



# Tecnológico de Monterrey

Análisis de Requerimientos de Software (Gpo 500)

Evidencia de Análisis de Requerimientos de Software. Entrega Medio Término.

Yaguit Joyce Montes López - A01114927

José Angel García Gómez - A01745865

Esteban Russek - A00827996

Prof. José María Castellanos Ortuño

31 de agosto de 2021

## **Situación Problema:**

¡Bienvenido! Ahora formas parte de la Fábrica digital “Tec21”. Lo que empezó como un pequeño departamento ha crecido increíblemente gracias al esfuerzo de todos sus colaboradores. Tanto, que recientemente se han mudado a un nuevo edificio con el fin de mejorar su infraestructura y tener mayor capacidad para el personal, mejorar los espacios de trabajo y contar con oficinas adecuadas para atender a los clientes.

La Directora de la fábrica digital, la Ing. Stepby Jobs, ha estado analizando la manera de delegar funciones y ahora que tiene una gran área bajo su dirección, requerirá más puestos y colaboradores de confianza. Actualmente, la Directora se encuentra en la selección de talento para una nueva coordinación que abrirá dentro de la fábrica, la cual se encargará de atender y supervisar los proyectos más grandes e importantes ¡ese es tu propósito! La Ingeniera Jobs pasa mucho tiempo atendiendo tantos asuntos de la agencia que no le permiten dedicar y concentrar el tiempo que le gustaría a cada proyecto, por ello, ha decidido que se apoyará en su personal de mayor confianza y se encuentra buscando los mejores candidatos para atender algunos proyectos, así como a un cliente muy especial, con el fin de no desatender la calidad de sus servicios. Como ves, tendrás una gran responsabilidad.

Uno de los primeros retos que te asignaron es desarrollar una aplicación móvil que ayude a contactar estudiantes que próximamente irán a su intercambio dentro del Tec21, y que desean interactuar entre ellos, para resolver inquietudes y compartir experiencias, a la vez que interactúan socialmente.

¿Qué actividades, procesos, herramientas y/o metodologías necesitarías conocer para resolver el problema?

## **Introducción**

Si bien, los modelos de ciclo de vida existentes para el desarrollo de software pueden ser útiles en cualquier caso, siempre va a haber una opción que mejor se adapte a los requerimientos o necesidades del proyecto. Por lo que es importante el considerar todos los modelos tomando en cuenta sus características principales para que de esta manera se pueda realizar la elección adecuada la cual nos ayude a realizar el reto con éxito y de la forma más eficiente posible.

### **Modelo de Ciclo de Vida:**

Consideramos que el modelo de desarrollo que más se adapta a la situación problema sería el **Incremental**, ya que este ciclo de vida se va adecuando a las necesidades del cliente, porque es menos complicado agregar las funcionalidades basadas en ellos y se adapta rápido a los cambios. Además, como se menciona en la descripción de la situación problema, esta fábrica digital ha ido creciendo conforme se ha desarrollado.

Por otra parte, las retroalimentaciones que se realizan por los usuarios en cada fase, nos ayudan a obtener mejores versiones hasta llegar a la final, por lo que el resultado debe de ser satisfactorio tanto para los usuarios como para el cliente.

### **Requerimientos Funcionales:**

- Conectar a los estudiantes que se encuentren de intercambio, hayan ido anteriormente o estén próximos a ir para que puedan interactuar entre ellos.
- Restricción de los usuarios que ingresan a la aplicación. Deberán ingresar con matrícula y contraseña.
- Hacer una sección en la aplicación para mostrar las preguntas más frecuentes de las personas que están próximas a irse de intercambio.
- Presentar un registro al usuario, para así que el sistema le seleccione personas que han ido o se van a ir al mismo lugar de intercambio.

### **Requerimientos no Funcionales:**

- Que la aplicación funcione en sistema IOS y Android sin restricciones.
- Las funcionalidades del sistema deben de responder al usuario en menos de 5 segundos.
- El sistema debe de operar correctamente con hasta 1,000 usuarios con sesiones iniciadas.
- Todo el sistema se debe de encontrar dentro de MiTEC.
- Solo gente que se va, está o se fue de intercambio puede acceder al sistema con su matrícula institucional.
- Debe contar con asistencia en línea.

- Proporcionar mensajes de error cuando se requiera.

