# AGRADECIMIENTOS

# A mi familia, por seguir animándome a continuar creciendo en la Universidad de Almería. A mis compañeros, porque desde el Grado hemos seguido apoyándonos continuamente en nuestras vidas. A mis amigos, por estar conmigo tanto en las buenas como en las malas. Al profesorado de esta universidad, donde gracias a ellos he podido adquirir estos nuevos conocimientos en esta ingeniería, y también y sobretodo a Manuel Torres Gil, por darme la oportunidad de realizar este proyecto donde he podido poner a prueba toda mi experencia profesional previa junto a lo aprendido en este máster para mi desarrollo y desempeño personal y laboral.

# Índice de contenidos

**Capítulo 01: Introducción**  
 1.1. Motivación  
 1.2. Objetivo General  
 1.3. Objetivos Específicos  
 1.4. Planificación  
 1.5. Estructura del Documento  
  
**Capítulo 02: Estado del Arte**  
 2.1. Introducción a la Seguridad en Aplicaciones  
 2.2. Métodos Tradicionales de Autenticación y Autorización  
 2.3. Tecnologías Modernas: SSO y Gestión de Identidades  
 2.4. Herramientas Principales: Keycloak y Vault  
 2.5. Contenedores, Orquestación y DevSecOps  
  
**Capítulo 03: Herramientas y Tecnologías**  
 3.1. Keycloak  
 3.2. Vault  
 3.3. Contenedores y Virtualización con Docker  
 3.4. Despliegue con Terraform  
  
**Capítulo 04: Diseño e Implementación**  
 4.1. Requerimientos Iniciales  
 4.2. Diseño de la Arquitectura  
 4.3. Implementación  
 4.4. Pruebas  
 4.5. Optimización  
  
**Capítulo 05: Resultados y Validación**  
 5.1. Análisis de la Solución  
 5.2. Comparación con Soluciones Tradicionales  
 5.3. Beneficios Observados  
  
**Capítulo 06: Conclusiones y Futuras Líneas de Trabajo**  
 6.1. Conclusiones Generales  
 6.2. Mejoras Propuestas  
 6.3. Extensiones del Trabajo  
  
**Capítulo 07: Bibliografía**

**Índice de figuras**

# Capítulo 01: Introducción

## 1.1. Motivación

La evolución tecnológica ha generado un incremento en la complejidad de las aplicaciones,   
haciéndolas vulnerables si no se cuenta con mecanismos sólidos de seguridad. Delegar la   
autenticación y la gestión de credenciales a plataformas especializadas permite a los   
desarrolladores concentrarse en la lógica del negocio, garantizando al mismo tiempo la   
seguridad de los datos y los usuarios.

## 1.2. Objetivo General

Diseñar e implementar una solución que integre Keycloak y Vault para gestionar la   
autenticación, autorización y protección de credenciales en aplicaciones que no poseen   
esta funcionalidad de forma nativa.

## 1.3. Objetivos Específicos

- Configurar Keycloak como servidor de identidad para autenticación y autorización.  
- Utilizar Vault para la protección y gestión de credenciales sensibles.  
- Implementar ambas herramientas en un entorno aislado mediante Docker.  
- Automatizar el despliegue utilizando Terraform para garantizar escalabilidad.

## 1.4. Planificación

El proyecto se planifica en fases iterativas:   
- Fase 1: Investigación y selección de tecnologías.  
- Fase 2: Diseño de la arquitectura.  
- Fase 3: Implementación y pruebas.  
- Fase 4: Documentación y conclusiones.

## 1.5. Estructura del Documento

La estructura del documento sigue un enfoque lógico, comenzando por la introducción y   
el estado del arte, seguido por la descripción de herramientas, diseño e implementación,   
y culminando con resultados, conclusiones y referencias bibliográficas.

**Capítulo 02: Estado del Arte**

**2.1 Introducción a la Seguridad en Aplicaciones**

La seguridad en aplicaciones ha evolucionado de simples validaciones de usuario a sistemas complejos que gestionan accesos, datos sensibles y amenazas en tiempo real. Con el incremento de ciberataques y vulnerabilidades, es crucial adoptar tecnologías que aseguren integridad, confidencialidad y disponibilidad.

