



UNIVERSIDAD NACIONAL DE SAN AGUSTÍN

***FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y
SERVICIOS***

Proyecto:
Videojuego Educativo de Carreras en Unity

SEMANA 1: Planificación Técnica y Arquitectura Base

DOCENTE:
Ing. Richart Smith Escobedo Quispe

CURSO:
TECNOLOGÍA DE OBJETOS

CARRERA:
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

REPOSITORIO:
<https://github.com/rescobedoq/ryc>

- INTEGRANTES:**
- Choquehuanca Zapana, Hernan Andy
 - Larico Rodriguez, Bryan Fernando
 - Mamani Anahua Victor Narciso
 - Portugal Portugal, Eduardo Sebastián

FECHA
10 - 11 - 2025

Revisión de la idea principal y objetivos

El proyecto consiste en el desarrollo de un videojuego educativo de carreras en Unity, donde el jugador conduce un vehículo a través de un circuito que combina competencia, aprendizaje y toma de decisiones en tiempo real.

Durante la carrera, el jugador se enfrenta a preguntas educativas que aparecen en puntos estratégicos del circuito. En cada punto, se muestran tres portales translúcidos, cada uno representando una posible respuesta.

- Si el jugador atraviesa el portal correcto, el vehículo recibe un impulso de velocidad temporal.
- Si elige una respuesta incorrecta, la velocidad se reduce momentáneamente, obligando al jugador a recuperar el ritmo con habilidad.

Esta dinámica integra educación y entretenimiento, fomentando la atención, rapidez mental y aprendizaje activo, al mismo tiempo que ofrece una experiencia divertida e interactiva.

1. Objetivos específicos:

- 1.1. Diseñar un entorno de carrera funcional y atractivo.
- 1.2. Integrar un sistema de preguntas y respuestas educativas.
- 1.3. Implementar mecánicas de retroalimentación inmediata (aumento o disminución de velocidad).
- 1.4. Aplicar patrones de diseño orientados a objetos para mantener un código modular y escalable.
- 1.5. Promover el aprendizaje lúdico mediante una experiencia inmersiva.

Selección de patrones de diseño

Patrón	Uso dentro del proyecto
Singleton	Para gestionar componentes globales como GameManager y AudioManager, evitando duplicación y garantizando control centralizado.
Observer	Para la comunicación entre el QuestionManager y otros componentes (como UIManager o CarController) cuando se responde una pregunta.
Factory	Para la creación de objetos del entorno (portales, efectos visuales, objetos recolectables) de manera flexible y extensible.
State	Para manejar los distintos estados del vehículo (normal, acelerado, penalizado) según las respuestas o condiciones del circuito.
Strategy	Para variar dinámicamente la dificultad del juego o la respuesta del vehículo según el nivel o progreso del jugador.