WEBIR - Centralización de información de casas de apuestas

Integrantes:

Juan Andres Nogueira

Alexis Artus

Andrés Kent

Federico Perez

Andrés Borges

Noviembre, 2019.

Introducción

Las casas de apuestas ofrecen la posibilidad de pronosticar resultados deportivos de distintas disciplinas, como ser fútbol y basquetbol. A su vez, dentro de cada disciplina, ofrecen diferentes tipos de apuestas. Por ejemplo, en fútbol, una modalidad de apuesta consiste en pronosticar los ganadores (o empate) de una serie de partidos. Otra modalidad de apuestas en fútbol consiste en pronosticar el resultado exacto de un partido.

En este trabajo nos centraremos en la disciplina fútbol con la primera modalidad de apuesta mencionada. En esta modalidad, para un partido dado, cada posible resultado tiene un dividendo asociado. Por ejemplo: si se enfrentan Peñarol y Defensor y tenemos los siguientes dividendos:

Gana Peñarol	Empate	Gana Defensor
1,35	2,25	3,20

Estos dividendos significan que, si el usuario pronostica el resultado correcto (gana Peñarol, empate o gana Defensor) recibirá “x” veces lo apostado. A modo de ejemplo: si el usuario pronostica que gana Peñarol, apuesta $50 y el pronóstico se cumple, recibirá $50*1,35 = $67.5.

En esta modalidad de apuestas, los usuarios generalmente deciden pronosticar los resultados de al menos 3 partidos para hacer crecer el dividendo total. Si el usuario acierta el pronóstico de todos los partidos, cobra X veces lo apostado, donde X es el producto de los dividendos de cada uno de los partidos. Ejemplo: el usuario pronostica 3 partidos con dividendos 1,34, 2,77 y 3,15 y acierta, en dicho caso cobrará 1,34 x 2,77 x 3,15 = 11,69 veces lo apostado.

Los dividendos de cada partido varían según la casa de apuesta y es aquí donde entra la utilidad del presente trabajo.

Problema a resolver

El problema que se intentará resolver en el presente trabajo es implementar un sistema que, dada una apuesta que un usuario desea realizar, encuentre la casa de apuestas que paga mejores dividendos para dicha apuesta.

En principio, nos limitaremos a las siguientes restricciones:

- Tener en cuenta únicamente la disciplina fútbol en la modalidad de apuesta mencionada en la introducción.
- Únicamente utilizar la liga de fútbol de primera división del Campeonato Uruguayo.
- Dos casas de apuestas de las cuales se extraerá la información. Las mismas serán Supermatch[1] y Bet365 México[2].

La extracción de información de las webs de cada una de las casas de apuestas se implementará de forma modular. Esto permitirá que, si en un futuro se desean agregar nuevas casas de apuestas, sea fácil realizarlo.

Enfoque de la solución

En nuestro trabajo se recupera la información de las casas de apuestas respecto a los partidos del campeonato de primera división del fútbol uruguayo. A cada posible resultado (ganar, empatar o perder) las distintas casas de apuestas le asignan un dividendo de paga, que si bien entre las distintas casas de apuesta utilizan algoritmos de predicción con cierta similitud los dividendos se diferencian.

En la web se le ofrece al usuario que seleccione la combinación de resultados de partidos que desee apostar, al momento de ir seleccionando se le muestra los dividendos totales diferentes que le pagaran las distintas casas de apuestas.

A la hora de recuperar la información de las webs de casas de apuestas nos encontramos con la dificultad de que no proveen APIs, por lo tanto la recuperación se debió realizar utilizando Web Scraping.

Diseño

Herramientas

A continuación se hace un listado de las herramientas a utilizar:

Ruby[3]

Ruby es un lenguaje de programación interpretado, reflexivo y orientado a objetos. Con el cual se utilizará la técnica de scraping para obtener la información necesaria de las páginas mencionadas. Con ello se generarán API Rest para el frontend.

SQLite[6]
SQLite es una biblioteca “in-process” que implementa un motor de base de datos SQL autónomo , sin servidor y transaccional. El código para SQLite es de dominio público y, por lo tanto, es de uso gratuito para cualquier propósito, comercial o privado. SQLite es la base de datos más implementada en el mundo con más aplicaciones de las que podemos contar, incluidos varios proyectos de alto perfil. A diferencia de la mayoría de las otras bases de datos SQL, SQLite no tiene un proceso de servidor separado. SQLite lee y escribe directamente en archivos de disco normales. Una base de datos SQL completa con múltiples tablas, índices, disparadores y vistas está contenida en un solo archivo de disco.