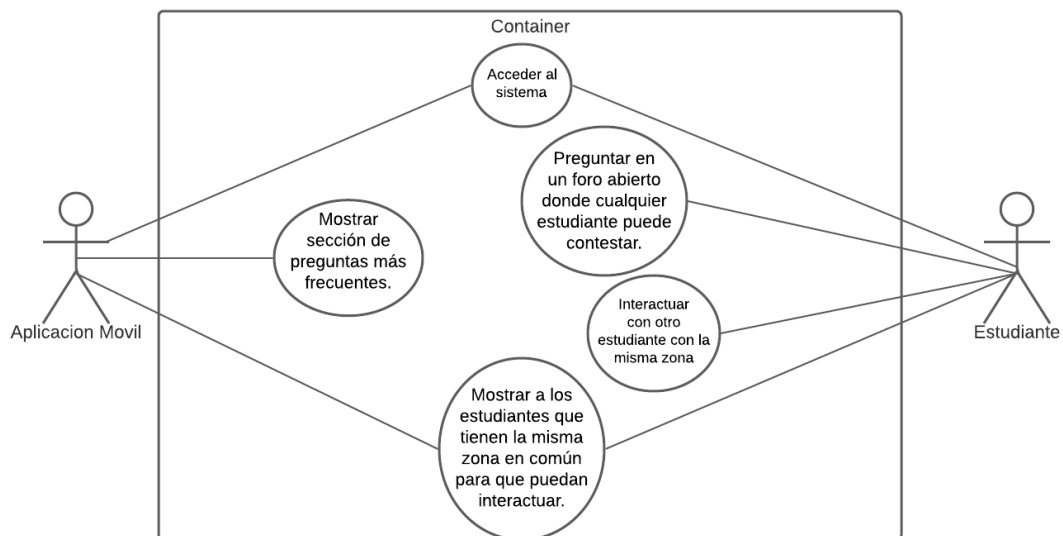
## Casos de Uso

Actores: Aplicación Móvil, Estudiantes

Acciones:

1. Acceder al sistema
2. Mostrar a los estudiantes que tienen la misma zona en común para que puedan interactuar.
3. Interactuar con otro estudiante de la misma zona.
4. Preguntar en un foro abierto donde cualquier estudiante puede contestar.
5. Mostrar sección de preguntas más frecuentes.

## Diagrama UML



## Definición de casos de uso

Definición de caso 1: Acceder al sistema

Actores involucrados: Aplicación móvil, Estudiante

Descripción:

1. Descargar la aplicación móvil.
2. Ingresar con tu usuario y contraseña institucional.
3. Evaluar si el estudiante está de intercambio, se fue o se va a ir.

Entradas: Usuario y Contraseña institucional

Salidas: Dar acceso al sistema

Flujos Alternos:

1. FA1. El usuario y contraseña no está registrado en la institución.
  - a. Mostrar mensaje de error.
  - b. Regresar al paso 2
2. FA2. El estudiante no está, fue ni se va a ir de intercambio.
  - a. Mostrar mensaje de error
  - b. Terminar programa

Definición de caso de uso 2: Mostrar a los estudiantes que tienen la misma zona en común para que puedan interactuar.

Actores involucrados: Aplicación Móvil.

Descripción:

1. La aplicación da acceso al estudiante.
2. El estudiante registra si está, se fue o se va a ir de intercambio.
3. Se registra el estatus del estudiante.
4. El estudiante ingresa su lugar de intercambio.
5. La aplicación muestra a todos los estudiantes con la misma zona en común y acomodados por su estatus.

Entradas: El estatus del estudiante, lugar de intercambio del estudiante

Salidas: Sección donde se muestran los estudiantes con los cuales puede interactuar.

Flujos Alternos:

1. FA1. El lugar que ingresa el estudiante no está registrado en la base de datos de la aplicación.
  - a. Despliega mensaje de solicitar verificar el lugar que ingresó.
  - b. Volver al paso 4.

Definición de caso de uso 3: Interactuar con otro estudiante de la misma zona.

Actores involucrados: Estudiante

Descripción:

1. Seleccionar la persona de la lista mostrada por el sistema
2. Enviar solicitud para comenzar a interactuar
3. Una vez la otra persona acepte la solicitud, se abrirá un chat privado para las dos personas.

Entradas: Persona Seleccionada

Salida: Chat privado de las dos personas.

Flujos alternos:

1. FA1. La persona a la que le enviaron solicitud no acepta interactuar.
  - a. Mostrar mensaje que rechazaron la solicitud.
  - b. Volver al paso 1

Descripción de caso de uso 4: Preguntar en un foro abierto donde cualquier estudiante puede contestar.

Actores involucrados: Estudiante

Descripción:

1. Ingresar al foro abierto
2. Realizar una pregunta pública
3. Esperar a que le respondan

Entradas: Pregunta sobre los intercambios

Salidas: Respuestas de cualquier persona.

Flujos alternos:

1. FA1. Nadie responde la pregunta.
  - a. Volver al paso 2

Descripción de caso de uso 5: Mostrar sección de preguntas más frecuentes.

Actores involucrados: Aplicación Móvil

Descripción:

1. Seleccionar las preguntas más frecuentes en el foro público.
2. Desplegar las preguntas en orden de más frecuente a menos en una sección aparte.

Entradas: Preguntas del foro.

Salidas: Selección de aquellas preguntas más frecuentes.

Flujos alternos:

1. FA1. No encuentra preguntas en el foro.
  - a. Mostrar mensaje en la sección de preguntas frecuentes que diga que no han realizado aún ninguna pregunta
2. FA2. No hay preguntas que se repitan.
  - a. Mostrar mensaje en la sección de preguntas frecuentes que diga que no hay preguntas repetidas.

## **Conclusión**

Al haber realizado todo esto, nos dimos una idea de cómo queremos que el software funcione y con qué ciclo de vida creemos que sería el más eficiente para evitar contratiempos en el desarrollo. Nos dimos cuenta que el ciclo de vida del software tenía que ser de tipo incremental porque se adapta a las necesidades del usuario, ya que con este ciclo es más fácil implementar más funcionalidades. Después, viendo cuáles requerimientos son funcionales y cuáles no, nos dimos cuenta cuáles acciones son parte del sistema, y cuáles son las funciones que el usuario verá en el sistema. Después utilizando el diagrama UML vimos que funciones aplican para ambos actores. Finalmente, con los casos de usos, se plantea mejor cómo funciona el software y cuáles serían los casos dependiendo lo que el usuario ponga en el sistema.

