

## **Documento SRS**

### **1. Introducción**

#### **1.1. Propósito**

#### **1.2. Alcance**

#### **1.3. Visualización general del proyecto**

#### **1.4. Definiciones**

### **2. Referencias**

### **3. Requisitos**

#### **3.1. Especificación de requisitos**

#### **3.2. Método de priorización de requisitos**

#### **3.3. Métodos de educación de requisitos**

#### **3.4. Descripción de requisitos**

### **4. Validaciones**

### **5. Anexos**

## **1. Introducción**

### **1.1. Propósito:**

En este documento presentaremos los requisitos para el desarrollo del sistema Web para la Gestión de Accesos y Aforo por COVID-19 que fueron obtenidos por medio de encuesta y reuniones realizada con los stakeholders, estas actividades se hicieron para poder entender las necesidades y expectativas que se tenían para el proyecto.

Este documento va dirigido tanto a los desarrolladores como a los clientes que servirá como un puente para ambas partes, por ello se utilizará el lenguaje natural en la descripción de los requisitos, también cabe aclarar que es un proyecto de tamaño mediano y de nivel universitario.

Este documento hace parte de los entregables del proyecto para el curso procesos de ingeniería de software del semestre 2020-II realizado por los estudiantes Juan Camilo Gomez, Diego Andres Vinasco, Juan David Arce.

### **1.2. Alcance:**

El alcance de este documento está en definir los requisitos y restricciones del proyecto que fueron acordados por ambas partes (desarrolladores y clientes), después de validar los requisitos no se permitirá cambios de estos y también sirve para que el cliente sepa si se cumplió todo lo especificado en un inicio con la entrega final del sistema.

El alcance del proyecto se trata de crear un sistema en el cual los locales puedan registrar la temperatura, si lleva el tapabocas, fecha y hora de los usuarios que entran. también se espera que las entidades de salud colaboren para brindar información de pacientes que se encuentren en el periodo de cuarentena para que los distintos locales eviten el ingreso de estos usuarios. Los usuarios podrán ver los distintos locales que han visitado y si se encuentran en periodo de cuarentena o no.

Se espera que los administradores puedan ver todos los locales, entidades de salud y usuarios, esto con el fin de regular las distintas cuentas.

### **1.3. Visualización general del documento:**

#### **Ver anexo 1 (Diagrama de contexto)**

En este documento se espera que el cliente pueda entender su contenido ya que se dan a conocer los requisitos y restricciones del sistema que fueron obtenidos por las actividades previamente mencionadas, los requisitos se explicarán más adelante en la sección 3 de este documento, aparte de dar a entender las necesidades del cliente, también se darán referencias de información adicional que se investigó en el proceso del desarrollo del sistema Web para la Gestión de Accesos y Aforo por COVID-19 en la sección 2 de este documento, también se anexaron las verificaciones que se harán al sistema para comprobar su correcta funcionalidad en la sección 4 y por

último tendremos anexos del documento como el diagrama de caso de usos, el diagrama de contexto y el diagrama de paquetes en la sección 5.

#### 1.4. Definiciones:

En esta sección detallaremos y explicaremos los términos técnicos usados en este documento para que el lector no tenga problemas en entender los requisitos.

**Apartado:** con esto nos referimos a una sección o pantalla del aplicativo web que pueda acceder a alguno de los roles en el sistema.

**Concurrente:** con esto nos referimos a la cantidad de usuarios que pueden usar el sistema al mismo tiempo

**Rol:** con esto nos referimos a las diferentes funciones que puede tener uno de los usuarios del sistema, por ejemplo, el rol entidad de salud que se encarga de informar al sistema de pacientes infectados o en espera de resultados o el rol administrador que se encarga de supervisar el sistema, locales, usuarios y entidades de salud.

