Métodos implementados con la solución de arreglos	Métodos implementados con la solución de listas	Cambios
`__init__`	`__init__`	En lugar de construir el directorio como un arreglo de tamaño fijo, se construye a partir de una lista, usando la clase Nodo.
`insertar_nuevo_alumno`	`insertar_nuevo_alumno`	El cambio radica en que, en lugar de extender el directorio, se utiliza la función __agregar_a_la_lista para agregar a los alumnos a la lista directorio.
`insertar_nuevo_profesor`	`insertar_nuevo_profesor`	El cambio radica en que, en lugar de extender el directorio, se utiliza la función __agregar_a_la_lista para agregar a los profesores a la lista directorio.
`insertar_nuevo_coordinador`	`insertar_nuevo_coordinador`	El cambio radica en que, en lugar de extender el directorio, se utiliza la función __agregar_a_la_lista para agregar a los coordinadores a la lista directorio.
`extender`		Se retiró este método, pues servía para hacer más grande el arreglo.
`buscar_indice`		Se retiró este método, pues al ser una lista, no podemos recorrerla por medio de índices.
`esta_vacio`	`esta_vacio`	Este cambió, pues en vez de definir el número de personas como 0, se usa la clase Lista y se verifica que el primer sea None o no.
`mostrar_persona`	`mostrar_persona`	El cambio principal radica en que no se uso un for para recorrer el directorio, sino que se usaron los nodos para hacerlo.
`tamanio_csv`		Se eliminó, pues no se utiliza.

lectura_csvs	lectura_csvs	El principal cambio es que ya no se ocupa un arreglo para almacenar los datos del archivo .csvs, sino que ahora se utiliza una lista dinámica.
escritura_csvs	escritura_csvs	La primera versión usa un arreglo fijo y construye manualmente las cadenas de atributos, mientras que la segunda emplea una lista dinámica e iteradores, simplificando la escritura al aprovechar el formato __str__ de cada objeto.
__particion		Se retiró, pues al ser una lista dinámica, no podemos usar QuickSort.
ordenar_directorio	ordenar_directorio	La primera versión aplica el algoritmo Quick Sort recursivamente en un arreglo, mientras que la segunda crea una nueva lista ordenada utilizando un comparador, iterando sobre la lista original y reorganizándola sin usar recursión.
__compare_strings		Se removió del directorio y se trasladó a un archivo separado llamado Comparadores.py.
nombre_comparador		Se removió del directorio y se trasladó a un archivo separado llamado Comparadores.py.
mostrar_informacion_contacto	mostrar_informacion_contacto	El principal cambio es que la primera versión trabaja con un arreglo fijo y ordena los contactos antes de buscarlos, mientras que la segunda usa una lista enlazada, omite la ordenación previa, y recorre la lista hasta encontrar el contacto, adaptándose mejor a la estructura dinámica.
eliminar_contacto	eliminar_contacto	El principal cambio es que la primera versión ocupaba un for para recorrer el directorio,

		mientras que la segunda versión funciona a través de los nodos.
menu_actualizar_alumno	menu_actualizar_alumno	No hubo cambios.
menu_actualizar_profesor	menu_actualizar_profesor	No hubo cambios.
menu_actualizar_coordinador	menu_actualizar_coordinador	No hubo cambios.
actualizar_alumno	actualizar_alumno	El principal cambio es que la primera versión busca al alumno por índice en un arreglo fijo y luego actualiza sus datos, mientras que la segunda utiliza una lista enlazada, buscando el nodo correspondiente para acceder y modificar el alumno de manera más dinámica.
actualizar_profesor	actualizar_profesor	El principal cambio es que la primera versión busca al profesor por índice en un arreglo fijo, mientras que la segunda utiliza una lista enlazada, buscando el nodo correspondiente para acceder y modificar el profesor de manera más dinámica.
actualizar_coordinador	actualizar_coordinador	El principal cambio es que la primera versión busca al coordinador por índice en un arreglo fijo, mientras que la segunda utiliza una lista enlazada, buscando el nodo correspondiente para acceder y modificar el coordinador de manera más dinámica.
mostrar_contactos_por_sueldo	mostrar_contactos_por_sueldo	El principal cambio es que la primera versión busca al contacto por índice en un arreglo fijo, mientras que la segunda utiliza una lista enlazada, buscando el nodo correspondiente para acceder y mostrar al contacto de manera más dinámica.
contiene	contiene	El mayor cambio radica en que en la primera versión se recorre el

		directorio por medio de un for, mientras que la segunda ocupa los nodos y el método contiene, propio de la interfaz Lista.
`eliminar`	`eliminar`	El principal cambio es que la primera versión ocupaba un for para recorrer el directorio, mientras que la segunda versión solo utiliza el método eliminar de la interfaz de Lista.
`eliminar_cel`	`eliminar_cel`	El principal cambio es que la primera versión ocupaba un for para recorrer el directorio, mientras que la segunda versión funciona a través de los nodos y el método eliminar de la interfaz de Lista..
`eliminar_email`	`eliminar_email`	El principal cambio es que la primera versión ocupaba un for para recorrer el directorio, mientras que la segunda versión funciona a través de los nodos y el método eliminar de la interfaz de Lista..
`buscar_indice_cum`		Se retiró, pues ya no se usa.
`buscar_indice_cel`		Se retiró, pues ya no se usa.
`buscar_contacto_celular`	`buscar_contacto_celular`	El principal cambio es que la primera versión busca al contacto por índice en un arreglo fijo, mientras que la segunda utiliza una lista enlazada, buscando el nodo correspondiente para acceder y mostrar a la persona.
`buscar_contacto_cum`	`buscar_contacto_cum`	El principal cambio es que la primera versión busca al contacto por índice en un arreglo fijo, mientras que la segunda utiliza una lista enlazada, buscando el nodo correspondiente para acceder y mostrar al contacto.
`__str__`	`__str__`	El principal cambio es que la primera versión utiliza un arreglo

		y ordena los contactos antes de categorizarlos y mostrarlos, mientras que la segunda recorre una lista enlazada, adaptándose a su estructura dinámica y verificando si está vacía antes de construir y retornar la cadena con la información de los contactos.
`mostrar_contactos_con_email`	`mostrar_contactos_con_email`	El principal cambio es que la primera versión ordena el arreglo y usa una copia profunda para eliminar contactos sin email antes de mostrar los resultados, mientras que la segunda recorre una lista enlazada, buscando y agrupando los contactos con un email específico sin crear copias, adaptándose mejor a la estructura dinámica..
`mostrar_contactos_por_carrera`	`mostrar_contactos_por_carrera`	El principal cambio es que la primera versión busca al contacto por índice en un arreglo fijo, mientras que la segunda utiliza una lista enlazada, buscando el nodo correspondiente para acceder y mostrar al contacto.
`mostrar_alumnos_o_profesores`	`mostrar_alumnos_o_profesores`	El principal cambio es que la primera versión busca al contacto por índice en un arreglo fijo, mientras que la segunda utiliza una lista enlazada, buscando el nodo correspondiente para acceder y mostrar al contacto.
	`__agregar_a_lista`	Método único que agrega una nueva persona a la lista de contactos.
	`buscar_nodo`	Método único que busca el nodo en el que se encuentra almacenada una persona.
	`buscar_nodo_nomcompleto`	Método único que busca el nodo en el que se encuentra almacenada una persona.