PUCE Virtual

CLASE

7

Programación orientada a objetos

Persistencia, auditoría, trazabilidad, patrones e integración en ciberseguridad

QS WORLD UNIVERSITY RANKINGS

Educación de calidad

INTRODUCCIÓN DE LA CLASE

GRADO / POSGRADO / TECNOLOGÍAS

Estudia a tu tiempo
son interrupciones

EDUCACIÓN EN LÍNEA DE CALIDAD

En el desarrollo de sistemas orientados a la ciberseguridad, la persistencia, la auditoría y la trazabilidad son componentes esenciales para garantizar el control y la protección de la información. La persistencia de datos permite almacenar de manera permanente información crítica como usuarios o configuraciones, mientras que la persistencia de eventos registra acciones relevantes del sistema, como intentos de acceso o fallos. En Java, bajo el paradigma de programación orientada a objetos, estos procesos se implementan mediante clases que encapsulan tanto los datos como los mecanismos de almacenamiento, facilitando una estructura modular, mantenible y alineada con buenas prácticas de desarrollo seguro.

Por su parte, la auditoría y la trazabilidad permiten supervisar y analizar el comportamiento del sistema a través del tiempo. La auditoría se enfoca en el registro de eventos importantes, mientras que la trazabilidad permite reconstruir paso a paso las acciones realizadas, lo cual es clave para la detección de incidentes y el análisis forense. El uso de registros o logs constituye una herramienta fundamental en este contexto, ya que proporciona evidencia sobre lo ocurrido dentro del sistema. Implementados en Java mediante clases y estructuras adecuadas, estos mecanismos contribuyen al desarrollo de aplicaciones seguras, capaces de identificar amenazas y responder de manera eficiente ante posibles ataques. En esta clase estudiaremos los temas citados.

13. PERSISTENCIA, AUDITORÍA, TRAZABILIDAD, PATRONES E INTEGRACIÓN EN CIBERSEGURIDAD

En el desarrollo de software actual, no basta con que las aplicaciones funcionen: deben ser seguras, trazables y robustas frente a amenazas cada vez más avanzadas. La programación orientada a objetos permite estructurar estos sistemas de forma modular y mantenible. Al integrar elementos como persistencia, auditoría, trazabilidad y patrones de diseño, se logran aplicaciones más seguras, confiables y controlables ante riesgos.

