

Instituto Tecnológico de Estudios Superiores de Monterrey



Evidencia final: Documento SRS

Evidencia del análisis de requerimientos de Software: documento SRS

Unidad Formativa:

Implementación del internet de las cosas Grupo 12

(TI 1015.12)

Alumnos (Equipo #6):

Jorge Daniel Cruz Case - A01634536
Álvaro Misael Pintor Alcantar - A01631795

Campus:

Guadalajara

Propósito del documento

El siguiente documento SRS tiene como propósito el análisis de la situación problema de la unidad formativa “TI 1015”, así como la definición de los requerimientos establecidos por la misma o por el mismo cliente. Además se definirán características importantes para el desarrollo del proyecto, como los actores, actividades del sistema y sus interacciones con los usuarios. Por último, tomando en cuenta los requerimientos y características definidas mediante el análisis de la problemática, se escogerá una metodología de desarrollo para el proyecto.

Dentro del mismo documento se hará uso de diversas herramientas tecnológicas para la construcción de las visualizaciones importantes para el desarrollo del proyecto, tales como, diagramas UML de casos de uso o de actividades. Esto nos permitirá realizar correctamente la validación y verificación de los requerimientos establecidos, así como nos permitirá conocer las limitaciones/restricciones del sistema.

Definición de la problemática

Dentro de la “Fábrica digital tec 21”, se nos asignó el desarrollo de una aplicación móvil para estudiantes que estén dentro o próximos a un intercambio dentro del plan de estudios Tec 21. La aplicación tiene el propósito de permitir a estos mismos estudiantes a conectar entre ellos, socializar e interactuar entre ellos. También para resolver inquietudes y compartir experiencias entre ellos o con una persona de apoyo.

Análisis de la problemática

Debido a la naturaleza del problema, será necesario un análisis más profundo de las necesidades o características deseadas de la aplicación, esto debido a que la definición de la aplicación es muy vaga.

Este tipo de limitaciones creadas por la ambigüedad del problema pueden ser resueltas mediante juntas o entrevistas con el cliente, en las mismas se podrán identificar las funcionalidades y atributos que la aplicación a desarrollar deberá tener. Así mismo, se pueden identificar requerimientos de la aplicación, tanto funcionales como no funcionales.

Una vez identificadas las funcionalidades y requerimientos de la aplicación, se podrá elegir un método de desarrollo eficiente que cumpla con las especificaciones del cliente.

Definiciones, acrónimos y abreviaciones

Para la resolución de la problemática asignada, así como el correcto desarrollo del proyecto deseado, es necesario tener un buen entendimiento de los conceptos, definiciones y abreviaturas utilizadas durante el documento.

- Ingeniería de requerimientos, Al proceso de **Descubrir, Analizar, Especificar/Documentar, Verificar y Validar** los servicios proporcionados por un software, así como sus restricciones, se le conoce como la ingeniería de requerimientos. (González, 2021).
- Requerimientos, estos no tienen una definición consistente dentro de la industria del software, estos cuentan con una dualidad en su entendimiento funcionamiento o uso. “Un requerimiento se visualiza como una declaración abstracta de alto nivel de un servicio que debe proveer el sistema o como una restricción de este” (Sommerville, s.f). Así como una definición detallada y formal de una función del sistema.
 - *Funcionales*, Son servicios que el sistema debe de proveer, así como su comportamiento ante situaciones específicas.
 - *No funcionales*, Son limitaciones sobre servicios o funciones que ofrece el sistema. (González, 2021).
- Diagrama de casos de uso, diagrama comúnmente utilizado para definir las interacciones que se tienen dentro de un sistema, así como del sistema con su entorno. Este cuenta con, **Actores, Casos de uso, Limitaciones del sistema, Asociaciones, Extensiones e inclusiones**. (González, 2021).
- Diagrama de actividades, Este representa el comportamiento de un sistema, enfatizando la secuencia de actividades que lleva a cabo y las condiciones que guardan estas actividades. Estos cuentan con, **Nodos iniciales y finales, Acciones, flujo de control, decisiones y fusiones**. (González, 2021).
 -
- Validación, también conocida como la validación de requerimientos, es el proceso de verificar que los requerimientos definidos se ajusten realmente al sistema que quiere el cliente. (Sommerville I.)