React[4]
React, es un framework de aplicaciones web basado en Javascript de código abierto y diseñada para crear interfaces de usuario con el objetivo de facilitar el desarrollo de aplicaciones en una sola página.

Selenium[5]
Selenium es un entorno de pruebas de software para aplicaciones basadas en la web. Selenium provee una herramienta de grabar/reproducir para crear pruebas sin usar un lenguaje de scripting para pruebas.

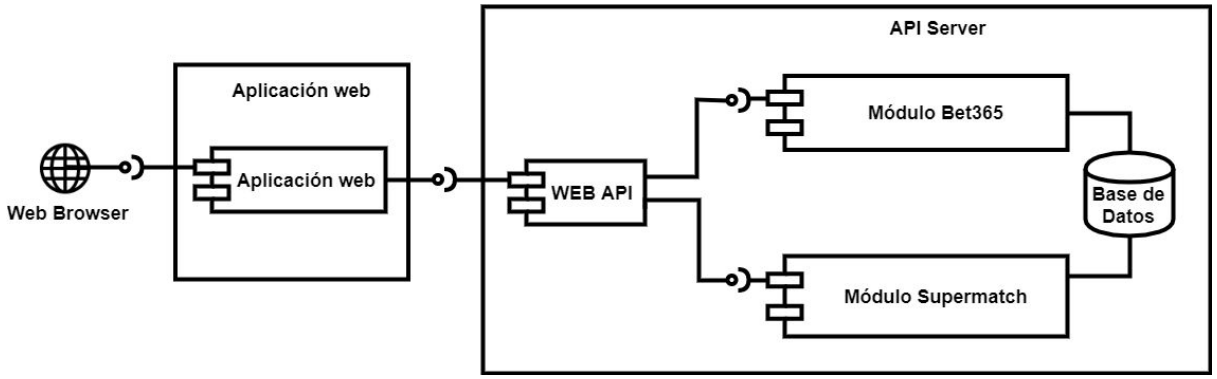
Implementación

Se decide utilizar Selenium para el scraping, ya que las fuentes elegidas cargaban la información dinámicamente. Selenium permite sortear este obstáculo ya que representa una simulación de navegación como si lo hiciera cualquier persona. Además, se cuenta con la funcionalidad de esperar a que cierta información sea cargada, de esta forma extraer los datos de forma certera.

Pero la utilización de Selenium también aportó un problema, la simulación de la navegación es costosa en tiempo. Es inviable levantar un navegador por cada fuente, y luego hacer la navegación por cada usuario que necesite acceder los datos, ya que los tiempos de respuestas serían muy largos, y el consumo de recursos muy elevado.

Para solucionar este inconveniente se implementa un respaldo de los partidos, es decir que los usuarios acceden directamente a los datos previamente procesados, incrementando mucho la velocidad de respuesta. Se tiene una base de datos en donde se guardan todos los partidos, identificados por los equipos y la fecha, y luego se implementa una carga automática que por defecto se ejecuta cada media hora, en donde se realiza la simulación una única vez.

Arquitectura



El diseño de la arquitectura consiste en dos grandes partes, por un lado el API Server implementado en Ruby el cual abarca el scraping, y proporciona una Api para poder consultar todos los datos obtenidos. Por eso en este servidor también se tiene la base de datos del sistema.

Por otro lado se tiene una aplicación web implementada en React, con un servidor propio en donde se presenta el fron-end del sistema y los usuarios pueden consultar a travez de un navegador web.

Funcionalidades y Uso

Cuando el usuario ingrese a la página se encontrará con el listado de partidos próximos a disputarse. Cada partido tiene 3 botones, donde los botones indican si gana el local, empatan o gana el visitante respectivamente.
Al hacer clic en una de las opciones en el sector derecho de la página se muestra los dividendos que pagan las distintas casas de apuestas. De esta forma el usuario puede determinar en qué casa de apuesta le es más redituable realizar la apuesta deseada.
Los dividendos son tomados de las distintas casas de apuestas, y al realizar una combinación de al menos dos partidos, la forma de calcular el dividendo total es mediante el producto de los dividendos individuales.

Evaluación y resultados

A partir de la implementación antes comentada se logró llegar al siguiente producto.

