Universidad de Girona
Escuela Politécnica Superior
Grado en Ingeniería Informática

# La seguridad como punto de partida del desarrollo web

Resumen

*Autor*
Sergi Bergillos Pedraza

*Tutor*
Antonio Bueno Delgado

Septiembre de 2021

# Capítulo 1

# Introducción

La seguridad y la privacidad de los datos personales es un derecho del usuario y una responsabilidad de las empresas tal y como recogen numerosas leyes o reglamentos a nivel europeo o español.

Sin embargo, el conocimiento adquirido durante el grado de ingeniería informática de la Universidad de Gerona (UdG) es insuficiente para preparar a los recién graduados para los desafíos del mundo laboral en temas de ciberseguridad.

A pesar de que este conocimiento puede ampliarse de forma autónoma en internet, la gran cantidad de información disponible al respecto puede resultar abrumadora.

El objetivo de este proyecto es de carácter divulgativo y didáctico: documentar los diferentes tipos de ciberataques contra sitios web. cómo se producen y cómo protegerse; y proporcionar una serie de consejos y buenas prácticas para diseñar e implementar un sitio web seguro.

Se implementará, además, dos plantillas en los *stacks* MEVN (MongoDB, Express.js, Vue.js y Node.js) y LAMP (Linux, Apache, MySQL y PHP) para realizar guías de implementación de las diferentes buenas prácticas y las medidas de protección descritas. Después, estas dos plantillas podrán usarse como un punto de partida de un sitio web seguro.

Dado el carácter divulgativo y didáctico del proyecto, la documentación se ha hecho en castellano y el documento y el código realizado se han licenciado con una licencia de código abierto.

# Capítulo 2

# Desarrollo

El primer paso que se ha hecho para desarrollar el proyecto ha sido el estudio de la viabilidad con el sistema TELOS, que puede encontrarse en el «Capítulo 2. Estudio de viabilidad» de la memoria, concretamente en la página 7.

TELOS es un acrónimo que define cinco factores esenciales para determinar la viabilidad del proyecto: «Technical», ¿es el proyecto técnicamente posible?; «Economical», ¿es el proyecto económicamente rentable?; «Legal», ¿es el proyecto legal?; «Operational», ¿cómo afecta el proyecto a las operaciones actuales?; y «Scheduling», ¿se puede terminar el proyecto a tiempo?

Una vez considerado viable, se ha escogido el sistema Kanban como metodología para gestionar el trabajo porque al no tratarse de un proyecto de final de grado cuyo núcleo es la implementación de una aplicación se ha descartado el uso de metodologías más encaradas a la programación como pueden ser los métodos Cascada o Agile. El funcionamiento y los componentes básicos del sistema pueden encontrarse en el «Capítulo 3. Metodología», pág. 11.

Después, en el «Capítulo 4. Planificación», pág. 13, se ha utilizado el sistema SMART para describir los tres objetivos principales del proyecto: «Recopilación de buenas prácticas para un desarrollo web seguro», «Recopilación de ciberataques: cómo se producen y cómo protegerse» y «Diseño e implementación de las plantillas de un sitio web seguro».

SMART es un acrónimo de: «Specific», el objetivo ha de especificarse tanto como sea posible; «Measurable», el objetivo ha de ser cuantificable; «Attainable», el objetivo ha de ser realista; «Relevant», el objetivo ha de merecer la pena; «Time-bound», el objetivo ha de tener una fecha límite o un final definido.

Esto ha permitido desglosar los objetivos más abstractos del proyecto en una lista de tareas realizables. La estimación inicial de estas tareas puede encontrarse en la «Tabla 4.1. Lista de tareas», pág. 18, donde se han agrupado por categoría y se han dado una estimación en horas para su realización.

El siguiente paso en el desarrollo del trabajo ha sido el «Capítulo 5. Marco de trabajo y conceptos previos», pág. 21, en el que se ha explicado la situación de la seguridad informática en la actualidad; y los conceptos considerados imprescindibles para seguir el trabajo, como pueden ser las *cookies*. Si bien, se han ido añadiendo nuevos conceptos a medida que se avanzaba el proyecto.

Un aspecto fundamental de este proyecto es su carácter divulgativo y educativo. Por este motivo, se ha dedicado el «Capítulo 6. Estudio y decisiones», pág. 33, a comentar por qué se ha escogido la licencia de código abierto MIT para las plantillas y se ha licenciado la memoria con una Creative Commons extremadamente permisiva como CC BY 4.0.

Además, para facilitar la cooperación con posibles colaboradores futuros, se han especificado varios estándares para Git, JavaScript y PHP, pág. 34-38; y definido una «Guía de Contribución» y un «Código de conducta» basado en el de Ruby, pp 127-128.

En la última sección de este capítulo, se ha defendido la elección de los dos *stacks* en que se han realizado las guías.

Finalmente, se han realizado las tareas de la «Tabla 4.1. Lista de Tareas», pág. 18, utilizando Trello[1] como herramienta para gestionar el tablero Kanban. El flujo habitual ha sido: 1) investigar la buena práctica o ciberataque; 2) documentar la buena práctica o ciberataque en la memoria; 3) investigar las opciones disponibles en MEVN y LAMP para implementar la buena práctica o las medidas de protección frente el ciberataque; y 4) realizar las guías específicas para las dos plantillas.

