REVECOM

REVISTA VENEZOLANA DE COMPUTACIÓN

ML • Java • Python • C# • C++ • Scala
Perl • Haskell • Lisp • JavaScript

Sociedad Venezolana de Computación

Vol. 8, No. 1
Junio 2021

ISSN: 2244-7040

SVC VENEZUELA Sociedad Venezolana de Computación

Revista Venezolana de Computación

REVECOM

Revista Venezolana de Computación

Sociedad Venezolana de Computación

Editores:
Eric Gamess, Wilmer Pereira, Yudith Cardinale

ISSN: 2244-7040

Vol. 8, No. 1
Junio 2021

Editorial

Bases de Datos Difusas

Hace más de cinco décadas que Lofti Zadeh propuso la Teoría de Conjuntos Difusos, la cual permite dar un tratamiento matemático y computacional a conceptos vagos del mundo real (L. A. Zadeh, "*Fuzzy Sets*", *Information and Control*, 8(3) 338–353, 1965). Desde entonces, los conjuntos difusos han hallado cabida en distintos tipos de aplicaciones en diversas áreas de ciencias y tecnologías de la computación y la automatización. Las "Bases de Datos" es una de esas áreas en que se ha explorado la aplicabilidad de los Conjuntos Difusos.

Los conjuntos clásicos se pueden definir mediante una función característica cuyo rango es {0,1}. Esta función simplemente asigna el valor 1 (verdadero) a los elementos que forman el conjunto, mientras que asigna el valor 0 (falso) a los elementos que están fuera del conjunto. Los conjuntos difusos generalizan esta concepción, permitiendo elementos gradualmente incluidos. Un conjunto difuso se define mediante una función de membresía cuyo rango es el intervalo real [0,1]. Esta función simplemente asigna el valor 1 a los elementos que están completamente incluidos en el conjunto, 0 a los elementos completamente excluidos, y un valor entre 0 y 1 a los elementos que están gradualmente incluidos. Cuanto más cerca de 1 esté el grado, indica que el elemento es más compatible con el concepto expresado por el conjunto.

El concepto de base de datos pudiera ser algo muy complejo. Existen diversos modelos de bases de datos en tres niveles: conceptual, lógico y físico. Sin embargo, es posible hacer una abstracción y simplificarlo en un nivel lógico. En esta simplificación, podemos pensar en una base de datos como una colección de conceptos, instanciados en conjuntos de registros, conformados por atributos de datos; la cual es persistente, puede ser modificada y consultada. Vamos a tomar esta concepción para exponer la aplicabilidad de la Teoría de Conjuntos Difusos a las Bases de Datos.

En una base de datos tradicional los conjuntos de registros que instancian cada concepto son conjuntos clásicos. Un registro de datos que está en el conjunto se considera que es una instancia del concepto. De manera que el concepto representado en la base de datos es algo rígido. Esto es adecuado para muchas situaciones del mundo real. Por ejemplo, si pensamos en los empleados de una organización, el conjunto de instancias es rígido. Pero, si queremos el conjunto de los buenos empleados, este conjunto no debería ser rígido. De modo que deberíamos tener bases de datos donde algunos conceptos permitan como instancia un conjunto difuso de registros.

En el mundo real pueden existir datos de los cuales no conocemos con certeza su valor, pero sí tenemos una idea de cuál puede ser. Estos datos imprecisos no son contemplados en las bases de datos tradicionales. Ellos se pueden representar mediante un tipo de restricción que se llama distribución de posibilidad, que consiste en el conjunto difuso de los posibles valores del dato impreciso. Por ejemplo, puede que de algunas de mis amistades desconozca su edad exacta, pero tengo una idea al menos del rango aproximado. Esto se pudiera almacenar en una base de datos que permita atributos con valores imprecisos. Aquí vemos otra aplicabilidad de los conjuntos difusos en las bases de datos.

Editorial

El lenguaje natural es muy rico en la expresión de condiciones lógicas. Estas expresiones se formalizan mediante la lógica simbólica. Particularmente la lógica de primer orden o lógica de predicados, que se basa en la teoría de conjuntos, con su conocida consecuencia de rigidez. Por ejemplo, el decir que dos objetos están cerca, en el mundo real, puede ser algo gradual y relativo, hay objetos que están más cerca que otros. Por ello, resulta conveniente la extensión de la lógica de predicados con la teoría de conjuntos difusos, dando lugar a la lógica difusa.

En la lógica difusa, un predicado se interpreta mediante un conjunto difuso. Así el valor de verdad de este predicado ya no es verdadero o falso, sino que es gradual en el intervalo real [0,1]. El grado 0 se interpreta como falso, mientras que el 1 es verdadero, y los grados intermedios son parcialmente ciertos, mientras más cerca esté de 1, más posiblemente cierto. Se da cabida a la representación de adverbios que modifican el sentido de predicados. Por ejemplo, la expresión “muy cerca” sería un predicado formado por el adverbio “muy” y el predicado base “cerca”. Estos adverbios se interpretan como operadores en [0,1]. Los conectores lógicos negación, conjunción, disyunción e implicación son interpretados mediante operadores en [0,1] que representan el sentido del conector. De hecho, existen diversas interpretaciones posibles para estos conectores.

También los cuantificadores lógicos se extienden en lógica difusa. Se habla de cuantificadores difusos absolutos, entre los cuales está el existencial, así como “cerca de diez”, “al menos cinco”, “a lo sumo tres”, entre otros. También hay cuantificadores difusos relativos, entre los cuales está el universal, pero también expresiones como “la mayoría”, “la minoría”, “la mitad”, entre otros. Las expresiones cuantificadas difusas se interpretan mediante el uso de integrales difusas que permiten combinar el conjunto difuso que representa el cuantificador, con el conjunto difuso que representa el predicado bajo la cuantificación y el conjunto difuso base sobre el cual se hace la cuantificación.

Las consultas a bases de datos tradicionales se basan en expresiones en lógica de predicados aplicadas en distintas estructuras de consultas. En las bases de datos difusas, estas expresiones se extienden con el uso de la lógica difusa. Entre las operaciones de consulta que se tienen usualmente en las bases de datos, están las operaciones de ordenamiento y de particionamiento. El ordenamiento se basa en relaciones de orden total en los dominios de los atributos. Al tener atributos de datos imprecisos, se hace necesario definir relaciones de orden sobre estos dominios, para que puedan ser usados en estas consultas. Por otro lado, el particionamiento se basa en relaciones de equivalencia definidas a partir de la igualdad de valores. Aplicando conjuntos difusos, la relación de igualdad, y las relaciones de equivalencia, pueden ser flexibilizadas a relaciones difusas que se llaman relaciones de similitud.

Durante las últimas dos décadas y medias, me dediqué a la investigación y desarrollo en bases de datos difusas. Es mucho el trabajo realizado y la satisfacción recibida, así como numerosos los aportes de diversos colegas por el mundo. “Pero todo esto, que antes valía mucho para mí, ahora, a causa de Cristo, lo tengo por algo sin valor. Aún más, a nada le concedo valor si lo comparo con el bien supremo de conocer a Cristo Jesús, mi Señor. Por causa de Cristo lo he perdido todo, y todo lo considero basura a cambio de ganarlo a él” (Filipenses 3:7-8, DHH)

Dr. Leonid Tineo
Profesor Jubilado de la Universidad Simón Bolívar

Revista Venezolana de Computación

ReVeCom (Revista Venezolana de Computación) es la primera revista venezolana arbitrada, periódica, digital, orienta a la publicación de resultados de investigación en el campo de la computación. ReVeCom fue creada por la SVC (Sociedad Venezolana de Computación) y tiene entre sus objetivos hacer conocer los trabajos de alta calidad investigativa que se realizan a nivel nacional, latinoamericano e internacional. La revista permite la divulgación de artículos con aporte original en castellano o inglés.

ReVeCom es una revista abierta para una mayor difusión de los resultados de investigación. Cuenta con una página web (https://svc.net.ve/revecom-informacion), donde se encuentran los trabajos publicados e información sobre la revista. La revista promueve la pluralidad de intereses, dando cabida a la divulgación de trabajos de todos los campos del conocimiento inherentes a la computación.

Este volumen de ReVeCom (Vol. 8, No. 1) corresponde a artículos de investigación en el campo de la computación que fueron seleccionados a través de un riguroso arbitraje por expertos del área. En esta oportunidad, el comité evaluador fue compuesto por:

Nombre	Afiliación
Rhadamés Carmona	Universidad Central de Venezuela – Venezuela
Ernesto Coto	Optellum Ltd. – Reino Unido
Jacinto Dávila	Universidad de Los Andes – Venezuela
Irvin Dongo	Universidad Católica San Pablo – Perú
Eric Gamess	Jacksonville State University – USA
Jonas Montilva	Universidad de Los Andes – Venezuela
Carlos Moreno	Universidad Central de Venezuela – Venezuela
Dinarle Ortega	Universidad de Carabobo – Venezuela
David Padua	University of Illinois – USA
Wilmer Pereira	Instituto Tecnológico Autónomo de México – México
Robinson Rivas	Universidad Central de Venezuela – Venezuela
Dulce Rivero	Universidad de Los Andes – Venezuela
Eugenio Scalise	Universidad Central de Venezuela – Venezuela

Directorio de la Sociedad Venezolana de Computación

Presidente:
Dr. Leonid Tineo (Universidad Simón Bolívar – Venezuela)

Vicepresidente:
Dr. Eric Gamess (Jacksonville State University – USA)

Secretario:
Dr. Wilmer Pereira (Instituto Tecnológico Autónomo de México – México)

Tesorero:
Dr. David Coronado (Universidad Simón Bolívar – Venezuela)

Coordinadora de Educación e Investigación:
MSc. Mildred Luces (Universidad Nacional Experimental de la Gran Caracas – Venezuela)

Coordinadora de Publicaciones:
Dra. Yudith Cardinale (Universidad Simón Bolívar – Venezuela)

Coordinadora de Eventos:
MSc. Soraya Carrasquel (Universidad Simón Bolívar – Venezuela)

Edición

Comité Editorial

Director:
Dr. Eric Gamess - Jacksonville State University, USA
Redes de computadores, computación de alto desempeño, simulación.

Coordinador del Comité Editorial:
Dr. Wilmer Pereira - Instituto Tecnológico Autónomo de México, México
Inteligencia artificial, robótica autónoma, aprendizaje automatizado.

Jefe de Redacción:
Dra. Yudith Cardinale - Universidad Simón Bolívar, Venezuela
Computación paralela, computación de alto desempeño, sistemas distribuidos, computación en la nube, arquitecturas paralelas, servicios web, web semántica.

Miembros del Comité Editorial

Dr. Carlos Acosta - Universidad Central de Venezuela, Venezuela
Computación paralela, computación de alto desempeño, computación reconfigurable y FPGAs, simulación paralela y distribuida, BigData.

Dr. Andrés Arcia-Moret - Xilinx Cambridge, Reino Unido
Simulación de redes, protocolos de transporte, redes inalámbricas.

Dr. Ernesto Coto - Optellum Ltd., Reino Unido
Computación gráfica, visualización científica, procesamiento digital de imágenes.

Dr. Francisco Luengo - Universidad del Zulia, Venezuela
Computación social, minería de texto.

Dr. Jonas Montilva - Universidad de Los Andes, Venezuela
Ingeniería del software, sistemas de información.

Dra. Masun Nabhan - Universidad Simón Bolívar, Venezuela
Inteligencia artificial, minería en datos, aplicaciones de inteligencia artificial para educación y discapacitados.

Dra. Dinarle Ortega - Universidad de Carabobo, Venezuela
Ingeniería del software, arquitectura del software, arquitecturas empresariales, modelado de procesos de negocio.

Dr. David Padua - University of Illinois, USA
Compiladores, computación de alto desempeño.

Dr. Leonid Tineo - Universidad Simón Bolívar, Venezuela
Bases de datos, lógica difusa, lenguajes artificiales, minería de datos.

Tabla de Contenido

	Editorial	ii
	Revista Venezolana de Computación	iv
	Directorio de la Sociedad Venezolana de Computación	v
	Comité Editorial	vi
	Tabla de Contenido	vii
1.	**Dispositivo de Evaluación de Poses Integrado a una Arquitectura Terapéutica IoT** Francisco Moreno, Walter Hernández	1-12
2.	**Un Experimento Curricular en Ciberética y Sociedad: Creación de un Curso de Innovación Responsable en Tiempos de Pandemia** Alejandro Teruel	13-24
3.	**Extracción de Esqueleto por Contracción de Mallado con Asignación Automática de Pesos** Moisés Berenguer, Francisco Moreno	25-36
4.	**Despliegue de un Clúster HPC Utilizando Contenedores Docker** José Hernández, Gianfranco Verrocchi, Yudith Cardinale, Ana Aguilera, Irvin Dongo	37-46
	Índice de Autores	47

Revista Venezolana de Computación
ISSN: 2244-7040
http://www.svc.net.ve/revecom
Vol. 8, No. 1, pp. 1-12, Junio 2021
Fecha recepción: 18/02/2021, Fecha aceptación: 14/06/2021

Dispositivo de Evaluación de Poses Integrado a una Arquitectura Terapéutica IoT

Francisco José Moreno Alvarez, Walter Hernández
francisco.j.moreno@ucv.ve, walter.hernandez@ciens.ucv.ve

Centro de Computación Gráfica, Universidad Central de Venezuela, Facultad de Ciencias, Caracas, Venezuela

Resumen: En el presente trabajo se describe el proceso de desarrollo e integración de un dispositivo inteligente a una arquitectura basada en *IoT* orientada a la terapia física. El dispositivo inteligente corresponde con un sistema que monitorea la postura del paciente mientras este realiza un conjunto de posturas propuestas por un especialista médico. El dispositivo procura ser de bajo costo monetario y no intrusivo, permitiéndole al paciente tener una libre movilidad durante el ejercicio. Para su desarrollo se proponen un conjunto de alternativas acerca del módulo encargado de la adquisición de las características del paciente (esqueleto). Como resultado se obtiene un sistema virtualmente agnóstico al *hardware* utilizado para la captura del esqueleto del paciente. De igual manera se proponen distintos mecanismos para comparar la postura del paciente contra la esperada, procurando realizar una detección eficaz y proveer de herramientas que le permitan tanto al paciente como al personal médico evaluar su rendimiento. Por otra parte, se indican las consideraciones tomadas al momento del desarrollo del dispositivo para poder integrarlo a la arquitectura IoT previamente implementada. Se describe el proceso para registrar los parámetros requeridos para configurar el dispositivo desde una herramienta *web*, el proceso para asignar un ejercicio a un paciente a través del dispositivo desarrollado, así como las consideraciones tomadas para evaluar los resultados provistos por el mismo. Finalmente, se comprueba la eficacia del método de comparación de poses, demostrando ser altamente efectivo independientemente de la posición y estatura del usuario. De igual manera, se compara la eficiencia de los mecanismos de adquisición planteados, proporcionando una alternativa capaz de operar inclusive en dispositivos móviles.

Palabras Clave: Poses; IoT; Terapia; Dispositivo Inteligente; Internet de las Cosas; Rehabilitación.

Abstract: In this work the development and integration of a smart device to a physical therapy *IoT* based architecture is presented. The smart device, supervises the patient while performing a set of postures configured by a medical specialist. The proposed device is designed to have a low monetary cost and to be unobtrusive throughout the therapy's execution, allowing the patient to move freely and with no physical restrictions. Several alternatives are studied to handle the extraction of the user posture defining characteristics, such as the skeleton. As a result, the proposed device is considered virtually agnostic to the *hardware* used to capture the user's skeleton. Furthermore, several techniques to compare the user's posture with the target pose are studied, pursuing an effective method which provides the tools required from the patient and therapist to qualify the therapy's execution performance. On the other hand, a set of considerations are taken into account to allow an easy integration of the device with an existing implementation of an *IoT* architecture. The process to register all the required device parameters through a web platform, including the steps to assign an exercise based on the smart device and all the requirements to evaluate the results of each exercise are also described. Finally, the effectiveness of the proposed technique to compare the poses is tested, proving to be highly effective independently of the location and height of the patient. Likewise, the efficiency of the skeleton acquisition techniques are demonstrated, providing an alternative capable of working even on mobile devices.

Keywords: Poses; IoT; Therapy; Smart Device; Internet of Things; Rehabilitation.

I. Introducción

Se estima que globalmente alrededor de 2.4 billones de personas conviven con algún tipo de afectación de salud que podría mejorarse directamente a través de la rehabilitación. Sin embargo, actualmente estas necesidades están siendo cubiertas en un porcentaje muy bajo. En países de mediano a bajo ingreso por ejemplo, más de la mitad de la población que requiere este tipo de servicio no lo recibe. Por otra parte los servicios de rehabilitación han detenido abruptamente su funcionamiento en aproximadamente un 60-70% de los casos, debido a las condiciones generadas por eventos como la pandemia por COVID-19 desde principios del año 2020 [1].

Así mismo, múltiples investigaciones [2][3] confirman la efectividad del ejercicio físico como una herramienta para la recuperación y prevención de ataques cerebro vasculares, e indican su capacidad para actuar como un agente de retardo de las enfermedades asociadas al envejecimiento. Por ende, es necesario que la población mantenga un estilo de vida activo y realice ejercicio. Dichos ejercicios deben ser diseñados y evaluados por personal especializado en las condiciones particulares del paciente. Dados los problemas de logística, infraestructura y capacidad monetaria que involucran prestar y recibir sesiones de ejercicio o terapia de manera presencial, se propuso una arquitectura basada en *IoT* [4] que permita al paciente realizar sesiones de terapia de manera remota y, al personal médico evaluar su progreso. Dicha arquitectura pretende aliviar la precariedad del sistema de salud, que se agrava aún más por situaciones extraordinarias como la mencionada pandemia del COVID-19.

En el trabajo actual se presenta un dispositivo terapéutico que forma parte de la arquitectura basada en *IoT* y que tiene como objetivo fortalecer la flexibilidad y movilidad de los pacientes que lo utilicen. Dicho dispositivo solo requiere el uso de un computador tradicional y una cámara *web* (en esta primera fase), haciéndolo de fácil acceso y sin requerir un gasto económico adicional para el paciente o el centro hospitalario que lo provea.

El resto del trabajo se presenta en 4 secciones. La Sección II describe brevemente algunos trabajos relacionados a la propuesta que se presenta en la Sección III. La Sección IV plantea las pruebas que se realizaron para evaluar su capacidad y efectividad de evaluar la calidad del ejercicio realizado por el paciente. Finalmente, la Sección V expone las conclusiones del trabajo y trabajos futuros.

