

The logo of the Instituto Tecnológico de Costa Rica (ITCR) is a dark blue square with the letters "ITCR" in white, centered within it.

ITCR

## Sistema PAM-TEC

Programa de adultos mayores del  
Instituto Tecnológico de Costa  
Rica.

Diseño de software

Adrián Díaz Meza  
María José Salazar Murillo

---

## Contenido

Resumen ejecutivo.....	2
Objetivos del sistema.....	3
Objetivo general.....	3
Objetivos específicos.....	3
Especificación de requerimientos .....	4
Requerimientos Funcionales.....	4
Requerimientos No funcionales .....	6
Requerimientos de usabilidad: .....	6
Requerimientos de confiabilidad: .....	6
Requerimientos de funcionalidad:.....	6
Requerimientos de soporte y configuración: .....	6
Prioridades.....	7
Funcionalidades.....	7
Factores de Calidad .....	8
Descripción de diseño de alto nivel. ....	9
Diagrama de arquitectura conceptual: .....	9
Diagrama de paquetes: .....	10
Diagrama de componentes: .....	11
Diagrama de despliegue:.....	12
Descripción detallada del sistema. ....	13
Diagrama de clases.....	13
Justificación de patrones usados.....	14
1.  Jugador-Rol.....	14
2.  Singleton.....	15
3.  Patrón Broker.....	16
4.  Patrón Jerarquía General.....	17
5.  Patrón Abstracción-Ocurrencia. ....	18
Problemas de diseño .....	19
Interacción con sistemas externos. ....	20

## Resumen ejecutivo.

PAM-TEC Programa Educativo para la Persona Adulta Mayor del Tecnológico de Costa Rica, es un programa donde se capacitan los adultos mayores en diferentes áreas, como computación, inglés, fotografía entre otras, a pesar de que es un programa ha venido teniendo bastante éxito durante los últimos trimestres, aún no cuenta con un sistema que automatice algunos de los procedimientos necesarios para poder gestionarlo correctamente.

Inicialmente se pretendía que sería una pequeña población la que se acercaría a este programa por lo cual no se veía la necesidad de un sistema para la gestión de dicho programa, sin embargo la población estudiantil que participa activamente ha venido creciendo, por lo cual se ha complicado la administración de este, por ejemplo y quizá el proceso más importante es el de matrícula, que es llevada a cabo por medio de los colaboradores del programa de forma manual, mediante el uso de hojas impresas o con la ayuda de editores de texto, lo cual conforme aumenta la población dicho proceso se vuelve más tedioso.

Es por estas razones que surge la necesidad de implementar un sistema computarizado que permita, tanto la matrícula de cada uno de los estudiantes, así como un sistema adecuado para manejo de toda la información que rodea este proceso. Información meramente administrativa, como información de tutores, horarios, aulas, materiales, entre otros.

El proyecto es totalmente sin fines lucrativos, lo que se busca es ayudar a la administración de este programa.

## Objetivos del sistema.

### Objetivo general.

Brindar a los integrantes del Programa Educativo para la persona adulta mayor del Instituto Tecnológico de Costa Rica (PAMTEC) la capacidad de procesar de forma rápida y eficiente el proceso de matrícula mediante un sistema de gestión de matrícula.

### Objetivos específicos.

1. Ofrecer mediante el sistema de gestión de matrícula: rapidez, seguridad de los datos, e información a tiempo real al Programa Educativo para la persona adulta mayor del Instituto Tecnológico de Costa Rica.
2. Mejorar la atención al cliente mediante el sistema de gestión de matrícula.
3. Proporcionar información oportuna y confiable, en el momento que se precise.
4. Realizar búsquedas para localizar un determinado curso, estudiante o profesor.
5. Generar reportes como: listados de alumnos, cursos y horarios entre otros.

## Especificación de requerimientos

En esta sección se detallarán cada uno de los requerimientos tanto funcionales como no funcionales que se requieren para que el sistema funcione correctamente.

Acrónimos utilizados.

Acrónimo	Significado
RQF	Requerimiento funcional
RQNF	Requerimiento <b>no</b> funcional

## Requerimientos Funcionales.

**RQF-001.** El sistema debe de permitir agregar, editar, eliminar y consultar información (nombre completo, cedula, fecha nacimiento, número de teléfono, dirección exacta) de los estudiantes de PAM-TEC.

**RQF-002.** El sistema debe permitir generar un historial respecto a la información de los estudiantes con los cursos aprobados, reprobados y abandonados.

**RQF-003.** El sistema deberá proveer las acciones básicas para agregar, editar, eliminar y consultar información (nombre, código, horario, tutor, aula, fecha inicio, fecha finalización, requisito, descripción) de los cursos.

**RQF-004.** El sistema debe de permitir agregar, editar, eliminar y consultar información (nombre completo, carné, fecha nacimiento, número de teléfono, dirección exacta, categoría de tutor (horas beca, voluntario, asistencia especial, administrador)) de los tutores de PAM-TEC.

