CICLO FORMATIVO DE GRADO SUPERIOR

DESARROLLO DE APLICACIONES WEB

MEMORIA

**SPECTRUM**

Proyecto Final

Autor: Máximo Díaz-Cano Sánchez

Fecha: 20/05/2022

ÍNDICE

1 - Introducción 1

Descripción de la aplicación 1

Objetivos y motivación 1

2 - Análisis 2

Requisitos funcionales 2

Requisitos no funcionales 2

3 - Planificación 3

Sprint 0 3

Sprint 1 3

Sprint 2 3

Sprint 3 3

Sprint 4 3

4 - Diseño 4

Diagramas de clases y componentes 4

Arquitectura del sistema 4

Interfaz gráfica 4

Diseño de la base de datos 5

Entorno de desarrollo 6

Servicios y librerías 6

5 - Pruebas 8

Plan de pruebas 8

Informe de pruebas 8

6 - Conclusiones 9

Conocimientos adquiridos 9

Mejoras futuras 9

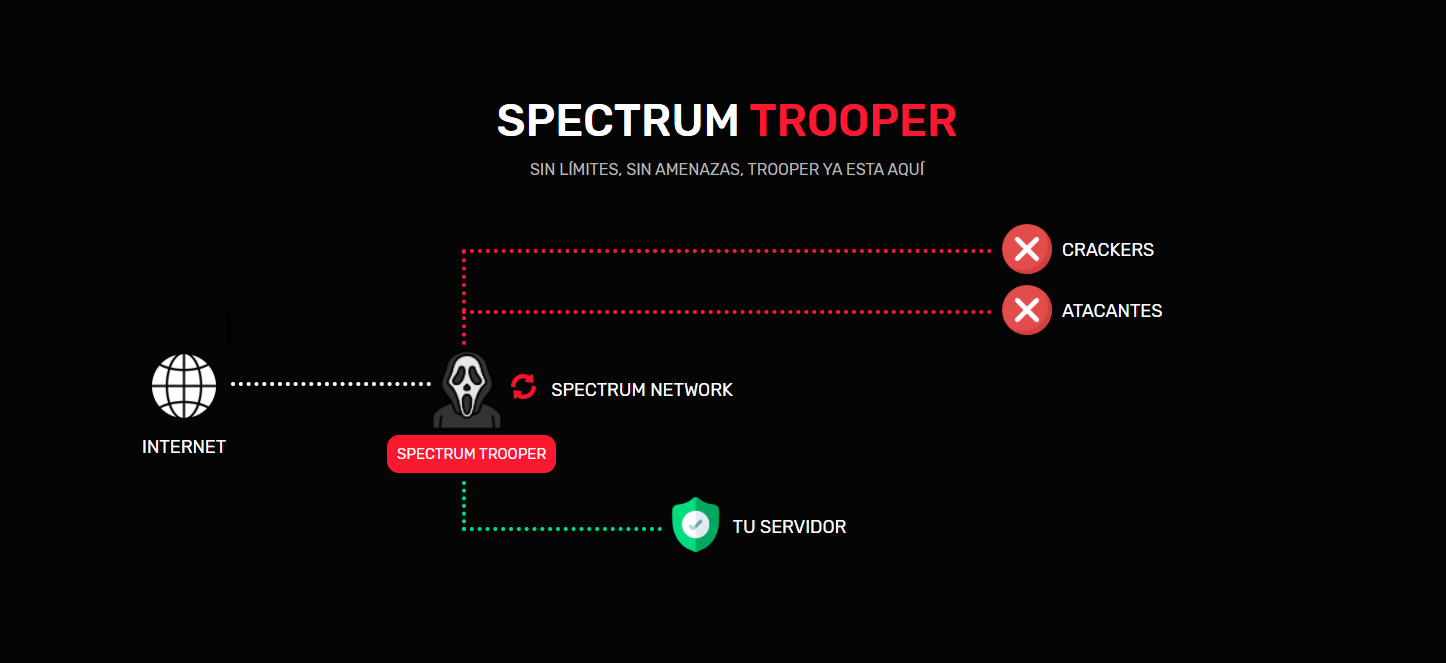
# Introducción

## Descripción de la aplicación

**Spectrum** es un sistema Anti-DDoS que te ayuda a ocultar y proteger tus servidores, manteniéndolos sin ataques, sin lag y sin caídas actuando como proxy inverso tcp/ip.