II. Trabajos Relacionados

Ahmad et al. [5] presentan un robot de asistencia social que motiva a personas de la tercera edad a realizar ejercicios de movilidad recomendados por el servicio nacional de salud de Inglaterra [6]. Mientras la persona realiza los ejercicios que el robot le propone, este evalúa que se realicen de la manera correcta y proporciona instrucciones y palabras de aliento al paciente para que complete el ejercicio.

Los movimientos del paciente son capturados mediante el uso de un *Microsoft Kinect* que genera la representación de su esqueleto. La postura registrada se evalúa mediante un modelo de cinemática angular que compara los ángulos de los huesos del paciente con un conjunto de parámetros preestablecidos. El modelo se complementa con zonas espaciales donde deben encontrarse las articulaciones del esqueleto registrado del paciente.

El sistema es probado con un conjunto de pacientes de la tercera edad, que manifestaron su satisfacción de contar con un asistente personal (el robot de asistencia social) que les proponga ejercicios y los motive durante su ejecución. En la Figura 1, se puede observar a un paciente siendo asistido por el robot propuesto en la investigación.

Figura 1: Paciente Utilizando el Sistema Propuesto por [5]

Por otra parte, Ian et al. [7] indican la importancia de fortalecer y mantener el tono muscular del cuerpo para la salud humana. Hacen énfasis en el Yoga como una herramienta para fortalecer la musculatura y mejorar la flexibilidad del prácticamente. Sin embargo, mencionan que el proceso de aprendizaje del Yoga tiende a ser entorpecido por los errores de juicio de los distintos instructores de Yoga. Por ende, proponen un sistema para aprender a realizar posturas de Yoga mediante el uso del computador.

En este sistema, el aprendiz es grabado por una cámara mientras realiza una postura siguiendo un conjunto de instrucciones. Las imágenes capturadas, se envían a un computador que extrae el esqueleto del usuario mediante una red neuronal denominada *OpenPose*[1]. Posteriormente, evalúa los ángulos de los huesos del estudiante mediante un modelo angular similar a [5]. En caso de que el estudiante realice correctamente la postura, su esqueleto es desplegado en color verde y en caso contrario en color rojo. El sistema se complementa utilizando una tabla (*hardware especializado*) que evalúa el balance del paciente. En la Figura 2 se puede observar a un practicante haciendo uso del sistema.

Finalmente Santosh et al. [8], proponen un sistema para aprender y reconocer posturas de Yoga (*asanas*) exclusivamente mediante el uso de técnicas basadas en aprendizaje profundo. En este caso, utilizan *OpenPose* al igual que [7] para identificar los puntos que correspondan con las articulaciones del usuario en una imagen. La red previa alimenta a una red neural recurrente con memoria a corto y largo plazo, que relaciona las posturas con información temporal para aprender la secuencia de movimientos que conlleva realizar correctamente una *asana*.

Los datos de entrenamiento para las redes, fueron creados a partir de vídeos de 15 individuos realizando *asanas*, y se probó con 12 individuos distintos a los de la fase de entrenamiento. Como resultado, se obtuvo un 90% de exactitud en la pruebas realizadas. En la Figura 3 se puede observar a un usuario haciendo uso del sistema propuesto.
En el sistema desarrollado en esta investigación se propone una alternativa al método de comparación angular utilizado

[1] https://github.com/CMU-Perceptual-Computing-Lab/openpose

Figura 2: Usuario Utilizando el Sistema Propuesto por [7]

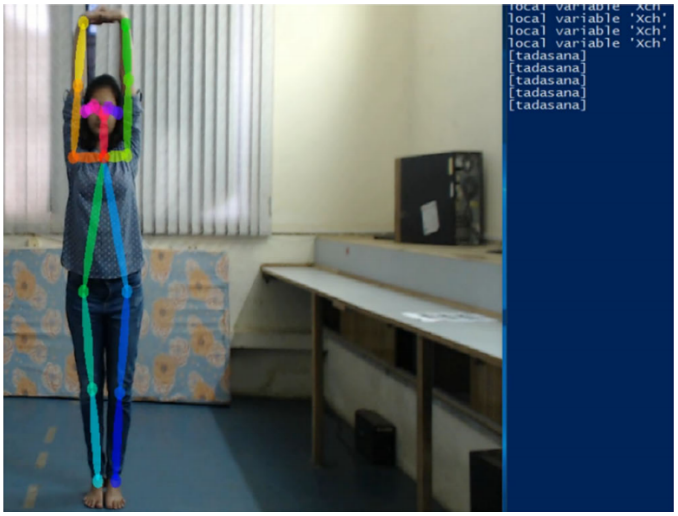

Figura 3: Usuario Utilizando el Sistema Propuesto por [8]

en [5] y [7], capaz de discernir entre poses distintas cuyos ángulos internos son los mismos, sin la necesidad de requerir información adicional. Se experimenta con distintos dispositivos de adquisición del esqueleto y se compara su eficiencia. Finalmente, se acopla la propuesta a la red de dispositivos terapéuticos que forman parte de la arquitectura basada en *IoT* producto de una investigación previa [4].

III. Propuesta

El dispositivo que se propone a continuación, tiene como objetivo evaluar el rango de movilidad del paciente mientras imita una postura propuesta por el personal médico. Para alcanzar la meta es necesario definir:

- Un mecanismo para capturar la postura actual del paciente.
- Un algoritmo que permita evaluar la similitud entre la postura del paciente y la propuesta por el personal médico.
- Herramientas que informen al paciente el grado de similitud contra la pose objetivo, para facilitar la tarea de imitarla.
- Elementos que le permitan al médico visualizar y diseñar la postura para el paciente y a este último, tener un punto de referencia de la postura a seguir.
- Finalmente, una aplicación que le permita al médico configurar la pose y su homólogo que evalúe la postura del paciente.

Es importante destacar la necesidad de desarrollar un dispositivo de fácil acceso, por ende, debe ser de bajo costo monetario y en la medida de lo posible no depender de *hardware* especializado que involucre un gasto adicional para el paciente o el centro hospitalario que lo provea.

A. Adquisición de Postura

Para identificar la postura del paciente es necesario conocer la posición en el espacio de cada una de las articulaciones que componen su esqueleto. Además, ya que la aplicación será utilizada por personas con movilidad reducida, es importante que el mecanismo de adquisición no dificulte la ya mermada movilidad del paciente. Dada estas observaciones, se experimentó con tres alternativas de adquisición no invasivas, que infieren la posición de las articulaciones del paciente a partir del análisis de imágenes.

La primera alternativa fue el *Microsoft Kinect*. Dispositivo especialmente desarrollado para la adquisición del esqueleto de múltiples usuarios dentro de su rango de visión [9]. Fue concebido originalmente para ser utilizado en conjunción con la consola de videojuegos *Xbox 360*, con la intención de desarrollar aplicaciones interactivas. Utiliza imágenes de profundidad (por láser), haciéndolo poco susceptible a las características de iluminación del entorno en el que se encuentre el usuario y, permitiéndole inferir la posición en tres dimensiones de 20 articulaciones sobre el mismo. La clara desventaja de trabajar con esta herramienta, es que el paciente o centro hospitalario debe contar con el *hardware* especializado *Microsoft Kinect*.

OpenPose surge como alternativa al no requerir de un dispositivo especializado de adquisición. Utiliza como insumo imágenes que pueden ser adquiridas a través de cualquier cámara convencional. *OpenPose* es una librería de *software* libre basada en aprendizaje profundo [10][11][12][13], capaz de extraer la posición 143 puntos claves por cada usuario presente en una imagen. Dichos puntos se descomponen en: 25 puntos generales para el cuerpo del usuario, 6 puntos que identifican cada parte del pie, 21 puntos por mano y 71 para la cara. Esta alternativa presenta un claro avance en precisión y cantidad de información registrada contra la previa. Sin embargo, requiere de un computador con alto poder de cómputo para alcanzar velocidades de procesamiento que se consideren interactivas. La librería cuenta con un apartado que permite la aceleración del proceso de clasificación mediante la tarjeta gráfica, pero es dependiente de la arquitectura propietaria *CUDA* del fabricante

de *hardware Nvidia*. Si bien es cierto que el trabajar con imágenes tradicionales presenta una ventaja ante la facilidad de acceso a la tecnología, trae como consecuencia que los puntos claves extraídos por la librería se encuentren ubicados en el espacio en dos dimensiones de la imagen.

Finalmente, *PoseNet* al igual que *OpenPose* es una librería basada en aprendizaje profundo [14][15] propuesta por *Google* y es capaz de extraer la posición 17 puntos claves en el cuerpo de cada usuario presente en una imagen. Para realizar la extracción de los mismos puede hacer uso de un modelo de clasificación diseñado para trabajar en dispositivos móviles y de bajo poder de cómputo [16] o un modelo de clasificación más preciso [17] pero más lento. Adicionalmente, *PoseNet* tiene la capacidad de ejecutarse directamente en un navegador *web*, al ser soportado por la versión *web* de *TensorFlow*[2]. Estas características hacen que sea un fuerte competidor contra las alternativas previamente mencionadas, ya que a pesar de ser menos preciso que *OpenPose*, tiene la capacidad de operar en computadoras convencionales a tasas interactivas y no requerir *hardware* especial para la adquisición de las imágenes a diferencia del *Microsoft Kinect*, por lo que se considera como la alternativa más accesible de las tres.

A pesar de que *PoseNet* se presenta como la mejor alternativa en lo que confiere al tema de accesibilidad, el sistema fue desarrollado para que pudiera operar independientemente del método de adquisición seleccionado. Por ende, se desarrolló un esqueleto estándar que será utilizado en las siguientes etapas del sistema de evaluación. El esqueleto estándar esta compuesto por 14 puntos claves que pueden ser extraídos de los esqueletos que proveen las tres alternativas y, en el caso específico de *PoseNet*, el punto clave correspondiente al inicio del cuello puede ser inferido como el punto medio entre los hombros del paciente. El esqueleto estándar puede ser visualizado en la Figura 4. Los colores utilizados para representar cada hueso fueron elegidos de una paleta de 21 colores de alto contraste, diseñada para facilitar su identificación y memorización por parte del usuario [18].

B. Algoritmo de Evaluación

Posterior a la adquisición de los puntos claves que componen el esqueleto del paciente, la postura que describa debe ser comparada contra la pose propuesta por el personal médico. Dicha comparación debe:

- Ser independiente de la posición del paciente en el espacio.
- Ser invariante ante la posible diferencia en estaturas de los pacientes.
- Permitir algún mecanismo de apoyo visual que guíe al paciente para alcanzar la pose deseada.

Una forma de comparar dos posturas, que es independiente de la posición espacial y es independiente a la estatura del paciente, es mediante el cálculo del ángulo existente entre cada par de huesos adyacentes, tanto en el esqueleto del paciente como en el que representa la pose objetivo. Si los ángulos calculados coinciden, entonces la pose corresponde con la esperada. El ángulo entre cada par de huesos adyacentes es calculado mediante la Ley del Coseno, la cual dicta: que dada la distancia de dos catetos adyacentes en un triángulo y el ángulo formado entre ambos, se puede obtener la distancia del cateto opuesto aplicando la ecuación (1):

Figura 4: Esqueleto Estandarizado

$$c^2 = a^2 + b^2 - 2ab\cos\alpha \tag{1}$$

Donde c representa el cateto opuesto, a y b los catetos adyacentes y α el ángulo formado entre ambos. Dicha fórmula es una generalización del teorema de Pitágoras y puede ser reordenada como (2) para obtener el ángulo que se forma entre los catetos adyacentes.

$$\alpha = \arccos\left(\frac{(a^2 + b^2 - c^2)}{2ab}\right) \tag{2}$$

En el caso de uso actual, cada par de huesos conforman los catetos adyacentes de un triángulo imaginario compuesto por las tres articulaciones que forman los huesos. Entonces, el ángulo entre un par de huesos puede ser calculado con (2). Sin embargo, dos esqueletos con posturas distintas, pueden contener los mismos ángulos entre cada par de huesos, lo que acarraría un falso positivo en la prueba de similitud. Por ende, el cálculo debe ser complementado con la evaluación de la relación espacial existente entre las articulaciones involucradas, tal como lo indica [19]. En la Figura 5, se puede observar como el ángulo del codo en ambas posturas es idéntico, sin embargo en la postura de la izquierda (A) la mano esta por encima del codo, mientras que en la imagen derecha (B) esta por debajo, lo que comprueba la importancia de evaluar las relaciones espaciales entre las articulaciones así como los ángulos.

[2]https://github.com/tensorflow/tfjs-models/tree/master/posenet

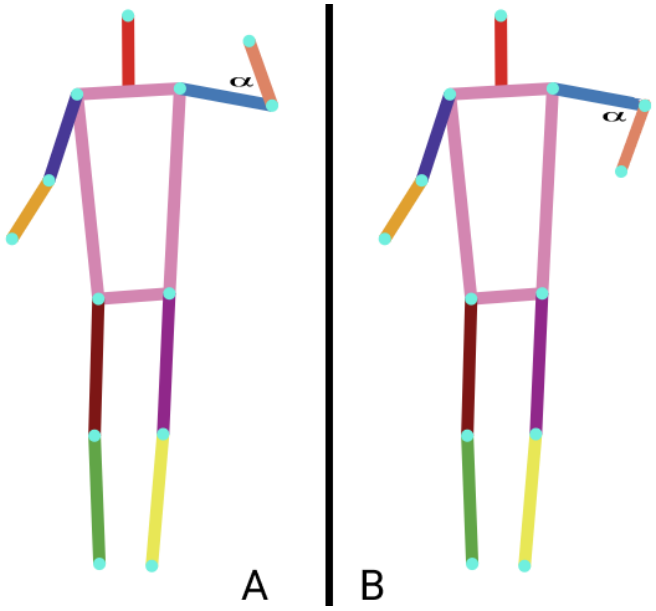

Figura 5: Dos Esqueletos con los Mismos Ángulos y Distintas Poses

El método que se propone en esta investigación es una modificación del discutido previamente. Dicho método evalúa la relación espacial entre cada articulación que compone el hueso y los ángulos entre cada uno, con un simple cálculo y un poco de preprocesamiento. El método se puede descomponer en los siguientes pasos:

1) Transformar la representación del esqueleto basada en puntos en el espacio, a una representación vectorial.
2) Calcular el ángulo existente entre los vectores que representen huesos homólogos en el esqueleto del paciente y el esqueleto objetivo.
3) Verificar que cada ángulo se encuentre dentro de un margen de tolerancia.

1) Representación Vectorial: En este primer paso, se toman las articulaciones de cada hueso del paciente y se genera el vector que compone cada uno de ellos. Es importante que el punto inicial y final de cada hueso en ambos esqueletos sea el mismo, debido a que dicha relación establece la dirección del vector. Esto quiere decir, que si se definió el vector que compone el antebrazo, como el vector que va del codo a la mano, en ambos esqueletos dicho hueso se debe calcular del codo a la mano y no en sentido arbitrario.

Cada vector calculado luego es normalizado. Al normalizarlo, se vuelve independiente de su posición inicial en el espacio y solo se conserva su orientación. La dirección del vector, encapsula la relación espacial entre cada articulación y permite validar que dos poses son idénticas.

2) Cálculo de Ángulos: En este caso, se calcula el ángulo existente entre la representación vectorial de los huesos homólogos en ambos esqueletos. Por ejemplo, al calcular el ángulo existente entre los vectores que corresponden al antebrazo del paciente y la pose objetivo, se compara si los mismos están orientados de la misma forma (ángulo entre los vectores igual a cero) y por ende corresponden parcialmente a la misma pose.

En este punto es que cobra vital importancia que los vectores que representan el mismo hueso, sean calculados utilizando el mismo orden en las articulaciones que lo comprenden, ya que en caso de estar alternados, correspondería con vectores cuyas direcciones son opuestas y el ángulo entre ellos no sería cero. Esta propiedad, permite discernir entre posturas cuyos ángulos internos son idénticos, pero la relación espacial entre las articulaciones que lo componen difiere.

El producto escalar es utilizado para calcular el ángulo entre ambos vectores. La ecuación (3) representa el producto escalar entre dos vectores, donde en el procedimiento actual no es necesario calcular la magnitud de los vectores ya que fueron normalizados en la etapa previa. En caso de que los vectores estén alineados, el ángulo entre ambos será cero y el producto escalar uno. El producto escalar es más eficiente que el método previo (2) ya que no requiere el cálculo del coseno y provee de la información necesaria para verificar la pose.

$$\begin{aligned} \cos(\alpha) &= \frac{a \cdot b}{|a||b|} \\ a \cdot b &= a_x b_x + a_y b_y \end{aligned} \tag{3}$$

3) Verificación del Margen de Tolerancia: Finalmente, se verifica si el producto escalar entre ambos vectores supera un margen de tolerancia impuesto por el personal médico. Dicho margen específica la diferencia en grados entre la dirección esperada del hueso y la lograda por el paciente. Al reducir el margen de tolerancia, se incrementa la dificultad del ejercicio.

C. Maniquí Virtual

El personal médico dispone de un maniquí virtual para configurar la pose que el paciente deberá realizar en la terapia. El maniquí esta compuesto por todos los huesos de la representación del esqueleto del paciente y el personal médico puede alterar la orientación o ángulo de cada uno de ellos, para generar una postura a seguir. Para facilitar el proceso de configuración, el esqueleto (del maniquí y del paciente) es representado como una estructura jerárquica (árbol). Cuando se aplica una transformación a un hueso, sus hijos se verán afectados por esta. Por ejemplo, si se rota el hueso correspondiente al brazo, el hueso del antebrazo rotará con este. El proceso de heredar las transformaciones del hueso padre es conocido como *Foward Kinematic* y se complementa con un conjunto de límites para los rangos de rotación de cada hueso, que previenen la configuración de poses anatómicamente incorrectas.

Finalmente, una pose inicial puede ser creada haciendo uso del algoritmo de detección de postura. El personal médico puede utilizar una foto de referencia o la cámara *web*, para que el sistema generé una pose inicial en base a la postura observada. Posteriormente, puede realizar los ajustes necesarios a la postura generada de manera manual. En la Figura 6 se puede observar el maniquí en distintas posturas que fueron configuradas a través del proceso descrito anteriormente.