**Tipos de datos:** en programación, los tipos de datos se refieren a datos numérico y su tipo numérico, si es entero, flotante, etc, también a los caracteres o cadena de caracteres, booleanos, etc

## 2. Referencias:

En esta sección daremos las referencias de páginas que se utilizaron para poder completar requisitos para el sistema web de gestión de acceso y aforo. La primera es una página que ayudó a sacar requisitos no funcionales, las siguientes páginas nos ayudarán a tener una base sobre los servicios que prestan este tipo de servicios web, además de sacar algunos requisitos funcionales y no funcionales.

[1]"Requerimientos no funcionales: Ejemplos", Pmoinformatica.com, 2020. [Online]. Available: <http://www.pmoinformatica.com/2015/05/requerimientos-no-funcionales-ejemplos.html>. [Accessed: 11- Nov- 2020].

[2]"Sistemas de Control de accesos y Control de aforo | AforoGes", *AforoGes*, 2020. [Online]. Available: <https://www.aforoges.es/>. [Accessed: 11- Nov- 2020].

[3]"Control de accesos y aforo - Shake Your Apps", *Shake Your Apps*, 2020. [Online]. Available: <https://shakeyourapps.com/servicios/sistema-de-gestion-de-accesos-para-eventos/>. [Accessed: 11- Nov- 2020].

[4]"Control de aforo: software para supermercados y retail. ¡Pregúntanos!", *Sonotrack*, 2020. [Online]. Available: <https://sonotrack.com/blog/sistema-control-de-aforo-personas/>. [Accessed: 11- Nov- 2020].

### **3. Requisitos**

#### **3.1. Especificación de requisitos:**

En esta sección mostraremos los requisitos en lenguaje natural para que el cliente pueda identificar de manera clara lo que significa cada uno de los requerimientos del sistema recolectados de las diferentes encuestas y reuniones, estos están divididos en apartados pertenecientes a los roles dentro del sistema o su categoría, en este documento incluimos tanto los requisitos funcionales como los no funcionales, las prioridades, ID's o quien se encarga de cada requisito es detallado en [5.3], la forma de describir los requisitos será [sujeto] [acción].

#### **3.2. Método de priorización de requisitos:**

Para priorizar los requisitos del sistema utilizamos la matriz de Wieggers donde se utilizan 4 criterios para evaluar su prioridad los cuales son beneficio, penalización, costo y riesgo donde cada uno va a tener 2, 1, 1, 0.5 puntos de costo respectivamente y cada requisito se valorará de 1 a 10 en cada uno de los criterios, estos valores serán analizados y discutidos hasta llegar a un acuerdo con el equipo desarrollador ya que estos se van a encargar de realizar las funciones o métodos para los algoritmos del sistema, no se incluye la opinión de los diferentes stakeholders puesto que puede ser difícil que se pongan de acuerdo con un valor o incluso que sientan que todos son prioritarios y con una puntuación muy alta.

#### **3.3. Métodos de educación de requisitos:**

En esta sección se describirán las distintas actividades que se utilizaron para poder recolectar los diferentes requisitos del sistema, se realizaron 3 diferentes encuestas donde se preguntaban cosas como, ¿qué debía hacer cada rol del sistema? ¿cuántas personas podrían utilizar el sistema?. Después de analizar todas las respuestas de las diferentes encuestas, se establecieron 4 tipos distintos de roles en el sistema, estos son, administradores, locales, usuarios y entidades de salud, cada uno de ellos tendrá diferentes pantallas con funcionalidades distintas, aclararemos que 1

encuesta fue respondida en una reunión virtual donde se grabó la reunión y posteriormente se escribieron todas las respuestas en un documento, las otras 2 fueron enviadas y respondidas por medio de correo electrónico, estas respuestas fueron leídas y analizadas para obtener la mayor cantidad de requisitos posibles, estas encuestas se agregaran en el portafolio para más información sobre las preguntas y respuestas de cada una de ellas, también se hizo una reunión revisando las diferentes pantallas de los roles descritos anteriormente donde se aclararon dudas con respecto a funcionalidades del sistema y también se comentaron correcciones de las diferentes pantallas, por último, se sacaron algunos requisitos mientras nos encontrábamos en clase y el profesor describía lo que esperaba que hiciera el sistema.

