

Requerimientos Funcionales para el Programa de Gestión de Matrícula de Dev Senior

El sistema de **gestión de matrícula de Dev Senior** permite a los usuarios registrar estudiantes en cursos, asignar docentes a cada curso, y definir días y horarios de clase. Está diseñado para operar en un entorno de consola, donde se realizan todas las operaciones a través de un menú interactivo. La información se gestiona temporalmente en listas sin persistencia de datos entre sesiones.

Requerimientos Funcionales

1. Registro de Estudiantes en Cursos

- **Descripción:** El sistema permite registrar estudiantes en los cursos de Dev Senior, limitados a los cursos de **Java** y **Python**.
- **Requerimientos:**
 - El sistema debe solicitar el **nombre del estudiante** y el **curso** al que desea matricularse (Java o Python).
 - Una vez registrado el estudiante, el sistema debe confirmar la matrícula y almacenar el nombre y el curso en la lista de estudiantes matriculados.
 - La matrícula debe ser registrada solo si el curso seleccionado es válido (Java o Python).
 - El sistema debe permitir la matrícula de múltiples estudiantes en una misma sesión.

2. Asignación de Docentes a los Cursos

- **Descripción:** La aplicación permite asignar docentes a los cursos disponibles en Dev Senior.
- **Requerimientos:**
 - Al seleccionar la opción de asignar docente, el sistema debe permitir al usuario seleccionar un curso (Java o Python) y luego ingresar el **nombre del docente**.
 - El docente asignado debe ser registrado en la lista de docentes junto con el curso correspondiente.

- Debe confirmar la asignación del docente una vez registrada.
- La asignación permite asociar un único docente a cada curso, aunque puede asignarse un docente diferente en una nueva sesión de trabajo si se requiere.

3. Definición de Horarios de Clase para los Cursos

- **Descripción:** La aplicación debe permitir definir los horarios de clase para cada curso.
- **Requerimientos:**
 - El sistema debe mostrar los cursos disponibles y solicitar al usuario que seleccione uno para definir su horario.
 - El usuario debe ingresar los **días de clase** (por ejemplo, Lunes y Miércoles) y la **hora de inicio** de la clase (por ejemplo, 6 pm).
 - El horario debe ser registrado en la lista de horarios con los datos de días y hora para el curso seleccionado.
 - El sistema debe confirmar la asignación del horario una vez registrado.

4. Consulta de Estudiantes Matriculados

- **Descripción:** La aplicación permite al usuario consultar la lista de estudiantes matriculados en los cursos de Dev Senior.
- **Requerimientos:**
 - Al seleccionar esta opción, el sistema debe mostrar todos los estudiantes matriculados en cada curso, junto con el nombre y curso de cada uno.
 - Si no hay estudiantes matriculados, debe mostrar un mensaje indicando que la lista de estudiantes está vacía.

5. Consulta de Docentes Asignados

- **Descripción:** El sistema debe permitir consultar los docentes asignados a cada curso.
- **Requerimientos:**
 - Al seleccionar esta opción, el sistema debe mostrar el nombre de cada docente asignado y el curso correspondiente.
 - Si no hay docentes asignados, debe mostrar un mensaje indicando que no hay docentes registrados.

6. Consulta de Horarios de los Cursos

- **Descripción:** El sistema permite consultar los horarios de clase definidos para cada curso.
- **Requerimientos:**
 - Al seleccionar esta opción, el sistema debe mostrar el curso, los días de clase, y la hora de inicio.
 - Si no hay horarios definidos, debe mostrar un mensaje indicando que no hay horarios registrados.

7. Salida del Programa

- **Descripción:** La aplicación permite al usuario finalizar la sesión de forma controlada.
- **Requerimientos:**
 - Al seleccionar la opción de salir, el sistema debe terminar el ciclo principal y mostrar un **mensaje de despedida**.
 - Una vez seleccionada esta opción, el programa ya no debe permitir realizar operaciones adicionales.