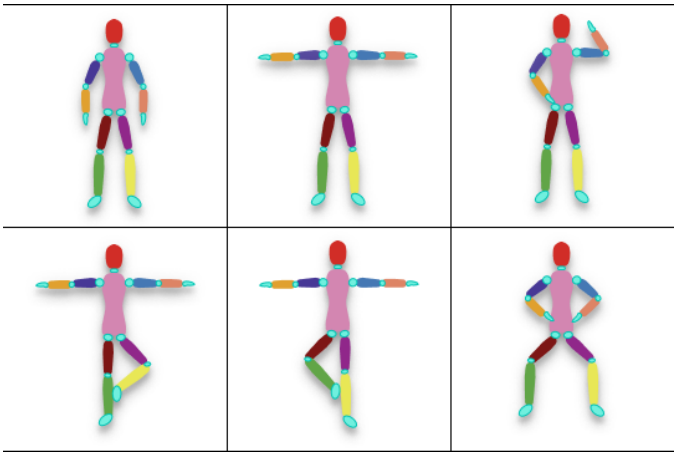

Figura 6: Maniquí Provisto por la Aplicación para la Configuración de las Poses Objetivo

D. Aplicación del Médico

La configuración del maniquí virtual descrita anteriormente se realiza a través de una aplicación *web* orientada al personal médico.

Un ejemplo de la interfaz en uso se puede observar en la Figura 7. En la parte A se representa el proceso de configuración del ángulo del brazo. La interfaz consta de una barra de identificación en la parte superior (1), en la cual se encuentra el nombre de la aplicación, un identificador de nuevos eventos ocurridos en el sistema y el control de sesión del usuario. Un panel a la izquierda (2) con las opciones presentes en el sistema, como: panel de control, estadísticas, asignación de la sesiones de terapia, edición de ejercicios (seleccionada actualmente), manejo de pacientes, manejo de personal médico y vista de los dispositivos terapéuticos registrados en el sistema. En la parte superior central (3) se encuentra un campo de texto donde se identifica la postura creada, seguido de tres botones (4) que permiten filtrar los huesos del esqueleto que se despliegan en el área de trabajo, entre los huesos evaluados en la postura y los que no son evaluados. Posteriormente se encuentra el esqueleto con la postura actual (6), el médico puede interactuar con el esqueleto manipulando las articulaciones que definen un hueso o modificando el ángulo de apertura del mismo en la barra lateral derecha (12). Las articulaciones se presentan en color celeste, el hueso actualmente seleccionado en un color morado oscuro (7), los huesos activos en morado claro (8) y los inactivos en blanco. Para el hueso activo, se despliega de manera visual el ángulo que se esta evaluando, representado por un semicírculo purpura (10) y un cono que representa el margen de tolerancia asociado al hueso (9). En la esquina superior derecha se encuentra un indicador de orientación en el espacio (5) que específica dónde está la izquierda, derecha, arriba y abajo del esqueleto actualmente visualizado. Dichas indicaciones cambiarán en base a la vista que se seleccione del maniquí, la cual puede ser modificada en la barra inferior central (11) entre la vista frontal y dos vistas laterales. Finalmente, el panel derecho contiene la configuración actual de todos los huesos del esqueleto, en el mismo el usuario podrá: activar y desactivar la evaluación de un hueso para la postura actual, cambiar el ángulo esperado de un hueso y su margen de tolerancia.

La parte B ilustra cómo cambia el estado visual de la interfaz al seleccionar el antebrazo en el esqueleto (1). En este caso, el antebrazo pasa a ser de color purpura y el brazo cambia a un morado claro (2). Además, se despliega el ángulo evaluado para el antebrazo (4) y el cono que representa el margen de tolerancia (3).

E. Aplicación del Paciente

Finalmente, en la Figura 8 se puede observar un ejemplo de la interfaz del paciente en cada uno de los distintos estados que despliega a medida que el usuario realiza el ejercicio solicitado.

Como se puede observar, se utiliza el maniquí como representación visual de la postura que debe realizar el paciente. La posición del maniquí cambia en base a la posición del paciente dentro de la imagen, esto con la intención de que el maniquí no interrumpa la visibilidad del paciente al realizar el ejercicio. A su vez, el color del maniquí corresponde con los mismos colores desplegados en el esqueleto detectado sobre el paciente, esto procura facilitarle el proceso de identificación de las extremidades involucradas en la pose actual.

Se puede observar en la parte A de la Figura 8, como se despliega un cono con el margen de tolerancia asociado al ángulo esperado del hueso. El color del cono es el mismo que el del hueso al que corresponda para facilitar la identificación del movimiento a realizar. Cuando el usuario mueve la extremidad actual dentro del cono, el mismo desaparece y se despliega el cono correspondiente al margen de tolerancia de la siguiente extremidad mal posicionada (como se ilustra en la parte B y C). La representación jerárquica del esqueleto, permite el despliegue progresivo de los conos de tolerancia en el orden en el que el paciente debe realizar los movimientos para alcanzar la postura. Por ende, no se despliega el cono de tolerancia de un hueso hijo cuando el padre está mal ubicado. El despliegue progresivo de los elementos, disminuye la carga cognitiva que recibe el paciente al servir de guía en los pasos a seguir para alcanzar el objetivo. El resto de la secuencia muestra al paciente siguiendo las instrucciones que el sistema dispone de manera visual hasta que logra completar la postura y se le indica la conclusión del ejercicio mediante un mensaje.

Además de los conos que representan el margen de tolerancia, se despliegan un conjunto de flechas que indican la dirección general en la que el paciente debería mover la extremidad para alcanzar la posición deseada. Dichas flechas parten del punto medio de la extremidad actualmente evaluada (siguiendo la estructura jerárquica) y su dirección es calculada siguiendo la ecuación (4), que calcula el vector de desplazamiento escalado c, necesario para llevar la representación vectorial del hueso actual a, al esperado b.

$$c = (b - a)|a| \tag{4}$$

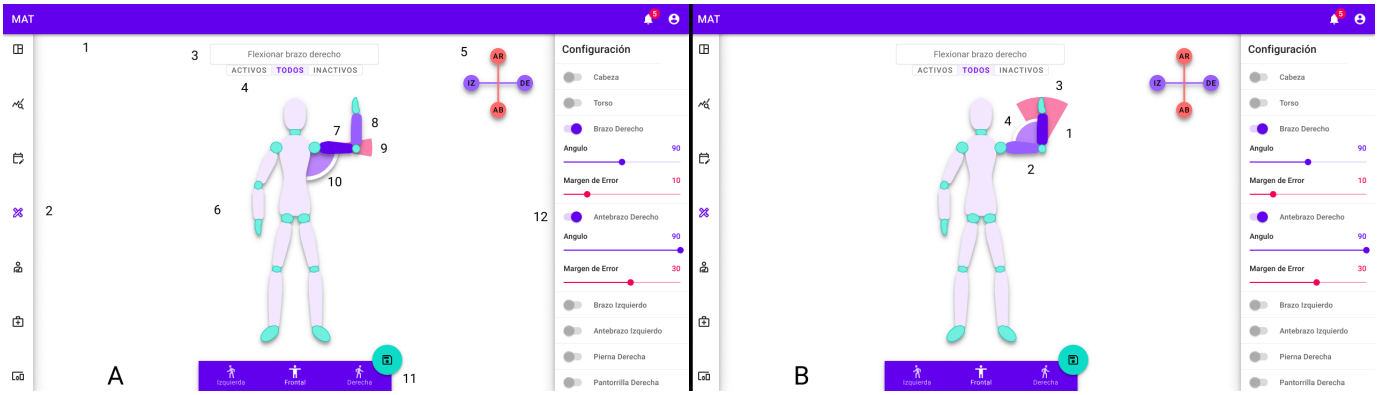

Figura 7: Vista de la Interfaz Utilizada por el Personal Médico para Configurar las Poses

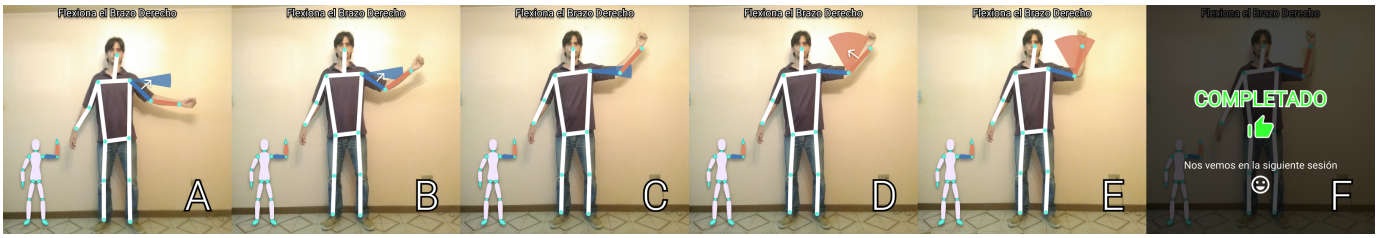

Figura 8: Secuencia de Interacción con la Aplicación del Paciente

F. Integración con Plataforma IoT

La propuesta descrita hasta el momento, constituye un caso de uso específico de un dispositivo inteligente a integrar dentro de la arquitectura basada en *IoT* planteada en [4]. Para integrarlo dentro de dicha arquitectura es necesario: registrar el dispositivo, los tipos de ejercicios que soporta y sus parámetros, el tipo de dato correspondiente a los resultados que otorga el dispositivo y, tomar las consideraciones de comunicación necesarias para que el dispositivo pueda comunicarse efectivamente con los servicios de la arquitectura.

1) Registro del Dispositivo: El registro del dispositivo se realiza mediante una interfaz *web* diseñada para los desarrolladores de dispositivos inteligentes. En la Figura 9 se puede ver una fracción de dicha interfaz, en esta el desarrollador del nuevo dispositivo indica:

- El nombre del dispositivo.
- Una breve descripción sobre su función.
- Un conjunto de palabras claves utilizadas para buscarlo en la lista de dispositivos registrados.
- El sitio *web* del desarrollador (opcional).
- La lista de ejercicios que soporta.

Cada ejercicio en la lista de ejercicios soportados, contiene los siguientes datos:

- Un descriptor (nombre) de tipo de ejercicio.
- Descripción.
- Un conjunto de palabras claves para buscar ese tipo de ejercicios.
- Una lista con los parámetros de configuración requeridos.

Por cada parámetro a configurar se debe indicar:

- El tipo de dato del parámetro, entre los cuales se soportan los tipos: entero, flotante, lógico, cadena de caracteres, listas y objetos.
- Nombre del parámetro.
- Valor por defecto.
- Si el parámetro es obligatorio u opcional.

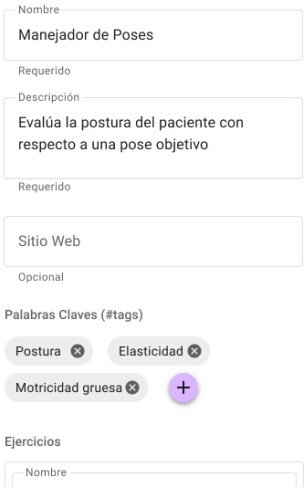

Figura 9: Configuración del Dispositivo

Dependiendo del tipo de dato del parámetro, se especifican restricciones a los valores que puede tener. Un tipo de dato numérico puede tener asociado un valor mínimo o máximo, una cadena de caracteres una longitud mínima o máxima y una lista tiene igualmente asociada una posible longitud mínima o máxima, además del tipo de dato por el que está compuesta. Construir un tipo dato "objeto", consiste en indicar una lista con los atributos que contiene, cada uno de los atributos tiene la misma configuración que un parámetro tradicional, lo que permite construir un objeto cuyos atributos sean a su vez

objetos.

Finalmente, los datos que registra el dispositivo son configurados mediante una lista de parámetros. Cuando el dispositivo genere los resultados de la ejecución de un ejercicio, serán validados con la configuración especificada en este punto. Adicionalmente, el dispositivo indicará un porcentaje global de éxito asociado al ejercicio. Dicho porcentaje es calculado por el dispositivo, ya que el mismo contiene la lógica necesaria para evaluar la calidad de un ejercicio realizado. Este porcentaje será utilizado como un número referencial para evaluar la calidad de la ejecución del paciente.

De igual manera, posterior a la configuración del dispositivo, la aplicación utilizada por el personal médico tendrá la capacidad de crear un formulario con controles para cada uno de los parámetros que componen un ejercicio de la terapia (similar al panel 12 de la Figura 7). En el caso específico de las poses, los parámetros del ejercicio corresponden con una lista de los objetos que definen la dirección del vector que representa cada hueso del esqueleto. Debido a que los parámetros son de tipo espacial y tienen relaciones jerárquicas entre si, resulta complejo configurarlos mediante campos numéricos tradicionales (configuración por defecto mostrada por la interfaz del médico). Por ende, se desarrolló una interfaz especial que facilite la configuración de dicho tipo de ejercicio, la cual se pudo observar en la Figura 7. En caso de que un nuevo dispositivo trabaje con parámetros complejos, que requieran el diseño de una interfaz particular para facilitar su manipulación, el desarrollador del mismo debe ponerse en contacto con el administrador de la arquitectura para realizar las adecuaciones pertinentes. Algunos ejemplos de datos complejos pudiesen ser: puntos geográficos (geolocalización), puntos en el espacio tridimensional o la representación de una postura mediante un maniquí virtual para el caso de uso particular del dispositivo presentado.

2) Consideraciones de Comunicación: Recordando [4], los dispositivos inteligentes se comunican con los servicios que componen la arquitectura, mediante el protocolo de comunicación *MQTT*[3] (*Message Queue Telemetry Transport* - Transporte de Telemetría de Cola de Mensajes).

El dispositivo enviará los resultados de un ejercicio a un tópico específico dispuesto para dicha función, suministrando el identificador único del ejercicio asignado y el resultado de dicho ejercicio. La estructura de un mensaje con estas características puede ser observada en la Figura 10, donde el campo "QUALITY" es el valor global de evaluación del ejercicio (rango $[0 - 100]$). Cuando la arquitectura registre los resultados, el dispositivo recibirá un mensaje indicando el nuevo estatus del ejercicio asignado como completado. De igual manera, si el dispositivo pierde la conexión con el servidor de mensajería, puede solicitar al momento de restablecer la conexión, las sesiones de terapia que pudieran haberle sido asignadas durante su tiempo de desconexión (ver Figura 11).

[3]http://mqtt.org

TOPIC:	exercise/save-results
ID:	c187738b-819f-4c02-85f8-af926430e514
RESULTS:	steps: 1500
	time: 216.000
QUALITY:	98

Figura 10: Campos de un Mensaje que Registra el Resultado de un Ejercicio de un Dispositivo Cuenta Pasos

TOPIC:	exercise/get-assigned
DEVICE ID:	e96122a5-0573-4e70-ab66-976f405ceddc
SINCE:	1c6b1f91-30db-4c0c-82ff-da6107e26b45

Figura 11: Solicitud de Últimos Ejercicios Asignados

IV. Pruebas

Todas las pruebas que se aplicaron al sistema, se realizaron en un equipo con las siguientes características:

- Procesador: Intel(R) Core(TM) I5-8250U.
- Memoria RAM: 8 GB.
- Gráficos Integrados: Intel(R) UHD 620.
- Sistema Operativo: Windows 10.

Y un dispositivo *Android* con las siguientes características:

- Procesoador: Snapdragon 430 1.4GHz.
- Memoria RAM: 3 GB.
- Sistema Operativo: Android 9.

A. Tiempo de Procesamiento

Se midieron los tiempos involucrados en cada una de las etapas que conllevan el proceso de identificar una postura, con la intención de detectar potenciales cuellos de botella que impidan que el producto final, pueda considerarse una aplicación en tiempo real. En la Tabla I se reflejan los tiempos de cada una de las etapas individuales del proceso de identificación de una postura. Como se puede observar, el tiempo total para procesar una postura es de 0,122 milisegundos, lo cual representa una tasa de 8.197 esqueletos procesados por segundo. Siendo la etapa más lenta de todo el algoritmo la normalización del esqueleto, que representa 43% del tiempo involucrado.

En la Tabla II se puede observar claramente las diferencias en velocidad de procesamiento de cada dispositivo de adquisición, donde se evidencia que el *Microsoft Kinect* es el método de adquisición más rápido, seguido por *PoseNet* y finalmente *OpenPose*. El tiempo de la identificación de la postura se mantiene constante ante todas las pruebas, debido a que solo depende del procesamiento de los esqueletos. El algoritmo de identificación de poses siempre realiza las mismas operaciones independientemente de la complejidad de la pose a evaluar, el procesamiento sería menor en caso de que la pose involucre

Tabla I: Tiempos por Etapas del Proceso de Comparación

Etapa	Tiempo	Fps
Esqueleto Estándar	0,042 ms	23.890
Normalización	0,053 ms	18.867
Comparación	0,027 ms	37.037
Total	0,122 ms	8.197

Tabla II: Tiempos de los Métodos de Adquisición

Método Adquisición	Detección Esqueleto	FPS
Kinect	33,05ms	30
Open Pose (CPU)	12.106,62 ms	0
Pose Net (MobiNet V1)	89,12 ms	11
Pose Net (ResNet 50)	109,63 ms	9
Pose Net (Android)	159,42 ms	6

una menor cantidad de huesos. Sin embargo, en las pruebas realizadas se consideró el peor caso posible, que sería que todos los huesos del esqueleto estén involucrados en la pose.

En base a los resultados expuestos en ambas tablas, se puede constatar que la velocidad del sistema reside en la del dispositivo de adquisición y que este representa un cuello de botella. Claramente la mejor alternativa es el *Microsoft Kinect* ya que permitiría que la aplicación se ejecute a 30 fotogramas por segundo, lo cual es una velocidad más que aceptable para una aplicación interactiva. Sin embargo, ya que dicha alternativa involucra adquirir un dispositivo de adquisición especializado, se considera que la mejor alternativa es el uso de *PoseNet*, que aunque es más lento, puede trabajar con una cámara *web* convencional. La aplicación con *PoseNet* se ejecuta a 9 fotogramas por segundo en su configuración más precisa y lenta y, a 11 fotogramas por segundo en la configuración menos precisa. Dicho rango se encuentra dentro del espectro en el que el humano percibe una secuencia de fotos, como un vídeo y no como imágenes separadas. Dada la velocidad de procesamiento, más la naturaleza de la aplicación, la cual involucra alcanzar una postura gradualmente sin necesidad de movimientos bruscos, permiten que *PoseNet* sea una alternativa viable para distribuir el sistema sin necesidad de requerir *hardware* especializado. De igual manera, se considera factible su configuración para dispositivos móviles, en la que el sistema fue capaz de operar a 6 fotogramas por segundo.

Finalmente, *OpenPose* no es capaz de alcanzar si quiera un fotograma por segundo con la configuración de *hardware* probada, por lo cual corresponde a la opción menos recomendada para la distribución del sistema. Para que *OpenPose* sea viable requiere de una tarjeta gráfica de alto poder de cómputo y por ende un mayor costo monetario. Sin embargo, es importante aclarar, que fue la alternativa que logró detectar poses complejas con mayor facilidad.

