

Análisis de Cabeceras JWT a través de Paradigmas de Programación

Aidan Pace

[2025-04-28 Mon]

Análisis de Cabeceras JWT a través de Paradigmas de Programación

- Una exploración entre lenguajes de técnicas de análisis de cabeceras JWT

Lo que cubriremos

Análisis de Cabeceras JWT a través de Paradigmas de Programación

- Una exploración entre lenguajes de técnicas de análisis de cabeceras JWT
- PyCon US 2025, 14 de mayo - 22 de mayo, 2025

Lo que cubriremos

Análisis de Cabeceras JWT a través de Paradigmas de Programación

- Una exploración entre lenguajes de técnicas de análisis de cabeceras JWT
- PyCon US 2025, 14 de mayo - 22 de mayo, 2025
- Aidan Pace (@aygp-dr)

Lo que cubriremos

Análisis de Cabeceras JWT a través de Paradigmas de Programación

- Una exploración entre lenguajes de técnicas de análisis de cabeceras JWT
- PyCon US 2025, 14 de mayo - 22 de mayo, 2025
- Aidan Pace (@aygp-dr)

Lo que cubriremos

- Contexto histórico y evolución de la autenticación

Análisis de Cabeceras JWT a través de Paradigmas de Programación

- Una exploración entre lenguajes de técnicas de análisis de cabeceras JWT
- PyCon US 2025, 14 de mayo - 22 de mayo, 2025
- Aidan Pace (@aygp-dr)

Lo que cubriremos

- Contexto histórico y evolución de la autenticación
- Estructura JWT y fundamentos (amigable para principiantes)

Análisis de Cabeceras JWT a través de Paradigmas de Programación

- Una exploración entre lenguajes de técnicas de análisis de cabeceras JWT
- PyCon US 2025, 14 de mayo - 22 de mayo, 2025
- Aidan Pace (@aygp-dr)

Lo que cubriremos

- Contexto histórico y evolución de la autenticación
- Estructura JWT y fundamentos (amigable para principiantes)
- Desafíos de la codificación Base64url

Análisis de Cabeceras JWT a través de Paradigmas de Programación

- Una exploración entre lenguajes de técnicas de análisis de cabeceras JWT
- PyCon US 2025, 14 de mayo - 22 de mayo, 2025
- Aidan Pace (@aygp-dr)

Lo que cubriremos

- Contexto histórico y evolución de la autenticación
- Estructura JWT y fundamentos (amigable para principiantes)
- Desafíos de la codificación Base64url
- Patrones de análisis de cabeceras en diferentes lenguajes

Análisis de Cabeceras JWT a través de Paradigmas de Programación

- Una exploración entre lenguajes de técnicas de análisis de cabeceras JWT
- PyCon US 2025, 14 de mayo - 22 de mayo, 2025
- Aidan Pace (@aygp-dr)

Lo que cubriremos

- Contexto histórico y evolución de la autenticación
- Estructura JWT y fundamentos (amigable para principiantes)
- Desafíos de la codificación Base64url
- Patrones de análisis de cabeceras en diferentes lenguajes
- Enfoques funcionales vs orientados a objetos

Análisis de Cabeceras JWT a través de Paradigmas de Programación

- Una exploración entre lenguajes de técnicas de análisis de cabeceras JWT
- PyCon US 2025, 14 de mayo - 22 de mayo, 2025
- Aidan Pace (@aygp-dr)

Lo que cubriremos

- Contexto histórico y evolución de la autenticación
- Estructura JWT y fundamentos (amigable para principiantes)
- Desafíos de la codificación Base64url
- Patrones de análisis de cabeceras en diferentes lenguajes
- Enfoques funcionales vs orientados a objetos
- Modismos y mejores prácticas específicos de cada lenguaje

Análisis de Cabeceras JWT a través de Paradigmas de Programación

- Una exploración entre lenguajes de técnicas de análisis de cabeceras JWT
- PyCon US 2025, 14 de mayo - 22 de mayo, 2025
- Aidan Pace (@aygp-dr)

