Universidad Latina De Costa Rica

Facultad de Ingenierías, Tecnologías de la Información y Comunicación

BIT 28 - Sistemas Operativos II

Proyecto Final

Portal de Tutorías y Soporte Académico

Elaborado Por:
Melissa Bermúdez Araya

Docente:
Carlos Andres Mendez Rodriguez

Fecha de Entrega:
Viernes 25 de Abril de 2025

Tabla de Contenidos

Resumen

El presente artículo propone el desarrollo y creación de un portal de tutorías y ayuda escolar para la comunidad de estudiantes de la Universidad Latina; esta herramienta, con una infraestructura tecnológica que asegura alta disponibilidad y buen rendimiento mediante el uso de balanceo de carga con HAProxy. La solución está enfocada a responder a una demanda creciente de asistencia académica, y busca garantizar que siga funcionando sin problemas, incluso ante grandes volúmenes de tráfico o fallos en los servidores. Se añadirán métodos para distribuir como Round-robin y Least Connections, formas para comprobar el estado (chequeos de salud) como también protecciones contra ataques grandes (DDoS) y un límite en el número de conexiones.

Introducción

Hoy en día, muchas instituciones educativas necesitan contar con plataformas tecnológicas que funcionen bien y estén disponibles todo el tiempo. Con clases virtuales, tareas en línea y tutorías personalizadas, es súper importante que los estudiantes puedan acceder fácil y rápido a la ayuda que necesitan. Pero cuando muchas personas usan el sistema al mismo tiempo, si no está bien preparado, puede fallar o volverse muy lento.

Por eso, nuestro proyecto propone crear un Portal de Tutorías y Soporte Académico, una página web donde los estudiantes puedan agendar tutorías, revisar materiales de estudio y comunicarse con sus tutores de forma sencilla. Lo más importante no es solo que funcione bien por fuera, sino que por dentro esté pensado para aguantar mucho tráfico, ser seguro y fácil de mantener.

Pensamos especialmente en los momentos más cargados del curso, como las semanas de exámenes, donde todos necesitan más apoyo. Si el sistema se cae justo ahí, se vuelve un problema. Para evitarlo, usamos HAProxy, que es una herramienta que reparte las conexiones entre varios servidores, así no se sobrecarga ninguno.

Además, hicimos un script (una especie de programa automático) que instala todo en Linux sin tener que hacerlo a mano, lo que ahorra tiempo y reduce errores. Y también incluimos certificados SSL de Let's Encrypt, que permiten que todo se cargue con HTTPS, es decir, de forma segura.

Con este proyecto, no solo ayudamos a mejorar cómo funciona la parte tecnológica en la educación, sino que también aprendimos a usar herramientas reales que se usan en empresas, lo cual es genial si queremos seguir estudiando o trabajando en el área de tecnología en el futuro.

Objetivos

Objetivo General

- Diseñar e implementar un portal web de tutorías y soporte académico para la Universidad Latina, asegurando alta disponibilidad y rendimiento eficiente.

Objetivos Específicos

- Desarrollar un portal web funcional que permita la gestión de tutorías académicas, incluyendo reservas, chat y acceso a recursos.
- Evaluar el rendimiento, disponibilidad y seguridad del sistema mediante pruebas controladas de carga y recuperación ante fallos.
- Documentar la arquitectura y buenas prácticas para su implementación futura en el entorno institucional.

Materiales y Métodos

Materiales utilizados

Para construir y desplegar el Portal de Tutorías y Soporte Académico, se utilizaron herramientas de software libre reconocidas por su fiabilidad en entornos reales de producción. La elección de cada una se enfocó en cumplir con los principios de alta disponibilidad, escalabilidad, seguridad y automatización.

