



# Informe Final

## Recuperación de información y Recomendaciones en la Web

Grupo 14

Francisco Betancor 4665908

Daniel Acuña 4927569

Profesora Libertad Tansini

## Introducción

Muchas veces se ve a los videojuegos en pc y en general como un medio de difícil acceso (principalmente en nuestro país) tanto por sus costos económicos como por falta de información de dónde y cómo consumir estos productos.

Debido a esto muchas personas quedan marginadas a no poder disfrutar de ellos u optan por adquirirlos de forma ilegal (descarga de copias desde páginas ilícitas o compras de copias piratas en ferias) perdiendo soporte oficial y funcionalidades de estos.

## Problema

Cuando un jugador quiere comprar un juego puntual en un momento dado, es probable que se dirija a la plataforma más conocida para esto y solo se quede con el precio que allí aparezca sin conocer páginas alternativas donde el juego en cuestión puede estar a menor precio con descuentos importantes de hasta el 75%. Gastando más dinero del necesario o en el peor de los casos no pudiéndose permitir el producto.

## Enfoque y descripción de la solución

Para solucionar este problema se hizo GünstigeSpiele, donde se reunieron precios de distintas paginas de ventas donde poder encontrar el menor de ellos, bajo la condición de estar dentro de la región en que se ofrecen estos precios, usando las conversiones de moneda para la comparación y mostrando los precios obtenidos en una misma moneda del país locatario de donde se quiera comprar(en nuestro caso pesos uruguayos consultando a las cotizaciones del BROU). De esta manera el usuario tendrá oportunidad de adquirir videojuegos a un precio accesible y de manera legal, además de conocer otros sitios de compra.

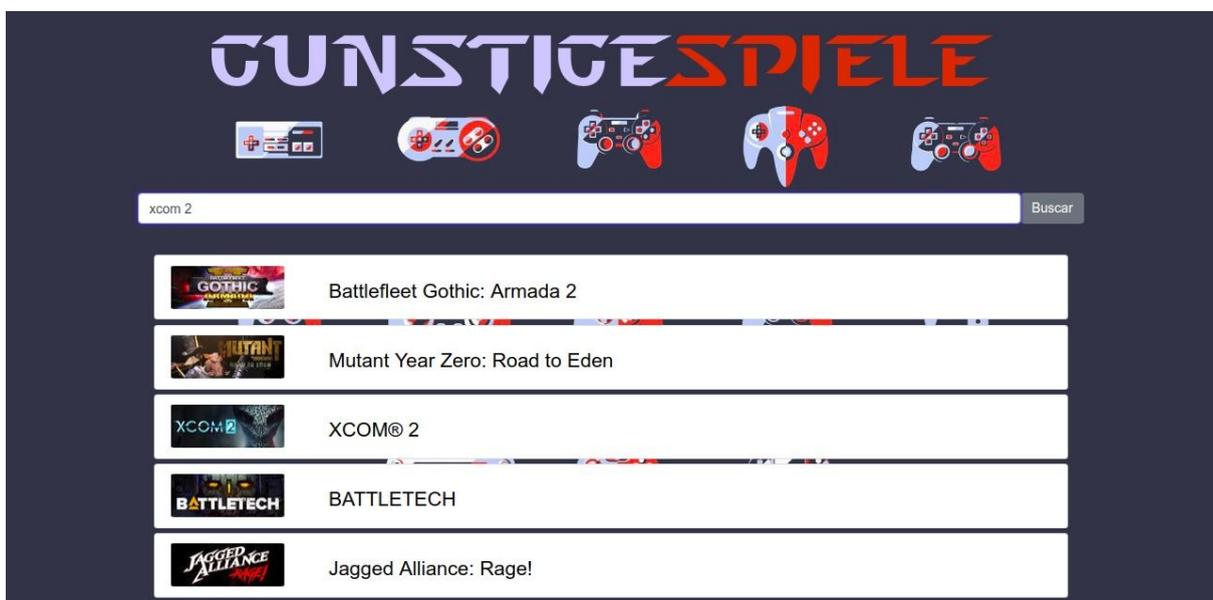
## Funcionalidades y uso

Su única y principal funcionalidad al momento es poder encontrar el mejor precio entre las tiendas definidas para un juego definido.

En primer lugar como página principal nos encontraremos con el título de la aplicación web y una barra de búsqueda donde poner el título del juego que el usuario está buscando.



Luego se listara los juegos que tengan un título similar al de la búsqueda para que el usuario seleccione uno de estos y así poder obtener el juego el nombre exacto del juego deseado y poder ir a la web de las distintas tiendas en busca de sus precios.