### **3.4. Descripción de requisitos:**

#### **Requisitos de locales:**

1. El sistema debe guardar los datos básicos del local, estos son: Tipo de identificación (rut o nit), número de identificación, el nombre del local, departamento, municipio, barrio, teléfono(s) de contacto(1 - 3), correo de contacto, contraseña, tipo de local / categoría, nombre de usuario.
2. El sistema debe permitir ver todas las personas que han visitado el local por medio de una tabla que aparecerá en la pantalla.
3. El sistema debe permitir guardar la temperatura con la que ingreso el usuario.
4. El sistema debe permitir guardar si un usuario tiene o no el tapabocas.
5. El sistema debe permitir recuperar la contraseña por medio de un formulario, después de llenar este formulario se debe enviar la solicitud de cambio a un administrador.
6. El sistema debe permitir modificar los datos de un local, excepto su tipo de identificación, número de identificación y nombre de usuario.
7. El sistema debe permitir buscar a un usuario por código QR.
8. El sistema debe permitir buscar a un usuario por su tipo de documento y número de documento.
9. El sistema debe tener una opción de ingreso a destiempo por si el sistema o el local presentan problemas como por ejemplo, fallas en la

conexión a internet, problemas de energía, problemas en el computador.

10. El sistema debe permitir agregar la fecha en el ingreso a destiempo para tener presente el día en que ingreso el usuario.
11. El sistema debe permitir agregar la hora en el ingreso a destiempo para tener presente la hora en que ingreso el usuario.
12. El sistema debe permitir agregar la temperatura en el ingreso a destiempo.
13. El sistema debe permitir agregar si el usuario trae el tapabocas en el ingreso a destiempo.
14. El sistema debe permitir filtrar los usuarios que han visitado los locales por fechas.
15. El sistema debe permitir ingresar al local por medio de un nombre de usuario y contraseña.
16. El sistema no debe permitir el ingreso a un usuario si está en un periodo de cuarentena.
17. El sistema debe verificar que todos los datos básicos estén completos, estos son: Tipo de identificación (rut o nit), número de identificación, el nombre del local, departamento, municipio, barrio, teléfono(s) de contacto(1 - 3), correo de contacto, contraseña, tipo de local / categoría, nombre de usuario.
18. El sistema debe verificar que los tipos de datos de los datos básicos sean correctos, estos son: Tipo de identificación (rut o nit), número de identificación, el nombre del local, departamento, municipio, barrio, teléfono(s) de contacto(1 - 3), correo de contacto, contraseña, tipo de local / categoría, nombre de usuario.
19. En el apartado donde aparece todos los usuarios que han ingresado al local, deben aparecer todos los datos básicos de un usuario, incluyendo la fecha y hora en que ingreso, temperatura y el tapabocas.
20. El sistema debe permitir la desactivación de la cuenta de un local por parte del dueño del local.

