**URL de presentación de proyecto:**

<https://youtu.be/-UMXbKC6Y0o>

**Miembros del equipo:**

* Alexis Ake Vela
* Diego DeGante
* Mario Chan Zurita
* Pedro Cauich Salazar

**Tabla de contenido**

1. **Introducción**
2. **Definición de proyecto**
   * 2.1 Problemática
   * 2.2 Objetivos
3. **Requerimientos**
   * 3.1 Actores del sistema
   * 3.2 Requerimientos de sistema
     + 3.2.1 Funcionales
     + 3.2.2 No funcionales
     + 3.2.3 Mapeo de requerimientos
4. **Casos de uso** 
   * 4.1 Diagrama de casos de uso
5. **Proceso** 
   * 5.1 Estándares de codificación
   * 5.2 Proceso de desarrollo
6. **Trabajo en equipo** 
   * 6.1 Roles
   * 6.2 Esquema de monitoreo
   * 6.3 Bitácoras
   * 6.4 Métrica para evaluar avance individual
   * 6.5 Avance grupal

**Sistema de alarmas relacionadas al recordatorio de recetas de medicamentos.**

**1. Introducción.**

MedicineReminer es una aplicación en la cual puedes guardar y administrar todas tus recetas médicas, permitiéndote llevar el control de tus medicamentos. Todo esto simplemente añadiendo el nombre del medicamento, la hora de la primera toma, los periodos y notas con las especificaciones médicas necesarias.

MedicineReminder llevará el control de los tiempos de ingesta de cada medicamento por ti, al igual que te mostrará todas las especificaciones necesarias al momento de tomar cada medicamento para que nunca olvides ningún detalle.

**2. Definición del proyecto.**

**2.1 Problemática:**

Al crearse el concepto de medicamentos, surge la necesidad de un consumo de estos moderado y periódico. Los medicamentos tienen el objetivo de mantener un estilo de vida a las personas, sin embargo, muchas de estas personas no son aptas para medir sus tiempos y mantener una responsabilidad óptima sobre sí mismas, es por eso que nuestro proyecto presenta una alternativa para cubrir esta necesidad de una manera eficiente.

**2.2 Objetivos:**

**Objetivo general:**

Presentar a los usuarios una forma fácil y rápida de organizar el uso de medicamentos.

**Objetivos específicos:**

A) Estructurar la información de los medicamentos de una manera eficiente y de fácil entendimiento.

B) Organizar los medicamentos en cuestión a si se encuentran activos o inactivos.

C) Notificar e indicar las especificaciones de ingesta para cada toma de cada medicamento.

D) Administrar las alarmas de acuerdo a las necesidades del usuario.

E) Automatizar el funcionamiento de las alarmas.

F) Compatible con más del 80% de dispositivos android.

G) Permitir al usuario la automatización de los procesos mediante la voz.

**3. Requerimientos**

**3.1 Actores del sistema:**

**Usuario:** persona que usa el sistema

Puede ingresar una nueva alarma de medicamento a su lista de alarmas a través de la opción de agregar alarma.

Puede editar una alarma ya creada previamente.

Puede eliminar una alarma de su lista de alarmas.

Puede agregar o eliminar una alarma haciendo uso únicamente del asistente inteligente.

Puede hacer una búsqueda en su lista de medicamentos ingresando el nombre del medicamento.

**3.2 Requerimientos de sistema**

**3.2.1 Funcionales**

|  |  |
| --- | --- |
| RF001 | Página principal |
| Descripción | Al iniciar la aplicación, el sistema mostrará un listado únicamente de las alarmas ya creadas por el usuario que se encuentran activas. El sistema dará la opción de administrar las alarmas. Las opciones que dará el sistema son las siguientes:   * Seleccionar una alarma de la lista * Activar asistente inteligente * Buscar alarma * Agregar alarma * Mostrar alarmas |

|  |  |
| --- | --- |
| RF002 | Seleccionar una alarma de la lista |
| Descripción | El sistema abrirá la alarma, mostrando todos los datos ingresados por el usuario. Dará la opción de editar la alarma o eliminarla. |

|  |  |
| --- | --- |
| RF003 | Eliminar alarma |
| Descripción | El sistema permitirá eliminar la alarma completamente. Teniendo en cuenta que, al eliminarla, los datos de la alarma ya no se podrán visualizar ni ser recuperados. |

|  |  |
| --- | --- |
| RF004 | Editar |
| Descripción | El sistema permitirá modificar cualquiera de los datos ya ingresados en alarmas ya previamente creadas por el mismo usuario. Teniendo en cuenta que, al modificarlos, se reiniciará el tiempo de duración que tenía el medicamente o hábito. |

|  |  |
| --- | --- |
| RF005 | Agregar |
| Descripción | El sistema permitirá agregar una alarma. Los datos que se deben de ingresar para agregar una alarma son:   * Nombre del medicamento * Hora de primera toma * Periodo de tiempo (cada cuanto tiempo se debe de tomar el medicamento) * Fecha en la cual se auto desactivará la alarma * Notas (recordatorios de especificaciones) |

|  |  |
| --- | --- |
| RF006 | Buscador de alarma |
| Descripción | En la parte superior se encontrará una barra de búsqueda el cual desplegará las opciones alarmas existentes relacionadas con la búsqueda, las opciones serán seleccionables y redirigirán al usuario a editar la alarma |

|  |  |
| --- | --- |
| RF007 | Mostrar alarmas |
| Descripción | El sistema mostrará una lista de todas las alarmas creadas por el usuario, tanto las que se encuentran activadas como las que se encuentran desactivadas. |

