

# Informe Final. Clash Royale

## Ingeniería de Software + Base de Datos.

Karen Dianelis Cantero Lopez. C311  
Luis Alejandro Rodriguez Otero C311  
Hector Miguel Rodriguez Sosa C311  
Sebastian Suarez Gomez C311

## 1 Introducción

El producto tiene como objetivo analizar los datos de los usuarios de Clash Royale y proporcionar la información más relevante sobre su comportamiento, patrones de juego, estrategias y rendimiento. Se utilizan análisis de métricas de participación y evaluación de la experiencia del usuario. En el presente informe se describe el proyecto. Se presentan los requerimientos específicos funcionales, no funcionales y del entorno. Las funcionalidades del producto se muestran a partir de un modelo de casos. En la arquitectura del proyecto se presentan la arquitectura general y la arquitectura de diseño del backend. En el modelo de datos se muestra el modelo de base de datos. Estos y otros aspectos significativos para la realización del producto se abordan a continuación.

## 2 Requerimientos Específicos

### Requisitos Funcionales

1. El usuario puede registrarse e iniciar sesión.
2. El usuario puede visualizar gráficamente la popularidad del juego; es decir, puede ver gráficamente la cantidad de batallas y desafíos en el año.
3. Realizar batallas entre jugadores simulando el resultado. Visualizar las cartas con sus características.
4. Realizar operaciones CRUD sobre las entidades.
5. Realizar consultas a la base de datos y exportar los resultados de dichas consultas en archivos .pdf o .csv.
6. Realizar un análisis del mazo del jugador.
7. Realizar análisis de métricas de participación: el sistema es capaz de analizar y proporcionar información detallada sobre la participación de los usuarios en el juego.

### Requisitos No Funcionales

1. Seguridad: El sistema garantiza la confidencialidad, integridad y autenticación de los datos de los usuarios. El acceso a la plataforma es controlado por nombre de usuario y contraseña. Una vez el usuario esté autenticado en la plataforma, se le asignará un token jwt. En caso de este expirar luego de haberse iniciado sesión, se necesitará que se autentique de nuevo.
2. Rendimiento: El sistema es eficiente y tiene un rendimiento óptimo. Asegura tiempos de respuesta rápidos y capacidad para manejar el volumen de datos y usuarios sin degradar la calidad del servicio.
3. Usabilidad: La interfaz del sistema es intuitiva, fácil de usar. Se siguen las mejores prácticas de diseño de experiencia de usuario (UX) para garantizar una interacción fluida y positiva.

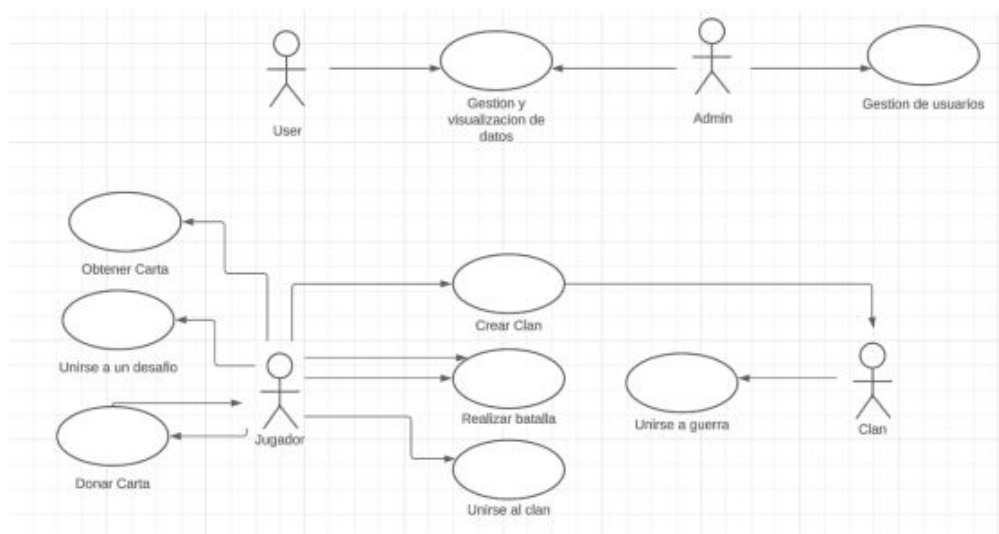
4. El sistema almacena todos los datos en una base de datos SQL Server. A través del sistema se pueden realizar consultas requeridas y algunas parametrizadas.
5. El sistema cuenta con un apartado FAQ donde se le darán respuesta a algunas preguntas técnicas que puedan tener los usuarios.
6. Disponibilidad: Existen tres tipos de usuarios (user, admin y superadmin), donde a cada uno se le garantiza el acceso a un determinado tipo de información.
7. Diseño e implementación: En el proyecto se emplea una arquitectura monolítica. Para la construcción del Backend, se usó .NET Core con C# y para el Frontend, se usó Angular. En el Backend, se usó una arquitectura de diseño basada en un dominio conocida como DDD. Para el trabajo con la base de datos se usó Entity Framework Core.

