|  |  |
| --- | --- |
| Mi Educación en Línea: ¿Te interesa contribuir a la seguridad ...  Sistema de Autenticación Centralizado  SAC | RESUMEN  [-- pendiente --]  Juan Carlos Luna Hernandez |

Tabla de Contenido

[**Introducción** 2](#_Toc179142225)

[**Objetivo** 2](#_Toc179142226)

[**Metodología** 4](#_Toc179142227)

[**Marco Conceptual** 6](#_Toc179142228)

[**Autenticación** 6](#_Toc179142229)

[**Tipos de Autenticación** 6](#_Toc179142230)

[**Términos Relacionados** 7](#_Toc179142231)

[**Marco Teórico** 10](#_Toc179142232)

[**Desarrollo** 10](#_Toc179142233)

[**Conclusiones** 12](#_Toc179142234)

[**Referencias** 13](#_Toc179142235)

**Introducción**

El control de acceso es un mecanismo fundamental en los sistemas de información que limita el acceso a datos y funcionalidades según el perfil del usuario. A través de la definición de roles y permisos, se garantiza la seguridad de la información y la eficiencia de los procesos.

Sin embargo, en muchos sistemas actuales, se han perdido la comprensión de cómo funciona este mecanismo a nivel técnico. La proliferación de herramientas y frameworks que automatizan gran parte del proceso ha llevado a una desconexión entre los desarrolladores y los principios básicos del control de acceso.

**Objetivo**

El desarrollo de un Sistema de Autenticación Centralizado (SAC) destinado a unificar y organizar el conocimiento sobre el control de acceso. A través de este sistema, se busca proporcionar una base sólida para comprender los conceptos clave y los componentes que intervienen en la gestión de accesos a sistemas y recursos.

**Objetivos secundarios**

* **Definir roles y permisos:** crear una estructura jerárquica de roles y asignar permisos específicos a cada uno.
* **Gestionar usuarios:** crear, modificar y eliminar cuentas de usuario, asociándolas a los roles correspondientes.
* **Integrar con otros sistemas:** proporcionar interfaces de programación (APIs) para que otros sistemas puedan autenticar usuarios y autorizar sus acciones.
* **Documentar el proceso:** crear una documentación detallada que explique el funcionamiento del sistema y los conceptos clave del control de acceso.

**Beneficios esperados**

* **A nivel de conocimiento y gestión**
  + **Unificación del conocimiento**: se consolida la información dispersa sobre control de acceso en una única fuente, facilitando su consulta y actualización.
  + **Establecimiento de estándares**: se definen criterios y terminologías comunes para describir los diferentes componentes y procesos del control de acceso, lo que mejora la comunicación y la colaboración entre equipos.
  + **Facilitación de la comprensión**: se ofrece una visión clara y estructurada de los conceptos fundamentales, lo que reduce la curva de aprendizaje para nuevos usuarios y facilita la capacitación del personal.
  + **Mejora en la toma de decisiones**: al contar con una base de conocimiento sólida, se pueden tomar decisiones más informadas y estratégicas en materia de seguridad y gestión de accesos.
  + **Optimización de procesos**: se identifican las mejores prácticas y se establecen procesos eficientes para la gestión de identidades y accesos.
* **A nivel técnico y operativo**
  + **Mayor seguridad**: se centraliza la gestión de las credenciales de acceso, lo que reduce el riesgo de vulnerabilidades y ataques.
  + **Reducción de costos**: se optimizan los recursos al evitar la duplicación de esfuerzos y la adquisición de múltiples herramientas.
  + **Mayor eficiencia**: se automatizan tareas repetitivas, lo que libera tiempo para que el personal se enfoque en actividades de mayor valor agregado.
  + **Escalabilidad**: el sistema puede adaptarse fácilmente a las necesidades cambiantes de la organización, permitiendo su crecimiento y expansión.
  + **Integración con otros sistemas**: el SAC puede integrarse con otros sistemas de información, lo que permite una gestión unificada de la identidad y el acceso.
* **A nivel estratégico**
  + **Aumento de la confianza**: un sistema de autenticación centralizado sólido y confiable mejora la percepción de seguridad de los usuarios y de los clientes.
  + **Cumplimiento normativo**: se facilita el cumplimiento de las regulaciones y estándares de seguridad vigentes.
  + **Mejora de la imagen corporativa**: una gestión eficaz de la seguridad de la información contribuye a la reputación de la organización.

