**✅ Módulo 1: Gestión de Usuarios (Inicio de sesión)**

🔖 **Título:** Implementar sistema de autenticación por consola

**Descripción:**  
Permite al usuario iniciar sesión utilizando un nombre de usuario y contraseña, validando su rol (administrador). Garantiza el acceso a funcionalidades correctas.

**Tareas específicas:**

* Crear clase Usuario con atributos: nombre, contraseña, rol.
* Crear clase Administrador heredando de Usuario.
* Crear lista predefinida de usuarios.
* Solicitar datos desde consola.
* Validar credenciales y rol.
* Redirigir al menú del sistema.

**✅ Módulo 2: Gestión de Aviones (Lista Enlazada Simple)**

🔖 **Título:** CRUD de aviones en lista enlazada simple

**Descripción:**  
Permite registrar, modificar, eliminar y consultar aviones mediante una lista enlazada simple. Cada nodo almacena un avión y referencia al siguiente.

**Tareas específicas:**

* Crear clase Avion con atributos: matrícula, tipo, estado, etc.
* Crear clase NodoLista y ListaAviones.
* Implementar método agregarAvion().
* Implementar buscarAvionPorMatricula().
* Implementar modificarAvion() (con búsqueda previa).
* Implementar eliminarAvion() en cualquier posición.
* Mostrar todos los aviones almacenados.

**✅ Módulo 3: Gestión de Tareas (Listas Enlazadas Circulares)**

🔖 **Título:** Gestión circular de tareas por avión

**Descripción:**  
Cada avión tiene asociada una lista circular de tareas de mantenimiento. Se pueden agregar, eliminar y recorrer en bucle infinito.

**Tareas específicas:**

* Crear clase Tarea con atributos: código, descripción, tipo.
* Crear clase NodoCircular y ListaTareasCirculares.
* Asociar la lista de tareas al objeto Avion.
* Implementar agregarTarea(Avion).
* Implementar eliminarTarea(Avion).
* Recorrer las tareas en forma circular mostrando las descripciones.

**✅ Módulo 4: Simulador de Tareas Urgentes (Pilas)**

🔖 **Título:** Procesar tareas urgentes en estructura tipo pila

**Descripción:**  
Administra tareas de emergencia con estructura LIFO. Las tareas más recientes se atienden primero.

**Tareas específicas:**

* Crear clase PilaTareasUrgentes.
* Crear método agregarTareaUrgente() (push).
* Crear método ejecutarTareaUrgente() (pop).
* Mostrar todas las tareas pendientes.
* Controlar que no se ejecute si la pila está vacía.

**✅ Módulo 5: Historial Recursivo de Tareas Finalizadas**

🔖 **Título:** Mostrar y buscar historial de tareas usando recursividad

**Descripción:**  
Permite recorrer e imprimir tareas finalizadas mediante métodos recursivos, y realizar búsquedas por tipo o descripción sin utilizar ciclos.

**Tareas específicas:**

* Crear clase HistorialRecursivo.
* Crear estructura para almacenar tareas finalizadas.
* Método recursivo para imprimir todas las tareas.
* Método recursivo para buscar tarea por tipo.
* Validar que la búsqueda sea exacta.

**✅ Módulo 6: Menú Principal del Sistema**

🔖 **Título:** Crear menú de navegación por consola

**Descripción:**  
Interfaz por consola que permite acceder a los módulos según el tipo de usuario. Controla la navegación y la entrada del usuario.

**Tareas específicas:**

* Crear clase MenuSistema.
* Mostrar opciones de menú al iniciar sesión.
* Validar opción ingresada.
* Redirigir a la lógica de cada módulo.
* Diferenciar opciones según rol (Administrador vs Técnico).