



**Desarrollo de componentes y servicios frontend y backend para plataforma de personas
digitales NATI**

Gilco Delgado Cadavid

Informe de práctica para optar al título de
Ingeniero de Sistemas

Tutor

Diana Margot Lopez Herrera
Ingeniería de sistemas.

Universidad de Antioquia
Facultad de Ingeniería
Ingeniería de sistemas
Medellín
2024

Cita	Delgado Cadavid [1]
Referencia	[1] Delgado Cadavid, “Desarrollo de componentes y servicios frontend y backend para plataforma de personas digitales NATI”, Semestre de industria, Ingeniería de sistemas, Universidad de Antioquia, Medellín, 2024.

Estilo IEEE (2020)



CENDOI

Repositorio Institucional: <http://bibliotecadigital.udea.edu.co>

Universidad de Antioquia - www.udea.edu.co

El contenido de esta obra corresponde al derecho de expresión de los autores y no compromete el pensamiento institucional de la Universidad de Antioquia ni desata su responsabilidad frente a terceros. Los autores asumen la responsabilidad por los derechos de autor y conexos.

Dedicatoria

A mi familia, pilar fundamental en mi paso por la universidad en lo económico, pero sobre todo, en lo anímico, gracias por apoyarme en cada paso.

Agradecimientos

A Diana Margot López.

A el semillero de videojuegos de la Universidad de Antioquia.

A mis amigos y compañeros de la carrera.

A Pablo Jaramillo y al equipo de Innovati S.A. que hicieron esta experiencia posible.

TABLA DE CONTENIDO

RESUMEN	5
ABSTRACT	6
I. INTRODUCCIÓN	8
II. PLANTEAMIENTO DEL PROBLEMA	9
III. JUSTIFICACIÓN	10
IV. OBJETIVOS	11
A. Objetivo general	11
B. Objetivos específicos	11
V. MARCO TEÓRICO	12
VI. METODOLOGÍA	15
VII. RESULTADOS	17
VIII. CONCLUSIONES	23
REFERENCIAS	26

LISTA DE FIGURAS

Fig. 1 Versión inicial de la herramienta de configuración rápida de avatares.	18
Fig. 2 Versión actual de la herramienta de configuración rápida de avatares	18
Fig. 3 Comparación de avatar en Unity y ThreeJS	21

RESUMEN

Este informe da cuenta del trabajo realizado durante el proceso de prácticas profesionales en la empresa de desarrollo de software e innovación Innovati SAS. El trabajo se centró en el desarrollo de la aplicación Nati, un software como servicio que ofrece asistentes virtuales para empresas. Nati es un conjunto de aplicaciones que se impulsa por grandes modelos de lenguaje basados en inteligencia artificial generativa para brindar servicios de atención al cliente, asesoramiento de productos y más.

Durante el tiempo de prácticas se abordó el proyecto en tres frentes. El primero como desarrollador en Unity, manteniendo y mejorando el proyecto en busca de la automatización de integración de avatares. El segundo como desarrollador frontend con el framework de JavaScript Angular, integrando características a la aplicación, principalmente la adopción de ThreeJS como motor gráfico adicional a Unity. El tercer frente, realizando vigilancia tecnológica en tecnologías de IA generativa para la creación de avatares fotorrealistas en tiempo real. Como resultado, el ecosistema Nati cuenta con una muy buena alternativa para el uso de personas digitales, más optimizada para la web y con mejor aspecto visual, además queda encaminada a la implementación de tecnologías de IA generativa de vanguardia.

***Palabras clave* — Persona Digital, Unity, WebGL, ThreeJS, Avatar, Angular, IA generativa**

ABSTRACT

This report provides an account of the work carried out during the professional internship at the software development and innovation company Innovati SAS. The work focused on the development of the Nati application, a software as a service that offers virtual assistants for companies. Nati is a suite of applications powered by large language models to provide customer service, product advice, and more.

During the internship period, the project was approached from three fronts. The first as a Unity developer, maintaining and improving the project in pursuit of automating avatar integration. The second as a frontend developer with the JavaScript framework Angular, integrating features into the application, mainly the adoption of ThreeJS as an additional graphics engine to Unity. The third involved technological surveillance in generative AI technologies for the creation of real-time photorealistic avatars. As a result, the Nati ecosystem has a very good alternative for the use of digital personas, more optimized for the web and with a better visual appearance, and is also on track for the implementation of cutting-edge generative AI technologies.

Keywords —Unity, Digital Person, WebGL, ThreeJS, Avatar, Angular, Generative AI

I. INTRODUCCIÓN

La práctica profesional, se realizó en la empresa Innovati SAS. Se ingresó como desarrollador Unity y frontend para colaborar activamente en los objetivos de la empresa con el producto de software Nati.

