

# Informe de Subproceso de matriculación

Grupo revisor: Henry Ludeña, Jonatan Claudio

Grupo revisado: Nick Valverde, Alejandro Minda

Caso de Uso: El usuario se encuentra en el sistema SAEW, dentro del módulo de matriculación y trata de agregar una materia.

## Objetivos:

- Analizar las irregularidades de los procesos diagrama de flujo del caso de uso.
- Garantizar un buen desarrollo de software enfocándose en una correcta implementación de la reingeniería de software.

## Introducción:

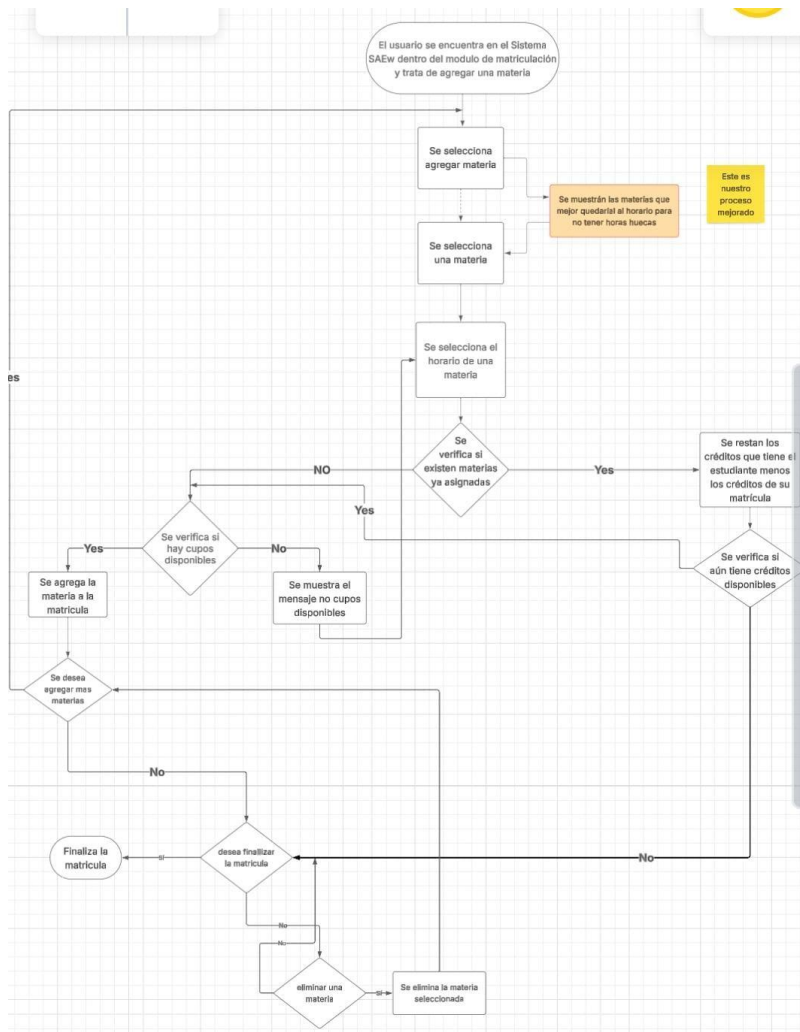
El presente análisis tiene como objetivo examinar el comportamiento del sistema SAEW durante la ejecución del caso de uso correspondiente a la acción del usuario al intentar **agregar una materia dentro del módulo de matriculación**.

Este escenario, aunque común en los procesos administrativos de gestión académica, puede presentar **irregularidades funcionales y de flujo** que comprometen la eficiencia del sistema y la experiencia del usuario.

A partir de la observación detallada del **diagrama de flujo** del caso de uso, se buscará identificar posibles fallos, redundancias o cuellos de botella que puedan afectar la lógica del proceso. Este diagnóstico servirá como base para **proponer mejoras bajo un enfoque de reingeniería de software**, con el fin de optimizar el desarrollo, asegurar la coherencia interna del sistema y fomentar una estructura sólida, escalable y mantenible.

