



Optimización Biomecánica del Ciclismo Asistida por Inteligencia Artificial

Autor:

Ing. Rodrigo Iván Goñi

Director:

MSc. Fernando Corteggiano (FNRC)

*Esta planificación fue realizada en el curso de Gestión de proyectos
entre el 01 de julio de 2025 y el 30 de abril de 2026.*

Índice

1. Descripción técnica-conceptual del proyecto a realizar	5
2. Identificación y análisis de los interesados	6
3. Propósito del proyecto	6
4. Alcance del proyecto	7
5. Supuestos del proyecto	7
6. Product Backlog	7
7. Criterios de aceptación de historias de usuario	8
8. Fases de CRISP-DM	9
9. Desglose del trabajo en tareas	10
10. Diagrama de Gantt	11
11. Planificación de Sprints.	11
12. Normativa y cumplimiento de datos (gobernanza).	13
13. Gestión de riesgos	13
14. Sprint Review	15
15. Sprint Retrospective	15

Registros de cambios

Revisión	Detalles de los cambios realizados	Fecha
0	Creación del documento	01 de julio de 2025

Acta de constitución del proyecto

Buenos Aires, 01 de julio de 2025

Por medio de la presente se acuerda con el Ing. Rodrigo Iván Goñi que su Trabajo Final de la Carrera de Especialización en Inteligencia Artificial se titulará “Optimización Biomecánica del Ciclismo Asistida por Inteligencia Artificial ” y consistirá en y consistirá en el desarrollo de un prototipo de un sistema inteligente que, mediante la integración de la detección de pose por redes neuronales y el análisis de datos de sensores, optimice los parámetros biomecánicos de la bicicleta para maximizar la potencia, eficiencia y minimizar el riesgo de lesiones del ciclista. El trabajo tendrá un presupuesto preliminar estimado de 600 horas y un costo estimado de \$15000, con fecha de inicio el 01 de julio de 2025 y fecha de presentación pública el 15 de mayo de 2026.

Se adjunta a esta acta la planificación inicial.

Dr. Ing. Ariel Lutenberg
Director posgrado FIUBA

Nombre del cliente
Empresa del cliente

MSc. Fernando Corteggiano
Director del Trabajo Final

1. Descripción técnica-conceptual del proyecto a realizar

La biomecánica en el ciclismo es un factor fundamental para mejorar el rendimiento y prevenir lesiones, basándose en el principio de adaptar la bicicleta a las características físicas del ciclista. Un ajuste incorrecto no solo puede causar lesiones, sino también disminuir la potencia de salida hasta en un 20 % [1]. Sin embargo, el acceso a un análisis biomecánico profesional presenta barreras significativas: las soluciones actuales suelen ser costosas, de baja disponibilidad y requieren visitas a laboratorios especializados.

El desafío principal de este proyecto es encontrar el balance óptimo entre la posición que maximiza la velocidad y aquella que minimiza el esfuerzo y el riesgo de lesiones. Frecuentemente, la postura más aerodinámica no es la más sostenible a largo plazo. Para abordar este problema, se propone el desarrollo de un sistema que ajuste automáticamente los parámetros de la bicicleta. Mediante el uso de inteligencia artificial, el sistema analizará la postura del ciclista para optimizar la potencia de salida y reducir la tensión muscular, lo que exige un enfoque de optimización multiobjetivo con diversas restricciones.

Las soluciones convencionales se basan en un análisis estático y puntual, realizado en un único día y dependiente en gran medida de la experiencia del biomecánico. La recomendación de repetir el ajuste anualmente, sumada a su alto costo y escasa disponibilidad, provoca que la mayoría de los ciclistas no mantengan una configuración óptima en sus bicicletas.

El valor fundamental de este sistema radica en ofrecer al ciclista la capacidad de realizar autoajustes frecuentes, de forma autónoma y a un costo significativamente menor que las alternativas tradicionales. Esto democratiza el acceso a una biomecánica de precisión, permitiendo una mejora continua del rendimiento y la prevención de lesiones.