**Componentes clave**

* **Repositorio de usuarios**: almacena la información de los usuarios, incluyendo credenciales y asociaciones a roles.
* **Motor de autorización**: evalúa las solicitudes de acceso y determina si el usuario tiene los permisos necesarios.
* **API de autenticación**: proporciona servicios para autenticar usuarios y obtener tokens de acceso.

**Metodología**

Haciendo una analogía con la construcción de un edificio sólido y seguro, el objetivo del proyecto SAC es la representación de la construcción del edificio, pero en el mundo digital. Para lograrlo, se plantea un proceso que combina la planificación detallada de la arquitectura con la agilidad y flexibilidad de adaptarse a los cambios.

**Fase de planificación: los cimientos**

Se comenzará diseñando los cimientos del edificio. Definiendo en detalle cada pieza del sistema: las tablas de la base de datos, las funciones que gestionarán los usuarios, roles y permisos, y las interfaces que permitirán a otros sistemas interactuar con SAC. Esta fase es la base para asegurar que toda la estructura sea coherente y sólida.

**Fase de desarrollo: construyendo las paredes**

Una vez completado los planos, se iniciará la construcción de las paredes del edificio. Esto implica escribir el código para implementar las funcionalidades del sistema. Se crearán funciones para registrar nuevos usuarios, iniciar sesión, asignar roles y permisos, y evaluar si un usuario tiene autorización para realizar una acción determinada.

**Fase de pruebas: asegurando la calidad**

Antes de que el edificio sea habitado, se realizan una serie de pruebas exhaustivas para asegurar de que todo funcione correctamente. Se probará cada pieza del sistema de forma individual y luego se integrarán para verificar que trabajen en conjunto. También se evaluará la seguridad del sistema para protegerlo de posibles ataques.

**Fase de implementación y mantenimiento: habitando el edificio**

Una vez construido y probado el edificio, será el momento de mudarse. Se instalará el sistema en el entorno de producción y se configurarán todos los componentes necesarios. Se capacitará a los usuarios para que puedan aprovechar al máximo las funcionalidades del sistema. Finalmente, se establecerá un plan de mantenimiento para realizar correcciones y mejoras a lo largo del tiempo.

A lo largo de todo el proceso, se seguirá una metodología ágil. Esto significa que se dividirá el proyecto en pequeñas partes, para realizar entrega de resultados de forma incremental. De esta manera, se obtendrá feedback (retroalimentación) de los usuarios de forma inmediata, lo que permitirá realizar ajustes en el camino, asegurando que el sistema se adapte a las necesidades reales.

**Marco Conceptual**

**Autenticación**

De acuerdo con Okta (2023) y Microsoft (2024), la autenticación es el proceso de verificación de la identidad de un usuario, dispositivo o servicio que intenta acceder a un sistema o recurso digital. Su objetivo principal es garantizar que solo las entidades autorizadas pueden ingresar y utilizar los sistemas, protegiendo así la integridad y confidencialidad de la información.

La autenticación actualmente se apoya de algunos componentes, como:

* **Factores de autenticación**, que son los elementos utilizados para verificar la identidad, como contraseñas, biometría (huellas dactilares, reconocimiento facial), tokens de seguridad, o una combinación de ellos.
* **Métodos de autenticación**, los cuales son los procedimientos utilizados para validar los factores de autenticación, como la comparación de contraseñas almacenadas, el análisis de patrones biométricos o la verificación de códigos de un solo uso (OTP).
* **Sistemas de gestión de identidades y acceso (IAM)**, que proporcionan un marco para gestionar las identidades de los usuarios, autenticarlos y autorizar su acceso a los recursos.

La autenticación es importante porque protege la información confidencial al limitar el acceso a usuarios autorizados, sin comprometer la integridad de los sistemas; de esta forma ayuda a las organizaciones a cumplir con las regulaciones de seguridad de la información, y al mismo tiempo permite a los usuarios a acceder a los servicios de manera segura y conveniente.

