

Hello, Job!

Preparación para tu entrevista de trabajo

Ayala Díaz, Alejandro de Jesús., Dzul Solís, Daniel Mauricio., Martín Ortiz, Miguel Ángel., Moreira Palma, Ángel Mauricio y Santamaría Sánchez, Julio

{15235400, 15246512, 15245739,
15245828}@modelo.edu.mx

Universidad Modelo

Resumen— Los egresados de Desarrollo de Tecnología y Software (DTS) de la Universidad Modelo enfrentan el reto de sentirse poco preparados para las entrevistas laborales, debido al nerviosismo y falta de práctica realista. Para solucionarlo, se creó Hello, Job!, un videojuego educativo que simula entrevistas técnicas de desarrollo de software. Los jugadores asumen el rol de candidatos, enfrentándose a preguntas técnicas y ejercicios de programación típicos de reclutamiento. Esta herramienta interactiva busca mejorar la confianza, lógica y habilidades de codificación.

Durante el desarrollo se cumplieron la mayoría de los objetivos, integrando modelos 3D, mecánicas funcionales y un editor de código. La inteligencia artificial genera preguntas dinámicas según el desempeño del jugador, eliminando diálogos predefinidos y enriqueciendo la experiencia. Las pruebas confirmaron que la aplicación simula eficazmente entrevistas técnicas, creando un entorno desafiante. Como proyección futura, se contempla expandir el proyecto a disciplinas como derecho o medicina, e implementar tecnologías inmersivas como realidad virtual. Hello, Job! es una propuesta escalable con alto potencial para mejorar la preparación profesional en múltiples campos.

Índice de Términos— *Videojuego educativo, simulación de entrevistas, modelos 3D, Realidad Virtual, entrevista de trabajo*

I. INTRODUCCIÓN

En la actualidad, el campo laboral para programadores se encuentra cada vez más saturado, debido al creciente número de aspirantes que buscan insertarse en la industria tecnológica. Ante esta competencia, las empresas tienden a seleccionar únicamente a los candidatos más preparados y competitivos, utilizando como principal filtro las entrevistas técnicas.

Dentro de la comunidad estudiantil de la Universidad Modelo, una parte considerable del alumnado de la carrera de Desarrollo de Software ha manifestado sentirse poco preparado para enfrentar una entrevista formal de programación. Esta inseguridad puede convertirse en un obstáculo determinante a la hora de conseguir un empleo.

Frente a esta problemática, el equipo Dream Team ha iniciado el desarrollo de un videojuego interactivo que simula entrevistas laborales reales en el ámbito del desarrollo de software. Esta herramienta integra escenarios realistas y desafíos de codificación en tiempo real, con el objetivo de preparar a los estudiantes de forma práctica y didáctica para el proceso de selección profesional.

II. FASE I: PREPARACIÓN Y PLANEACIÓN

A. Antecedentes

Los desarrolladores de software en México son profesionales encargados de crear, diseñar, implementar y mantener software, aplicaciones y sistemas informáticos. Este rol es esencial en diversas industrias, desde la medicina y la educación hasta el entretenimiento y la gestión de la información (Hireline, s.f.).

Como en cualquier entorno laboral, existen factores de estrés ocupacional que afectan a todos los empleados. Los estudios sugieren que los trabajos en TI suelen exigir largas jornadas laborales, horarios de viaje impredecibles, disponibilidad constante y una necesidad intensificada de mantenerse actualizado con tecnologías en constante cambio, especialmente en roles técnicos. Además, el entorno de desarrollo de software se caracteriza por plazos estrictos, un enfoque profundo en el cliente y la necesidad de un conocimiento técnico y especializado considerable. (Nayak, 2014).

Por un afortunado accidente, la industria del software parece haber reinventado un instrumento rudimentario pero efectivo para inducir estrés de manera confiable en los sujetos, lo que generalmente se manifiesta como ansiedad de desempeño (Wilson & Roland, 2002). La entrevista técnica tiene una sorprendente similitud con el Trier Social Stress Test, un procedimiento utilizado durante décadas por psicólogos y considerado el "estándar de oro" para inducir estrés de manera confiable. (Kirschbaum, Pirke & Hellhammer, 1993).