Lo que cubriremos

- Contexto histórico y evolución de la autenticación
- Estructura JWT y fundamentos (amigable para principiantes)
- Desafíos de la codificación Base64url
- Patrones de análisis de cabeceras en diferentes lenguajes
- Enfoques funcionales vs orientados a objetos
- Modismos y mejores prácticas específicos de cada lenguaje
- Consideraciones de seguridad y ataques comunes

Análisis de Cabeceras JWT a través de Paradigmas de Programación

- Una exploración entre lenguajes de técnicas de análisis de cabeceras JWT
- PyCon US 2025, 14 de mayo - 22 de mayo, 2025
- Aidan Pace (@aygp-dr)

Lo que cubriremos

- Contexto histórico y evolución de la autenticación
- Estructura JWT y fundamentos (amigable para principiantes)
- Desafíos de la codificación Base64url
- Patrones de análisis de cabeceras en diferentes lenguajes
- Enfoques funcionales vs orientados a objetos
- Modismos y mejores prácticas específicos de cada lenguaje
- Consideraciones de seguridad y ataques comunes

Contexto Histórico de la Autenticación

- Autenticación temprana: Pares de nombre de usuario/contraseña

Contexto Histórico de la Autenticación

- Autenticación temprana: Pares de nombre de usuario/contraseña
- Sesiones del lado del servidor con cookies (con estado)

Contexto Histórico de la Autenticación

- Autenticación temprana: Pares de nombre de usuario/contraseña
- Sesiones del lado del servidor con cookies (con estado)
- Surgimiento de la autenticación basada en tokens (sin estado)

Contexto Histórico de la Autenticación

- Autenticación temprana: Pares de nombre de usuario/contraseña
- Sesiones del lado del servidor con cookies (con estado)
- Surgimiento de la autenticación basada en tokens (sin estado)
- Estandarización JWT (RFC 7519, mayo 2015)

Contexto Histórico de la Autenticación

- Autenticación temprana: Pares de nombre de usuario/contraseña
- Sesiones del lado del servidor con cookies (con estado)
- Surgimiento de la autenticación basada en tokens (sin estado)
- Estandarización JWT (RFC 7519, mayo 2015)
- Flujos de autenticación modernos (OAuth 2.0, OIDC)

Conceptos Básicos de JWT

Repaso de la Estructura JWT

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjMONTY3ODkwIn0.dozjgNryP4.

Tres segmentos codificados en base64url separados por puntos:

1. **Cabecera** (algoritmo y tipo de token)
2. **Carga útil** (reclamaciones/claims)
3. **Firma**

```
digraph {  
    rankdir=LR;  
    node [shape=box, style=filled, fillcolor="#e6f3ff", fontname="monospace"];  
    edge [fontname="Arial"];
```

JavaScript (Navegador)

```
const authHeader = "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOi.  
const token = authHeader.split(' ')[1];
```

```
// IMPORTANTE: ¡En producción, verifica la firma antes de analizar!  
// Este ejemplo es solo para demostración
```

```
// Decodificar la parte de la cabecera  
const headerPart = token.split('.')[0];  
const decodedHeader = JSON.parse(atob(headerPart));  
console.log(decodedHeader);
```

Nota: `atob()` maneja base64 pero no específicamente base64url

Patrones Comunes y Variaciones

1. **Extracción de tokens**: Dividir por espacio o regex

Análisis de Rendimiento entre Lenguajes

Lenguaje	Tiempo de análisis (s)	Uso de memoria (KB)
Rust	5.2	1.8
JavaScript	24.7	12.3

Patrones Comunes y Variaciones

1. **Extracción de tokens:** Dividir por espacio o regex
2. **Manejo de Base64url:**

Análisis de Rendimiento entre Lenguajes

Lenguaje	Tiempo de análisis (s)	Uso de memoria (KB)
Rust	5.2	1.8
JavaScript	24.7	12.3

Patrones Comunes y Variaciones

1. **Extracción de tokens:** Dividir por espacio o regex
2. **Manejo de Base64url:**
 - Reemplazo de caracteres (- → +, _ → /)

Análisis de Rendimiento entre Lenguajes

Lenguaje	Tiempo de análisis (s)	Uso de memoria (KB)
Rust	5.2	1.8
JavaScript	24.7	12.3

Patrones Comunes y Variaciones

1. **Extracción de tokens:** Dividir por espacio o regex
2. **Manejo de Base64url:**
 - Reemplazo de caracteres (- → +, _ → /)
 - Cálculo de relleno