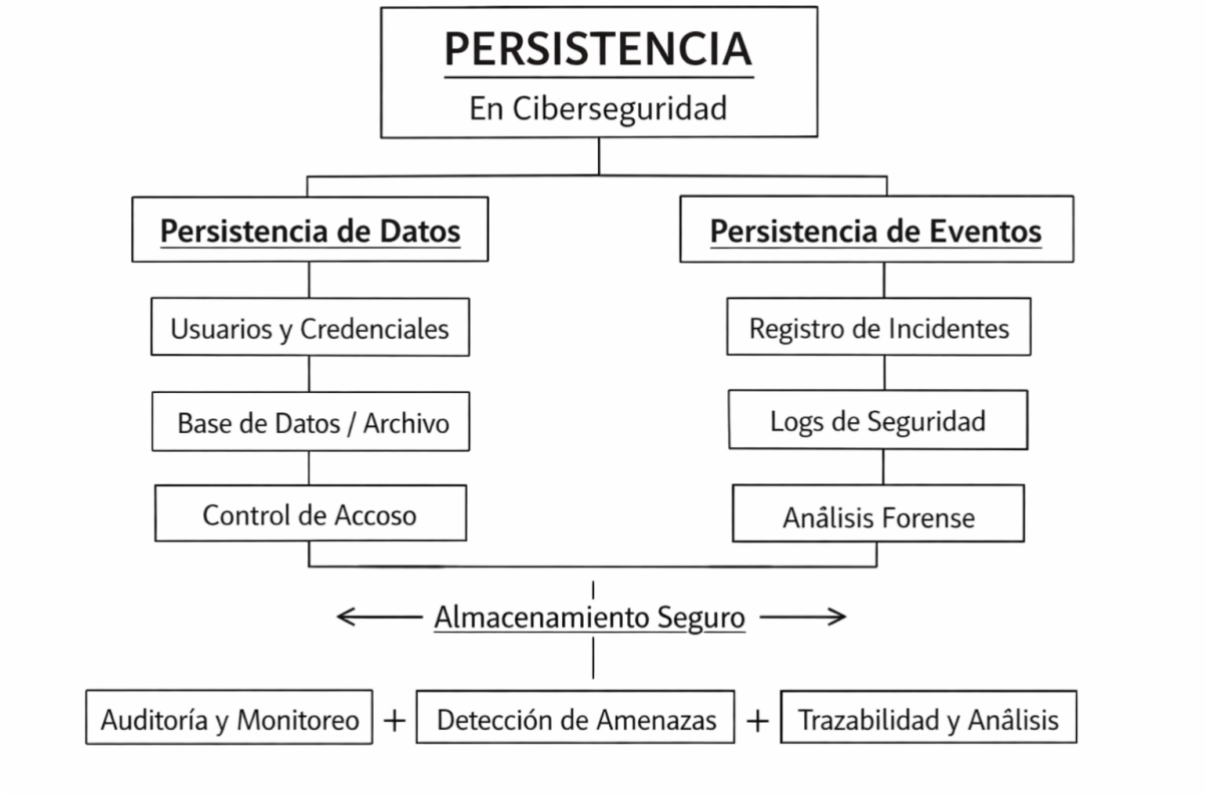
Persistencia

La persistencia en POO es la capacidad de guardar información de forma permanente para su posterior recuperación. En ciberseguridad, es clave porque permite registrar actividades, detectar comportamientos sospechosos y realizar auditorías. Se implementa mediante clases que gestionan el almacenamiento de datos, favoreciendo un diseño modular y mantenible (Hernández Bejarano, 2023).

La figura 1 presenta una visión general de la persistencia en ciberseguridad dentro de un sistema informático. Se distinguen los componentes persistencia de datos y de eventos.

Figura 1

Persistencia en ciberseguridad



Autor: Héctor Avalos Silva

13.1 Persistencia de datos

La persistencia de datos consiste en almacenar información estructurada en medios permanentes como archivos o bases de datos. En sistemas de acceso seguro, esto implica gestionar usuarios, contraseñas (en forma de hash), roles y permisos.

El uso de POO permite separar la lógica de negocio de la lógica de persistencia, lo cual mejora la seguridad y escalabilidad del sistema (Blanco Fernández, 2020).

La figura 2 ilustra cómo los datos persistentes deben ser protegidos, monitoreados y auditados para garantizar su seguridad frente a amenazas.

Figura 2

Persistencia de datos

Autor: Héctor Avalos Silva

Ejemplo: Persistencia de usuarios en un sistema de acceso seguro

a) Clase Usuario

```
//Clase Usuario representa un usuario del sistema
class Usuario {
   private String username;
   private String passwordHash
   private String rol;

   public Usuario(String username, String passwordHash, String rol) {
      this.username = username;
      this.passwordHash = passwordHash;
      this.rol = rol;
   }

   public String getUsername() {
      return username;
   }

   public String getPasswordHash() {
      return passwordHash;
   }

   public String getRol() {
      return rol;
   }
}
```

Esta clase representa la entidad principal del sistema. El uso de hash de contraseña evita almacenar credenciales en texto plano (Oviedo Regino, 2015).

b) Repositorio de usuarios (persistencia en archivo)

```
import java.io.*;

// Clase que gestiona la persistencia de usuarios en archivo
class UsuarioRepository {
```

```
    private static final String FILE = "usuarios.txt";
    public void guardarUsuario(Usuario usuario) {

        // Bloque try-with-resources para asegurar el cierre automático del archivo
        try (BufferedWriter bw = new BufferedWriter(new FileWriter(FILE, true))) {

            // Escribe los datos del usuario separados por comas
            bw.write(usuario.getUsername() + "," +
                    usuario.getPasswordHash() + "," +
                    usuario.getRol());

            bw.newLine();

        } catch (IOException e) {

            e.printStackTrace();
        }
    }

    // Método que busca un usuario en el archivo por su nombre
    public Usuario buscarUsuario(String username) {

        // Bloque try-with-resources para lectura del archivo
        try (BufferedReader br = new BufferedReader(new FileReader(FILE))) {

            String linea;

            // Itera sobre cada línea del archivo
            while ((linea = br.readLine()) != null) {

                // Divide la línea en partes usando la coma como separador
                String[] datos = linea.split(",");

                // Verifica si el username coincide con el buscado
                if (datos[0].equals(username)) {

                    // Retorna un nuevo objeto Usuario con los datos encontrados
                    return new Usuario(datos[0], datos[1], datos[2]);
                }
            }

        // Manejo de errores de lectura
        } catch (IOException e) {

            e.printStackTrace();
        }
        return null;
    }
}
```