Un cliente se conecta a **Spectrum** por un puerto, este, copia y envía su buffer de paquetes al servidor de destino, lo mismo ocurre de forma inversa, copia y envía los paquetes del servidor como respuesta al cliente.

Imagen explicativa creada por mi sobre cómo funciona **Spectrum**.



## Objetivos y motivación

El objetivo de este proyecto es detectar y mitigar cualquier ataque a un servidor cliente, sin que este note el ataque, y poder monitorizarlo desde su respectivo panel y ajustar sus filtros Anti-DDoS para ser más contundente.

# Análisis

Descripción de los requisitos funcionales y no funcionales.

## Requisitos funcionales

Descripción de los requisitos funcionales (puedes utilizar diagramas de casos de uso).

* **Licencias**: Cada cliente tiene asignada una licencia para su proxy Spectrum, cada licencia consta de una Spectrum-Key (clave interna) y una Client-Key (clave externa), el cliente nunca tiene acceso a la Spectrum-Key, ya que es la que se utiliza para gestionar de forma interna la comunicación de Spectrum con la API.

* **Spectrum WEB**: es el nombre con el que se conoce al panel web que utiliza Spectrum, desarrollado en PHP (conectado a la base de datos) y JavaScript para su interfaz junto a su diseño HTML5, en el cual se puede ver las estadísticas de cada proxy de cada cliente, los cuales deben identificarse utilizando el nombre usado en su sitio y su Client-Key. En Spetsnaz puedes ver y gestionar los baneos y otros ajustes del firewall de Spectrum y demás configuraciones en desarrollo.

## Requisitos no funcionales

Todos aquellos requisitos que no estén directamente relacionados con la funcionalidad de la aplicación: de seguridad, de rendimiento, de memoria, relacionados con la versión del S.O. o con la resolución de las pantallas, etc.

* **Spectrum API**: En un dominio externo en un hosting comunicado con una base de datos se utiliza una API propia desarrollada en PHP 7.1, la cual permite ir revocando datos y estadísticas en tiempo real que llegan de cada Spectrum, los cuales se identifican con la Spectrum-Key que tiene asignado.
* **Spectrum**: el propio proxy tcp/ip en sí, desarrollado en C# por el que pasan todas las conexiones y obtiene sus estadísticas.

# Planificación

A continuación se detallan los sprints del proyecto.

## Sprint 0

En este primer sprint se hará un análisis de requerimientos necesarios para el proyecto. Se necesita planificar que tareas se realizarán en los siguientes sprints y documentarse acerca de la tecnología del proyecto y de su contexto.

## Sprint 1

El objetivo de este sprint es crear la base de datos y estudiar modificaciones.

## Sprint 2

En este tercer sprint se desarrolla el panel, lo que es el frontend.

## Sprint 3

En el cuarto sprint se desarrolla el panel web con el backend.

## Sprint 4

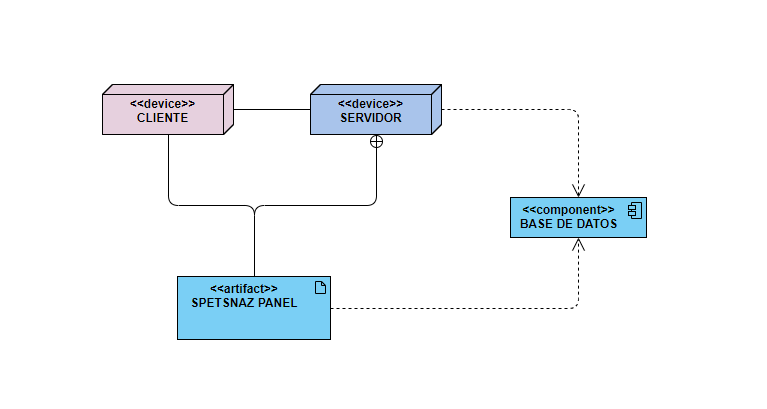
Pruebas, despliegue y documentación.