Análisis de Rendimiento entre Lenguajes

Lenguaje	Tiempo de análisis (s)	Uso de memoria (KB)
Rust	5.2	1.8
JavaScript	24.7	12.3

Patrones Comunes y Variaciones

1. **Extracción de tokens:** Dividir por espacio o regex
2. **Manejo de Base64url:**
 - Reemplazo de caracteres (- → +, _ → /)
 - Cálculo de relleno
 - Disponibilidad de decodificador seguro para URL (ventaja JVM)

Análisis de Rendimiento entre Lenguajes

Lenguaje	Tiempo de análisis (s)	Uso de memoria (KB)
Rust	5.2	1.8
JavaScript	24.7	12.3

Patrones Comunes y Variaciones

1. **Extracción de tokens**: Dividir por espacio o regex
2. **Manejo de Base64url**:
 - Reemplazo de caracteres (- → +, _ → /)
 - Cálculo de relleno
 - Disponibilidad de decodificador seguro para URL (ventaja JVM)
3. **Análisis JSON**: Nativo vs bibliotecas

Análisis de Rendimiento entre Lenguajes

Lenguaje	Tiempo de análisis (s)	Uso de memoria (KB)
Rust	5.2	1.8
JavaScript	24.7	12.3

Patrones Comunes y Variaciones

1. **Extracción de tokens**: Dividir por espacio o regex
2. **Manejo de Base64url**:
 - Reemplazo de caracteres (- → +, _ → /)
 - Cálculo de relleno
 - Disponibilidad de decodificador seguro para URL (ventaja JVM)
3. **Análisis JSON**: Nativo vs bibliotecas
4. **Manejo de errores**: Diferencias idiomáticas

Análisis de Rendimiento entre Lenguajes

Lenguaje	Tiempo de análisis (s)	Uso de memoria (KB)
Rust	5.2	1.8
JavaScript	24.7	12.3

Consideraciones de Seguridad

Mejores Prácticas de Seguridad JWT

- Siempre verificar firmas antes de analizar o usar la carga útil

Ataques JWT Comunes

Consideraciones de Seguridad

Mejores Prácticas de Seguridad JWT

- Siempre verificar firmas antes de analizar o usar la carga útil
- Usar algoritmos fuertes (preferir RS256/ES256 sobre HS256)

Ataques JWT Comunes

Consideraciones de Seguridad

Mejores Prácticas de Seguridad JWT

- Siempre verificar firmas antes de analizar o usar la carga útil
- Usar algoritmos fuertes (preferir RS256/ES256 sobre HS256)
- Implementar gestión de claves adecuada (rotación, almacenamiento seguro)

Ataques JWT Comunes

Consideraciones de Seguridad

Mejores Prácticas de Seguridad JWT

- Siempre verificar firmas antes de analizar o usar la carga útil
- Usar algoritmos fuertes (preferir RS256/ES256 sobre HS256)
- Implementar gestión de claves adecuada (rotación, almacenamiento seguro)
- Establecer tiempos de vida de token apropiados (tokens de acceso de corta duración)

Ataques JWT Comunes

Consideraciones de Seguridad

Mejores Prácticas de Seguridad JWT

- Siempre verificar firmas antes de analizar o usar la carga útil
- Usar algoritmos fuertes (preferir RS256/ES256 sobre HS256)
- Implementar gestión de claves adecuada (rotación, almacenamiento seguro)
- Establecer tiempos de vida de token apropiados (tokens de acceso de corta duración)
- Incluir reclamaciones esenciales (iss, sub, exp, aud, iat)

Ataques JWT Comunes

Consideraciones de Seguridad

Mejores Prácticas de Seguridad JWT

- Siempre verificar firmas antes de analizar o usar la carga útil
- Usar algoritmos fuertes (preferir RS256/ES256 sobre HS256)
- Implementar gestión de claves adecuada (rotación, almacenamiento seguro)
- Establecer tiempos de vida de token apropiados (tokens de acceso de corta duración)
- Incluir reclamaciones esenciales (iss, sub, exp, aud, iat)