- Actores, se le conoce como actores, a aquellas entidades comúnmente externas al sistema, estos interactúan o tienen una relación con el mismo. Un ejemplo de un actor es el usuario o incluso el mismo sistema.
- Modelo de ciclo de vida, un modelo de ciclo de vida se refiere al método de desarrollo que se utiliza dentro del desarrollo de un proyecto de software, véase los métodos ágiles como SCRUM. Estos dictan el flujo de desarrollo, así como eventos importantes y fechas a considerar.

Actores del sistema

Los actores de un sistema son todas aquellas entidades que son comúnmente externas al sistema en sí que interactúan y guardan una relación con el mismo. Los actores en este caso para nuestro sistema son los siguientes:

- Estudiante: Este tipo de actor será el que mayormente use la aplicación pues la mayor cantidad de usuarios dentro del sistema serán de este tipo. Cada estudiante será capaz de construir su propio perfil con la información que el desee y tiene la característica de poder interactuar con otros estudiantes.
- Sistema: El sistema en sí mismo lo consideramos un actor, ya que interactúa con todos los tipos de usuarios y es necesario tener en cuenta su papel en muchas sino todas las interacciones del sistema.
- Mentor: Este actor será el encargado de orientar y resolver dudas de los estudiantes acerca del proceso de intercambio o del ambiente del campus al cual se irá de intercambio por lo que el mentor tiene que tener la característica de pertenecer a dicho campus.
- Encargado de comunicaciones del TEC: Este actor será el encargado de crear los anuncios y publicaciones que aparecerán para todos los usuarios en la parte principal del Feed.

Características de los usuarios:

- Un usuario común, es decir un estudiante este será una persona de entre 15 a 21 años aproximadamente. Será obligatoriamente alguien que esté próximo a su intercambio o está dentro del mismo.

- Un administrador o Encargado de comunicaciones del TEC, será un docente de sistemas o comunicaciones especializado en la actualización de información dentro de la aplicación.
- Un mentor será un usuario especial, el cual será un tutor o mentor estudiantil escogido por el Tec. Este si es un tutor sabrá acerca del comportamiento de los alumnos y podrá ayudarlos en temas psicológicos, y si es un mentor podrá ayudar con su experiencia en la escuela.

Limitaciones/Restricciones del sistema:

- La presente aplicación estará disponible solo para estudiantes pertenecientes al ITESM así como para otras universidades que cuenten con algún tipo de convenio de intercambio y será posible ser descargada por medio de la AppStore y google play.
- El desarrollo del proyecto deberá de ser finalizado en el periodo de tiempo especificado durante la entrevista con el cliente, en este caso, el periodo de tiempo será de 6 meses.

Modelo de desarrollo elegido:

Debido a la naturaleza de la aplicación requerida, la misma será utilizada más que nada por estudiantes del Tec de Monterrey. Es conocido que las aplicaciones del Tec de Monterrey no son las mejores, ni en usabilidad, ni en disponibilidad, estas tienden a ser muy lentas, a fallar y a ser poco claras en sus instrucciones o utilidades. Por lo cual, si estamos desarrollando una aplicación para estudiantes o docentes del Tec de Monterrey, y tomando en cuenta lo dicho anteriormente, sentimos que tener en mano la opinión o retroalimentación de aquellas personas que realmente usan la aplicación es esencial para el desarrollo de la misma. Por esto decidimos irnos por un modelo de desarrollo de tipo Incremental, esto debido a que es necesario tener en cuenta a aquellas personas que utilizarán la aplicación constantemente.