**RQF-005.** El sistema debe de permitir la asignación de roles a cada uno de los tutores registrados.

**RQF-006.** El sistema debe permitir la relación de tutores a cursos.

**RQF-007.** El sistema debe permitir la matrícula de un estudiante, asociando estudiante a cursos.

**RQF-008.** El sistema debe de permitir a los administradores agregar o modificar estado del curso, por estudiante.

**RQF-009.** El sistema debe de permitir retirar a un estudiante de un curso ya antes matriculados.

**RQF-010.** El sistema debe de mostrar solo los cursos que un estudiante dado puede matricular.

**RQF-011.** El sistema permite generar reportes de matrícula.

**RQF-012.** El sistema genera diferentes tipos de consultas y/o reportes que podrán ser procesadas y visualizadas por los diferentes usuarios dependiendo de su rol.

**RQF-013.** El proyecto debe de ejecutarse desde el browser.

**RQF-014.** El sistema debe de poder visualizarse desde desktop, tabletas y Smartphone.

**RQF-015.** El sistema deberá de estar conectado al API de Google Drive.

**RQF-016.** El sistema debe de hacer uso al menos de 5 patrones de diseño.

**RQF-017.** El sistema debe permitir el acceso a través de un dominio público.

## Requerimientos No funcionales

### Requerimientos de usabilidad:

**RQNF-001.** El sistema deberá contar con un manual de usuario para cada una de las herramientas.

### Requerimientos de confiabilidad:

**RQNF-002.** El sistema deberá terminar las sesiones abiertas en caso de un cierre inesperado.

**RQNF-003.** El control de acceso a la información del sistema dependerá al rol de cada persona.

### Requerimientos de funcionalidad:

**RQNF-004.** El sistema debe responder a las consultas en un tiempo menor a 10 segundos.

**RQNF-005.** El proceso de autenticación para cualquier usuario deberá ser igual o menor a 5 segundos.

**RQNF-006.** El sistema con respecto a la parte de información general deberá estar disponible 24/7, todos los días del año.

### Requerimientos de soporte y configuración:

**RQNF-007.** El sistema debe estar en capacidad de permitir en el futuro el desarrollo de nuevas funcionalidades, modificar o eliminar funcionalidades después de su construcción y puesta en marcha inicial.

**RQNF-008.** El sistema debe operar de manera independiente del navegador que se utilice.

**RQNF-009.** Los encargados de la administración del sistema serán capaces de usar todas las funciones del sistema después de un total de 6 horas de entrenamiento.

## Prioridades

### Funcionalidades

El sistema está comprendido de ciertas funcionalidades antes mencionadas en los requerimientos. Aunque todas son importantes para conformar el sistema las desglosaremos por orden de prioridad:

1. **Gestión de la matrícula:** El sistema deberá gestionar la matrícula de un estudiante, asociando cursos para un periodo específico. También debe ser capaz de retirar un curso matriculado, si así el estudiante lo desea o de cambiar un curso por otro que más le convenga. Se deben mostrar sólo los cursos que puede matricular el estudiante. También el sistema será capaz de generar un reporte para la matrícula.
2. **Gestión de estudiantes:** El sistema deberá proveer las acciones básicas para agregar, editar y eliminar estudiantes. Así como generar un historial del estudiante con los cursos aprobados, reprobados.
3. **Gestión de cursos:** El sistema deberá proveer las acciones básicas para agregar, editar y eliminar cursos.
4. **Gestión de tutores:** El sistema deberá proveer las acciones básicas para agregar, editar y eliminar tutores. Así como asociar un tutor a los diferentes cursos que este puede impartir.



## Factores de Calidad

Para todo sistema se deben cumplir ciertos factores como lo son: confiabilidad, Usabilidad, eficiencia y mantenibilidad. Cada factor tiene una importancia diferente de acuerdo al sistema. A continuación desglosaremos cada una de los factores de acuerdo a la importancia que posee en el sistema de matrícula de PAMTEC.

Factor de Calidad	Justificación
1. Confiabilidad	Es el de mayor importancia para este sistema ya que lo más importante es que haga lo que tiene que hacer y que se pueda matricular con éxito ya que por un fallo en este factor se podría perder la oportunidad de llevar algún curso o problemas durante el curso lectivo.
2. Usabilidad	Debe ser un sistema sencillo de usar para todos los tutores. No se debe de tener complicaciones al momento de utilizarlo ni dificultades para entender como este funciona ya que el sistema puede ser manejado por los diferentes tutores y debería ser comprensible para todos, tanto para los tutores nuevos como para los existentes.
3. Eficiencia	Necesitamos que el sistema pueda funcionar con pocos recursos y con poca memoria ya que se desea acceder a él por cualquier medio que posea internet ya sea celular, Tablet, entre otros. Además de que el tiempo de respuesta y la persistencia es muy importante en este sistema se espera que sea rápido y que el sistema no se caiga.
4. Mantenibilidad	Consideramos este factor como el menos importante en comparación con los anteriores este factor es la capacidad que tiene un producto para ser modificado y aunque es un factor de suma importancia en este caso no se cree que sea tan importante para lograr cumplir la funcionalidad del sistema y poder matricular.

