19-2-2024

Lineamientos

Para el desarrollo de software

Cuatrimestre: 8

Grupo: C

Carrera: IDGS

Profesor: Narváez

Figueroa José Christian

Presentan:

Bahena Castillo Luis Eduardo

Barrios Tecorral Oscar Miguel

Mata Nieto Iván Samuel

Reynoso Macedo Brayan

Rodriguez Rodriguez Cristian

Rogel Valentin Diego Jared

UNIVERSIDAD TECNOLÓGICA
EMILIANO ZAPATA DEL ESTADO DE MORELOS

División Académica
de Tecnologías de la
Información y Comunicación

Contenido

Lineamientos de seguridad en el desarrollo:

Consideraciones generales:

- **Seguridad desde el diseño:** La seguridad debe ser una consideración fundamental desde las primeras etapas del desarrollo.
- **Enfoque en la defensa en profundidad:** Implementar múltiples capas de seguridad para minimizar el impacto de un ataque.
- **Minimizar el acceso:** Otorgar a los usuarios solo los permisos necesarios para realizar sus tareas.
- **Mantener el software actualizado:** Aplicar parches de seguridad y actualizaciones de software con regularidad.
- **Capacitación del personal:** Asegurar que el equipo de desarrollo comprenda las mejores prácticas de seguridad.

Implementación de librerías y frameworks de seguridad:

- Utilizar librerías y frameworks de seguridad confiables y con buen soporte.
- Configurar correctamente las librerías y frameworks según las necesidades del proyecto.
- Mantener las librerías y frameworks actualizados a la última versión.

Consultas seguras de base de datos:

- Utilizar siempre parámetros para las consultas SQL y evitar concatenar cadenas.
- Validar la entrada del usuario para prevenir ataques de inyección SQL.

Validación de datos:

- Validar todos los datos de entrada del usuario para prevenir ataques de scripting entre sitios (XSS), inyección de código y otros tipos de ataques.
- Implementar reglas de validación para cada tipo de dato (longitud, formato, valores permitidos).
- Utilizar mensajes de error descriptivos para ayudar al usuario a corregir los errores.

Implementación de mecanismos de autenticación:

- Utilizar un mecanismo de autenticación robusto, como JWT.
- Almacenar las contraseñas de forma segura utilizando un algoritmo de hash.

Manejo de sesiones:

- Invalidar las sesiones después de un período de inactividad.
- Implementar un mecanismo de "token jwt" para proteger las acciones que modifican datos.

Calidad de código:

- Escribir código limpio, bien documentado y fácil de entender.
- Realizar pruebas de seguridad estáticas y dinámicas para detectar vulnerabilidades.
- Implementar un proceso de revisión de código para asegurar la calidad del mismo.

Creación de base de datos y datos iniciales:

- Crear la base de datos con los permisos de acceso mínimos necesarios.
- Insertar los datos iniciales de forma segura utilizando scripts SQL parametrizados.

- Eliminar los datos de prueba y scripts de desarrollo antes de la producción.

Manejo de errores y excepciones:

- Manejar las excepciones de forma segura para evitar que se revele información sensible.
- No mostrar mensajes de error que revelen información sensible al usuario final.

Tecnologías de preferencia o recomendadas:

Considerando que vamos a codificar aplicaciones web, hemos seleccionado estas tecnologías como las mejores para esto

Lenguaje y Framework para Backend:

- **Java 11 o posterior:** Lenguaje robusto y escalable para aplicaciones web empresariales.
- **Spring Boot 2 o posterior:** Framework de desarrollo Java que facilita la creación de aplicaciones web Spring.

Dependencias para Backend:

- **Spring Boot 2 o superior:** Framework Java que simplifica la creación de aplicaciones web Spring.
- **Spring Data JPA:** Facilita el acceso a bases de datos relacionales mediante JPA.
- **Spring Boot Starter Web:** Proporciona configuraciones predeterminadas para crear aplicaciones web RESTful.
- **Spring Boot Starter Thymeleaf:** Integra el motor de plantillas Thymeleaf para generar vistas web.
- **Spring Boot Starter Validation:** Permite la validación de datos de entrada.

- **Spring Boot Starter Mail:** Habilita el envío de emails desde la aplicación.
- **JJWT 0.9.1 o superior:** Librería para generar y validar tokens JWT (opcionales para autenticación basada en tokens).
- **Spring Boot Starter Security:** Añade funcionalidades de seguridad como autenticación y autorización (opcional).
- **Lombok:** Elimina código boilerplate Java (opcional).
- **MySQL Connector Java 8 o superior:** Controlador JDBC para bases de datos MySQL.

Lenguaje y Framework para Frontend:

- **Vue.js 2:** Framework JavaScript popular para crear interfaces de usuario interactivas.
- **JavaScript:** Lenguaje esencial para el desarrollo web front-end.

