

# Proyecto Integrador de la Asignatura

## Introducción

El proyecto se formula para integrar los conceptos que comprenden la asignatura. Tiene en cuenta aspectos pedagógico-didácticos inherentes como: temática, contexto (área, asignatura, nivel, grupo, actores, período de tiempo, competencias a desarrollar), estrategias didácticas para el aprendizaje colaborativo y que permitan crear una organización fuerte e innovadora en la enseñanza y aprendizaje, se planificaron propuestas que sean pertinentes para los actores educativos con quienes se desarrollará el proyecto. En cuanto a la problemática planteada y la selección y adecuación de contenidos se corresponden a la significación lógica, representatividad, actualidad, relevancia y contextualización. De esta manera se permita intervenir con idoneidad y conocimiento, concretar las acciones, motivar el análisis, la reflexión y la acción en favor del aprendizaje de los estudiantes de Licenciatura en Sistemas de Información que cursan en el presente cuatrimestre la asignatura Algoritmo y Estructuras de Datos 2.

A fin de realizar el seguimiento, acompañamiento y comunicación con todos y cada uno de los equipos de alumnos que intervienen en el proyecto, se habilitará una sección en el espacio virtual de la asignatura.

## Objetivos de aprendizaje

Se pretende que los alumnos aprendan a:

- Construir programas que requieran el uso de estructuras de datos compuestas, manipulación de archivos, el uso de recursividad, construir y desarrollar tipos de datos abstractos, incorpore una primera aproximación de programación de placas tipo Arduino, seleccionar y usar estratégica algoritmos a fin de lograr la optimización y eficiencia de los módulos creados.
- Combinar las estructuras y las técnicas aplicadas.
- Adquirir nociones básicas para autogestionar sus proyectos de programación de software.

## Competencias por desarrollar

- Competencias tecnológicas:
  - **Identificar, formular y resolver problemas.**  
Resultado: Un programa que resuelva uno o más problemas determinados
  - 
  - **Utilizar de manera efectiva técnicas y herramientas de aplicación para desarrollar software**  
Resultado: Manejo de metodologías de programación (programación de a pares) y herramientas como Google Meet, Jitsi, Zoo, Trello o similares. Uso del IDE.
- Competencias sociales, políticas y actitudinales:

- **Desempeñarse de manera efectiva en equipos de trabajo.**

Resultado: Utilización de la herramienta Trello e incrementos de trabajo presentados en cada etapa del proyecto

- **Aprender en forma continua, autónoma y de manera colaborativa.**

Resultado: apropiarse de diversos conceptos de estructuras de datos compuestas, como ser: listas, listas enlazadas y desarrollar habilidades para aplicar técnicas de programación como el corte de control sobre archivos de datos y recursividad, primera aproximación en la programación de placas tipo Arduino.

## Destinatarios

Alumnos del curso 2021 de la Asignatura: Algoritmos y Estructuras de Datos II (1er año - 2do cuatrimestre) de la Carrera: Licenciatura en Sistemas de Información (FaCENA)

## Estructura de la experiencia

La experiencia se llevará a cabo a lo largo del cuatrimestre mediante el desarrollo de un proyecto integrador. Además, el alumno deberá recurrir a los materiales dispuestos por la cátedra y la bibliografía del programa de la asignatura.

El grupo de cuatro (4) alumnos trabajara con la técnica de “a pares”, y resolviendo actividades de complejidad creciente los estudiantes irán avanzando en el proceso de “aprender haciendo”.

A fin de aprender con temas lúdicos y motivadores, los estudiantes diseñarán una solución software, crearán un programa en C para resolver un problema sobre alguno de los siguientes temas u otros que puedan proponer:

- Software de los personajes de cómics
- software de juegos educativos para PC
  - Exploro los rincones de mi casa
  - Aprendo a sumar y restar
  - Menú de juegos didácticos (Ej. algunas opciones del juego “peque TIC”)
- software de juegos interactivo para niños
  - software para juego que recoge letras y arma palabras
  - software para juego para los más pequeños para conocer el reino animal a través de imágenes.
  - Software para juego para unir puntos y descubrir imágenes
- Otros

Para la asignatura “Algoritmos y Estructuras de Datos 2” de la carrera Licenciatura en Sistemas de Información. La resolución se realizará en etapas a lo largo del cuatrimestre a medida que se vaya avanzando en los temas propios de la materia que tienen que ver con estructuras de datos y técnicas de programación necesarias que un profesional de Sistemas debe conocer, para poder diseñar e implementar programas de computación.