Si una entrevista técnica es esencialmente una versión de facto del Trier Social Stress Test, entonces las implicaciones pueden ser profundas. En lugar de medir a los pocos que responden correctamente en un tiempo adecuado, las empresas probablemente están evaluando la capacidad de aquellos que se desempeñan bien bajo estrés (Fox, 2019).

Se ha observado varias situaciones en las que la naturaleza de la entrevista interfiere con la capacidad del candidato para realizar su tarea. Varios cambios en el procedimiento de la entrevista pueden reducir los efectos del estrés y la carga cognitiva (de Alwis, Murphy & Minto, 2008). Por ejemplo, el formato de la entrevista técnica es sustancialmente diferente de cómo los desarrolladores realizan sus actividades de programación en el día a día. Para reducir el estrés, se recomienda proporcionar al candidato un ejercicio de calentamiento que le brinde la oportunidad de familiarizarse con el entorno de la entrevista, experimentar el formato de la entrevista y hacer preguntas sobre el proceso de la entrevista técnica. (Hunter, 2009)

La investigación muestra que el juego de roles y el entrenamiento en entornos de realidad mixta conducen a entrevistas de trabajo exitosas (Walker et al., 2016). Los investigadores han demostrado resultados positivos en la instrucción asistida por tecnología para personas con discapacidades intelectuales y del desarrollo (IDD, por sus siglas en inglés) (Burke et al., 2021).

De manera similar, un sistema de orientación basado en juego de roles para entrevistas de trabajo, que utiliza un robot androide con apariencia humana,

ha sido efectivo en la enseñanza de habilidades para entrevistas de trabajo en individuos con trastorno del espectro autista. Aunque los hallazgos de estos estudios son prometedores, estas intervenciones requieren dispositivos tecnológicos avanzados, como software de realidad virtual costoso o hardware robótico, que no están disponibles de manera gratuita para todos. Se deben explorar métodos virtuales más entables para enseñar estas habilidades (Kumazaki et al., 2019)

B. Planteamiento del problema

Los jóvenes egresados de la carrera de Desarrollo de Tecnología y Software (DTS) de la Escuela de Ingeniería de la Universidad Modelo campus Mérida no se sienten preparados para afrontar una entrevista de trabajo. Uno de los principales problemas que mencionan es el nerviosismo provocado por la falta de preparación para estas.

Para abordar esta problemática de manera innovadora, se propone el desarrollo de un videojuego simulador que no solo prepare a los jóvenes, sino que también haga el aprendizaje más dinámico y entretenido. En este, el jugador asumirá el rol de un candidato que se enfrenta a una entrevista de trabajo para un puesto en el área de desarrollo de software. A lo largo del juego, se presentarán escenarios realistas con preguntas sobre la experiencia y los conocimientos técnicos del aspirante, así como ejercicios destinados a ser resueltos mediante programación.

C. Objetivos

Objetivo general.

Desarrollar un videojuego simulador de entrevistas de trabajo del sector de software que permita a los estudiantes recientemente egresados practicar las habilidades requeridas, así como las situaciones a las que pueden exponerse en estas, con la finalidad de que los profesionales tengan una mejor preparación y sus probabilidades de ser aceptados aumenten.

Objetivos específicos.

1. Crear modelos 3D de buena calidad que sigan con la línea estética del juego y, en su conjunto, mejoren la atmósfera y la inmersión.

2. Idear problemas, preguntas y ejercicios para presentarle al jugador durante la entrevista que contribuyan a mejorar la preparación de la persona para la situación real.

