Guardamos el sample rate del audio)
    2. bitDepth: int (Guardamos el bit Depth del audio)
  - ii. Métodos:
    1. Audio(int ID, String autor, String Hashtags, String url, int kilobytes, int sampleRate, int bitDepth): Constructor (Es el constructor de la clase)
    2. toString(): void (Despliega un mensaje)
    3. getSampleRate(): int
    4. getBitDepth(): int
- j. Video (Default para los posts multimedia con video)
  - i. Propiedades:
    1. frameRate: int (Guardamos los framerates del video)
  - ii. Métodos:
    1. Video(int ID, String autor, String Hashtags, String url, int kilobytes, int frameRate): Constructor (Es el constructor de la clase)
    2. toString(): void (Se usa para desplegar un mensaje)
    3. getFrameRate(): int