Diagrama de modelación del proyecto:



Este ciclo de 4 fases se repetirá el número de iteraciones necesarias hasta que el proyecto esté completamente acabado, es decir que no hay un número de iteraciones definidas, pues cada proyecto es diferente.

Requerimientos formulados a través de la entrevista

Requerimientos Funcionales:

1. Se tendrá un registro, formulario o base de datos con los datos personales del usuario, los cuales serán:
 - a. Nombre(s)
 - b. Apellidos
 - c. Campus
 - i. De origen
 - ii. De destino de intercambio
 - d. Matricula
 - e. Carrera
 - f. Edad
 - g. Semestre actual, variando de semestre máximo dependiendo de la carrera.
 - h. Contraseña
 1. La contraseña deberá ser de mínimo 8 caracteres y deberá tener por lo menos un número, una letra mayúscula, 1 letra minúscula y 1 carácter especial.

- i. Correo electrónico institucional
 - i. Contando con el “@tec.mx” o “@itesm.mx”
 - j. Contacto del alumno o tutor
 - i. Emergencia
 - ii. Personal
2. Se tendrá un proceso de inicio de sesión para la aplicación, la cual pedirá lo siguiente:
 - a. Nombre de usuario, Correo electrónico institucional “@tec.mx” o “@itesm.mx” o su matrícula.
 - b. Una contraseña de al menos 8 caracteres, mínimo 1 número, una letra mayúscula y un carácter especial.
3. Se le mostrará al usuario 4 menús, uno después del otro, en estos se le mostrarán ciertas actividades o gustos para entender al usuario, estos menús tendrán las siguientes características:
 - a. Se tendrá un banco de preguntas dentro de la memoria interna de la aplicación o desde una base de datos
 - i. Las preguntas serán ingresadas por un administrador cada 2 a 3 meses.
 - ii. Las preguntas pueden llegar a ajustarse dependiendo de las respuestas ingresadas por el usuario en menús pasados.
 - b. Las respuestas a las preguntas serán almacenadas en el mismo lugar dónde se almacenan los datos personales del usuario en el punto 1.
4. Dependiendo de los datos ingresados por el usuario como carrera o semestre, se le pondrá en contacto con un asesor académico:
 - a. El asesor tendrá acceso a los datos personales de sus alumnos mediante una llave única generada por la aplicación la cual le permitirá acceder a la misma, la clave será construida de la siguiente manera:
 1. Los primeros 2 dígitos serán 2 caracteres que reflejarán su posición dentro del tec de Monterrey al momento de ser escogido para la aplicación, ejemplo, Alumno es A0, docente L0 & trabajador T0.
 - a. Estos dígitos están propuestos a cambio.
 2. Los siguientes 5 dígitos corresponden a los últimos 5 dígitos de su clave institucional.
 3. Los siguientes 2 dígitos corresponden al campus al que pertenece, por ejemplo si es del campus Monterrey los dígitos serán 01.

4. Los últimos 2 dígitos corresponden al año de entrada al programa de la aplicación.
 - b. La llave única tendrá un formato así, “L0-37654-01-21”, teniendo que ser ingresada sin los guiones.
 - c. La contraseña se le hará llegar por medio del correo que haya sido utilizado para crear la cuenta en la aplicación.
 - i. De no ser posible hacer contacto con el asesor se contactará a un asesor de repuesto.
5. El programa tomará los datos ingresados por el usuario y generará un perfil general dependiendo de lo siguiente:
 - a. Edad
 - b. Nombre(s) y apellidos
 - c. Campus de residencia actual
 - d. Género
 - i. Hombre
 - ii. Mujer
 - iii. Otro
 - e. Hobbies o pasatiempos
 - f. Foto de perfil elegida por el usuario
 - i. Formatos: PNG, JPEG, JPG.
 - g. Una biografía con un pequeño resumen de aquellas cosas que el usuario cree que los demás deben de saber:
 - i. Máximo 300 caracteres
 - ii. Compatible con links de otras redes sociales
6. La aplicación asignará un grupo de personas que tengan datos parecidos a aquellos ingresados por el usuario, estos dependen de:
 - a. Campus de residencia actual
 - b. Campus de origen
 - c. Hobbies o pasatiempos
7. Se contará con un sistema de mensajes instantáneos para la comunicación entre los usuarios de la aplicación:
 - a. Contará con un buscador específico de usuarios para la comunicación, el usuario deberá ingresar:
 - i. Nombre completo del usuario a buscar