## Descripción de diseño de alto nivel.

En esta sección se describen los componentes principales del sistema y el modo en que interactúan entre sí para lograr los objetivos del diseño. Por lo cual se describen los siguientes diagramas:

1. Diagrama de arquitectura conceptual.
2. Diagrama de paquetes.
3. Diagrama de componentes.
4. Diagrama de despliegue.

Diagrama de arquitectura conceptual:

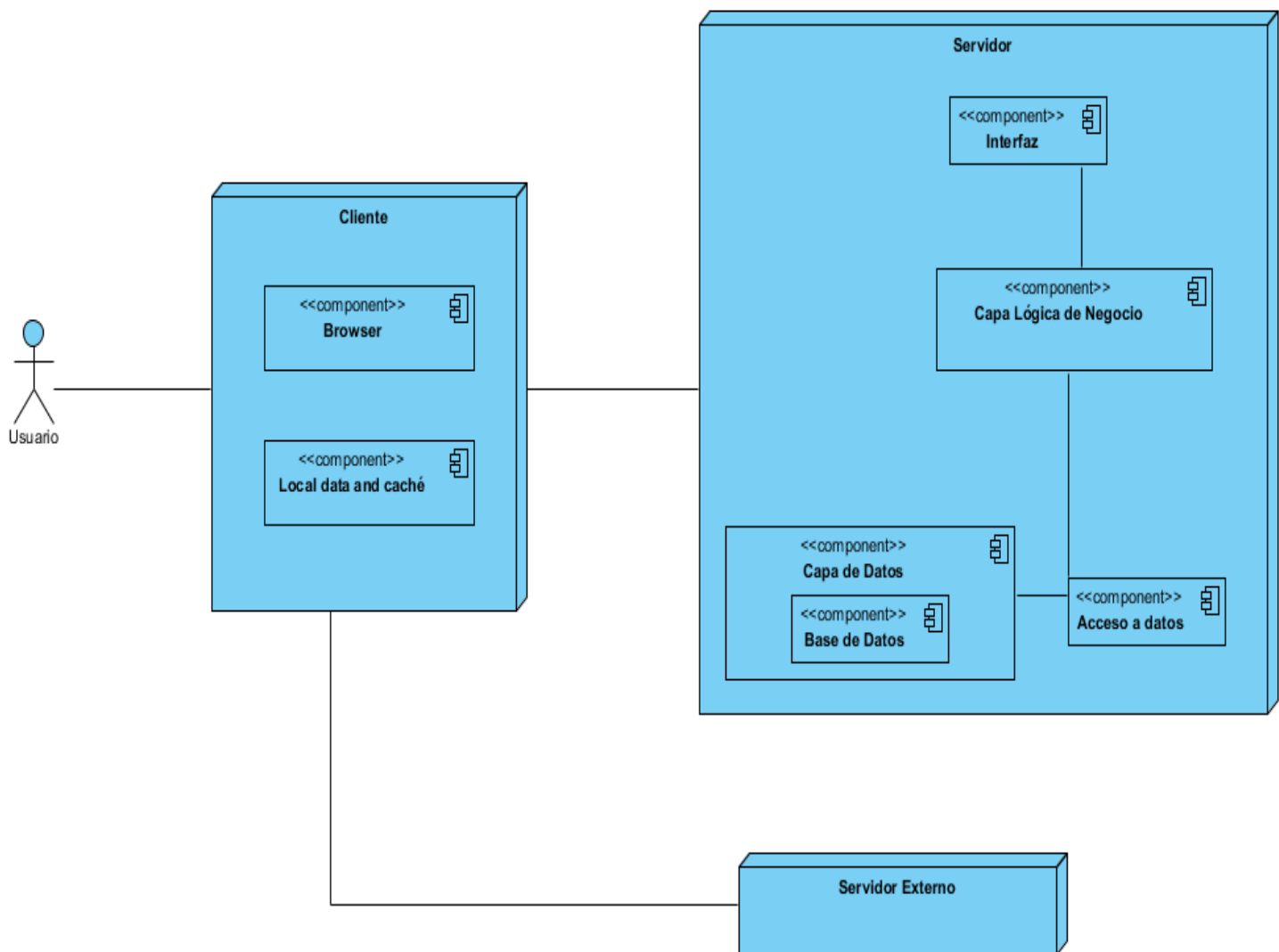


Diagrama de paquetes:

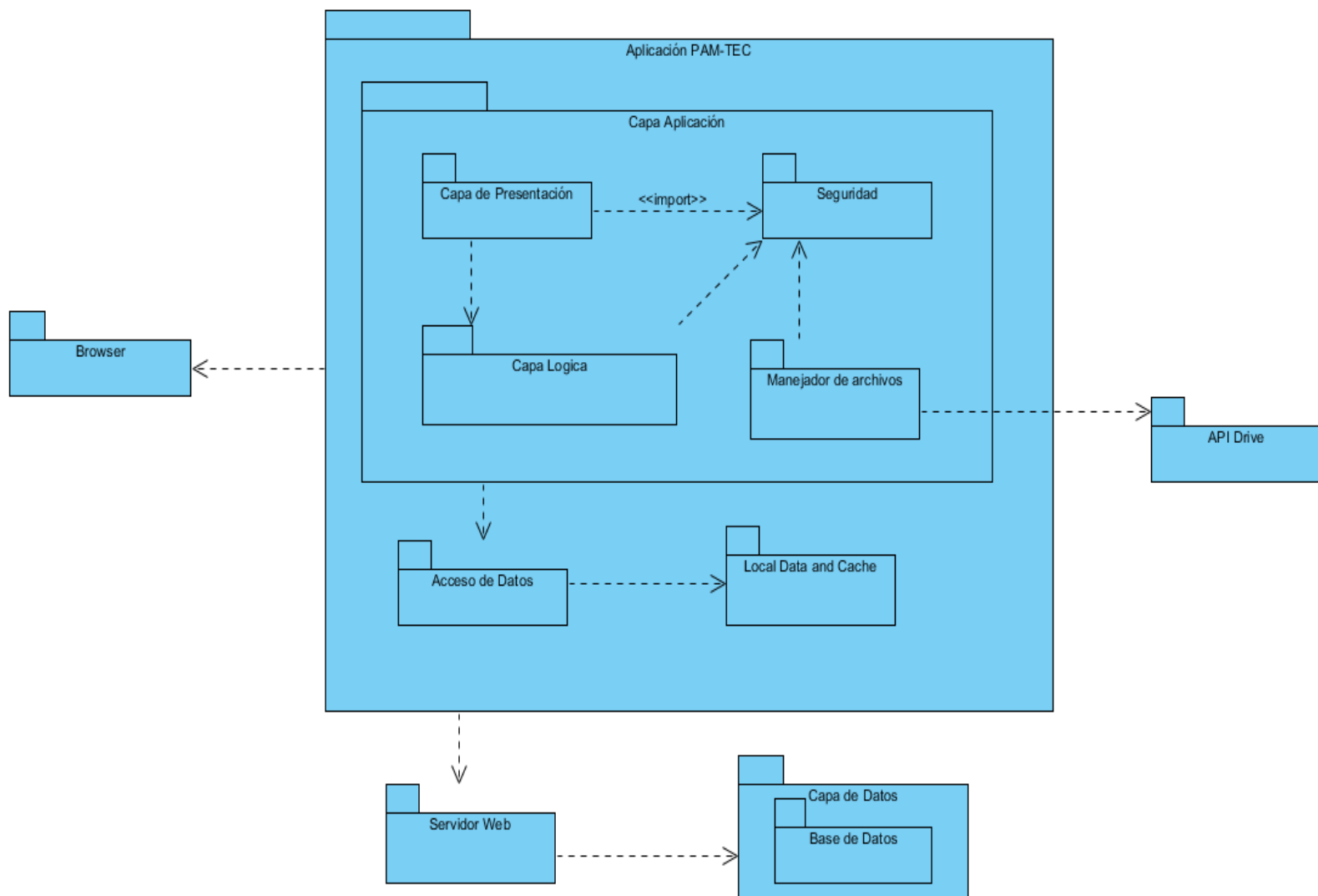


Diagrama de componentes:

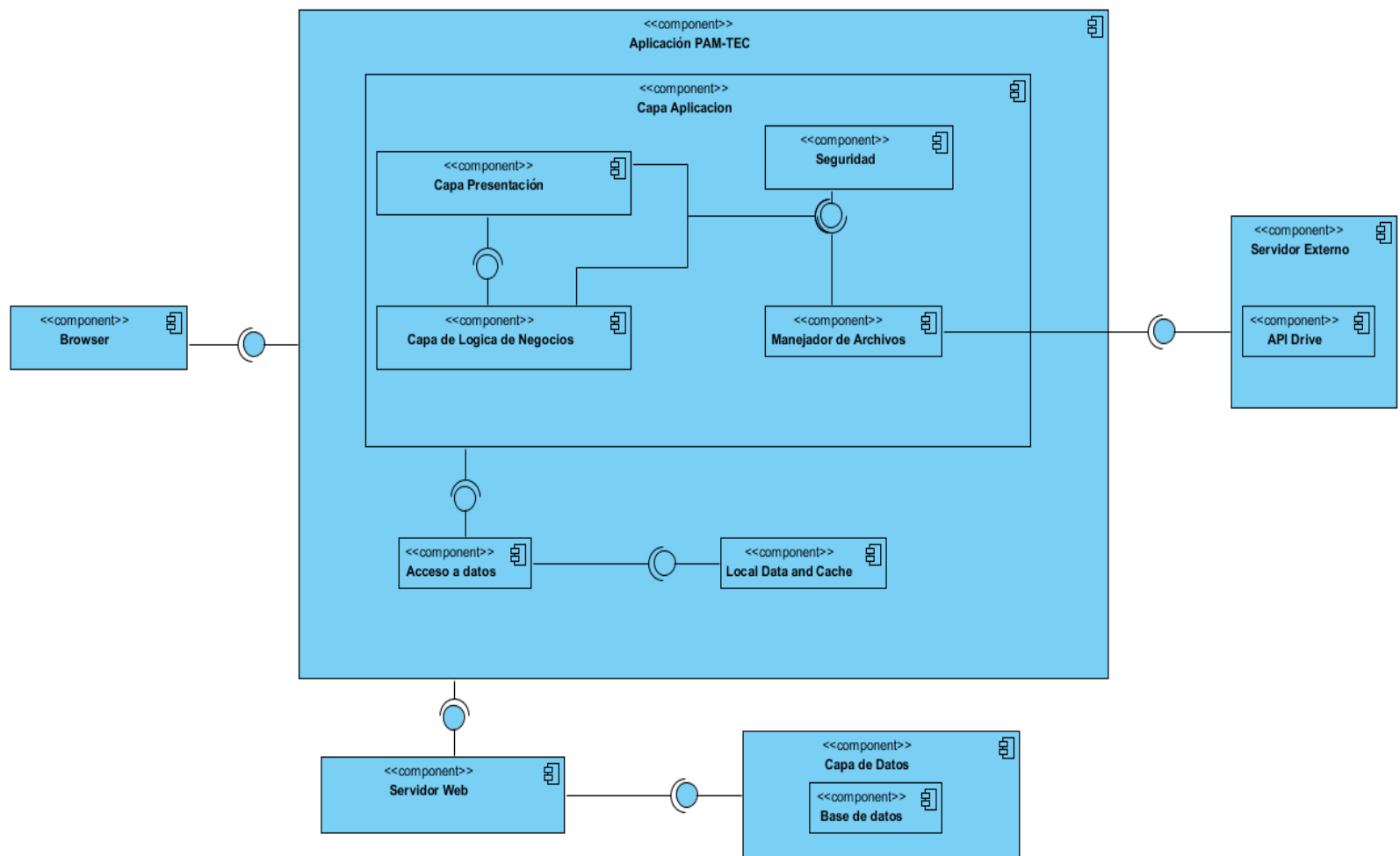
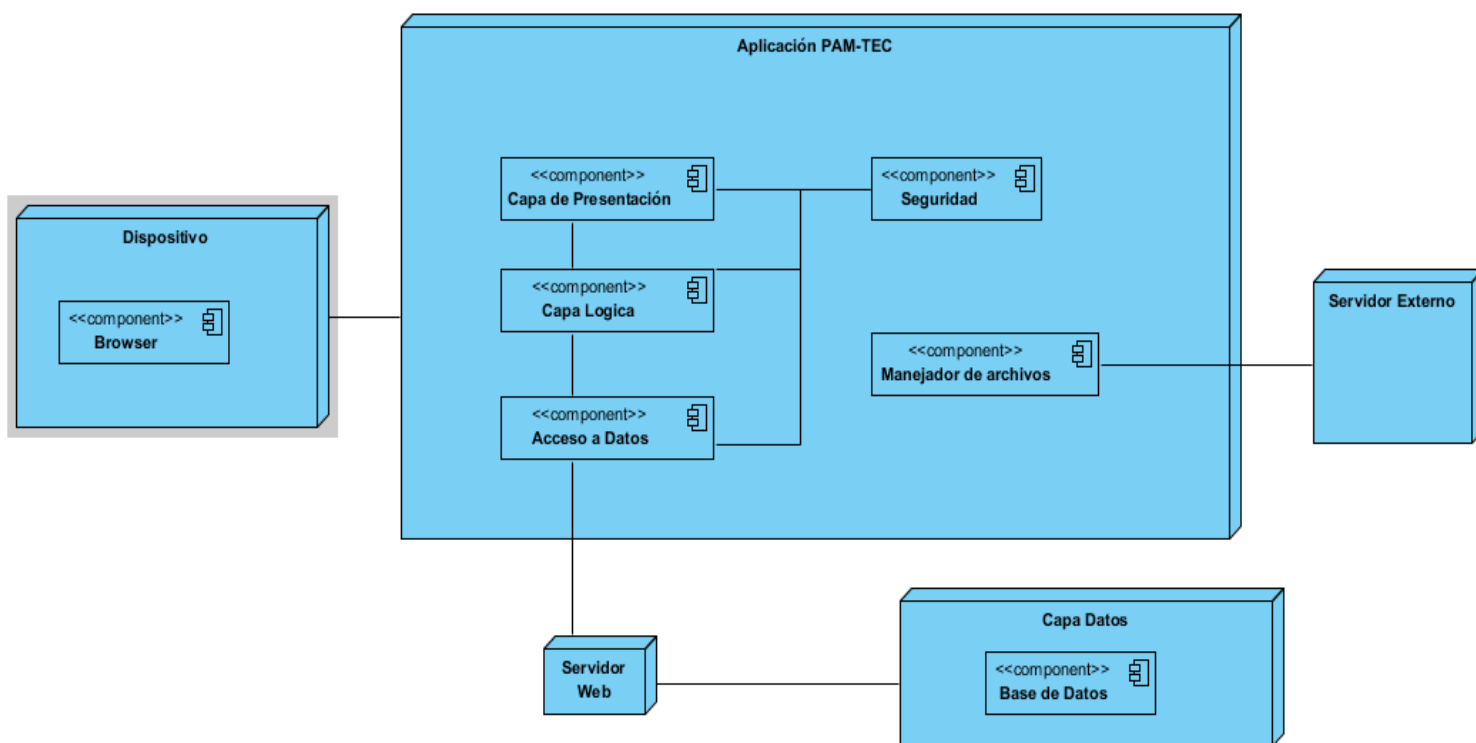


