Universidad Nacional de Loja
Computación
Autores: Josue Torres, Alexis Roman.
Fecha: 19/10/2025

1. **Resumen técnico del proyecto (entorno, lenguaje, framework).**

Arquitectura General

El sistema está diseñado siguiendo una **arquitectura de microservicios**, donde las responsabilidades de negocio están desacopladas en servicios independientes. La comunicación entre la aplicación web y móvil, y el backend, se centraliza a través de un **API Gateway**, que actúa como único punto de entrada, simplificando la seguridad y el acceso de rutas.

Entorno de Desarrollo

Se implementará un entorno basado en contenedores para garantizar la consistencia y escalabilidad del sistema.

- **Contenerización: Docker**. Los microservicios, junto con la aplicación web y la base de datos, se empaquetan en su propio contenedor Docker.
- **Control de Versiones: Git** para el control de código fuente, alojado en la plataforma **GitHub**

Detalle de Componentes: Lenguajes y Frameworks

Backend (Microservicios)

- **Lenguaje: Python 3.12**
- **Framework: FastAPI**. Elegido por su alto rendimiento, su facilidad para crear APIs RESTful modernas y su generación automática de documentación a través de Swagger.
- **Base de Datos: MariaDB**. Se utilizará una sola base de datos para todos los microservicios, sabemos que no es lo mas optimo para mantener la autonomía pero es más sencillo de implementar.
- **Protocolo de Comunicación: API RESTful** utilizando **JSON** como formato de intercambio de datos.

Frontend - Aplicación Web (Para Gerentes)

Este componente funciona como un **Backend-for-Frontend (BFF)**.

- **Lenguaje (Backend): Python 3.12**
- **Framework (Backend): FastAPI**. Actúa como servidor web que gestiona las sesiones del gerente, orquesta las llamadas a los microservicios y sirve las vistas.
- **Tecnología (Frontend/Cliente): HTML, CSS, y JavaScript**. No se descarta el utilizar un framework moderno como **React, Vue.js o Angular** para construir una interfaz de usuario dinámica.

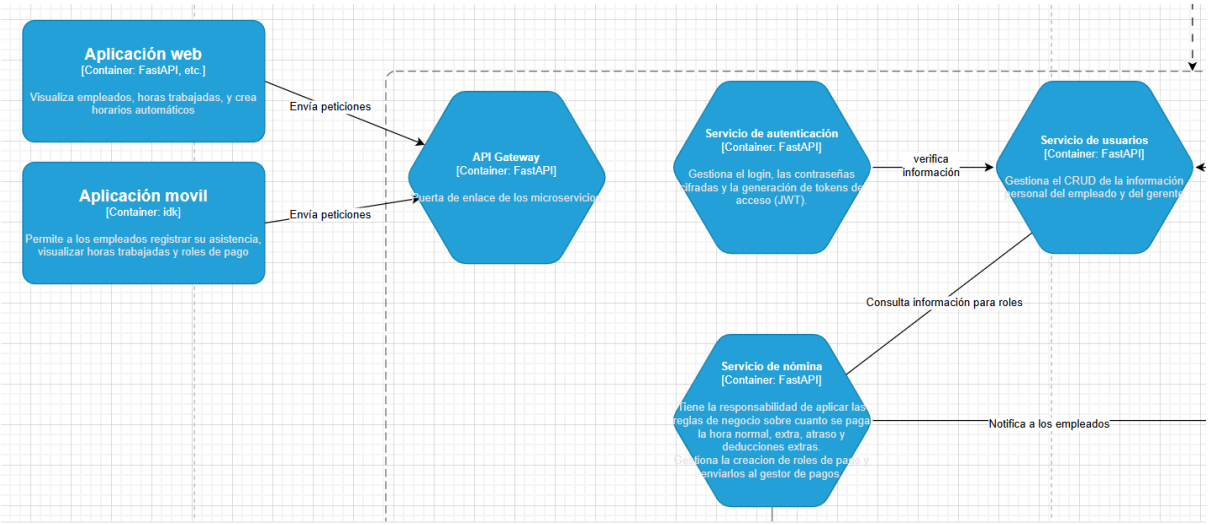
2. **Identificación de al menos 5 vulnerabilidades del OWASP Top 10 (2021) relevantes para su backend.**
3. **Descripción del riesgo y del posible impacto de cada vulnerabilidad en su sistema.**
4. **Medidas de mitigación implementadas o planificadas, explicando cómo se aplicaron en el código o la configuración.**