Aquí se aplica el patrón Repository, que encapsula el acceso a datos y permite cambiar fácilmente el mecanismo de almacenamiento (Phillips & Plott, 2021).

c) Servicio de autenticación

```
// Clase que gestiona la lógica de autenticación
class AuthService {
```

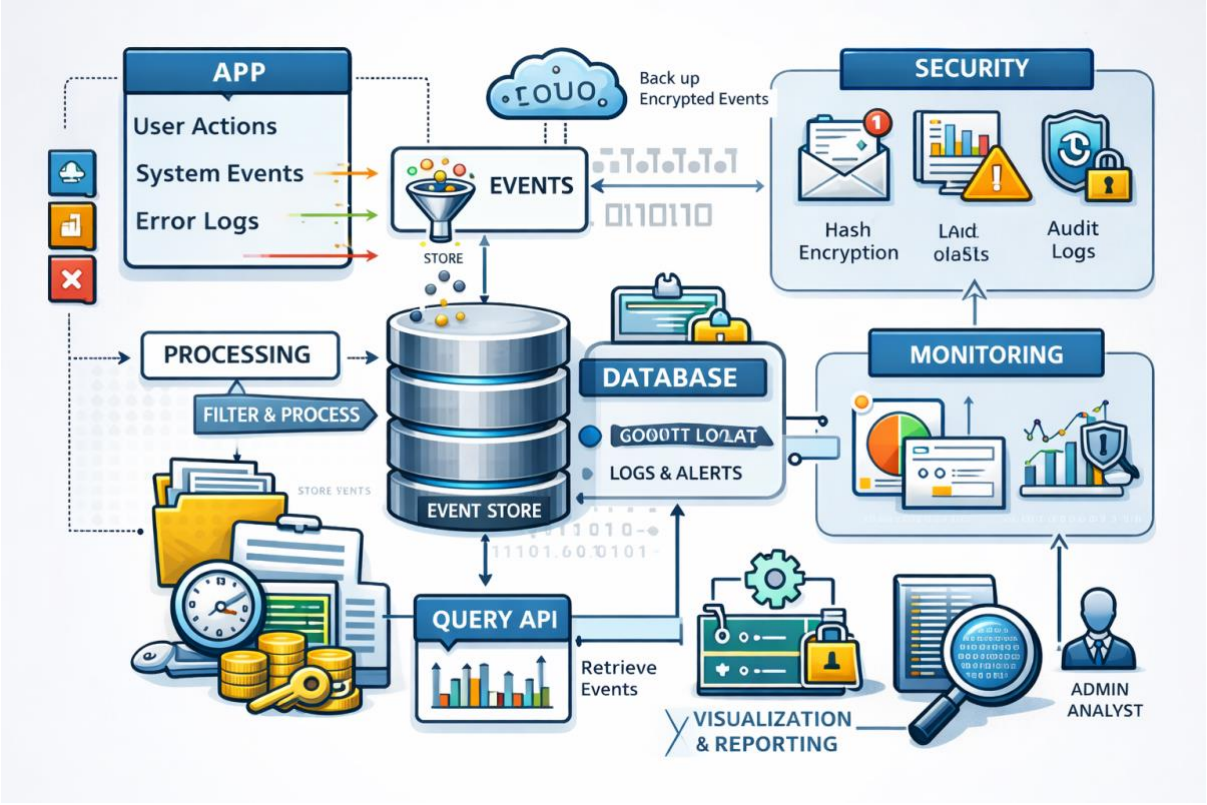
```
    private UsuarioRepository repo;

    public AuthService(UsuarioRepository repo) {

        this.repo = repo;
    }

    public boolean login(String username, String passwordHash) {

        Usuario usuario = repo.buscarUsuario(username);

        if (usuario == null) {
            return false;
        }

        return usuario.getPasswordHash().equals(passwordHash);
    }
}
```

Este servicio separa la lógica de autenticación del almacenamiento, cumpliendo el principio de responsabilidad única (Gervais, 2025).

13.2 Persistencia de eventos

La persistencia de eventos consiste en registrar cada acción relevante del sistema como un evento inmutable.
En ciberseguridad, esto es esencial para auditoría de accesos, detección de ataques (fuerza bruta, accesos no autorizados), análisis forense.
Este enfoque se relaciona con el patrón Event Sourcing, donde el estado del sistema se reconstruye a partir de eventos (Phillips & Plott, 2021).
La figura 3 ilustra cómo los eventos del sistema son capturados, almacenados, monitoreados y protegidos, permitiendo auditoría, trazabilidad y detección de amenazas.