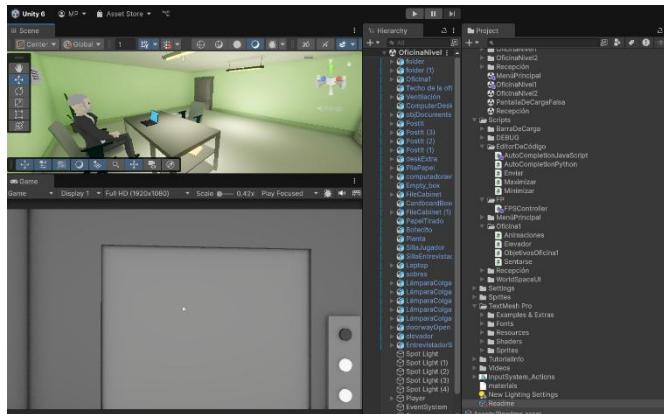
3. Diseñar e implementar mecánicas de juego que enriquezcan la experiencia, manteniéndola entretenida pero fiel a la realidad.

III. FASE II: DISEÑO E IMPLEMENTACIÓN

A. Diseño

Durante la realización del diseño se buscó que la estética del juego fuera minimalista, sin texturas muy detalladas o gráficos realistas. Esta decisión de diseño fue por dos motivos principales: la estética acogedora que este enfoque proporciona y la mayor facilidad de implementación.

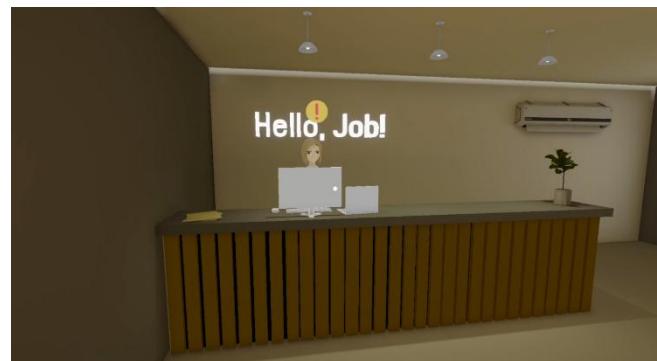
Para el desarrollo del videojuego se escogió el motor de videojuegos Unity Engine 6, el cual emplea el lenguaje de programación C# para el codificado.



Pantalla del entorno de desarrollo de Unity



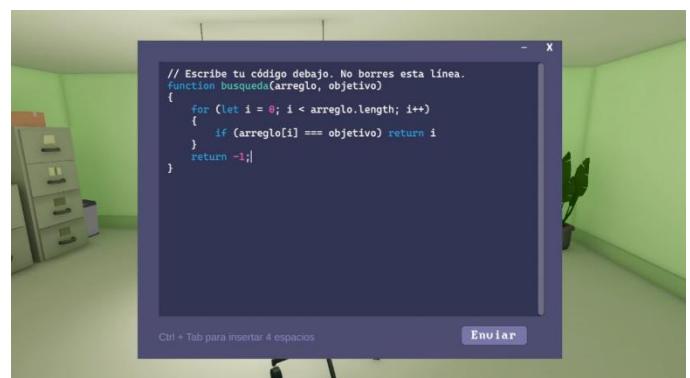
Pantalla del menú de selección de apodo y lenguaje a utilizar en la entrevista técnica.



Escena de la recepción inicial, en la que el usuario tiene la primera interacción con la recepcionista (quien dará instrucciones básicas al usuario)



Escena de la sala de entrevistas nivel principiante. Sería esta la escena en la que se desarrollaría la dinámica de entrevista.



Ventana del editor de código integrado, el cual cuenta con autocompletado de caracteres especiales y resaltado de sintaxis.

Para el desarrollo del backend se utilizó Node.js como entorno de ejecución principal, gracias a su eficiencia en aplicaciones orientadas a eventos y su capacidad para manejar múltiples conexiones concurrentes. Se implementó un servidor REST utilizando Express.js, lo que permitió estructurar de forma clara las rutas, middlewares y controladores necesarios para la interacción entre el juego (Unity) y los servicios externos. Uno de los componentes clave fue la integración con Supabase, una

plataforma backend-as-a-service basada en PostgreSQL que permitió la gestión de usuarios, almacenamiento y autenticación mediante JWT, facilitando una configuración segura y escalable sin necesidad de desarrollar toda la lógica desde cero.