Ataques JWT Comunes

- Ataque "alg": "none" - El atacante elimina el requisito de validación de firma

Consideraciones de Seguridad

Mejores Prácticas de Seguridad JWT

- Siempre verificar firmas antes de analizar o usar la carga útil
- Usar algoritmos fuertes (preferir RS256/ES256 sobre HS256)
- Implementar gestión de claves adecuada (rotación, almacenamiento seguro)
- Establecer tiempos de vida de token apropiados (tokens de acceso de corta duración)
- Incluir reclamaciones esenciales (iss, sub, exp, aud, iat)

Ataques JWT Comunes

- Ataque "alg": "none" - El atacante elimina el requisito de validación de firma
- Confusión de algoritmo - Cambiar de asimétrico (RS256) a simétrico (HS256)

Consideraciones de Seguridad

Mejores Prácticas de Seguridad JWT

- Siempre verificar firmas antes de analizar o usar la carga útil
- Usar algoritmos fuertes (preferir RS256/ES256 sobre HS256)
- Implementar gestión de claves adecuada (rotación, almacenamiento seguro)
- Establecer tiempos de vida de token apropiados (tokens de acceso de corta duración)
- Incluir reclamaciones esenciales (iss, sub, exp, aud, iat)

Ataques JWT Comunes

- Ataque "alg": "none" - El atacante elimina el requisito de validación de firma
- Confusión de algoritmo - Cambiar de asimétrico (RS256) a simétrico (HS256)
- Manipulación de tokens - Modificar reclamaciones sin invalidar la firma

Consideraciones de Seguridad

Mejores Prácticas de Seguridad JWT

- Siempre verificar firmas antes de analizar o usar la carga útil
- Usar algoritmos fuertes (preferir RS256/ES256 sobre HS256)
- Implementar gestión de claves adecuada (rotación, almacenamiento seguro)
- Establecer tiempos de vida de token apropiados (tokens de acceso de corta duración)
- Incluir reclamaciones esenciales (iss, sub, exp, aud, iat)

Ataques JWT Comunes

- Ataque "alg": "none" - El atacante elimina el requisito de validación de firma
- Confusión de algoritmo - Cambiar de asimétrico (RS256) a simétrico (HS256)
- Manipulación de tokens - Modificar reclamaciones sin invalidar la firma
- Inyección de tokens - Usar un token de un contexto en otro

Consideraciones de Seguridad

Mejores Prácticas de Seguridad JWT

- Siempre verificar firmas antes de analizar o usar la carga útil
- Usar algoritmos fuertes (preferir RS256/ES256 sobre HS256)
- Implementar gestión de claves adecuada (rotación, almacenamiento seguro)
- Establecer tiempos de vida de token apropiados (tokens de acceso de corta duración)
- Incluir reclamaciones esenciales (iss, sub, exp, aud, iat)

Ataques JWT Comunes

- Ataque "alg": "none" - El atacante elimina el requisito de validación de firma
- Confusión de algoritmo - Cambiar de asimétrico (RS256) a simétrico (HS256)
- Manipulación de tokens - Modificar reclamaciones sin invalidar la firma
- Inyección de tokens - Usar un token de un contexto en otro

Consideraciones de Seguridad

Mejores Prácticas de Seguridad JWT

- Siempre verificar firmas antes de analizar o usar la carga útil
- Usar algoritmos fuertes (preferir RS256/ES256 sobre HS256)
- Implementar gestión de claves adecuada (rotación, almacenamiento seguro)
- Establecer tiempos de vida de token apropiados (tokens de acceso de corta duración)
- Incluir reclamaciones esenciales (iss, sub, exp, aud, iat)

Ataques JWT Comunes

- Ataque "alg": "none" - El atacante elimina el requisito de validación de firma
- Confusión de algoritmo - Cambiar de asimétrico (RS256) a simétrico (HS256)
- Manipulación de tokens - Modificar reclamaciones sin invalidar la firma
- Inyección de tokens - Usar un token de un contexto en otro

Consideraciones de Seguridad

Mejores Prácticas de Seguridad JWT

- Siempre verificar firmas antes de analizar o usar la carga útil
- Usar algoritmos fuertes (preferir RS256/ES256 sobre HS256)
- Implementar gestión de claves adecuada (rotación, almacenamiento seguro)
- Establecer tiempos de vida de token apropiados (tokens de acceso de corta duración)
- Incluir reclamaciones esenciales (iss, sub, exp, aud, iat)

