

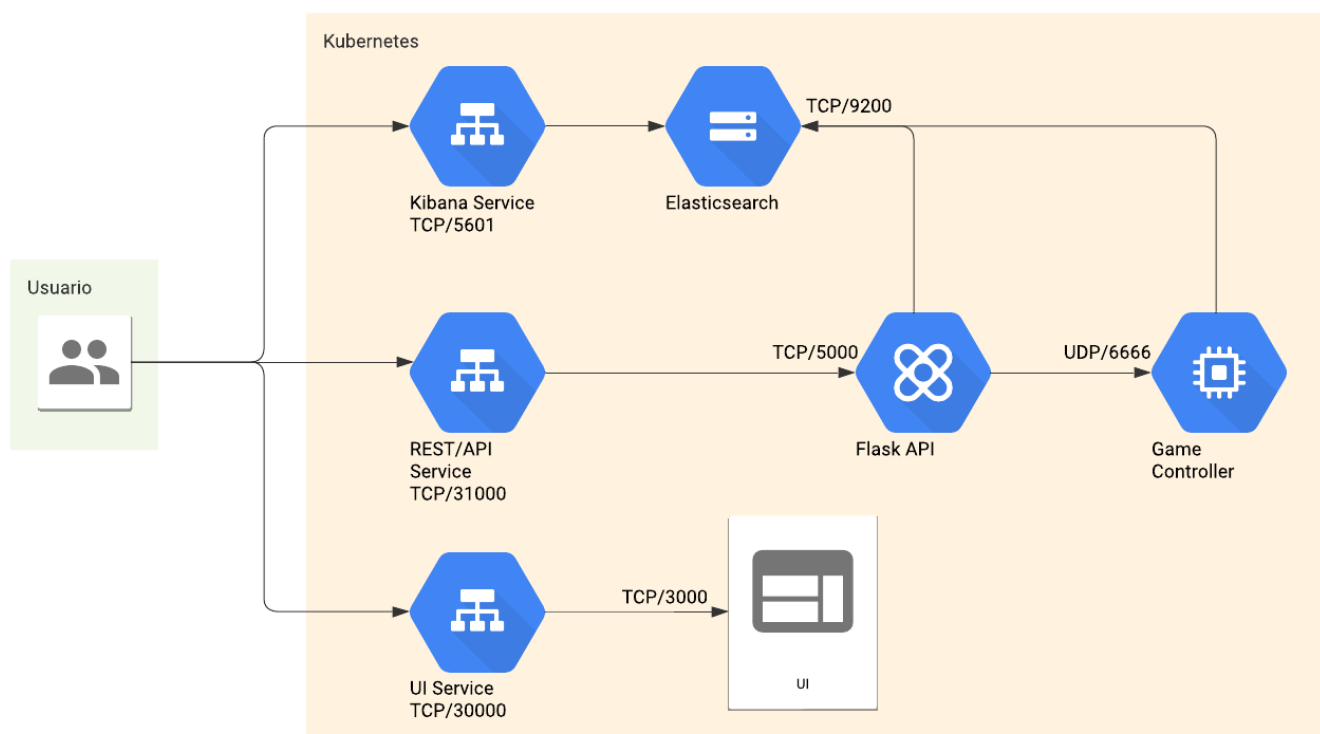
# Documentación - Crazy Chess

## Estudiantes:

- Ary-El Durán Ballesteró | 2018102445
- Isaac David Ortega Arguedas | 2018189196
- Zhong Jie Liu Guo | 2018319114