En la sección dentro del espacio virtual de la asignatura los equipos dispondrán de:

Información necesaria para la comprensión del proyecto; disponibilidad de contenido temático (apuntes, ensayos, paper, videos, link, etc. para el desarrollo del proyecto; tutoriales de las herramientas a utilizar; fechas de presentación de los avances; formas de evaluación de los avances y de la presentación; comunicación asincrónica con el equipo docente que acompaña el desarrollo del proyecto; fechas, si es necesario, de encuentros sincrónicos a realizar; fechas y horas para coordinar las tutorías que fueran necesarias para cada equipo.

## Metodología del trabajo

Las estrategias didácticas a implementar facilitarán el aprendizaje colaborativo y permitirán que los estudiantes aprendan a utilizar programas como [Trello](#) y metodologías ágiles como [Programación de a pares](#) para su uso en situaciones propias del desempeño de un profesional de [Sistemas](#).

En el diseño y desarrollo de estrategias se conjugaron los objetivos y las actividades lo que permitió definir los criterios de acción. Entre las estrategias didácticas se consideran: lectura y análisis del material teórico/práctico de la asignatura, trabajo en equipo, uso de herramientas de software, programación e implementación de estructuras de datos y técnicas de programación, estrategias de autorreflexión, autoevaluación y/o metacognición (rutina de pensamiento).

Para evaluar se implementará una rúbrica que se adjunta más adelante.

Como recursos se utilizarán: material teórico práctico y videos explicativos generados por los docentes de la asignatura

Para realizar este trabajo se utilizará [Programación de a pares \(Pair Programming\)](#), que es una metodología de desarrollo de software donde dos personas (programadores) situados en una sola computadora desarrollan código de una aplicación software. El equipo de programadores se concentra en el código que se está escribiendo.

Como cada equipo este compuesto por cuatro alumnos, pertenecientes todos a la misma comisión de práctica, estos se irán alternando en la sustitución del “par de programadores” que propone la técnica.

Cada equipo de trabajo, cuatro alumnos, deberá designar a uno de los alumnos para que desempeñe el rol de “líder coordinador” del equipo. Este alumno será el referente del equipo ante los docentes.

Algunas de las actividades que deberá desempeñar este alumno líder, entre otras, son:

- Colaboración y comunicación permanente con el equipo de docente
- Coordinar la planificación de las distintas fases del proyecto
- Dirección y coordinación de recursos
- Gestión de las relaciones externas (docentes y otros pares) del proyecto
- Gestión de riesgos derivados de la ejecución
- Capacidad de intervención

En la filosofía de **Programación de a pares (Pair Programming)**, en general, el responsable de “escribir” el código se conoce como el **conductor**, mientras que el otro /s desarrollador/es es llamado **revisor**, este último revisa continuamente lo que se está codificando o escribiendo, funcionando como un observador activo, y debe concentrarse en el plan de acción. Idealmente, ambos miembros del equipo deberían cambiar sus funciones para obtener mejores resultados. Durante todo el tiempo que han estado juntos, el par de dúo se mantiene en contacto, permitiendo que cada integrante del equipo participe y ayude a delinear la dirección del código.

Esta filosofía se adaptará a un equipo de cuatro alumnos, donde, por ejemplo, uno asumirá el rol de conductor (escribir el código), mientras que los otros tres desarrolladores (alumnos), serán los revisores del código.

Durante el desarrollo del proyecto, el equipo de trabajo, si cree conveniente y/o necesario, alternara los roles.

### Programación de a pares “remota” o a distancia

Cada desarrollador trabajará desde una computadora diferente. Se utilizará algún tipo de aplicación que les permita comunicarse y compartir la pantalla del ordenador, como: Google meet, zoom, jitsi.org, ciscowebex, etc.