### Requisitos de entorno

1. Framework: .NET Core 7 y Angular.
2. Lenguaje de programación: C# y Typescript.

## 3 Funcionalidades del producto

Modelo de casos



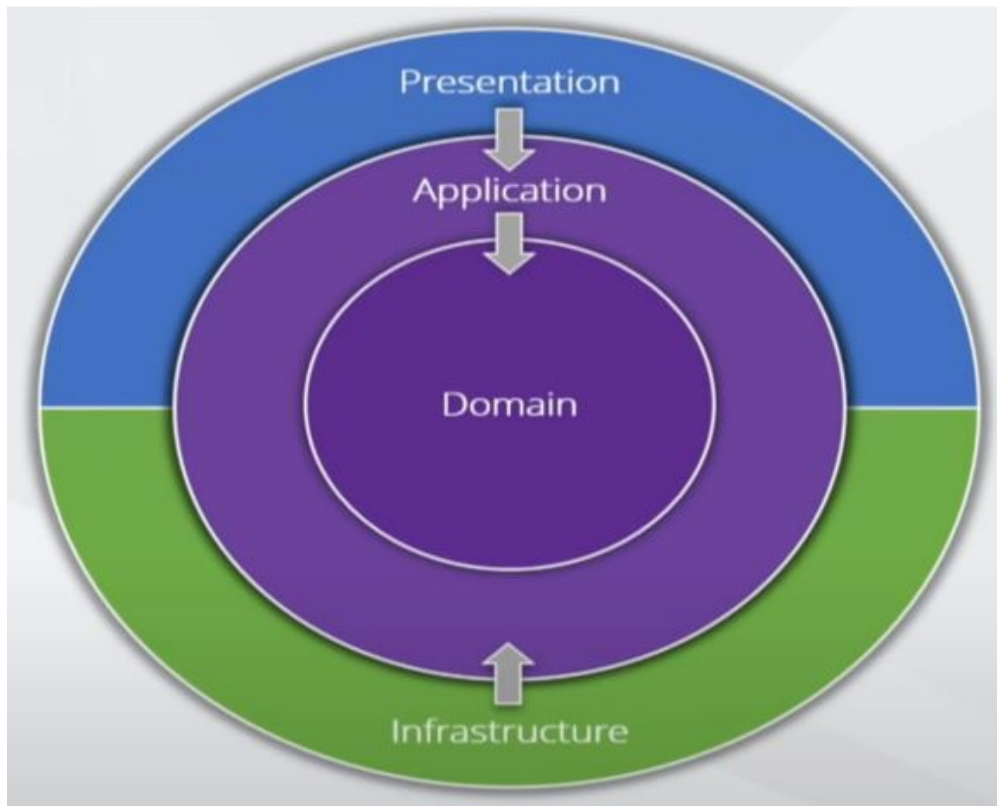
## 4 Enfoque Metodológico

Para la creación, se dividió el proyecto teniendo en cuenta las destrezas de los individuos y cada uno trabajó en un sector específico. El jefe de equipo repartió las tareas basándose en lo anterior.

## 5 Arquitectura

En el proyecto en general, se utiliza una arquitectura monolítica donde los componentes se integran en una sola unidad y son compatibles entre sí.

## Diseño de Backend



Se escogió esta arquitectura de diseño para tener el proyecto correctamente separado y garantizar las mejores prácticas de programación. Se emplea el patrón de diseño CQRS, donde se separan las responsabilidades en commands y queries.