# Diseño

## Diagramas de clases y componentes

## Arquitectura del sistema

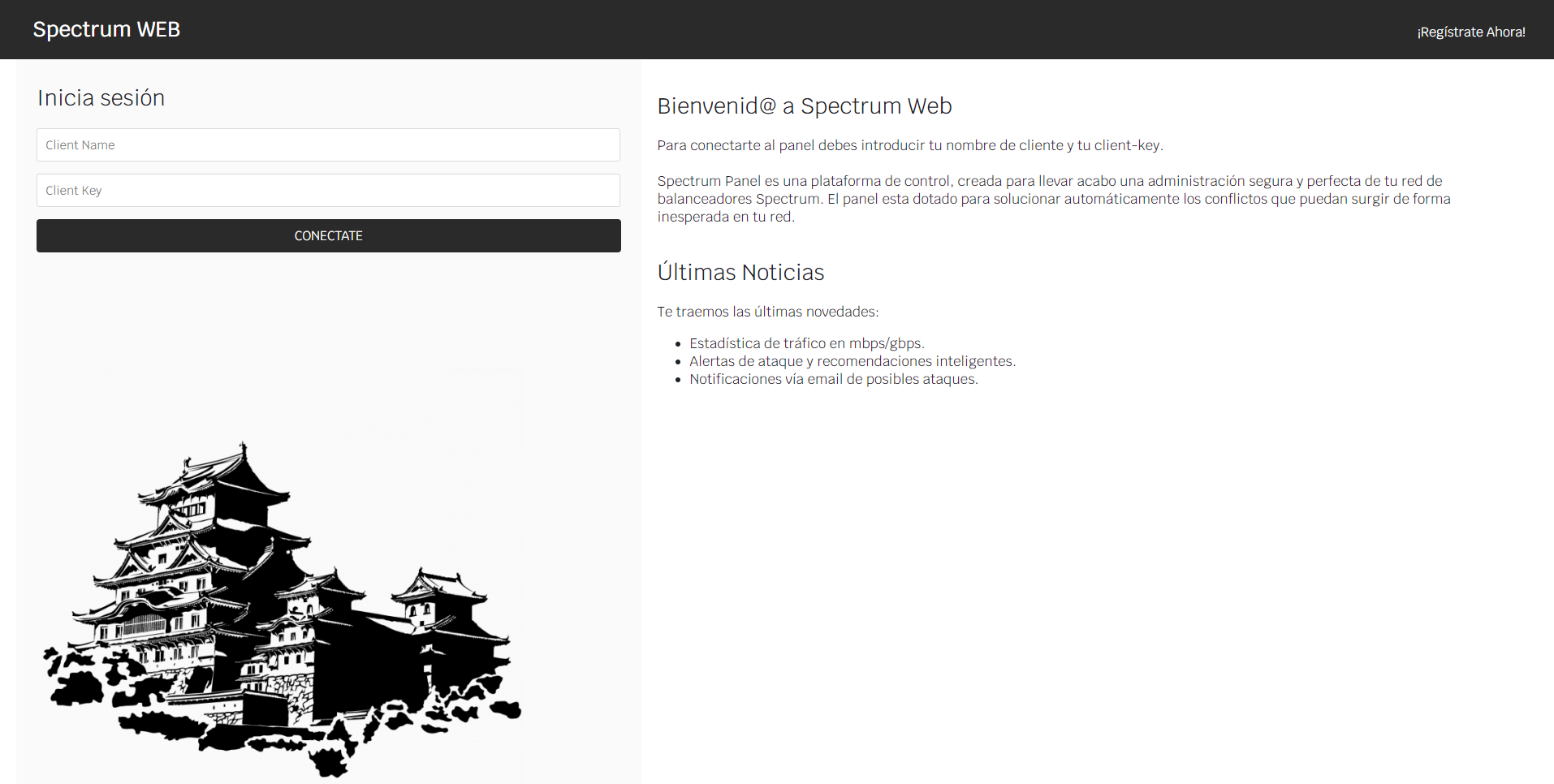
Describir la arquitectura del sistema (diagrama de despliegue)