Para lograr estos objetivos, el sistema propuesto se estructura en una serie de módulos interconectados, como se ilustra en el diagrama de bloques de la Figura 1 a continuación:

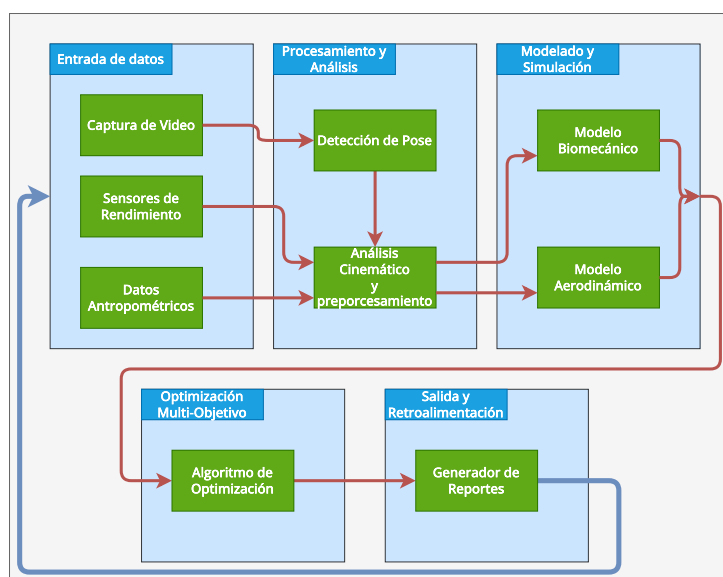


Figura 1. Diagrama en bloques del sistema.

- **Entrada de datos:** Este módulo es el encargado de recolectar datos de las distintas fuentes de información. Se compone de una fuente de video, sensores de rendimiento y los datos antropométricos del ciclista.
- **Procesamiento y análisis:** Este módulo toma los datos de entrada y los procesa. Con la ayuda de una red neuronal de detección de pose, añade los datos de posición del ciclista al sistema. Luego, en un módulo de preprocesamiento, los datos se filtran, sincronizan, limpian y completan.
- **Modelado y Simulación:** Con ayuda de un modelo físico y aerodinámico, se predice cómo impactarán los cambios de los parámetros de la bicicleta en el rendimiento.
- **Optimización Multi-Objetivo:** Con un algoritmo genético, se buscará maximizar la performance del objetivo en un rango adecuado de posición, tratando de minimizar la resistencia aerodinámica y teniendo en cuenta la morfología y el nivel del ciclista.
- **Salida y Retroalimentación:** Con los datos de la optimización, se generará un reporte de recomendaciones y posibles configuraciones de la bicicleta. Una vez finalizado el reporte, se recomienda ajustar los parámetros para volver a iniciar el análisis.

2. Identificación y análisis de los interesados

Rol	Nombre y Apellido	Organización	Puesto
Auspiciante	-	-	-
Cliente	Ciclistas de distintos tipos	-	-
Impulsor	-	-	-
Responsable	Ing. Rodrigo Iván Goñi	FIUBA	Alumno
Colaboradores	-	-	-
Orientador	MSc. Fernando Corteggiano	FNRC	Director del Trabajo Final
Equipo	-	-	-
Opositores	-	-	-
Usuario final	-	-	-

- **Cliente:** Ciclistas de distintos tipos que buscan optimizar la comodidad o el rendimiento de su bicicleta a través de ajustes personalizados.
- **Responsable:** Rodrigo Goñi es el líder del proyecto de optimización biomecánica asistida por IA, originado por una necesidad personal de ajustar su bicicleta. Es ingeniero en IA y robótica.
- **Orientador:** Fernando Corteggiano, Ingeniero Electricista y Magister en Ciencias de la Ingeniería de la UNRC, es director y profesor adjunto, ya ha dirigido diversas tesis, va a aportar su experiencia en electrónica, telecomunicaciones y software, para definir el alcance y requerimientos del sistema.

3. Propósito del proyecto

¿Por qué se hace el proyecto? ¿Qué se quiere lograr?

Se recomienda que sea solo un párrafo que continúe con la idea de la frase “el propósito de este proyecto es...” (omitir la frase, ya que está en el título de la sección).

4. Alcance del proyecto

¿Qué se incluye y que no se incluye en este proyecto?

Se refiere al trabajo que se va a hacer para entregar el producto o resultado especificado.