21. El sistema debe permitir la reactivación de la cuenta de un local por parte del dueño del local.

**Requisitos de usuario:**

22. El sistema debe permitir crear una cuenta de usuario.
23. El sistema debe permitir a un usuario desactivar su cuenta en el sistema.
24. El sistema debe permitir a un usuario la reactivación de su cuenta en el sistema.
25. El sistema debe almacenar los datos básicos de un usuario, estos son: Nombre, Apellidos, tipo de documento, número de documento, género, fecha de nacimiento, departamento de residencia, municipio de residencia, barrio, dirección, teléfono de contacto(1), correo, contraseña, nombre de usuario.
26. El sistema debe permitir al usuario ver los locales a los que ha entrado permitiéndole filtrar por fechas.
27. El sistema debe permitir al usuario saber el estado de su salud (si se ha hecho alguna prueba de covid, si el resultado está en proceso, si está infectado o no).
28. El sistema debe permitir modificar los datos básicos, excepto el tipo de documento, el número de documento y el nombre de usuario.
29. El sistema debe generar una petición de modificación a los administradores en caso de que el usuario haga alguna modificación de sus datos básicos, excepto el tipo de documento, el número de documento y el nombre de usuario.
30. El sistema debe validar que todos los campos estén completos, estos son: Nombre, Apellidos, tipo de documento, número de documento, género, fecha de nacimiento, departamento de residencia, municipio de residencia, barrio, dirección, teléfono de contacto(1), correo, contraseña, nombre de usuario.
31. El sistema debe validar que los tipos de datos concuerdan con los campos, estos son: Nombre, Apellidos, tipo de documento, número de documento, género, fecha de nacimiento, departamento de residencia, municipio de residencia, barrio, dirección, teléfono de contacto(1), correo, contraseña, nombre de usuario.

- 32. El sistema debe generar un código QR después de que el usuario cree su cuenta de usuario.
- 33. El sistema debe permitir ingresar al usuario por medio de un nombre de usuario y una contraseña.
- 34. El sistema debe permitir recuperar la contraseña al usuario por medio de un formulario.
- 35. El sistema debe generar una petición de cambio de contraseña a los administradores para aceptar o rechazar la modificación.
- 36. El sistema debe permitir cerrar la sesión en cualquier pantalla del usuario.

**Requisitos entidad de salud:**

- 37. El sistema debe permitir crear una cuenta a una entidad de salud.
- 38. El sistema debe permitir ver los usuarios registrados en el sistema a la entidad de salud por medio de una tabla donde se mostrará su número de documento, tipo de documento, nombres y apellidos.
- 39. El sistema debe permitir recuperar la contraseña por medio de un formulario a una entidad de salud después de que el usuario haga la petición.
- 40. El sistema debe permitir a la entidad de salud reportar a los usuarios que están infectados o no después de realizar la prueba.
- 41. El sistema debe permitir a la entidad de salud reportar los usuarios que están en espera de los resultados.
- 42. El sistema debe poner en un periodo de cuarentena a un usuario que ha dado positivo para covid de 14 días.
- 43. El sistema debe poner en un periodo de cuarentena a un usuario que está en espera de sus resultados hasta que las pruebas salgan.
- 44. El sistema debe permitir ingresar a una entidad de salud por medio de un nombre de usuario y contraseña.
- 45. Después de crear una cuenta de una entidad de salud, se debe enviar una petición de aprobación a una cuenta de un administrador para que pueda acceder al sistema.



- 46. El sistema debe permitir hacer una búsqueda de resultados de los usuarios registrados en el sistema.
- 47. El sistema debe permitir modificar los datos básicos de una entidad de salud.

**Requisitos de administradores:**

- 48. El sistema debe permitir crear una cuenta administrador.
- 49. El sistema debe permitir desactivar una cuenta administrador por el mismo administrador.
- 50. El sistema debe permitir reactivar una cuenta administrador por el mismo administrador.
- 51. El sistema debe permitir ingresar a un administrador por un nombre de usuario y una contraseña.
- 52. El sistema debe permitir consultar todos los usuarios registrados en el sistema a un administrador.
- 53. El sistema debe permitir consultar todos los locales registrados en el sistema a un administrador.
- 54. El sistema debe permitir consultar todas las entidades de salud registradas en el sistema a un administrador.
- 55. El sistema debe tener un apartado donde el administrador pueda aceptar o rechazar peticiones de modificación de un local.
- 56. El sistema debe tener un apartado donde el administrador pueda aceptar o rechazar peticiones de modificación de un usuario.
- 57. El sistema debe permitir agregar, consultar, eliminar y modificar nuevos barrios.
- 58. El sistema debe permitir agregar, consultar, eliminar y modificar nuevas categorías de locales.
- 59. El sistema debe permitir aceptar o rechazar las nuevas categorías para los locales.
- 60. El sistema debe guardar los datos básicos de un administrador, estos son: Nombres, Apellidos, Username, Contraseña. tipo de documento y número de documento.