Principales tecnologías:

- Sistema operativo: Ubuntu Server 20.04 LTS, ejecutado en una máquina virtual.
- Lenguaje de programación: Python 3.10.
- Framework del backend: Flask, por su ligereza y facilidad para crear APIs REST.
- Servidor WSGI: Gunicorn, encargado de gestionar los procesos de Flask.
- Servidor web y proxy inverso: NGINX, para servir el contenido y canalizar las peticiones al backend.
- Balanceador de carga: HAProxy, configurado para distribuir tráfico y facilitar la escalabilidad.
- Certificación SSL: Certbot junto con Let's Encrypt, para ofrecer una conexión segura mediante HTTPS.
- Base de datos: SQLite, seleccionada por su simplicidad y utilidad en entornos de prueba o bajo consumo.
- Control de versiones: Git y GitHub, para llevar un registro claro de cada cambio realizado.
- Automatización del despliegue: Un script en Bash que agiliza todo el proceso de instalación y configuración.

Metodología

1. Diseño del Backend: Se definió una arquitectura RESTful con Flask, capaz de gestionar usuarios (estudiantes y tutores) y sus sesiones de tutoría.
2. Base de Datos: SQLite fue integrada y configurada para ser inicializada automáticamente al ejecutar la aplicación, facilitando la portabilidad del sistema.
3. Despliegue Automatizado: Se desarrolló un único script en Bash que realiza todos los pasos necesarios: instalación de dependencias, configuración de Gunicorn, NGINX, HAProxy, habilitación de SSL y arranque automático.

4. Alta Disponibilidad y Balanceo de Carga: Se integró HAProxy para permitir balanceo round-robin entre instancias (en este caso locales), con posibilidad de escalar horizontalmente.
5. Seguridad: Se utilizó Certbot con Let's Encrypt para asegurar el portal con HTTPS, protegiendo la integridad y confidencialidad de los datos en tránsito.
6. Versionamiento y documentación: Se usó Git para el control de cambios y para documentar la evolución del proyecto, permitiendo revertir o auditar cada configuración realizada.

Script del Servicio Web

```
#!/bin/bash
# Script de despliegue del Portal de Tutorías y Soporte Académico
# Autor: Melissa Bermúdez
# Curso: Alta Disponibilidad y Balanceo de Carga

set -e

# 1. Actualización del sistema
sudo apt update && sudo apt upgrade -y

# 2. Instalación de dependencias
sudo apt install -y python3 python3-pip python3-venv git nginx haproxy certbot python3-certbot-nginx

# 3. Clonar el repositorio del portal
sudo git clone https://github.com/usuario/portal-tutorias.git /opt/tutorias
cd /opt/tutorias

# 4. Crear entorno virtual e instalar dependencias
python3 -m venv venv
source venv/bin/activate
pip install -r requirements.txt

# 5. Configurar Gunicorn como servicio
cat <<EOF | sudo tee /etc/systemd/system/tutorias.service
[Unit]
Description=Portal de Tutorías
After=network.target
```

```
[Service]
User=www-data
WorkingDirectory=/opt/tutorias
ExecStart=/opt/tutorias/venv/bin/gunicorn -w 4 -b 127.0.0.1:5000 app:app
Restart=always

[Install]
WantedBy=multi-user.target
EOF

sudo systemctl daemon-reexec
sudo systemctl enable tutorias
sudo systemctl start tutorias

# 6. Configurar HAProxy
cat <<EOF | sudo tee /etc/haproxy/haproxy.cfg
global
    log /dev/log local0
    maxconn 4096

defaults
    log     global
    mode    http
    timeout connect 5000ms
    timeout client  50000ms
    timeout server  50000ms

frontend http-in
    bind *:8080
    default_backend app_servers

backend app_servers
    balance roundrobin
    server app1 127.0.0.1:5000 check
EOF

sudo systemctl restart haproxy

# 7. Configurar NGINX como proxy con HTTPS
```

```
cat <<EOF | sudo tee /etc/nginx/sites-available/tutorias
server {
    listen 80;
    server_name 190.113.110.160;

    location / {
        proxy_pass http://localhost:8080;
        proxy_set_header Host \$host;
        proxy_set_header X-Real-IP \$remote_addr;
    }
}
EOF

sudo ln -s /etc/nginx/sites-available/tutorias /etc/nginx/sites-enabled/
sudo nginx -t && sudo systemctl reload nginx

# 8. Obtener certificado SSL de Let's Encrypt (esto puede fallar con IP pública)
sudo certbot --nginx -d 190.113.110.160 --non-interactive --agree-tos -m melissa.bermudez1@ulatina.net || echo "⚠️ No se pudo emitir certificado para IP. Considera usar un dominio."

# 9. Redireccionar HTTP a HTTPS
sudo sed -i '/listen 80;/a return 301 https://\$host\$request_uri;' /etc/nginx/sites-available/tutorias
sudo systemctl reload nginx

# 10. Mensaje final
echo -e "\n Despliegue completado. Accede al portal en: https://190.113.110.160"
```