Dependencias para Frontend

- **Bootstrap 4 o superior:** Framework CSS para crear interfaces de usuario atractivas y receptivas.
- **Axios 0.27 o superior:** Librería para realizar solicitudes HTTP.
- **Vee-validate 4.4 o superior:** Librería para la validación de formularios.
- **Moment.js 2.29 o superior:** Librería para manipular fechas y horas.
- **Lodash 4.17 o superior:** Librería de utilidades JavaScript.

Base de datos:

- **MySQL 8 o posterior:** Base de datos relacional popular por su facilidad de uso y bajo costo.

Consideraciones:

- Esta combinación de tecnologías ofrece un buen equilibrio entre rendimiento, escalabilidad y facilidad de desarrollo.
- MySQL es una base de datos bien documentada y con una amplia comunidad de usuarios.
- Spring Boot facilita la creación de aplicaciones web Spring con una configuración mínima.
- Vue.js es un framework ligero y flexible que permite crear interfaces de usuario modernas y atractivas.
- JavaScript es un lenguaje esencial para el desarrollo web front-end y ofrece una amplia gama de bibliotecas y frameworks.
- Las dependencias elegidas son adecuadas para desarrollar una aplicación web Spring Boot completa con acceso a base de datos, validación, plantillas, envío de emails y funcionalidades opcionales de seguridad y autenticación.
- Versión de Spring Boot Starter Mail: La versión 2.6.4 está desactualizada. Se recomienda actualizar a la versión más reciente (actualmente 2.7.7).
- Dependencias opcionales: Puedes eliminar Lombok y Spring Boot Starter Security si no las necesitas.

Uso de tecnologías:

Lenguaje:

- **Java:** Lenguaje de programación robusto y escalable para aplicaciones web empresariales.
- **Spring Boot:** Framework Java que facilita la creación de aplicaciones web Spring.
- **Spring Data JPA:** Facilita el acceso a bases de datos relacionales mediante JPA.
- **Spring Boot Starter Web:** Proporciona configuraciones predeterminadas para crear aplicaciones web RESTful.
- **Spring Boot Starter Thymeleaf:** Integra el motor de plantillas Thymeleaf para generar vistas web.
- **Spring Boot Starter Validation:** Permite la validación de datos de entrada.
- **Spring Boot Starter Mail:** Habilita el envío de emails desde la aplicación.
- **JJWT:** Librería para generar y validar tokens JWT (opcionales para autenticación basada en tokens).
- **Spring Boot Starter Security:** Añade funcionalidades de seguridad como autenticación y autorización (opcional).
- **Lombok:** Elimina código boilerplate Java (opcional).

Patrones de diseño:

- **Patrón Service:** Define una capa de negocio que encapsula la lógica de la aplicación.
- **Patrón Repository:** Se encarga de almacenar y recuperar datos de la base de datos.

Almacenamiento:

- **MySQL:** Base de datos relacional popular por su facilidad de uso y bajo costo.

Tipos de consultas:

- **Consultas SELECT:** Se utilizan para recuperar datos de la base de datos.
- **Consultas INSERT:** Se utilizan para insertar datos en la base de datos.
- **Consultas UPDATE:** Se utilizan para actualizar datos en la base de datos.
- **Consultas DELETE:** Se utilizan para eliminar datos de la base de datos.
- **Consultas JOIN:** Se utilizan para combinar datos de dos o más tablas.

Uso de repositorios de integración

1. Repositorios:

- Tipo: GitHub
- Acceso: Leer/escribir para el equipo.
- Estructura: Reflejar la arquitectura del proyecto.

2. Almacenamiento:

- Límite: 1 GB por repositorio.
- Eliminación: Archivos sin modificar en más de 1 año.

3. Ramas:

- Estrategia: Feature branching.
- Nomenclatura: Nombre de la característica.
- Merges: En la rama principal antes de producción.

4. Commits:

- Comentarios: Descripción breve del cambio.
- Revisiones: Requeridas antes de fusionar en la rama principal.

Lineamientos para el Desarrollo y Consumo de APIs:

Integración de Servicios:

- **Tipo de integración:** REST
- **Formato de datos:** JSON / parámetros de query http

Autenticación de Servicios:

- **Mecanismos:** JWT
- **Esquemas de autorización:** Bearer
- **Niveles de granularidad:** Por usuario, por recurso, por acción
- **Gestión de claves:** Almacenamiento seguro

Interoperabilidad:

- **Especificación de rutas:**
  - Recursos: Nombres en plural, sustantivos
  - Acciones: Verbos en infinitivo
  - Parámetros: En la ruta o en el cuerpo de la solicitud
  - Códigos de estado HTTP: Indicar el resultado de la operación

- **Paginación:**

Spring Data JPA ofrece el objeto Page, una interfaz que encapsula información esencial para la paginación:

  - Contenido total: El número total de elementos en el conjunto de datos.
  - Número de elementos por página: El tamaño de cada página (por ejemplo, 10, 20, 50).
  - Número de página actual: El índice de base cero de la página actual que se está mostrando (comienza en 0).
  - Contenido de la página actual: Una lista de objetos que representan los datos en la página actual.
  - Información adicional: Dependiendo de la implementación, el objeto Page también podría proporcionar detalles de navegación como enlaces a la siguiente y a la anterior página.

