Videojuego de ajedrez con modo de juego contra la máquina

Anexo IV: Plan de seguridad

Trabajo de Fin de Grado

GRADO EN INGENIERÍA INFORMÁTICA

Junio de 2025

Autor

Óscar Sánchez Rubio

Tutores

Luis Augusto Silva Zendron

Gabriel Villarrubia González

Índice

Contenidos

1. Introducción 1
2. Plan de seguridad 2
   2.1. Identificación de activos 2
   2.2. Evaluación de riesgos 2
   2.3. Priorización de la protección 3
   2.4. Toma de precauciones 3
3. Bibliografía 4

1. Introducción

El presente anexo tiene como propósito hacer una breve descripción del plan de seguridad del proyecto *software*, cuyo objetivo es identificar los posibles activos de la organización (ya sea personal involucrado, *hardware*, *software*, sistemas o datos que componen el proyecto informático) y establecer las medidas necesarias para proteger estos activos relacionados con el desarrollo del proyecto (Mejía R., 2025), en este caso, con el desarrollo de este Trabajo de Fin de Grado sobre un motor de ajedrez.

2. Plan de seguridad

2.1. Identificación de activos

En el contexto del desarrollo de este TFG se han podido identificar los siguientes activos que deben ser protegidos:

- **Código fuente del proyecto**: Todo el código implementado en C# supone el núcleo principal del proyecto y debe mantenerse asegurado y alejado de sujetos no deseables.
- ***Software* de desarrollo**: El conjunto de programas, como el motor de videojuegos Unity y el editor de código Visual Studio Code, implicados en la creación del código anteriormente expuesto, son también objeto de protección.
- **Desarrollador del proyecto**: En este caso, al tratarse de uno solo, es un punto vital del proyecto.
- **Documentación**: Todos los ficheros que contienen información detallada del diseño, funcionamiento, pruebas y resultados del proyecto deben ser protegidos. En concreto, estos son la memoria y los anexos que le corresponden al TFG.
- **Dependencias externas**: Librerías de terceros utilizadas para mejorar o simplificar la implementación, como es el caso de TextMeshPro.
- **Ordenador personal**: En este proyecto, el desarrollador empleó su ordenador personal como principal herramienta *hardware* durante el desarrollo.
- **Plataformas externas**: Actualmente, se almacenan en Google Drive la última versión compilada de la aplicación, junto con diversas anotaciones y documentos que han sido elaborados por y para el desarrollo del proyecto.

2.2. Evaluación de riesgos

A continuación, se enumeran las amenazas más relevantes para los activos identificados en el apartado anterior, y se exponen las consecuencias y la probabilidad de ocurrencia de cada una de ellas:

- **Pérdida del código fuente**: Supondría un retraso crítico en el desarrollo, aunque no sería muy común de que ocurriese puesto que existen respaldos de este.
- **Cambios de políticas en el *software* de desarrollo**: Actualmente, Visual Studio Code en un programa de código abierto y Unity permite no pagar licencias en proyectos no comerciales, sin embargo, nada asegura que esto vaya a ser así siempre, y más en el caso de Unity, cuando estos cambios en las políticas ya han ocurrido en el pasado (Bromberg, 2024). Esto puede suponer un retraso sustancial en el desarrollo del proyecto.
- **Exposición de la documentación**: Al tratarse de un proyecto que no innova demasiado en el ámbito de la creación de un motor de ajedrez, el hecho de que la documentación pueda ser expuesta a sujetos externos al proyecto, no supone demasiado riesgo.

- **Vulnerabilidades en librerías externas**: Esto podría permitir la ejecución de código malicioso en los sistemas implicados en el desarrollo o uso de la aplicación, y es un riesgo muy considerable. Sin embargo, las librerías externas empleadas en este proyecto son muy populares, por lo que se presupone que contarán con un nivel de seguridad, en cuanto a su producto, bastante alto.

- **Obsolescencia de las librerías externas**: La posibilidad de que las librerías externas no sean actualizadas con el tiempo, puede provocar incompatibilidades con distintos módulos del proyecto. De nuevo, son librerías muy populares, esto no ocurrirá pronto.

- **Infección por *malware***: A través del equipo personal, lo que podría afectar el código o robar información. Aunque no es tan improbable de que ocurra, sí que es bastante difícil que un virus en concreto perjudique información del proyecto.

- **Daños físicos en el ordenador personal**: Podría provocar la pérdida irreversible de datos relativos al proyecto, lo que puede afectar gravemente no existían copias de seguridad del mismo. Su probabilidad no es muy alta.

- **Indisponibilidad de plataformas externas**: La caída de servicios como Google Drive, podría ocasionar la perdida de una parte de demasiado grande de la información que no está respaldada. En cambio, Google Drive presenta un historial bastante fiable en cuanto a la disponibilidad de sus servicios.

2.3.Priorización de la protección

Una vez se ha realizado este análisis, es el momento de decidir qué activos resultan más importantes e interesantes para empezar a proteger. El principal activo a proteger puede considerarse el ordenador personal del desarrollador, puesto que almacena las versiones más recientes del código fuente y de la documentación, siendo un proyecto muy centralizado en este aspecto.

Como antes se ha mencionado, el código fuente contenido en el ordenador personal, es vulnerable tanto a *malwares* como a pérdidas de cualquier tipo, por ejemplo, accidentales. Además, esto mismo puede sucederle a la documentación, por lo que el equipo personal debe ser el foco de la protección para garantizar la seguridad.

2.4.Toma de precauciones

Una primera precaución para evitar estas vulnerabilidades es emplear un sistema de control de versiones, que mantenga almacenadas las últimas versiones, tanto del código fuente, como de la documentación. Otra solución podría ser realizar copias de seguridad automáticas hacia un servidor externo en que la información se encuentre asegurada.

En cuanto a los problemas menores, mantener actualizadas las bibliotecas externas y aplicaciones empleadas en el desarrollo, es una buena práctica para impedir que ocurran problemas que surjan de vulnerabilidades en estos componentes.

3. Bibliografía

Bromberg, M. (2024, Septiemble 12). *A message to our community: Unity is canceling the Runtime Fee*. Retrieved from Unity: https://unity.com/blog/unity-is-canceling-the-runtime-fee

Mejía R., F. A. (2025, Enero 29). *Pasos para elaborar el plan de seguridad informática*. Retrieved from LinkedIn: https://www.linkedin.com/pulse/pasos-para-elaborar-el-plan-de-seguridad-inform%C3%A1tica-mej%C3%ADa-r--sbfie/