A través de alguna de esas apps, el conductor comenzará compartiendo su pantalla para que el revisor pueda ver lo que está escribiendo.

Para gestionar las tareas por hacer y coordinar las videoconferencias una opción es utilizar la herramienta Trello disponible en <https://trello.com> .

A lo largo del desarrollo del proyecto cada equipo ira construyendo un informe de lo actuado en cada etapa. Dicho informe será utilizado en todos y cada una de las presentaciones parciales y la presentación final.

Dicho informe debe contener:

1. Nombre del proyecto
2. Participantes
3. Descripción de proyecto (software a desarrollar)
4. Justificación del proyecto
5. Planificación del Proyecto (Relacionar con el ciclo de vida seleccionado).
6. Detalle de cada etapa
  - a. Descripción de las actividades
  - b. Código (en lenguaje “C”) de los módulos realizado
  - c. Herramientas utilizadas
7. Conclusiones

### Descripción de las etapas y actividades que se desarrollaran

### **Etapas 1: Semana de repaso**

1. Crear un tablero en Trello. En el mismo deberán ir cargando las tareas por hacer, del modo que les parezca más adecuado para realizar un seguimiento del trabajo: las tareas por hacer o pendientes, las tareas en ejecución, las tareas completas y realizadas.  
En dicho tablero se deberán incluir como usuarios (email) los cuatro (4) alumnos mas los docentes de la comisión de practica que le corresponda al equipo de trabajo.
2. Crear un programa en lenguaje “C” que permita generar un archivo binario de datos en el contexto del software que desarrollaran. Para ello será necesario que se permita ingresar por teclado los datos que fuesen necesarios. El programa deberá construirse con “estructuras” y contener las funciones necesarias.

### **Etapas 2: Estructuras compuestas: listas (implementación estática mediante arrays)**

1. Modificar el procedimiento de “ingresar datos” para que a medida que se ingresen los datos, se almacene la estructura que contiene los datos en una **lista estática** de datos.
2. Escribir las funciones necesarias para que el usuario pueda verificar los datos que posteriormente se grabaran en el archivo:
  - Mostrar datos (recorrer la lista)
  - Eliminar un dato (¿a partir de la selección del código de identificación del dato o del nombre?)
3. Modificar la función que graba los datos para que, en lugar de grabar de un dato a la vez, los grabe a medida que recorra la lista de datos.

### **Etapas 3: Estructuras compuestas enlazadas (implementación dinámica mediante listas enlazadas)**

1. Modificar la función de “IngresarDatos ()” para que a medida que se ingresen los datos, se almacene la estructura que contiene los datos en una **lista enlazada de datos** en lugar de la lista estática programada en la etapa anterior.
2. Modificar las funciones necesarias para que el usuario pueda verificar los datos que posteriormente se grabaran en el archivo (Mostrar lista de datos, eliminar un dato, etc.) para que apliquen a las funciones correspondientes a las listas.
3. Modificar la función que graba los datos para que recorra la lista enlazada de datos, en lugar de recorrer la lista estática y a medida que acceda a un nodo con contenido correspondiente a un dato lo grabe en el archivo de datos.

### **Etapas 4: Corte de control**

1. Agregar una función `informeDeDatos ()` que permita generar un reporte donde fuese necesario aplicar una estrategia de “corte de control”.

### **Etapas 5: Recursividad**

Modificar al menos dos funciones para que apliquen la técnica de la recursión:

### **Etapas 6: Programación de placas tipo Arduino**

Crear un módulo, asociado al software que se está desarrollando, que permita, programando placa de Arduino, simular algún comportamiento, situación de los datos ingresados.

El equipo deberá prever en el módulo que:

- Respecto del hardware el módulo debe contener, al menos, el uso de leds, display, sensores y botones.
- Programar las funciones necesarias para simular el comportamiento y/o situación “pensada”.
- Para almacenar las combinaciones de ataques, el grupo deberá proponer la estructura de datos más adecuada y justificar su propuesta.