## Diagrama de Arquitectura

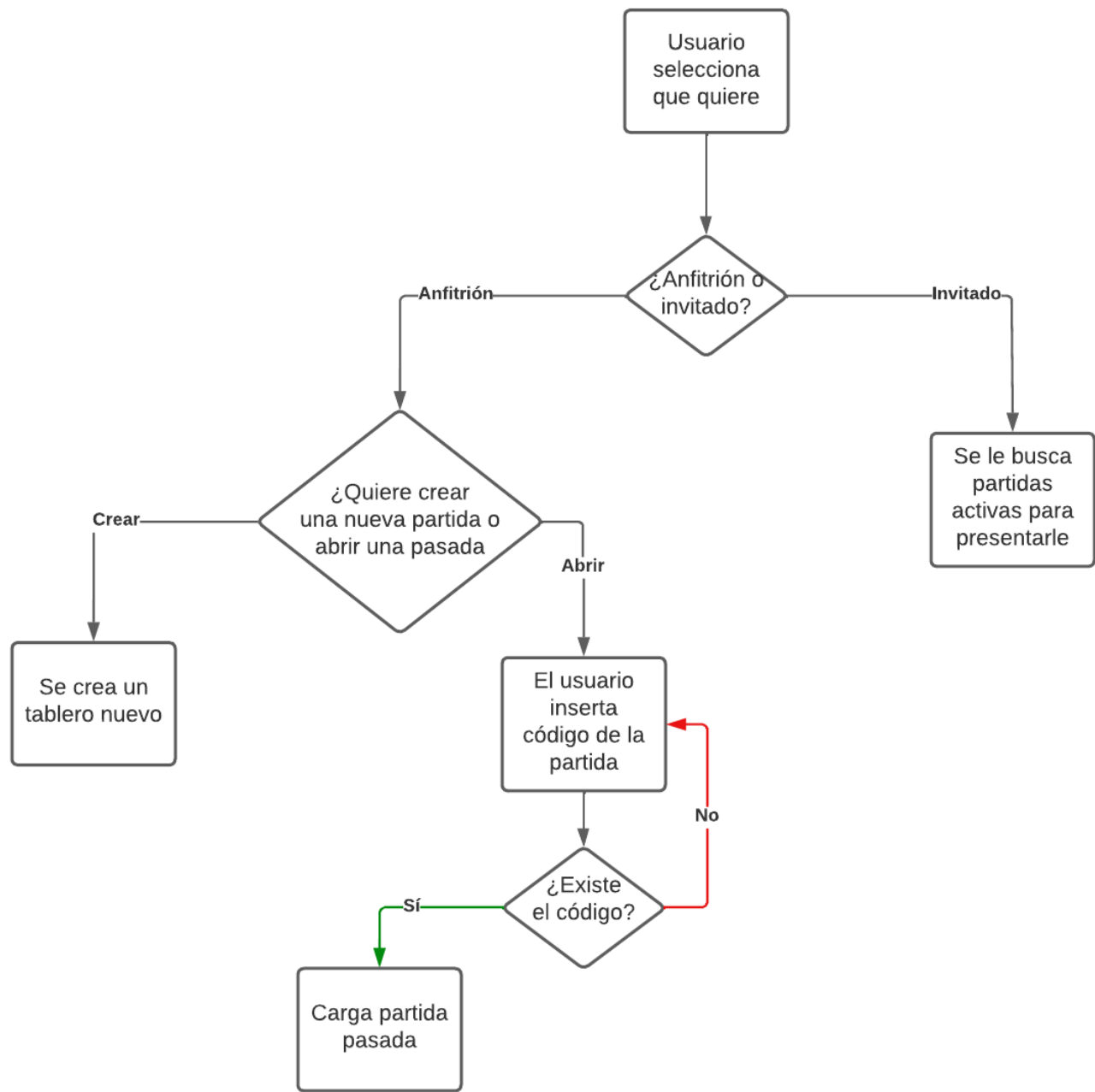
La base de este programa se tomó de la especificación del proyecto. Es bastante similar en el sentido de los puertos y las conexiones.