**Tipos de Autenticación**

En los últimos años, los mecanismos de autenticación han evolucionado para implementar métodos que aseguren un acceso adecuado a los sistemas. Se identifican los siguientes tipos de autenticación:

* **Basada en contraseña**: este es el método más común en la mayoría de los sistemas, por su facilidad de uso por parte de los usuarios del sistema, sin embargo, es el más vulnerable a ataques de seguridad.
* **Multifactor**: además de la contraseña, combina varios factores de autenticación para aumentar la seguridad; por ejemplo: biometría, tokens o la combinación de estas.
* **Sin contraseña**: la autenticación se realiza solamente con métodos de biometría o tokens de hardware o software, relacionadas a una cuenta de usuario, para eliminar la necesidad de contraseñas.
* **Basada en certificados**: se emplean certificados digitales para verificar la identidad.

Se puede observar que la autenticación es una práctica esencial en la seguridad de la información. Al implementar mecanismos de autenticación robustos, las organizaciones pueden proteger sus sistemas y datos de manera efectiva. La elección del método de autenticación adecuado dependerá de factores como el nivel de seguridad requerido, la complejidad del sistema y la experiencia de los usuarios.

**Términos Relacionados**

Los siguientes términos son esenciales para comprender cómo se gestiona la seguridad en los sistemas informáticos modernos. Al trabajar juntos, permiten a las empresas proteger sus datos y garantizar que solo las personas autorizadas puedan acceder a ellos. Es decir, garantizan el control y la protección de los recursos informáticos.

**Identidad**

La identidad se refiere a la representación única de una persona, dispositivo o entidad dentro de un sistema informático. En otras palabras: es ***quién eres***. Esta representación puede estar asociada a atributos como nombre de usuario, dirección de correo electrónico, números de identificación o características biométricas. Sirve como base para autenticar y autorizar el acceso a recursos y servicios.

El concepto de identidad se enfoca en la autenticación y autorización de usuarios y recursos, garantizando la **integridad** (*las personas autorizadas pueden modificar la información*), **confidencialidad** (*datos legibles solo por usuarios autorizados*), **irrefutabilidad** (*el usuario no puede negar las acciones que realizó*) y **disponibilidad** (*disponibilidad de los recursos cuando sean necesarios*) de la información. (Kurth, 2024)

**Credenciales**

Las credenciales, según la capacitación gratuita sobre seguridad de Fortinet (2024), son los elementos de información que se utilizan para verificar la identidad de un sujeto (persona o entidad) durante el proceso de autenticación. Es decir: ***probar que eres tú***. Las credenciales más comunes son:

* **Nombre de usuario y contraseña**: es la combinación de un nombre único y una cadena de caracteres secreta.
* **Tokens**: son dispositivos físicos o virtuales que generan códigos de un solo uso para autenticación.
* **Certificados digitales**: son archivos electrónicos que contienen información sobre la identidad de un sujeto y una firma digital.
* **Datos biométricos**: son características físicas únicas de una persona, como huellas dactilares, reconocimiento facial, iris o voz.

También se pueden categorizar las credenciales por diferentes factores, como: ***algo que se sabe*** (contraseña); ***algo que se posee*** (tokens y certificados); y ***algo que se es*** (biometría).

**Autorización**

Fernández (2023) y Oteir (2023), explican que la autorización es el proceso de determinar los permisos o privilegios específicos que se conceden a un usuario autenticado. Una vez verificada la identidad, la autorización define qué acciones puede realizar el usuario dentro del sistema, como leer, escribir, modificar o eliminar datos, ejecutar programas o acceder a recursos específicos. La autorización se basa en roles, políticas y perfiles de acceso predefinidos. Básicamente es ***lo que puedes hacer***. Por ejemplo, un usuario puede autenticarse en una aplicación, pero la autorización decide qué funcionalidades puede utilizar y qué datos puede ver.

**Tipo de autorización**

Existen diversos modelos de autorización, cada uno con sus características y aplicaciones:

* **RBAC**: el control de acceso basado en roles es el más común y asigna permisos a los usuarios en función de sus roles dentro la organización. Simplifica la gestión de permisos y reduce los riesgos de errores.
* **ABAC**: el control de acceso basado en atributos ofrece una mayor granularidad, permitiendo conceder acceso a los recursos en función de atributos específicos del usuario, como su departamento o ubicación.
* **MAC**: el control de acceso obligatorio impone políticas de acceso a nivel de sistema, brindando un control centralizado y estricto.
* **DAC**: el control de acceso discrecional permite a los usuarios individuales controlar el acceso a los recursos que poseen, ofreciendo mayor flexibilidad, pero también mayor riesgo.
* Finalmente, el control de acceso **basado en reglas** permite definir reglas personalizadas para otorgar acceso, adaptándose a escenarios más complejos.