**3.2.2 No funcionales**

|  |  |
| --- | --- |
| RNF001 | Conteo automático |
| Descripción | El sistema empezará su conteo de duración de acuerdo al tiempo ingresado por el usuario una vez guardado el recordatorio, con los usos que sean posibles durante ese tiempo. |

|  |  |
| --- | --- |
| RNF002 | Suspensión automática |
| Descripción | El sistema desactivará automáticamente los recordatorias una vez que se llegue a la fecha indicada como fecha de auto desactivación. |

|  |  |
| --- | --- |
| RNF003 | Notificaciones |
| Descripción | El sistema mandará la notificación de uso del medicamente cada que cumple el periodo de ingresado (hr) con los siguientes datos:   * Nombre del medicamento * Notas agregadas por el usuario |

|  |  |
| --- | --- |
| RNF004 | Asistente Inteligente |
| Descripción | Se podrá acceder a la aplicación mediante el asistente inteligente d Google, el usuario podrá únicamente agregar o eliminar alarmas haciendo uso del asistente |

|  |  |
| --- | --- |
| RNF005 | Sistemas compatibles |
| Descripción | La aplicación se desarrollaá con android 4.0.3(Ice Cream Sandwich) API 15 para tener una compatibilidad del 100% con cualquier dispositivo. |

**3.2.3 Mapeo de requerimientos**

|  |  |
| --- | --- |
| **Requerimiento** | **Objetivo** |
| Funcionales |  |
| RF001 | A |
| RF002 | A |
| RF003 | C |
| RF004 | D |
| RF005 | D |
| RF006 | A |
| RF007 | B |
| No Funcionales |  |
| RNF001 | E |
| RNF002 | E |
| RNF003 | C |
| RNF004 | G |
| RNF005 | F |

**4. Casos de uso.**

**CU01**: Agregar alarma de medicina.  
**Descripción**: El usuario desea agregar la alarma de una medicina.  
**Secuencia**:

1. El usuario abre la aplicación.
2. El usuario presiona el botón de añadir alarma.
3. El usuario coloca todos los datos requeridos.
4. La alarma comienza a funcionar al momento que el usuario presiona guardar.
5. El usuario puede agregar otra alarma o regresar a la página principal presionando la flecha para regresar.

**Salidas alternativas:**

Si el usuario ya no desea crear alguna alarma, presiona la flecha para regresar a la página principal.

**CU02**: Editar alarma de medicina.  
**Descripción**: El usuario desea editar la alarma de una medicina.  
**Secuencia**:

1. El usuario abre la aplicación.
2. El usuario selecciona la opción buscar.
3. El usuario coloca en la barra de búsqueda el nombre de la alarma que desea editar.
4. El usuario selecciona el nombre de la alarma.
5. El usuario modifica el o los datos que desea cambiar.
6. El usuario presiona el botón de guardar.
7. El usuario presiona la flecha para regresar a la página principal.

**Salidas alternativas:**

Si el usuario ya no desea editar alguna alarma, presiona la flecha para regresar a la página principal.

**CU03**: Eliminar alarma de medicina.  
**Descripción**: El usuario desea eliminar la alarma de una medicina.  
**Secuencia**:

1. El usuario abre la aplicación.
2. El usuario selecciona la opción buscar.
3. El usuario coloca en la barra de búsqueda el nombre de la alarma que desea eliminar.
4. El usuario selecciona el nombre de la alarma.
5. El usuario presiona el botón de eliminar.
6. El usuario presiona la flecha para regresar a la página principal.

**Salidas alternativas:**

Si el usuario ya no desea eliminar alguna alarma, presiona la flecha para regresar a la página principal.

