

# Informe de Cierre

*Planificación y Gestión de Proyectos Informáticos*

<https://github.com/pabsanper/ACME-Cycling.git>



**GRUPO 3.8**

Alberto Benítez Morales

David Sabugueiro Troya

Pablo Santos Pérez

Francisco Javier Vázquez Monge

Álvaro Paradas Borrego

**CLIENTE:** José González Enríquez

**FECHA:** 9/12/2022

## Control de cambios

Versión	Fecha	Tipo	Responsables	Descripción	Motivación
1.0	8/12/22	Creación	David Sabugueiro	Creación y relleno del documento	Realizar uno de los documentos para el cierre del proyecto
1.1	9/12/22	Cierre	Pablo Santos	Puesta a punto del documento y cierra	Proceder con el cierre del proyecto



## Tabla de contenidos

<b>Control de cambios</b>	<b>2</b>
<b>Tabla de contenidos</b>	<b>3</b>
<b>1. Descripción de proyecto</b>	<b>4</b>
<b>2. Resumen de rendimiento</b>	<b>4</b>
<b>3. Variaciones</b>	<b>5</b>
<b>4. Gestión de beneficios</b>	<b>6</b>
<b>5. Necesidades del proyecto</b>	<b>6</b>
<b>6. Riesgos e incidencias</b>	<b>6</b>



## 1. Descripción de proyecto

Desde ACME-Cycling ponemos a su disposición un amplio catálogo de los mejores componentes para la bicicleta de sus sueños, pudiendo elegir entre las piezas que el cliente desee, con el fin de poder personalizarla al completo.

La totalidad del comercio se desarrollará de forma online mediante la página web que, diseñada por un grupo de ingenieros del software, está construida mediante el framework de desarrollo Django, con el fin de ofrecer al cliente una interfaz rápida y limpia.

El criterio de éxito sería el correcto funcionamiento de la tienda web, así como su cierre y entrega dentro de los límites económicos y temporales previamente establecidos.

## 2. Resumen de rendimiento

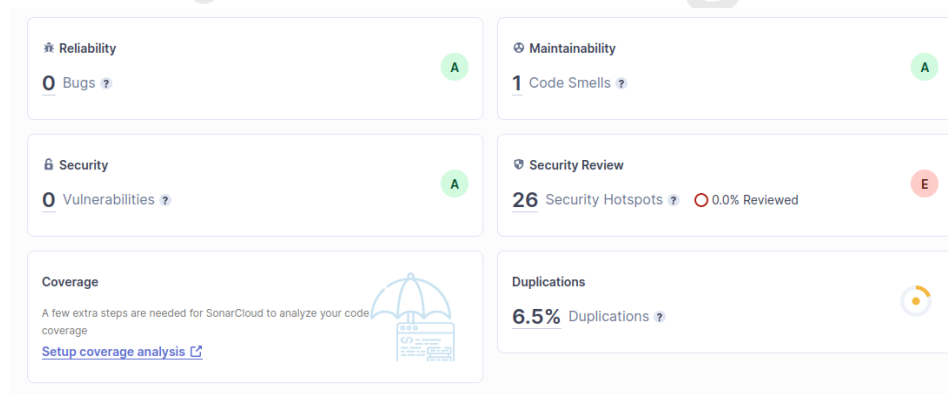
	Objetivos	Criterios de finalización	Evidencia
<b>Alcance</b>	El objetivo principal logrado respecto al alcance ha sido el haber cerrado el proyecto tal y como se esperaba, para ello ha sido necesario cumplir todos y cada uno de los diferentes documentos que habían sido estimados a formar parte del producto final, así como todos los requisitos detallados en el documento de requisitos	Respecto al objetivo, el principal criterio de finalización además del cumplimiento de los plazos y costes, sería la correcta funcionalidad de la aplicación web, además del funcionamiento de todos y cada uno de los requisitos detallados.	Como evidencia de ello se puede comprobar que el producto se ha entregado dentro de los costes permitidos, dentro de los plazos permitidos. Además de la correcta funcionalidad de la aplicación web.
<b>Calidad</b>	El principal objetivo es el cumplimiento de los siete principios de la gestión de la calidad: <ul style="list-style-type: none"> <li>• Enfoque en el cliente</li> <li>• Liderazgo</li> <li>• Compromiso</li> </ul>	Los criterios que corresponden al cumplimiento de las auditorías de calidad. A su vez en el proceso de ejecución se han realizado análisis de código periódicos	Por la parte de documentación el cumplimiento de las siguientes normas: <ul style="list-style-type: none"> <li>• ISO 9000:2005</li> <li>• ISO 9001:2008</li> <li>• ISO 9004:2009</li> </ul>