En este contexto, el análisis no solo se centrará en la corrección de errores existentes, sino en la **mejora integral del diseño del proceso**, lo cual es fundamental para garantizar la calidad del producto final y una experiencia satisfactoria para el usuario.

Desarrollo:



## Resumen:

### 1. Inicio del proceso:

El usuario accede al sistema SAEW, dentro del módulo de matriculación, e inicia el proceso de agregar una materia.

### 2. Selección de materia:

- Se selecciona la opción "Agregar materia".
- Se elige una materia específica.
- Se selecciona el horario correspondiente.

### 3. Proceso optimizado (etapa mejorada):

- Se muestran los horarios que evitan tener horas huecas, facilitando una mejor organización para el estudiante.

### 4. Validación de materias ya asignadas:

- El sistema verifica si esa materia ya ha sido asignada.

- Si **ya está asignada**, se verifica si hay **cupos disponibles**.
  - Si hay cupos: se agrega la materia a la matrícula.
  - Si no hay cupos: se muestra un mensaje indicando que no hay disponibilidad.
- Si **no está asignada**, se verifica si el estudiante **aún tiene créditos disponibles**.
  - Si sí tiene créditos: se agregan y se restan del total disponible.
  - Si no tiene créditos suficientes: el sistema impide la adición.

#### 5. **Ciclo de decisión:**

- Luego de cada adición, el sistema pregunta si se desea agregar otra materia.
  - Si **sí**, se repite el proceso.
  - Si **no**, se pregunta si desea **finalizar la matrícula**.

#### 6. **Opción de eliminación** (antes de finalizar):

- Si el usuario no finaliza la matrícula directamente, puede optar por **eliminar una materia** previamente agregada.

#### 7. **Finalización del proceso:**

El usuario confirma la finalización del proceso de matrícula.

## **Correcciones:**

**Se muestran las materias que mejor quedarían al horario para no tener horas huecas:**

- Este subproceso no debería encontrarse en esta zona, sino que debería de estar en la zona posterior a la elección las materias verificando los créditos y los cupos, esto debido a que depende de los cupos y los horarios para la selección de una materia y de esa manera ya ordenando las horas huecas de los horarios.
- Se debería implementar un flujo mas específico de tal manera que se explique como va a ser el desarrollo de dicho proceso, debido a que hasta ahora queda muy ambiguo.

**Se verifica si existen materias ya asignadas:**

- No tiene sentido, debido a la ambigüedad, es decir, no debería haber una relación entre una materia previamente ya escogió y otra que se va a escoger.

Se agrega la materia a la matrícula:

- Después de este proceso debería haber una flecha (dirección) que continúe el proceso con “Se restan los créditos que tiene el estudiante menos los créditos de su matrícula”, de manera que no solo se agregue la materia sino que también deben restarse los créditos.

## **Recomendaciones:**

- Se debería añadir un flujo donde se filtre la aparición de materias recomendadas según el número de créditos disponibles. Por ejemplo: si el estudiante solo tiene 2 créditos deberían solo aparecer las materias que necesiten 2 créditos o menos.
- Después de la elección del horario se debería hacer dos condicionales, le primero es verificar si hay cupos disponibles y el segundo verificar si tiene créditos disponibles, es decir, el condicional “Se verifica si existen materias ya asignadas” es innecesario.

## **Conclusión:**

En conclusión, al analizar el diagrama de flujo nos podemos dar cuenta de que varios procesos necesitan ser reestructurados y modificados también en su orden, esto es importante tener presente debido a la importancia de cada uno de los pasos a tener en cuenta al implementar estos procesos en un software funcional. Por tal motivo siempre es necesario hacer un análisis tanto de la arquitectura como del diseño previos a la entrega del producto de la fase del ciclo de vida del desarrollo del software el cual es el diseño de software para que de esa manera en la etapa de implementación se tenga un esquema completo de cada uno de los procesos que se deben implementar a fin de que no haya errores ni defectos en el sistema y garantizar la calidad al cliente.