



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Universidad Nacional de Colombia - sede Bogotá
Facultad de Ingeniería
Departamento de Sistemas e Industrial
Curso: Ingeniería de Software 1 (2016701)

Taller 01

Lista de los requerimientos identificados dados por el usuario.

1. Un sistema que centraliza las actividades e información de la empresa.
2. Cada sistema debe contar con un encargado principal y permitir la creación de roles de otro tipo de empleados: "Personal".
3. Asignación de reglas comunes entre refugios, como plantillas pre-establecidas para donaciones.
4. Ingreso y registro de personas.
5. Historial de estancia de cada usuario.
6. Registro y seguimiento de donaciones.
7. Aplicación funcional en celulares android y computadores.
8. Seguimiento de los turnos del personal.
9. Pre-registro.
10. Seguimiento de disponibilidad de camas.
11. Búsqueda por filtros.
12. Permitir un chat entre sedes con el fin de transmitir información de manera eficiente.
13. Panel de vista general

Clasificación de los requerimientos en Funcionales y No Funcionales

<hr/>	
FUNCIONALES	NO FUNCIONALES
<hr/>	
Ingreso y registro de personas	Aplicación funcional en celulares Android y sistema operativo Windows.
Historial de estancia de cada usuario.	
Registro y seguimiento de donaciones.	Un sistema que centraliza las actividades e información de la empresa.
Seguimiento de los turnos del personal.	Plantillas pre-establecidas para el registro de ingresos, donaciones, etc.
Permitir un Pre-registro dentro del sistema en otras sedes tales como iglesias.	
Seguimiento de disponibilidad de camas.	
Búsqueda por filtros.	
<hr/>	

Vista general del sistema.

Cada sistema debe contar con un encargado principal y permitir la creación de roles de otro tipo de empleados: "Personal".

Panel de vista general.

Apartado de chat entre sedes con el fin de transmitir información de manera eficiente.

NOTA: Aquí tomamos y empezamos a darle un Alias a los requerimientos, en la literatura a veces encontrarán para acortar RF (requerimiento funcional) y RNF (requerimiento no funcional) entonces lo manejaremos así por el resto del curso.

El usuario de rol Y deberá poder administrar X cosas por medio de una interfaz Y cosa

Mo	S	Co	W
RF_1: Ingreso y registro de personas por medio de un formulario en donde se detallen los siguientes datos (nombre completo,numero de cedula(si tiene), edad, sexo, ciudad de origen, estado de salud, fecha y hora de ingreso, firma o huella, observaciones adicionales, numero de cama a asignar)	RF_7: El usuario con rol Personal deberá poder buscar la información de Personas refugiadas o en el centro por fechas, nombre o ID, asimismo, como el historial de instancia de personas y turnos del personal	RF_11: El usuario administrador y Personal deberá poder tener acceso a un chat en tiempo real con miembros de su misma sede, junto con otro chat con otras sedes	RF_13: Permitir pre-registro desde otras sedes (como iglesias) mediante un formulario remoto vinculado al sistema.
RF_2: El usuario de rol Personal deberá poder ver la disponibilidad de camas por medio de una interfaz: Ocupación de camas	RF_8: El usuario con rol Administrador deberá poder visualizar y filtrar la información de Ingreso y registro de personal, historial de instancia, asignación de camas, turnos de personal, necesidades urgentes por medio de una interfaz: Panel general de los centros.	RF_12: El usuario administrador y Personal deberá poder tener acceso a una interfaz en donde se establezcan necesidades urgentes de la sede(s).	
RF_3: El usuario de rol Personal podrá registrar el ingreso de donaciones.	RF_9: Búsquedas con filtros avanzados en todos los módulos: personas, turnos, camas, donaciones, etc.	RF_14: Plantillas pre-establecidas para el registro de ingresos, donaciones, etc.	

RF_4: El usuario de rol Administrador podrá dar seguimiento a las donaciones: Ver donaciones, generar la solicitud de traslado de centro de las donaciones, cambiar la ubicación de sede de las donaciones por medio de una interfaz control donaciones	RF_10: Chat en tiempo real entre miembros de la misma sede y con otras sedes (acceso para personal y administrador).		
RF_5: El usuario administrador podrá otorgar roles a empleados, tales como "Personal".	RNF_4: Interfaz intuitiva y accesible para usuarios con conocimientos técnicos básicos		
RF_6: El administrador podrá ver el historial de estancias de las personas	.		
RNF_1: El sistema deberá estar disponible como aplicación móvil (Android) y aplicación web compatible con Windows.			
RNF_2: La arquitectura del sistema debe centralizar la información de todas las sedes.			
RNF_3: Disponibilidad del sistema del 99% anual con soporte para al menos 100 usuarios concurrentes.			
RNF_4: Cifrado de datos sensibles y cumplimiento de normativas de protección de datos personales.			

1. Organizar por MoSCoW

NOTA: desde este punto los requerimientos pasan a ser requisitos debido a que como desarrolladores hemos observado las implicaciones (ya existe un análisis).