### **Etapas 7: (optativa) Selección de estrategia algorítmica para optimizar y hacer más eficiente al algoritmo**

De las actividades realizadas en cualquiera de las etapas del 2 al 6, el equipo deberá seleccionar algún módulo a fin de utilizar alguna estrategia algorítmica para lograr mayor eficiencia u optimización del mismo. Luego deberá implementar la propuesta y comprobar la optimización y/o eficiencia lograda.

## Entregas

Las fechas de presentación de cada una de las entregas (parciales y final) se informará a los equipos en tiempo y forma a través de la sección “Proyecto Integrador” del espacio virtual de la asignatura. La exposición de la presentación del avance no deberá superar los 15 minutos. Deberá desarrollarse un documento (archivo magnético), para entregar; la documentación completa realizada.

La presentación y exposición de la primera entrega se hará utilizando alguna herramienta a tal fin (ej.: power point, genially, canva, lucidchart, etc.).

## Parciales

### **Primera entrega (avance)**

Al finalizar las actividades de la segunda etapa (listas, implementación estática mediante arrays) y antes de iniciar las actividades de la tercera (estructuras compuestas enlazadas) etapa es necesario que cada equipo presente el avance y conclusión de lo realizado.

Dicha presentación deberá contener la primera parte del informe del proyecto integrador, a saber,

1. Nombre del proyecto
2. Participantes
3. Descripción de proyecto (software a desarrollar)
4. Justificación del proyecto
5. Planificación del Proyecto (Relacionar con el ciclo de vida seleccionado).
6. Detalle de las etapas 1 y 2.
  - a. Descripción de las actividades
  - b. Código (en lenguaje “C”) de los módulos realizado
  - c. Herramientas utilizadas

### **Segunda entrega (avance)**

Al finalizar las actividades de la quinta (Recursividad) etapa y antes de iniciar las actividades de la sexta etapa (Programación de placas tipo Arduino) es necesario que cada equipo presente el avance y conclusión de lo realizado.

Dicha presentación deberá contener la primera parte del informe del proyecto integrador, presentado en la primera entrega, más la descripción de lo siguiente, a saber,

1. Detalle de las etapas 3, 4 y 5.
  - a. Descripción de las actividades
  - b. Código (en lenguaje “C”) de los módulos realizado
  - c. Herramientas utilizadas

## Final

### Tercera entrega (final)

Al finalizar las actividades de la séptima (optimización y eficiencia) etapa (optativa) se procederá a la entrega y presentación final del proyecto integrador. Dicha presentación deberá contener lo anteriormente presentado, más:

1. Detalle de las etapas 6 y 7.
  - a. Descripción de las actividades
  - b. Código (en lenguaje “C”) de los módulos realizado
  - c. Herramientas utilizadas
2. Conclusiones

## Evaluación, revisión y reflexión

### Evaluación

Las evaluaciones se realizarán de manera iterativa e incremental a medida que se realicen entregas parciales (3) del proyecto mediante una rúbrica, y de la presentación final del proyecto integrador.

Se realizarán mediante encuentros inter-grupales, que favorecerán la interacción de estudiantes, con la mediación del docente.

Se realizarán exposiciones para chequear avances y reflexionar acerca del desarrollo del trabajo en equipo, de manera de tomar distancia y analizar el camino recorrido.

Finalizada cada una de las etapas de actividades, se implementará la coevaluación entre equipos de trabajo con la finalidad de abrir paso al diálogo y la confrontación académica, el desarrollo de actitudes y habilidades argumentativas sobre los diseños y modificación de funciones de acuerdo a los criterios utilizados para su desarrollo. Utilizarán una lista de cotejo con los criterios a evaluar en esta instancia.

En esta actividad de co evaluación, utilizando herramientas de la plataforma de Moodle, cada equipo evaluara a otro, apoyado con la lista de criterios a evaluar. Dispondrán de un tiempo para hacerlo.

