**Especificación de requerimientos de software**

**Software Requirements Specification (SRS)**

**Sistema para gestión de liga NBA**

Integrantes:

Jurgen Adolfo Sanclemente Moncada

Juan Pablo Tascón

Mauricio Bonilla

Nicolas Ramírez

Brian Andrade

Septiembre 2022

Versión 1.0

**Índice General**

1. **Introducción**
   1. *Propósito.*
   2. *Ámbito del Sistema.*
   3. *Definiciones, Acrónimos y Abreviaturas.*
   4. *Referencias.*
   5. *Visión General del Documento.*
2. **Descripción general**

*2.1 Perspectiva del producto*

*2.2 Funciones del producto*

*2.3 Características de los usuarios*

*2.4 Suposiciones y dependencias*

*2.5 Requerimientos futuros*

1. **Requerimientos**

*3.1 Requisitos Funcionales*

*3.2 Requerimientos no funcionales e información para desarrolladores*

1. **Introducción**

En el siguiente documento se explicarán y analizarán los requisitos del proyecto “Sistema gestión Liga NBA”, desarrollado por los estudiantes: Jurgen Sanclemente, Nicolas Ramírez, Juan Pablo Tascón, Mauricio Bonilla y Brian Andrade, del programa Ingeniería de sistemas, Universidad San Buenaventura Cali.

El siguiente documento se basa en otro documento siguiendo la guía de requerimientos de software de la IEE (Std. 830-1993).

**1.1 Propósito**

Este documento pretende dar a conocer el funcionamiento general del proyecto “Sistema gestión Liga NBA” que está dirigido para los desarrolladores anteriormente mencionados y al usuario final.

* 1. **Ámbito del sistema**
* Nombre del sistema: SGLN “Sistema gestión Liga NBA”
* El sistema se encargará de llevar las estadísticas de la actual temporada de la NBA, permitirá la inscripción de equipos y todo lo que estos conllevan, también la planeación de encuentros y registros de información que el sistema se encargara de analizar, organizar e imprimir para el usuario final.
* El principal beneficio del sistema es la facilidad que se le otorga a los organizadores de la temporada de llevar un registro y análisis de la información, ahorrando tiempo y recursos, la meta del sistema es ser la primera opción para la administración de las temporadas.
  1. **Definiciones, Acrónimos y Abreviaturas**
* SGLN: “Sistema gestión liga NBA”
* Administrador: “Persona encargada de ingresar la información y recibirla por parte del sistema”
* BD: “Base de datos y/o archivos donde el sistema recopila la información”
* UML: “Lenguaje de modelo unificado”
* DP: “Diagrama de procesos”
* AN: “Árbol de navegación”
  1. **Referencias**

Protocolos de la W3C.

<https://www.w3.org/standards/webarch/protocols>

Software Requirements Specification (SRS) Template

<https://www.alemana.cl/aac/cur/pdf/SRS+SIA.pdf>

SRS Ejemplo, Sistema Tarifado de Transito

<https://es.slideshare.net/JuAnJoShL/srs-ejemplo-sistema-tarifado-de-transito>

NBA web page

<https://www.nba.com/>

NBA App: basketball en vivo

<https://play.google.com/store/apps/details?id=com.nbaimd.gametime.nba2011&hl=es_CO&gl=US>

* 1. **Visión general del documento**

El documento se divide en 4 secciones:

* Sección 1 se enfoca en la explicación del documento y lo que el lector encontrara.
* Sección 2 es básicamente la descripción general enfocada al sistema y al futuro administrador de este.
* Sección 3 trata sobre los requisitos específicos, con términos especiales para desarrolladores.
* Sección 4 son apéndices donde podrá encontrar imágenes del árbol de navegación, además links directos a las historias de usuario, en las que se basa el sistema.

1. **Descripción General**

Se identificarán factores como el contexto del sistema como el tiempo de desarrollo y sus fases.

* 1. **Perspectiva del producto**

El producto final permite una administración eficiente de la información relacionada a la temporada de la liga NBA, donde se encuentran las inscripciones de equipos, creación de partidos y recopilación de datos en base a estos encuentros, todos esos datos el sistema permitirá una fácil abstracción por parte del administrador, ya que el mismo software los digerirá y mostrará en un lenguaje de alto nivel, contando con una interfaz de fácil acceso.

* 1. **Funciones del producto**
* Registro de equipos.
* Registro de resultados obtenidos en partidos.
* Visualización de tabla con los rankings de los equipos.
  1. **Características del administrador**

El sistema cuenta con un tipo de usuario final llamado “administrador”:

* Esta persona trabaja para la empresa NBA o posee una relación con esta para poder modificar el sistema.
* Tiene conocimientos del deporte y por ende le resulta fácil entender los diferentes términos que se encontrara dentro del software.
* Habilidades básicas en estadística, abstracción de información y tecnología.
  1. **Restricciones**

El sistema solo podrá ser modificado por personas anexas a la empresa que supervisa la “NBA”, de resto no posee más limitantes, ya que el software es de uso local y no necesita ningún tipo de conexión vía internet

* 1. **Suposiciones y dependencias**

Para el correcto funcionamiento del sistema, se debe tener pre instalado el software SGLN, en la maquina donde se necesite usar y otorgarle los permisos necesarios para el manejo de archivos.

* 1. **Requerimientos futuros**

En un principio el software contara con 3 módulos principales (inscripción de equipos, registro de partidos y visualización de estadísticas), en un futuro se pueden llegar a ampliar estos requerimientos.