La “Plataforma de Personas Digitales Nati” es una solución tecnológica que automatiza procesos de interacción empresa-usuario mediante el uso de herramientas de procesamiento de lenguaje natural y de la representación de personas mediante avatares tridimensionales. La intención de Nati es brindar asistencia y gestionar información de los usuarios según sus necesidades. En ese sentido, la plataforma es capaz de llevar a cabo entrevistas, proporcionar asesoramiento sobre productos y resolver solicitudes y trámites de manera eficiente sin necesidad de intervención humana.

El ecosistema Nati se conforma a partir de aplicaciones web y backend que se comunican entre sí con el fin de proporcionar las funcionalidades requeridas por los clientes, modular y eficientemente. Nati se presenta como una plataforma de software como servicio con una estructura flexible que permite adaptarse a las necesidades de diferentes usuarios mediante una estructura multitenant que, además, facilita la configuración de su uso a través de un panel de administración tipo low-code con un alto grado de personalización.

Durante la práctica profesional se toman como principales objetivos el mantenimiento y automatización del proyecto de Unity para un proceso de integración de avatares rápido, la adición al proyecto de la librería de renderizado 3D optimizada para la web llamada ThreeJS y la vigilancia tecnológica en IA generativa para la creación de avatares fotorrealistas que interactúen con el usuario en tiempo real.

A continuación se presentan detalles técnicos y del proceso de práctica importantes

II. PLANTEAMIENTO DEL PROBLEMA

Innovati SAS requiere un practicante que pueda realizar las tareas de desarrollador de software con conocimientos en Unity y desarrollo frontend, específicamente en la presentación de un avatar en un entorno 3D usando la plataforma para navegadores WebGL.

Una de las aplicaciones de la plataforma Nati, encargada de presentar el asistente virtual al usuario y generar la interacción mediante una interfaz de chat, está basada en dos tecnologías que conviven para entregar la experiencia: Unity para la renderización del modelo 3D en la pantalla y Angular para la interfaz gráfica de usuario. Sin embargo, esta convivencia presenta problemas de rendimiento y fidelidad gráfica de parte de Unity, pues cargar su instancia en el navegador puede ser un proceso pesado. Esto crea la necesidad de explorar las posibles soluciones. Una de ellas es la integración de otro motor gráfico, nativo para la web, que alivie la carga y a su vez mejore la calidad visual del avatar. Con este objetivo en mente, la empresa plantea una secuencia de acciones de exploración, especificadas a continuación:

- Integración de la librería 3D de JavaScript ThreeJS a la aplicación web. Así se mantiene el comportamiento del asistente virtual, y se mejora el rendimiento de la aplicación.
- Adaptación del panel de administración de Nati, de manera que permita la elección del motor gráfico que se desea usar, ofreciendo los asistentes virtuales propios de cada uno.
- Exploración de alternativas fotorrealistas para el avatar. Implica aumentar la oferta de opciones para los clientes y sus necesidades

De estos tres objetivos específicos, el primero y el tercero están dentro de la línea de acción del desarrollador de Unity y de la aplicación web; en ese sentido, la práctica está orientada a trabajar en la resolución de dichos apartados.

El planteamiento del problema entonces se puede resumir en la necesidad de la aplicación frontend de Nati de ampliar su oferta en motores gráficos alineados con las características propias de un entorno web, siendo su primer objetivo replicar el comportamiento de Unity en ThreeJS.

III. JUSTIFICACIÓN

Como miembro del semillero de videojuegos de la Universidad de Antioquia desde los primeros semestres de la carrera, en donde se han desarrollado un cúmulo de conocimientos útiles en temas relacionados al desarrollo de entornos virtuales, movimiento tridimensional y animación, es deseable tener la oportunidad de ejercer la práctica académica en una empresa en donde tales conocimientos son fundamentales en su visión de negocio, ya que representa un reto de aplicación de saberes pero también de profundización en otras técnicas y tecnologías que promueven el desarrollo profesional.

El proyecto representa un reto a nivel técnico, ya que requiere un traspaso de los conocimientos adquiridos en una tecnología específica como Unity, al motor 3D ThreeJS, que se puede fundamentar en otros conceptos y estructuras lógicas y matemáticas. Lo anterior precisa entonces de un ingeniero de sistemas en formación que pueda abstraer los fundamentos en el funcionamiento de motores gráficos para lograr los objetivos en la implementación de los asistentes virtuales en ThreeJS.

Las funcionalidades y optimizaciones que se lograron con Unity para que funcionara correctamente en la aplicación web, aunque excelentes, están en un punto en donde hay poco margen de mejora. Por eso, uno de los retos principales es traducir optimizaciones específicas de Unity a ThreeJS, entre estas están el sistema de Addressables que permite la carga dinámica del avatar en la aplicación web desde la nube, así como la implementación del sistema de animaciones y sincronización de labios.

Adicionalmente, Innovati requiere un desarrollador flexible, dispuesto a la adquisición de conocimientos y a la vigilancia tecnológica constante con el fin de sentar las bases para la innovación y mejora continua de la plataforma Nati. Esto incluye la búsqueda de otras alternativas de representación digital de personas basadas en el fotorrealismo, así como la implementación de nuevas características en la aplicación web con Angular.