Explicitar todo lo quede comprendido dentro del alcance del proyecto. Por ejemplo:

El proyecto incluye:

- Ítem 1.
- Ítem 2.
 - Subítem 1.
 - Subítem 2.
 - ...
- ...

Explicitar además todo lo que no quede incluido (“El presente proyecto no incluye...”)

5. Supuestos del proyecto

“Para el desarrollo del presente proyecto se supone que: ...”

- Supuesto 1.
- Supuesto 2.
- ...

Por ejemplo, se podrían incluir supuestos respecto a disponibilidad de tiempo y recursos humanos y materiales, sobre la factibilidad técnica de distintos aspectos del proyecto, sobre otras cuestiones que sean necesarias para el éxito del proyecto como condiciones macroeconómicas o reglamentarias.

6. Product Backlog

El Product Backlog debe organizarse en cuatro *épicas* fundamentales del proyecto. Cada épica debe contener al menos dos historias de usuario que describan funcionalidades clave.

El Product Backlog debe permitir interpretar cómo será el proyecto y su funcionalidad. Se deben indicar claramente las prioridades entre las historias de usuario y si hay alguna opcional.

Las historias de usuario deben ser breves, claras y medibles, expresando el rol, la necesidad y el propósito de cada funcionalidad. También deben tener una prioridad definida para facilitar la planificación de los sprints.

Cada historia de usuario debe incluir una ponderación en *Story Points*, un número entero que representa el tamaño relativo de la historia. El criterio para calcular los Story Points debe indicarse explícitamente.

Las historias deben seguir el formato: “*Como [rol], quiero [tal cosa] para [tal otra cosa]*”.

Las épicas deben estructurarse de la siguiente forma:

- **Épica 1**
 - HU1
 - HU2
- **Épica 2**
 - HU3
 - HU4
- **Épica 3**
 - HU5
 - HU6
- **Épica 4**
 - HU7
 - HU8

Reglas para definir historias de usuario:

- Ser concisas y claras.
- Expresarlas en términos cuantificables y medibles.
- No dejar margen para interpretaciones ambiguas.
- Indicar claramente su prioridad y si son opcionales.
- Considerar regulaciones y normas vigentes.

7. Criterios de aceptación de historias de usuario

Los criterios de aceptación deben establecerse para cada historia de usuario, asegurando que se cumplan las condiciones necesarias para que la funcionalidad sea validada correctamente.

Cada historia debe tener criterios medibles, específicos y verificables. Deben permitir validar que se cumple con las necesidades del usuario.

Se estructuran de forma análoga a las épicas del backlog:

- **Épica 1**
 - Criterios de aceptación HU1
 - Criterios de aceptación HU2
- **Épica 2**
 - Criterios de aceptación HU3
 - Criterios de aceptación HU4
- **Épica 3**
 - Criterios de aceptación HU5
 - Criterios de aceptación HU6
- **Épica 4**
 - Criterios de aceptación HU7
 - Criterios de aceptación HU8

Reglas para definir criterios de aceptación:

- Medibles y verificables.
- Especificar cuándo una historia se considera completada.
- Incluir condiciones específicas.
- No ambiguos.
- Probables de testear funcional o técnicamente.
- Mínimo 3 criterios por HU.

8. Fases de CRISP-DM

1. **Comprensión del negocio:** objetivo, valor agregado de IA, métricas de éxito.
2. **Comprensión de los datos:** tipo, origen, cantidad, calidad.
3. **Preparación de los datos:** características clave, transformaciones necesarias.
4. **Modelado:** tipo de problema, algoritmos posibles.
5. **Evaluación del modelo:** métricas de rendimiento.
6. **Despliegue del modelo (opcional):** tipo de despliegue y herramientas.

9. Desglose del trabajo en tareas

A partir de cada HU, descomponer en tareas concretas, técnicas y medibles:

- Duración estimada: entre 2 y 8 h. Evitar tareas genéricas.
- Si una tarea excede 8 h, dividirla.
- Indicar prioridad relativa (Alta, Media, Baja).

Historia de usuario	Tarea técnica	Estimación	Prioridad
HU1	Tarea 1 HU1	6 h	Alta
HU1	Tarea 2 HU1	8 h	Alta
HU2	Tarea 1 HU2	5 h	Media
HU2	Tarea 2 HU2	6 h	Alta
...