Vulnerabilidad	Descripción	Posible impacto en el sistema	Medidas de mitigación
A04:2021-Insecure Design	Deficiencias en el diseño del sistema que pueden llevar a fallos catastróficos en producción.	La falta de implementación en el diseño de un API Gateway abre la puerta para diversos fallos de seguridad y fallos de lógica.	- Implementar un API Gateway, donde se centralice toda la autenticación, validación de tokens y autorización antes de reenviar a los microservicios - Asegurar que cada microservicio solo tenga los permisos estrictamente necesarios para operar. - implementar microservicios especializados en realizar una sola acción de la lógica de negocio.
A06:2021-Vulnerable and Outdated Components	Utilizar versiones de frameworks, librerías o dependencias con vulnerabilidades conocidas.	Un atacante podría encontrar una vulnerabilidad en una versión antigua de FastAPI que le permita ejecutar código malicioso directamente en el servidor.	Utilizar las ultimas versiones mas estables de cada tecnología.
A07:2021-Identification and Authentication Failures	Fallas que permiten a un atacante suplantar la identidad de un usuario correcto, como por ejemplo permitir contraseñas débiles o exponer los tokens de sesión.	El servicio de autenticación maneja `/auth/login`, `/auth/logout` y `/auth/refresh`. Si no se implementa un tiempo de expiración adecuado o no se invalidan tokens en `/auth/logout`, un token comprometido podría seguir siendo válido.	- Utilizar tokens de acceso de corta duración, como 1 minutos, y tokens de refresco - Exigir contraseñas con una longitud mínima, mayúsculas, minúsculas, números, y símbolos - Guardar las contraseñas en la base de datos utilizando algoritmos de hashing
A01:2021-Broken Access Control	Algún usuario puede ver o modificar datos o funcionalidades	Los endpoints como `/users/{user_id}`, `/users/{user_id}/pa`	- Definir roles claros (**"Empleado"**, **"Gerente"**) y asociar permisos específicos a cada uno.

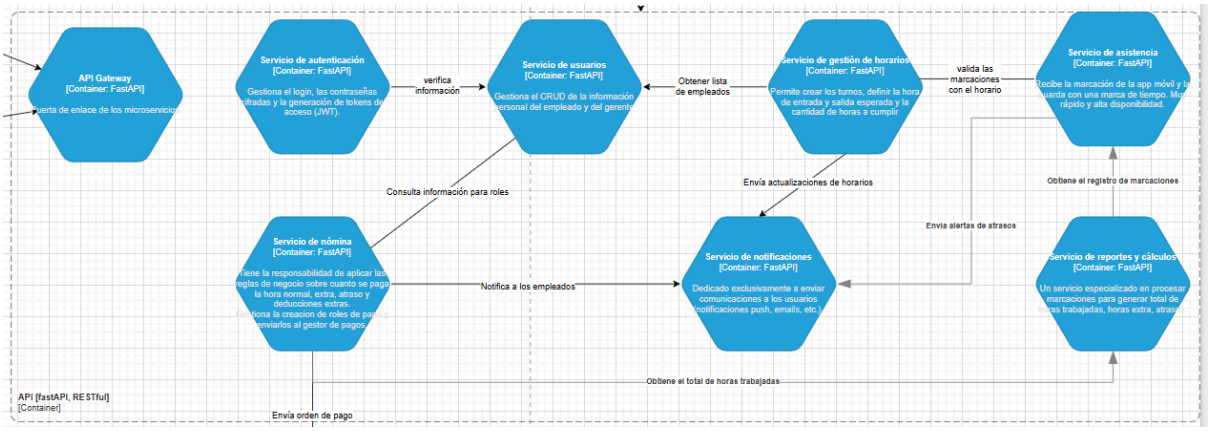
	que no le corresponden.	`yrolls` o `/reports/hours/{user_id}` pueden ser vulnerables si el sistema no valida correctamente que el usuario autenticado tenga permisos para acceder a esos recursos. Por ejemplo, un empleado podría consultar o modificar los datos de otro si el ID se pasa por parámetro.	- Verificar el rol en cada petición a los microservicios
A08:2021-Software and Data Integrity Failures	Falta de validaciones para asegurar que los datos que se envían sean correctos y estén en el formato indicado, antes de guardarlos en la bd.	Los endpoints como `/users/register`, `/payrolls/generate` o `/attend/in` aceptan datos de entrada sin mostrar validaciones robustas. Si estos datos se almacenan o se usan en cálculos, pueden permitir **inyecciones de código o datos maliciosos**.	- Los datos recibidos en el backend deben ser validados correctamente, por ejemplo una marcación no puede ser en una fecha u hora futura