61. El sistema debe tener un apartado donde se puedan ver las sugerencias escritas por los usuarios del sistema.
62. El sistema debe tener un apartado donde se puedan ver las quejas escritas por los usuarios del sistema.
63. El sistema debe informar a un administrador si otro administrador quiere cambiar su contraseña.
64. El sistema debe permitir el cambio de contraseña de otro administrador.
65. El sistema debe permitir filtrar los locales por categorías.
66. El sistema debe permitir filtrar los usuarios por tipo de documento.
67. El sistema debe permitir filtrar los usuarios por barrios.
68. El sistema debe permitir filtrar los usuarios por departamentos.
69. El sistema debe permitir filtrar los usuarios que estén infectados.
70. El sistema debe permitir ver a todos los usuarios que han visitado un local.
71. El sistema debe permitir ver toda la información básica de un usuario, estos son: Nombre, Apellidos, tipo de documento, número de documento, género, fecha de nacimiento, departamento de residencia, municipio de residencia, barrio, dirección, teléfono de contacto(1), correo, contraseña, nombre de usuario.
72. El sistema debe permitir ver toda la información básica de un local, estos son: Tipo de identificación (rut o nit), número de identificación, el nombre del local, departamento, municipio, barrio, teléfono(s) de contacto(1 - 3), correo de contacto, contraseña, tipo de local / categoría, nombre de usuario.
73. El sistema debe permitir ver toda la información básica de una entidad registrada.
74. El sistema debe permitir consultar los locales que ha visitado un usuario.
75. El sistema debe permitir aceptar o rechazar a un administrador nuevas cuentas de locales.

76. El sistema debe permitir aceptar o rechazar a un administrador nuevas cuentas de entidades de salud.

**Requisitos generales del sistema:**

77. El sistema debe soportar al menos 5000 usuarios concurrentes.
78. El sistema debe asegurar que las contraseñas sean de al menos 10 caracteres.
79. El sistema debe validar que las contraseñas contengan al menos un número.
80. El sistema debe validar que las contraseñas contengan al menos una letra.
81. El sistema debe validar que las contraseñas contengan al menos un carácter especial.
82. El sistema debe responder a un usuario que esté ingresando al sistema en menos de 2 segundos.
83. El sistema debe responder a un usuario que esté filtrando información en menos de 10 segundos.
84. Los datos modificados en la base de datos deben ser actualizados a los usuarios en menos de 3 segundos.
85. El sistema debe cerrar sesión automáticamente tras 15 minutos de inactividad.
86. El tiempo de aprendizaje de los usuarios deberá ser menor a 4 horas en promedio.
87. El sistema debe contar con manuales de usuario estructurados adecuadamente.
88. El sistema debe tener una disponibilidad del 98% de las veces en que un usuario intente acceder.
89. El tiempo para iniciar o reiniciar el sistema no podrá ser mayor a 15 minutos.
90. El sistema será desarrollado para las plataformas PC.

**4. Validaciones:**

Para validar los requisitos del sistema tendremos una cita con uno de los stakeholders que tiene contacto con los demás interesados y además cuenta con más experiencia en levantamiento de requisitos, el profesor gustavo, donde él leerá todos los requisitos y aprobará, corregirá o añadirá nuevos.

Los requisitos se analizaron en la fase de pruebas con un documento de pruebas donde se describirán el tipo de pruebas en el que se van a realizar y la manera en cómo se debe esperar que responda después de realizar dichas pruebas.

**5. Anexos:**

- 5.1. Diagrama de contexto (adjunto en el portafolio).**
- 5.2. Diagrama de casos de uso (adjunto en el portafolio).**
- 5.3. Excel con los requisitos priorizados (adjunto en el portafolio).**
- 5.4. Diagrama de paquetes (adjunto en el portafolio).**