**Técnicas de autorización**

Para implementar estos tipos de autorización, se utilizan diversas técnicas, por ejemplo: (Rodríguez, 2019)

* **JWT**: los JSON Web Tokens son una forma compacta y segura de transmitir información de autenticación y autorización, ideal para aplicaciones web.
* **SAML**: el Security Assertion Markup Language es un protocolo estándar para el intercambio de información de autenticación y autorización entre diferentes sistemas, utilizado comúnmente en entornos empresariales.
* **OpenID**: es un protocolo de autenticación descentralizado que permite a los usuarios iniciar sesión en múltiples sitios web con un único conjunto de credenciales.
* **OAuth**: es un marco de autorización que permite a aplicaciones de terceros acceder a recursos de usuario en otros servicios, como plataformas de redes sociales.

**Acceso**

El acceso se refiere al ***derecho*** de un usuario ***a utilizar un recurso o servicio***. El acceso se otorga a los usuarios autenticados y autorizados, y está limitado por los permisos asignados. Los tipos de acceso pueden incluir:

* **Acceso físico**: acceso a instalaciones o dispositivos físicos.
* **Acceso lógico**: acceso a sistemas informáticos, aplicaciones y datos.
* **Acceso remoto**: acceso a recursos desde una ubicación remota.

**Marco Teórico**

[*proporciona el sustento teórico, relacionando el trabajo de investigación con las teorías y modelos existentes; por ejemplo:*]

* *Revisar las principales teorías relacionadas con la seguridad de la información, como la autenticación multifactor, los modelos de amenaza, y los principios de seguridad.*
* *Describir los diferentes modelos de IAM (Gestión de Identidades y Accesos) y sus características, seleccionado aquellos que sean más relevantes para el SAC.*
* *Mencionar las normas y estándares internacionales relacionados con la seguridad de la información y la gestión de identidades, como ISO/IEC 270001, NIST 800-63, etc.*
* *Presentar ejemplos de sistemas de autenticación similares al SAC, analizando sus fortalezas y debilidades.*

**Desarrollo**

[*agregar aquí el detalle para el desarrollo del proyecto*]