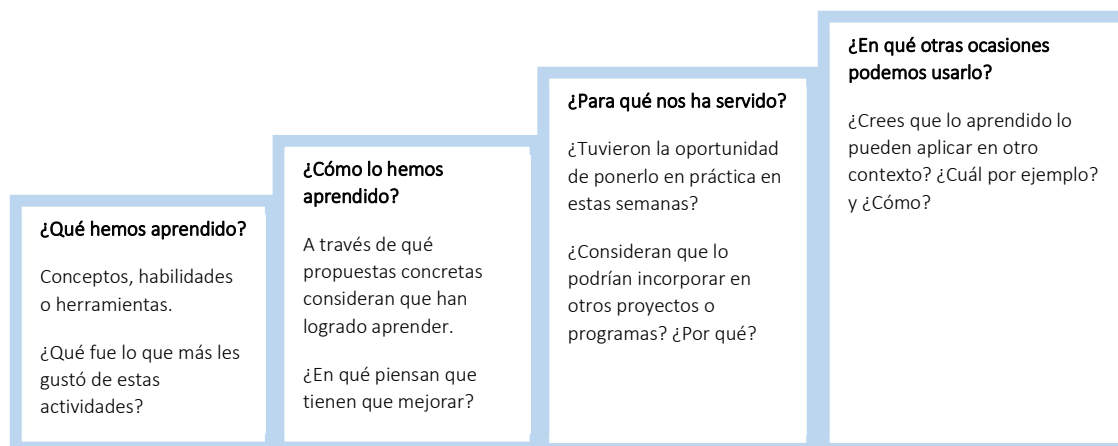
Este será un momento de discusión grupal (estudiantes y docentes). Presentación de dificultades y fortalezas detectadas en ambos grupos.

#### Momento de reflexión en equipo: Retrospectiva

Al finalizar cada etapa, cada equipo deberá realizar un momento de reflexión (retrospectiva) del equipo de trabajo. Preguntas disparadoras (ejemplo):

- ¿Qué les pareció Trello? ¿Qué ventajas y desventajas detectan en la utilización de la herramienta?
- ¿Les resultó fácil de usar? ¿Les parece que pueden aplicarla en otros proyectos?
- ¿Cómo anduvieron con Pair Programming? ¿Les resultó útil?
- ¿Ventajas y desventajas del entorno de programación?
- ¿Qué aprendieron de nuevo? ¿Qué fue lo que más les gustó de estas actividades?
- ¿En qué piensan que tienen que mejorar? ¿Qué pueden hacer para mejorar?

Se espera que incluyan un breve texto en el que describan los aprendizajes que construyeron a lo largo del proyecto, explicitando también qué actividades concretas los ayudaron a adquirirlos y cómo dan cuenta que lo aprendieron. Con esta finalidad se utilizará como herramienta la escalera de metacognición, donde encontrarán para cada “escalón” una guía de preguntas específicas.



#### Diario de reflexión

Se aplicará el diario de reflexión como herramienta de autoaprendizaje y de metacognición, que es un “diario personal y académico” que permite escribir para aprender y permite objetivizar la realidad en la que los alumnos reflexionen y escriban acerca de su proceso de aprendizaje, centrado en los siguientes aspectos: el desarrollo conceptual que ha alcanzado; los sentimientos y actitudes que experimentó durante su aprendizaje; y los procesos mentales y procedimentales que siguió para lograr el aprendizaje.

El diario permitirá que los alumnos respondan a preguntas como:

- ¿Qué he aprendido con .....?



- ¿Qué estrategia apliqué para aprender?
- ¿Qué sentimientos me despertó este proceso de aprendizaje?
- ¿Qué fue lo que más me gustó de estas actividades?
- ¿En que podría mejorar? ¿Qué puedo hacer para mejorar?

Compartirán sus reflexiones individuales con el grupo.