Ataques JWT Comunes

- Ataque "alg": "none" - El atacante elimina el requisito de validación de firma
- Confusión de algoritmo - Cambiar de asimétrico (RS256) a simétrico (HS256)
- Manipulación de tokens - Modificar reclamaciones sin invalidar la firma
- Inyección de tokens - Usar un token de un contexto en otro

Consideraciones de Seguridad

Mejores Prácticas de Seguridad JWT

- Siempre verificar firmas antes de analizar o usar la carga útil
- Usar algoritmos fuertes (preferir RS256/ES256 sobre HS256)
- Implementar gestión de claves adecuada (rotación, almacenamiento seguro)
- Establecer tiempos de vida de token apropiados (tokens de acceso de corta duración)
- Incluir reclamaciones esenciales (iss, sub, exp, aud, iat)

Ataques JWT Comunes

- Ataque "alg": "none" - El atacante elimina el requisito de validación de firma
- Confusión de algoritmo - Cambiar de asimétrico (RS256) a simétrico (HS256)
- Manipulación de tokens - Modificar reclamaciones sin invalidar la firma
- Inyección de tokens - Usar un token de un contexto en otro

Consideraciones de Seguridad

Mejores Prácticas de Seguridad JWT

- Siempre verificar firmas antes de analizar o usar la carga útil
- Usar algoritmos fuertes (preferir RS256/ES256 sobre HS256)
- Implementar gestión de claves adecuada (rotación, almacenamiento seguro)
- Establecer tiempos de vida de token apropiados (tokens de acceso de corta duración)
- Incluir reclamaciones esenciales (iss, sub, exp, aud, iat)

Ataques JWT Comunes

- Ataque "alg": "none" - El atacante elimina el requisito de validación de firma
- Confusión de algoritmo - Cambiar de asimétrico (RS256) a simétrico (HS256)
- Manipulación de tokens - Modificar reclamaciones sin invalidar la firma
- Inyección de tokens - Usar un token de un contexto en otro

Comparación de Implementación entre Lenguajes

Característica	JavaScript	Python	Rust	Clojure	TypeScript
Seguridad de tipos	Limitada	Opcional	Fuerte	Dinámica	Fuerte
Manejo de Base64	Manual	Incorporado	Crates	JVM	Manual
Manejo de errores	try/catch	Excepciones	Result	Monádico	try/catch
Rendimiento	Medio	Bajo	Alto	Medio	Medio
Bibliotecas JWT	Muchas	Varias	Pocas	Pocas	Muchas

JWT en Producción

- Validación de tokens en API Gateway

Comparación de Implementación entre Lenguajes

Característica	JavaScript	Python	Rust	Clojure	TypeScript
Seguridad de tipos	Limitada	Opcional	Fuerte	Dinámica	Fuerte
Manejo de Base64	Manual	Incorporado	Crates	JVM	Manual
Manejo de errores	try/catch	Excepciones	Result	Monádico	try/catch
Rendimiento	Medio	Bajo	Alto	Medio	Medio
Bibliotecas JWT	Muchas	Varias	Pocas	Pocas	Muchas

JWT en Producción

- Validación de tokens en API Gateway
- Autorización de microservicios

Comparación de Implementación entre Lenguajes

Característica	JavaScript	Python	Rust	Clojure	TypeScript
Seguridad de tipos	Limitada	Opcional	Fuerte	Dinámica	Fuerte
Manejo de Base64	Manual	Incorporado	Crates	JVM	Manual
Manejo de errores	try/catch	Excepciones	Result	Monádico	try/catch
Rendimiento	Medio	Bajo	Alto	Medio	Medio
Bibliotecas JWT	Muchas	Varias	Pocas	Pocas	Muchas

JWT en Producción

- Validación de tokens en API Gateway
- Autorización de microservicios
- Implementaciones de Inicio de Sesión Único