IV. OBJETIVOS

A. Objetivo general

Mejorar el rendimiento de la aplicación web Nati en su componente de asistente digital animada introduciendo ThreeJS como librería gráfica nativa para la web, manteniendo y mejorando la calidad de los avatares en su apartado gráfico, de animación y de mecánicas asociadas.

B. Objetivos específicos

- Comprender la arquitectura limpia simple de Nati-Unity para contribuir en la optimización y automatización del proyecto.
- Interiorizar los conceptos clave en el desarrollo de proyectos 3D con ThreeJS para la web y plantear una arquitectura adecuada para su integración en la aplicación Nati.
- Construir el módulo de ThreeJS en la aplicación, desde el proceso de integración de nuevos avatares, la carga de estos en la interfaz desde la nube, la sincronización labial, las animaciones y la comunicación con los servicios pertinentes.
- Ejercer vigilancia tecnológica en el uso de IA generativa para la creación de avatares fotorrealistas que se comuniquen en tiempo real, generando propuestas viables para el futuro de la aplicación.
- Integrar nuevas funcionalidades en la aplicación web Angular de acuerdo con las necesidades de los clientes.

V. MARCO TEÓRICO

A continuación se presentan herramientas, tecnologías y conceptos necesarios para comprender las acciones llevadas a cabo en el marco de la práctica académica:

Persona Digital: Acorde a lo planteado por Innovati. Una persona digital es un avatar humano con capacidades para escuchar, comprender e interactuar con seres humanos expresando emociones y generando empatía[1].

Procesamiento de lenguaje natural: NLP (de sus siglas en inglés Natural Language Processing) es una rama de la inteligencia artificial que busca mezclar la lingüística computacional, entendida como la interpretación del lenguaje humano a partir de ciertas reglas, con modelos estadísticos y del campo del machine learning. En general, los casos de uso se enfocan en otorgar a diferentes dispositivos como móviles, computadoras e IoT, la capacidad de reconocer, comprender y generar texto y voz, incluso en tiempo real. Ejemplos de NLP se puede encontrar en traductores de idiomas, chatbots de atención al cliente, autenticación de usuarios por voz y generación de texto[2].

ThreeJS: Three.js es una librería 3D cuyo objetivo principal es facilitar a desarrolladores la creación de contenido tridimensional en la web. La librería está altamente relacionada con el sistema WebGL, específicamente encapsula el código de bajo nivel de WebGL en funciones que hacen la parte más complicada, de manera que el desarrollador simplemente las use para crear escenas, luces, objetos, etc., sin programar complejos algoritmos tridimensionales [3].

ThreeJS es una de las tecnologías fundamentales en el desarrollo de prácticas, mientras que posee cierta desventaja respecto a Unity en cuanto la facilidad de uso a través de una interfaz de usuario y en la preparación de texturas y materiales del modelo 3D de una persona digital, es compensada por el alto rendimiento y calidad que presenta dentro de la aplicación.

API: Abreviación de “Interfaz de programación de aplicaciones”, se trata de un conjunto de reglas y protocolos que definen las formas en las que una aplicación puede comunicarse con otra. La interfaz entonces actúa como un “contrato de servicio” entre aplicaciones que hacen cosas distintas pero que pueden compartir ciertos datos para su funcionamiento. El formato de este contrato se basa en solicitudes y respuestas, es decir, un sistema que requiere cierta

información la solicita a otro sistema, que consecuentemente, responde al primero con los datos solicitados [4].

IA generativa: Se refiere a un tipo de la inteligencia artificial que tiene la capacidad de generar ideas y contenidos que no han sido creados con anterioridad por los seres humanos. Dentro de los ejemplos más comunes se encuentran producción de conversaciones, imágenes, videos y sonidos. Su funcionamiento está muy asociado con los modelos de procesamiento de lenguaje natural, los cuales identifican relaciones existentes entre la gran cantidad de datos con que se entrenan, para luego producir contenido de acuerdo a las peticiones hechas por los usuarios en lenguaje común [5].

Este es un concepto fundamental dentro del desarrollo de la práctica académica. Como empresa enfocada en la innovación, Innovati precisa de la implementación de tecnologías de vanguardia que agreguen valor a sus productos y servicios, especialmente a la plataforma Nati, por lo que la vigilancia tecnológica en IA generativa para mejoras en las funcionalidades de la aplicación y especialmente del avatar, es un objetivo claro.

Talking-heads: Se trata de un campo de la inteligencia artificial generativa que busca recrear en video la parte superior del cuerpo de un ser humano de manera fotorrealista a partir de un audio, esto implica la generación de los movimientos faciales o visemas, además de la sincronización del movimiento labial con la voz proporcionada en el audio. A esta técnica se le conoce como “audio driven talking face” [6]. El acercamiento común hacia este objetivo es a través de usos de arquitecturas de deep-learning que permiten multimodalidad en los datos de entrada para proporcionar un resultado como salida del modelo.