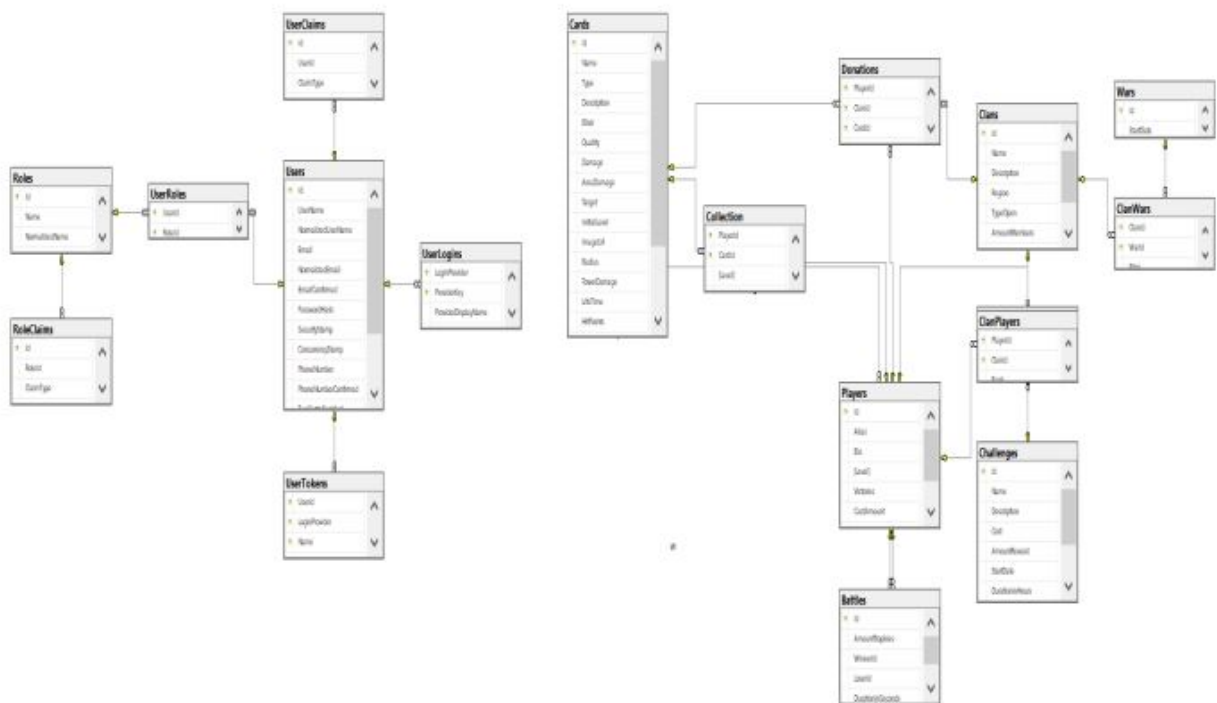
1. Domain: Entidades, excepciones, errores y posibles partes que no necesiten de ninguna otra capa.
2. Application: Interfaces y posibles objetos que interactúen con el Domain.
3. Infrastructure: Diferentes implementaciones de las interfaces de repositorio y todo lo relacionado con la base de datos.

## 6 Patrones de Visualización

Una gráfica de líneas muestra la popularidad del juego por mes (cantidad de batallas en el año). Mientras el proyecto avance se irán agregando más formas de visualización de datos.

## 7 Modelo de Datos

Base de datos:

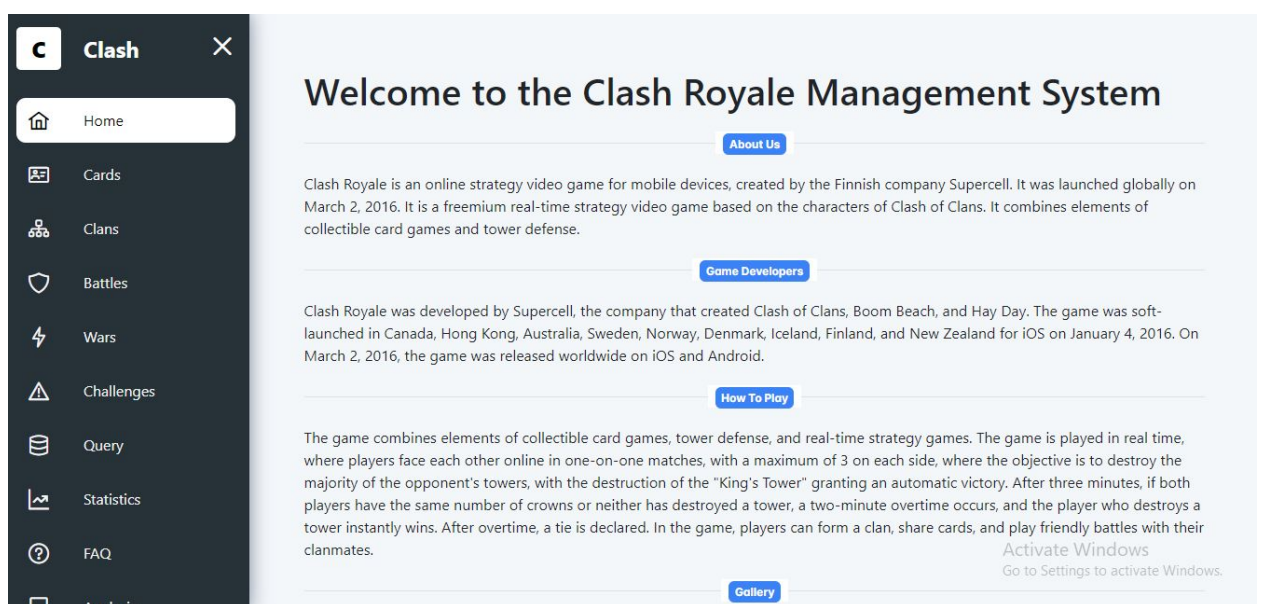


## 8 Diccionario de Datos

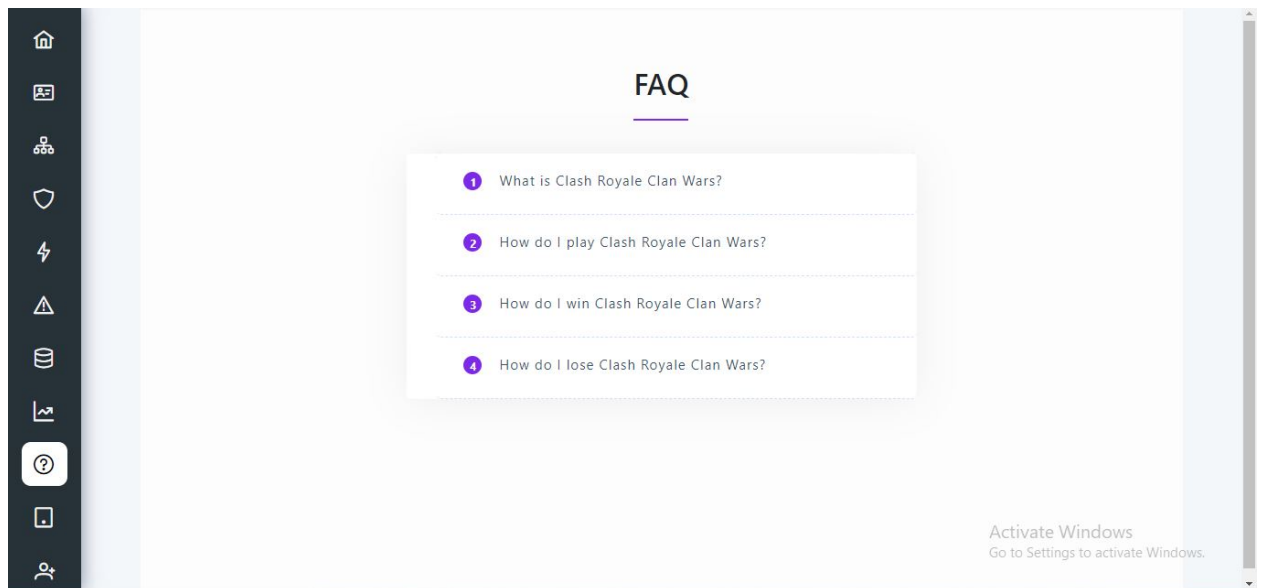
## 9 Esquema de las clases definidas

## 10 Opciones del sistema

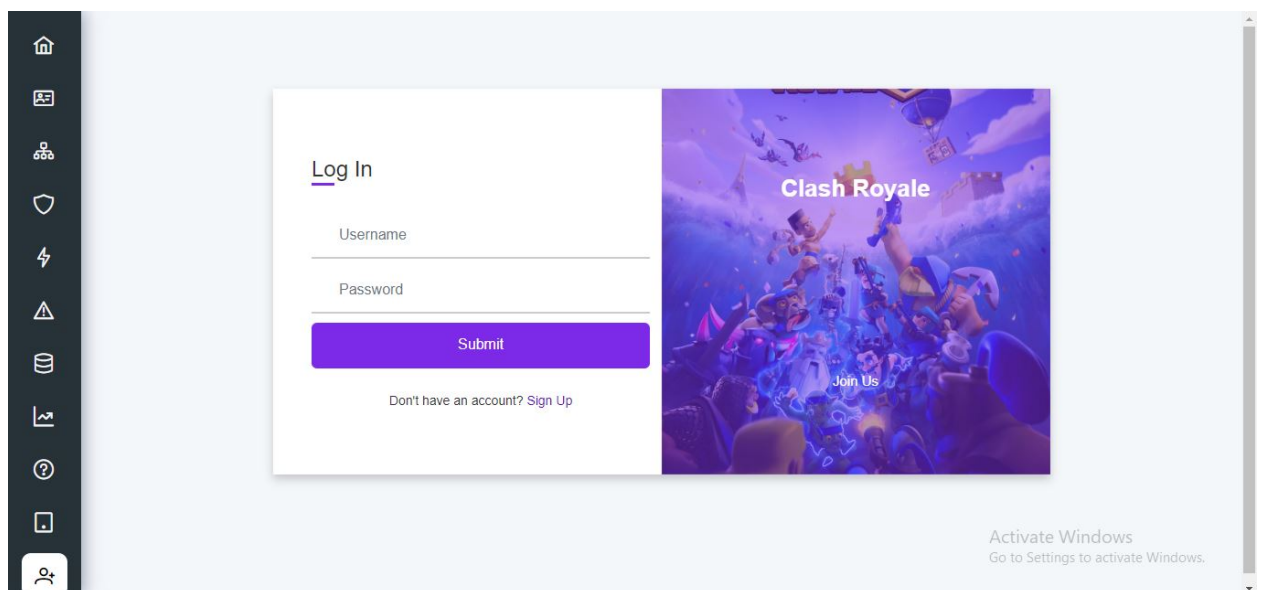
