



UNIVERSIDAD  
**NACIONAL**  
DE COLOMBIA

**Universidad Nacional de Colombia - sede Bogotá**  
**Facultad de Ingeniería**  
**Departamento de Ingeniería de Sistemas**  
**Curso: Ingeniería de Software**

**Estudiantes: Grupo 1: Formula 1 - (2025-1) Martes y Jueves.**

Requerimientos funcionales y no funcionales

- Crear o Actualizar registro de personas
- Ver disponibilidad de camas
- Administración de recursos entre centros
- Asignación de camas
- Ver historial de estadía
- Administración de turnos de personal (Crear, Modificar, Eliminar)
- Administración de turnos de centro (Registro de novedades)
- Registro y seguimiento de donaciones
- Panel general
- Estandarizar reglas
- Aplicación disponible en Móvil y PC
- Actualización en tiempo real de la información de los centros

FUNCIONALES	NO FUNCIONALES
RF1.1: Crear registro	RNF1: Aplicación disponible en Móvil y PC
RF1.2: Actualizar registro de personas	RNF2.1: Actualización bajo demanda de la información de los centros
RF2: Ver disponibilidad de camas	RNF2.2: Actualización en tiempo real de la información de los centros
RF3: Administración de recursos entre centros	RNF3: Privacidad de la información de los usuarios
RF4: Asignación de camas	
RF5: Ver historial de estadía	
RF6: Administración de turnos de personal (Crear, Modificar, Eliminar)	

RF7: Administración de turnos de centro (Registro de novedades)	
RF8: Registro y seguimiento de donaciones	
RF9: Panel general para usuarios administrativos	
RF10: Estandarizar reglas	

## 2. MoSCoW

### ❖ Must have

- RNF1: La aplicación que pide el cliente debe estar disponible en Android y en PC
- RNF2.1: La información de cada centro debe poder ser editada y visualizada, bajo demanda.
- RF1.1: Creación de registro: Un usuario debe poder crear un registro de una persona que ingresa a un centro por primera vez. Este registro contiene información como nombre, edad, género, estado de salud, después de esto el sistema debe redirigir al usuario para registrar una visita en el centro.
- RF1.2: Actualizar registro: Agregar una visita al centro que incluya actualización de los datos (si los hay), registrar pertenencias ingresadas, asignación de cama.
- RF2: Ver disponibilidad de camas para realizar la asignación inmediata de un centro en específico
- RF3: El sistema debe tener una opción para visualizar los recursos de otros centros como disponibilidad de camas, cantidad de alimento, disponibilidad de otros recursos (kit médico, kit de aseo, etc) y debe ser posible realizar una solicitud de uso de un recurso de otro centro, la cual puede ser aceptada o rechazada por el centro que recibe la solicitud
- RF4: Un usuario de un centro debe poder asignar 1 cama a un registro de visita de una persona de acuerdo a la disponibilidad del centro
- RF9: Panel general para usuarios administrativos en donde se ve toda la información relevante de los centros (recursos, turnos, donaciones)

### ❖ Should have

- RF5: Ver historial de estadía: Si bien es importante, es más prioritario establecer nuevos registros con base al software que queremos implementar, aquí almacenaremos el tiempo de estadía de cada paciente registrado.
- RF6: Se modificarán y mostrarán los turnos laborales de los funcionarios que participan en la organización.
- RF7: Administración de turnos de centro, y registrar novedades si las hay (cantidad de personas atendidas, conflictos entre visitantes, agotamiento de recursos, crear acta del turno, etc)
- RF8.1: Se debe de poder visualizar y tener un registro de las donaciones que entran cada cierto tiempo, así como el centro donde ingresaron
- RNF2.2: La información de cada centro debe poder ser editada y visualizada, en tiempo real.

### ❖ Could have

- RF8.2: Se debería tener un sistema de recomendación para dirigir donaciones a los centros que la necesitan según el registro de recursos.

### ❖ Won't

- RNF3: Aunque no es probable que una base de datos de un centro como estos sea atacada, a corto plazo, un sistema de seguridad podría ser útil

para proteger los datos de las personas que asisten a los centros. Se puede implementar sistemas de protección de datos.

- RF10: La estandarización de reglas es responsabilidad de los administrativos del centro.

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

Poker Planning:

4 developers

Requisito	Priorización	Estimación	Argumento
RF1.1	MUST	5	Tiene una dificultad media alta, pues se necesitan varias rondas de testing para ver si los datos se almacenan correctamente
RF1.2	MUST	5	Es una tarea medianamente compleja dado que requiere una conexión con los registros actuales y generar visitas que puedan ser fáciles de rastrear
RF2	MUST	2	Aunque es una funcionalidad que solo requiere mostrar información, para esto es necesario llamar la base de datos y que estos estén actualizados
RF3	MUST	8	Es una tarea bastante compleja pues requiere administrar la mayoría de datos que se almacenan y mostrarlos de una manera organizada. Además, requiere una forma de comunicación entre centros para enviar una solicitud de uso de un recurso, como la respuesta a la misma.
RF4	MUST	3	Tiene una dificultad media pues deben ser actualizados dos registros: el de la persona que se registra y el de disponibilidad de camas.
RF5	MUST	8	Esta tarea requiere la creación de permisos para usuarios de manera que, únicamente los usuarios administrativos puedan acceder a este panel. Además, se requiere la visualización de todos los datos almacenados de un centro, lo que implica llamados a la base de datos.
RF5	SHOULD	2	Es una tarea que sólo requiere llamado a la base de datos

RNF1	MUST	5	Tiene dificultad media alta debido a que el cambio del tipo de dispositivo puede requerir nuevamente el desarrollo de la aplicación, la dificultad depende del entorno de desarrollo.
RNF2.1	MUST	5	La dificultad media alta se debe a que toca manejar versiones de los registros adecuadamente para que no existan conflictos al actualizar
RF6	SHOULD	3	Esta tarea tiene dificultad media pues únicamente los usuarios con permiso administrativo pueden crear y modificar turnos, mientras que cualquier usuario puede ver los turnos.
RF7	SHOULD	5	Si bien es una tarea que tiene una dificultad y estructura parecida a otras en las que pusimos 8, aquí ya hemos trabajado en resolver las mayores dificultades de esta estructura de funcionamiento.
RF8.1	SHOULD	5	Teniendo en cuenta las implementaciones previas, es un requisito que se puede complementar por extensión.
RNF2.2	SHOULD	5	Esta tarea requiere un paso extra al requisito RNF 2.1 pues se necesitaría una conexión a un servicio externo que nos permita hacer consultas y actualizaciones en tiempo real sin comprometer el rendimiento del sistema
RF8.2	COULD	5	La dificultad recae es crear un algoritmo de recomendación que tome los datos que estamos almacenando y de una sugerencia de centro de destino de la donación, para de esta manera equilibrar la cantidad de recursos en todos los centros
RNF3	WON'T	5	En una etapa temprana es complejo estimar en qué consistiría la privacidad y protección de datos, pero puede implicar modificaciones en la base de datos (qué se almacena y cómo se almacena)
RF10	WON'T	1	Aunque esta tarea no nos corresponde como desarrolladores, su implementación es de dificultad baja ya que sería agregar condiciones a las funcionalidades actuales