B. Calidad de Detección

Con el objeto de verificar la efectividad del sistema para discernir entre poses correcta e incorrectamente realizadas, se aplicaron pruebas sobre un conjunto de datos provisto por la Universidad Politécnica de Cataluña [20]. Los datos están compuestos por imágenes de 12 actores realizando un total de 10 poses de distintas complejidades. Entre los actores se encuentran 10 personas de sexo masculino y 2 de sexo femenino de distintas alturas y contexturas. Es importante destacar, que las imágenes que componen el conjunto de datos no fueron capturadas haciendo uso del dispositivo terapéutico propuesto en este trabajo. Por ende, los autores de las imágenes califican la pose realizada por el actor en base a su similitud general con respecto a la pose objetivo, lo que trae como consecuencia

Tabla III: Imágenes por Pose e Individuo

	Pose				
Individuo	**1**	**2**	**3**	**4**	**5**
1	38	49	33	43	37
2	33	41	32	36	36
3	25	26	26	27	31
4	28	30	36	25	30
5	34	37	42	35	31
6	33	34	28	34	35
7	37	27	34	29	25
8	47	31	37	39	31
9	30	39	35	34	27
10	33	34	26	30	33
11	35	32	36	25	28
12	28	23	28	27	29

que dos posturas que no son realizadas exactamente igual, sean calificadas como la misma pose. La Figura 12 ilustra el problema descrito anteriormente, donde los 3 actores están realizando la misma pose; sin embargo, se puede observar claramente que existen diferencias entre las ejecuciones de cada uno.

Aunque el conjunto de datos está agrupado en 5 poses básicas y 5 poses complejas, en las pruebas se utilizó exclusivamente el grupo de poses básicas (ver Figura 13). Esta decisión se tomo en base a dos criterios: las poses calificadas como "básicas", involucran más movimientos distintivos en las articulaciones de los sujetos; por otra parte, las poses "complejas" requieren un mayor esfuerzo de resistencia y equilibrio por parte de los actores (mantener piernas flexionadas o elevar una pierna en el aire), estas características no las hacen más complejas en cuanto a la evaluación de ángulos se refiere, pero si enfatiza la variabilidad en la ejecución de la pose entre cada sujeto, lo que acarrea mayores errores de calificación referentes a la exactitud de la pose realizada, por parte de los autores del conjunto de datos.

Para realizar las pruebas se seleccionó un individuo como referencia y por cada pose se eligió una imagen que identifique la misma, por lo tanto 5 imágenes (una por cada pose) fueron utilizadas como referencia para el individuo seleccionado. Una vez definidas las poses bases, se comparó cada pose contra todas las posturas realizadas por cada uno de los individuos en el conjunto de prueba, incluyendo el individuo utilizado como referencia. En la Tabla III se puede observar la cantidad de imágenes (posturas) grabadas por pose de cada individuo, en la cual se puede evidenciar que al realizar las pruebas, la proporción de poses incorrectas con la que se compara la pose referencial es muy superior a las poses correctas; por ejemplo, la pose 0 será comparada contra 1954 posturas, de las cuales 401 (20.52%) corresponden con el intento de realizar la misma pose (pose 0) y 1553 (79.48%) con el intento de realizar una pose distinta.

De las pruebas se extrajo la matriz de confusión presente en la Tabla IV. Para todas las poses se evaluó cada uno de los huesos del esqueleto que representa al usuario. La columna del margen de tolerancia, corresponde con los grados de diferencia tolerados entre cada par de huesos comparado en el esqueleto. Para todos los huesos se utilizó el mismo margen de tolerancia,

Figura 12: Errores Cometidos por los Usuarios para Imitar la Misma Pose Base

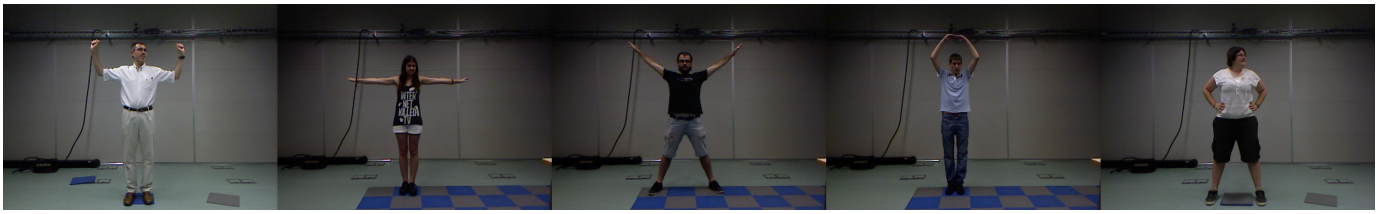

Figura 13: Poses Básicas Utilizadas en las Pruebas

Tabla IV: Matriz de Confusión

Margen de Tolerancia	VP	VN	FP	FN
≈ 0	5	7816	0	1949
5	173	7816	0	1781
10	278	7816	0	1676
15	478	7816	0	1476
20	750	7816	0	1204
25	1256	7816	0	698
30	1486	7816	0	468
35	1694	7789	27	260
40	1890	7745	71	64
45	1918	7613	203	36

sin embargo, en el dispositivo propuesto, el personal médico puede especificar un margen de tolerancia individual por cada hueso. Para calificar una pose como correctamente clasificada se utilizó las etiquetas (identificador de pose realizada) provistas en el conjunto de datos de prueba. Como se puede observar, con un margen de tolerancia muy pequeño (≈ 0) solo se califican como correctamente realizadas 5 poses (VP), estas corresponden con la comparación que se realiza entre la pose seleccionada como referencia y ella misma, este comportamiento es esperado debido a que el grado de exactitud exigido por cada hueso es muy estricto, lo que hace que sea prácticamente imposible para un individuo realizar la postura exigida. A medida que se incrementa el margen de tolerancia, los errores cometidos al realizar la misma pose son permitidos y por ende la cantidad de verdaderos positivos incrementa y la cantidad de falsos negativos decrece. Por otra parte, se puede observar que al incrementar demasiado el margen de tolerancia aparecen falsos positivos, esto se debe a que la cantidad de grados de diferencia permitida entre cada hueso es tan laxa, que dos posturas diferentes pueden corresponder con el intento de realizar la pose seleccionada como referencia, este comportamiento es esperado y es deseado para evaluar pacientes que tengan rangos de movilidad muy reducidos.

Tomando en cuenta los motivos por los cuales surgen los falsos positivo y negativos, se puede concluir que el comportamiento es el esperado y el correcto, debido a que no representan una mala calificación por parte del algoritmo de comparación de poses, sino una mala asignación de la etiqueta de la pose en el conjunto de datos, afirmación que es posible ya que la etiqueta no corresponde con el margen de tolerancia asociada a la pose objetivo, ya que si el mismo es muy bajo, el nivel de exigencia para realizar correctamente la pose es muy alto y la mayoría de los intentos por realizarla serán rechazados y en caso contrario cualquier intento será calificado como correcto. Es importante destacar que aunque el conjunto de datos utilizado no esta diseñado con los niveles de exactitud requeridos para una aplicación médica, la cantidad de falsos positivos generados durante las pruebas realizadas es muy pequeña respecto a la muestra completa ($<= 10\%$). Se considera que para las características del sistema, un falso negativo carece de importancia respecto a un falso positivo, debido que un falso positivo representa aprobar la ejecución de la terapia del paciente aunque el mismo no intentase realizar la pose asignada. Sin embargo, un falso negativo involucra exigirle al paciente mayor esfuerzo para cumplir con el grado de exigencia impuesto por el personal médico. Como el dispositivo cuenta con diversos indicadores visuales del progreso del paciente: como el despliegue del esqueleto evaluado, el maniquí virtual de referencia, los conos de margen de tolerancia y flechas que indican la dirección del movimiento esperado y, que además dichos elementos son presentados de manera progresiva para guiar al paciente a cumplir con el objetivo, se considera que el sistema es capaz de ayudar al paciente a superar los falsos negativos.

Finalmente, es necesario indicar que durante las pruebas se utilizó el algoritmo de detección de esqueletos *PoseNet*, el cual presentó errores de detección en el 11.4% (223 imágenes) de las imágenes procesadas. En la Figura 14 se perciben

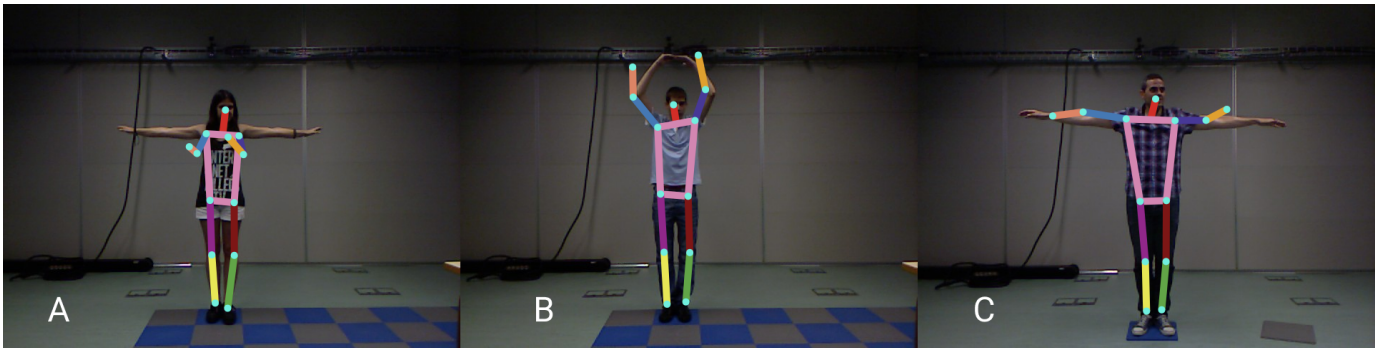

Figura 14: Errores de Detección de Esqueleto

algunos errores de detección, donde el esqueleto detectado no corresponde con la ubicación real del individuo. Estos errores de detección son producto de condiciones del ambiente en el que se realizó la prueba. Podemos observar que el tono de piel del individuo A puede confundirse con la pared de fondo; por otra parte, la disposición del cableado, tuberías y separadores en la pared de fondo pueden ser el motivo por el cual confunda la posición de alguna de las articulaciones de los individuos B y C. Ante estos problemas, se incluyó al sistema una lista de recomendaciones para minimizar los errores producto de una mala detección del esqueleto, como lo son:

- Utilizar ropa de distinto color al del fondo de la imagen.
- No utilizar ropa con rayas, preferiblemente prendas unicolor y no holgada.
- Procurar que el fondo de la imagen (lugar donde se realice la terapia) contenga la menor cantidad de elementos posibles.
- Remover elementos que obstruyan la visión del paciente.
- Contar con buena iluminación en el lugar donde se realizará la terapia.
- Alejarse de la cámara hasta que todo el rango de movimiento del paciente pueda ser capturado en la imagen.

Debido a que el paciente podrá observar el resultado de la detección del esqueleto sobre su cuerpo en tiempo real, se considera que será capaz de identificar una mala detección y los potenciales elementos que pueden estar generando dichos problemas en las imágenes capturadas por el dispositivo al consultar las sugerencias provistas.

V. Conclusión y Trabajos Futuros

En el presente trabajo se describió y probó un dispositivo capaz de evaluar la postura del paciente y compararla contra una pose diseñada por el personal médico. El sistema propuesto es de bajo costo monetario y preciso. Además, está diseñado para disminuir la carga cognitiva generada en el paciente al suministrarle indicaciones graduales y claras de los pasos que debe seguir para poder realizar el ejercicio.

El dispositivo presentado se complementa con una interfaz *web* que le permite al personal médico configurar y adaptar los ejercicios propuestos al paciente, mediante el uso de un maniquí virtual que representa la postura objetivo para este. Además, se indica el procedimiento para incluir un dispositivo dentro de la arquitectura basada en *IoT* previamente propuesta: desde su configuración, hasta las consideraciones que deben tomarse para realizar una comunicación efectiva con los servicios que comprenden la arquitectura.

Finalmente, se comprueba exitosamente la capacidad del algoritmo de comparación de poses diseñado, para evaluar correctamente las poses realizadas por el paciente. Siendo además capaz de trabajar con distintos mecanismos de adquisición de las características que definen su esqueleto.

Se propone como trabajo futuro, investigar más métodos de adquisición que mejoren el tiempo de respuesta y precisión del sistema, cumpliendo con la premisa de no involucrar la adquisición de *hardware* especializado costoso que dificulte su uso. De igual manera, se deben desarrollar otros dispositivos terapéuticos que sigan esta misma línea de pensamiento y fortalezcan la red de salud generada por la arquitectura *IoT*.

Referencias

[1] World Health Organization, *Rehabilitation*, World Health Organization, report, 2020.

[2] K. Bandi, P. Pravati, P. Gopal, E. Jayasettiaseelon, Y. Sreekanth, M. Sridhar, and G. Gaurav, *Effect of Yoga Therapy on Heart Rate, Blood Pressure and Cardiac Autonomic Function in Heart Failure*, Journal of clinical and diagnostic research : JCDR, vol. 8, no. 1, pp. 14–6, 2014.

[3] L. Bezants, *The Role of Exercise in the Management of Stroke*, 2016.

[4] F. Moreno and W. Hernandez, *Sistema de Apoyo a la Terapia Física Basado en el Internet de las Cosas*, Revista Venezolana de Computación, vol. 6, no. 1, pp. 28–38, 2019.

[5] A. Lotfi, C. Langensiepen, and S. Yahaya, *Socially Assistive Robotics: Robot Exercise Trainer for Older Adults*, Technologies, vol. 6, pp. 32, 2018.

[6] Gov UK, *Exercises for Older People*, https://www.nhs.uk/live-well/exercise/physical-activity-guidelines-older-adults/, 2019.

[7] I. Gregory and S. Tedjojuwono, *Implementation of Computer Vision in Detecting Human Poses*, in proceedings of the 2020 International Conference on Information Management and Technology (ICIMTech), pp. 271–276, 2020.

[8] S. Yadav, A. Singh, A. Gupta, and J. Raheja, *Real-time Yoga Recognition Using Deep Learning*, Neural Computing and Applications, vol. 31, 2019.

[9] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, *Real-Time Human Pose Recognition in Parts from Single Depth Images*, in proceedings of Communications of the ACM, vol. 56, pp. 1297–1304, 2011.

[10] Z. Cao, G. Hidalgo, T. Simon, S. Wei, and Y. Sheikh, *OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019.

[11] Z. Cao, T. Simon, S. Wei, and Y. Sheikh, *Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields*, in proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017), Honolulu, HI, USA, 2017.

[12] T. Simon, H. Joo, I. Matthews, and Y. Sheikh, *Hand Keypoint Detection in Single Images using Multiview Bootstrapping*, in proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017), Honolulu, HI, USA, 2017.

[13] S. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, *Convolutional Pose Machines*, in proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016), Las Vegas, NV, USA, 2016.

[14] G. Papandreou, T. Zhu, N. Kanazawa, A. Toshev, J. Tompson, C. Bregler, and K. Murphy, *Towards Accurate Multi-person Pose Estimation in the Wild*, in proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017), Honolulu, HI, USA, 2017.

[15] G. Papandreou, T. Zhu, L. Chen, S. Gidaris, J. Tompson, and K. Murphy, *PersonLab: Person Pose Estimation and Instance Segmentation with a Bottom-Up, Part-Based, Geometric Embedding Model*,in proceedings of the 2018 European Conference on Computer Vision (ECCV), Munich, Germany, 2018.

[16] A. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*, in proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017), Honolulu, HI, USA, 2017.

[17] K. He, X. Zhang, S. Ren, and J. Sun, *Deep Residual Learning for Image Recognition*, in proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016), Las Vegas, NV, USA, 2016.

[18] P. Green-Armytage, *A Colour Alphabet and the Limits of Colour Coding*, Color: Design & Creativity, vol. 5, pp. 1–23, 2010.

[19] P. Borkar, M. Pulinthitha, and A. Pansare, *Match Pose – A System for Comparing Poses*, International Journal of Engineering Research & Technology (IJERT), vol. 8, no. 1, pp. 28–38, 2019.

[20] J. Hidalgo and J. Casas, *Body Pose Dataset*, https://imatge.upc.edu/web/resources/body-pose-dataset, 2015.

Revista Venezolana de Computación
ISSN: 2244-7040
http://www.svc.net.ve/revecom
Vol. 8, No. 1, pp. 13-24, Junio 2021
Fecha recepción: 05/03/2021, Fecha aceptación: 20/07/2021

Un Experimento Curricular en Ciberética y Sociedad: Creación de un Curso de Innovación Responsable en Tiempos de Pandemia

Alejandro Teruel[1]
teruel@usb.ve

[1] Departmento de Computación y Tecnología de la Información, Universidad Simón Bolívar, Caracas, Venezuela

Resumen: En este artículo se motiva y describe la creación de un curso electivo experimental sobre el enfoque de Innovación Responsable de la Unión Europea como escalón hacia la creación de uno o más cursos en Ciberética y Sociedad que cubran los tópicos claves recomendados incluidos para el área del conocimiento denominado "Social Issues and Professional Practice". en el Curriculum de Ciencias de la Computación de la ACM y la IEEE Computer Society. La electiva fue diseñada para dictarse en línea dadas las restricciones imperantes en Venezuela durante la pandemia COVID-19 y se montó sobre un curso en línea abierto masivo (MOOC) de la Universidad de Delft para la plataforma edX. Se reportan brevemente los resultados obtenidos después de dictar la asignatura por primera vez.

Palabras Clave: Ética; Sociedad; Innovación Responsable; Educación; Computación; MOOC.

Abstract: This paper justifies and describes the creation of an experimental elective course on the European Union's Responsible Innovation framework as a stepping stone to the creation of one or more courses on Cyberethics and Society to cover the Association of Computer Science (ACM) and IEEE Computer Society's 2013 Computer Science Curriulum's "Social Issues and Professional Practice" knowledge area. The new course was designed to be taught online due to Venezuela's COVID-19 pandemia restrictions at the time and it was built on a University of Delft massively open online course (MOOC) available on the edX platform. The results obtained after teaching the course for the first time are reported briefly.

Keywords: Engineering Ethics; Society; Responsible Innovation; Education; Computer Science.