Unity: Es un motor de desarrollo de experiencias 2D y 3D multiplataforma, comúnmente asociado al desarrollo de videojuegos. Es una de las herramientas más usadas por su bajo costo (gratis en muchos casos), su curva de aprendizaje notablemente menor al de productos de su estilo. A partir de su interfaz de usuario, presenta facilidades en la implementación de lógica, construcción de mundos, físicas, animación y un largo etcétera. Como lenguaje de programación usa C# [7]. Entre otras ventajas, posee una gran comunidad activa de usuarios que constantemente colaboran en la resolución de problemas y generación de recursos de aprendizaje.

WebGL™: Es una especificación para la renderización de gráficos 3D en la gran mayoría de navegadores web, tanto en escritorio como en móviles. El estándar es expuesto a través de un API en JavaScript y está basado en OpenGL. Soporta diferentes usos como la aceleración de hardware en la GPU y el procesamiento de efectos visuales, todo esto en combinación con elementos HTML como el canvas para exponerlo en la interfaz de usuario dentro del navegador. [8].

Angular: Es un framework diseñado para el desarrollo de SPA (Single Page Applications) que ofrece una forma robusta de creación de páginas web dinámicas y responsivas basada en componentes que, además, son reusables. Estos componentes constan, esencialmente, de tres archivos: HTML, CSS y TypeScript, siendo este último un “superconjunto” de ECMAScript6, lo cual le da las funcionalidades de Javascript más otras que amplían sus capacidades. Esta separación a su vez permite desacoplar la lógica de la página y su estética [9].

VI. METODOLOGÍA

La metodología de trabajo se basó en SCRUM con algunas alteraciones que obedecen a las dinámicas de desarrollo propias de la empresa que se han venido gestando en los últimos años con las experiencias en distintos proyectos, especialmente Nati. A continuación se presentan las actividades que la componen.

Para empezar, el product manager en conjunto con el equipo de trabajo, plantea una serie de objetivos que debe alcanzar la plataforma Nati en los próximos meses. Estos vienen determinados a partir de múltiples fuentes: vigilancia tecnológica en las que se descubren mejoras técnicas o de experiencia de usuario, retroalimentación de los clientes, nuevos objetivos comerciales, etc. A partir de las definiciones alcanzadas, el equipo puede formar los objetivos a mediano y largo plazo, organizados normalmente en épicas o features.

Después de las pautas iniciales, el product owner se encarga de concretar la concepción de los objetivos en historias de usuario, asignando las prioridades adecuadas en términos de tiempo y esfuerzo por parte del equipo. Así, quedan determinadas las responsabilidades y los pasos generales a seguir para la construcción constante de la plataforma en los sprints venideros.

Todo el seguimiento del trabajo se realiza dentro de sprints, cuya duración es de 10 días cada uno. La actividad inicial dentro de estos es la planificación, la cual consta de un reunión del scrum master y el product owner y el equipo para descomponer las historias de usuario en tareas asignadas a integrantes del equipo específicos. Una historia de usuario puede involucrar una o más componentes de la aplicación, ya sea de parte de los servicios backend o de las aplicaciones frontend.

De la planeación se desprenden las actividades del equipo durante los días del sprint. Además, se reservan tres días por semana, o seis días por sprint, en los cuales el equipo destina un tiempo corto para que cada miembro comparta los últimos avances, inconvenientes y trabajos próximos con el resto. El contexto dado facilita la coordinación y cooperación entre las partes involucradas en la implementación de una historia de usuario, incluyendo el testeado de los trabajos completados por partes de la persona encargada de asegurar la calidad.

La finalización del sprint implica una revisión y aceptación por parte del product owner de las metas propuestas por el equipo, por lo que en esta actividad, llevada a cabo el último día del sprint o el día inicial del siguiente, se exponen los resultados en la implementación de las historias de usuario comprometidas. En el proceso se explican los procedimientos que fueron necesarios para la finalización, en caso de un resultado exitoso, mientras que en caso contrario, se detallan los motivos puntuales por los cuales no se pudo llevar a cabo el compromiso y se pospone, si así se considera, para el próximo sprint.

Por último, como ejercicio útil para dar por cerrado un sprint y comenzar uno nuevo, se hace una retroalimentación. Mediante herramientas didácticas que permiten al equipo expresar sus pensamientos respecto al desempeño técnico y humano, se genera una deliberación de ideas y argumentos cuyo fin es la mejora constante en todos los ámbitos profesionales. Las conclusiones quedan debidamente consignadas en la herramienta y se tienen en cuenta para la siguiente iteración.

VII. RESULTADOS

Por ser Nati un producto de propiedad de Consultoría GP S.A.S, marca Innovati Software, algunos de los conceptos técnicos considerados secreto industrial no se plasman aquí, y se da fé mediante las cartas de certificación de la compañía del correcto cumplimiento. A continuación, se presentan algunos aspectos técnicos relevantes de la práctica que pueden ser revelados a la comunidad académica.