Además, para la ejecución y evaluación de ejercicios de código en tiempo real, se integró la API de Judge0, lo cual permitió compilar y ejecutar código en diferentes lenguajes directamente desde el backend. También se aplicaron buenas prácticas de estructuración del código, como separación por capas (rutas, controladores, servicios), y el uso de herramientas como Postman para pruebas de endpoints y validación de flujos antes de su conexión con el frontend.

La principal delimitación del proyecto es la capacidad relativamente limitada de la inteligencia artificial con la que se comunicará el producto pues, con intención de no elevar más los costos del proyecto, se está optando por el modelo gratuito de Gemini de Google, la cual es más lenta que los modelos de paga. Otra limitación viene dada por los recursos mínimos que debe tener el equipo en el que se ejecute el juego, el cual, si bien no tiene un apartado gráfico muy exigente, emplea tecnologías que pueden requerir de una buena capacidad de cómputo y conexión de red de calidad.

B. Protocolo de pruebas

Al proyecto se le realizaron pruebas en cinco secciones: menú principal, recepción del juego, escena de desarrollo de la entrevista, editor de código integrado y backend.

Las pruebas realizadas para el frontend (las primeras cuatro secciones) comprenden las relacionadas con mecánicas de juego, funcionalidad de interfaces, configuraciones y el apartado visual. La prueba del backend consistió en una llamada de prueba a la API de Gemini para validar que esta responda correctamente y con la información que se le solicita.

En general, la inmensa mayoría de las pruebas se desarrollaron exitosamente, a excepción de dos, correspondientes al apartado del frontend. Cabe aclarar que las pruebas que no resultaron exitosas no corresponden a funcionalidades necesarias para el correcto funcionamiento del juego, mas ya se está

trabajando en resolver los problemas que se presentaron durante las pruebas. Ver la tabla del protocolo de pruebas en la sección de anexos para mayor detalle.

C. Análisis de resultados

Tras la realización de las pruebas correspondientes, se analizaron los resultados obtenidos. La inmensa mayoría de los resultados esperados fueron obtenidos en la cantidad mínima de intentos. Esto nos indica que la mayoría de los objetivos específicos se cumplieron satisfactoriamente. Además, ninguna de las pruebas fallidas corresponde a funciones críticas del producto, sino más bien a comodidades y añadidos que elevan la calidad del producto final.

En cuanto al primer objetivo —crear modelos 3D de buena calidad que respetaran la línea estética del juego y aportaran a la atmósfera e inmersión—, se logró satisfactoriamente. Esto se debe a que se trabajó con modelos licenciados y softwares especializados que facilitaron su integración al entorno del juego. Los elementos visuales, combinados con el diseño de escenas, brindaron al proyecto una estética distintiva que mejora notablemente la experiencia del usuario.

Respecto al objetivo de diseñar los problemas, preguntas y ejercicios que se le presentan al jugador durante la entrevista simulada, se superaron las expectativas. Aunque se contaba con una base inicial de preguntas, la incorporación de inteligencia artificial permitió una mayor adaptabilidad en tiempo real, eliminando la necesidad de configurar rutas de respuesta predefinidas. Esto no solo enriqueció la dinámica del juego, sino que también incrementó el grado de personalización para cada sesión, acercando aún más la experiencia a una entrevista real.

Por último, el diseño e implementación de las mecánicas de juego representaba uno de los retos más complejos del proyecto. La dificultad radicaba en lograr que el juego no solo fuera funcional, sino también entretenido y coherente con la realidad de una entrevista de programación. Sin embargo, este objetivo también fue cumplido: se desarrollaron mecánicas sólidas, intuitivas y atractivas, que lograron mantener el interés del jugador sin comprometer el realismo de la simulación.

Neurobiology of Stress, 6, 113–126.
<https://doi.org/10.1016/j.ynstr.2016.11.001>

IV. CONCLUSIONES

La creación de Hello, Job! representa un paso significativo hacia la formación práctica de estudiantes que buscan insertarse en un entorno laboral cada vez más exigente. Al combinar la lógica de un videojuego con situaciones reales de entrevistas técnicas, el proyecto no solo atiende una necesidad inmediata dentro del área de desarrollo de software, sino que también propone una forma innovadora y accesible de entrenar habilidades críticas para la empleabilidad en la nueva realidad con la IA como actor principal.