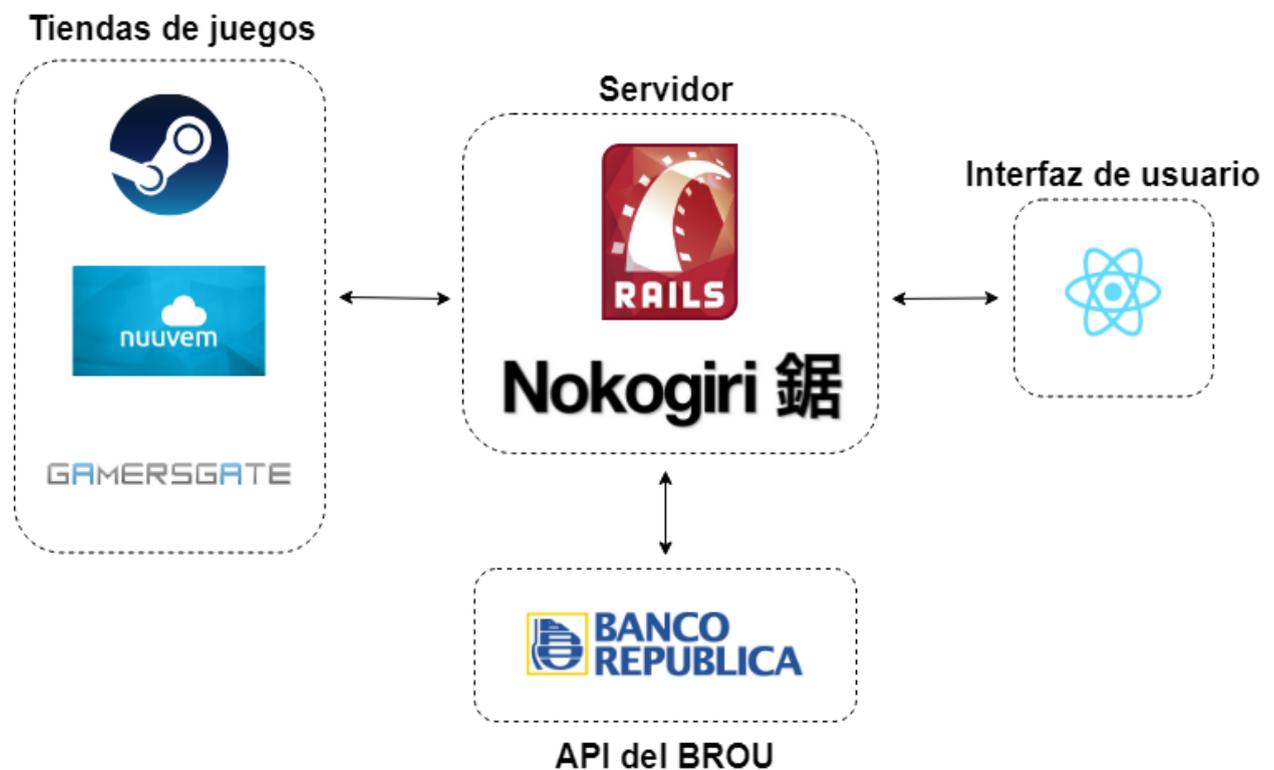
Una vez seleccionado el juego se listarán los precios de las distintas tiendas en orden ascendente dejando un link directo a la página del artículo para que el usuario pueda ir a comprar el juego.

# GUNSTIGESPIELE

## XCOM® 2

TIENDA	PRECIO	DISPONIBLE	URL
Nuuvem	R\$ 29.99 - (\$U 264)	✗ No disponible para su región	<a href="https://www.nuuvem.com/item/xcom-2">https://www.nuuvem.com/item/xcom-2</a>
Gamers Gate	US\$ 15 - (\$U 498)	✓	<a href="https://latam.gamersgate.com/DD-XCOM2-GN/xcom-2">https://latam.gamersgate.com/DD-XCOM2-GN/xcom-2</a>
Steam	\$U 771	✓	<a href="https://store.steampowered.com/app/268500">https://store.steampowered.com/app/268500</a>

## Arquitectura



## Implementación

Para la implementación primero se eligió la plataforma Steam como la raíz donde hacer la primer búsqueda de juegos. Para realizar la primer búsqueda se utilizó la url de búsqueda de Steam agregando como parámetro por url el texto de la búsqueda hecha en nuestra aplicación. Luego se muestra al usuario los resultados que nos proporciona steam, a partir de este punto se estudia el código fuente de la página para poder extraer información de cada uno de estos. Toda esta información se recolecta en el backend y se envía al front end en formato JSON.