## Diagramas de Flujo

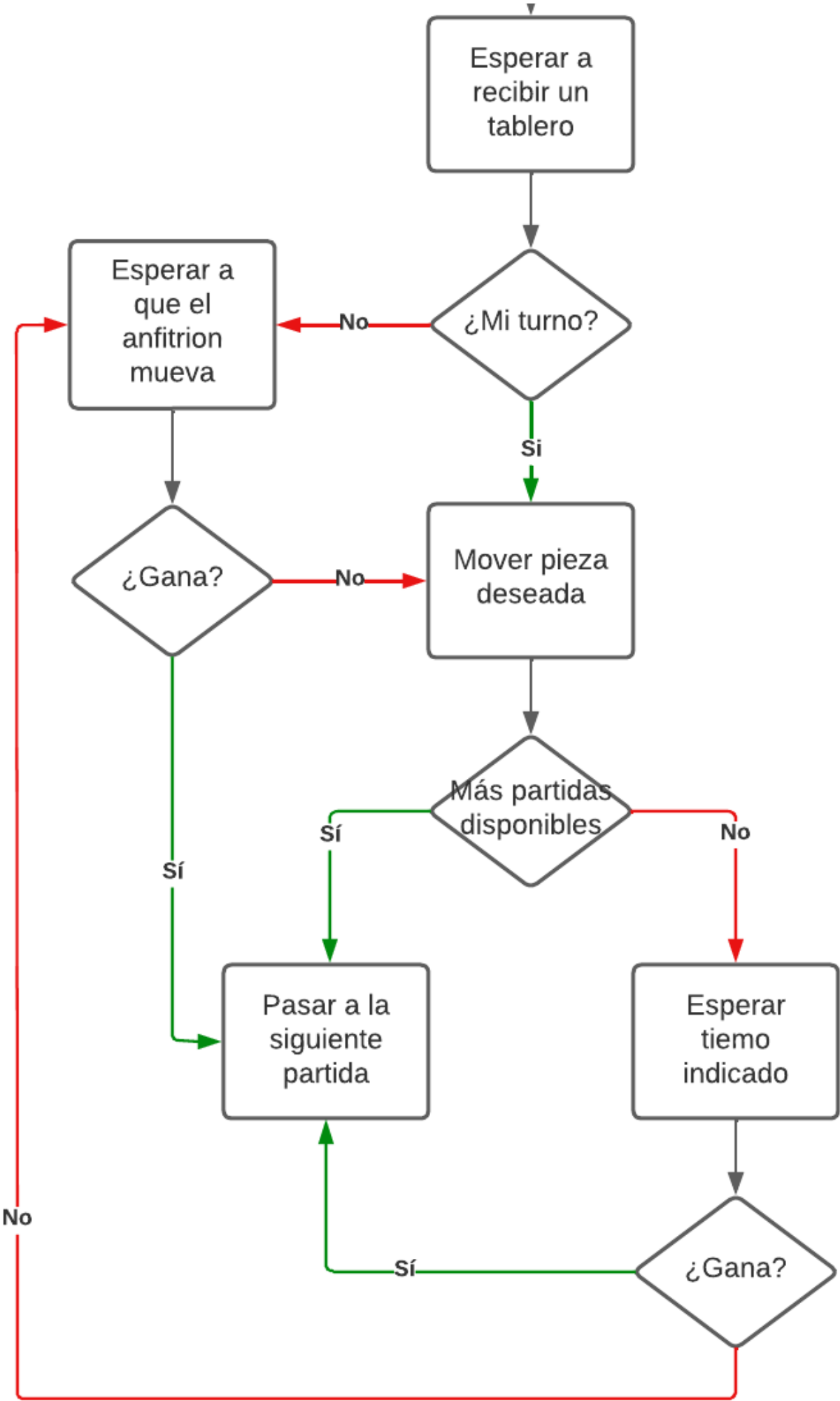
Para este apartado, se mostrarán tres diagramas para las tres historias de usuario: crear un juego, jugar como invitado y jugar como anfitrión.