Figura 3
Persistencia de eventos



Autor: Héctor Avalos Silva

Ejemplo: Registro de eventos de seguridad

a) Clase Evento de Seguridad

```
import java.time.LocalDateTime;

// Clase que representa un evento de seguridad (auditoría)
class EventoSeguridad {

   private String usuario;
   // Atributo que almacena el tipo de acción (LOGIN_EXITOSO, LOGIN_FALLIDO)
   private String accion;
   private String ip;
   private LocalDateTime fecha;

   // Constructor que inicializa los atributos del evento
   public EventoSeguridad(String usuario, String accion, String ip) {
      this.usuario = usuario;
      this.accion = accion;
      this.ip = ip;
      this.fecha = LocalDateTime.now();
   }

   public String serializar() {

      return fecha + "|" + usuario + "|" + accion + "|" + ip;
   }
}
```

Los eventos son inmutables, lo que garantiza la integridad de la información registrada (Guardati Buemo, 2016).

b) Almacenamiento de eventos

```
import java.io.*;

// Clase que gestiona la persistencia de eventos de seguridad
class EventStore {

    private static final String FILE = "eventos.log";

    public void guardarEvento(EventoSeguridad evento) {

        // Bloque try-with-resources para escritura segura
        try (BufferedWriter bw = new BufferedWriter(new FileWriter(FILE, true))) {

            // Escribe el evento serializado en el archivo
            bw.write(evento.serializar());
            bw.newLine();

        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Este componente permite mantener un historial completo de actividades, fundamental para auditoría y monitoreo.

Integración de los componentes de persistencia de datos y eventos

```
// Clase que integra autenticación y auditoría
class SistemaAccesoSeguro {

    // Servicio de autenticación
    private AuthService authService;

    // Sistema de almacenamiento de eventos
    private EventStore eventStore;

    // Constructor que recibe las dependencias
    public SistemaAccesoSeguro(AuthService authService, EventStore eventStore) {
        this.authService = authService;
        this.eventStore = eventStore;
    }

    // Método que gestiona un intento de inicio de sesión
    public void intentarLogin(String username, String passwordHash, String ip) {

        // Llama al servicio de autenticación
        boolean autenticado = authService.login(username, passwordHash);

        if (autenticado) {
            System.out.println("Acceso concedido");
            // Registra evento de login exitoso
            eventStore.guardarEvento(
                new EventoSeguridad(username, "LOGIN_EXITOSO", ip)
            );

        } else {

            System.out.println("Acceso denegado");
```

```
            // Registra evento de login fallido
            eventStore.guardarEvento(
                new EventoSeguridad(username, "LOGIN_FALLIDO", ip)
            );
        }
    }
}
// Clase principal del sistema
public class SistemaSeguro {

    public static void main(String[] args) {

        // Crea instancia del repositorio de usuarios
        UsuarioRepository repo = new UsuarioRepository();

        // Crea instancia del servicio de autenticación
        AuthService authService = new AuthService(repo);

        // Crea instancia del almacén de eventos
        EventStore eventStore = new EventStore();

        Usuario usuario = new Usuario("admin", "123hash", "ADMIN");

        repo.guardarUsuario(usuario);

        SistemaAccesoSeguro sistema =
              new SistemaAccesoSeguro(authService, eventStore);