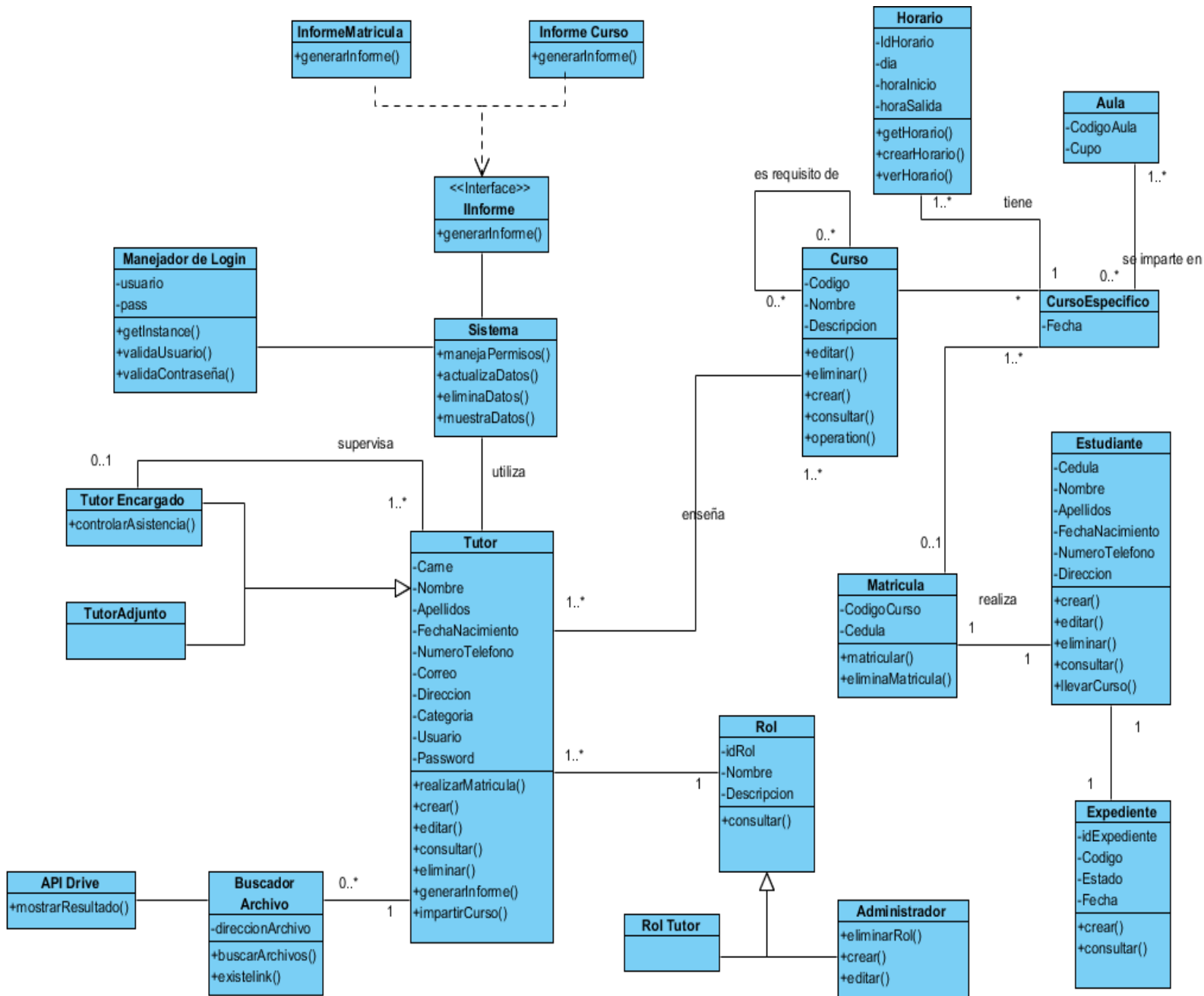
Diagrama de despliegue:



## Descripción detallada del sistema.

En esta sección se presenta una descripción detallada del sistema con el propósito de representar los objetos fundamentales del sistema, es decir los que percibe el usuario y con los que espera tratar para completar su tarea en vez de objetos del sistema o de un modelo de programación.