## Rúbrica para evaluar cada etapa

	Muy bueno	Bueno	Regular	Debe mejorar	Observaciones
Gestión del proyecto mediante el uso de Trello	El tablero refleja adecuadamente la gestión del proyecto: la división de tareas es adecuada, y se utilizaron estados apropiados para las tareas. Se visualiza interacciones de todos los miembros del equipo.	El tablero refleja medianamente bien la gestión del proyecto: la división de tareas es bastante adecuada. Se visualiza interacciones de todos los miembros del equipo.	El tablero no refleja la gestión de todo el proyecto: la división de tareas es demasiado general. Se visualiza interacciones de algunos miembros del equipo.	El tablero no refleja la gestión del proyecto, o no se utilizó Trello. No se visualiza interacciones entre los miembros del equipo.	
Descripción de los resultados de las reflexiones	El texto que describe las reflexiones refleja respuestas a todas las preguntas sugeridas, y se ven otras reflexiones adicionales.	El texto que describe las reflexiones refleja respuestas a muchas de las preguntas sugeridas.	El texto que describe las reflexiones refleja respuestas a pocas de las preguntas sugeridas.	El texto que describe las reflexiones no refleja adecuadamente respuestas a las preguntas sugeridas.	
Incluyen los conceptos dados durante esta etapa adecuadamente en el programa	Entienden los conceptos y los aplican adecuadamente al programa.	La aplicación de los conceptos no es del todo adecuada, pero obtienen los resultados.	Aplican medianamente bien los conceptos, pero los resultados no son adecuados.	No aplican los conceptos adecuadamente ni obtienen los resultados esperados.	
Estilo de programación	El código es comprensible, utiliza indentaciones adecuadamente, la denominación de las variables y funciones es descriptiva y apropiada, utiliza comentarios.	El código es comprensible, utiliza indentaciones en algunos casos, la denominación de las variables y funciones es descriptiva y apropiada.	El código es comprensible, utiliza indentaciones en pocos casos, la denominación de las variables y funciones no es descriptiva ó no es apropiada.	El código es desprolijo y poco claro.	
Funcionalidad de la solución propuesta (El programa funciona correctamente)	El código funciona perfectamente.	El código funciona para la mayoría de los casos (más del 70%).	El código brinda menos del 70% de las funcionalidades requeridas.	El código no compila o da error al ejecutarlo.	

## Ejemplo de Guía de desarrollo (para los estudiantes)

---

### A) Programando con personajes de cómics

#### *Descripción de la propuesta*

A lo largo de todo el cuatrimestre se irá desarrollando un programa de forma iterativa e incremental. La temática será orientada a los fans de cómics.

Iremos aplicando las nuevas estructuras de datos y técnicas de programación que los estudiantes conocerán en esta materia en un programa, para que experimenten las diferentes formas de utilizarlas y cómo pueden combinarse de diversas formas.

Será importante que comprendan acerca de personajes de cómics, porque cuando se programa es un plus conocer o aprender lo más que se pueda acerca del tema relacionado a la aplicación.

#### Metodología del trabajo

Las estrategias didácticas a implementar facilitarán el aprendizaje colaborativo y permitirán que los estudiantes aprendan a utilizar programas como [Trello](#) y metodologías ágiles como [Programación de pares](#) para su uso en situaciones propias del desempeño de un profesional de [Sistemas](#).

#### Actividades

##### **Etapas 1: Semana de repaso**

3. Crear un tablero en Trello. En el mismo deberán ir cargando las tareas por hacer, del modo que les parezca más adecuado para realizar un seguimiento del trabajo: las tareas por hacer o pendientes, las tareas en ejecución, las tareas completas y realizadas.
4. Crear un programa en lenguaje "C" que permita generar un archivo binario de datos de personajes de cómics. Para ello será necesario que se permita ingresar por teclado los siguientes datos de los personajes de cómics: universo al que pertenece (1- DC, 2- Marvel), código de personaje, nombre del personaje (Por ejemplo: Superman, Thor, Mujer Maravilla, Capitán América, Magneto, etc.), género (F-femenino, M-masculino), rol (H-héroe/heroína, V- villana/o), cantidad de películas en las que participó, superpoder principal. El programa debe contener las siguientes funciones:
  - a) `ingresarPersonaje(tPersonaje)` que se ocupe de ingresar los datos de un personaje
  - b) `grabarPersonajes ()` que se ocupe de grabar los datos en el archivo, hasta que el usuario decida que no desea ingresar más personajes
  - c) `mostrarPersonajesGrabados()` que se ocupe de mostrar los personajes grabados en el archivo