Criterios para estimar tiempos:

- Considerar dificultad, complejidad técnica y grado de incertidumbre de cada tarea.
- Evitar subestimar el esfuerzo requerido. Si una tarea supera las 8 h, dividirla en subtareas.
- Basar la estimación en la experiencia propia o en referencias de tareas similares.

Sobre la prioridad:

- Asignar una prioridad relativa (Alta, Media o Baja) según la relevancia funcional de la tarea y su impacto en los entregables.
- Priorizar tareas que estén vinculadas a criterios de aceptación de las HU o que sean necesarias para desbloquear otras.
- Incluir tareas opcionales solo si están bien justificadas.

Recomendaciones generales:

- Incluir al menos dos tareas por historia de usuario.
- El total estimado debe ser coherente con la planificación global de unas 600 horas (la cantidad de horas sugeridas para un trabajo de posgrado).
- Este desglose se utilizará en las secciones siguientes para armar el diagrama de Gantt (sección 10) y la planificación de sprints (sección 11).
- Recordar que la calidad del desglose impacta directamente en la calidad de la planificación del proyecto.

10. Diagrama de Gantt

El diagrama de Gantt debe representar de forma visual y cronológica todas las tareas del proyecto, abarcando aproximadamente 600 horas totales, de las cuales entre 480 y 500 deben destinarse a tareas técnicas (desarrollo, pruebas, implementación) y entre 100 y 120 a tareas no técnicas (planificación, documentación, escritura de memoria y preparación de la defensa).

Consignas y recomendaciones:

- Incluir tanto tareas técnicas derivadas de las HU como tareas no técnicas generales del proyecto.
- El eje vertical debe listar las tareas y el eje horizontal representar el tiempo en semanas o fechas.
- Utilizar colores diferenciados para distinguir tareas técnicas y no técnicas.
- Las tareas deben estar ordenadas cronológicamente y reflejar todo el ciclo del proyecto.
- Iniciar con la planificación del proyecto (coincidente con el inicio de Gestión de Proyectos) y finalizar con la defensa, próxima a la fecha de cierre del trabajo.
- Configurar el software para mostrar los códigos del desglose de tareas y los nombres junto a cada barra.
- Asegurarse de que la fecha final coincida con la del Acta Constitutiva.
- Evitar tareas genéricas o ambiguas y asegurar una secuencia lógica y realista.
- Las fechas pueden ser aproximadas; ajustar el ancho del diagrama según el texto y el parámetro `x unit`. Para mejorar la apariencia del diagrama, es necesario ajustar este valor y, quizás, acortar los nombres de las tareas.

Herramientas sugeridas:

- Planner, GanttProject, Trello + plugins
<https://blog.trello.com/es/diagrama-de-gantt-de-un-proyecto>
- Creately (colaborativa online)
<https://creately.com/diagram/example/ieb3p3ml/LaTeX>
- LaTeX con `pgfgantt`:
<http://ctan.dcc.uchile.cl/graphics/pgf/contrib/pgfgantt/pgfgantt.pdf>

Incluir una imagen legible del diagrama de Gantt. Si es muy ancho, presentar primero la tabla y luego el gráfico de barras.

11. Planificación de Sprints

Organizar las tareas técnicas del proyecto en sprints de trabajo que permitan distribuir de forma equilibrada la carga horaria total, estimada en 600 horas.

Consigna:

- Completar una tabla que relacione sprints con HU y tareas técnicas correspondientes.
- Incluir estimación en horas para cada tarea.
- Indicar responsable y porcentaje de avance estimado o completado.
- Contemplar también tareas de planificación, documentación, redacción de memoria y preparación de defensa.

Conceptos clave:

- Una épica es una unidad funcional amplia; una historia de usuario es una funcionalidad concreta; un sprint es una unidad de tiempo donde se ejecutan tareas.
- Las tareas son el nivel más desagregado: permiten estimar tiempos, asignar responsables y monitorear progreso.

Duración sugerida:

- Para un proyecto de 600 h, se recomienda planificar entre 10 y 12 sprints de aproximadamente 2 semanas cada uno.
- Asignar entre 45 y 50 horas efectivas por sprint a tareas técnicas.
- Reservar 100 a 120 h para actividades no técnicas (planificación, escritura, reuniones, defensa).