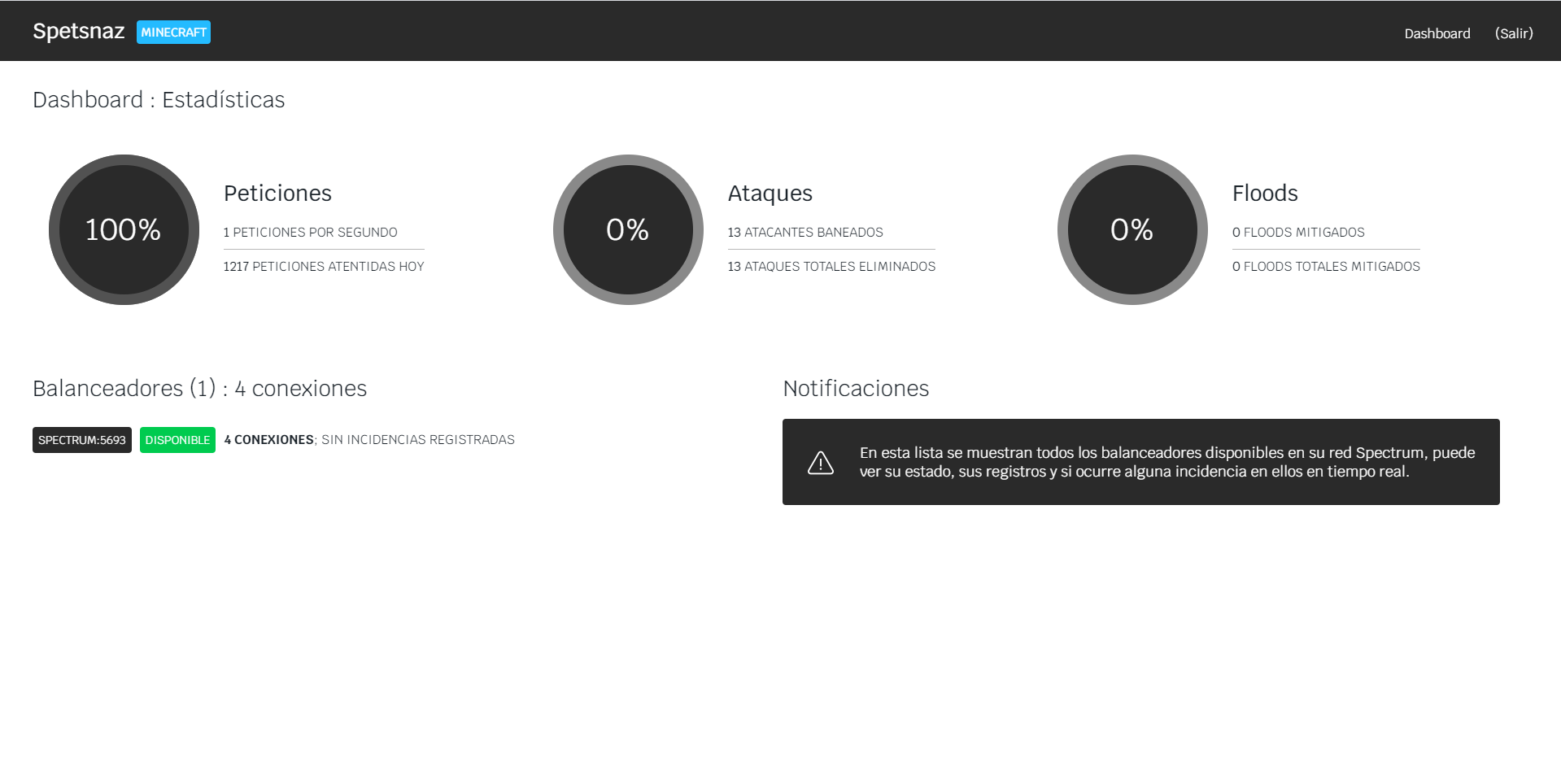
## Interfaz gráfica

Incluir aquí el diseño de las pantallas, su descripción y la navegación entre ellas.

