

Taller #1: Levantamiento de requerimientos y análisis de requisitos

Refugio Horizonte

Grupo: ZFC Fans

Integrantes:

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) Generar un sistema de organización entre los refugios y sus empleados
- (req2) Consultar en tiempo real, una base de datos que indique la disponibilidad de las camas en cada refugio.
- (req3) Definir un protocolo estándar de ingreso, gestión, y salida de un habitante de calle en el sistema.
- (req4) Aplicabilidad del protocolo teniendo en cuenta las necesidades de cada equipo
- (req5) Crear un sistema de ingresos a las distintas sedes del refugio.
- (req6) Incluir en la base de datos un registro histórico de ingresos y salidas.
- (req7) Implementar un sistema de administración financiera unificado.
- (req8) Implementar un sistema de pre-registro.
- (req9) Construir un sistema de agenda de los turnos del personal.
- (req10) Crear un aplicativo para escritorio, con una GUI que permita acceder a todas las funcionalidades definidas de la primera aplicación
- (req11) Funcionalidad en Mobile.
- (req12) Implementar la primera versión funcional de la aplicación en tres meses.
- (req13) Procurar que el desarrollo y mantenimiento de la aplicación se ajuste a un presupuesto reducido
- (req14) Asegurar que la aplicación sea fácil e intuitiva de usar.
- (req15) Implementar un sistema único para todos los centros de refugiados.
- (req16) Filtrado de información, por fechas, centros, personas, etc..
- (req17) Definir e implementar un sistema de roles para los usuarios de la aplicación.
- (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) Consultar en tiempo real, una base de datos que indique la disponibilidad de las camas en cada refugio.- (RF_2) Definir un protocolo estándar de ingreso, gestión, y salida de un habitante de calle en el sistema.- (RF_3) Crear un sistema de ingresos a las distintas sedes del refugio.- (RF_4) Incluir en la base de datos un registro histórico de ingresos y salidas.- (RF_5) Implementar un sistema de administración financiera unificado.- (RF_6) Implementar un sistema de pre-registro.- (RF_7) Construir un sistema de agenda de los turnos del personal.- (RF_8) Crear un aplicativo para escritorio, con una GUI que permita acceder a todas las funcionalidades definidas de la primera aplicación- (RF_9) Implementar un sistema único para todos los centros de refugiados.- (RF_10) Incluir en la función de consulta histórica, la habilidad de filtrar los resultados de búsqueda.- (RF_11) Definir e implementar un sistema de roles para los usuarios de la aplicación.- (RF_12) Autenticación del usuario dentro del sistema de acuerdo a su rol.- (RF_13) Crear un formato de ingreso.- (RF_14) Crear un verificador básico de aceptabilidad del refugiado.- (RF_15) Incluir un formato de condiciones especiales.- (RF_16) Crear un registro de pertenencias.- (RF_17) Crear un sistema de egresos de las distintas sedes del refugio.	<ul style="list-style-type: none">- (RNF_1) Generar un sistema de organización entre los refugios y sus empleados- (RNF_2) Aplicabilidad del protocolo teniendo en cuenta las necesidades de cada equipo.- (RNF_3) Funcionalidad en Mobile.- (RNF_4) Implementar la primera versión funcional de la aplicación en tres meses.- (RNF_5) Procurar que el desarrollo y mantenimiento de la aplicación se ajuste a un presupuesto reducido.- (RNF_6) Asegurar que la aplicación sea fácil e intuitivo de usar

3. Organizar por MoSCoW

MUST (DEBE - obligatorio sin él el sistema pierde sentido o no funciona)