Manual de Instalación del Portal de Tutorías y Soporte Académico

Requisitos Previos

- Una máquina virtual limpia con **Ubuntu Server 20.04 o superior**
- Acceso como usuario con privilegios de **sudo**
- Conexión a Internet
- IP pública o dominio (opcional para HTTPS con Let’s Encrypt)

Pasos de Instalación

1. Conectar a la máquina virtual

Conéctate por consola o vía SSH:

ssh usuario@ip_publica

2. Crear y ejecutar el script de despliegue
    a. Crea un archivo llamado instalar.sh:
    nano instalar.sh
    b. Pega el contenido completo del script proporcionado anteriormente.
    c. Hazlo ejecutable:
    chmod +x instalar.sh
    d. Ejecuta el script:
    sudo ./instalar.sh

Nota: Este script realiza automáticamente la instalación y configuración del entorno, aplicación, proxy reverso, balanceador de carga, HTTPS y más.

Acceso a la Aplicación

Después de ejecutar el script, accede al sistema desde tu navegador:

https://190.113.110.160

Resultados

Después de ejecutar el script automatizado en una máquina virtual con Ubuntu, logramos desplegar con éxito un portal web completamente funcional, accesible desde Internet mediante la IP pública: https://190.113.110.160. Entre los logros más importantes, destacamos:

✔ Despliegue sin complicaciones: Todo se configuró correctamente en una sola ejecución, sin necesidad de hacer ajustes manuales durante el proceso.

✔ Backend robusto: El servidor Flask se encuentra funcionando detrás de Gunicorn y NGINX, lo que permite atender varias conexiones al mismo tiempo de forma eficiente.

✔ Balanceo listo para escalar: HAProxy ya está operando como balanceador de carga a nivel local, lo cual prepara el sistema para distribuir la carga y soportar fallos cuando se escalan las instancias en el futuro.

✔ Conexión segura: Se implementaron certificados SSL válidos con Let's Encrypt, permitiendo que el acceso al portal se realice a través de HTTPS de forma segura.

✔ Alta disponibilidad: Todos los servicios web se configuraron como servicios del sistema con systemd, lo que asegura que se reinicien automáticamente en caso de fallos o reinicios del sistema.

✔ Rutas probadas: Se comprobó que las rutas principales del sistema (/estudiantes, /tutores y /tutorias) funcionan correctamente mediante pruebas con herramientas como curl, Postman y acceso desde el navegador.

✔ Trazabilidad garantizada: Toda la configuración se almacenó y documentó en un repositorio Git, lo que facilita llevar un control claro y ordenado de los cambios realizados.

Discusión

A lo largo del desarrollo de este proyecto, quedó claro que construir una aplicación web no se trata solo de que funcione, sino de cómo lo hace bajo presión, cómo se adapta al crecimiento y qué tan segura es para quienes la usan. En un entorno educativo, donde cientos de estudiantes pueden depender del sistema para agendar una tutoría justo antes de un examen, no es aceptable que la plataforma se caiga o responda lentamente.