I. Introducción

Desde 1991, las recomendaciones curriculares del ACM y la IEEE Computer Society para las Ciencias de la Computación (Computer Science) incluyen el área del conocimiento denominado "Social Issues and Professional Practice". sobre aspectos éticos, legales y sociales de la Computación. En las recomendaciones curriculares más recientes [1] se cita:

> "Los estudiantes de pregrado necesitan también entender los aspectos básicos culturales, sociales, legales y éticos inherentes a la disciplina de la Computación. Deben entender de dónde viene la disciplina, dónde está y hacia dónde va. También deben entender el rol individual que les toca desempeñar en este proceso, así como deben apreciar las preguntas filosóficas, los problemas técnicos y los valores estéticos que conforman una parte importante en el desarrollo de la disciplina.
>
> Los estudiantes también necesitan desarrollan la habilidad para hacer preguntas serias sobre el impacto social de la computación y para evaluar las respuestas propuestas. Los futuros profesionales deben ser capaces de anticipar el impacto que puede tener introducir un producto determinado en un ambiente determinado. Ese producto, ¿mejorará o empeorará la calidad de vida? ¿Qué impacto tendrá sobre individuos, grupos e instituciones?
>
> Finalmente los estudiantes deben estar conscientes de los derechos legales básicos de vendedores y usuarios de hardware y de software, y necesitan apreciar los valores éticos que fundamentan esos derechos. Los futuros profesionales deben comprender la responsabilidad que les toca ejercer y las posibles consecuencias de fracasos. Deben comprender sus propias limitaciones así como las limitaciones de sus herramientas."

En el curriculum del 2013 se reconoció que el enorme impacto que había tenido la Computación en la vida y sociedad contemporánea no había dejado de crecer, planteando cada vez más desafíos a la sociedad y a los profesionales de la Computación y áreas afines que son responsables de introducir, gestionar, operar, mantener y mejorar las aplicaciones y la

infraestructura computacional. Desde entonces los impactos son cada vez más profundos y más extendidos y el clamor social sobre las distorsiones y las injusticias que pueden introducir o acrecentar los sistemas basados en la Computación así como las bondades y esperanzas que se vierten sobre ellos obligan al profesional de la Computación a atender los aspectos éticos y sociales de manera más profunda y cuidadosa.

Este artículo cubrirá los precedentes para abordar temas de Ciberética y Sociedad para la carrera de cinco años de Ingeniería de Computación en la Universidad Nacional Experimental Autónoma Simón Bolívar en Venezuela y describirá dos propuestas de creación de un curso en el área para complementar los temas tratados en cursos más específicos como Ingeniería de Software 1. Se describe brevemente los resultados obtenidos después de la primera vez que se dictó la segunda de las propuestas, una propuesta basada en el enfoque de Innovación Responsable de la Unión Europea montado sobre un MOOC de la Universidad de Delft en la plataforma edX.

II. Antecedentes en la Universidad Simón Bolívar

La Universidad Simón Bolívar (USB) es una universidad nacional pública creada en 1969. La carrera de Ingeniería de la Computación en la USB se empezó a dictar en 1973. Siempre ha tenido una alta demanda, seleccionándose entre 1 de cada 5 hasta 1 de cada 8 aspirantes a la carrera en la Universidad. Los egresados de la carrera son muy demandados por la industria y aquellos que han querido continuar estudios de postgrados en la propia universidad, otras universidades nacionales o internacionales de primera línea han sido muy exitosos en los estudios realizados.

Los planes de estudio de la Universidad Simón Bolívar contienen un conjunto de asignaturas de Estudios Generales que proporcionan una importante formación cultural, social y ética. Estos conocimientos son básicos y comunes a la formación integral característica de la USB. En el Apéndice A se proporcionan algunas notas adicionales al respecto. Estos cursos pueden proporcionar algunos elementos básicos que preparan, concientizan o sensibilizan al estudiante a aspectos culturales, sociales o éticos pero sin atarlos específicamente a la Computación. En los estudios conducentes al título de Ingeniero de la Computación en la Universidad Simón Bolívar (USB), los temas éticos, legales y sociales inherentes a la carrera se han venido trabajando de manera fragmentada y muchas veces, implícita en asignaturas obligatorias y electivas de Ingeniería de Software, Bases de Datos, Redes de Computadores, Robótica, Sistemas de Información y Ciberseguridad. La asignatura obligatoria Ingeniería de Software 1 incluye, desde hace más de 20 años, explícitamente un tema dedicado a aspectos legales y éticos de la Ingeniería de Software muy orientado hacia las leyes venezolanas sobre la informática y la deontología asociada a códigos éticos profesionales como el código ético del Colegio de Ingenieros de Venezuela y el código de la ACM. Este material se cubre en poco más de una semana (4-6 horas) de las doce semanas del curso.

III. Una Primera Propuesta

Las recomendaciones curriculares del ACM y la IEEE Computer Society indican que el material sobre ética y sociedad es mejor cubrirlo entre una asignatura obligatoria y algunos módulos cortos en otros cursos de la disciplina. En concordancia con esta idea, se consideró que se debía crear una asignatura específica dedicada a Ciberética y Sociedad.

Se decidió diseñar la asignatura mediante un proceso que tomara en cuenta:

1. Una revisión bibliográfica sobre el tema de diseño y dictado de este tipo de curso.
2. La identificación de profesores de la Universidad con experticia en el tema,
3. Los retos surgidos por la pandemia de COVID-19 que incluyeron la necesidad de dictar la asignatura a distancia, el tomar en cuenta los muy pobres niveles de acceso a Internet existentes enVenezuela y el bajo nivel de adquisición de la mayoría de profesores y estudiantes dada la profunda crisis que viven los venezolanos.
4. El aprovechamiento de material ya existente, primordialmente de acceso abierto, evitando en lo posible el síndrome NIH (Not Invented Here).
5. La vinculación del curso a los valores éticos de la Universidad.
6. Un carácter experimental para el curso, abierto a su evolución en el tiempo y gracias a la experiencia que se fuera adquiriendo, particularmente al ir mejorando el ajuste del curso a las características y necesidades de la carrera, la Universidad y el país.

Se creó un foro (Grupo Google) interdisciplinario dedicado a la discusión de Ciberética y Sociedad al cual se unieron profesores de las áreas de Computación, Ciencias Políticas, Ciencias Sociales y Ciencias Económicas y Administrativas entre otros. Adicionalmente se ha ido creando una lista bibliográfica en el área y, después de dos años de trabajo, se elaboró una primera propuesta curricular que fue entregado a la Coordinación de Computación de la USB a principios del 2020.

Crear un curso en Ciberética y Sociedad plantea al menos tres retos importantes:

1. ¿Qué contenidos y competencias atender en el curso
2. ¿Cuál debe ser la estrategia pedagógica del curso?
3. ¿Dónde encontrar o cómo formar los profesores para el curso?

Estas preocupaciones se han venido trabajando en la literatura tanto específica a la Computación [2]-[9] como general a la Ingeniería [10][11][12]. Las propuestas incluyen el uso de ciencia ficción [5], juicios simulados [6], debates [11], recomendaciones para rúbricas de corrección [7] y una propuesta específica para incluir sustentabilidad en la Ingeniería de Software [9]. Cabe destacar la recopilación de más de 115 programas en el área [8] y el muy completo y rico texto de la National Academy of Engineering [10].

En esta primera propuesta, que fue posteriormente retirada de la consideración de la Coordinación de Computación, a solicitud de su autor, se incluyeron 32 tópicos agrupados en siete temas: (1) Fundamentos de la ciberética, (2) Acceso a la computación, (3) Privacidad y libertades civiles, (4) Ciberética y desarrollo sustentable, (5) Ciberética y cultura, (6) Propiedad intelectual, y (7) Uso de herramientas colaborativas.

En la sección de la propuesta sobre la estrategia metodológica se advirtió que:

> "No se considera factible cubrir todos los temas mencionados previamente en detalle por lo que se procurará cubrir al menos un tópico dentro de cada tema. Preferiblemente los tópicos a abordar de todos los temas menos el primero serán de mutuo acuerdo entre el profesor y los estudiantes, hasta que se haya obtenido suficiente experiencia como para determinar qué tópicos deben cubrirse en este curso y cuáles pueden incorporarse a otros cursos. El nivel de detalle de cobertura del primer tema (Fundamentos de la Ciberética) será determinado por el profesor de la asignatura."

Las competencias para el curso se ciñeron a las recomendaciones curriculares del ACM y IEEE Computer Society [1], teniéndose también el debido cuidado para vincular el curso explícitamente al Decálogo de Valores de la Universidad Simón Bolívar [13].

En cuanto a la estrategia pedagógica del curso, la riqueza de propuestas de la literatura sugirieron no atarse a ninguna inicialmente sino ir explorando distintas estrategias para determinar cuál sería más factible y efectiva para las circunstancias actuales.

El reto de escoger una estrategia pedagógica estaba muy vinculado al reto de encontrar profesores para el curso. Ninguno de los profesores de Computación interesados en el área tiene formación en Ética o en Ciencias Sociales por lo que se intentó crear alianzas con profesores de otros Departamentos para dictar un curso interdisciplinario o, en su defecto, multidisciplinario como se había hecho en el pasado para otras asignaturas. Las posibles alianzas se exploraron a través del grupo Google sobre Ciberética y Sociedad mencionado previamente y contactos personales, dentro y fuera de la Universidad. Esto condujo, gracias a la recomendación del Prof. Rafael Tomás Caldera, actualmente adscrito a la Universidad Monteávila, a una muy valiosa asesoría por parte del Prof. Raúl González Fabré de la Universidad de Comillas en Madrid. Lamentablemente, el contexto de la crisis humanitaria compleja venezolana acelerada por la pandemia de COVID-19, la fuga de profesores debido a la crisis en general y el colapso de sueldos universitarios a niveles internacionales de pobreza crítica y las crecientes cargas y responsabilidades académicas y administrativos de quienes quedaban en la Universidad no permitieron consolidar un equipo docente para el curso, por lo que, pese al interés de varios profesores, el reto quedó en manos de un solo profesor.

IV. La Segunda Propuesta

Uno de los riesgos de dictar un curso como el planteado en la primera propuesta es que el curso no "fluya"; trata un conjunto de temas más o menos interesantes pero dispares y poco integrados, lo que tienta a hacer el curso más informativo que formativo. Este riesgo se incrementa cuando el profesor no tiene formación formal en Ética y más en una situación de pandemia que obliga a dictar el curso en línea, en un país con serias deficiencias en el acceso a Internet.

Esto llevó a poner de lado la primera propuesta por los momentos.

Se tuvo la oportunidad de cursar el MOOC sobre Innovación Responsable de la Universidad de Delft [14] y de leer el libro de Sarah Spiekermann sobre el enfoque basado en valores [15]. Estas dos experiencias permitieron intuir las posibles bondades de armar un curso cuyo hilo conductor podía ser la Ética de Tecnología orientada al proceso de análisis y diseño en el desarrollo de software.

El enfoque de Diseño Basado en Valores (VBD) [15] postula la necesidad de incluir requisitos no funcionales basados en valores éticos, e incorporar en el proceso de análisis y diseño de software el levantamiento de los valores claves a incorporar en el producto a desarrollar, la cuidadosa conceptualización e investigación de esos valores y la atención debida a la incidencia sobre los valores y situación de todos los afectados por el producto ("direct and indirect stakeholders"). VBD puede considerarse como una variante del enfoque de Diseño Sensitivo a Valores (VDS) de Batya Friedman y sus colegas y colaboradores [16].

El enfoque de Innovación Responsable de la Unión Europea surgió como un lineamiento clave en programa de investigación e innovación Horizon2020. La idea de base fue que el diseño e implantación de innovaciones debía prestarle particular, temprana y continua atención al posible impacto ético, social y ambiental de las innovaciones propuestas, de forma que las innovaciones realmente contribuyeran a un mejor mundo. El diseño orientado a valores y la Investigación e Innovación Responsables han sido definidos como:

> "Design for Values means making design choices explicit to and for reasons of moral and social values throughout the entire design or engineering processes" [17]

> "The EU […] defines [Responsible Research and Innovation (RRI)] as follows: an interactive process where societal actors, researchers and innovators actively cooperate to co-define, co-design and co-construct solutions, services and products that are socially acceptable and sustainable and resolve important societal issues." [18]

> "[...F]or an innovative organization or process to be praised as being "responsible", this would imply -among other things- that those who initiated it and were involved in it must be acknowledged as moral and responsible agents. In other words, they have to:
>
> - Obtain - as much as possible - relevant knowledge on (i) the consequences of the outcomes of their actions and (ii) the range of options open to them;
> - Evaluate all outcomes and options effectively in terms of relevant moral values (including, but not limited to, well-being, justice, equality, privacy, autonomy, safety, security, sustainability, accountability, democracy and efficiency) […]
> - Use these two considerations as requirements for the design and development of new technology, products and services, leading to moral improvement." [18]

Se estimó que los retos y riesgos asociados con la estrategia pedagógica, la falta de profesores formados en el área y la necesidad de dictar el curso en modalidad de curso en línea debido a las restricciones ocasionadas en Venezuela por pandemia de COVID-19 podían volverse más manejables al montar el curso de la USB sobre el MOOC de la Universidad

de Delft. A tal efecto, y a pesar que el MOOC tiene una licencia de Creative Commons que permitiría reusar la mayor parte de sus elementos, se escribió a la Universidad de Delft para ponerlos al tanto de la intención de montar un curso de la USB sobre el MOOC. Ellos autorizaron tal uso, condicionado al carácter de universidad pública de la USB.

Según la propuesta planteada a la Coordinación de Computación, las competencias principales a desarrollar al cursar las asignaturas son:

1. Formular preguntas de carácter ético con respecto a tecnología (nueva o emergente).
2. Incluir requerimientos éticos al levantar requerimientos no funcionales.
3. Tomar en cuenta la posible existencia de riesgos desconocidos (incertidumbre profunda) de tecnologías nuevas.
4. Presentar y defender reflexiones críticas sobre tecnologías nuevas o emergentes desde un punto de vista ético y de riesgos (casos de estudio).
5. Aplicar los conceptos de Innovación Responsable y Diseño Sensitivo a Valores (VSD) a casos sencillos y reseñar sus posibles consecuencias para el proceso de diseño.
6. Analizar una innovación tecnológica, preferiblemente asociada a la Computación, propuesta para resolver algún aspecto de los Objetivos de Desarrollo Sostenible (ODS) de la UNESCO o de los Grandes Desafíos de la Ingeniería de la National Academy of Engineering, aplicable a Venezuela o Latinoamérica para determinar si constituye una innovación responsable […]

En cuanto a los temas y tópicos a tratar se propuso seguir muy de cerca la estructura del MOOC de la Universidad de Delft, según se aprecia en la Tabla I:

Tabla I: Temas y Tópicos del MOOC de la Universidad de Delft

Tema 1	Innovación Responsable como aplicación de la Ética • Qué es la Innovación Responsable • El Problema del Tranvía desde el punto de vista del ingeniero • Responsabilidad individual y colectiva • El Problema de las Muchas Manos: ¿quién es el responsable?
Tema 2	Dilemas morales. Razón y Emociones. Institutos y valores • El Dilema de la Sobrecarga Moral • Razón, emociones y valores en la toma de decisiones • Caso de estudio: Responsabilidad en la Robótica • Sustentabilidad, ética y tecnología: la Tragedia de los Bienes Comunes
Tema 3	Comprensión e identificación de riesgos • Evaluación Tecnológica y la Innovación Responsable • Gestión de incertidumbre e ignorancia • Caso de estudio: Energía nuclear • Caso de estudio: Inteligencia Artificial • Enfoques para la evaluación del impacto de tecnologías
Tema 4	Análisis de riesgos e Ingeniería de Seguridad ("Safety Engineering") • Gestión de riesgos: Análisis de costo-beneficio • Gestión de riesgos: El compromiso entre riesgo y seguridad • Evaluación cuantitativa avanzada de riesgos • Caso de estudio: Vehículos autónomos • Caso de estudio: Ciberseguridad
Tema 5	¿Cómo surgen las innovaciones? • Innovación radical. • Innovación incremental. • Caso de estudio: Refrigerantes • Factores que inciden en la innovación • Caso de estudio: Innovación frugal
Tema 6	El diseño orientado a valores • Introducción • Aplicación • Caso de estudio: Armas autónomas, ¿quién hala el gatillo?

El programa completo de la asignatura incluyendo todas las competencias a desarrollar se encuentran en el Apéndice C. Posteriormente se evaluó otro MOOC sobre el tema desarrollado por la Universidad de Edimburgo sobre la misma plataforma edX, pero se mantuvo la decisión de trabajar sobre el MOOC de la Universidad de Delft.

El MOOC de la Universidad de Delft fue diseñado para que le resultara atractivo a quienes están interesados en tecnología por lo que incluye casos de estudio y ejemplos de muy variados campos de la Ingeniería. Muestra una comprensible orientación a ejemplos de interés en la Unión Europea y particularmente en Holanda. Estos sesgos se pueden ver en la selección de ejemplos sobre la tecnología de reactores nucleares, control de inundaciones por mar y armas autónomas. Otros temas son de interés más general pero en disciplinas percibidas como alejadas de la Computación como lo son el tratamiento de aguas negras, la sobre-pesca, el uso de nano-partículas en bloqueadores soleares y la sustitución de refrigerantes que dañan la capa de ozono. En el MOOC se incluyeron ejemplos y casos aplicadas a áreas percibidas como de, o muy afines a la Computación como vehículos autónomos, control de refinerías, robots en la medicina e inteligencia artificial.