A. Mantenimiento y automatización del proyecto Unity.

Dado que el rol adquirido inicialmente se enfocó en el desarrollo del proyecto Unity de Nati, fue mi responsabilidad mantener y mejorar sus características. Se hizo especial énfasis en la integración de nuevos avatares y en la automatización de este proceso, ya que seguía siendo en parte muy manual y por ende existía un margen de error en los pasos a seguir. Si bien el proyecto ya tenía avances significativos en este aspecto, el objetivo fue aumentar lo máximo posible la facilidad en su manejo, de manera que cualquier persona con un mínimo conocimiento en Unity pudiera interiorizar su funcionamiento con mayor facilidad.

Dentro de los avances logrados en este aspecto, se incluyen la configuración automática en el tamaño de las texturas del avatar, así como la organización de sus materiales, lo cual antes era tedioso y consumía mucho tiempo. Por otro lado, la creación de gestos faciales era inflexible, creando el mismo conjunto para todos los avatares; esto ignora la posibilidad de que la malla 3D de cada avatar varíe y por tanto el mismo gesto puede verse mal o bien en diferentes avatares. Por lo anterior, se introdujeron archivos de configuración propios para cada avatar y fácilmente modificables. En la figura 1 y 2 se puede observar el antes y el después de la ventana de automatización de Unity.

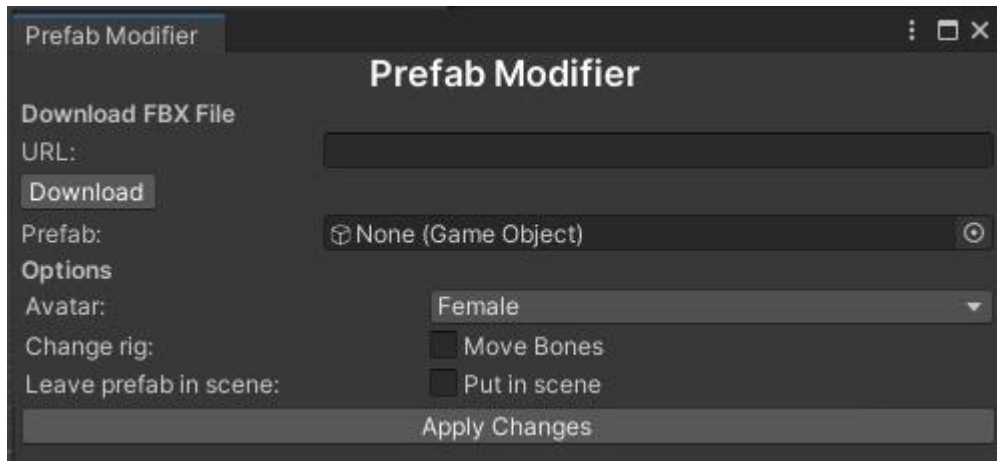


Fig. 1 Versión inicial de la herramienta de configuración rápida de avatares.