## Diagrama de clases.



## Justificación de patrones usados.

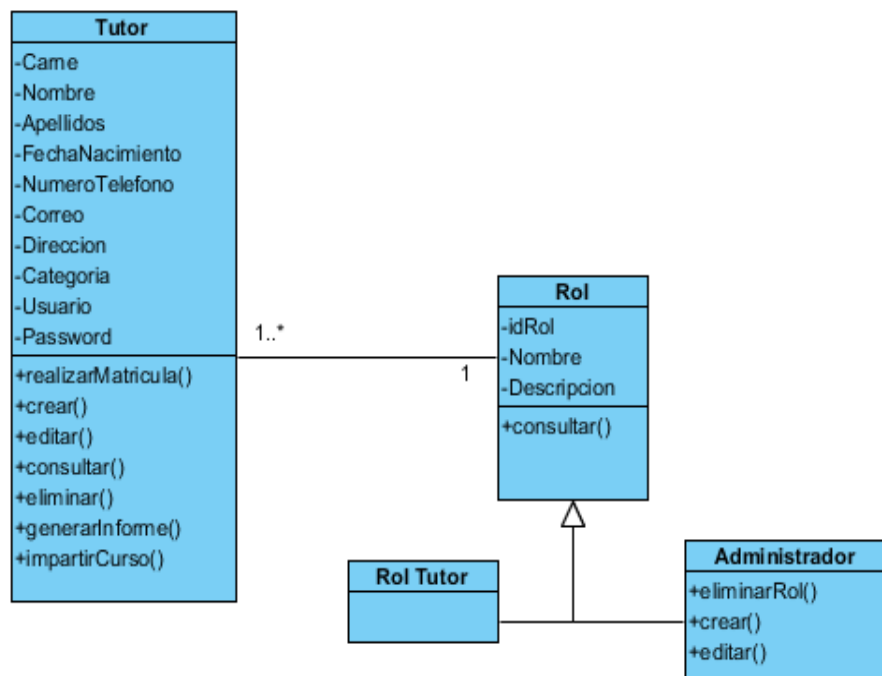
En esta sección se describen la situación donde se utilizaron los patrones de diseño para resolver algún problema en específico, así como sus respectivos diagramas.

### 1. Jugador-Rol.

En este caso se seleccionó este patrón debido a que un tutor tiene dos roles, un rol que es el de tutor y otro rol que es el de administrador sin embargo un tutor puede tener ambos roles, debido a que una persona que imparte clases puede ser tutor y también a su vez puede ser el administrador del programa.

Ejemplo: En este caso podemos mencionar a la encargada del programa de PAM-TEC que es la administradora, sin embargo ella también es tutora en algunos cursos.

A continuación se muestra en diagrama de este caso:



## 2. Singleton.

La idea de este patrón es proveer un mecanismo para limitar el número de instancias de una clase, en este caso la idea es dentro del login. Es decir este patrón busca garantizar que una clase sólo tenga una instancia y proporcionar un punto de acceso global a ella.

En este caso el login lo que hace es recibir peticiones de usuarios y verifica si el usuario es válido o no. En nuestro caso el encargado de realizar este proceso es una clase llamada manejador de login.

A continuación se muestra la imagen del diagrama correspondiente a este patrón:

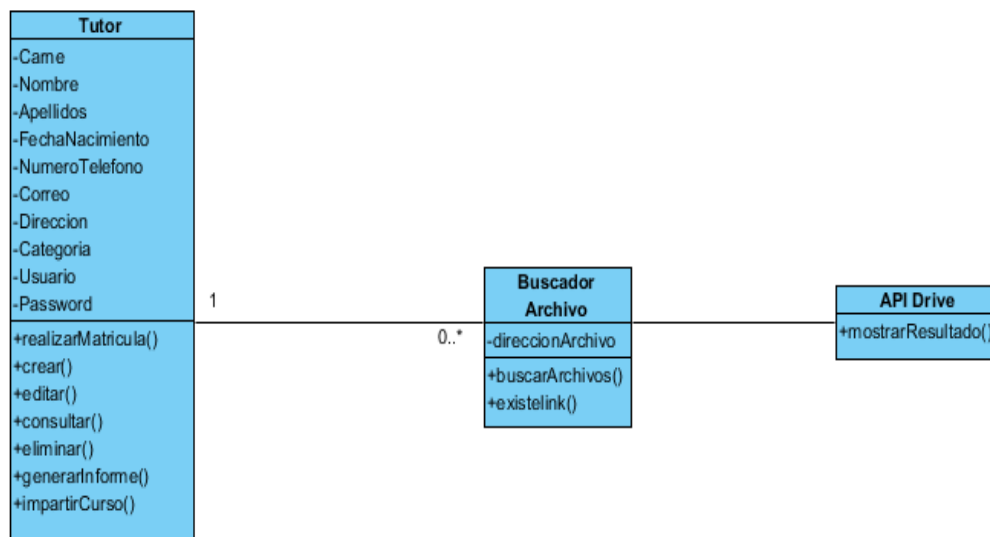




### 3. Patrón Broker.

Este es un patrón de arquitectura que se utiliza para estructurar sistemas de software distribuidos con componentes desacoplados que interactúan por invocaciones de servicios remotos, donde los componentes invocadores no saben que componentes invocados están en otras capas físicas. En este caso lo utilizamos para conectarnos con la API de Google Drive, para el manejo de archivos como por ejemplo los programas de cursos, entonces cuando un tutor quiere ver información de un determinado programa de algún curso, se direccionará a un documento de google drive y ahí podrá visualizar la información sin importar donde este almacenada.