Se consideró importante hacer algunos ajustes al MOOC para que fuera más motivante y asequible para los estudiantes de Ingeniería de Computación en la USB. Se elaboró un sitio web para el curso (https://sites.google.com/usb.ve/innova-responsable-de-software/aspectos-organizativos/bienvenida) que es una meta-guía al MOOC y puede accesarse escribiéndole al autor. El sitio web establece un orden diferente de cobertura de los temas, agrega lecturas o videos considerados más relevantes a la Computación, elimina o recomienda considerar ciertas lecturas y videos del MOOC como opcionales para el curso USB, y reemplaza los elementos evaluativos del MOOC por elementos propios al curso para permitirle al profesor del curso un mayor acceso y mayor retroalimentación a los estudiantes del curso. Algunos de estos elementos siguen muy de cerca los elementos evaluativos del curso mientras que otros son muy diferentes. Una decisión importante fue decidir cuáles temas y particularmente casos o ejemplos conservar del área de Ingeniería no directamente relacionada con la Computación. Dada que la carrera incluye seis asignaturas en Estudios Generales lo que representa más de 12% de la carga crediticia de la carrera, se decidió, algo optimistamente, para la primera dictada del curso, conservar la mayoría de los temas y ejemplos que no son de Computación. Las modificaciones más importantes tienen que ver con la eliminación del tema avanzado sobre manejo de riesgos basado en redes bayesianas (se consideró que los estudiantes no tenían la preparación necesaria para entender y aprovechar el tema) y la eliminación del proyecto final del MOOC, el cuál versaba sobre las armas autónomas y su reemplazo por un proyecto sobre los Objetivos de Desarrollo Sustentable (ODS) de la UNESCO y los Grandes Desafíos de la Ingeniería definidos por la National Academy of Engineering, un cambio inscrito inscrito en la idea de ajustar más el curso al contexto venezolano. Para que el proyecto sobre ODS tuviera mejor

cabida en el curso, se agreguó un tema al curso que versa sobre ODS, los Grandes Desafíos de la Ingeniería, la Computación Verde y la Sostenibilidad Computacional. También se hicieron cambios importantes en el tema del Diseño Sensitivo a Valores (VSD del MOOC), notablemente para incorporar elementos adicionales del Diseño Basado en Valores (VBD) para resaltar dos ejemplos de investigación e innovación responsable en la USB. Uno de los ejemplos fue el desarrollo de posibles aplicaciones para el "lodo rojo", un producto secundario altamente contaminante producido por la siderúrgica venezolana nacional y otro fue un cuidadoso estudio sobre el uso de bombas lacrimógenas con fecha de expiración vencida en la represión de protestas contra el gobierno venezolano. El primer ejemplo permitió referenciar el Parque Tecnológico Sartenejas, la incubadora de empresas innovadoras de la Universidad, También se agregó como caso de estudio de lo que significa justicia algorítmica los polémicos algoritmos usados en Venezuela para la asignación de aspirantes a carreras en universidades públicas. Adicionalmente el profesor incluyó algunos ejemplos o tareas con lecturas sobre el uso de la computación en el área de la salud, el uso de la inteligencia artificial en la selección de posibles empleados, los TIC verdes y la sostenibilidad computacional (software para apoyar el cumplimiento de ODS). Más detalles sobre los cambios realizados pueden encontrarse en un reporte interno [19].

Dados los problemas de acceso a Internet que se vive en Venezuela, también se decidió hacer el curso, como el MOOC, lo más asíncrono posible.

En cuanto a la estrategia pedagógica, se consideró muy importante que los estudiantes tuvieran oportunidad de estar expuestos a diferentes posiciones éticas que los obligaran a practicar discusiones en equipo sobre ética e impacto social y que el curso prestara especial interés a identificar valores y trabajar requerimientos éticos durante los procesos de análisis y diseño del sistema, evitando quedarse en la teoría o los fundamentos filosóficos de la Ética de la Tecnología. Por ello, se requirió que la mayor parte de las tareas se resolvieran en equipo y se varió tanto el tamaño de los equipos (de 2 a 6 miembros según la tarea) como la constitución de los miembros que conformaran los equipos. La variación en conformación también estimulaba la comunicación entre estudiantes, que dado el uso del MOOC y el diseño asíncrono del curso de la USB, podía verse reducido.

Dada la prohibición de acceso al campus universitaria decretada por el Ejecutivo el 18 de marzo 2020 ante la pandemia de COVID-19 y que no ha sido levantada a la fecha (15 de septiembre 2021), el curso se diseñó para ser tomado en línea. Inicialmente el curso se construyó sobre el MOOC de la Universidad de Delft, un sitio Google propio, el uso de Google Classroom, documentos Google compartidos entre estudiantes y/o el profesor para la elaboración, retroalimentación y discusión pre-corrección, revisión por pares y corrección de tareas, un Google Group en la modalidad de foro de debate, Google Forms para elaborar algunos quices y las encuestas del curso, un grupo Whatsapp para anuncios al grupo, consultas de grupo, así como el uso de Whatsapp, correo electrónico o teléfono para consultas individuales. Los estudiantes consideraron confuso el uso de tantas herramientas e incómodo el salto entre ellos, por lo que durante el propio curso se dejó de lado Google Classroom y Google Group, este último en buena parte por haber prescindido de la opción de foro-debate.

Se diseñó una encuesta de entrada y una de salida del curso. La encuesta de entrada estuvo diseñada para explorar los conocimientos previos y el interés de los estudiantes por los aspectos éticos y sociales, así como su experiencia con cursos a distancia, particularmente MOOC. La encuesta de salida fue mucho más detallada y proponía evaluar la percepción de los estudiantes en cuanto al dominio logrado en las competencias previstas para el curso, la idoneidad de los contenidos, la opinión que les mereció la estrategia y herramientas pedagógicas, más cualquier opinión adicional que tuvieran a bien hacer.

V. El Trabajo Final Sobre Objetivos de Desarrollo Sustentable (ODS)

Para tener una mejor idea de la adaptación principal que se hizo al MOOC, es conveniente revisar brevemente el Trabajo Final del curso, cuya evaluación corresponde al 45% de la evaluación del curso y que debía entregarse en varias etapas a lo largo de cinco semanas.

En la primera etapa (5%), cada estudiante debía leer la descripción de los 17 ODS y escoger los tres ODS que más le llamaran la atención, indicando brevemente lo que más le llamaba la atención de ellos y si la informática puede contribuir a alcanzarlos.

En la segunda etapa (5%), cada estudiante debía analizar los indicadores de dos metas de los tres ODS que más le llamaron la atención.

En la tercera etapa (sin nota) del proyecto, después de trabajar los tópicos de TIC verdes y sostenibilidad computacional, se forman equipos de 3 estudiantes que deben hacer una revisión bibliográfica con miras a escoger una innovación ya propuesta en la literatura o precedentes para una propuesta, preferiblemente asociada a la Computación, que consideran puede contribuir a algún aspecto de los ODS.

En la cuarta etapa (5%), el equipo analiza los Perfiles de ODS por país en el sitio web correspondientes a Venezuela y otro país latinoamericano escogido por el equipo para comparar sus resultados en dos ODS incluyendo y prestando particular atención al ODS a que estaría adscrito la innovación que escogió el equipo. El equipo debe resaltar cualquier anomalía o datos que le llamen la atención, indicar a cuál país le va mejor y, a vuelo de pájaro, indicar en cuál país considera que podría tener mayor impacto positivo la innovación escogida, siempre justificando las respuestas.

En la quinta y última etapa (30%), el equipo debe integrar el trabajo hecho en las etapas anteriores y analizar la innovación propuesta con base al siguiente esquema, justificando sus respuestas:

1. Breve descripción de la innovación a analizar, indicando claramente el ODS a que pertenece.
2. Cómo o por qué aplica a Venezuela o Latinoamérica. Estado de cumplimiento del ODS pertinente en Venezuela y otro país latinoamericano.
3. Afectados (stakeholders) por la innovación.

4. Proponer un modelo causal (Dinámica de Sistemas) para el ODS que supuestamente atiende la innovación. ¿Qué tipo de apalancamiento atiende? Tomando este modelo en cuenta responde qué problema resuelve la tecnología.
5. Identificar y analizar al menos tres valores, tres antivalores y tres riesgos involucrados en el desarrollo de la innovación. ¿Cómo se relaciona la propuesta con el Patrimonio Ética de la USB? ¿Qué emociones creen que suscitaría entre sus afectados? Traten de entrevistar algunos posibles afectados y reporten tanto los valores e intereses, los argumentos y las emociones que muestren. Conceptualicen los tres valores.
6. ¿Qué dilemas o problemas éticos --tales como el Problema del Tranvía, la Tragedia de los Comunes, el Problema de las Muchas Manos, Dilema de Collingridge-- se podrían presentar al desarrollar e implantar la innovación? Expliquen bien cómo se presentan o porqué consideran que no se presentan. ¿Cómo podría la tecnología contribuir a resolver, diluir o desplazar esos dilemas o problemas?
7. ¿Se presentan conflictos entre valores? Proponer una investigación empírica en el sentido del enfoque de Diseño Orientado a Valores.
8. ¿Qué posibles efectos, de primer, segundo y tercer orden pueden anticipar que tenga la innovación? ¿Consideran que se deberán tomar decisiones bajo riesgo, bajo incertidumbre y/o bajo ignorancia? ¿Puedes prever algún desarrollo tecnológico adicional de la innovación?
9. ¿Cómo toman en cuenta la posible existencia de riesgos desconocidos que pudieran presentarse en el desarrollo e implantación de la innovación? (Aplicación del Principio de Precaución).
10. Proponer una jerarquía de valores, normas y requerimientos, identificando los requerimientos éticos y sociales para el proyecto.
11. ¿A cuál(es) de las dimensiones de la sostenibilidad – ambiental, individual, social, económica, técnica – y en que grado de importancia relativa podría contribuir la innovación?
12. ¿Consideran que las respuestas anteriores cubren lo necesario para que la innovación pueda considerarse que cubre los requerimientos de una Innovación Responsable en Venezuela o Latinoamérica? ¿Y el enfoque de Diseño Basado en Valores? En caso que no pueda ser considerada una Innovación Responsable, propongan, si lo consideran factible, un cambio para que, a su juicio, satisfaga mejor los criterios necesarios de una Innovación Responsable.
13. Con base en sus respuestas, ¿cambiarían tu respuesta de la tarea anterior respecto a en cuál de los dos países escogidos cree el equipo que podría tener mayor impacto positivo la innovación escogida?
14. Referencias (tienen que haberse citado en al menos una de las secciones anteriores).

La evaluación final del proyecto tomará en cuenta la calidad del informe, la honestidad al citar ideas tomados de otros, y cómo toman en cuenta la retroalimentación del profesor.

VI. Resultados del Curso

El curso se dictó a siete estudiantes de Ingeniería de Computación en el trimestre Enero-Marzo 2021. Uno de ellos retiró el curso.

Los resultados detallados de las dos encuestas puede encontrarse en un reporte interno [20]. En líneas generales, los estudiantes consideraron que alcanzaron un nivel entre satisfactorio y muy satisfactorio en el dominio de las competencias y los contenidos del curso y consideraron el nivel y el impacto del curso en su futuro profesional entre positivo y muy positivo. Sin embargo, dos estudiantes sintieron haber adquirido poca competencia en lo que se refiere a identificar y conceptualizar valores, antivalores y riesgos (particularmente antivalores). En cuánto a los temas y tópicos, en líneas generales todos los estudiantes consideraron que su cobertura fue justo, con la excepción de la cobertura sobre los TIC verdes y los Grandes Desafíos de la Ingeniería que tres y dos de los siete estudiantes consideraron insuficiente respectivamente y los Objetivos de Desarrollo Sustentable que dos estudiantes consideraron exagerado.

Tanto los estudiantes como el profesor estuvieron de acuerdo con que el curso resultó demasiado largo (se llevó 14 semanas y no 12 como estaba planificado) por lo que algunos tópicos deben recortarse, particularmente los más alejados de la Computación y el entorno venezolano como lo es la seguridad de los reactores nucleares. De hecho sólo dos estudiantes estuvieron de acuerdo con que el balance entre ejemplos y casos de estudio tomados del área de Computación y los tomados de otras áreas estuvo bien.

En cuanto a la estrategia pedagógica debe destacarse el aspecto asíncrono del curso, el trabajo en equipos variables y múltiples ciclos de retroalimentación en las tareas tipo informes compartidos con el profesor y a veces, adicionalmente, con pares evaluadores. En general los estudiantes expresaron su desideratum de agregar más sincronicidad al curso, hubieron opiniones divididas sobre las bondades de variar tanto la membrecía en equipos y apreciaron el esfuerzo y calidad de la retroalimentación dada.

Los estudiantes consideraron al curso interesante, actual y formativo y lo calificaron como exigente y exitoso.

VII. Conclusiones y Recomendaciones

Esta experiencia muestra cómo puede diseñarse un curso centrado en ética aplicada a la Tecnología en una institución que carezca de profesores expertos en el área de Ciberética, montándolo sobre un MOOC. Se puede seguir una metodología como la aplicada en esta experiencia:

1. Identificar los profesores con experiencia o interés por el área, así como los tópicos de ciberética y sociedad manejados en los distintos cursos existentes en la institución.
2. Consultar con tales profesores e invitarlos a formar un equipo de trabajo para trabajar sobre las recomendaciones del ACM y la IEEE Computer Society como punto de partida al esfuerzo.
3. Llevar a cabo una extensa revisión bibliográfica, la preocupación por el tema ha sido bien documentada en la literatura.

4. Analizar la posibilidad de aprovechar material ya desarrollado para tratar los temas incluyendo los MOOC. Seleccionar cuidadosamente los tópicos y ejemplos a incluir, teniendo en cuenta que el área está en plena efervescencia y crecimiento.
5. Diseñar el curso para pocos estudiantes y en forma experimental, evitando en la medida de lo posible clases magistrales y quices, buscando mecanismos que permitan la reflexión, la discusión y mucha retroalimentación sobre los escritos presentados por los estudiantes y tomando en cuenta las limitaciones de acceso a Internet que puedan tener ellos y sus profesores.
6. Diseñar y analizar cuidadosamente encuestas de entrada y salida para el curso que permitan evaluar los aciertos, desaciertos y mejoras posibles del curso.

Hay suficiente información publicada en la literatura para dar luces sobre las competencias, tópicos, estrategias pedagógicas y estrategias de evaluación para tal diseño. Sin embargo puede que se requiera adaptar el MOOC a las características de co-aprendizaje tanto de los estudiantes a tomar el curso como del profesor que lo dictaría.

En el caso de elaborar un curso para estudiantes de Computación o Ingeniería de Software se recomienda sobre todo enfocar el curso a la elaboración y análisis de requerimientos éticos en el proceso de diseño de software. Adicionalmente se recomienda incorporar algunos tópicos y discusiones en otros cursos de la carrera primero para luego crear un curso dedicado al tema, sin apuntar inicialmente a cubrir todos los tópicos recomendados como obligatorios en el Curriculum para Ciencias de la Computación [1].

Finalmente, debe destacarse que el MOOC de la Universidad de Delft sobre el que se construyó el curso [14] resulta un excelente punto de partida para el diseño de un curso en Ciberética y Sociedad.

AGRADECIMIENTOS

Mis agradecimientos a los profesores Rafael Tomás Caldera (Universidad Monteávila), Raúl González Fabre (Universidad de Comillas) y Nelson Tepedino (Universidad Simón Bolívar) por su apoyo y acertadas recomendaciones en mis temerarias incursiones al área de la Ética, a la Coordinación de Computación de la Universidad Simón Bolívar por permitirme crear y dictar la nueva asignatura, a los investigadores Christoph Becker y Birgit Penzenstadler de la Universidad de Toronto por sus sugerencias y estímulo, al Prof. Joost Groot Kormelink y la Universidad de Delft por el permiso de aprovechar el material desarrollado para el MOOC sobre el cual se basó el curso, a los evaluadores anónimos y el editor de la Revista Venezolana de Computación por sus valiosa recomendaciones, tanto de fondo como de forma para este artículo. Finalmente agradezco de manera especial a los estudiantes que tomaron el curso y al profesor Alfonso Reinoza (Universidad Simón Bolívar) quien tomó el curso como oyente por su acertadas observaciones.

REFERENCIAS

[1] Joint Task Force on Computing Curricula Association for Computing Machinery(ACM) and IEEE Computer Society: *Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*, ACM and IEEE Computer Society, December 20 2013. ISBN: 978-1-4503-2309-3, DOI: 10.1145/2534860, http://dx.doi.org/10.1145/2534860

[2] C. Fiesler. *What Our Tech Ethics Crisis Says About the State of Computer Science Education*, .How we get to next., December 5 2018. https://howwegettonext.com/what-our-tech-ethics-crisis-says-about-the-state-of-computer-science-education-a6a5544e1da6

[3] S. Jones. *Doing the Right Thing: Computer Ethics Pedagogy Revisited*, Journal of Information, Communication and Ethics in Society, 14 (1). pp. 33-48. 2016. ISSN 1477-996X. http://eprints.mdx.ac.uk/19166/ (Preprint).

[4] C. D. Martin, C. Huff, D. Gotterbarn, K. Miller. *Implementing a Tenth Strand in the CS Curriculum*, Comm. ACM, December 1996.

[5] E. Burton, J. Goldsmith, and N. Mattei. *How to Teach Computer Ethics through Science Fiction*. Commun. ACM 61, 8 (2018), 54-64. https://doi.org/10.1145/3154485

[6] R. L. Canosa, J. M. Lucas. *Mock Trials and Role-playing in Computer Ethics Courses*, in Proceedings of the ACM SIGCSE Technical Symposium on Computer Science Education. 148-152. https://doi.org/10.1145/1352322.1352187

[7] B. Moskal, K. Miller, and L. A. Smith King. *Grading Essays in Computer Ethics: Rubrics Considered Helpful*, SIGCSE '02: Proceedings of the 33rd SIGCSE Technical Symposium on Computer Science Education, February 2002, pp. 101-105 https://doi.org/10.1145/563340.563380

[8] C. Fiesler, N. Garrett, and N. Beard. *What Do We Teach When We Teach Tech Ethics? A Syllabi Analysis*, ACM SIGCSE '20, March 11-14, 2020.

[9] B. Penzenstadler, S. Betz, C. C. Venters, R. Chitchyan, J. Porras, N. Seyff, L. Duboc, and C. Becker. *Everything is Interrelated: Teaching Software Engineering for Sustainability*, in Proceedings of the 40th International Conference on Software Engineering June, 2018.

[10] Infusing Ethics Selection Committee, Center for Engineering Ethics and Society, National Academy of Engineering. *Infusing Ethics Into the Development of Engineers: Exemplary Education Activities and Programs*, National Academies Press 2016.

[11] D. Johnson. *Engineering Ethics: Contemporary and Enduring Debates*, Yale University Press 2020.

[12] F. F. Benya, C. H. Fletcher, and R. D. Hollander. *Practical Guidance on Science and Engineering Ethics Education for Instructors and Administrators*: Papers and Summary from a Workshop December 12, 2012. National Academy of Engineering, National Academies Press 2012. http://nap.edu/18519

[13] Universidad Simón Bolívar, *Patrimonio Ético*, 2015, https://equinoccio.com.ve/index.php/colecciones/alma-mater

[14] Universidad de Delft. *Responsible Innovation: Ethics, Safety and Technology*, edX MOOC 2019. https://www.edx.org/course/responsible-innovation-ethics-safety-and-technology

[15] S. Spiekermann. *Ethical IT Innovation: A Value-Based System Design Approach*, CRC Press 2016.

[16] B. Friedman, P. H. Kahn Jr., and A. Borning. *Value Sensitive Design and Information Systems*, in P. Zhang & D. Galletta (Eds.), "Human-Computer Interaction in Management Information Systems: Foundations", M. E. Sharpe 2006.