Comparación de Implementación entre Lenguajes

Característica	JavaScript	Python	Rust	Clojure	TypeScript
Seguridad de tipos	Limitada	Opcional	Fuerte	Dinámica	Fuerte
Manejo de Base64	Manual	Incorporado	Crates	JVM	Manual
Manejo de errores	try/catch	Excepciones	Result	Monádico	try/catch
Rendimiento	Medio	Bajo	Alto	Medio	Medio
Bibliotecas JWT	Muchas	Varias	Pocas	Pocas	Muchas

JWT en Producción

- Validación de tokens en API Gateway
- Autorización de microservicios
- Implementaciones de Inicio de Sesión Único

Comparación de Implementación entre Lenguajes

Característica	JavaScript	Python	Rust	Clojure	TypeScript
Seguridad de tipos	Limitada	Opcional	Fuerte	Dinámica	Fuerte
Manejo de Base64	Manual	Incorporado	Crates	JVM	Manual
Manejo de errores	try/catch	Excepciones	Result	Monádico	try/catch
Rendimiento	Medio	Bajo	Alto	Medio	Medio
Bibliotecas JWT	Muchas	Varias	Pocas	Pocas	Muchas

JWT en Producción

- Validación de tokens en API Gateway
- Autorización de microservicios
- Implementaciones de Inicio de Sesión Único

Depuración y Solución de Problemas

Problemas Comunes de JWT y Soluciones

- **Firma inválida** - Verificar coincidencia de claves, consistencia de algoritmo

Herramientas de Depuración

Depuración y Solución de Problemas

Problemas Comunes de JWT y Soluciones

- **Firma inválida** - Verificar coincidencia de claves, consistencia de algoritmo
- **Tokens expirados** - Verificar sincronización de reloj cliente/servidor

Herramientas de Depuración

Depuración y Solución de Problemas

Problemas Comunes de JWT y Soluciones

- **Firma inválida** - Verificar coincidencia de claves, consistencia de algoritmo
- **Tokens expirados** - Verificar sincronización de reloj cliente/servidor
- **Tokens mal formados** - Inspeccionar codificación, asegurar manejo adecuado de base64url

Herramientas de Depuración

Problemas Comunes de JWT y Soluciones

- **Firma inválida** - Verificar coincidencia de claves, consistencia de algoritmo
- **Tokens expirados** - Verificar sincronización de reloj cliente/servidor
- **Tokens mal formados** - Inspeccionar codificación, asegurar manejo adecuado de base64url
- **Reclamaciones faltantes** - Validar estructura del token contra esquema esperado

Herramientas de Depuración

Depuración y Solución de Problemas

Problemas Comunes de JWT y Soluciones

- **Firma inválida** - Verificar coincidencia de claves, consistencia de algoritmo
- **Tokens expirados** - Verificar sincronización de reloj cliente/servidor
- **Tokens mal formados** - Inspeccionar codificación, asegurar manejo adecuado de base64url
- **Reclamaciones faltantes** - Validar estructura del token contra esquema esperado
- **Desajuste de algoritmo** - Confirmar que el alg de cabecera coincide con la implementación

Herramientas de Depuración

Depuración y Solución de Problemas

Problemas Comunes de JWT y Soluciones

- **Firma inválida** - Verificar coincidencia de claves, consistencia de algoritmo
- **Tokens expirados** - Verificar sincronización de reloj cliente/servidor
- **Tokens mal formados** - Inspeccionar codificación, asegurar manejo adecuado de base64url
- **Reclamaciones faltantes** - Validar estructura del token contra esquema esperado
- **Desajuste de algoritmo** - Confirmar que el alg de cabecera coincide con la implementación

Herramientas de Depuración

- Depurador JWT en línea (jwt.io)

Depuración y Solución de Problemas

Problemas Comunes de JWT y Soluciones

- **Firma inválida** - Verificar coincidencia de claves, consistencia de algoritmo
- **Tokens expirados** - Verificar sincronización de reloj cliente/servidor
- **Tokens mal formados** - Inspeccionar codificación, asegurar manejo adecuado de base64url
- **Reclamaciones faltantes** - Validar estructura del token contra esquema esperado
- **Desajuste de algoritmo** - Confirmar que el alg de cabecera coincide con la implementación

Herramientas de Depuración

- Depurador JWT en línea (jwt.io)
- Bibliotecas JWT específicas de lenguaje con opciones de depuración