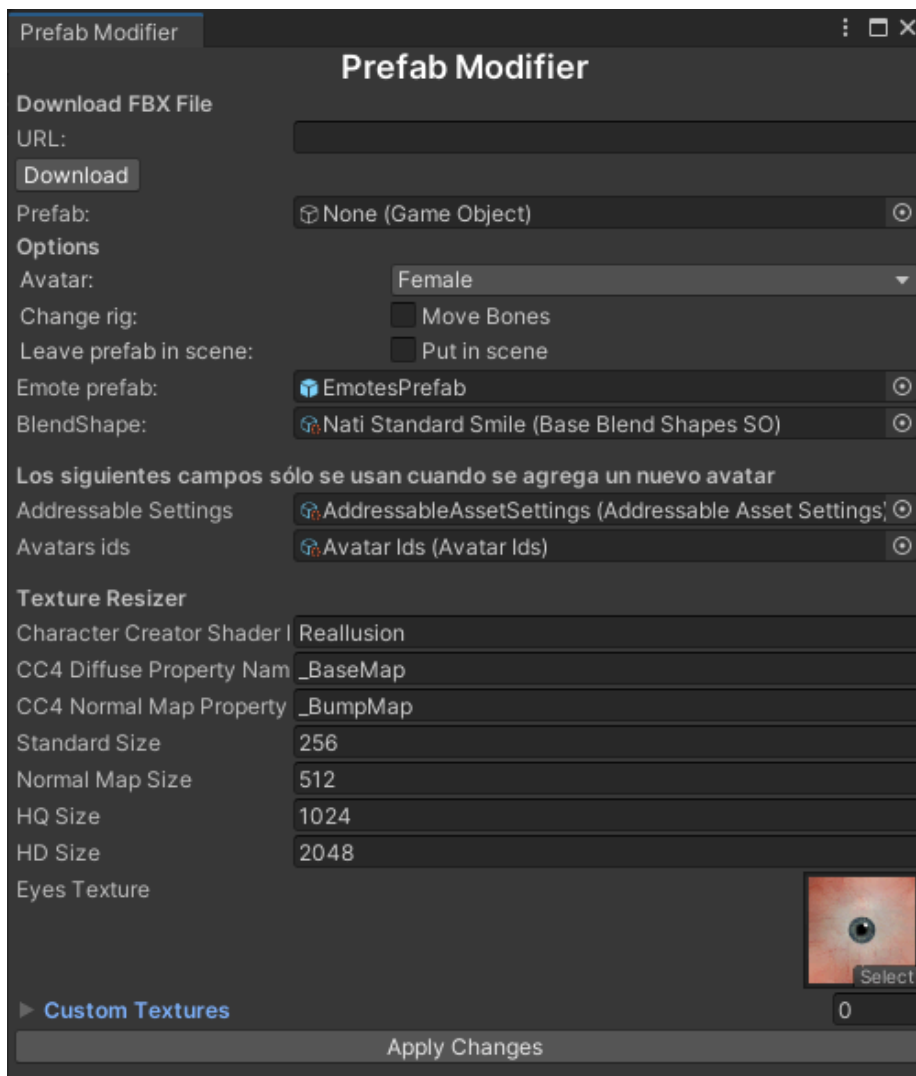


Fig. 2 Versión actual de la herramienta de configuración rápida de avatares.

Por último, el sistema que se usaba para manejar las expresiones faciales de las personas digitales se basaba en una librería externa, la cual sobre-complejizaba el comportamiento de las expresiones y además generaba muchas dependencias a clases internas que realmente no eran necesarias. Por esta razón, se construyó un algoritmo de control de gestos y transiciones entre estos, hecho a la medida de las necesidades del proyecto.

B. Adición al proyecto web de motor gráfico nativo.

Unity, como herramienta de desarrollo de videojuegos y experiencias tridimensionales goza de un gran prestigio, y con razón, pues las posibilidades que concede a sus usuarios son amplias y ofrece además actualizaciones constantes y una comunidad activa. Es también conocida su amplia oferta de plataformas en las cuales se puede desplegar un producto, incluyendo Windows, Playstation, Nintendo Switch, Xbox, plataformas de realidad virtual y aumentada, y, adicionalmente, para la web a través de WebGL.

Sin embargo, a pesar de soportar desarrollos en la web, la compatibilidad que Unity ofrece para Nati se ve en gran medida reducida por distintas razones, principalmente por la necesidad de cargar el motor dentro de la web, el cual por cierto puede resultar pesado para aplicaciones con cierta carga gráfica. Además, si se desea integrar Unity dentro de una aplicación web, inevitablemente la complejidad en ambas partes aumenta debido a las consideraciones adicionales para establecer una comunicación efectiva.

Debido a lo anterior, se propuso la integración dentro del proyecto de un motor gráfico que optimice y facilite el desarrollo de las personas digitales. La elección fue ThreeJS, una librería para javascript que permite la renderización de gráficos tridimensionales en el navegador. El proceso consistió en implementar en el proyecto web los componentes y servicios necesarios para que una persona digital hiciera, como mínimo, lo mismo que hace en Unity, pero ahorrando los pormenores mencionados anteriormente.

Se implementó todo el flujo de creación y carga de avatares 3D a través de código, ya que la librería no cuenta con una interfaz gráfica. Procesos como añadir un modelo 3D, organizar materiales y texturas, gestionar escenas y renderizados se convierten en algoritmos en lenguaje

TypeScript. En ese sentido, el primer paso fue profundizar en los métodos y utilidades principales de Threejs, para lo cual fue útil realizar algunas demos de carga de modelos 3D animados y renderización de escenas.

El proceso de carga de avatares fue uno de los que más cambió. Los modelos se trataban en formato FBX, el cual tiene una alta compatibilidad con Unity, pero no tanto con ThreeJS, en donde se suelen usar otros formatos como GLB o GLTF. Esto implicó la definición de un nuevo proceso de exportación desde las herramientas de edición 3D para adaptarse al requerimiento. Además, el algoritmo de carga del avatar implica también configurar las mallas, materiales, texturas y animaciones que contiene el modelo, sin embargo, pese a que de todo lo anterior hay bastante, se logró el objetivo, contando con la ventaja de que al ser hecho a través de código, no es necesaria la configuración manual de todo esto, cosa que sí se debía hacer en Unity.

Adicional a lo anterior, se buscó que el motor también contara con la flexibilidad de importar las personas digitales desde la nube de Nati, esto quiere decir que dentro del proyecto web sólo se implementó el código que estructura y da comportamiento al modelo 3D, esto sin usar el a veces tedioso sistema de Addressables de Unity, que está más pensado para usarse dentro de su misma nube y no tanto en almacenamientos externos.

En cuanto a rendimiento, se alcanzó el resultado esperado, logrando una reducción en los tiempos de carga del avatar de alrededor de 10 segundos, gracias, especialmente, a que ya no hubo necesidad de cargar la instancia de Unity dentro de la aplicación. Es un logro importante tanto a nivel técnico como de experiencia de usuario, pues sin duda los clientes pueden interactuar más rápida e intuitivamente con la aplicación. Por último, vale recalcar que en ThreeJS no hay ningún tipo de logo o marca de agua que indique su uso, al contrario de Unity, que siempre muestra su logo al iniciar la aplicación y que retrasaba unos segundos más la interacción.