**2.2 Métodos Tradicionales de Autenticación y Autorización**

Sistemas basados en contraseñas: Explicación de limitaciones como la reutilización y debilidad de contraseñas.

Autenticación multifactor (MFA): Introducción como mejora, pero con desafíos de implementación.

Problemas en soluciones tradicionales: Dificultades para integrar seguridad en sistemas modernos distribuidos.

**2.3 Tecnologías Modernas: SSO y Gestión de Identidades**

Single Sign-On (SSO): Beneficios para usuarios y desarrolladores.

OAuth 2.0 y OpenID Connect: Normas para la autenticación moderna, su importancia en Keycloak.

Gestión de identidades: Introducción al concepto de Identity as a Service (IdaaS).

**2.4 Herramientas Principales: Keycloak y Vault**

Keycloak: Análisis de características, incluyendo soporte para SSO, OAuth2 y LDAP.

Vault: Gestión de secretos con seguridad reforzada, ejemplos de uso en proyectos reales.

**2.5 Contenedores, Orquestación y DevSecOps**

Contenedores: Ventajas de Docker frente a virtualización tradicional.

Orquestación con Kubernetes: Automatización para despliegues de sistemas seguros.

DevSecOps: Enfoque para integrar seguridad en todas las fases del ciclo de desarrollo.

**Capítulo 03: Herramientas y Tecnologías**

**3.1 Keycloak**

Instalación y configuración básica.

Uso de realms, roles y mapeos de usuarios.

Integración con aplicaciones mediante protocolos estándar como OAuth2.

**3.2 Vault**

Conceptos clave: secretos, policies y autenticación.

Configuración de backend de almacenamiento.

Ejemplo práctico: manejo de credenciales dinámicas.

**3.3 Contenedores y Virtualización con Docker**

Creación de imágenes Docker para Keycloak y Vault.

Redes virtuales para la comunicación entre servicios.

Ejemplo: configuración de volúmenes para persistencia de datos.

**3.4 Despliegue con Terraform**

Automatización de infraestructura como código.

Ejemplo: despliegue de un clúster Docker con Keycloak y Vault.

Ventajas de Terraform frente a otras herramientas de orquestación.

**Capítulo 04: Diseño e Implementación**

**4.1 Requerimientos Iniciales**

Especificación funcional y no funcional.

**4.2 Diseño de la Arquitectura**

Diagrama de la arquitectura: componentes, flujos de datos y dependencias.

Justificación de elecciones tecnológicas.

**4.3 Implementación**

Configuración de Keycloak para SSO.

Integración de Vault con aplicaciones mediante API REST.

Ejemplo: implementación de roles y secretos dinámicos.

**4.4 Pruebas**

Estrategias de pruebas: unitarias, integrales y de carga.

Ejemplo: simulación de ataques para validar seguridad.

**4.5 Optimización**

Escalabilidad del sistema.

Mejora en tiempos de respuesta.

Propuestas para minimizar fallos.  
  
  
  
  
  
  
  
  
  
  
  
  
**Capítulo 05: Resultados y Validación**

**5.1 Análisis de la Solución**

Evaluación del cumplimiento de objetivos iniciales.

Métricas clave: tiempo de autenticación, reducción de vulnerabilidades.

**5.2 Comparación con Soluciones Tradicionales**

Comparación de eficiencia y facilidad de uso.

Análisis de coste-beneficio.

**5.3 Beneficios Observados**

Mayor seguridad en la gestión de identidades.

Reducción en la complejidad de mantenimiento.

Flexibilidad en integración con aplicaciones heterogéneas.

**Capítulo 06: Conclusiones y Futuras Líneas de Trabajo**

**6.1 Conclusiones Generales**

Resumen del impacto del proyecto.

Reflexión sobre el aprendizaje adquirido y su relevancia en la industria.

**6.2 Mejoras Propuestas**

Inclusión de autenticación basada en biometría.

Implementación de IA para detección proactiva de anomalías.

**6.3 Extensiones del Trabajo**

Integración con otras herramientas como Kubernetes y Jenkins.

Adaptación para empresas de diferentes sectores.

**Capítulo 07: Bibliografía**