Una vez se selecciona el juego el front end envía al backend el id único que lo identifica junto con el nombre recuperados anteriormente y de esta manera obtener datos exactos del juego para empezar a recolectar los precios de las distintas tiendas. A partir de este punto el backend recoge toda la información utilizando la gema nokogiri tanto de los precios en las tiendas como de las cotizaciones en el brou que maneja cada tienda y se encarga de darle un formato estándar para comunicarse con el frontend de tal manera que este tenga que hacer el mínimo esfuerzo para presentarla. Para extraer la información de cada web también se tuvo que estudiar el código fuente de cada una y utilizar selectores de CSS para obtener la información. Dado que son datos que cambian cada día o incluso cada hora por cada búsqueda se debe extraer la información de la web sin utilizar una base de datos.

## Desafíos

Uno de los primeros desafíos a los que nos presentamos fue lo poco que se acomodaba las tecnologías elegidas en el informe de avance a la naturaleza de lo que queríamos hacer, lo cual nos llevo un par de días de investigación dándonos cuenta que era casi un problema para estudiar por sí mismo y poco explorado por lo que apreciamos. Por lo tanto decidimos casi forzosamente a cambiar hacia rails y nokogiri el cual hacía exactamente lo que necesitábamos.

El segundo gran desafío fue, dada la naturaleza de nuestra aplicación, tener que comparar el mismo juego de distintas páginas sin poder estar seguros de realmente se estuviera comparando el mismo producto y no dos distintos con nombre muy parecido.

Dado que teníamos que lidiar con los buscadores de cada una de estas tiendas y de los resultados quedarnos con el juego que realmente era, sumado al hecho de que los nombres de los juegos a veces solo se diferencian en un número o de símbolos especiales como notas de música o “Marcas registradas” , se tuvo que manejar

distintos algoritmos para estipular que se tratara del mismo juego con cierto margen de error, hasta que al final dimos con el que nos dio una efectividad de 100%, quitando todo los símbolos no ASCII del nombre y buscando una igualdad absoluta en el nombre.

El tercer desafío fue obtener información crucial como las cotizaciones del brou y darle un formato único a toda la información reunida de las distintas páginas. Además se intentó agregar más tiendas para hacer más completa las comparaciones, sin embargo en las búsquedas de estas no se devolvía los resultados en una página “estática” sino que una vez cargada la página se hacía la consulta de forma “dinámica” lo cual dificulto la obtencion de informacion de estas.

El cuarto desafío fue los distintos estados en los que puede estar un juego y la forma inconsistente en cómo una misma página lo manifiesta teniendo que abarcar manualmente cada una de estas a medida que sea posible. Por ejemplo para el caso de las regiones o de los juegos que todavia no estan disponibles.

## Evaluación y resultados

A pesar de los desafíos anteriormente mencionados las pruebas hechas han dado resultados casi exactos en todas nuestras pruebas. Aunque debido a las pocas tiendas con las que se hace la comparación de los precios los resultados que aporta están limitados actualmente. Sin embargo se ha podido cubrir la mayoría de estados de los juegos en cada página y los resultados en estas tres tiendas son robustos lo cual es un buen prototipo para continuar. Además pudimos apreciar la gran utilidad de recolectar información de la web como también sus limitaciones y la complejidad que puede alcanzar en casos particulares.

## Trabajo Futuro

Sin duda el margen de mejora de nuestra aplicación es enorme. La primera y más obvia mejora es en la cantidad de tiendas, agregando cada una de las tiendas oficiales conocidas (las cual se cuentan por decenas) se obtendrían los verdaderos precios más bajos, además de mejorar la aplicación agregando entradas individuales para cada juego y recolectando solo la mínima información necesaria (dado que tanto el nombre, como el link del producto y la imagen son fijos) en vez de recolectar todo como se hace actualmente agregando una importante complejidad que está sujeta a un margen de error más grande. Esto nos lleva a la segunda mejora...

La segunda gran mejora sería agregar una base de datos donde se pudieran hacer gráficas y guardar datos relevantes como los mínimos históricos en los precios de cada juego para que la persona que vaya a comprar pueda verlo y decidir si la oferta actual es lo suficientemente buena o debe esperar un poco más para obtener el mejor precio.

La tercer gran mejora sería crear un sistema de usuarios donde estos pudieran hacer una "wishlist" agregando los juegos que quieren comprar en un futuro y que el sistema les avise cuando estos alcancen o igualen su precio mínimo histórico en una oferta, además de que el sistema pueda recomendarle juegos en oferta que sean similares a los que ellos tienen agregados en la wishlist (lo cual implicaría tener que implementar un sistema de categorización y tags para los juegos ).

La cuarta y última gran mejora que mencionaremos sería ampliar el espectro de la aplicación y no solo limitarse a juegos para PC, sino a juegos para otros dispositivos y también ofertas de hardware para videojuegos como consolas o periféricos de PCs.

## Referencias

<https://www.rubydoc.info/github/sparklemotion/nokogiri>

<https://reactjs.org/docs/getting-started.html>

<https://ryanstutorials.net/css-tutorial/css-selectors.php>

<https://reactstrap.github.io/>