Importante:

- En proyectos individuales, el responsable suele ser el propio autor.
- Aun así, desagregar tareas facilita el seguimiento y mejora continua.

Conversión opcional de Story Points a horas:

- 1 SP \approx 2 h como referencia flexible.
- Tener en cuenta aproximaciones tipo Fibonacci.

Recomendaciones:

- Verificar que la carga horaria por sprint sea equilibrada.
- Usar sprints de 1 a 3 semanas, acordes al cronograma general.
- Actualizar el % completado durante el seguimiento del proyecto.
- Considerar un sprint final exclusivo para pruebas, revisión y ajustes antes de la defensa.

Cuadro 1. Formato sugerido

Sprint	HU o fase	Tarea	Horas / SP	Responsable	% Completado
Sprint 0	Planificación	Definir alcance y cronograma	10 h	Alumno	100 %
Sprint 0	Planificación	Reunión con tutor/cliente	5 h	Alumno	50 %
Sprint 0	Planificación	Ajuste de entregables	6 h	Alumno	25 %
Sprint 1	HU1	Tarea 1 HU1	6 h / 3 SP	Alumno	0 %
Sprint 1	HU1	Tarea 2 HU1	10 h / 5 SP	Alumno	0 %
Sprint 2	HU2	Tarea 1 HU2	7 h / 5 SP	Alumno	0 %
...
Sprint 5	Escritura	Redacción memoria	50 h / 34 SP	Alumno	0 %
Sprint 6	Defensa	Preparación exposición	20 h / 13 SP	Alumno	0 %

12. Normativa y cumplimiento de datos (gobernanza)

En esta sección se debe analizar si los datos utilizados en el proyecto están sujetos a normativas de protección de datos y privacidad, y en qué condiciones se pueden emplear.

Aspectos a considerar:

- Evaluar si los datos están regulados por normativas como GDPR, Ley 25.326 de Protección de Datos Personales en Argentina, HIPAA u otras según jurisdicción y temática.
- Determinar si el uso de los datos requiere consentimiento explícito de los usuarios involucrados.
- Indicar si existen restricciones legales, técnicas o contractuales sobre el uso, compartición o publicación de los datos.
- Aclarar si los datos provienen de fuentes licenciadas, de acceso público o bajo algún tipo de autorización especial.
- Analizar la viabilidad del proyecto desde el punto de vista legal y ético, considerando la gobernanza de los datos.

Este análisis es clave para garantizar el cumplimiento normativo y evitar conflictos legales durante el desarrollo y publicación del proyecto.

13. Gestión de riesgos

a) Identificación de los riesgos (al menos cinco) y estimación de sus consecuencias:

Riesgo 1: detallar el riesgo (riesgo es algo que si ocurre altera los planes previstos de forma negativa)

- Severidad (S): mientras más severo, más alto es el número (usar números del 1 al 10). Justificar el motivo por el cual se asigna determinado número de severidad (S).
- Probabilidad de ocurrencia (O): mientras más probable, más alto es el número (usar del 1 al 10). Justificar el motivo por el cual se asigna determinado número de (O).

Riesgo 2:

- Severidad (S): X.
Justificación...
- Ocurrencia (O): Y.
Justificación...

Riesgo 3:

- Severidad (S): X.
Justificación...
- Ocurrencia (O): Y.
Justificación...

b) Tabla de gestión de riesgos: (El RPN se calcula como $RPN = S \times O$)

Riesgo	S	O	RPN	S*	O*	RPN*

Criterio adoptado:

Se tomarán medidas de mitigación en los riesgos cuyos números de RPN sean mayores a...

Nota: los valores marcados con (*) en la tabla corresponden luego de haber aplicado la mitigación.

c) Plan de mitigación de los riesgos que originalmente excedían el RPN máximo establecido:

Riesgo 1: plan de mitigación (si por el RPN fuera necesario elaborar un plan de mitigación). Nueva asignación de S y O, con su respectiva justificación:

- Severidad (S*): mientras más severo, más alto es el número (usar números del 1 al 10). Justificar el motivo por el cual se asigna determinado número de severidad (S).
- Probabilidad de ocurrencia (O*): mientras más probable, más alto es el número (usar del 1 al 10). Justificar el motivo por el cual se asigna determinado número de (O).