Planeamos expandir su alcance hacia otras disciplinas como derecho o medicina, adaptando escenarios y dinámicas específicas a cada campo profesional. Además, consideramos la integración de tecnologías emergentes como la realidad virtual (VR), lo cual permitiría una experiencia aún más inmersiva y realista. De este modo, Hello, Job! puede convertirse en una plataforma transversal para la preparación profesional, adaptable a múltiples contextos educativos y laborales.

REFERENCIAS

- Hireline. (s.f.). Perfil y sueldo de un Perfil de Desarrollador / Developer en México. Recuperado de <https://hireline.io/mx/encyclopedia-de-perfiles-de-tecnologia/desarrollador-it>.
- Freeman P, Aspray W. (1999) The supply of information technology workers in the United States. Computing Research Association. 1999. Available at http://archive.cra.org/reports/wits/chapter_2.html.
- Deloitte. (2020) Mental health and employers. Refreshing the case for investment. 2020. Available At <https://www2.deloitte.com/content/dam/Deloitte/uk/Documents/consultancy/deloitte-uk-mental-health-and-employers.pdf>.
- Allen, A. P., Kennedy, P. J., Dockray, S., Cryan, J. F., Dinan, T. G., & Clarke, G. (2017). The Trier Social Stress Test: Principles and practice. Neurobiology of Stress, 6, 113–126. <https://doi.org/10.1016/j.ynstr.2016.11.001>
- Kirschbaum, C., Pirke, K. M., & Hellhammer, D. H. (1993). The "Trier Social Stress Test": A tool for investigating psychobiological stress responses in a laboratory setting. *Neuropsychobiology*, 28(1-2), 76–81. <https://doi.org/10.1159/000119004>
- Wilson, G. D., & Roland, D. (2002). Performance anxiety. En *The Science and Psychology of Music Performance: Creative Strategies for Teaching and Learning* (pp. 47–61).
- Fox, C. (2019). It's time to retire the whiteboard interview. Hackernoon. Recuperado de <https://hackernoon.com/its-time-to-retire-the-whiteboard-interview-qyr32sd>.
- Hunter, A. (2009). High-tech rascality: Asperger's Syndrome, hackers, geeks, and personality types in the ICT industry. *New Zealand Sociology*, 24(2), 215–222.
- Walker, Z., Vasquez, E., & Wienke, W. (2016). The impact of simulated interviews for individuals with intellectual disability. *Journal of Educational Technology & Society*, 19(1), 76–88.
- Burke, S. L., Li, T., Grudzien, A., & Garcia, S. (2021). Brief report: Improving employment interview self-efficacy among adults with autism and other developmental disabilities using virtual interactive training agents (ViTA). *Journal of Autism and Developmental Disorders*, 51(2), 741–748.
- Kumazaki, H., Muramatsu, T., Yoshikawa, Y., Matsumoto, Y., Ishiguro, H., Mimura, M., & Kikuchi, M. (2019). Role-play-based guidance for job interviews using an android robot for individuals with autism spectrum disorders. *Frontiers in Psychiatry*, 10, Article 239.
- Clark, D. B., Tanner-Smith, E. E., & Killingsworth, S. S. (2016). Digital games, design, and learning: A systematic review and meta-analysis. *Review of Educational Research*, 86(1), 79–122. <https://doi.org/10.3102/0034654315582065>
- Behroozi, M., Parnin, C., & Barik, T. (2019). Hiring is broken: What do developers say about technical interviews? *Visual Languages & Human-Centric Computing (VL/HCC)*, 1–9. <https://doi.org/10.1109/VLHCC.2019.8818836>
- Ford, D., Barik, T., Rand-Pickett, L., & Parnin, C. (2017). The tech-talk balance: What technical interviewers expect from technical candidates. 2017 IEEE/ACM 10th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE), 43–48. IEEE.
- Saunders, T., Driskell, J. E., Johnston, J. H., & Salas, E. (1996). The effect of stress inoculation training on anxiety and performance. *Journal of Occupational Health Psychology*, 1(2), 170. <https://doi.org/10.1037/1076-8998.1.2.170>