        sistema.intentarLogin("admin", "123hash", "192.168.1.1");
        sistema.intentarLogin("admin", "incorrecto", "192.168.1.2");
    }
}
```

Análisis del enfoque de ciberseguridad

El sistema permite registrar accesos y eventos, identificar fallos y mantener trazabilidad, lo que facilita detectar ataques y analizar incidentes. La persistencia de estos datos permite reconstruir comportamientos, garantizar la integridad de la información y realizar auditorías confiables (Phillips & Plott, 2021).
La tabla 1 resume características de aspectos sobre la persistencia de datos, objetos y eventos.

Tabla 1

Tabla comparativa de la persistencia de datos, objetos y eventos

Aspecto	**Persistencia de Datos**	**Persistencia de Objetos**	**Persistencia de Eventos**
Definición	Almacenamiento permanente de información	Almacenamiento de objetos de la POO	Almacenamiento de acciones o sucesos del sistema
Enfoque	Datos en general (tablas, archivos)	Instancias de clases (objetos Java)	Registros de actividades (logs)
Ejemplo	Usuarios, contraseñas, configuraciones	Objeto Usuario, Cuenta, Producto	Login, intentos fallidos, cambios de sistema
Tecnología común	Bases de datos (MySQL, PostgreSQL)	ORM (JPA, Hibernate)	Logs (Log4j, SIEM, archivos de registro)

Relación con POO	No depende directamente de POO	Totalmente ligado a la POO	Se apoya en POO para estructurar eventos
Uso en ciberseguridad	Protección de información crítica	Gestión estructurada de datos seguros	Auditoría, trazabilidad, detección de ataques
Ventaja principal	Garantiza disponibilidad de datos	Facilita desarrollo modular y mantenible	Permite monitoreo y análisis de incidentes
Riesgo si falla	Pérdida de información	Inconsistencia entre objetos y base de datos	Imposibilidad de detectar o investigar ataques

Autor: Héctor Avalos Silva

Invito a revisar el siguiente video a fin de reforzar el tema de persistencia:

- **Título:** Persistencia y Streams en Java - Teoría
- **Descripción:** Teoría y herramientas básicas para gestionar la persistencia en Java
- **Enlace**: Persistencia y Streams en Java - Teoría

14. AUDITORÍA Y TRAZABILIDAD

En el contexto de la ciberseguridad, la auditoría y la trazabilidad son mecanismos esenciales para garantizar la transparencia, el control y la seguridad de los sistemas informáticos. Desde la programación orientada a objetos, estos conceptos se implementan mediante clases que permiten registrar, almacenar y analizar eventos del sistema, manteniendo una estructura organizada y reutilizable.

La correcta implementación de auditoría y trazabilidad permite detectar incidentes de seguridad, reconstruir acciones realizadas por los usuarios y cumplir con normativas y estándares de seguridad (Hernández Bejarano, 2023)

14.1 Concepto de auditoría en sistemas informáticos y la importancia de la trazabilidad de eventos

Auditoría en sistemas informáticos

La auditoría es el proceso de registrar, monitorear y analizar las acciones que ocurren dentro de un sistema. En ciberseguridad, esto incluye intentos de acceso, cambios en configuraciones, uso de recursos y actividades sospechosas.

El objetivo es garantizar que todas las acciones puedan ser revisadas posteriormente, permitiendo detectar vulnerabilidades o comportamientos anómalos (Oviedo Regino, 2015).

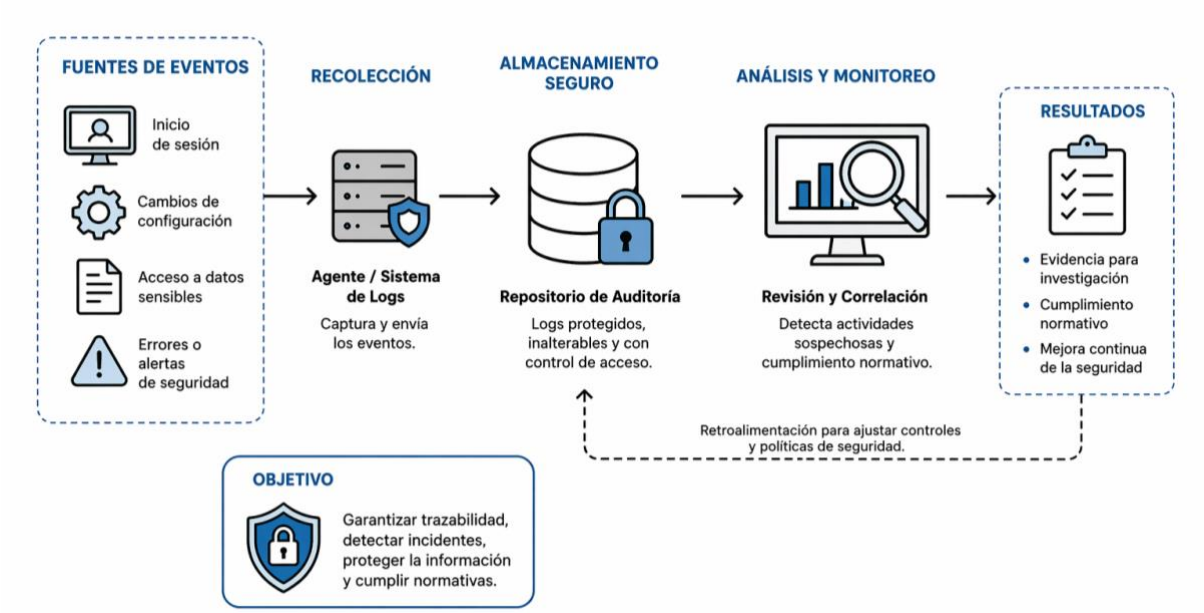
Trazabilidad de eventos

La trazabilidad permite seguir y registrar eventos dentro de un sistema, incluyendo datos como usuario, tiempo y tipo de acción. Es clave para reconstruir incidentes y realizar análisis forense. En POO, se implementa mediante clases que gestionan eventos y separan la auditoría de la lógica del negocio, mejorando la organización y mantenibilidad del código (Blanco Fernández, 2020).

La figura 4 permite tener visibilidad, control y capacidad de respuesta ante amenazas en sistemas desarrollados bajo un enfoque OO.

Figura 4

Auditoría y trazabilidad en ciberseguridad



Autor: Héctor Avalos Silva

Ejemplo: Evento de auditoría

