

Taller_01.pdf

Cristian Leonardo Castañeda Olarte

Alan Ryan Cheyne Gómez

Juan Jerónimo Gómez Rubiano

David Santiago Velásquez Gómez

- 1.** Generar una lista de todos los requerimientos identificados dados por el usuario.
 - (req1) Coordinación interna del refugio
 - (req2) Confirmar disponibilidad de camas en tiempo real
 - (req3) Implementación de un protocolo general
 - (req4) Aplicabilidad del protocolo teniendo en cuenta las necesidades de cada equipo
 - (req5) Crear un sistema de ingresos
 - (req6) Consulta de información histórica de los refugios
 - (req7) Registro unificado de donaciones
 - (req8) Implementar un sistema de pre-registro
 - (req9) Almacenar y agendar turnos de personal
 - (req10) Manejar un panel general de la aplicación
 - (req11) Funcionalidad en Desktop y Mobile
 - (req12) Implementar primer versión funcional en tres meses
 - (req13) Sistema que se ajuste a un presupuesto reducido
 - (req14) Asegurar que el sistema sea fácil e intuitivo de usar
 - (req15) Implementar un sistema único para todos los centros
 - (req16) Filtrado de información, por fechas, centros, personas, etc..
 - (req17) Definir roles de los usuarios
 - (req18) Autenticación y acceso a diferentes funcionalidades según el rol del usuario.

- 2.** Dividir los requerimientos en funcionales o no funcionales

FUNCIONALES	NO FUNCIONALES
<ul style="list-style-type: none">- (RF_1) Confirmar disponibilidad de camas en tiempo real- (RF_2) Implementación de un protocolo general- (RF_3) Crear un sistema de ingresos- (RF_4) Consulta de información histórica de los refugios- (RF_5) Registro unificado de donaciones	<ul style="list-style-type: none">- (RNF_1) Coordinación interna del refugio- (RNF_2) Aplicabilidad del protocolo teniendo en cuenta las necesidades de cada equipo- (RNF_3) Funcionalidad en Desktop y Mobile- (RNF_4) Implementar primer versión funcional en tres meses

<ul style="list-style-type: none"> - (RF_6) Implementar un sistema de pre-registro - (RF_7) Almacenar y agendar turnos de personal - (RF_8) Manejar un panel general de la aplicación - (RF_9) Implementar un sistema único para todos los centros - (RF_10) Filtrado de información, por fechas, centros, personas, etc.. - (RF_11) Definir roles de los usuarios 	<ul style="list-style-type: none"> - (RNF_5) Sistema que se ajuste a un presupuesto reducido - (RNF_6) Asegurar que el sistema sea fácil e intuitivo de usar
--	--

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.

3. 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_3) Crear un sistema de ingresos
- (RF_7) Almacenar y agendar turnos de personal
- (RF_9) Implementar un sistema único para todos los centros
- (RF_11) Definir roles de los usuarios
- (RNF_1) Coordinación interna del refugio
- (RNF_5) Sistema que se ajuste a un presupuesto reducido

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

- (RF_1) Confirmar disponibilidad de camas en tiempo real
- (RF_2) Implementación de un protocolo general
- (RF_4) Consulta de información histórica de los refugios
- (RF_5) Registro unificado de donaciones
- (RF_8) Manejar un panel general de la aplicación
- (RNF_4) Implementar primer versión funcional en tres meses
- (RNF_6) Asegurar que el sistema sea fácil e intuitivo de usar
- (RF_10) Filtrado de información, por fechas, centros, personas, etc..

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

- (RF_6) Implementar un sistema de pre-registro
- (RNF_2) Aplicabilidad del protocolo teniendo en cuenta las necesidades de cada equipo

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

- (RNF_3)Funcionalidad en Desktop y Mobile

Tabla general:

Must	<ul style="list-style-type: none"> - (RF_3) Crear un sistema de ingresos - (RF_7) Almacenar y agendar turnos de personal - (RF_9) Implementar un sistema único para todos los centros - (RF_11) Definir roles de los usuarios - (RNF_1)Coordinación interna del refugio - (RNF_5) Sistema que se ajuste a un presupuesto reducido
Should	<ul style="list-style-type: none"> - (RF_1) Confirmar disponibilidad de camas en tiempo real - (RF_2) Implementación de un protocolo general - (RF_4) Consulta de información histórica de los refugios - (RF_5) Registro unificado de donaciones - (RF_8) Manejar un panel general de la aplicación - (RNF_4) Implementar primer versión funcional en tres meses - (RNF_6) Asegurar que el sistema sea fácil e intuitivo de usar - (RF_10) Filtrado de información, por fechas, centros, personas, etc..
Could	<ul style="list-style-type: none"> - (RF_6) Implementar un sistema de pre-registro - (RNF_2)Aplicabilidad del protocolo teniendo en cuenta las necesidades de cada equipo
Wont	<ul style="list-style-type: none"> - (RNF_3)Funcionalidad en Desktop y Mobile

4. 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_3	MUST	5	Requiere manejo de bases de datos, e implementación de API
RF_7	MUST	5	Implica trabajo ya que significa llevar a cabo un estudio en profundidad de los horarios del personal.
RF_9	MUST	13	Requiere integrar prácticamente todos los otros requisitos funcionales
RF_11	MUST	2	Muy fácil de definir en el backend
RNF_1	MUST	1	Este involucra poco trabajo ya que se ve como una suma de los demás requerimientos por lo que una vez resueltos los demás este se soluciona por sí solo
RNF_5	MUST	21	Necesita amplia experiencia en Ingeniería de Software, para hacer una buena estimación de tiempos, recursos necesarios en condiciones ideales, y estrategias para ahorrar recursos
RF_1	SHOULD	8	La actualización en tiempo real presenta desafíos adicionales de hardware y diseño de la aplicación
RF_2	SHOULD	1	Se puede diseñar con el cliente, tras esclarecer el flujo de trabajo deseado
RF_4	SHOULD	5	La implementación tiene una lógica muy fácil, pero requiere consulta y transcripción de datos manuales y electrónicos no integrados, lo cual consume tiempo
RF_5	SHOULD	5	Requiere estudiar el flujo de donaciones de cada centro además de los respectivos destinos de estas a la vez que hace las veces de inventario.
RF_8	SHOULD	2	Los requerimientos de backend son bastante sencillos, sobre todo comparados con los otros requerimientos funcionales
RF_10	SHOULD	3	Es fácil realizar queries de bases de datos para obtener consultas específicas
RNF_4	SHOULD	21	Dependiendo del equipo de trabajo, implementar todo en este periodo de tiempo puede ser desafiante
RNF_6	SHOULD	8	Necesita primero definir el diseño básico, y luego aplicar ideas pensadas en usuarios no expertos, que permita una navegación ágil

RF_6	COULD	8	Tiene requisitos adicionales, pero es en general muy similar a RF_3
RNF_2	COULD	13	Implica estudiar todo el hardware de cada refugio y replantear la funcionalidad en base a los recursos.
RNF_3	WONT	34	Requiere replantear toda la funcionalidad del programa, así como crear su propio diseño, frontend, etc.