ANEXOS

Anexo 1: Listado de softwares utilizados

- Canva: Software web utilizado en la realización de diseños preliminares y conceptuales.
- Blender: Software libre empleado para la creación y manipulación de algunos modelos 3D.
- Unity Engine 6: Motor de videojuegos en el que se realizó el ensamblaje del proyecto.
- Visual Studio: Entorno de Desarrollo Integrado (IDE) en el que se realizó la programación de los scripts del frontend.
- Visual Studio Code: Editor de código en el que se realizó la programación del backend.
- C#: Lenguaje de programación compilado en el que se escribieron los scripts del juego.
- JavaScript/Node.js/Express.js: Implementación del lenguaje de programación JavaScript para el desarrollo del backend.
- Supabase: Plataforma backend-as-a-service basada en PostgreSQL que permitió la gestión de usuarios, almacenamiento y autenticación mediante JWT

Anexo 2: Simulación

Este enlace redirige al video del producto en funcionamiento:

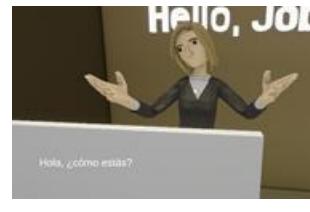
https://drive.google.com/file/d/11zUV5L1mydxYK_QuAIPO4tXUJsiAKwDj/view?usp=sharing

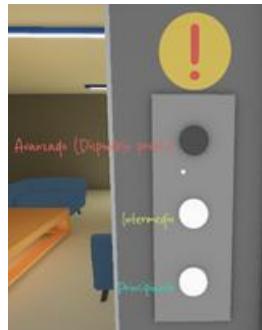
Anexo 3: Protocolo de pruebas

1. Menú principal		
Prueba	Resultado esperado	Resultado obtenido
Pantalla inicial funcional y responsive	La pantalla inicial (previa al menú de opciones) debe reaccionar a la pulsación de una tecla y realizar un recorrido de cámara hacia el menú principal.	<u>Primer intento:</u> La pantalla fue responsive y reaccionó correctamente al input. 
Funcionalidad de los botones	Los botones deben realizar su acción correspondiente (Iniciar partida, mostrar créditos o salir al escritorio)	<u>Primer intento:</u> Los botones responden de manera adecuada y realizan las acciones que les corresponden. 
Menú de ajustes	Se abre el menú de ajustes al presionar su ícono, se muestran las opciones y cada opción cambia la configuración interna.	<u>Primer intento:</u> Se verificó que todas las opciones aplican sus respectivos cambios adecuadamente. 
Menú de selección de apodo/lenguaje	El apodo ingresado y el lenguaje seleccionado se	<u>Primer intento:</u> Al ingresar posteriormente al archivo de

	guardan en las preferencias del jugador.	preferencias se verificó que sí se guardan correctamente los datos proporcionados.
--	--	--

2. Escena de la recepción

Prueba	Resultado esperado	Resultado obtenido
Animaciones de la recepcionista	Se activan diferentes animaciones para el modelo 3D de la recepcionista según la situación.	<p><u>Primer intento:</u> Las animaciones cambian correctamente a la que corresponde a la situación actual.</p>  
Diálogos	Los diálogos de la recepcionista cambian al presionar la tecla espacio.	<p><u>Primer intento:</u> Los diálogos cambian correctamente.</p>  
Marcador de objetivos	El marcador de objetivos actualiza su posición cuando el objetivo actual cambia.	<p><u>Primer intento:</u> El marcador de objetivo se dirige a la posición del siguiente objetivo a realizar.</p>

		  
Botones del elevador (selector de dificultad)	Cada uno de los botones disponibles (principiante e intermedio) llevan a su respectivo nivel de dificultad.	<p><u>Primer intento (nivel principiante):</u> La escena del nivel principiante se carga correctamente.</p>  <p><u>Segundo intento (nivel intermedio):</u> Error. La escena no se carga, ya que aún no se ha realizado el enlace del botón con esta.</p>
3. Escena de entrevista		
Prueba	Resultado esperado	Resultado obtenido
Animaciones del entrevistador	El entrevistador tiene una animación “idle” y animaciones distintas para situaciones específicas.	<u>Primer intento:</u> Error. El entrevistador tiene la animación “idle” (animación para cuando no esté haciendo nada) pero no se han implementado las demás.