Al implementar herramientas como HAProxy para balancear las cargas y NGINX como proxy inverso, logramos distribuir el tráfico de forma eficiente, asegurando que el portal siguiera disponible incluso cuando varias personas lo usaban al mismo tiempo. Además, al integrar HTTPS con Let's Encrypt, pudimos proteger los datos y generar más confianza en los usuarios.

Un punto que marcó la diferencia fue la automatización. El hecho de que todo el despliegue se pudiera hacer con un solo script facilita muchísimo repetir el proceso en otras máquinas o ambientes. Esto no solo ahorra tiempo, sino que reduce errores, algo clave cuando el sistema tiene que estar disponible sin margen de falla.

También nos dimos cuenta de que no hace falta tener una aplicación muy compleja para aplicar buenas prácticas. Incluso un portal sencillo, como el que desarrollamos, puede ser potente si está bien pensado desde la infraestructura. Y eso abre la puerta para que otras universidades o instituciones pequeñas puedan implementar soluciones similares, sin grandes presupuestos.

En resumen, el proyecto no solo cumplió con el objetivo de crear un portal funcional, sino que nos permitió aplicar en un contexto real conceptos como alta disponibilidad, escalabilidad, seguridad y rendimiento. Más allá del código, fue una oportunidad para pensar como arquitectos de soluciones que pueden marcar la diferencia en la experiencia de aprendizaje de muchas personas.

Conclusión

Este proyecto nos permitió demostrar que es totalmente viable poner en marcha un servicio web funcional, con un enfoque sólido en alta disponibilidad, seguridad, escalabilidad y automatización, sin necesidad de recurrir a una infraestructura compleja o costosa.

La implementación del "Portal de Tutorías y Soporte Académico" integra herramientas clave como NGINX, Gunicorn, HAProxy, Let's Encrypt y Git, cada una cumpliendo un papel fundamental dentro de la arquitectura del sistema.

Logramos un despliegue automatizado, seguro y listo para producción (aunque en una escala reducida), con la capacidad de crecer fácilmente. Ya sea sumando nuevas instancias o migrando a servidores más potentes, el sistema está preparado para adaptarse.

Este trabajo sienta las bases para que el portal evolucione en un entorno universitario real, brindando un soporte técnico y académico confiable a los estudiantes, y demostrando que la tecnología puede ser una gran aliada en la educación.

Referencias Bibliográficas

- B, G., & B, G. (2025, 11 abril). Bash Script: qué es, cómo escribir uno y ejemplos. ES Tutoriales. https://www.hostinger.com/es/tutoriales/bash-script-linux
- Create a web server on a compute instance. (2021, 28 octubre). Oracle Help Center. https://docs.oracle.com/es/learn/lab_compute_instance/index.html
- Tang, L. (2024, 29 julio). Optimización de la Seguridad en Servidores Web con Let's Encrypt y Certbot - LatinCloud.com. LatinCloud Blog. https://latincloud.com/blog/optimizacion-seguridad-web-lets-encrypt/
- C, J. (2025, 13 marzo). Balanceo de Carga con HAProxy - Terranode. Terranode. https://terranode.net/blog/balanceo-de-carga-con-haproxy/
- Boada, D., & Boada, D. (2025, 18 marzo). Cómo configurar un proxy inverso NGINX en 2025. ES Tutoriales. https://www.hostinger.com/es/tutoriales/como-configurar-proxy-inverso-nginx
- Camisso, J., & Juell, K. (2020, 11 junio). Cómo presentar aplicaciones de Flask con Gunicorn y Nginx en Ubuntu 20.04. DigitalOcean. https://www.digitalocean.com/community/tutorials/how-to-serve-flask-applications-with-gunicorn-and-nginx-on-ubuntu-20-04-es
- Losapuntesde. (2025, 14 enero). Desarrollo de una API REST con Flask en Python: Creación de una interfaz de programación de aplicaciones RESTful. Apuntes de Programador. https://apuntes.de/python/desarrollo-de-una-api-rest-con-flask-en-python-creacion-de-una-interfaz-de-programacion-de-aplicaciones-restful/#gsc.tab=0