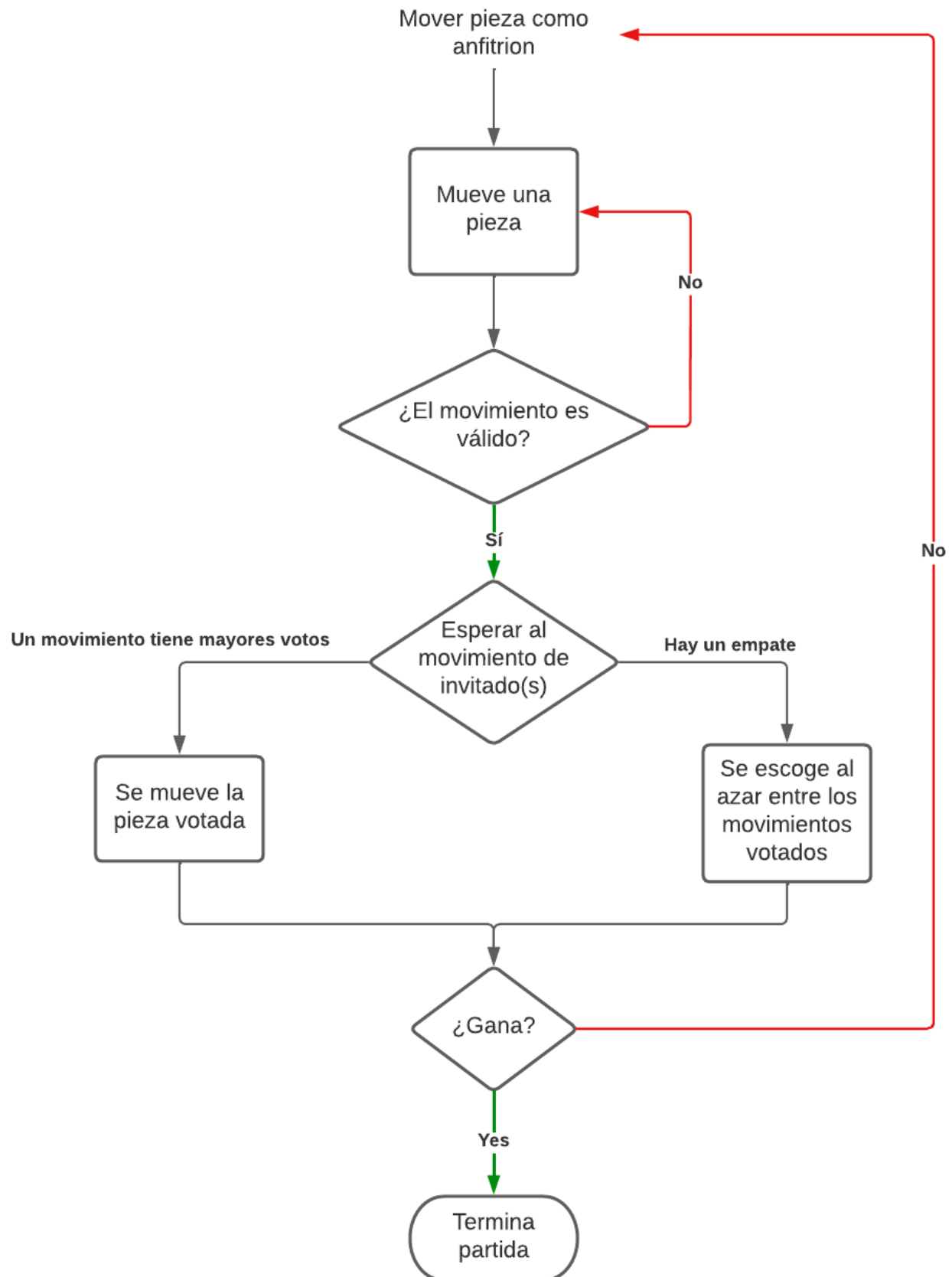
Crear o abrir partida



Jugar como invitado







## Prerequisitos para el proyecto

Para este proyecto, se asume que tiene instalado los siguientes programas:

- Docker Desktop(ver [link](#) )
- Kubernetes ([Habilitar Kubernetes desde Docker Desktop](#))
- Helm ([ver instalación](#))
- kubectl (Si no se instala por defecto, vea [link](#))
- (Opcional) Lens (ver [link](#))

## Scripts para comenzar a usarlo

Los comandos se pueden dividir en dos partes: la creación de las imágenes y la corrida en Kubernetes. El primero usará el docker-compose que ayudará a construir las imágenes necesarios de las aplicaciones. Con el programa Lens, puede revisar de forma visual el progreso de la instalación.