1. El sistema cuenta con una barra de navegación plegable a la izquierda de la pantalla

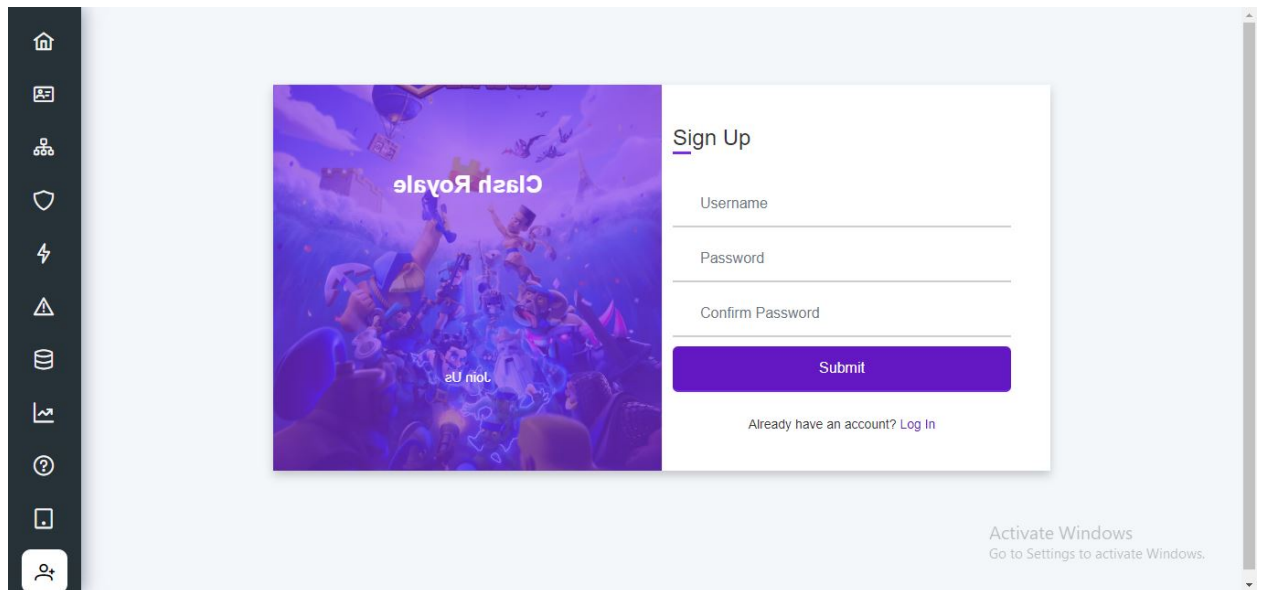


2. El sistema cuenta con un apartado de FAQ donde se le darán respuesta a algunas preguntas técnicas que puedan tener los usuarios.