[17] P. Vermaas. *Opening*, 178th Dies Natalis Symposium: Design for Values, 10th January 2020. TU Delft. https://collegerama.tudelft.nl/Mediasite/Play/6067b450e77c4466b2e5c0dddfce4a3f1d

[18] J. Groot Kormelink. *Responsible Innovation: Ethics and Risks of New Technologies*, 2nd Edition. TU Delft 2019. https://doi.org/10.5074/t.2019.006

[19] A. Teruel. *Cambios Hechos al MOOC de la Universidad de Delft para Llegar al Curso EP5008 en la USB*. Reporte interno, 2021.

[20] A. Teruel. *Un Experimento Curricular en Ciberética y Sociedad: Evaluación de un Curso de Innovación Responsable en Tiempos de Pandemia*. Reporte interno, 2021.

[21] Universidad Simón Bolívar. *Reglamento General de la Universidad Nacional Experimental Simón Bolívar*, Gaceta Oficial de la República Bolivariana de Venezuela, Año CXXVIII - Mes VII. Caracas: Viernes 27 de abril de 2001, Número 37.186.

[22] Decanato de Estudios Profesionales, Decanato de Estudios Tecnológicos, Decanato de Estudios Generales. *Catálogo Integrado de Estudios*, Universidad Simón Bolívar 2016.

[23] C. Álvarez. *Análisis y Estructuración de los Estudios Generales del Ciclo Profesional (1996-1997)*, Universidad Simón Bolívar 1997.

[24] Universidad Simón Bolívar. *Reglamento de Formación Complementaria y Servicio Comunitario de los Estudiantes de Pregrado de la Universidad Simón Bolívar*, Universidad Simón Bolívar, 6 de Julio 2016.

APÉNDICE A: LOS ESTUDIOS GENERALES EN LA UNIVERSIDAD SINÓN BOLÍVAR

Las carreras de pregrado de cinco años en la Universidad Simón Bolívar contienen entre 195 y 220 créditos y contienen un ciclo básico de un año y un ciclo profesional de cuatro años. Todo profesional graduado de la USB requiere cursar 18 créditos (6 asignaturas) en asignaturas de Estudios Generales durante su ciclo profesional. El Reglamento General de la Universidad Simón Bolívar [21] establece que:

> Artículo 33. Los programas de las carreras estarán integrados por asignaturas específicas de la correspondiente disciplina. Los estudios generales tienen por finalidad poner al estudiante en contacto con las diferentes áreas del saber, desarrollar habilidades y destrezas para el estudio y la indagación intelectual, así como contribuir a la formación de profesionales dotados de una amplia visión cultural y crítica […]

Según el Catálogo de Estudios de la Universidad Simón Bolívar elaborado conjuntamente por el Decanato de Estudios Profesionales, el Decanato de Estudios Tecnológicos y el Decanato de Estudios Generales [22]:

> "El Programa de Estudios Generales del Ciclo Profesional está orientado hacia la formación integral de los estudiantes de la Universidad Simón Bolívar. Está conformado por un conjunto de asignaturas de carácter electivo que se cursan a lo largo del Ciclo Profesional de cada carrera. El programa permite relacionar los diferentes campos del saber científico y humanístico en busca de una formación personal y profesional integral, contextualizada y pertinente a la realidad nacional e internacional.
>
> Los objetivos de los Estudios Generales en el Ciclo Profesional son los siguientes:
>
> - Contribuir a desarrollar en el estudiante una visión crítica y actualizada acerca de la cultura nacional, latinoamericana, occidental y universal.
> - Contribuir a desarrollar en el estudiante una actitud positiva hacia la interdisciplinariedad del conocimiento científico y humanístico en función de la adquisición de una personalidad intelectual analítica.
> - Familiarizar al estudiante con los enfoques conceptuales y metodológicos de las ciencias sociales y humanísticas en función del estudio de los problemas y las necesidades del hombre contemporáneo.
> - Ofrecer al estudiante un conjunto de experiencias orientadas a favorecer el desarrollo de sus capacidades para la autogestión y autorrealización."

En la Universidad Simón Bolívar se oferta una amplia variedad de Estudios Generales en áreas tan diversas como Ciencias Sociales, Literatura, Arte, Filosofía, Economía, Historia, Psicología, Derecho Biología, Urbanismo y Ambiente. El estudiante es libre de escoger las asignaturas de Estudios Generales a cursar a lo largo de su carrera. Si bien los Estudios Generales proporcionan una formación en aspectos básicos culturales, sociales y éticos, no necesariamente proporciona aspectos "...inherentes a la disciplina de la Computación".

En su momento, en una reflexión crítica sobre los Estudios Generales, el Decano de Estudios Generales, el Prof. Cristian Álvarez [23] puntualizó que:

> "... la intención de los Estudios Generales es la de propiciar en el educando una voluntad de cultura, un deseo de cultivar sus facultades, una actitud que busque incesantemente el crecimiento intelectual y espiritual, tratando de comprender la realidad a la que pertenece. Aprender a conocer la realidad, lo que incluye la propia condición humana, su relación con el entorno y los métodos y resultados del saber que cultiva; aprender a asumir las responsabilidades que implican los alcances y limitaciones de ese pensar; recordar, recobrar la reflexión sobre lo humano y sus saberes a partir de los temas y áreas del conocimiento constituyen el fin de los cursos de Estudios Generales."

El estudiante también recibe una formación complementaria [24]:

> Artículo 4: La Formación Complementaria tiene como principal objetivo contribuir a la Formación Integral del estudiante, promoviendo el desarrollo de actitudes, valores y conocimientos que propendan a sensibilizar al estudiante frente a la realidad del país y el reforzamiento de valores de ética, solidaridad, participación, respeto y estima de la esfera de lo colectivo.
>
> Artículo 9: Todos los estudiantes deben inscribir y aprobar, al menos, un curso reconocido por la Universidad que plantee aspectos de la problemática social, económica o ambiental del país en general o de varias o alguna de sus comunidades en particular. Los programas del Ciclo Básico y de Formación General incluirán contenidos sobre ética, capital social, responsabilidad social y valores ciudadanos [...]

APÉNDICE B: LA PRIMERA PROPUESTA – CIBERÉTICA Y SOCIEDAD

A. Número Total de Horas Semanales

Teóricas: 3 Prácticas: 0 Laboratorio: 0 Créditos: 3

B. Objetivo General

Explorar la dimensión ética y social de la Computación y desarrollar un sentido de la importancia de estos temas en el desarrollo profesional y personal.

C. Objetivos Específicos

1. Formular y defender una posición respecto a un aspecto ético relacionado con la tecnología informática.
2. Distinguir elementos utilitarios o deontológicos en argumentos ciberéticos.
3. Describir al menos cinco desafíos éticos relacionados con la computación.

4. Describir los resultados de una discusión sobre aspectos éticos como una posición de consenso o diferencias aceptables de opinión.
5. Demostrar el dominio de métodos efectivos verbales y escritos para explicar una posición relativa a las responsabilidades sociales de desarrolladores de software y otros profesionales de TI.
6. Demostrar conocimiento básico de al menos dos objetivos de desarrollo sostenible y el potencial, positivo o negativo de la Informática para alcanzarlos.
7. Ser capaz de trabajar en equipo para presentar, en forma crítica y consensuada, una acción relacionada con la Computación con impacto social o ambiental.
8. Identificar posibles elementos del Decálogo de Valores de la USB relacionados con la Ciberética.
9. Presentar y defender opiniones personales sobre un dilema o un aspecto polémico de la ciberético.
10. Analizar un curso de acción en el contexto de varias culturas, comunidades y países.

D. Contenido Programático

Tema I: Fundamentos de la ciberética

- Teorías éticas (Utilitarismo, Deontología, Ética basada en virtudes) y la toma de decisiones.
- Suposiciones y valores morales. La Teoría de Fundamentos Morales (Moral Foundation Theory).
- Argumentos éticos, incluyendo la noción de desacuerdos constructivos.
- Repaso de códigos éticos (ACM, IFIP, Project Management Institute, Colegio de Ingenieros de Venezuela) pertinentes al ejercicio profesional de la Ingeniería de la Computación.

Tema II: Acceso a la computación

- Accesabilidad: el punto de vista ético y el punto de vista legal.
- La brecha digital: Diferencias en el acceso a la tecnología digital y sus posibles consecuencias en la igualdad de género, clase o nivel social, etnicidad, países y regiones.
- Crecimiento y control de Internet.

Tema III: Privacidad y libertades civiles

- Fundamentos filosóficos del derecho a la privacidad.
- Fundamentos legales de la protección a la privacidad.
- Implicaciones para la privacidad de la recopilación generalizada de datos para alimentar bases y repositorios de datos, sistemas de vigilancia y computación en la nube.
- Consecuencias de privacidad diferencial.
- Ciber-acoso y discriminación.
- Políticas aceptables de uso informático en el lugar de trabajo.
- Soluciones basadas en tecnología para la protección de la privacidad. Propiedades y características de una aplicación que pueden contribuir a proteger la privacidad.
- Libertad de expresión y el Internet.

Tema IV: Ciberética y desarrollo sustentable

- El profesional de la Computación y las políticas públicas.
- Análisis de factores ambientales y culturales que pueden contribuir a la sostenibilidad de una solución informática (p. ej. políticas organizacionales, viabilidad económica, consumo de recursos).
- Impactos globales sociales y ambientales en el uso y eliminación de dispositivos y suministros informáticos.
- Impactos ambientales en la selección de diseños en áreas como algoritmos, sistemas de operación, redes o interacción personas-computadoras.
- Análisis de posibles consecuencias.
- Disenso ético y denuncias.

Tema V: Ciberética y cultura

- Implicaciones sociales de la Computación en un mundo con Redes de Computadoras.
- Posibles efectos de los medios sociales en el individualismo, el colectivismo y la cultura.
- Computación ubicua y consciente de contexto.
- La relación entre culturas regionales y dilemas éticos.

Tema VI: Propiedad intelectual

- Fundamentos filosóficos de la propiedad intelectual.
- Derechos de propiedad intelectual.
- Propiedad intelectual digital intangible.
- Fundamentos legales para la protección de propiedad intelectual.
- Gestión de derechos digitales.
- Copyrights, patentes, secretos industriales, marcas comerciales.
- Plagio.

Tema VII: Uso de herramientas colaborativas

E. Estrategias Metodológicas

No se considera factible cubrir todos los temas mencionados previamente en detalle por lo que se procurará cubrir al menos un tópico dentro de cada tema. Preferiblemente los tópicos a abordar de todos los temas menos el primero serán de mutuo acuerdo entre el profesor y los estudiantes, hasta que se haya obtenido suficiente experiencia como para determinar qué tópicos deben cubrirse en este curso y cuáles pueden incorporarse a otros cursos. El nivel de detalle de cobertura del primer tema (Fundamentos de la Ciberética) será determinado por el profesor de la asignatura.

Se combinarán clases magistrales, discusiones en clase, casos de estudio, reportes escritos y la presentación de tópicos y/o casos preparados en equipo. Para ello se explorarán las estrategias metodológicas sugeridas en en el reporte “Infusing Ethics into the Development of Engineers” (National Academy of Engineering, 2016) y en http://www.onlineethics.org.

F. Estrategias de Evaluación

Se explorarán varias estrategias de evaluación con entre tres y seis evaluaciones en el curso, prestando particular atención a las estrategias reportadas en el reporte “Infusing Ethics into the Development of Engineers” (National Academy of Engineering, 2016).

G. Fuentes de Información

- C. Fiesler, What Our Tech Ethics Crisis Says About the State of Computer Science Education, December 2018. https://howwegettonext.com/what-our-tech-ethics-crisis-says-about-the-state-of-computer-science-education-a6a5544e1da6 [Excelente artículo para motivar el curso]
- L. C. Kaczmarczyk, *Computers and Society: Computing for Good*, Chapman & Hall/CRC Textbooks in Computing, 2011.
- https://openmindplatform.org/use-openmind
- http://www.onlineethics.org
- T. Bynum, *Computer and Information Ethics*, The Stanford Encyclopedia of Philosophy (Summer 2018 Edition), https://plato.stanford.edu/entries/ethics-computer
- H. Abelson, K. Ledeen, and H. Lewis, *Blown to Bits: Your Life, Liberty, and Happiness After the Digital Explosion*, Addison-Wesley, 2008. http://www.bitsbook.com
- H. T. Tavani, *Ethics and Technology: Controversies, Questions, and Strategies for Ethical Computing*, 4th edition. Wiley, 2013.
- G. W. Reynolds, *Ethics in Information Technology*, 5th edition. Cengage, 2015.
- M. J. Quinn, *Ethics for the Information Age*, 6th edition. Pearson, 2015.
- K. E. Himma and H. T. Tavani, *The Handbook of Information and Computer Ethics*, Wiley, 2008.
- R. Schultz, *Contemporary Issues in Ethics and Information Technology*, IRM Press, 2006.
- H. Nissenbaum, *Privacidad Amenazada: Tecnología, Política y la Integridad de la Vida Social*, Oceano, 2011.
- R. A. Schultz, *Information Technology and the Ethics of Globalization: Transnational Issues and Implications*, Information Science Reference, 2010.
- A. Adam, *Gender, Ethics, and Information Technology*, Palgrave, 2005.
- I. Zorn, S. Maass, E. Rommes, C. Schirmer, H. Schelhowe, *Gender Designs IT: Construction and Deconstruction of Information Society Technology*, VS Verlag für Sozialwissenschaften, 2007.
- Estudiantes de la Maestría en Informática de Negocios de la Universidad de Viena: *Mind your Data: Your Guide to Regain Privacy and Control!*, Enero 2018. https://www.wu.ac.at/fileadmin/wu/d/i/ec/Privacy_Broschüre/Brochure_2702.pdf
- P. Lin, K. Abney, and G. A. Bekey, *Robot Ethics 2.0: From Autonomous Cars to Artificial Intelligence*, Oxford University Press, 2017.
- Organización de las Naciones Unidas, *Los Objetivos de Desarrollo Sostenible*, 2018. http://www.undp.org/content/undp/es/home
- United Nations Statistics Division Statistical Services Branch, *The Sustainable Development Goals Report*
- National Academy of Engineering, *The Grand Challenges of Engineering*, 2017. http://www.engineeringchallenges.org
- B. E. Drushel and K. German, *The Ethics of Emerging Media: Information, Social Norms, and New Media Technology*, The Continuum International Publishing Group, 2011.
- J. Sullins, *Information Technology and Moral Values*, The Stanford Encyclopedia of Philosophy (Spring 2016 Edition), https://plato.stanford.edu/archives/spr2016/entries/it-moral-values
- H. Tavani, *Search Engines and Ethics*, The Stanford Encyclopedia of Philosophy (Fall 2016 Edition), https://plato.stanford.edu/archives/fall2016/entries/ethics-search
- C. Anderson (Rapporteur), *Overcoming Challenges to Infusing Ethics into the Development of Engineers: Proceedings of a Workshop*, The National Academies Press, 2017.
- D. J. Gunkel, *The Machine Question: Critical Perspectives on AI, Robots, and Ethics*, MIT Press, 2012.
- S. L. Edgar, *Morality and Machines: Perspectives on Computer Ethics*, Jones and Bartlett, 1997.

APÉNDICE C: LA SEGUNDA PROPUESTA – UNA ASIGNATURE SOBRE INNOVACIÓN RESPONSABLE

A. Número Total de Horas Semanales

Teóricas: 3 Prácticas: 0 Laboratorio: 0 Créditos: 3

B. Objetivo General

Motivar y facilitarle al estudiante la adquisición de competencias básicas necesarias para aplicar, personal y profesionalmente, el enfoque de Innovación Responsable. Tal enfoque está fundamentado en valores éticos que ayudan a desarrollar soluciones socio-técnicas a problemas regionales y globales. El corazón del enfoque lo constituye el análisis, la reflexión y el debate público sobre los principios éticos, los riesgos y la aceptabilidad moral de nuevas tecnologías emergentes. El curso hará especial hincapié en tecnologías asociadas a la Computación.

C. Objetivos Específicos

1. Formular preguntas de carácter ético con respecto a tecnología (nueva o emergente).
2. Identificar y conceptualizar al menos tres posibles valores éticos asociados al uso o a la propuesta de desarrollo de una tecnología, tres antivalores y tres riesgos.
3. Incluir requerimientos éticos al levantar requerimientos no funcionales.
4. Explicar en sus propias palabras los aspectos más característicos y el rol de la Ética en la Innovación Responsable.
5. Identificar posibles elementos del Decálogo de Valores de la USB relacionados con la Innovación Responsable.
6. Describir cómo la tecnología puede contribuir a resolver, diluir o desplazar algunos dilemas éticos.
7. Distinguir entre diversos tipos de innovaciones (radical, de nicho, incremental, frugal) y conocer las condiciones de éxito para cada tipo.
8. Tomar en cuenta la posible existencia de riesgos desconocidos (incertidumbre profunda) de tecnologías nuevas.

9. Analizar riesgos asociados a tecnologías, tanto retroactivamente ("backward looking") como proactivamente ("forward looking").
10. Presentar y defender reflexiones críticas sobre tecnologías nuevas o emergentes desde un punto de vista ético y de riesgos (casos de estudio).
11. Aplicar los conceptos de Innovación Responsable y Diseño Sensitivo a Valores (VSD) a casos sencillos y reseñar sus posibles consecuencias para el proceso de diseño.
12. Analizar una innovación tecnológica, preferiblemente asociada a la Computación, propuesta para resolver algún aspecto de los Objetivos de Desarrollo Sostenible (ODS) de la UNESCO o de los Grandes Desafíos de la Ingeniería de la National Academy of Engineering, aplicable a Venezuela o Latinoamérica para determinar si constituye una innovación responsable y en caso contrario proponer, de considerarlo factible, un cambio para que satisfaga mejor los criterios necesarios de una innovación responsable.

D. Contenido Programático

Tema I: Innovación responsable como aplicación de la ética

- Qué es la Innovación Responsable
- El Problema del Tranvía desde el punto de vista del ingeniero
- Responsabilidad individual y colectiva
- El Problema de las Muchas Manos: ¿quién es el responsable?