Riesgo 2: plan de mitigación (si por el RPN fuera necesario elaborar un plan de mitigación).

Riesgo 3: plan de mitigación (si por el RPN fuera necesario elaborar un plan de mitigación).

14. Sprint Review

La revisión de sprint (*Sprint Review*) es una práctica fundamental en metodologías ágiles. Consiste en revisar y evaluar lo que se ha completado al finalizar un sprint. En esta instancia, se presentan los avances y se verifica si las funcionalidades cumplen con los criterios de aceptación establecidos. También se identifican entregables parciales y se consideran ajustes si es necesario.

Aunque el proyecto aún se encuentre en etapa de planificación, esta sección permite proyectar cómo se evaluarán las funcionalidades más importantes del backlog. Esta mirada anticipada favorece la planificación enfocada en valor y permite reflexionar sobre posibles obstáculos.

Objetivo: anticipar cómo se evaluará el avance del proyecto a medida que se desarrollen las funcionalidades, utilizando como base al menos cuatro historias de usuario del *Product Backlog*.

Seleccionar al menos 4 HU del Product Backlog. Para cada una, completar la siguiente tabla de revisión proyectada:

Formato sugerido:

HU seleccionada	Tareas asociadas	Entregable esperado	¿Cómo sabrás que está cumplida?	Observaciones o riesgos
HU1	Tarea 1	Módulo funcional	Cumple criterios de aceptación definidos	Falta validar con el tutor
	Tarea 2			
HU3	Tarea 1	Reporte generado	Exportación disponible y clara	Requiere datos reales
	Tarea 2			
HU5	Tarea 1	Panel de gestión	Roles diferenciados operativos	Riesgo en integración
	Tarea 2			
HU7	Tarea 1	Informe trimestral	PDF con gráficos y evolución	Puede faltar tiempo para ajustes
	Tarea 2			

15. Sprint Retrospective

La retrospectiva de sprint es una práctica orientada a la mejora continua. Al finalizar un sprint, el equipo (o el alumno, si trabaja de forma individual) reflexiona sobre lo que funcionó bien, lo que puede mejorarse y qué acciones concretas pueden implementarse para trabajar mejor en el futuro.

Durante la cursada se propuso el uso de la **Estrella de la Retrospectiva**, que organiza la reflexión en torno a cinco ejes:

- ¿Qué hacer más?
- ¿Qué hacer menos?

- ¿Qué mantener?
- ¿Qué empezar a hacer?
- ¿Qué dejar de hacer?

Aun en una etapa temprana, esta herramienta permite que el alumno planifique su forma de trabajar, identifique anticipadamente posibles dificultades y diseñe estrategias de organización personal.

Objetivo: reflexionar sobre las condiciones iniciales del proyecto, identificando fortalezas, posibles dificultades y estrategias de mejora, incluso antes del inicio del desarrollo.

Completar la siguiente tabla tomando como referencia los cinco ejes de la Estrella de la Retrospectiva (*Starfish* o estrella de mar). Esta instancia te ayudará a definir buenas prácticas desde el inicio y prepararte para enfrentar el trabajo de forma organizada y flexible. Se deberá completar la tabla al menos para 3 sprints técnicos y 1 no técnico.

Formato sugerido:

Sprint tipo y N°	¿Qué hacer más?	¿Qué hacer menos?	¿Qué mantener?	¿Qué empezar a hacer?	¿Qué dejar de hacer?
Sprint técnico - 1	Validaciones continuas con el alumno	Cambios sin versión registrada	Pruebas con datos simulados	Documentar cambios propuestos	Ajustes sin análisis de impacto
Sprint técnico - 2	Verificar configuraciones en múltiples escenarios	Modificar parámetros sin guardar historial	Perfiles reutilizables	Usar logs para configuración	Repetir pruebas manuales innecesarias
Sprint técnico - 8	Comparar correlaciones con casos previos	Cambiar parámetros sin justificar	Revisión cruzada de métricas	Anotar configuraciones usadas	Trabajar sin respaldo de datos
Sprint no técnico - 12 (por ej.: "Defensa")	Ensayos orales con feedback	Cambiar contenidos en la memoria	Material visual claro	Dividir la presentación por bloques	Agregar gráficos difíciles de explicar