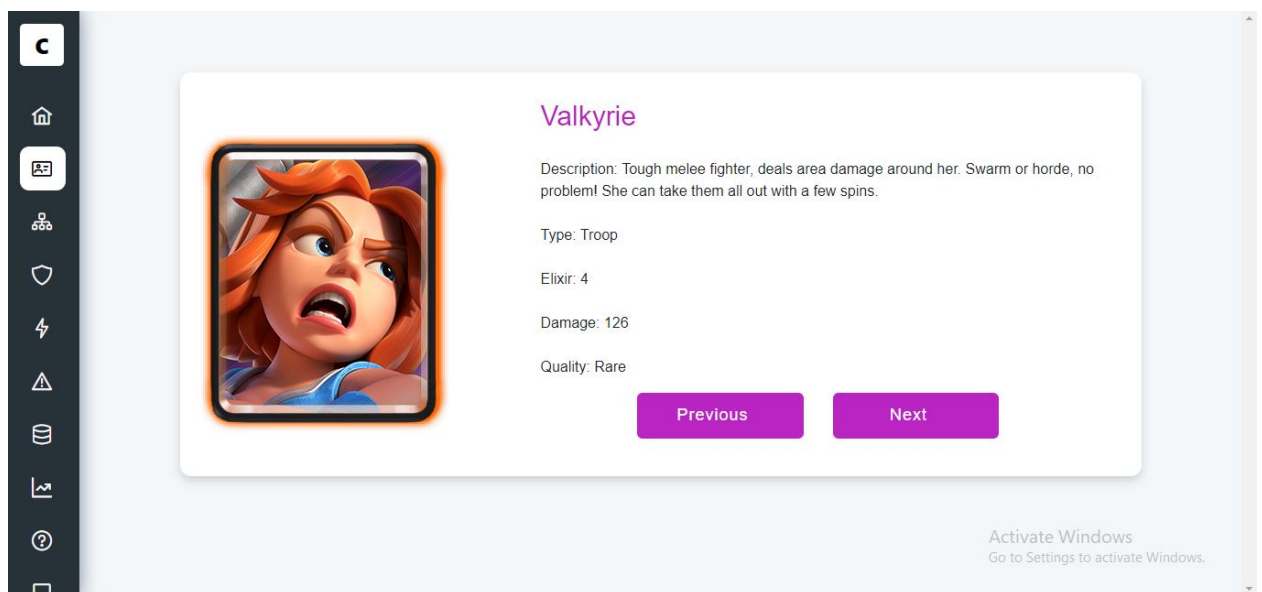


3. Los usuarios del sistema pueden autenticarse con su nombre de usuario y contraseña, o se pueden registrar