## Instalación de Elasticsearch y Kibana

Se va a instalar de primero Elasticsearch y Kibana ya que se ocupan las credenciales de acceso para su uso. Se usará como base la guía [Quickstart](#) de Elastic Cloud en Kubernetes, pero se van a mostrar los comandos necesarios. Para una explicación más detallada, vea el link anterior. Los siguientes comandos son para una primera instalación.

```
kubectl create -f https://download.elastic.co/downloads/eck/2.4.0/crds.yaml kubectl  
apply -f https://download.elastic.co/downloads/eck/2.4.0/operator.yaml
```

Luego, se usarán los archivos ubicados en la carpeta **db**, elastic.yaml y kibana.yaml. En una terminal, ubíquese en la carpeta **db**, luego haga pull a las imágenes si se requieren.

```
docker pull docker.elastic.co/elasticsearch/elasticsearch:8.4.0  
docker pull docker.elastic.co/kibana/kibana:8.4.0
```

Cuando las descargas terminen, corra los siguientes comandos y espere un tiempo para que los servidores terminen de levantarse.

```
kubectl apply -f elastic.yaml  
kubectl apply -f kibana.yaml
```

Para revisar si ha terminado, ejecute `kubectl get elastic` y `kubectl get kibana`. Si todo sale bien, aparecerá la salud (HEALTH) en verde.

Para acceder a kibana, se debe hacer port-forwarding con `kubectl` o hacerlo por medio de Lens.

```
kubectl port-forward service/quickstart-kb-http 5601
```

Luego, va a aparecer un log-in, el usuario es **elastic** y la contraseña puede averiguarse corriendo el siguiente comando: `kubectl get secret quickstart-es-elastic-user -o=jsonpath='{.data.elastic}' | base64 --decode; echo`. Si está en Windows, se recomienda correrlo en una terminal de WSL o de Unix.

Si quiere detener los pods, vaya a la carpeta **db** y corra los siguientes comandos.

```
docker delete -f elastic.yaml docker delete -f kibana.yaml
```

## Creación de imágenes

Ahora, hay que crear las imágenes para que se ejecuten en los pods. La ruta de la terminal tiene que ser en la carpeta **ProyectoOpcional**. Luego, se construirán utilizando el comando `docker-compose build <nombre-imagen>`. Va a tomar algunos minutos para descargar las dependencias.

```
cd <a la ruta del proyectoOpcional>
docker-compose build api
docker-compose build frontend
docker-compose build game-controller
```

Si necesita eliminar las imágenes creadas y todo lo relacionado a ello se usa `docker-compose down --rm all`. Si hizo un cambio en el código, simplemente ingrese `docker-compose build <nombre-imagen>` de la imagen modificada.

## Levantar los pods en Kubernetes

Ahora, se van a levantar los pods en Kubernetes con las imágenes apropiadas utilizando helm charts. Primero hay que dirigirse a la carpeta **app-deployment**. Luego se utiliza el siguiente comando para montar la arquitectura en el Kubernete.

```
## Crear los pods
helm install app-deployment .
## Detener y eliminar los pods
helm delete app-deployment
```

Lo anterior va a levantar múltiples pods y servicios relacionados al proyecto. Espere unos segundos y para utilizar CrazyChess, abra su navegador y escriba `http://localhost:30000/` para abrir la UI. Si desea ver la API, la url es la siguiente: `http://localhost:31000/` Si todo salió bien, le aparecerá el menú principal de la aplicación.

## Recomendaciones

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.
- 7.
- 8.
- 9.
- 10.

## Conclusiones

- 1.
- 2.
- 3.
- 4.
- 5.

- 6.
- 7.
- 8.
- 9.
- 10.