MUST (DEBE - OBLIGATORIO SIN ÉL EL SISTEMA PIERDE SENTIDO O NO FUNCIONA)

- RF_1: Historial de estancia
- RF_2: Roles y permisos de usuarios
- RNF_1: Un sistema que centraliza las actividades e información de la empresa, en donde todos los funcionarios de la empresa usen la misma aplicación.
 - una aplicación que comparte la información de los refugios en una misma base de datos
- RNF_2: Compatibilidad en android y windows

SHOULD (DEBERÍA - NO OBLIGATORIO PERO APORTA MUCHO A LA EXP DE USUARIO)

- RF_3: Turnos del personal
- RF_4: Panel general

COULD (PODRÍA - ES UN PLUS USUALMENTE PERSONALIZACIÓN, SE DESARROLLA USUALMENTE CUANDO QUEDA TIEMPO)

- RF_5: Plantillas pre-establecidas en para el registro de ingresos, donaciones, etc.
- RF_6: Apartado de chat entre sedes con el fin de transmitir información de manera eficiente.

WONT (NO NECESARIO - PUEDE ESPERAR AL PUNTO QUE ESTÉ O NO NO PASA NADA SI NUNCA SE DESARROLLA)

- RF_7: Permitir un Pre-registro dentro del sistema en otras sedes tales como iglesias

2. Realizar una estimación de tiempo/esfuerzo con Fibonacci sobre cada requisito de cuánto se demoraría

NOTA: usualmente se dice que los puntos de fibonacci son esfuerzo pero ese “esfuerzo” lo define el equipo, usualmente aconsejo a los equipos en los que participo que ese “esfuerzo” sea determinado en Días entonces en esta actividad la estimación será los días que consideran puede tomar el desarrollo

NOTA: Manejen colorcitos tipo así por casilla, es sugerencia, más ustedes pueden optar por facilitar la lectura y no sea un simple cuadro, pueden modificar a gusto si quieren.

Requisito	Priorización	Estimación	Argumento
RF_1	MUST	5	Es bastante fácil implementar un formulario y almacenar los datos en una base de datos, siendo que la infraestructura de la base de datos ya esté implementada
RF_2	MUST	5	Ver la disponibilidad de camas es solo

			consultar la base de datos y traerme las camas que estén disponibles y ponerlas en un espacio que haga parte de la interfaz principal por lo que es un procedimiento mas o menos sencillo
RF_3	MUST	3	Al igual que el requisito se llena un formulario (una operación básica) que luego esa información se publica en un base de datos(fácil si la infraestructura de la base de datos ya está
RF_4	MUST	13	Es complicada no por lo difícil sino porque son muchas tareas. Ya no solo es que el sistema se traiga la información de la base de datos sino que el usuario necesita una interfaz sencilla en la que se pueda rastrear las donaciones y aparte también se necesita interactuar para generar las solicitudes de traslado
RF_5	MUST	5	En general no es tan difícil ya que con un campo role en la tabla de usuario de la base de datos y un middleware para verificar permisos se podría hacer más o menos rápido
RF_6	MUST	5	Al igual que las camas es pedirle una solicitud a una api que me traiga la lista de estancias de una persona desde la base de datos y ponerlas en un espacio de la interfaz general, puede ser una interfaz específica de personas
RNF_1	MUST	5	No es tan difícil ya que algunas tecnologías para desarrollar software solo necesitan pequeños cambios para tener un producto en varios tipos de dispositivos
RNF_2	MUST	21	Es bastante complicado ya que se necesitan primero manejar una base de datos compartida que funcione en tiempo real, también se necesita un buen sistema de seguridad, se necesita conexión a internet en todas las sedes, y una gestión de fallos para que un error en una sede no afecte las demás
RNF_3	MUST	8	Si se tiene buen capital se puede contratar un servicio confiable que me permita manejar esa cantidad de usuarios. El único tema sería gestionar con el proveedor
RF_7	SHOULD	5	No es complicado, se necesita construir una api adecuada , así mismo una interfaz que permita estas búsquedas de manera intuitiva

RF_8	SHOULD	5	.Al igual que el anterior se necesita hacer un backend adecuado, pero no debería ser tardado
RF_9	SHOULD	8	Construcción de backend, procedimiento usual(filtros), pero más filtros
RF_10	SHOULD	13	Un chat en tiempo real puede ser nivel moderado ya que se necesitan ciertos servicios como webSockets y manejar conexiones activas
RNF_4	SHOULD	8	Se necesita tener conocimientos de UI y UX y se necesita diseñar cada interfaz de la manera correcta. Puede que no sea tan difícil si se tiene en cuenta la cantidad de plantillas ya probadas que hay publicadas
RNF_2	MUST	5	No es tan difícil ya que algunas tecnologías para desarrollar software solo necesitan pequeños cambios para tener un producto en varios tipos de dispositivos
RF_12	COULD	8	Es hacer la interfaz con un espacio de texto si se quiere y un espacio donde están publicadas las necesidades de las otras sedes esto se traería desde la base de datos compartida
RF_13	WON'T	5	Es sencillo porque es darle el sistema a otra persona y asignarle un rol, así podrá hacer un preregistro por medio de un formulario

NOTA FINAL:

Hasta este punto hemos transformado los requerimientos en requisitos, y hemos realizado una priorización y estimación de las tareas iniciales, esto por ahora es un borrador, es lo que se mostrará al cliente/stakeholder para su validación, cuando tenemos una aprobación procedemos a la siguiente sección, creación de Historias de usuario y Casos de uso.