- (RF_3) **Crear un sistema de ingresos a las distintas sedes del refugio.** El sistema debe incluir un *formato de ingreso*, un *verificador básico de aceptabilidad del refugiado*, una *función de asignación de camas*, un *formato de requerimientos especiales*, y un *registro de pertenencias al momento del ingreso*.
- (RF_7) **Construir un sistema de agenda de los turnos del personal.** El sistema debe tener integrado un calendario. Debe poder identificar a los usuarios del aplicativo e implementar roles (empleado, supervisor y administrador) de cada usuario. El *empleado* puede consultar la agenda, crear notas, realizar comentarios y solicitudes de cambios. El *supervisor* puede realizar todas las acciones del empleado, pero también puede crear, modificar y eliminar turnos. El *administrador* puede realizar todas las acciones del empleado, crear, modificar y eliminar usuarios, así como realizar actualizaciones a los parámetros que pueda usar la agenda (crear nuevas sedes, cambiar el número de empleados por sede, u otras excepciones a definir más adelante)
- (RF_9) **Implementar un sistema único para todos los centros de refugiados.** El sistema debe tener múltiples refugios en la base de datos. Un refugio es una etiqueta, que se aplica para las funciones de asignación de camas, datos del habitante, agenda, donaciones y canal de comunicaciones
- (RF_11) **Definir e implementar un sistema de roles para los usuarios de la aplicación.** Los roles a construir son: Empleado, Supervisor, Administrador. Se diferencian entre ellos en la asignación de permisos para el uso del aplicativo. El *empleado* puede usar el aplicativo de ingresos, salidas, modificaciones básicas a la gestión de usuarios, usar el chat grupal de la sección de comunicaciones y configuración básica del usuario. Tiene permisos de solo lectura para la agenda, otros aspectos de comunicaciones, donaciones y consultas históricas. El *Supervisor* puede manejar la gestión de usuarios, agenda y donaciones. El *Administrador* puede realizar modificaciones a los usuarios y maneja de comunicaciones (Newsletter y reglamento)
- (RF_12) **Autenticación del usuario dentro del sistema de acuerdo a su rol.** Cada usuario contará con un nombre de usuario y una contraseña con los cuales podrá ingresar al sistema. Una vez se encuentre dentro del sistema, el usuario encontrará que su dashboard abarca solo las funcionalidades a las que tiene acceso.
- (RF_13) **Crear un formato de ingreso.** El formato de ingreso contiene datos personales (Nombre, Cédula, Edad, Sexo, Ciudad de origen, fecha, hora de ingreso y firma) y un examen físico estándar (Peso, altura, lesiones, estado de salud, observaciones especiales).
- (RF_14) **Crear un verificador básico de aceptabilidad del refugiado.** El verificador básico debe revisar si la persona a ingresar al refugio ha sido vetada de futuros ingresos por diferentes causas (comportamientos violentos, desacato de reglas, etc.).
- (RF_15) **Incluir un formato de condiciones especiales.** El formato de requerimientos especiales debe permitir anotar si el refugiado tiene condiciones físicas o psiquiátricas, si necesita medicamentos, y una sección libre para anotar información adicional.

- (RF_16) **Crear un registro de pertenencias.** El registro de pertenencias debe poder enlistar cada pertenencia, por tipo (u objeto no especificado), y características distintivas, así como realizar una asignación de lockers.
- (RF_17) **Crear un sistema de egresos de las distintas sedes del refugio.** El sistema de egresos debe aplicar a habitantes debidamente registrados, realizar una verificación básica de seguridad (evitar egresos de personas en condiciones de vulnerabilidad extremas) y anotar distintos motivos de egreso (Voluntario o forzado)
- (RNF_1) **Generar un sistema de organización entre los refugios y sus empleados.** El aplicativo debe permitir la consulta, comunicación, y actualización de información de todos los refugios de manera centralizada e inmediata. Se usa el mismo aplicativo, con las mismas reglas, para todos los refugios, y sus funcionalidades.
- (RNF_5) **Procurar que el desarrollo y mantenimiento de la aplicación se ajuste a un presupuesto reducido.** La planeación del desarrollo y mantenimiento de la aplicación busca optimizar costos y esfuerzo, dando como resultado que sea lo más económica posible.

SHOULD (DEBERÍA - no obligatorio pero aporta mucho a la exp de usuario)

- (RF_1) **Consultar en tiempo real, una base de datos que indique la disponibilidad de las camas en cada refugio.** El sistema debe asegurar que el registro de ingresos y salidas del refugio sea inmediato, y actualizar la base de datos en cada registro. Igualmente, en el proceso de ingreso, debe realizar una consulta a las camas disponibles, así como detectar colisiones (por ejemplo, dos empleados aplican a la misma cama al mismo tiempo).
- (RF_2) **Definir un protocolo estándar de ingreso, gestión, y salida de un habitante de calle en el sistema.** Este protocolo debe estar establecido en la sección de reglamento del módulo de comunicaciones. Debe tener lenguaje preciso, y especificar el manejo de excepciones.
- (RF_4) **Incluir en la base de datos un registro histórico de ingresos y salidas.** Este registro debe incluir información de usuario, empleados responsables, línea de tiempo (fechas de ingreso, salida, novedades), refugio, cama.
- (RF_5) **Implementar un sistema de administración financiera unificado.** El sistema deberá permitir el manejo de nómina, facturas, gastos, ingresos y donaciones de todas las sedes. Cada una de estas categorías deberá contar con un registro que almacena la información relevante de cada área.
- (RF_8) **Crear un aplicativo con una GUI que permita acceder a todas las funcionalidades definidas de la primera aplicación.** La aplicación debe tener las siguientes secciones: Ingresos, Agenda, Gestión de usuarios, Comunicaciones, Donaciones, Consulta histórica, Salidas, Configuración de usuario y Cerrar sesión. También debe asignar los permisos de acceso y modificación de acuerdo al rol del usuario.
- (RF_10) **Incluir en la función de consulta histórica, la habilidad de filtrar los resultados de búsqueda.** El filtro debe ser realizado por: Fecha, refugio, cama, usuario, empleado.
- (RF_16) **Crear un canal de comunicaciones.** El canal de comunicaciones debe ser fácil de acceder en la API. Incluye chats grupales, newsletter (para difusión de notificaciones globales por parte de los administradores), reglamento de la empresa y documentación del aplicativo.

