SPORTS SCHEDULING

1. **ANÁLISIS**

En el ámbito deportivo del Futbol Profesional Colombiano, actualmente se realizan programación de partidos para la liga Águila de forma prácticamente aleatoria sin ningún algoritmo de optimización, a excepción de unas reglas básicas como los partidos clásicos en una fecha determinada o que el ganador de la Superliga debe de jugar de primero en la liga.

Actualmente, en la facultad de matemáticas e ingeniería de la Konrad Lorenz lleva a cabo un proyecto el cual pretende usar algoritmos para crear automáticamente un fixture (calendario deportivo) para los partidos de la Liga Águila en Colombia, basándose en algoritmos como el cuadrado latino para organizar de forma automática dicho calendario y a su vez usar y calcular los vectores creados para asi calcular las distancias que debe recorrer cada equipo tomando en cuenta si es local o visitante, de manera que el fixture creado sea no solo automático, sino que sea el mas optimo a nivel de distancias recorridas en total por todos los equipos, de forma que pretende ahorrar dinero a la Federación Colombiana de Futbol y a su vez, ahorrar a los jugadores del estrés de vuelos largos y consecutivos, lo cual afecta en su rendimiento deportivo.

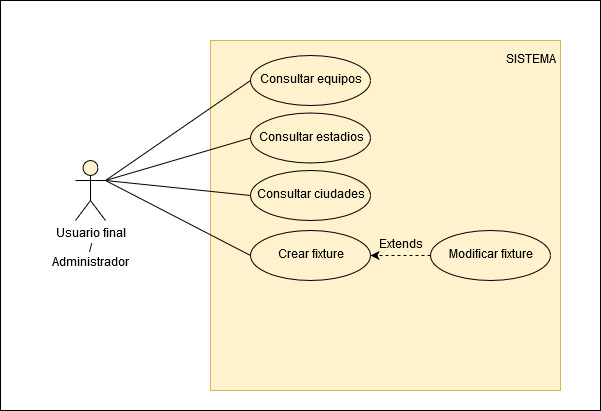
La Federación Colombiana de Futbol es la encargada de cubrir los gastos de viaje de los equipos y no sus clubes, por lo que por año se gasta un dinero aleatorio ya que depende del fixture que quede en el año el que determine el numero de viajes totales que deben realizar los equipos y las distancias que estos recorren, la cual si es aleatoria, no se tiene un estimado de cuanto se gastara al año, sin embargo, con un software que calcula la distancia aproximada de cada estadio y en base a eso definimos los costos de los viajes, podemos realizar un fixture con los costos mínimos de viaje en el año dando un estimado en pesos colombianos.

Por tal motivo se pretende realizar el software “SPORTS SCHEDULING” que pretende innovar en el marco del Futbol en Colombia con el fin de cubrir esta necesidad que se tiene de cambiar el viejo marco de creación de los horarios de los partidos y asi realizar un fixture más inteligente, que ahorre dinero en gastos de viaje y disminuir estrés en los jugadores.

1. **DISEÑO**

Como primer paso se diseño el diagrama de casos de uso, el cual por el momento es sencillo ya que los requerimientos son de un solo usuario final el cual es el que usara la aplicación, luego se realizaron las respectivas especificaciones de los casos de uso. Para poder describir la estructura del sistema se diseño el diagrama de clases. Finalmente se diseñó el modelo entidad/relación, el cual con respecto a las historias de usuario.

* 1. **Ejemplo de los casos de uso**

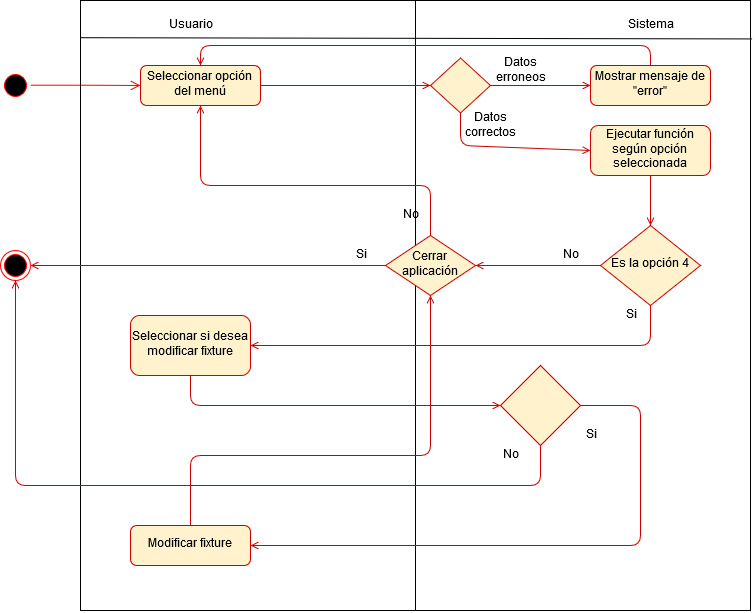


*Diagrama 1: Caso de uso general.*

En el diagrama de casos de uso (Diagrama 1) se presentan las diferentes acciones que puede realizar el usuario final con el sistema “SPORTS SCHEDULING”.

**Usuario final / Administrador:** Es el encargado de interactuar con la aplicación, el encargado de ver los equipos que pertenecen y juegan en la Liga Aguila y sus respectivos estadios y en las ciudades en las cuales se encuentran ubicados, tiene permisos de no solo crear los fixtures sino de modificarlos con el fin de cuadrarlo al gusto, teniendo en cuenta que las modificaciones afectan el costo de los viajes final calculado en la aplicación.

* **Consultar equipos:** Esta acción será realizada por el usuario final / administrador, el cual le permitirá listar los equipos que pertenecen a la Liga Águila.
* **Consultar estadios:** Esta acción será realizada por el usuario final / administrador, el cual le permitirá listar los estadios de los equipos pertenecientes a la Liga Águila.
* **Consultar ciudades:** Esta acción será realizada por el usuario final / administrador, el cual le permitirá listar las ciudades de los equipos pertenecientes a la Liga Águila.
* **Crear fixture:** Esta acción será realizada por el usuario final / administrador, el cual le permitirá crear un fixture o calendario deportivo.
* **Modificar fixture:** Esta acción podrá ser realizada por el usuario final / administrador después de crear el fixture, el cual le permitirá modificar a gusto el fixture.
  1. **Ejemplo de los diagramas de actividades**

****

*Diagrama 2: Diagrama de actividades.*

Este diagrama muestra todo el proceso de uso de la aplicación a manos del usuario final o administrador y todas las opciones y posibles rutas que puede seguir este único usuario.

1. **IMPLEMENTACIÓN**

La implementación del diseño se realizó a través de programación orientada a objetos, utilizando el lenguaje de programación Java, librería de componentes Vector y Arrays.

Para mantener un control de versiones adecuado, se utilizo GitHub como repositorio.

1. **PLATAFORMA**

Lenguaje de programación: Java.

1. **AMBIENTE**

Esta aplicación es de escritorio.