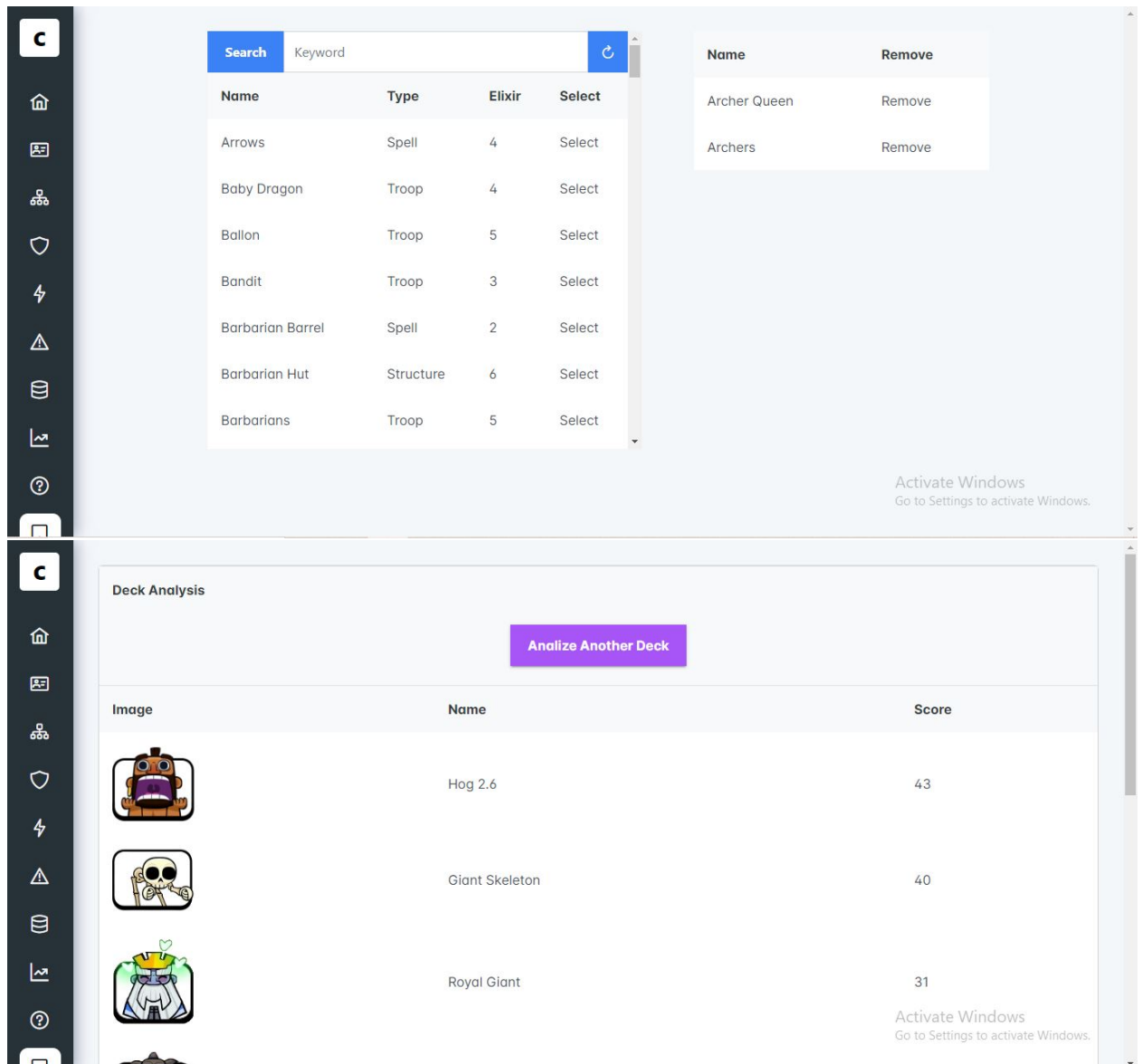




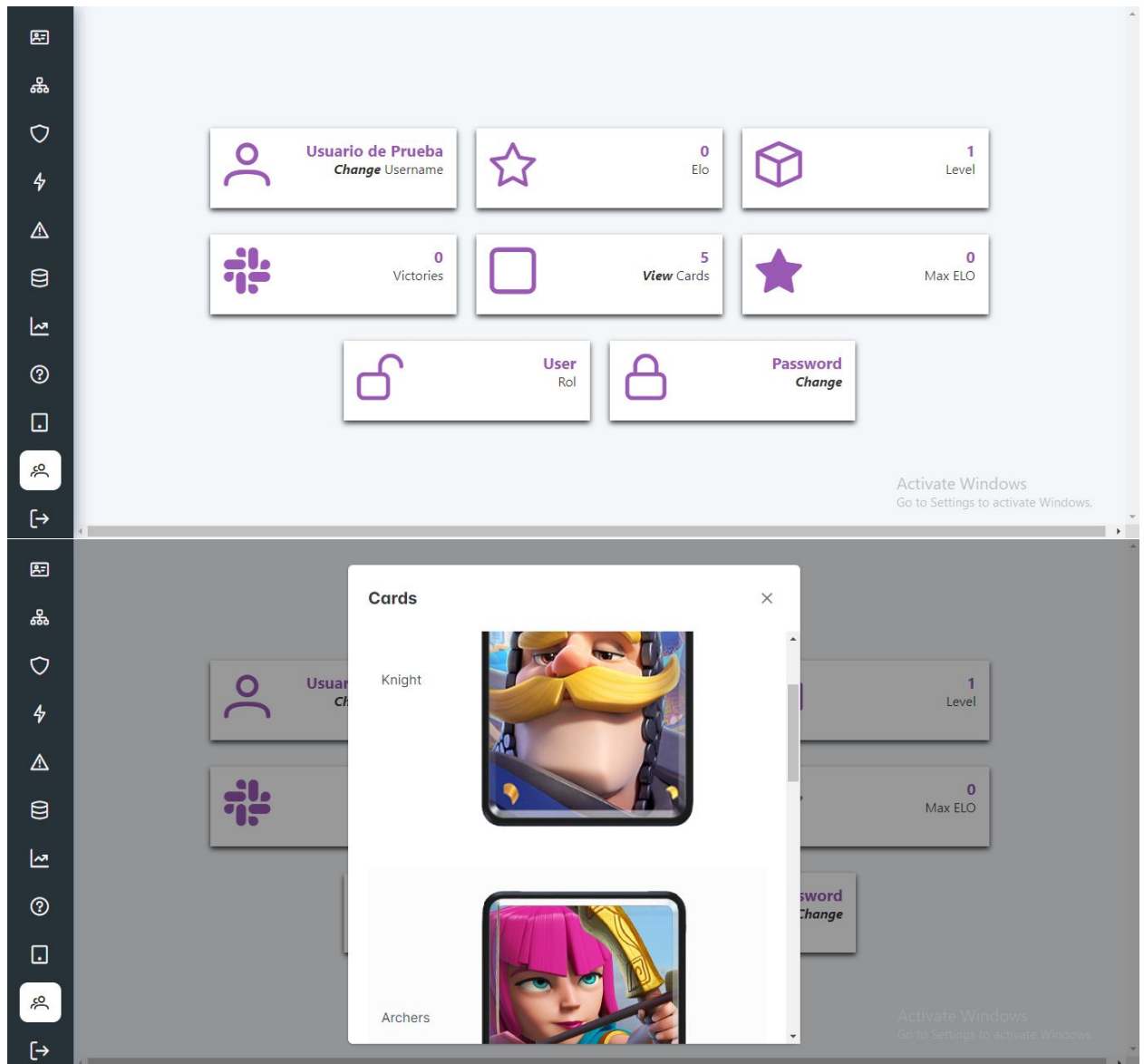
4. Los usuarios pueden visualizar las cartas con sus propiedades