Así, los capítulos «7. Recopilación de buenas prácticas para un desarrollo seguro», «8. Recopilación de ciberataques: qué son, cómo se producen y como protegerse» y «10. Implementación y pruebas» han avanzado más o menos en paralelo.

Concretamente, la recopilación de buenas prácticas incluye: «El protocolo HTTPS», «Registro», «Tratamiento de los errores», «Almacenamiento de datos sensibles» y la «Validación de las entradas», pp 43-62.

Y la recopilación de ciberataques incluye: «Ataque DoS», «Ataque de credenciales», «Ataque CSRF», «Ataque XSS» y, por último, «Ataque de inyección de código», pp 63-76.

Por último, las dos plantillas pueden encontrarse en el Github del autor: *cardona-node is the server-side application of the Cardona MEVN stack*[2] y *cardona-lamp is the server-side application of the Cardona LAMP stack*[3].

# Capítulo 3

# Conclusiones

Se han tratado doce de las veinte cuestiones que la Fundación OWASP define en *OWASP Proactive Controls*[4] y *OWASP Top Ten Web Application Security Risks*[5]:

- A1:2017-Injection: se han explicado los diferentes tipos de ataques de inyección de código y las diversas medidas de protección que el desarrollador puede aplicar en la sección «8.5. Ataque de inyección de código», pág. 74.
- A3:2017-Sensitive Data Exposure: se ha explicado como almacenar los datos sensibles de forma segura y legal en la sección «7.4. Almacenamiento de datos sensibles», pág. 54.
- A6:2017-Security Misconfiguration: se ha explicado la correcta configuración HTTP y del tratamiento de los errores en las secciones «7.1. El protocolo HTTPS», pág. 43, y «7.3. Tratamiento de los errores», pág. 50, respectivamente. Dos casos muy habituales de errores de configuración en las aplicaciones.
- A7:2017-Cross-Site Scripting XSS: se ha explicado este ataque, y sus posibles prevenciones y mitigaciones, en la sección «8.4. Ataque XSS», pág. 72.
- A10:2017-Insufficient Logging & Monitoring: se ha explicado esta debilidad en la sección «7.2. Registro», pág. 47.
- C1: Define Security Requirements: este es el objetivo principal del proyecto, ofrecer a los posibles desarrolladores web una serie de buenas prácticas y medidas de protección frente ciberataques para un desarrollo web seguro.
- C2: Leverage Security Frameworks and Libraries: aunque no se ha explicado este tema en ningún momento, sí que los *frameworks* escogidos y bibliotecas utilizados son mantenidos con regularidad y no se les conocen vulnerabilidades.
- C4: Encode and Escape Data: se ha explicado junto con el siguiente control y en diversos ataques como una protección frente ataques de inyección de código.
- C5: Validate All Inputs: se ha explicado la validación de entradas en la sección «7.5. validación de las entradas», pág. 60.
- C8: Protect Data Everywhere: se ha explicado el almacenamiento de datos sensibles en la sección «7.4. Almacenamiento de datos sensibles», pág. 54.

- C9: Implement Security Logging and Monitoring: se ha explicado el registro de los eventos de la aplicación en la sección «7.2. Registro», pág. 47.
- C10: Handle All Errors and Exceptions: se ha explicado el tratamiento de errores en la sección «7.3. Tratamiento de los errores», pág. 50.

Aunque la experiencia obtenida durante la realización del proyecto ha sido sin duda positiva, no ha estado libre de errores. Algunos de estos son: estimación incorrecta de las horas necesarias para realizar las tareas de investigación e implementación; falta de un estándar para la escritura del documento; y falta de una hoja de ruta concreta con el orden de las tareas a realizar.

Asimismo, también hay múltiples formas en el que trabajo se puede ampliar o mejorar: finalizar la lista de tareas de la «Tabla 4.1 Lista de Tareas», pág. 18; adaptar las recopilaciones y guías a formato Markdown para añadirlas a la Wiki del proyecto en el Github; realizar nuevas guías en otros *stacks* como WISA (Windows, IIS, SQL Server y ASP.NET) o Spring Boot (máquina virtual de Java, Tomcat, Spring Data y Java); e implementar un *framework* inspirando en Laravel con Node.js y Express.

# Bibliografía


[1] Sergi Bergillos Pedraza. *Tablero Kanban: La seguridad como punto de partida del desarrollo web*. Disponible en https://trello.com/b/YKGkYXTH/tablero-kanban-la-seguridad-como-punto-de-partida-del-desarrollo-web. Accedido por última vez el 1 de septiembre de 2021 (vid. pág. 3).

[2] Sergi Bergillos Pedraza. *cardona-node is the server-side application of the Cardona MEVN stack*. Disponible en https://github.com/SBergillos/cardona-node. Accedido por última vez el 22 de julio de 2021 (vid. pág. 3).

[3] Sergi Bergillos Pedraza. *cardona-lamp is the server-side application of the Cardona LAMP stack*. Disponible en https://github.com/SBergillos/cardona-lamp. Accedido por última vez el 21 de agosto de 2021 (vid. pág. 3).

[4] OWASP Contributors. *OWASP Proactive Controls*. Disponible en https://owasp.org/www-project-proactive-controls/. Accedido por última vez el 3 de agosto de 2021 (vid. pág. 4).

[5] OWASP Foundation, Inc. *OWASP Top Ten Web Application Security Risks | OWASP*. Disponible en https://owasp.org/www-project-top-ten/. Accedido por última vez el 4 de julio de 2021 (vid. pág. 4).