- ii. Correo electrónico institucional
 - iii. Matrícula
- b. Los mensajes contarán con:
 - i. Usuarios activos
 - ii. Chats recientemente abiertos
 - iii. Crear un nuevo chat
 - iv. Crear un grupo
 - v. Buscar a un usuario o a un chat específico
- c. Se podrá mandar mensajes sin un límite de caracteres
- d. Se podrá mandar mensajes de voz con un máximo de 2 minutos
- e. Se podrá mandar archivos
 - i. Multimedia
 - 1. Máximo de 64 kBytes en fotos
 - 2. 128 kBytes en audios o videos
 - ii. Misceláneos
 - 1. Archivos de texto
 - 2. Programación
- 8. El menú de inicio de sesión deberá desplegarse cada vez que se abra la aplicación.
 - a. Se le pedirá al usuario si desea guardar sus datos de sesión para autocompletar en usos posteriores.
 - b. Se deberá mostrar una opción posterior al inicio de sesión preguntándole al usuario si desea guardar sus datos de sesión para autocompletar en usos posteriores, o que no se cierre la sesión al cerrar la aplicación.
 - c. Contará con una función de recuperación de contraseña, llegara el caso de que se perdiera.
 - i. Se le mandará al usuario un correo de restablecimiento de contraseña.
 - 1. Etc

Requerimientos No Funcionales:

- 1. Los datos privados deberán de estar privados del ojo público, siendo posible su acceso por el mismo alumno o por su asesor:

- a. Al momento de querer acceder a estos datos se le solicitará al usuario ingresar su contraseña. Si el solicitante es el asesor, este deberá ingresar su clave única proporcionada por el sistema.
 - b. De fallar al proporcionar credenciales correctas, se le negará el acceso y se le avisará al usuario de actividad sospechosa en su perfil, el aviso llegará por:
 - i. Correo electrónico institucional
 - ii. Mensaje directo por la aplicación
2. Los usuarios de la aplicación podrán acceder a datos generales del usuario mediante su perfil, estos datos serán:
 - a. Nombre(s) y apellidos
 - b. Correo electrónico institucional
 - c. Matrícula
 - d. Campus de origen y residencia actual
 - e. Género
 - f. Biografía
 - g. foto de perfil
3. El usuario tendrá la posibilidad de modificar opciones de visibilidad y privacidad:
 - a. Que usuarios podrán ver su perfil
 - i. Usuarios elegidos por el usuario
 - ii. Usuarios con mensajes recientes
 - iii. Nadie
 - iv. Usuarios del mismo campus de residencia actual
 - v. Usuarios del mismo campus de origen
 - vi. Usuarios de su grupo asignado por la aplicación
 - vii. Público
 - b. Que usuarios le podrán mandar mensajes
 - i. Usuarios del mismo campus de residencia actual
 - ii. Usuarios del mismo campus de origen
 - iii. Usuarios de su grupo asignado por la aplicación
 - iv. Usuarios selectos
 - v. Nadie
 - vi. Público
4. La aplicación está abierta a ser programable a cualquier lenguaje que se crea óptimo:

- a. C++ por su optimización y utilidad
 - b. Java por su compatibilidad con sistemas operativos de un celular
- 5. La aplicación estará disponible solamente en celulares:
 - a. IOS
 - b. Android
 - c. Posible compatibilidad con sistemas windows.
- 6. La optimización de la aplicación deberá de ser alta, con tiempos de carga relativamente lentos sin contar conexiones a servidores o bases de datos:
 - a. Algoritmos con complejidad computacional baja, de preferencia evitar algoritmos con complejidad de $O(n^2)$.
 - b. Los procesos deberán de correr a menos de 10000 ms.
- 7. La aplicación contará con opciones de accesibilidad:
 - a. Menú del idioma que se prefiera para la aplicación
 - i. Inglés
 - ii. Español
 - b. Opciones para el cambio de los colores de la aplicación:
 - i. Modos daltónicos
 - ii. Modo oscuro / brillante
- 8. La aplicación debe de tener un diseño moderno y elegante, siendo fácil de usar y entender:
 - a. El usuario debe de entender las opciones de la aplicación sin mucho problema
 - b. El usuario debe de poder encontrar lo que busca sin mucho problema
 - c. Los colores utilizados deben ser amigables con la vista
 - d. La UI / UX debe de seguir los estándares puestos por el Tec de Monterrey y su curso de IoT.
- 9. El proceso de desarrollo del software deberá de ser abierto hacia los clientes, ellos teniendo control de lo que se hace:
 - a. Los clientes recibirán actualizaciones mensuales acerca del proceso o avance que se lleva a cabo de la aplicación.
 - b. El proceso será transparente para el cliente.

Conclusión del proyecto

El sistema que se nos dio de tarea desarrollar es una aplicación con el propósito de permitirle a estudiantes próximos a su intercambio o que ya se encuentran en este la comunicación entre ellos, capacidad de compartir sus experiencias y resolver dudas que se presenten entre estos.

Para la solución de la problemática se emplearon distintos métodos de desarrollo de software y ingeniería de requerimientos, como la identificación de actores, definición de usuarios, planteamiento del modelo de vida del desarrollo del proyecto, así como herramientas para visualización de requerimientos como diagramas UML (casos de uso y actividades) y entrevistas y preguntas con el cliente para solución de dudas y validación de requerimientos. El sistema planteado cuenta con una gran cantidad de opciones para satisfacer lo establecido en la problemática, como interacción entre usuarios, solución de dudas y compartimiento de experiencias.

También se manejan funciones establecidas en los requerimientos, manejandolos de la manera en que todas las funciones cuenten con características básicas e importantes, así como métodos de mantenimiento o consulta.

Conclusiones del curso

El análisis de requerimientos de software es el primer paso para el proceso de desarrollo de software, o por lo menos debería serlo, ya que abre paso a todos los demás pasos del proceso de desarrollo.

A lo largo de este curso comprendimos la importancia de la planeación y estandarización del software analizando casos reales como fue la “Crisis del software” en 1970, lo que nos dio una idea de lo que pasaría si nos brincamos todo este proceso y solo programamos.

Gracias a esta materia analizamos las ventajas que tiene una buena planeación en un proyecto de software ya que podemos disminuir la probabilidad de tener fallas en la finalización de éste.

Analizamos los modelos de ciclo de vida, como los métodos ágiles para tiempos de desarrollo cortos o los métodos de cascada para un desarrollo ordenado. También aprendimos al manejo y análisis de los requerimientos de un sistema, aprendiendo a cuestionar al cliente, hacer entrevistas y tener al cliente actualizado para un correcto funcionamiento del sistema.

Así mismo, no solo analizamos teóricamente cuál es el proceso para una correcta planeación, sino que también la pusimos en práctica por medio de la situación problema planteada.

El curso en sí mismo nos pareció muy interesante y muy completo ya que brinda una muy buena perspectiva de qué esperar cuando realizamos un gestionamiento de proyectos de software.

Lecciones aprendidas

A lo largo de este curso aprendimos muchos conceptos nuevos así como la importancia de hacer una buena planificación de un proyecto de software.