Uso del idioma en el código:

- **Nombres de variables y funciones:** Descriptivos, en español
- **Comentarios:** Explicativos, en español
- **Mensajes de error:** Claros y concisos, en español
- **Documentación:** Completa y actualizada, en español

Pruebas:

- Pruebas de integración
- Pruebas de seguridad

Seguridad:

- Validación de entrada y salida
- Protección contra ataques XSS

Aseguramiento de la calidad de software

Caso de Uso	Peligro	Amenaza	Impacto	Probabilidad	Control
Acceso al Sistema	Robo de credenciales	Actores maliciosos podrían obtener las credenciales de los usuarios mediante ataques de fuerza bruta, phishing o malware.	Pérdida de acceso al sistema, robo de información personal o financiera.	Alta	Implementar mecanismos de autenticación fuertes, usar contraseñas seguras y únicas, educar a los usuarios sobre las amenazas de seguridad.
Cierre de Sesión	Sesiones no cerradas	Los usuarios podrían dejar sus sesiones abiertas sin supervisión, lo que permitiría el acceso no autorizado al sistema.	Acceso no autorizado al sistema, robo de información personal o financiera.	Media	Implementar políticas de tiempo de espera de sesión, ofrecer la opción de cerrar sesión automáticamente, educar a los

Caso de Uso	Peligro	Amenaza	Impacto	Probabilidad	Control
					usuarios sobre la importancia de cerrar sesión correctamente.
Gestión de Perfil	Manipulación de datos	Actores maliciosos podrían modificar los datos de perfil de los usuarios para suplantarlos o cometer fraudes.	Robo de identidad, acceso no autorizado a información personal o financiera.	Media	Implementar controles de acceso a la información personal, permitir la edición de datos solo después de la verificación de identidad, realizar auditorías de seguridad.

Caso de Uso	Peligro	Amenaza	Impacto	Probabilidad	Control
Ver Eventos Pendientes	Denegación de servicio	Un ataque DDoS podría sobrecargar el sistema y evitar que los usuarios vean sus eventos pendientes.	Interrupción del servicio, pérdida de información sobre eventos importantes.	Media	Implementar medidas de protección DDoS, escalar la infraestructura según sea necesario, monitorear el rendimiento del sistema.
Marcar Eventos como Completados	Falsificación de datos	Actores maliciosos podrían modificar el estado de los eventos para obtener beneficios indebidos.	Pérdida de integridad de la información, fraude.	Baja	Implementar mecanismos de verificación de datos, registrar auditorías de las modificaciones a los eventos, realizar análisis

Caso de Uso	Peligro	Amenaza	Impacto	Probabilidad	Control
					de patrones de comportamiento.
Gestión de Usuarios (Administrador)	Abuso de privilegios	Los administradores podrían utilizar sus privilegios para acceder a información confidencial o realizar acciones no autorizadas.	Robo de información, daño al sistema, fraude.	Baja	Implementar el principio de mínimo privilegio, realizar auditorías de las acciones de los administradores, educar a los administradores sobre la responsabilidad de sus roles.

Caso de Uso	Peligro	Amenaza	Impacto	Probabilidad	Control
Gestión de Servicios (Administrador)	Configuración incorrecta	La configuración incorrecta de los servicios podría generar errores, vulnerabilidades o interrupciones del servicio.	Interrupción del servicio, pérdida de datos, vulnerabilidades de seguridad.	Media	Implementar procesos de control de cambios, realizar pruebas exhaustivas antes de la implementación de nuevos servicios, documentar los procedimientos de configuración.

Caso de Uso	Peligro	Amenaza	Impacto	Probabilidad	Control
Gestión de Paquetes de Servicios (Administrador)	Errores en la definición de paquetes	Los errores en la definición de los paquetes de servicios podrían generar confusiones o problemas en la prestación de los servicios.	Insatisfacción del cliente, errores en la facturación, problemas en la entrega de servicios.	Media	Implementar procesos de validación de la información de los paquetes, realizar pruebas exhaustivas antes de la publicación de nuevos paquetes, documentar claramente las características de los paquetes.

Caso de Uso	Peligro	Amenaza	Impacto	Probabilidad	Control
Visualización de Presentación de Paquetes (Administrador)	Información engañosa	Los paquetes de servicios podrían presentar información incorrecta o incompleta, lo que podría llevar a los usuarios a tomar decisiones equivocadas.	Insatisfacción del cliente, pérdida de clientes, problemas de reputación.	Media	Implementar procesos de validación de la información de los paquetes, presentar la información de manera clara y precisa, ofrecer mecanismos de comparación de paquetes.