Un proceso crucial en Unity era la reducción del tamaño de algunas texturas para lograr que las personas digitales pesaran lo mínimo posible en beneficio de la velocidad de carga de la aplicación, que ThreeJS optimice de por sí este proceso, posibilitó la mejoría en la calidad gráfica de las texturas al conservar un tamaño más grande, logrando que el avatar se vea mucho mejor

que en Unity. En la figura 3 se puede hacer la comparación entre ambos resultados, en donde se puede apreciar la mayor resolución en partes como el pantalón y las manos.



Fig. 3 Avatar Nati en Unity a la izquierda, avatar de ThreeJS a la derecha.

Por último, el resto de funcionalidades que se encontraban en Unity fueron trasladadas a su versión en ThreeJS, incluyendo el sistema de animaciones, de lipsync, de control de gestos y de movimientos. En ocasiones el código resultante fue bastante similar, mientras que en otros apartados requirió aproximaciones diferentes, por ejemplo, el sistema de animaciones se tuvo que implementar sin ayudas de interfaces de usuario, logrando como resultado la misma calidad obtenida en Unity en cuanto a las animaciones y transiciones.

En definitiva, el soporte de personas digitales optimizadas para la web con ThreeJS fue una adición valiosa para la variedad de opciones que ofrece la plataforma Nati, brindando al cliente una versión liviana para casos de uso en donde la velocidad y eficiencia del asistente digital es crucial. El impacto fue tal, que actualmente es la opción prioritaria a la hora de agregar

nuevas personas digitales y además, la mayoría de clientes están usando esta versión dada la mejora en la experiencia de uso de Nati.

C. Vigilancia tecnológica en generación de avatares con IA.

Dentro de los intereses de Innovati con su plataforma Nati está la incorporación de tecnologías de vanguardia en temas relacionados a la inteligencia artificial que se puedan aplicar con un criterio claro para la generación de valor a los clientes. Dentro de las tendencias en personas digitales se encuentra la generación de avatares fotorrealistas que se comunican en tiempo real con los usuarios, en una experiencia que puede resultar más inmersiva y comunicativa y por tanto, de utilidad para la plataforma en su propuesta de negocio.

En este contexto, una parte de la práctica se enfocó en la investigación del estado del arte en estos tópicos en cuanto a modelos, arquitecturas y técnicas con el fin de perfilar las opciones más adecuadas de acuerdo a las necesidades de Nati. haciendo énfasis en los siguientes apartados:

- Calidad del vídeo, teniendo en cuenta tanto su peso como su tamaño en ancho y alto.
- Calidad del movimiento de labios y su sincronización con el audio.
- Calidad en los movimientos de la cabeza, buscando la mayor naturalidad posible.
- Tiempo de generación, pensando en la necesidad de una conversación lo más cercana posible al tiempo real.
- Licencia de uso MIT o similares.
- Requisitos y tiempo de entrenamiento del modelo para avatares nuevos.

A partir de estos lineamientos, se hizo un trabajo en equipo con estudiantes de la materia Proyecto Integrador I del pregrado de Ingeniería de Sistemas de la Universidad de Antioquia, quienes estuvieron interesados en desarrollar su proyecto en generación de avatares fotorrealistas con inteligencia artificial en apoyo a Innovati. A través de reuniones periódicas se pudieron presentar avances no sólo en materia de investigación sino también de implementación y pruebas de distintos modelos.

Como resultado, se eligió un modelo que reúne muchos de los requisitos previamente listados, con este se construyó una aplicación backend de generación de vídeo en tiempo real, la cual consumía una aplicación frontend en donde se podían ver los resultados. Aunque fuera sólo

una demo, el usuario podía escribir lo que deseaba dijera el avatar y este lo vocalizaba con el vídeo creado, demostrando entonces el potencial de la tecnología para futuros desarrollos en la plataforma Nati.

VIII. CONCLUSIONES

Se puede decir con seguridad que los objetivos de las prácticas fueron alcanzados exitosamente, y además, dieron un plus valioso al estado actual de la plataforma Nati, tanto así que las líneas abiertas en la práctica (ThreeJS e IA generativa) siguen siendo tópicos en constante evolución.

Se logró una transferencia de conocimiento adecuada de los proyectos involucrados, comprendiendo a cabalidad las partes fundamentales tanto del proyecto de Unity como del frontend 3D, lo que además permitió los desarrollos en ThreeJS de acuerdo a los objetivos establecidos por la compañía.

En Unity se hicieron mejoras en la automatización del proceso de integración de avatares, reduciendo los tiempos y pasos necesarios para terminarlo, además de su complejidad. Como resultado, el proyecto se encuentra en un estado estable en donde son pocas las modificaciones que se hacen.