Index:

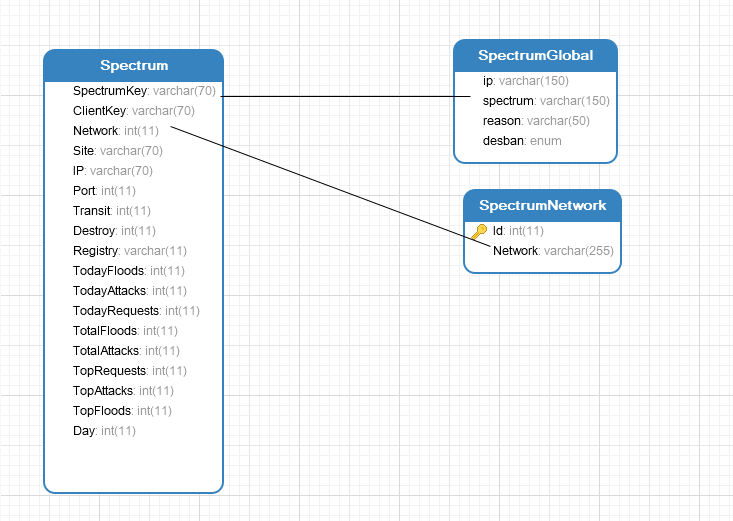


Dashboard:



## Diseño de la base de datos

Descripción de la base de datos local y del backend. Se recomienda utilizar diagramas de clases, así como la descripción de las diferentes relaciones y sus atributos.



**Spectrum : Tabla**

* **SpectrumKey**: Licencia de comunicación interna.
* **ClientKey**: Licencia de comunicación externa.
* **Network**: Número de red a la que pertenece (por si hay varios servidores distintos).
* **Site**: Nombre del cliente al que protege.
* **IP**: IP de destino a la que enviará el tráfico.
* **Port**: Puerto de destino al que enviará el tráfico.
* **Transit**: Media de conexiones que puede recibir (para aplicar filtros de limitaciones).
* **Destroy**: En caso de ser robado, si se pone un 1 se autodestruye.
* **TodayFloods**: Ataques por flood recibidos hoy.
* **TodayAttacks**: Ataques por otro motivo recibidos hoy.
* **TodayRequets**: Peticiones recibidas hoy.
* **TotalFloods**: Floods totales mitigados.
* **TotalAttacks**: Ataques totales mitigados.
* **TopRequests**: Top de conexiones alcanzadas hasta el día de hoy.
* **TopFloods**: Top de floods hasta el día de hoy.
* **TopAttacks**: Top de ataques hasta el día de hoy.
* **Day**: Controla el día de la última actualización para reiniciar estadísticas temporales (TodayX).

**SpectrumGlobal : Tabla**

* **IP**: IP del objetivo baneado.
* **Spectrum**: SpectrumKey del cliente al que pertenece este baneo.
* **Reason**: Motivo del baneo.
* **Desban**: Si se pone 1 uno, se manda la solicitud para desbanearlo del sistema.

**SpectrumNetwork : Tabla**

* **Id**: Identificador de la red.
* **Network**: IP de la red que asocia.

## Entorno de desarrollo

## Servicios y librerías

En este punto se explicarán las diferentes tecnologías usadas para realización del proyecto así como los elementos más importantes para entender el sistema.

**Spetsnaz (Spectrum Panel Web)**

* PHP 7.0 Laravel, MariaDB, Javascript, HTML5 y CSS3.
* Entorno de desarrollo: Visual Studio Code

# Pruebas

## Plan de pruebas

Ya mostrado en videoconferencia al tutor asignado.

## Informe de pruebas

Funciona correctamente como se esperaba.

# Conclusiones

## Conocimientos adquiridos

Mejoras de capacidad en desarrollo web.

## Mejoras futuras

Implementación completa de baneos y tipos de ataques.