PARTIDOS			SUPERMATCH			BET365		
Cerro largo	X	Racing	1.65	3.6	3.8	1.66	1.89	3.6
Plaza Colonia	X	Juventud de las piedas	1.6	3.35	4.45	1.69	3.4	4.55
Danubio	X	Liverpool	2.55	2.9	2.45	2.58	3	2.5
Fenix	X	Progreso	2.65	3.15	2.2	2.75	3.2	2.15
Cerro	X	Rampla	1.7	3.3	3.9	1.8	3.35	3.85
River	X	Defensor SP	2.3	3.05	2.6	2.25	3	2.67
Boston River	X	Wanderers	2.1	3.2	2.8	2.16	3.3	2.9
Nacional	X	Peñarol	2.3	3.3	2.45	2.35	3.35	2.5
TOTAL:				368.99			422.50	

En donde como se puede ver se seleccionan los partidos a apostar y se muestran los dividendos de cada sitio de apuestas.

Por otro lado, con el servicio mencionado en la sección de implementación se logró obtener la información en formato json. El resultado del mismo es el siguiente:


```
{
      "id":8,
      "visitante":"Peñarol",
      "local":"Nacional",
      "dividendoLocalSM":2.3,
      "dividendoVisitanteSM":2.45,
      "dividendoEmpateSM":3.3,
      "dividendoLocalBet":2.35,
      "dividendoVisitanteBet":2.50,
      "dividendoEmpateBet":3.35,
      "fecha":"2019-11-17T16:00:00.000Z",
      "clave":"NacionalPeñarol20191117",
      "created_at":"2019-11-11T22:47:28.171Z",
      "updated_at":"2019-11-11T22:47:28.171Z"
}
```

Como se puede ver se tiene toda la información que se necesita tener en la web para una buena experiencia de usuario.

Las pruebas realizadas constatan la correcta información de partidos con sus respectivos dividendos en las dos casas de apuestas. Debido a que nos limitamos a solo dos casas de apuestas la comparación de dividendos finales para el usuario están limitados actualmente. Sin embargo debido al diseño modular del sistema creemos que es un buen prototipo para continuar. Asimismo destacamos la gran utilidad de recuperar información en la web sin dejar de tener en cuenta sus limitaciones y la complejidad que puede alcanzar en casos particulares.

Conclusiones

Evaluamos que el sistema puede llegar a tener una mejor utilidad a medida que se agreguen nuevas ligas, nuevos deportes y nuevas casas de apuestas. A la hora de diseñar el sistema se tuvo en cuenta que permita agregar casas de apuestas sin tocar código ya existente, que la transición sea simplemente agregar un módulo que obtenga la información de la casa de apuesta deseada y la normalice con las herramientas que sean más adecuadas para dicha web.

Trabajos a Futuro

- Como primer trabajo a futuro consideramos agregar más ligas al sistema, en la actualidad por simplicidad sólo se utilizó la liga Uruguaya de fútbol de primera división. Esto puede ser realizado sin alterar las funcionalidades actuales gracias a la implementación por módulos que se tiene. Incluso se podría agregar sobre otros deportes.

- Se podría desarrollar una aplicación mobile (por ejemplo utilizando React Native) para que los usuarios puedan consultar desde sus teléfonos celulares. En este caso al usar una tecnología como React Native se podría reutilizar mucha de la lógica que se tiene en la web. Esto permitiría una rápida implementación y no tener que reinventar la rueda para la lógica a desarrollar.

- En la actualidad la normalización de los datos es basada en un diccionario. Dada la realidad con la que trabajamos no es un problema, pero al escalar y teniendo como consecuencia que la cantidad de equipos y casas de apuestas crezcan implica que al agregar un equipo se debe chequear a mano en el diccionario como maneja el nombre del equipo cada casa de apuestas y lograr un acuerdo.

- Basándonos en las búsquedas que decidan realizar los usuarios se podrían generar estadísticas y recomendar a futuros usuarios sobre cuáles son las combinaciones más consultadas.

Referencias

[1] Supermatch
https://www.supermatch.com.uy/

[2] Bet365
https://www.bet365.mx/#/HO/

[3] Ruby
https://www.ruby-lang.org/es/

[4] React
https://es.reactjs.org

[5] Selenium
https://www.seleniumhq.org/

[6] SQLite
https://www.sqlite.org/