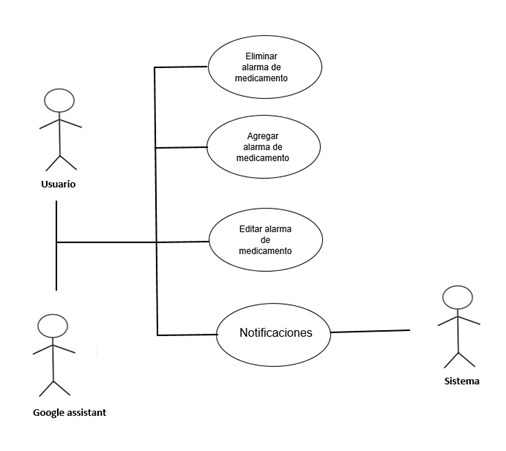
**CU04**: Agregar alarma de medicina usando el asistente inteligente.  
**Descripción**: El usuario desea agregar la alarma de una medicina usando únicamente el asistente inteligente.  
**Secuencia**:

1. El usuario abre la aplicación.
2. El usuario selecciona la opción asistente.
3. El usuario, como ejemplo, le puede decir al asistente: “Agregar alarma”.
4. Se hará una serie de preguntas y respuestas entre el usuario y el asistente para adquirir todos los datos necesarios.
5. Al finalizar, el asistente dirá: “Alarma agregada”, se cerrará automáticamente y la alarma estará activa a partir de ese momento.
6. El usuario se mantendrá en la página principal durante todo el proceso.

**Salidas alternativas:**

Si el usuario ya no desea agregar alguna alarma, puede decir: “cancelar”, cancelando automáticamente el proceso.

**4.1 Diagrama de casos:**



**5. Proceso.**

**5.1 Estándares de codificación**

Se empleará el uso del estándar básico de programación:

Los nombres de las variables serán sustantivos que hagan referencia al contexto de la variable, la primera letra de la primera palabra será minúscula y el inicio de cada palabra, mayúscula, no se empleará el uso de guion o guion bajo entre palabras, las palabras estarán unidas una tras otra.

Uso correcto de la indentación, se empleará el uso de llaves, en donde después de abrir una llave se creará un salto de línea y las líneas de código se desplazarán un TAB por cada llave abierta, es decir, si se abren dos llaves, la línea de código debe moverse dos TAB, el cierre de llave va después de la última línea de código.

Funciones, las funciones usarán el mismo tipo de nombrado que las variables excepto que en vez de sustantivos serán verbos que dan contexto a la acción de la función.

Archivos, los archivos creados que sean requeridos durante el proceso de desarrollo, se nombraran de forma que hagan referencia al contexto y/o contenido del mismo

Librerías, las librerías a emplear serán las proporcionadas por el propio lenguaje y las creadas por el mismo equipo de desarrollo

Comentarios, para las funciones se emplearán principalmente dos tipos de comentarios, el generado por la extensión Javadoc Tools de la aplicación Visual Studio Code, y comentarios sencillos, preferentemente agregados dentro del comentario de Javadoc, en donde se explica brevemente la función

Como documentación se utilizará JavaDoc proveniente del Java Developer Kit

**5.2 Proceso de desarrollo.**

Se empleará el método scrum, pero enfocándonos en que el sistema sea iterativo-incremental en donde se llevará la necesidad de tener un análisis, un diseño, la codificación y las pruebas (en ese mismo orden), para cada que se le agregue una funcionalidad conforme pase el tiempo para permitir la evolución del producto.

Herramientas:

Como principal entorno de desarrollo se empleará Android Studio, como principal punto de trabajo entre los miembros del equipo se utilizará GitHub, se usará Visual Studio Code como apoyo para la generación de la documentación interna del codigo.

**6. Trabajo en equipo**

**6.1 Roles:**

Alexis Ake: *Scrum team*

Diego de Gante: *Scrum team*

Mario Chan: *Scrum team* ***/ Scrum master***

Pedro Cauich: *Scrum team*

**6.2 Esquemas de monitoreo:**

El monitoreo del equipo estará contenido en el calendario, donde se irán apuntando los avances logrados cada determinado tiempo y serán comparados con los avances planeados.

**6.3 Bitácoras:**

Conforme avanzamos en el proceso de desarrollo, iremos anotando cada una de los avances y cambios que se vayan haciendo al sistema, para tener un control sobre el proceso.

<https://1drv.ms/w/s!AnplW1evppzUh2FakbGST47N7krN?e=BUNP7L>

**6.4 Métrica para evaluar avance individual:**

Al terminar con cada sprint, en la reunión de revisión, se realizarán pruebas unitarias a cada módulo trabajado, se analizará entonces la eficiencia del código, así como su extensión. También añadiremos la revisión del tiempo dedicado de cada integrante del equipo como un recurso adicional para la medición.

El porcentaje será dado se acuerdo a las pruebas unitarias en el “retrospective sprint”, por lo que cada junta deberán tener un porcentaje de 100% para que al final se saque un promedio entre los 6 sprint para obtener su % de contribución.

*El desglose de la métrica individual esta agregada junto con la bitácora.*

retrospective sprint\_ finalpercentage:

Alexis Ake: 100%

Diego de Gante: 100%

Mario Chan: 100%

Pedro Cauich: 100%

**6.5 Avance grupal:**

Haremos reuniones donde se analizará el avance del proyecto en relación a lo establecido, para poder dar paso a la siguiente etapa de desarrollo, tomando en cuenta la asistencia y su avance en la construcción del proyecto para saber que cada uno cumpla con su parte establecida.