- (RNF_4) **Implementar una primera versión funcional de la aplicación en tres meses.** En menos de tres meses, se debe entregar una aplicación funcional y probada por el equipo de desarrollo. Debe incluir la GUI, los módulos individuales de manera funcional, y conexión a otros elementos como la base de datos, o una API de un sistema de comunicaciones (como WhatsApp o Slack).
- (RNF_6) **Asegurar que el sistema sea fácil e intuitivo de usar.** La interfaz debe tener un contenido legible y no ambiguo. Cada módulo debe indicar las opciones disponibles, y una breve descripción del módulo cuando se hace hover sobre el botón del módulo/función en cuestión.

COULD (PODRÍA - es un plus usualmente personalización, se desarrolla usualmente cuando queda tiempo)

- (RF_6) **Implementar un sistema de pre-registro.** El sistema consiste en un formulario que puede llenar un habitante por fuera del refugio. El formulario recibe los datos personales, y la fecha tentativa de llegada. El módulo de ingresos debe poder generar un registro cuando se llena el pre-registro, y se completa con la información faltante, cuando llega el habitante de calle al refugio, o si no llega.
- (RNF_2) **Aplicabilidad del protocolo teniendo en cuenta las necesidades de cada equipo.** Desarrollar módulos o métodos específicos para un refugio. Esto se define basado en el refugio registrado por el usuario, cambiando la GUI, o una funcionalidad del aplicativo en particular.

WONT (NO NECESARIO - puede esperar al punto que esté o no no pasa nada si nunca se desarrolla)

- (RNF_3) **Funcionalidad en Mobile.** Crear una aplicación móvil comparable a la versión para escritorio.

Tabla general

Must	<ul style="list-style-type: none"> - (RF_3) Crear un sistema de ingresos a las distintas sedes del refugio. - (RF_7) Construir un sistema de agenda de los turnos del personal. - (RF_9) Implementar un sistema único para todos los centros. - (RF_11) Definir e implementar un sistema de roles para los usuarios del sistema. - (RF_12) Autenticación del usuario dentro del sistema de acuerdo a su rol. - (RF_13) Crear un formato de ingreso. - (RF_14) Crear un verificador básico de aceptabilidad del refugiado. - (RF_15) Incluir un formato de condiciones especiales. - (RF_16) Crear un registro de pertenencias. - (RF_17) Crear un sistema de egresos de las distintas sedes del refugio.
------	--

	<ul style="list-style-type: none"> - (RNF_1) Generar un sistema de organización entre los refugios y sus empleados. - (RNF_5) Procurar que el desarrollo y mantenimiento de la aplicación se ajuste a un presupuesto reducido.
Should	<ul style="list-style-type: none"> - (RF_1) Consultar en tiempo real, una base de datos que indique la disponibilidad de las camas en cada refugio. - (RF_2) Definir un protocolo estándar de ingreso, gestión, y salida de un habitante de calle en el sistema. - (RF_4) Incluir en la base de datos un registro histórico de ingresos y salidas. - (RF_5) Implementar un sistema de administración financiera unificado. - (RF_8) Crear un aplicativo con una GUI que permita acceder a todas las funcionalidades definidas de la primera aplicación. - (RF_10) Incluir en la función de consulta histórica, la habilidad de filtrar los resultados de búsqueda. - (RNF_4) Implementar una primera versión funcional de la aplicación en tres meses. - (RNF_6) Asegurar que la aplicación sea fácil e intuitiva de usar.
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 Mobile

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

Requisito		Estimación	Argumento
RF_3	Crear un sistema de ingresos a las distintas sedes del refugio.	5	Dado que este requisito depende de los requisitos funcionales (RF) 13, 14, 15, 16 y 17. El esfuerzo adicional para la finalización de este requisito radica en el acoplamiento de dichos requisitos.
MUST			
RF_7	Construir un sistema de agenda de los turnos del personal.	5	Implica trabajo ya que significa llevar a cabo un estudio en profundidad de los horarios del personal. También toca implementar una aplicación de calendario
MUST			