La integración de la librería 3D ThreeJS a la aplicación web, permitió no sólo replicar y, en muchos aspectos, mejorar las funcionalidades de Unity, sino también una naturalidad en la implementación y la comunicación con los servicios y componentes de Angular que mejoró significativamente la complejidad del código. Por otro lado, aspectos críticos de Nati, como lo son la velocidad de carga y la calidad de las personas digitales, se vieron beneficiados por las ventajas de poseer un motor nativo para la web.

Por otro lado, y gracias al conocimiento adquirido a lo largo de la práctica académica en desarrollo frontend con Angular y en entendimiento de requisitos establecidos por diferentes clientes, se pudieron hacer contribuciones significativas en términos de funcionalidad, experiencia de usuario y calidad del software que implican una mejoría del estado de Nati respecto a cómo estaba al momento de tomar el control del proyecto web.

La vigilancia tecnológica se volvió un pilar fundamental en los objetivos planteados en inteligencia artificial generativa para avatares fotorrealistas. Durante el proceso se aprendieron procedimientos adecuados en la investigación del estado del arte en campos tecnológicos,

teniendo en cuenta las búsquedas en bases de datos bibliográficas y dando un énfasis especial en la disponibilidad de los recursos de código expuestos en los repositorios de las investigaciones para validar los hallazgos y comparar resultados. Esto último fue importante para avanzar en la visión de Nati, ya que si en algo se caracteriza Innovati es en la orientación al resultado, es decir, la investigación como medio para generar propuestas concretas iterativamente, validadas a través de prototipos que permitan tomar decisiones adecuadas en aras de aumentar el valor de la plataforma para sus usuarios.

Adicionalmente, llevar a cabo la práctica académica en una StartUp como Innovati, permitió el acercamiento a muchas tecnologías, conceptos y técnicas de las cuales no se tenía previo conocimiento, esto gracias a la flexibilidad que se da a los desarrolladores para que se proyecten profesionalmente hacia temas que le interesen y que se trabajen dentro de la empresa. Esto fue fundamental para descubrir y explorar nuevas alternativas profesionales que, sin duda, serán valiosas en el futuro.

Por otra parte, el trabajo en equipo fue determinante para llevar a cabo los diferentes objetivos planteados. Fue común trabajar semana a semana con uno o más miembros del equipo enfocados en otras partes del conjunto de aplicaciones de Nati, para la implementación de distintas funcionalidades. Lo anterior, potenciado por el ambiente de trabajo profesional, pero también amistoso que se encuentra dentro de la organización.

En términos metodológicos, el marco SCRUM, con las adaptaciones propias de Innovati, se asemeja en gran medida a los aprendizajes que se adquieren en muchas de las materias de la carrera en ese sentido. Además, proyectos como Fábrica Escuela que facilitan adquirir e interiorizar algunos de los roles típicos del marco, permitieron sentir una familiaridad con los eventos SCRUM que fueron llevados a cabo a lo largo de la práctica.

Por último, cabe recalcar la relevancia de ser miembro del semillero de videojuegos de la Universidad de Antioquia. Las bases conceptuales obtenidas durante los talleres y proyectos en los que se tuvo la oportunidad de participar, permitieron el entendimiento, el apropiamiento y el desarrollo de los proyectos de Unity, ThreeJS y generación de avatares. Haciendo énfasis en ThreeJS, en donde fue de gran impacto el conocimiento en motores de videojuegos, pues no fue

necesario un extenso estudio de la librería para entender sus bases, puesto que, esencialmente, son las mismas que Unity.

REFERENCIAS

- [1] Nati.ai, 2022. [en línea]. Disponible en: <https://nati.ai/>
- [2] IBM, ¿Qué es el procesamiento del lenguaje natural (PLN)? 2021. [en línea]. Disponible en: <https://www.ibm.com/es-es/topics/natural-language-processing>
- [3] Three.js, three.js manual. 2024. [en línea]. Disponible en: <https://threejs.org/manual/#en/fundamentals>
- [4] AWS, What is an API? - application programming interface explained. 2024. [en línea]. Disponible en: <https://aws.amazon.com/what-is/api/>
- [5] Stryker, C. y Scapicchio, M., ¿Qué es la ia generativa?. 2024. [en línea]. Disponible en: <https://www.ibm.com/es-es/topics/generative-ai>
- [6] Ye, Zhenhui, et al. "Geneface++: Generalized and stable real-time audio-driven 3d talking face generation." *arXiv preprint arXiv:2305.00787*, 2023. [en línea]. Disponible en: <https://arxiv.org/abs/2305.00787>
- [7] Unity Technologies, Unity - Unity. 2024. [en línea]. Disponible en: <https://unity.com/es>
- [8] Khronos Group, WebGL. 2011. [en línea]. Disponible en: <https://www.khronos.org/api/webgl>
- [9] Geetha, G. et al. "Interpretation and analysis of Angular Framework", *International Conference on Power, Energy, Control and Transmission Systems (ICPECTS)*, 2022, pp 1-6. doi:10.1109/ICPECTS56089.2022.10047474.