			
Sentarse e interactuar	El jugador puede sentarse y levantarse para iniciar la interacción con el entrevistador.	<u>Primer intento:</u> El jugador se puede sentar correctamente para iniciar la interacción. 	

4. Editor de código integrado

Prueba	Resultado esperado	Resultado obtenido
Resaltado de sintaxis	El texto introducido en el campo de texto del editor de código resalta palabras clave del código con colores distintos.	<u>Primer intento:</u> El resultado de sintaxis funciona correctamente. Se resaltan nombres de funciones, tipos de dato, estructuras de control de flujo, cadenas y valores numéricos. 
Autocompletado de caracteres especiales	Si se abren comillas, paréntesis, corchetes o llaves se cierran automáticamente.	<u>Primer intento:</u> Los caracteres mencionados se autocompletan correctamente.
Indentado automático	Se generan espaciados automáticos para resaltar los niveles del código.	<u>Primer intento:</u> Error. El indentado automático no se ha terminado de implementar y, por lo tanto, no está funcionando.

5. Backend

Prueba	Resultado esperado	Resultado obtenido
Respuesta de la API de Gemini	Al mandarle una solución de ejercicio de código a la API de Gemini, esta	<u>Primer intento:</u> La API responde correctamente y devuelve lo que

	devuelve un JSON con los valores de las claves “éxito”, “stdout”, “status” y “feedback”, que luego serán las que se mostrarán al usuario.	se le solicita al enviarle una solución.
--	---	--

Anexo 4: Diagrama de Gantt



UNIVERSIDAD MODELO
ESCUELA DE INGENIERÍA
DTS

PLAN DE PROYECTO

Materia: PROYECTOS II Grupo: DTS-2A Profesor: Kenia Osorio.

Alumnos: Alejandro de Jesús Ayala
Díaz, Daniel Mauricio Dzul Solís,
Angel Mauricio Moreira Palma,
Miguel Angel Martín Ortiz, Julio
Santamaría Sánchez

No.	Actividades	Responsable	Entrega	Febrero				Marzo				Abril				Mayo				Junio				
				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	29	20	
Proyecto - Hello Job																								
1	Desarrollo de propuesta de proyecto	Alejandro	4 de febrero																					
2	Entrevistas y levantamiento de requisitos.	Miguel	10 de febrero																					
3	Cotización y elección de tecnologías	Daniel	18 de febrero																					
4	División y delegación de tareas.	Alejandro	18 de febrero																					
5	Desarrollo de modelos 3d en blender.	Mauricio	28 de febrero																					
6	Desarrollo de mockups y bocetos de niveles.	Mauricio	28 de febrero																					
7	Diseño de personajes	Julio	28 de marzo																					
8	Conexion API y Backend	Alejandro	16 de mayo																					
9	Programación de mecánicas	Mauricio	16 de mayo																					
10	Integración de sonido/música	Miguel y Daniel	23 de mayo																					
11	Desarrollo de plan de contenidos y medios	Julio	26 de mayo																					
12	Pruebas internas	Miguel y Daniel	4 de junio																					
13	Redacción de documentación	Julio	5 de junio																					
14	Presentación final	Alejandro	11 de junio																					

Anexo 5: Listado de costos

Concepto	Costo mensual	Costo anual
Licencias de software	\$14,167	\$170,004
Nómina (sueldos)	\$50,000	\$600,000
Servidores	\$1,545	\$18,540
TOTAL	\$65,712	\$788,544