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

**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.

**Objetivos:**

**Objetivo general:**

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

**Objetivos específicos:**

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

**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 activar o desactivar alarmas de su lista de alarmas.

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

**Requerimientos de sistema**

**Funcionales**

|  |  |
| --- | --- |
| RF001 | Página principal |
| Descripción | Al iniciar la aplicación, el sistema mostrará un listado de todas las alarmas ya creadas por el usuario. El sistema dará la opción de administrar las alarmas. Las opciones que dará el sistema son las siguientes:  Una lista de las alarmas ya creadas  Editar  Agregar |

|  |  |
| --- | --- |
| RF002 | 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. |

|  |  |
| --- | --- |
| RF003 | Agregar |
| Descripción | El sistema permitirá agregar una alarma. Los datos que se deben de ingresar para agregar una alarma son:  Nombre del medicamento  Dosis  Periodo de tiempo (cada cuanto tiempo se debe de tomar el medicamento)  Duración (durante cuantos días estará activa la alarma)  Notas (recordatorios de especificaciones) |

|  |  |
| --- | --- |
| RF004 | Buscador de alarma |
| Descripción | En la página principal del sistema, 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 |

**No funcionales**

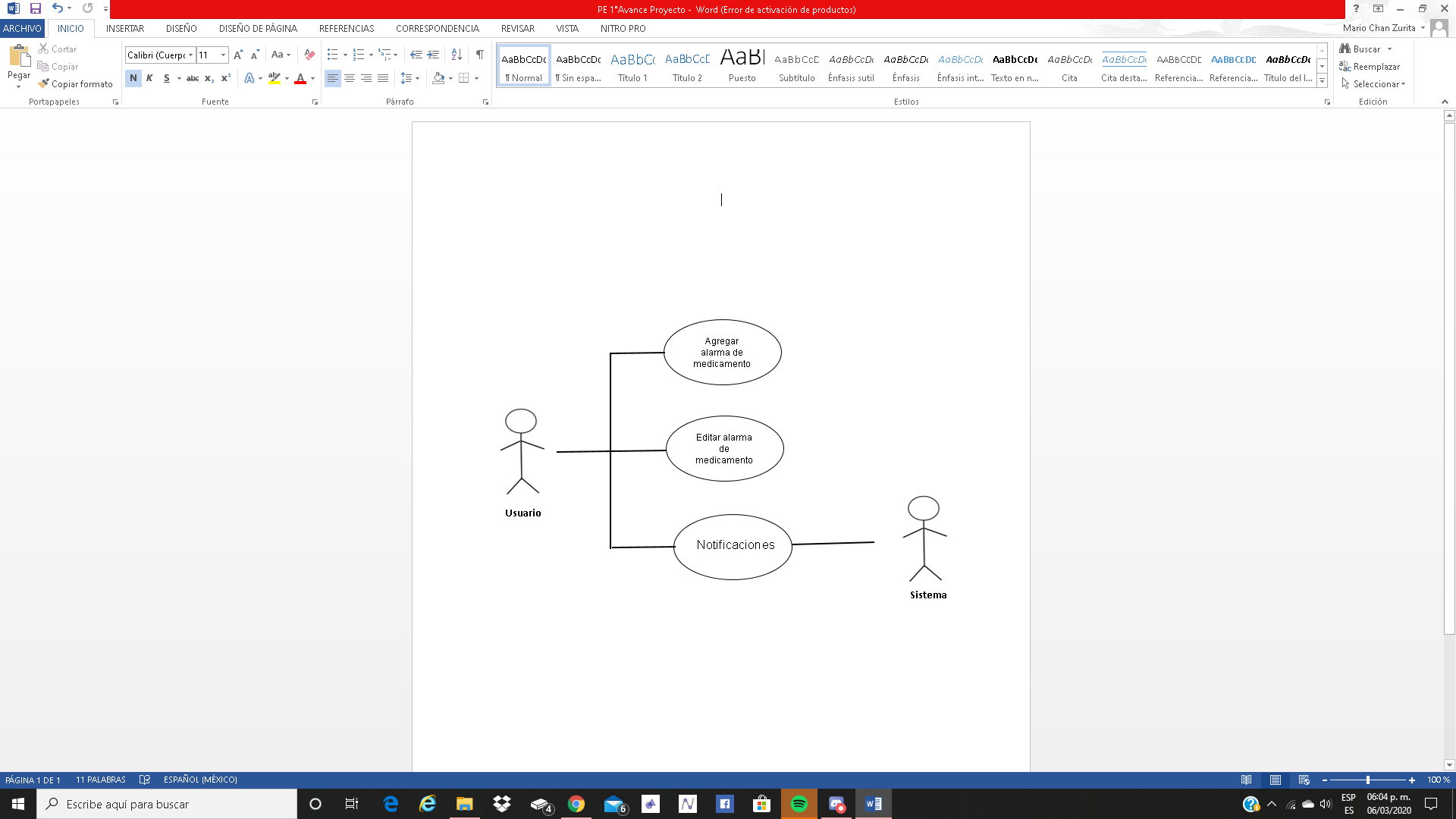
|  |  |
| --- | --- |
| RNF001 | Opciones de alarma |
| Descripción | Al comenzar a sonar una alarma, el sistema dará las siguientes opciones:  Posponer alarma (Pospone la alarma 5 minutos)  Apagar alarma (inicia el siguiente periodo de tiempo durante el tiempo establecido) |

|  |  |
| --- | --- |
| RNF002 | 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. |

|  |  |
| --- | --- |
| RNF003 | Suspensión automática |
| Descripción | El sistema anulará automáticamente los recordatorias una vez que se cumpla la duración del medicamente ingresado por el usuario (días). |

|  |  |
| --- | --- |
| RNF004 | Notificaciones |
| Descripción | El sistema mandará la notificación de uso del medicamente cada que cumple el periodo de ingresado (hr) con las notas del medicamento ya establecido, sin modificarse en caso de que se posponga la alarma. |

**Diagrama de casos:**



**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

**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.

Organización:

Alexis Ake: *Scrum team*

Diego de Gante: *Scrum team*

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

Pedro Cauich: *Scrum team*

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.

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.

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 4 sprint para obtener su % de contribución.

Ejemplo:

retrospective sprint\_1:

Alexis Ake: 100%

Diego de Gante: 80%

Mario Chan: 100%

Pedro Cauich: 80%

**.**

**.**

**.**

retrospective sprint\_ finalpercentage:

Alexis Ake: 97.5%

Diego de Gante: 99%

Mario Chan: 100%

Pedro Cauich: 96.5%

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.