5. Los usuarios pueden visualizar gráficamente la popularidad del juego; es decir, pueden ver gráficamente la cantidad de batallas en el año. Además, se pueden visualizar los 5 clanes más populares en una gráfica de pastel, y se puede visualizar en una gráfica de barras, la distribución de batallas ganadas por usuarios
6. Los usuarios pueden visualizar las tablas de Clanes, Batallas, Guerras y Desafíos. También puede visualizar los resultados de las 6 consultas que se realizan.
7. Un usuario puede realizar un análisis a un mazo de su elección y ver los resultados de la consulta.



8. Si el usuario está autenticado y su rol es 'User', entonces puede ver sus estadísticas como jugador, incluyendo las cartas de su mazo. Además puede cambiar su nombre de usuario y contraseña.



9. Si el usuario está autenticado y su rol es 'Admin' o 'Superadmin', puede realizar operaciones CRUD sobre las tablas, así como donar cartas, asignar cartas, agregar un jugador a un clan, agregar un clan a una guerra
10. Si el usuario está autenticado y su rol es 'Superadmin' puede cambiar los roles a los usuarios y cambiar sus contraseñas.

## 11 Salidas del sistema

1. Todas las tablas de la base de datos que se pueden visualizar (Jugadores, Clanes, Batallas, Guerras, Desafíos) se muestran usando el mismo componente, variando solamente las columnas que se muestran. Se tiene las funcionalidades de ordenar en orden ascendiente o descendiente por columnas y filtrar por una columna. También se cuenta con la funcionalidad de exportar los datos a pdf, csv y xlsx (se puede exportar todos los datos o los que se seleccionen manualmente). Los resultados de las consultas predeterminadas también se muestran en este mismo componente.





download (9).csv

	A	B	C	D	E	F	G	H
1	Name	Description	Region	Public	Member Cou	Trophies in v	Min Trophies	
2	The Boring R Window in C	SEA		false	10	64	2835	
3	The Roaming Whale in US	Dubai		false	7	2861	741	
4	The Boring F Truck in WEL	Canada		false	7	1761	3435	
5	The Boring R Truck in Dub	Dubai		false	5	936	218	
6	The Roaming Sniper in WE	WEU		true	4	674	1965	
7	The Blazing F Fighter in W	CIS		false	6	2810	1658	
8	The Gay Boat Fighter in Du	CIS		false	5	669	3006	
9	The Gay Dutch Dutchman in	LATAM		false	6	1351	2103	
10								
11								
12								

- Mostramos en una gráfica de líneas la popularidad del juego (cantidad de batallas por mes por año). Esta gráfica se puede exportar a pdf. FOTOOOOOOOOOOOOOOO
- Mostramos en una gráfica de barras la distribución de batallas ganadas por jugador (cuántos jugadores tienen x cantidad de batallas ganadas). Esta gráfica se puede exportar a pdf.
- Mostramos en una gráfica de pastel los 5 clanes más populares .Esta gráfica se puede exportar a pdf.

## 12 Modificaciones sobre el diseño de base de datos propuesto al inicio