	<ul style="list-style-type: none"> <li>• Enfoque basado en procesos</li> <li>• Mejora</li> <li>• Toma de decisiones basadas en evidencias</li> <li>• Buenas relaciones en la administración</li> </ul>		Por otro lado en cuanto a desarrollo, el código base es un código limpio de acuerdo con las métricas establecidas en SonarCloud.
--	--	--	--

### 3. Variaciones

En cuanto a las variaciones de tiempo y dinero con respecto a lo planificado, en primer lugar, cabe decir que las aproximaciones fueron bastante mejor de lo esperado, como se puede ver:

- Programación: No hubo desviación ninguna, ya que todos los requisitos fueron ejecutados correctamente
- Costes: Hubo desviación de -40.85€, por lo que la estimación ha sido bastante precisa a nivel general, pero mejorable a nivel de paquetes de control.
- Calidad: La calidad obtenida fue la esperada y este es el análisis de código resultante en SonarCloud



- El código está limpio de bugs y vulnerabilidades.
- Tan solo tenemos un Code Smell que proviene de haber llamado a la misma plantilla en más de una función en 'views.py'. Hemos considerado que está bien tal como está y no cambiaremos eso.

- Los security hotspots vienen de contraseñas que hemos usado en el código pero solo las usamos en local, por lo que no es ningún problema. Las que sí usamos están guardadas en una variable secret de GitHub.
- Por último, el código duplicado se da en su mayoría en los tests, a lo cuál no le damos importancia. El resto son duplicaciones mínimas de estructuras usadas por ejemplo en el carrito y el desplegable del carrito.

## **4. Gestión de beneficios**

Como ya se ha detallado anteriormente, el producto final ha cumplido tanto con el alcance como con la calidad esperada. Se ha conseguido, entre otros, con una buena planificación y estimación de las tareas, así como un desempeño correcto del equipo de desarrollo, siguiendo en todo momento las métricas de calidad ya definidas, así como cumplimiento de los plazos establecidos

## **5. Necesidades comerciales**

El proyecto final entregado al cliente cumple con las necesidades del negocio, tras haber cumplido todos y cada uno de los requisitos detallados en el documento de requisitos del Plan de Gestión del Alcance, así como todos y cada uno de los componentes requeridos por el cliente para el producto final. Todas las tareas del equipo de dirección han sido completadas en sus fechas correspondientes y con la calidad esperada, así como todas las tareas del equipo de desarrollo, que han cumplido con la calidad esperada, tras haber sido desarrolladas durante tres iteraciones.

## **6. Riesgos e incidencias**

Para los riesgos planificados en el Plan de Gestión de Riesgos teníamos acciones preventivas que han sido suficientes para no tener problemas. Sin embargo han habido otras incidencias que también hemos sabido solventar correctamente.

Riesgo/Incidencia	Respuesta	Comentarios
Problemas configuración entorno con otras asignaturas	Crear un nuevo entorno de virtualización de python	De esta manera se evitan los conflictos de python con otra asignatura
Configuración de cada uno de los módulos añadidos	A base de prueba y error, hemos encontrado las versiones adecuadas de Django, Stripe y Pillow	Hemos usado Django 3.0, Pillow 9.3.0 y Stripe 2.55.2
Diferente disponibilidad durante la semana de los miembros del equipo, por lo cual las tareas que dependen unas de otras se ven influidas	Creación de un documento con las disponibilidades de los integrantes con el fin de planear el reparto de tareas mejor	Este documento ha sido un excel en el que cada miembro proponía su disponibilidad
Al operar con distintas aplicaciones se necesitan diferentes archivos base.html	Hemos tenido que duplicar el mismo archivo base.html en todas las aplicaciones para poder acceder a este mismo	Este error fue solucionado y dejamos un solo base.html
Incompatibilidad de ciertas versiones de distintas herramientas	Documentarnos en internet sobre compatibilidad de funciones para poder ejecutar los módulos añadidos	Concretamente las versiones de psycopg2=2.8.4 y en el PaaS la versión de Python 3.10
En el PaaS usado, Pythonanywhere, la base de datos Postgres era de pago	Hemos cambiado la base de datos a SQLite	Todo funciona correctamente tras esto