Comparando ambas tareas, mayoría de métodos cambiaron debido a la implementación de listas

| Que hace | Métodos con arreglos clase Directorio | Métodos con Listas clase Directorio | Implementación con Listas |
| --- | --- | --- | --- |
| Mostrar la información completa de un contacto, con sólo proporcionar su nombre | busquedaN  busquedaN1#Auxiliar | BusquedaN | Se ajustó la implementación del método usando el iterator visto en clase  y no se requiere de un método auxiliar . |
| Mostrar la información completa de un contacto, dado su nombre y rol (alumno, profesor o director). | busquedaNR  busquedaNR1#Auxiliar | BusquedaNR | Se ajustó la implementación del método usando el iterator visto en clase  y no se requiere de un método auxiliar . |
| Método para buscar a una persona con su correo | busquedaC  busquedaC1#Auxiliar | busquedaC | Se ajustó la implementación del método usando el iterator visto en clase  y no se requiere de un método auxiliar . |
| Insertar una persona | insertper | (clase Lista)agregar | Este método funciona por el metodo agregar visto en clase el cual está en la clase Lista . |
| Busca y borra a una persona por un nombre dado. | borra\_persona | Eliminar los datos de un contacto, a partir del nombre.  No hay un método explícito. | Usaremos el método para eliminar el objeto de la lista combinado con el de búsqueda por nombre. |
| Busca y borra a una persona por un correo dado, | borra\_persona\_c | Método para borrar personas ingresando su correo.  No hay un método explícito. | Usaremos el método para eliminar el objeto de la lista combinado con el de búsqueda por correo |
| Actualiza los datos de una persona. | actualizar\_persona | Actualizar la información de alguno de los contactos, a partir del nombre. | Usaremos el método de búsqueda por nombre y el de sustitución de la clase Lista. |
| Muestra el directorio | imprimeArregloDir | Mostrar Directorio.  ###falta xD | En la implementación de listas se usa el iterator visto en clase.  asaff |
| Algoritmo de búsqueda | bubblesort | No hay | En la implementación de listas no es necesario usar un algoritmo de búsqueda |
| Mostrar sólo alumnos o sólo profesores, según la elección del usuario. Los contactos se deben mostrar ordenados de forma ascendente. | cuentatipo | BusquedaT | En la implementación de listas se usa el iterator visto en clase.  y no es necesario usar Bubble Sort. |
| Mostrar a todos los contactos con correo electrónico. La información se mostrará agrupada por categoría. | cuentacorr | BusquedaCC???’ | Combinando el método eliminar y de búsqueda por correo |
| Mostrar a todos los contactos de una carrera en particular. La información se mostrará agrupada por la categoría alumno. | cuentatipocarrAlum | BusquedaCarrAl | Se ajustó la implementación del método usando el iterator visto en clase. |
| Mostrar a todos los contactos de una carrera en particular. La información se mostrará agrupada por la categoría profesor. | cuentatipocarrProf | BusquedaCarrPr | Se ajustó la implementación del método usando el iterator visto en clase. |
| Mostrar a todos los contactos de una carrera en particular. La información se mostrará agrupada por la categoría coordinador. | cuentatipocarrCor | BusquedaCarrCor | Se ajustó la implementación del método usando el iterator visto en clase. |
| Mostrar a todos los contactos con un sueldo determinado. La información se mostrará agrupada por categoría (profesor o coordinador). | cuentatiposueldoCor | Filtrotipo  BusquedaSueCor | En la implementación de listas se usa el iterator visto en clase.  También requiere del método auxiliar Filtrotipo. |
| Mostrar a todos los contactos con un sueldo determinado. La información se mostrará agrupada por categoría.  (profesor o coordinador). | cuentatiposueldoProf | Filtrotipo  BusquedaSueProf | En la implementación de listas se usa el iterator visto en clase.  También requiere del método auxiliar Filtrotipo. |
|  | toString |  |  |
| Mostrar a todos los contactos con correo electrónico. La información se mostrará agrupada por categoría. | cuentatipo | BusquedaCC | Funciona con el método eliminar y de búsqueda por correo y con la implementación en listas usa el iterator visto en clase. |
| Es para convertir un archivo csv a una lista para poder usarlo dentro del programa(Lectura) | No hay | archivoALista | Se basa en el archivo subido por el profesor. |
| Permite leer las líneas de un archivo y almacenarlas en un arreglo por categoría (alumno, profesor o coordinador). | No hay | EscribirLista | Se basa en el archivo subido por el profesor. |

Como podemos ver, ,es mas optimo usar Listas pues anteriormente al usar arreglos teníamos que borrar todo y volver a hacerlo si queremos agrandar de tamaño nuestro directorio.

Ahora con Listas eso no pasa, pues no es necesario declarar el tamaño de la misma ya que su tamaño no es estático.Ya no se requiere bubble Sort.