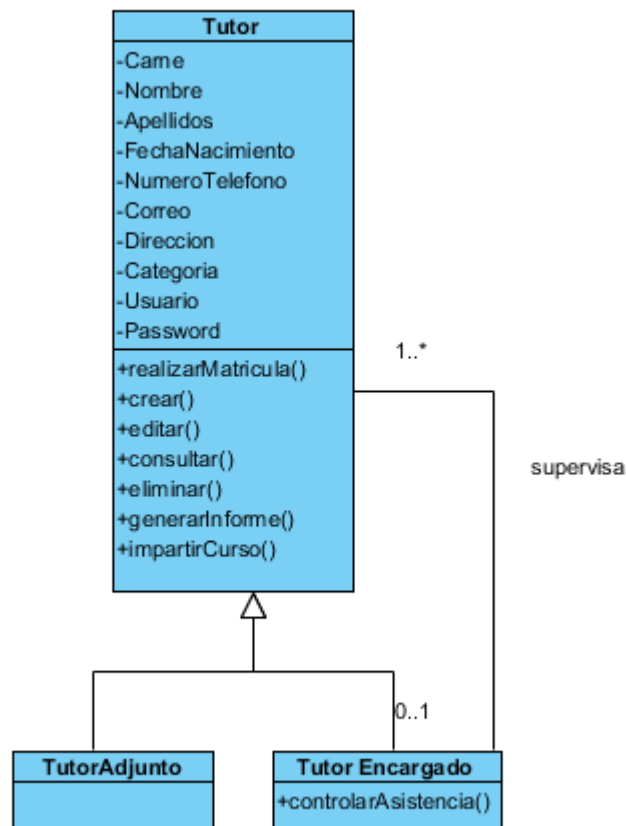
A continuación se muestra en diagrama de este caso:



#### 4. Patrón Jerarquía General.

La necesidad de utilización de este patrón se da debido a que un tutor puede ser un tutor adjunto o un tutor encargado, la gran diferencia de estos dos es que el tutor encargado es el responsable del tutor adjunto, velar por los materiales necesarios para impartir las clases, entre otros, en cambio el tutor adjunto solo se encarga de ayudarle al tutor encargado a dar la clase. Es importante tener en cuenta que una clase la pueden impartir varios tutores, sin embargo siempre hay un único tutor encargado.

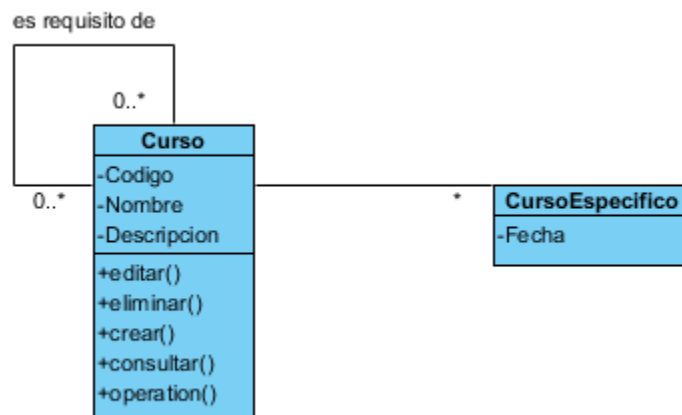
A continuación se muestra en diagrama de este caso, donde vemos que un tutor encargado puede supervisar a otros tutores siendo este un tutor también:



## 5. Patrón Abstracción-Ocurrencia.

Para este caso como puede haber varios cursos de un mismo curso, por ejemplo el curso compu1 se imparte en 3 horarios diferentes en 3 aulas diferentes, con tutores diferentes, por lo cual se utiliza este patrón para solucionar este problema de que había varios cursos con el mismo identificador, además comparten información entre ambas clases.

A continuación se muestra una imagen del diagrama con la implementación de este patrón:



## Problemas de diseño

Actualmente, la matrícula es realizada por los colaboradores del programa de forma manual. Al no utilizar una herramienta especializada para este fin, se presentan algunos problemas que se desean resolver con este sistema como lo son:

- Dificultad del manejo de la información de un estudiante: actualmente no existe una manera de ver el historial de un estudiante
- Pérdida de información: debido a que no existe un sistema especializado para esto, se ha sufrido una gran pérdida de información ya que muchas cosas se manejan en hojas que se pierden o archivos que se borraron y no se tiene ningún sistema que posea la información.
- Mal manejo de cupos por curso: Al no existir un sistema que maneje el control de cupos por curso se han dado problemas de sobrepasos de cupos en los cursos
- Mala gestión de cursos: Han existido muchos problemas de cursos sin aulas asignadas o sin tutores asignados.
- Desorden de información: al tener información en papeles o no tener un sistema que gestiona la información se tienen desordenes de información que conlleva a pérdida de la misma.
- Mala gestión de tutores: Problemas referentes a la gestión de tutores a cursos asignado información personal de estos.

## Interacción con sistemas externos.

En este caso se requiere la conexión con al menos un API externo a la aplicación. Pensando en que actualmente PAM-TEC hace uso de google drive para el manejo de alguna de su información más importante, que a su vez debe de ser compartida con todos los miembros del grupo de tutores o encargados de llevar a cabo el programa. Además de que en programa de adultos mayores se maneja información en la nube, como programas de cursos, lista de estudiantes, entre otros documentos importantes del programa. Se optó por la utilización del API que nos brinda google drive para acceder a este y poder subir archivos o visualizar documentos desde la aplicación de PAM-TEC.

Un poco sobre Google Drive, es que es una aplicación que nos brinda servicio de almacenamiento online, fue creado por Google, dispone de una API que nos permite trabajar con diferentes lenguajes de programación para poder gestionar todos los ficheros almacenados desde nuestra propia aplicación, es importante destacar que en este caso se manejará en .Net, para conectar a google drive.