```
import java.time.LocalDateTime;

// Clase que representa un evento auditado
class EventoAuditoria {

   private final String usuario;
   private final String accion;
   private final String ip;
   private final LocalDateTime fecha;

   public EventoAuditoria(String usuario, String accion, String ip) {
      this.usuario = usuario;
      this.accion = accion;
      this.ip = ip;
      this.fecha = LocalDateTime.now();
   }

   // Método que convierte el evento en formato texto
   public String serializar() {
      return fecha + "|" + usuario + "|" + accion + "|" + ip;
   }
}
```

Análisis

- El evento es inmutable, lo que evita manipulación
- Se garantiza integridad de la información
- Se facilita auditoría y trazabilidad

El diseño OO permite modelar eventos como entidades claras y reutilizables (Gervais, 2025).

14.2 Uso de registros para auditoría

Los registros (logs) son archivos o sistemas donde se almacenan los eventos generados por el sistema. En ciberseguridad, los logs permiten detectar intentos de intrusión, analizar fallos de autenticación y monitorear actividad del sistema.

El uso de registros estructurados mejora la capacidad de análisis y respuesta ante incidentes (Guardati Buemo, 2016).

Ejemplo: Registro de auditoría

```
import java.io.*;

// Clase que gestiona los registros de auditoría
class AuditoriaLogger {

   private static final String FILE = "auditoria.log";

   public void registrarEvento(EventoAuditoria evento) {

      try (BufferedWriter bw = new BufferedWriter(new FileWriter(FILE, true))) {

         bw.write(evento.serializar());

         bw.newLine();

      } catch (IOException e) {
         e.printStackTrace();
      }
   }
}
```

Integración con sistema de acceso seguro

```
// Sistema que integra autenticación y auditoría
class SistemaAccesoSeguroAuditoria {

   private AuthService authService;
   private AuditoriaLogger logger;

   public SistemaAccesoSeguroAuditoria(AuthService authService, AuditoriaLogger logger) {
      this.authService = authService;
      this.logger = logger;
   }

   // Método de login con auditoría
   public void intentarLogin(String username, String passwordHash, String ip) {

      boolean autenticado = authService.login(username, passwordHash);
      if (autenticado) {
         System.out.println("Acceso concedido");