Depuración y Solución de Problemas

Problemas Comunes de JWT y Soluciones

- **Firma inválida** - Verificar coincidencia de claves, consistencia de algoritmo
- **Tokens expirados** - Verificar sincronización de reloj cliente/servidor
- **Tokens mal formados** - Inspeccionar codificación, asegurar manejo adecuado de base64url
- **Reclamaciones faltantes** - Validar estructura del token contra esquema esperado
- **Desajuste de algoritmo** - Confirmar que el alg de cabecera coincide con la implementación

Herramientas de Depuración

- Depurador JWT en línea (jwt.io)
- Bibliotecas JWT específicas de lenguaje con opciones de depuración
- Herramientas de inspección Base64

Depuración y Solución de Problemas

Problemas Comunes de JWT y Soluciones

- **Firma inválida** - Verificar coincidencia de claves, consistencia de algoritmo
- **Tokens expirados** - Verificar sincronización de reloj cliente/servidor
- **Tokens mal formados** - Inspeccionar codificación, asegurar manejo adecuado de base64url
- **Reclamaciones faltantes** - Validar estructura del token contra esquema esperado
- **Desajuste de algoritmo** - Confirmar que el alg de cabecera coincide con la implementación

Herramientas de Depuración

- Depurador JWT en línea (jwt.io)
- Bibliotecas JWT específicas de lenguaje con opciones de depuración
- Herramientas de inspección Base64

Conclusión

Perspectivas entre Paradigmas

Paradigma	Fortalezas	Aplicación JWT
Orientado a objetos	Encapsulación, herencia	Token con métodos de validación
Funcional	Composición, inmutabilidad	Tubería de transformación para análisis
Procedural	Simplicidad, rendimiento	Validadores ligeros
Reactivo	Manejo de eventos	Verificación de token en flujos asíncrono

Conclusiones Principales

1. La codificación Base64url requiere atención especial

Conclusión

Perspectivas entre Paradigmas

Paradigma	Fortalezas	Aplicación JWT
Orientado a objetos	Encapsulación, herencia	Token con métodos de validación
Funcional	Composición, inmutabilidad	Tubería de transformación para análisis
Procedural	Simplicidad, rendimiento	Validadores ligeros
Reactivo	Manejo de eventos	Verificación de token en flujos asíncrono

Conclusiones Principales

1. La codificación Base64url requiere atención especial
2. Cada lenguaje tiene ventajas idiomáticas de análisis

Conclusión

Perspectivas entre Paradigmas

Paradigma	Fortalezas	Aplicación JWT
Orientado a objetos	Encapsulación, herencia	Token con métodos de validación
Funcional	Composición, inmutabilidad	Tubería de transformación para análisis
Procedural	Simplicidad, rendimiento	Validadores ligeros
Reactivo	Manejo de eventos	Verificación de token en flujos asíncrono

Conclusiones Principales

1. La codificación Base64url requiere atención especial
2. Cada lenguaje tiene ventajas idiomáticas de análisis
3. Los enfoques funcionales brillan para tuberías de transformación

Conclusión

Perspectivas entre Paradigmas

Paradigma	Fortalezas	Aplicación JWT
Orientado a objetos	Encapsulación, herencia	Token con métodos de validación
Funcional	Composición, inmutabilidad	Tubería de transformación para análisis
Procedural	Simplicidad, rendimiento	Validadores ligeros
Reactivo	Manejo de eventos	Verificación de token en flujos asíncrono

Conclusiones Principales

1. La codificación Base64url requiere atención especial
2. Cada lenguaje tiene ventajas idiomáticas de análisis
3. Los enfoques funcionales brillan para tuberías de transformación
4. Seguridad primero: siempre verificar firmas antes de analizar

Conclusión

Perspectivas entre Paradigmas

Paradigma	Fortalezas	Aplicación JWT
Orientado a objetos	Encapsulación, herencia	Token con métodos de validación
Funcional	Composición, inmutabilidad	Tubería de transformación para análisis
Procedural	Simplicidad, rendimiento	Validadores ligeros
Reactivo	Manejo de eventos	Verificación de token en flujos asíncrono