##### **Etapas 2: Estructuras compuestas: listas (implementación estática mediante arrays)**

4. Modificar el procedimiento de **ingresarPersonaje (tPersonaje)** para que a medida que se ingresen los datos de un personaje, se almacene la estructura que contiene los datos en una **lista estática** de personajes.
5. Escribir las funciones necesarias para que el usuario pueda verificar los personajes que posteriormente se grabaran en el archivo:
  - **mostrarListaPersonajes(tArrayPersonajes)**: Mostrar personajes (recorrer la lista)
  - **eliminarPersonaje(tArrayPersonajes, int)**: Eliminar un personaje (a partir de la selección del código de personaje o del nombre? )
6. Modificar la función que graba los personajes para que, en lugar de grabar de un personaje a la vez, los grabe a medida que recorra la lista de personajes.

### **Etapas 3: Estructuras compuestas enlazadas (implementación dinámica mediante listas enlazadas)**

4. Modificar la función **ingresarPersonaje (tPersonaje)** para que a medida que se ingresen los datos de un personaje, se almacene la estructura que contiene los datos en una **lista enlazada de personajes** en lugar de la lista estática programada en la etapa anterior.
5. Modificar las funciones necesarias para que el usuario pueda verificar los personajes que posteriormente se grabaran en el archivo (Mostrar lista de personajes, eliminar un personaje, etc.) para que apliquen a las funciones correspondientes a las listas.
6. Modificar la función que graba los personajes para que recorra la lista enlazada de personajes, en lugar de recorrer la lista estática y a medida que acceda a un nodo con contenido correspondiente a un personaje lo grabe en el archivo de datos.

### **Etapas 4: Corte de control**

2. Agregar una función **informePersonajes ()** que permita generar el siguiente reporte:

\*\*\* Personajes registrados \*\*\*

Universo DC		
	Personaje	género
	Mystique	Femenino
	Joker	Masculino
Cantidad de villanos: xx	Cantidad de héroes: xx	
Cantidad de películas DC: xx		
Universo Marvel		
	Personaje	género
	Hulk	Masculino
	Loki	Masculino
	Capitana Marvel	Femenino
	Thor	Masculino

	Doctor Strange	Masculino
Cantidad de villanos: xx	Cantidad de héroes: xx	
Cantidad de películas Marvel: xx		

### Etapa 5: Recursividad

1. Modificar las siguientes funciones para que apliquen la técnica de la recursión:
  - a) La función **ingresarPersonajes()**.
  - b) La función **mostrarPersonajes()** que recorre la lista enlazada para mostrar al usuario cuáles son los personajes que se van a grabar.

### Etapa 6: Programación de placas tipo Arduino

Se propone un juego de lucha por turnos, basado en combinaciones de ataques para dos jugadores en Arduino.

Respecto del hardware el juego debe comprender:

- 2 leds de diferente color, donde cada led indica el turno de un jugador.
- 4 leds de otro color, donde cada uno indica el tipo de ataque que debe realizar el jugador (Por ejemplo: patada, golpe de puño, saltar, superpoder).
- 4 botones, cada uno se corresponde con un led de tipo de ataque.
- 1 botón, que indica cuando comienza la batalla.

Las reglas del juego son las siguientes:

- El juego será por turnos, cada jugador debe elegir un color.
- Cuando el led de un color se encienda, el juego, por medio de los leds de ataque, indicará cual es la combinación de ataques que debe realizar el jugador de turno.
- El jugador de turno, ingresa la combinación de ataque mediante los botones, si es correcta, concluye su turno.
- En cada turno, se incorpora un nuevo elemento a la combinación de ataques.
- La batalla finaliza cuando uno de los jugadores, se equivoque en la combinación de ataques.

Para almacenar las combinaciones de ataques, el grupo deberá proponer la estructura de datos más adecuada y justificar su propuesta.

### Etapa 7: Selección de estrategia algorítmica para optimizar y hacer más eficiente al algoritmo

De las actividades realizadas en cualquiera de las etapas del 2 al 6, el equipo deberá seleccionar algún módulo a fin de utilizar alguna estrategia algorítmica para lograr mayor eficiencia u optimización del mismo. Luego deberá implementar la propuesta y comprobar la optimización y/o eficiencia lograda.