1. **Requerimientos específicos**

En la próxima sección se expondrá a profundidad los requerimientos (funcionalidades) del sistema.

**3.1 Requerimientos funcionales**

Administrador al inscribir equipos

|  |  |
| --- | --- |
| **Cod:** | 1000 |
| **Descripción** | El administrador puede ingresar información de jugadores |
| **Descripción detallada** | El sistema solicitara los datos de los jugadores dependiendo del equipo seleccionado |
| **Precondiciones** | Se selecciona la opción “Inscribir equipo” en el menú principal |
| **Entradas** | Nombre del jugador, numero en el dorsal, altura y peso |
| **Proceso** | El sistema guardara la información de cada jugador en el respectivo equipo |
| **Salidas** | El sistema carga la interfaz del siguiente jugador hasta completar los 12, después pasa a para añadir al trabajador del equipo |
| **Postcondiciones** | El jugador queda registrado en el equipo y se podrá utilizar en diferentes acciones |
| **Roles involucrados** | Administrador |
| **Verificación** | Si |

|  |  |
| --- | --- |
| **Cod:** | 1001 |
| **Descripción** | El administrador puede ingresar información del personal de los equipos |
| **Descripción detallada** | Se permite ingresar un trabajador al registrar un equipo |
| **Precondiciones** | Se tuvo que haber seleccionado el menú “Inscribir equipo” donde se tuvo que llenar la información de los 12 jugadores |
| **Entradas** | Se pedirá la información de los trabajadores de los equipos como nombre, experiencia∫ y cargo |
| **Proceso** | Se guardará el empleado en el registro |
| **Salidas** | Se pasa al menú de inscribir los datos del equipo |
| **Postcondiciones** | Los empleados de los equipos quedaran registrados en los archivos |
| **Roles involucrados** | Administrador |
| **Verificación** | Si |

|  |  |
| --- | --- |
| **Cod:** | 1002 |
| **Descripción** | Inscribir equipos |
| **Descripción detallada** | El administrador puede inscribir un equipo al sistema |
| **Precondiciones** | Se debió seleccionar la opción en el menú “Inscribir equipo”, haber llenado los 12 jugadores y la información del trabajador. |
| **Entradas** | Se debe ingresar el nombre del equipo, locación, el número de temporadas ganadas. |
| **Proceso** | Los datos del equipo quedaran registrados en el sistema |
| **Salidas** | Se carga un mensaje de retroalimentación exitosa. |
| **Postcondiciones** | El equipo queda registrado en los archivos de texto del sistema |
| **Roles involucrados** | Administrador |
| **Verificación** | Si |

Administrador al crear partidos

|  |  |
| --- | --- |
| **Cod:** | 2001 |
| **Descripción** | Ingreso de información de partido |
| **Descripción detallada** | El administrador puede ingresar los datos de un partido finalizado |
| **Precondiciones** | Se debió seleccionar la opción “Crear partido” en el menu principal |
| **Entradas** | Se ingresa información como puntajes por parte de ambos equipos y numero de faltas totales.  Nombres de ambos equipos, fecha de juego, nombre a el árbitro, nombre de estadio, equipo ganador y espectadores totales.  Al cargar la información de los jugadores individualmente se debe ingresar:  Puntos realizados por el jugador, asistencias, rebotes, tapones, robos de balón y faltas. |
| **Proceso** | El sistema recopilara la información del partido finalizado además de los jugadores y pasara a la fase de realizar cálculos |
| **Salidas** | Se muestra un mensaje de éxito |
| **Postcondiciones** | El partido quedará en el historial de juegos por temporada, cada jugador recibirá una actualización en sus estadísticas, al igual que los equipos |
| **Roles involucrados** | Administrador |
| **Verificación** | Si |

Administrador al modelar estadísticas de equipos

|  |  |
| --- | --- |
| **Cod:** | 3000 |
| **Descripción** | Cargar lista de equipos |
| **Descripción detallada** | El administrador podrá ver los mejores equipos organizados en una tabla |
| **Precondiciones** | Haber seleccionado el menú “Estadísticas” |
| **Entradas** |  |
| **Proceso** | El software buscará todos los equipos registrados y mostrará la información |
| **Salidas** | En una interfaz, se visualizará por cada equipo su nombre y estarán organizados, en la parte más alta el que tenga más partidos ganados y entre más abajo, menos ganados. |
| **Postcondiciones** | La información debe ser clara, además debe presentarse la opción de escoger alguno de los equipos |
| **Roles involucrados** | Administrador |
| **Verificación** | Si |

**3.2 Requerimientos no funcionales e información para desarrolladores**

* El sistema actualizara la información en tiempo real y archivara todos los datos en la BD (archivos locales).
* El sistema tendrá una interfaz intuitiva de fácil navegación
* La visualización de los datos se realizará de forma detallada

El lenguaje de diseño será modelado unificado (UML), el patrón de arquitectura en “N-capas”, el lenguaje de desarrollo será JAVA y los datos se guardarán en archivos tipo. texto.

El repositorio principal se encuentra en GitHub (organización Trompitas-team-SA) <https://github.com/orgs/Trompitas-team-SA/dashboard>

Manejo de requerimientos controlados desde atlassian.net (Jira software) <https://art-anime-01.atlassian.net/jira/software/projects/LIGA/boards/3/backlog?selectedIssue=LIGA-10>

Se utilizará una metodología ágil llamada SCRUM iterativa.

Para información adicional ingrese al GitHub de la organizacion.