RF_9	Implementar un sistema único para todos los centros.	13	Requiere integrar prácticamente todos los otros requisitos funcionales.
MUST			
RF_11	Definir e implementar un sistema de roles para los usuarios del sistema.	2	Puede ser implementado de manera rápida en el backend de la aplicación.
MUST			
RF_12	Autenticación del usuario dentro del sistema de acuerdo a su rol.	5	Se puede lograr implementando sistemas de autenticación ya existentes, pero las pruebas tienen que ser estrictas
MUST			
RF_13	Crear un formato de ingreso.	1	El formato de ingreso se puede desarrollar rápidamente tanto de manera física como dentro de la aplicación. En cuanto a la integración con la aplicación sólo necesita retoques en el frontend.
MUST			
RF_14	Crear un verificador básico de aceptabilidad del refugiado.	1	Sólo requiere realizar consultas a la base de datos, y verificar el formulario.
MUST			
RF_15	Incluir un formato de condiciones especiales.	1	Involucra pequeñas modificaciones en el backend. Es necesario aclarar con el cliente cuáles son los requerimientos especiales más frecuentes
MUST			
RF_16	Crear un registro de pertenencias.	2	Requiere su propia tabla de base de datos. Además, su implementación es muy similar a la de otros módulos
MUST			
RF_17	Crear un sistema de egresos de las distintas sedes del refugio.	2	Si bien puede requerir su propio módulo, es más simple que el módulo de ingreso en términos de lógica de negocio requerida.
MUST			
RNF_1	Generar un sistema de organización entre los refugios y sus empleados.	1	Este requerimiento por sí solo involucra poco trabajo adicional ya que se entiende como un acople de los requerimientos funcionales (RF), por lo que una vez resueltos estos, este se soluciona por sí solo.
: MUST			
RNF_5	Procurar que el desarrollo y mantenimiento de la aplicación se ajuste a	13	Se necesita amplia experiencia en Ingeniería de Software, para hacer una buena estimación de tiempos, recursos necesarios en condiciones ideales, y estrategias para ahorrar recursos
MUST			

	un presupuesto reducido.		
RF_1 SHOULD	Consultar en tiempo real, una base de datos que indique la disponibilidad de las camas en cada refugio.	5	La actualización en tiempo real presenta desafíos adicionales de hardware y diseño de la aplicación.
RF_2 SHOULD	Definir un protocolo estándar de ingreso, gestión, y salida de un habitante de calle en el sistema.	1	Se puede diseñar con el cliente, tras esclarecer el flujo de trabajo deseado.
RF_4 SHOULD	Incluir en la base de datos un registro histórico de ingresos y salidas.	5	La implementación de un historial requiere lógica sencilla de registros, sin embargo la consulta y transcripción de los datos manuales y electrónicos no integrados exige tiempo.
RF_5 SHOULD	Implementar un sistema de administración financiera unificado.	13	La implementación de un sistema que permita la administración, manejo y registro del flujo de caja en general requiere migrar de un sistema físico a uno digital, el cuál implica trasladar grandes cantidades de información. Adicionalmente, requiere abstraer toda la lógica de negocio hacia la aplicación.
RF_8 SHOULD	Crear un aplicativo con una GUI que permita acceder a todas las funcionalidades definidas de la primera aplicación.	5	Una vez desarrollada toda la lógica de negocio en el backend de la aplicación, basta con consumir los servicios en una GUI. Sin embargo, el aplicativo requiere tareas de diseño y maquetación de las interfaces.
RF_10 SHOULD	Incluir en la función de consulta histórica, la habilidad de filtrar los resultados de búsqueda.	3	Para consultas específicas se pueden utilizar queries, estos son fáciles de emplear ya que representan una solicitud precisa para extraer los datos.
RNF_4 SHOULD	Implementar una primera versión funcional de la aplicación en tres meses.	13	Se deben considerar las capacidades del equipo de trabajo para la entrega de una versión inicial en el tiempo de entrega estipulado.
RNF_6 SHOULD	Asegurar que la aplicación sea fácil e intuitivo de usar	8	Depende en buena medida del RF_8. Pero una vez desarrollada la GUI se pueden hacer algunos ajustes de accesibilidad. Adicionalmente, requiere de la retroalimentación del cliente.

RF_6	Implementar un sistema de pre-registro.	2	Se puede desarrollar de manera muy similar al RF_3. Sin embargo, como requiere de toda la lógica de negocio previamente desarrollada, se puede realizar con algunos ajustes a la aplicación.
COULD			
RNF_2	Aplicabilidad del protocolo teniendo en cuenta las necesidades de cada equipo	13	Implica estudiar todo el hardware de cada refugio y replantear la funcionalidad en base a los recursos.
COULD			
RNF_3	Funcionalidad en Mobile	34	Requiere replantear toda la funcionalidad de la aplicación, así como crear su propio diseño, frontend, etc.
WONT			