

Planificación y Gestión de Proyectos Informáticos



Tipo de documento: Sprint Retrospective Iteración 1

Nombre del proyecto: Adquisición de obras de arte en formato físico

Grupo: 2.12

Fecha: 19/11/2023

SPRINT RETROSPECTIVE

NOMBRE DEL PROYECTO:	Adquisición de obras de arte en formato físico (ArtUS)
CÓDIGO DEL PROYECTO:	2.12
DIRECTOR DEL PROYECTO:	Jesús Torres Valderrama
FECHA DE ELABORACIÓN:	19/11/2023

HISTORIAL DE VERSIONES			
FECHA Y HORA	N° DE VERSIÓN	DESCRIPCIÓN	ELABORADO POR
19/11/23 - 16:00	1.0	Primera versión del documento	Equipo de desarrollo
20/10/23 - 17:00	2.0	Revisión del documento	Equipo de desarrollo

PROPÓSITO DEL SPRINT RETROSPECTIVE

El propósito principal del Sprint Retrospective es permitir que el equipo de desarrollo inspeccione y adapte sus procesos para mejorar continuamente su eficacia y colaboración de cara a las siguientes iteraciones que debe realizar para el proyecto.

Este documento proporciona un espacio dedicado para que el equipo reflexione sobre el Sprint recién finalizado. Se revisan los logros, los desafíos, las dificultades y los éxitos alcanzados durante el Sprint.

Asimismo, se identifican oportunidades para mejorar tanto los procesos de trabajo como la dinámica del equipo. Se buscan áreas donde se pueda optimizar el flujo de trabajo, la comunicación, la colaboración o cualquier aspecto que afecte la eficiencia y la calidad del trabajo. Todo esto se recoge en el plan de acción.

METODOLOGÍA DEL EQUIPO DE DESARROLLO

SPRINT RETROSPECTIVE

Gestión de issues.

En este Sprint, se creó una tarea por elemento o conjunto de elementos del product backlog, se evitó realizar asignaciones grupales y se hizo uso de las tareas épicas para denominar aquellas que tenían una mayor carga de trabajo. A la hora de repartirlas, se intentó buscar que no hubiera dependencias entre tareas de distintos miembros en la medida de lo posible. Como principales puntos a mejorar en las próximas iteraciones se encuentran:

- Repartición de tareas. Aunque el reparto ha sido aceptable, este podría ser aún mejor buscando como objetivo que cada miembro del equipo de desarrollo pueda trabajar de forma totalmente independiente en sus tareas y por tanto ser más eficiente y rápido.
- Estimaciones de las tareas. A la hora de realizar un reparto lo más equitativo posible, se usó de referencia las estimaciones hechas por el equipo, buscando que todos los miembros tuvieran un número similar de puntos de esfuerzo. Sin embargo, en el desarrollo del Sprint, se observó como el esfuerzo real requerido para realizar ciertas tareas no correspondía con el que se suponía y algunas personas tuvieron una carga de trabajo mayor que otras.
- Gestión de las tareas mejorable. Han sucedido errores puntuales como que una tarea permanezca en “In review” a pesar de haber sido finalizada y su Pull Request haber sido aceptada.

Gestión de ramas.

En este Sprint se ha usado la metodología GitFlow, y la gestión de ramas ha sido adecuada:

- Seguimiento adecuado de la política de ramas definida. Se ha trabajado en ramas de feature asociadas a issues que trataban de agregar funcionalidades al sistema, refactor para aquellas tareas de mejora de códigos o hotfix en caso de querer hacer un arreglo cuando ya se ha subido el código. Luego, se ha volcado todo el código en una rama develop que sirve de intermediaria entre las otras ramas y main.

Cabe destacar aparte, que la nomenclatura de las ramas ha sido correcta en todo momento.

Commits y Pull Requests.

En general el equipo sigue correctamente la política dictada de commits excepto por algunos fallos puntuales que se cometen. Los errores más destacables son los siguientes:

SPRINT RETROSPECTIVE

- Nomenclatura incorrecta de un commit - [Titulo: arreglo index.html](#). Debería tener una etiqueta feat, fix, refactor, test en lugar de “título”.
 - Incumplimiento de nombre de commit - [fix, Arreglo de errores en el detalle de producto](#). Debería seguir la estructura de los commits.
- El equipo debe cuidar la forma de realizar los commits y pull requests y evitar al máximo posible los fallos mencionados previamente.

REFLEXIÓN DEL SPRINT

LOGROS / ÉXITOS	DESAFÍOS / DIFICULTADES
<ul style="list-style-type: none">• El equipo ha logrado realizar el trabajo requerido para la iteración.• Lo realizado funciona correctamente y la interfaz se ve bonita.• Los miembros han sabido organizarse y coordinarse correctamente con otros miembros.• Aprender Django desde 0 y saber implementarlo correctamente para la iteración.	<ul style="list-style-type: none">• Mantener una comunicación sostenida con los miembros.• Decidir el idioma de programación.• Decidir la política y estructura de commits, pull requests.

PLAN DE ACCIÓN

START	STOP	CONTINUE DOING
<ul style="list-style-type: none">• Realizar un mejor reparto de las tareas entre los miembros para que sea equitativo.• Hacer una subdivisión de roles dentro del equipo de trabajo, en desarrolladores y documentadores.	<ul style="list-style-type: none">• Tomar decisiones sin consultarlo con el equipo de dirección.• Realizar commits con una nomenclatura incorrecta.	<ul style="list-style-type: none">• Cumplimiento de la política de commits y ramas definidas.• Hacer una revisión de las distintas implementaciones antes de que estas se acepten. Para aceptarla, debe haber al menos una valoración positiva.• Seguir trabajando de manera diaria y consistente.

SPRINT RETROSPECTIVE

<ul style="list-style-type: none">● Fomentar una mejor comunicación entre los miembros del equipo y con el equipo de dirección.● Realizar solicitudes de cambio cuando haga falta realizar un cambio en el proyecto.● Realizar reuniones semanales con el equipo de dirección para determinar la calidad del proyecto y preguntar dudas.		<ul style="list-style-type: none">● Cada miembro se encarga de una tarea a la vez, es decir, una tarea en "In progress" por persona.● Se crean las pull request asociadas inmediatamente después de que la tarea sea movida a la columna de "In review".● Mover las tareas dentro de GitHub para representar correctamente el estado en el que se encuentra cada una.
--	--	---