|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CONTROL DE VERSIONES | | | | | |
| *Versión* | *Hecha por* | *Revisada por* | *Aprobada por* | *Fecha* | *Motivo* |
| 1.0 | Miguel Pacheco | Melissa Cervantes | Melissa Cervantes | 15/06/06 | Realizar un acta de reunión de retrospectiva para el sprint 1 |

ACTA DE REUNIÓN DE RETROSPECTIVA SPRINT I

|  |  |  |  |
| --- | --- | --- | --- |
| Proyecto | Genesis | | |
| Fecha y hora | 28/06/06 | Convocada por | David Chris |
| Lugar | Google Meet | Facilitador | Melissa Cervantes |
| Objetivo | Seguimiento de las actividades que se realizaran para el sprint 1 | | |

|  |  |  |
| --- | --- | --- |
| ASISTENTES | | |
| *PERSONA* | *CARGO* | *EMPRESA* |
| Melissa Cervantes | Scrum Master | Genesis |
| Miguel Pacheco | Product Owner | Genesis |
| David Chris | Scrum | Genesis |
| Stephany Vargas | Scrum | Genesis |

|  |  |  |
| --- | --- | --- |
| AGENDA | | |
| *Actividad* | Responsable | Tiempo |
| revisión del objetivo de la retrospectiva | Melissa Cervantes | 5 minutos |
| ¿Qué salió bien en el Sprint? | David Chris | 20 minutos |
| ¿Qué se puede mejorar? | Stephany Vargas | 20 minutos |
| Definición de Acciones y Compromisos | Miguel Pacheco | 10 min |
| Cierre de la Retrospectiva | Melissa Cervantes | 5 minutos |
|  |  |  |

|  |  |
| --- | --- |
| CONCLUSIONES | |
| 1 | Para las búsquedas en vivo se realizó con la herramienta WebSockets lo cual nos produjo un retraso de 5 horas, lo que suponía unas 20 horas, lo cual sería un total de 25 horas ,por lo cual se recomienda el uso de otras herramientas para la búsqueda de pujas en vivo tales como AJAX o Redis reducen la carga de tiempos de respuesta. |
| 2 | Para el backend en el filtrado de búsqueda se ha utilizado PostgreSQL con el lenguaje Python y la librería Django lo cual ha permitido que se mejoren los tiempos ahorrándonos 5 horas lo que suponía un trabajo de 25 horas, reduciéndose a 20 horas. Por esto se recomienda seguir utilizando las mismas herramientas. |
| 3 | Para la muestra del historial se usó la tecnología anteriormente mencionada WebSockets lo cual produjo retraso de 5 horas por lo cual se sugiere el uso de herramientas como React. |
| 4 | La validación de compras a través de la generación de certificados digitales, activada tras la adjudicación de una subasta, fue implementada utilizando la biblioteca React y el framework Angular. Gracias a esta arquitectura tecnológica, la funcionalidad se desarrolló y entregó dentro del plazo estimado de 20 horas establecido en la historia de usuario correspondiente al sprint. |
| 5 | Para la implementación de los formularios de registro de usuario, se utilizó jQuery en el frontend, lo cual generó un retraso de aproximadamente 7 horas respecto al tiempo estimado inicialmente. Aunque el desarrollo estaba planificado para completarse en 20 horas, se requerieron finalmente 27 horas debido a las limitaciones y complejidades de esta tecnología. Por ello, se recomienda optar por herramientas más modernas, mantenibles y eficientes como React, Angular o el uso adecuado de HTML5, que ofrecen una mayor productividad y mejor experiencia de desarrollo. |
| 6 | Se utilizó PHP como servidor para implementar la funcionalidad de subastas en vivo, lo cual generó un retraso de aproximadamente 4 horas debido a sus limitaciones para manejar eventos en tiempo real. Para este tipo de aplicaciones, se recomienda el uso de servidores como Node.js o Django, que están mejor preparados para gestionar conexiones persistentes y comunicación en tiempo real con mayor eficiencia. |