         logger.registrarEvento(
            new EventoAuditoria(username, "LOGIN_EXITOSO", ip)
         );
      } else {
         System.out.println("Acceso denegado");
         logger.registrarEvento(
            new EventoAuditoria(username, "LOGIN_FALLIDO", ip)
         );
      }
   }
```

}

Análisis del uso de logs en ciberseguridad

Este enfoque permite registrar accesos, detectar ataques y monitorear actividades, facilitando la respuesta a incidentes y el análisis del sistema. La auditoría y la trazabilidad son claves para garantizar control, transparencia y confiabilidad, mientras que en POO se implementan mediante clases bien estructuradas que permiten construir sistemas seguros, robustos y auditables (Guardati Buemo, 2016).

La tabla 2 presenta aspectos sobre auditoria y trazabilidad enfocada en ciberseguridad.

Tabla 2

Tabla comparativa de aspectos referentes a auditoría y trazabilidad

Aspecto	**Auditoría**	**Trazabilidad**
Definición	Proceso de registro y revisión de eventos del sistema	Capacidad de seguir el rastro de acciones o datos
Objetivo	Evaluar seguridad y detectar irregularidades	Reconstruir lo ocurrido paso a paso
Enfoque	Control y supervisión	Seguimiento detallado
Qué registra	Eventos relevantes (login, cambios, accesos)	Secuencia completa de acciones y su origen
Nivel de detalle	General (eventos importantes)	Alto (historial completo)
Uso en ciberseguridad	Detectar ataques y cumplir normativas	Investigar incidentes y análisis forense
Ejemplo	Registro de intentos de login	Seguimiento de un usuario desde login hasta acciones
Tecnologías comunes	Logs, SIEM, monitoreo	Logs detallados, correlación de eventos
Relación entre ambos	Usa registros para analizar	Se basa en esos registros para reconstruir hechos
Riesgo si falla	No detectar amenazas	No poder entender ni probar un incidente

Autor: Héctor Avalos Silva

La tabla 3 es una comparativa en varios aspectos sobre los temas persistencia, auditoría y trazabilidad enfocados en ciberseguridad.

Tabla 3

Tabla comparativa sobre persistencia, auditoría y trazabilidad

Aspecto	**Persistencia**	**Auditoría**	**Trazabilidad**
Definición	Almacenamiento permanente de datos	Registro y revisión de eventos del sistema	Seguimiento del rastro de acciones o datos
Objetivo	Garantizar disponibilidad e integridad de la información	Detectar irregularidades y evaluar seguridad	Reconstruir eventos y entender incidentes
Enfoque	Guardar información	Control y monitoreo	Seguimiento detallado
Qué gestiona	Datos, objetos y eventos	Eventos relevantes	Secuencia completa de eventos
Uso en ciberseguridad	Proteger datos críticos	Detectar ataques y cumplir normativas	Análisis forense y seguimiento de incidentes
Ejemplo	Base de datos de usuarios o logs	Registro de intentos de acceso	Ruta completa de un ataque desde su origen

Tecnologías comunes	Bases de datos, JPA, Hibernate	Logs, SIEM, monitoreo	Logs correlacionados, herramientas de análisis
Nivel de detalle	Medio	Medio	Alto
Dependencia	Base para los demás	Depende de la persistencia	Depende de auditoría y persistencia
Riesgo si falla	Pérdida de información	No detectar amenazas	No poder analizar ni reconstruir incidentes

Autor: Héctor Avalos Silva

Accede al siguiente video para reforzar el tema de auditoría y su funcionamiento:

- **Título:** Auditoría Informática Qué Es, Cómo Funciona y Por Qué Es CRUCIAL en el Mundo Digital
- **Descripción:** El video explica el concepto de auditoría informática, cómo funciona, por qué es importante.
- **Enlace:** Auditoría Informática Qué Es, Cómo Funciona y Por Qué Es CRUCIAL en el Mundo Digital

Glosario:

Blanco Fernández, Y. (2020). *Introducción a la programación orientada a objetos.* Andavira.

Gervais, L. (2025). *Aprender Programación Orientada a Objetos con Java.* (eni, Ed.) https://doi.org/978-2-409-05028-2

Guardati Buemo, S. (2016). *Estructuras de datos básicas: Programación orientada a objetos con Java.* Alfaomega.

Hernández Bejarano, M. &. (2023). *Programación Orientada a Objetos en Java: Buenas prácticas. Ingeniería de sistemas.* https://doi.org/978-958-792-463-3

Oviedo Regino, E. M. (2015). *Lógica de programación orientada a objetos.* Ecoe Ediciones. https://doi.org/978-958-711-136-3

Phillips, D., & Plott, S. F. (2021). *Python Object-Oriented Programming: Build robust and maintainable object-oriented Python applications and libraries* (4ta ed.). Packt Publishing Limited. https://doi.org/978-1801077262

Definiciones:

- **Persistencia:** La persistencia es la capacidad de un sistema o aplicación para almacenar y conservar datos de manera permanente, de modo que estos puedan recuperarse y utilizarse incluso después de que el sistema se apague o se reinicie.
- **Auditoría:** La auditoría es el proceso de registrar, revisar y analizar las actividades de un sistema con el fin de verificar su correcto funcionamiento, detectar errores o identificar posibles problemas de seguridad.