Conclusiones Principales

1. La codificación Base64url requiere atención especial
2. Cada lenguaje tiene ventajas idiomáticas de análisis
3. Los enfoques funcionales brillan para tuberías de transformación
4. Seguridad primero: siempre verificar firmas antes de analizar

Conclusión

Perspectivas entre Paradigmas

Paradigma	Fortalezas	Aplicación JWT
Orientado a objetos	Encapsulación, herencia	Token con métodos de validación
Funcional	Composición, inmutabilidad	Tubería de transformación para análisis
Procedural	Simplicidad, rendimiento	Validadores ligeros
Reactivo	Manejo de eventos	Verificación de token en flujos asíncrono

Conclusiones Principales

1. La codificación Base64url requiere atención especial
2. Cada lenguaje tiene ventajas idiomáticas de análisis
3. Los enfoques funcionales brillan para tuberías de transformación
4. Seguridad primero: siempre verificar firmas antes de analizar

Conclusión

Perspectivas entre Paradigmas

Paradigma	Fortalezas	Aplicación JWT
Orientado a objetos	Encapsulación, herencia	Token con métodos de validación
Funcional	Composición, inmutabilidad	Tubería de transformación para análisis
Procedural	Simplicidad, rendimiento	Validadores ligeros
Reactivo	Manejo de eventos	Verificación de token en flujos asíncrono

Conclusiones Principales

1. La codificación Base64url requiere atención especial
2. Cada lenguaje tiene ventajas idiomáticas de análisis
3. Los enfoques funcionales brillan para tuberías de transformación
4. Seguridad primero: siempre verificar firmas antes de analizar

Conclusión

Perspectivas entre Paradigmas

Paradigma	Fortalezas	Aplicación JWT
Orientado a objetos	Encapsulación, herencia	Token con métodos de validación
Funcional	Composición, inmutabilidad	Tubería de transformación para análisis
Procedural	Simplicidad, rendimiento	Validadores ligeros
Reactivo	Manejo de eventos	Verificación de token en flujos asíncrono

Conclusiones Principales

1. La codificación Base64url requiere atención especial
2. Cada lenguaje tiene ventajas idiomáticas de análisis
3. Los enfoques funcionales brillan para tuberías de transformación
4. Seguridad primero: siempre verificar firmas antes de analizar

Conclusión

Perspectivas entre Paradigmas

Paradigma	Fortalezas	Aplicación JWT
Orientado a objetos	Encapsulación, herencia	Token con métodos de validación
Funcional	Composición, inmutabilidad	Tubería de transformación para análisis
Procedural	Simplicidad, rendimiento	Validadores ligeros
Reactivo	Manejo de eventos	Verificación de token en flujos asíncrono

Conclusiones Principales

1. La codificación Base64url requiere atención especial
2. Cada lenguaje tiene ventajas idiomáticas de análisis
3. Los enfoques funcionales brillan para tuberías de transformación
4. Seguridad primero: siempre verificar firmas antes de analizar

Conclusión

Perspectivas entre Paradigmas

Paradigma	Fortalezas	Aplicación JWT
Orientado a objetos	Encapsulación, herencia	Token con métodos de validación
Funcional	Composición, inmutabilidad	Tubería de transformación para análisis
Procedural	Simplicidad, rendimiento	Validadores ligeros
Reactivo	Manejo de eventos	Verificación de token en flujos asíncrono

Conclusiones Principales

1. La codificación Base64url requiere atención especial
2. Cada lenguaje tiene ventajas idiomáticas de análisis
3. Los enfoques funcionales brillan para tuberías de transformación
4. Seguridad primero: siempre verificar firmas antes de analizar

Conclusión

Perspectivas entre Paradigmas

Paradigma	Fortalezas	Aplicación JWT
Orientado a objetos	Encapsulación, herencia	Token con métodos de validación
Funcional	Composición, inmutabilidad	Tubería de transformación para análisis
Procedural	Simplicidad, rendimiento	Validadores ligeros
Reactivo	Manejo de eventos	Verificación de token en flujos asíncrono

Conclusiones Principales

1. La codificación Base64url requiere atención especial
2. Cada lenguaje tiene ventajas idiomáticas de análisis
3. Los enfoques funcionales brillan para tuberías de transformación
4. Seguridad primero: siempre verificar firmas antes de analizar