5. **Evidencias de verificación: capturas de pruebas, fragmentos de código o resultados de herramientas de análisis.**

Implementación de un API Gateway: Desde el diseño se ha implementado un API Gateway para manejar toda la autorización y verificación antes de llamar a los microservicios, mejorando así la seguridad.



Microservicios especializados: Cada microservicio se encarga de una sola acción dentro del sistema, lo que nos permite una mayor modularidad, evitando la dependencia excesiva.



Componentes desactualizados: Se ejecutó la herramienta pip-audit, la cual analiza las dependencias del proyecto y reporta vulnerabilidades conocidas en las librerías instaladas.

```
$ pip-audit
Found 1 known vulnerability in 1 package
Name Version ID                  Fix Versions
---- ------- ------------------- ------------
pip  24.0    GHSA-4xh5-x5gv-qwph
```

El resultado indica una vulnerabilidad en la versión 24.0 de pip si bien pip no se utiliza directamente en la ejecución del sistema (solo para la gestión de dependencias), se recomienda actualizarlo a la versión más reciente para reducir riesgos potenciales en el entorno virtual.

Integridad de los datos:
Para esta prueba se intenta verificar que los datos enviados por el usuario sean correctos en cuanto a formato y tipo. En este caso se intentó crear un usuario con un email invalido y se obtuvo como respuesta un 402, previniendo crear un usuario sin el formato de datos correcto.

Curl

```
curl -X 'POST' \
  'http://127.0.0.1:8000/users/register' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "nombres": "Alexis",
  "apellidos": "Roman",
  "email": "12345",
  "celular": "string",
  "rol": "string",
  "password": "string"
}'
```

Request URL

```
http://127.0.0.1:8000/users/register
```

Server response

Code	Details
422	Error: Unprocessable Entity

Response body


```
{
  "detail": [
    {
      "type": "value_error",
      "loc": [
        "body",
        "email"
      ],
      "msg": "value is not a valid email address: An email address must have an @-sign.",
      "input": "12345",
      "ctx": {
        "reason": "An email address must have an @-sign."
      }
    }
  ]
}
```


Download

6. **Reflexión personal: qué aprendió sobre seguridad durante esta unidad y cómo mejoraría el diseño futuro del sistema.**

En esta unidad aprendimos sobre la importancia de mantener una seguridad robusta en cada sistema, ya que esto garantiza la protección de la información, la integridad de los datos y la continuidad de funcionamiento frente a posibles amenazas o vulnerabilidades.

La aplicación de la seguridad debe comenzar desde la fase del diseño del sistema, incorporando principios de seguridad que permitan anticipar riesgos y minimizar fallos. Esto implica definir políticas de uso, usar mecanismos de autenticación y cifrado adecuados, etc.

Para el futuro se pretende implementar todas las medidas de mitigación registradas en este documento, las cuales serán el pilar fundamental de nuestro sistema, así mismo en el camino se irán descubriendo otras posibles fallas que se irán mejorando.