|  |
| --- |
| **Considerando la tecnología:** MySQL como base de datos y PHP para la lógica de negocio, se propone la siguiente metodología, combinando elementos de la metodología Ágil y Waterfall:  **Fase 1: Planificación y Diseño**   * **Análisis de requerimientos:**   + Definición detallada de los roles y permisos necesarios.   + Diseño de la estructura de la base de datos MySQL (tablas, relaciones, índices).   + Diseño de la arquitectura del sistema, incluyendo componentes y flujo de datos.   + Identificación de las interfaces de usuario (API REST). * **Diseño técnico:**   + Selección de frameworks PHP (Laravel, Symfony, etc.) para agilizar el desarrollo.   + Diseño de las clases y funciones PHP para la gestión de usuarios, roles, permisos y autenticación.   + Definición de las reglas de negocio para la autorización. * **Planificación de tareas:**   + Creación de un diagrama de Gantt para visualizar el cronograma del proyecto.   + Estimación del tiempo requerido para cada tarea.   **Fase 2: Desarrollo**   * **Desarrollo de la base de datos:**   + Creación de las tablas en MySQL según el diseño.   + Carga de datos iniciales (si aplica). * **Desarrollo del backend:**   + Implementación de las funcionalidades en PHP:     - Registro de usuarios.     - Inicio de sesión.     - Gestión de roles y permisos.     - Motor de autorización (evaluación de permisos).     - API REST para la interacción con otros sistemas. * **Desarrollo del frontend (opcional):**   + Si se requiere una interfaz de usuario, se desarrollará utilizando tecnologías como HTML, CSS y JavaScript.   **Fase 3: Pruebas**   * **Pruebas unitarias:**   + Verificación del correcto funcionamiento de cada módulo de forma aislada. * **Pruebas de integración:**   + Validación de la interacción entre los diferentes componentes del sistema.   + Pruebas de las API REST. * **Pruebas de seguridad:**   + Evaluación de la vulnerabilidad del sistema a ataques comunes (inyección SQL, XSS, etc.). * **Pruebas de rendimiento:**   + Medición de la capacidad del sistema para manejar un gran número de usuarios y solicitudes.   **Fase 4: Implementación y Mantenimiento**   * **Deployment:**   + Instalación del sistema en el entorno de producción.   + Configuración de servidores web y base de datos. * **Capacitación:**   + Capacitación de los usuarios finales en el uso del sistema. * **Mantenimiento:**   + Corrección de errores y mejoras continuas.   + Monitoreo del sistema para detectar problemas y optimizar el rendimiento.   **Metodología Ágil**   * **Iteraciones cortas:** Dividir el proyecto en iteraciones cortas (sprints) para obtener feedback temprano y realizar ajustes. * **Desarrollo incremental:** Entregar funcionalidades de forma incremental para obtener valor de negocio más rápido. * **Colaboración:** Fomentar la colaboración entre los miembros del equipo a través de reuniones diarias y revisiones de código.   **Consideraciones Adicionales**   * **Seguridad:** Implementar medidas de seguridad robustas para proteger la información de los usuarios (encriptación, gestión de contraseñas seguras, etc.). * **Escalabilidad:** Diseñar el sistema para que pueda escalar horizontalmente en caso de crecimiento. * **Documentación:** Crear una documentación completa del sistema, incluyendo la arquitectura, el código y las guías de usuario. * **Versionamiento:** Utilizar un sistema de control de versiones (Git) para gestionar los cambios en el código.   **Herramientas Recomendadas:**   * **Base de datos:** MySQL Workbench para diseño y administración. * **Desarrollo PHP:** PhpStorm, Visual Studio Code. * **Control de versiones:** Git. * **Gestión de proyectos:** Trello, Jira.   **Adaptación de la Metodología:**  Esta propuesta es un punto de partida y puede adaptarse según las necesidades específicas del proyecto. Por ejemplo, si el proyecto es muy grande, se puede considerar dividirlo en subproyectos más pequeños.  **¿Qué te parece esta propuesta? ¿Hay algún aspecto que te gustaría profundizar o alguna otra metodología que te interese?**  **Adicionalmente, podríamos explorar temas como:**   * **Implementación de autenticación de dos factores.** * **Integración con servicios de directorio activo.** * **Gestión de sesiones y tokens de acceso.** * **Mejores prácticas para la seguridad en aplicaciones web.** |

**Conclusiones**

[*agregar aquí las conclusiones sobre el proyecto o los resultados finales*]

**Referencias**

Fernández, L. (2023, 13 febrero). Qué significa autenticación y la autorización. *RedesZone*. https://www.redeszone.net/tutoriales/seguridad/diferencias-autenticacion-autorizacion/

Flores, M. (2023, 16 agosto). Control de accesos en seguridad informática, tipos y ejemplos. *Delete Technology*. https://www.deletetechnology.com/blog/control-de-accesos-en-seguridad-inform%C3%A1tica-tipos-y-ejemplos

Fortinet. (2024). *¿Qué son las credenciales de ingreso o inicio de sesión?* https://www.fortinet.com/lat/resources/cyberglossary/login-credentials

Kurth, M. (2024, 26 abril). ¿Qué es la seguridad informática? definición del concepto. *Teclab*. https://teclab.edu.ar/tecnologia-y-desarrollo/que-es-la-seguridad-informatica/

Microsoft. (2024). *¿Qué es la autenticación? definición y métodos*. Seguridad de Microsoft. https://www.microsoft.com/es-mx/security/business/security-101/what-is-authentication

Okta. (2023). *¿Qué es la autenticación? Definición y usos - Auth0*. Auth0. https://auth0.com/es/intro-to-iam/what-is-authentication

Oteir, N. (2023, 3 marzo). ¿En qué se diferencian la autenticación y la autorización? *INTROSERV*. https://introserv.com/es/blog/%C2%BFen-que-se-diferencian-la-autenticacion-y-la-autorizacion/

Rodríguez, J. F. (2019, 31 diciembre). *Implementación de sistema SSO basado en Shibboleth*. https://openaccess.uoc.edu/handle/10609/108146

Toledo, R. (2024, 19 marzo). Control de contraseñas en la seguridad informática. *Cibernos*. https://www.grupocibernos.com/blog/control-de-contrasenas-en-la-seguridad-informatica