RECURSOS DE PROFUNDIZACION

Recurso 1:

- **Título:** Persistencia en ciberseguridad con Java con paradigma orientado a objetos.
- **Descripción:** Se desarrolla un programa en java que permite almacenar y gestionar credenciales de usuarios de manera persistente y con un nivel básico de seguridad.

- **Documento:** Recurso de profundización 1 Ejercicio peristencia.docx

Recurso 2:

- **Título:** Auditoría y trazabilidad con base de datos (SQLite vía JDBC), detección automática de fuerza bruta y ataques.
- **Descripción:** El programa realiza auditoría y trazabilidad con base de datos (SQLite vía JDBC), y realiza detección automática de fuerza bruta y detección de ataques.
- **Documento:** Recurso de profundización 2 Ejercicio auditoria.docx

Actividades de evaluacion

PREGUNTAS DE AUTOEVALUACION

INSTRUCCIONES	Responda las siguientes preguntas de opción múltiple seleccionando la alternativa correcta en cada caso. Cada pregunta presenta un contexto relacionado con ciberseguridad, persistencia, auditoría y trazabilidad en sistemas desarrollados con Java y programación orientada a objetos.					
Titulo	Evaluación sobre Persistencia, Auditoría y Trazabilidad en Ciberseguridad					
Dificultad		Baja	X	Media		Alta
RDAs Evaluados:		RDA 1				
		RDA 2				
	X	RDA 4				
Secciones evaluadas		13. Persistencia, auditoría, trazabilidad. 13.1 Persistencia de datos 13.2 Persistencia de eventos 14. Auditoría y trazabilidad 14.1 Concepto de auditoría en sistemas informáticos y la importancia de la trazabilidad de eventos. 14.2 Uso de registros para auditoría				
PREGUNTA 1	Un sistema desarrollado en Java almacena información de usuarios en una base de datos para garantizar que los datos no se pierdan al cerrar la aplicación. ¿Qué concepto de ciberseguridad se está aplicando principalmente?					
CORRECTA	Persistencia de datos					
Incorrecta 1	Auditoría de eventos					
Incorrecta 2	Trazabilidad de acciones					
Incorrecta 3	Monitoreo en tiempo real					
Retroalimentación de las respuestas						

CORRECTA	La persistencia de datos permite almacenar información de manera permanente, asegurando su disponibilidad. Según Hernández Bejarano & Baquero Rey (2023), es fundamental en sistemas seguros.
Incorrecta 1	La auditoría se enfoca en registrar eventos, no en almacenar datos de forma permanente.
Incorrecta 2	La trazabilidad permite seguir el rastro de acciones, no almacenar información.
Incorrecta 3	El monitoreo en tiempo real implica supervisión continua, no almacenamiento de datos.
PREGUNTA 2	Un sistema registra cada intento de inicio de sesión (exitoso o fallido) junto con la IP y la fecha. ¿Qué tipo de persistencia se está aplicando?
CORRECTA	Persistencia de eventos
Incorrecta 1	Persistencia de datos
Incorrecta 2	Persistencia de objetos
Incorrecta 3	Persistencia temporal
Retroalimentación de las respuestas	
CORRECTA	La persistencia de eventos registra acciones del sistema, lo cual es clave para la seguridad y el análisis posterior (Plott & Phillips, 2021).
Incorrecta 1	La persistencia de datos se refiere a almacenar información como usuarios, no eventos.
Incorrecta 2	La persistencia de objetos es una forma de almacenar estructuras, pero no se centra en eventos de seguridad.
Incorrecta 3	La persistencia implica almacenamiento permanente, no temporal.
PREGUNTA 3	Un administrador analiza los registros para identificar una secuencia de intentos fallidos desde una misma IP. ¿Qué concepto está utilizando principalmente?
CORRECTA	Trazabilidad
Incorrecta 1	Persistencia
Incorrecta 2	Cifrado
Incorrecta 3	Autenticación
Retroalimentación de las respuestas	
CORRECTA	La trazabilidad permite reconstruir acciones y analizar eventos, siendo clave para detectar ataques (Plott & Phillips, 2021).
Incorrecta 1	La persistencia solo almacena datos, no permite seguir el rastro de eventos.

Incorrecta 2	El cifrado protege información, pero no permite analizar secuencias de eventos.
Incorrecta 3	La autenticación valida identidad, no analiza comportamientos.
PREGUNTA 4	En un sistema seguro, los logs se utilizan para registrar accesos, errores y acciones del usuario. ¿Cuál es el propósito principal de estos registros?
CORRECTA	Permitir auditoría y análisis de seguridad
Incorrecta 1	Aumentar la velocidad del sistema
Incorrecta 2	Reducir el uso de memoria
Incorrecta 3	Eliminar errores automáticamente
Retroalimentación de las respuestas	
CORRECTA	Los logs permiten auditoría, monitoreo y análisis de incidentes, siendo esenciales en ciberseguridad (Guardati Buemo, 2007).
Incorrecta 1	Los logs no mejoran directamente la velocidad del sistema.
Incorrecta 2	Registrar eventos puede incluso aumentar el uso de recursos.
Incorrecta 3	Los logs no corrigen errores, solo los registran para análisis.

PUCE Virtual

síguenos en:

www.pucevirtual.puce.edu.ec