Algunos de los conceptos que más nos llevamos fue el de un buen análisis de requerimientos de software. Aprendimos que hay diferentes maneras de obtener los requerimientos de un proyecto, y uno de ellos es por medio de una entrevista directamente con el cliente. Esto tiene muchas ventajas ya que si el encargado del desarrollo de software está presente en dicha entrevista, puede asegurarse de entender más exactamente lo que en realidad quiere el cliente.

Así mismo, comprendimos que este paso es sumamente importante ya que en caso de que este paso se haya hecho de manera incorrecta, entonces el cliente obtendrá algo completamente diferente a lo que necesitaba. Es por esto que hay una serie de pasos a seguir para la recabación de dichos requerimientos y que no sean ambiguos, no hay que dejar nada a la interpretación.

Otro concepto que se nos hizo muy importante fue el de “modelo de desarrollo de software”. Aprendimos que hay diferentes tipos de modelos y analizamos sus pros, sus contras y sus diferencias. Así mismo, identificamos los diferentes roles que hay en un modelo de desarrollo y analizamos cuál era su trabajo.

Este curso nos brindó muchas lecciones las cuales sin duda las podremos utilizar en el desarrollo de próximos proyectos.

Anexos

Diagrama de actividades:

<https://drive.google.com/file/d/1u-DboGiA2c8IBFuR506W6BW1wfd0-Rxa/view?usp=sharing>

Diagrama de casos de uso:

https://drive.google.com/file/d/1EZld59fD1HhXabRosOF_CGX6Rqm0qH0J/view?usp=sharing

Link al video de la entrevista:

<https://drive.google.com/file/d/1SDAGBO9goEQyIDxSiO1IjM0zWUYcMzkS/view?usp=sharing>

Referencias

González C., Partida A., Sanromán A. (2021). *Módulo 2_2: Tipos de requerimientos*.

[Archivo PDF]. Recuperado de:

https://tecmx-my.sharepoint.com/:p:/g/personal/ana_sanroman_tec_mx/EcleOagFv9RBk5nUk7Yza7sB9Nu-CfAkymd0GtA2KmaaXg?e=7a3cl1.

González C., Partida A., Sanromán A. (2021). *Módulo 3: 3 Diagrama de actividades UML*.

[Archivo PDF]. Recuperado de:

https://tecmx-my.sharepoint.com/:p:/g/personal/ana_sanroman_tec_mx/ESUV0fhQEWJAHEFlN5J8uAIBC53fHyUn4joHrPrGV2uN0A?e=KiNcx9

González C., Partida A., Sanromán A. (2021). *Módulo 3: 3 Diagrama y documentación de Casos de USO*. [Archivo PDF]. Recuperado de:

https://tecmx-my.sharepoint.com/:p:/g/personal/ana_sanroman_tec_mx/EWeX1WFKNENKo-QTeY3FTZQB2q421U8sZSP7OCrRjKI5g?e=ZadsJMhttps://tecmx-my.sharepoint.com/:p:/g/personal/ana_sanroman_tec_mx/ESUV0fhQE

González C., Partida A., Sanromán A. (2021). *TC1015-Módulo 2: Ingeniería de requerimientos*. [Archivo PDF]. Recuperado de:

https://tecmx-my.sharepoint.com/:p:/g/personal/ana_sanroman_tec_mx/EcTvfx0XRCJHmKkxKjbS7TwBLF7D895TD2ZT1qsYHETPUg?e=OxhWU2

Morales I. (2017). *Hablemos un poco sobre Ingeniería de Requerimientos*. U-gob.

Recuperado de:

<https://u-gob.com/hablemos-un-poco-sobre-ingenieria-de-requerimientos/>

Sommerville I. (s.f.). *Requerimientos de software, 6ª edición, capítulo 5*. [Archivo PDF].

Recuperado de:

[https://www.uv.mx/personal/fcastaneda/files/2015/08/F_Capitulo_5_Requerimientos_
del_software.pdf](https://www.uv.mx/personal/fcastaneda/files/2015/08/F_Capitulo_5_Requerimientos_del_software.pdf)