Tema II: Dilemas morales. Razón y Emociones. Institutos y valores

- El Dilema de la Sobrecarga Moral
- Razón, emociones y valores en la toma de decisiones
- Caso de estudio: Responsabilidad en la Robótica
- Sustentabilidad, ética y tecnología: la Tragedia de los Bienes Comunes

Tema III: Comprensión e identificación de riesgos

- Evaluación Tecnológica y la Innovación Responsable
- Gestión de incertidumbre e ignorancia
- Caso de estudio: Energía nuclear
- Caso de estudio: Inteligencia Artificial
- Enfoques para la evaluación del impacto de tecnologías

Tema IV: Análisis de riesgos e Ingeniería de Seguridad ("Safety Engineering")

- Gestión de riesgos: Análisis de costo-beneficio
- Gestión de riesgos: El compromiso entre riesgo y seguridad
- Evaluación cuantitativa avanzada de riesgos
- Caso de estudio: Vehículos autónomos
- Caso de estudio: Ciberseguridad

Tema V: ¿Cómo surgen las innovaciones?

- Innovación radical.
- Innovación incremental.
- Caso de estudio: Refrigerantes
- Factores que inciden en la innovación
- Caso de estudio: Innovación frugal

Tema VI: El diseño orientado a valores

- Introducción
- Aplicación
- Caso de estudio: Armas autónomas, ¿quién hala el gatillo?
- Los Objetivos de Desarrollo Sustentable (ODS)

E. Estrategias Metodológicas, Didácticas o de Desarrollo de la Asignatura

Este curso plantea una serie de retos importantes, por cuanto:

1. Requiere fortalecer o desarrollar competencias cognoscitivas, afectivas e interpersonales. De hecho coloca en el centro de la atención a la Ética, una disciplina poco aplicada en la carrera más allá de lo que representan las normas que aplican qua estudiante. Esto requiere un cambio de actitud y el manejo de argumentos "blandos" o "humanistas" muy distintos a los argumentos de carácter algorítmico tradicionalmente enfatizados en la carrera.
2. Plantea como objetivo construir el curso fuertemente fundamentado en tecnologías digitales disponibles (TDD) de forma que pueda ser cursado a distancia en caso que las circunstancias así lo obliguen.
3. Debe destacarse la escasez de profesores especialistas en el tema y la dificultad a la fecha (Junio 2020) de conformar un equipo interdisciplinario de profesores dispuestos o con la disponibilidad necesaria para abordar estos importantes retos. Se espera que la existencia y promoción de la asignatura pueda incluso servir de estímulo para la creación de tal equipo.

Por ende se ha decidido basar la asignatura en un curso masivo abierto en línea (MOOC) desarrollado por especialistas de la Universidad de Delft. De esta manera se espera atacar (2), (3) y, parcialmente (1). El MOOC usa las estrategias características de su plataforma, a saber, videos cortos de entre 10 y veinte minutos para presentar el material clave, quices sencillos (de verdadero/falso y de múltiples opciones) con retroalimentación inmediata para chequear la comprensión básica del material dado, participación en foros en lo que se espera que los estudiantes hagan preguntas críticas y respondan las preguntas de otros, la inclusión de lecturas recomendadas adicionales, el seguimiento detallado al avance del estudiante por los elementos del curso y un trabajo final que es evaluado por pares. El MOOC además incluye acceso a un libro de texto y casos de estudio. Protagonizan los videos, especialistas de nivel internacional que conforman un verdadero equipo interdisciplinario en el área de la Facultad de Tecnología, Política y Gestión de esa universidad. Sin embargo la evaluación asociada al MOOC es superficial y a juicio del proponente de la asignatura de la USB, no proporciona el nivel de retroalimentación necesario para alcanzar un buen dominio del tema. Adicionalmente los estudiantes de la carrera pueden tener dificultades con algunos ejemplos o casos de estudio para los que están insuficientemente preparados o que se alejan de la tecnología asociada con la Computación. Por tales motivos se incluyen sesiones adicionales, presenciales o virtuales, de debate, revisión y aclaración de conceptos clave, cerradas al grupo de estudiantes inscritos. Adicionalmente el trabajo final deberá elaborarse y presentarse en equipos de 2-3 estudiantes

permitiendo así mayor sinergia, conservando una evaluación por pares pero agregando una evaluación profesoral de los mismos. Se reorienta y profundiza las características del trabajo final hacia un área de mayor interés y aplicabilidad al medio venezolano y latinoamericano como lo son los ODS. Se aspira invitar otros profesores a sesiones especiales de discusión o a participar en calidad de jurado profesoral. Se estima que el conjunto de exigencias llevan al MOOC a un esfuerzo cónsono con una asignatura trimestral de 3 créditos.

El encapsulamiento de un MOOC dentro del curso lleva el curso a tener características muy similares a un "aula invertida" (en Inglés, "flipped classroom"). – el estudiante debe cubrir por su cuenta el material del MOOC antes de participar en las sesiones adicionales.

Es conveniente aclarar el uso que se hará de tecnologías digitales disponibles. Se espera utilizar una plataforma de apoyo al aprendizaje como Classroom para coordinar el curso y servir de repositorio de objetos de aprendizaje adicionales al MOOC. Las sesiones de debate o aclaración se harán utilizando una herramienta como Google Group, que tiene la ventaja de permitir mayor asincronicidad o, en caso que se amerite mayor interactividad WhatsApp, Zoom, YouTube o similar – dependiendo de la calidad del servicio de Internet y los requerimiento de registro. Las encuestas, y algunas de las evaluaciones por tema, se apoyarán en Google Forms, Survey Monkey o similares. El trabajo final deberá hacerse en GoogleDocs y compartirse con el profesor y los pares evaluadores.

En la medida de lo posible, se explorará la posibilidad de permitirle al estudiante avanzar a su propio ritmo, particularmente en lo que se refiere a los tópicos I-VI. De esta manera un estudiante puede seguir el cronograma al pie de la letra, mientras que otro cubre cada grupo de tres tópicos en dos semanas, dejándose la tercera semana libre para atender otras asignaturas.

Finalmente debe advertirse que este curso requiere, por parte del estudiante, el dominio del Inglés ya que a la fecha (Junio 2020), los videos no incluyen subtítulos en Castellano y las lecturas, en su gran mayoría, no han sido traducidas al Castellano.

F. Estrategias de Evaluación

Al culminar cada uno de los tópicos I-VI, cada estudiante deberá contestar mediante un formulario Google o similar, una serie de preguntas cortas donde demuestre su dominio de los conceptos cubiertos. Adicionalmente participará en un foro/debate presencial o virtual (Google group o similar, en el caso virtual) muy similar en estilo a los foros del MOOC sobre el que se apoya el curso pero donde se prestará mayor cuidado a la retroalimentación a los aportes al foro. Las respuestas y aportes por tópico forman el 65% del peso de la evaluación del curso, y el peso de cada tópico no será menor al 10% del valor del curso.

El 35% restante de la evaluación del curso corresponde a un trabajo final. El trabajo final se elabora por un equipo de 2 a 3 estudiantes y se entrega por escrito en un documento compartido. El trabajo final será evaluado por un equipo de 2-3 estudiantes (pares) distinto al que escribió el trabajo. El equipo evaluador hace una presentación crítica del trabajo; el equipo autor tiene la oportunidad de defenderse ante las críticas o contestar preguntas sobre el trabajo. Del 35% de la nota dedicada al trabajo final, cada estudiante recibe 20% por su contribución al trabajo escrito y a la defensa de éste, y el 15% restante por su contribución a la presentación crítica de otro trabajo.

G. Fuentes de Información

- C. Fiesler, *What Our Tech Ethics Crisis Says About the State of Computer Science Education*, December 2018. https://howwegettonext.com/what-our-tech-ethics-crisis-says-about-the-state-of-computer-science-education-a6a5544e1da6 [Excelente artículo para motivar el curso]
- Universidad de Delft, *Responsible Innovation: Ethics, Safety and Technology*, https://www.edx.org/course/responsible-innovation-ethics-safety-and-technolog [El corazón del curso]
- European Commission, *Horizon 2020 Work Programme 2018-2020: 16. Science with and for Society*, European Commission Decision C(2020)1862 of 25 March 2020. https://ec.europa.eu/research/participants/data/ref/h2020/wp/2018-2020/main/h2020-wp1820-swfs_en.pdf
- Organización de las Naciones Unidas, *Los Objetivos de Desarrollo Sostenible*, 2018. http://www.undp.org/content/undp/es/home
- United Nations Statistics Division Statistical Services Branch, *The Sustainable Development Goals Report*
- National Academy of Engineering, *The Grand Challenges of Engineering*. 2017. http://www.engineeringchallenges.org
- National Academy of Engineering, *Online Ethics Center for Engineering and Science*. http://www.onlineethics.org
- J. Sullins, *Information Technology and Moral Values*, The Stanford Encyclopedia of Philosophy (Spring 2016 Edition), https://plato.stanford.edu/archives/spr2016/entries/it-moral-values [Introducción a un subconjunto reducido de valores éticos asociados a TI]
- S. Spiekermann. Ethical IT Innovation: A Value-Based System Design Approach, CRC Press 2016 [Texto complementario y de referencia para el curso]

Revista Venezolana de Computación
ISSN: 2244-7040
http://www.svc.net.ve/revecom
Vol. 8, No. 1, pp. 25-36, Junio 2021
Fecha recepción: 30/01/2021, Fecha aceptación: 25/07/2021

Extracción de Esqueleto por Contracción de Mallado con Asignación Automática de Pesos

Moisés E. Simón A. Berenguer Gómez, Francisco José Moreno Alvarez
moises.berenguer@ciens.ucv.ve, francisco.j.moreno@ucv.ve

Centro de Computación Gráfica, Universidad Central de Venezuela, Caracas, Venezuela

Resumen: En este trabajo se propone un método para realizar la extracción del esqueleto de un objeto tridimensional definido por una malla geométrica. La propuesta se basa en una investigación previa que utiliza el *suavizado laplaciano* para contraer un modelo hasta formar su esqueleto curvo. En esta, se evalúan un conjunto de modificaciones para incrementar su eficiencia. Los cambios planteados, se basan en pruebas aplicadas que determinaron la combinación óptima entre: la selección del método de resolución del sistema de ecuaciones que determina la contracción del mallado y, el algoritmo de permutaciones que se utiliza en conjunción a él. Debido a la naturaleza paralela de las fases que componen el algoritmo, la solución es desarrollada tanto en CPU como en GPU, utilizando la arquitectura propietaria de NVIDIA, CUDA. Además, se complementa el algoritmo original con un conjunto de etapas posteriores que adaptan el esqueleto resultante para su uso directo en aplicaciones de diseño 3D, permitiendo deformar y realizar animaciones al modelo original a partir del esqueleto generado. Finalmente, se compara la eficiencia del algoritmo propuesto contra el original, procesando objetos tridimensionales con distinto conteo de polígonos y características topológicas. Como resultado, la implementación desarrollada en CPU es 1.7 veces más rápida que el método original y la desarrollada en GPU es 5.4 veces más rápida, donde esta última exhibe un crecimiento lineal ante el incremento de los polígonos de la malla original, a diferencia del comportamiento exponencial de las soluciones en CPU.

Palabras Clave: Extracción de Esqueleto; Suavizado Laplaciano; Asignación de Pesos.

Abstract: In the current study a set of modifications to a previous mesh-based skeleton extraction technique, which compressed a mesh into its curved skeleton by applying a laplacian smoothing, are presented. All the suggested changes are supported by efficiency tests that determined the optimal combination between several linear equation solvers and their matrix permutation algorithms. The proposed method was developed and tested in CPU and GPU using the proprietary architecture of NVIDIA, CUDA. Also, a set of post-processing steps were included to enable its use as an animation tool in any 3D design suite. To prove the robustness of the algorithm, all tests were applied over a group of 3D models with an increasingly polygon count and topological complexity. As a result, the proposed method proves to be 1.7 times faster than the original in its CPU version and exhibits an exponential time growth as the number of processed polygons increases. Likewise, the GPU adaptation is 5.4 times faster and exhibits a linear-time growth.

Keywords: Skeleton Extraction; Laplacian Smoothing; Weight Assigments.

I. Introducción

Un esqueleto curvo es una representación simplificada de la geometría y topología de un objeto, su estructura es utilizada para navegar escenas al generar un camino libre de colisiones sobre la misma [1], propiedad que explota el área médica, para navegar virtualmente a través de los órganos del cuerpo humano, abarcando procedimientos tradicionalmente invasivos como: colonoscopia [2], broncoscopia [3], angioscopia [4], entre otros. También resulta de utilidad para segmentar información médica proveniente de resonancias magnéticas [5][6], detectar calcificaciones y aneurisma [7], alinear múltiples vistas de un objeto (utilizando el esqueleto como punto de referencia) [8][9], metamorfosis (transformar un objeto en otro) [10], animaciones [11][12], entre otros.

Dado el amplio abanico de casos de uso de los esqueletos curvos en distintas áreas de investigación, en este trabajo se plantea una alternativa para extraer el esqueleto curvo a partir de mallados geométricos. Se utiliza como base el método propuesto por Au et al. [13], al cual se le modifican algunas de sus etapas para incrementar su eficiencia. A partir de las modificaciones propuestas, se experimentó con múltiples técnicas en cada una de las etapas seleccionadas, lo que permitió evaluar el rendimiento de cada técnica bajo las mismas condiciones y seleccionar la combinación que demuestra el mejor rendimiento. Finalmente, se agregan las

fases necesarias al método optimizado, para que su resultado se pueda utilizar de forma transparente en cualquier software de diseño tridimensional, para animar y manipular (mediante el esqueleto) el mallado original.

El resto del trabajo se presenta en 5 secciones. La Sección II describe brevemente algunos trabajos relacionados a la propuesta que se presenta en la Sección III. La Sección IV plantea las pruebas realizadas al sistema para obtener la combinación de técnicas óptimas para generar el esqueleto curvo. Finalmente, la Sección V expone las conclusiones y trabajos futuros.

II. Trabajos Relacionados

Una forma u objeto normalmente es descrita por los límites que conforman su silueta. Blum [14], describe un objeto mediante un conjunto de discos ubicados en su interior, cada disco está alineado al eje central del objeto y abarca su volumen, la estructura que definen corresponde con el esqueleto o estructura central de un objeto.

A partir de la definición anterior, surgen múltiples técnicas para generar el esqueleto de un objeto de manera automática, en las que el esqueleto (de volumen aproximado a cero) de un modelo tridimensional, se puede calcular mediante un modelo matemático diseñado para este propósito [15]. El objeto tridimensional puede ser representado mediante: un mallado [1], una nube de puntos, una representación topográfica o volumétrica, entre otras. El proceso de extraer el esqueleto de un modelo tridimensional es denominado "esqueletonización".

Las diversas técnicas de esqueletonización difieren en aspectos teóricos como la definición del esqueleto y, en aspectos prácticos como: la discretización del espacio (vóxeles[2] vs mallas), los algoritmos de extracción de esqueleto y los parámetros que establecen. Cada uno de estos aspectos y la sensibilidad inherente de los esqueletos al ruido en la representación del objeto, hace que cada método produzca esqueletos diferentes para el mismo modelo base, lo que causa variaciones en el desempeño al utilizar el esqueleto generado y plantea un reto de investigación que se mantiene en la actualidad.

A. Métodos Volumétricos

El proceso de esqueletonización basado en volumen (*Voxel Based Methods*), extrae el esqueleto de un modelo tridimensional haciendo uso de voxelización. Voxelizar, involucra englobar todo el modelo en una caja contenedora o vóxel, que luego se subdivide iterativamente hasta alcanzar una representación volumétrica aproximada del objeto original. Posteriormente, se aplica un proceso (que varía entre cada autor) para conseguir la representación correspondiente al único camino de vóxeles que expone todas las características topológicas del modelo inicial. Según Bertrand y Aktouf [16], el camino resultante se denomina esqueleto curvo y solo describe la silueta del objeto al que pertenece, sin nodos o información adicional. Por ende, es necesario aplicar una última etapa en la que se agregan las características necesarias que adaptan el esqueleto a su caso de uso particular.

Los métodos de adelgazamiento y propagación de límites o *thinning*, se basan en el concepto de punto simple [17] y generan el esqueleto curvo removiendo capa a capa los vóxeles que conforman el objeto, hasta conseguir la representación más delgada posible.

Los métodos de adelgazamiento direccional solo eliminan los vóxeles de una dirección particular (norte, sur, arriba, abajo, este y oeste) en cada iteración, variando la cantidad de direcciones y condiciones para identificar los puntos finales [18]. Estos métodos son sensibles al orden en que se procesan las direcciones y, los esqueletos resultantes pueden no estar centrados dentro del modelo.

Los métodos de adelgazamiento secuenciales dividen el espacio discreto en varios subconjuntos denominados subcampos. En cada iteración, se selecciona uno de ellos y se eliminan los vóxeles pertenecientes a él. Se puede usar un número diferente de subcampos en tres dimensiones: 2, 4 y 8 [19]. Según Borgefors et al. [20], estos algoritmos consideran para su eliminación, a todos los puntos límite en una iteración de adelgazamiento, pero para mantener la topología, el vecindario debe ser inspeccionado para decidir si un vóxel debe eliminarse.

B. Métodos Geométricos

La esqueletonización basada en geometría (*Mesh Based Methods*) procesa directamente la malla que representa el objeto tridimensional. No requiere de una representación intermedia (vóxeles), a diferencia de la basada en volumen.

Au et al. [13] proponen un método que contrae la geometría aplicando iterativamente un suavizado laplaciano que elimina el ruido sobre la superficie y sus detalles desplazando los vértices en dirección a las normales. Si dicho proceso se aplica sin ningún tipo de restricción, los vértices del mallado se suavizarían progresivamente hasta converger en un solo punto, por lo cual se utilizan puntos de anclaje para restringir el desplazamiento de los vértices. Como resultado, se obtiene un mallado degenerado de volumen aproximado a cero, que es la contracción máxima que se puede realizar sobre el objeto sin perder sus características principales. Ya que el resultado final del algoritmo es el esqueleto curvo del mallado, debe ser complementado con etapas que agreguen las características necesarias para el caso de uso particular en que el que será empleado. Por ejemplo, si el esqueleto se utilizará en una animación, es necesario segmentarlo en articulaciones y extremidades.

En esta investigación, se modifica el método propuesto por Au et al. [13], alterando algunas de sus fases con la intención de mejorar su eficiencia. Principalmente, se analiza la forma de los sistemas de ecuación que se involucran en la etapa de

[1] Colección de triángulos y vértices que aproximan una superficie en tres dimensiones.

[2] Vóxel (*Volume Pixel*): Representación mínima de un elemento en una matriz en tres dimensiones. Equivalente 3D de un píxel en 2D.

contracción geométrica y se proponen alternativas para obtener las soluciones.

III. Propuesta

El algoritmo que se propone a continuación, construye el esqueleto[3] de un objeto a partir de la malla que representa su geometría. Además, el esqueleto se complementa con la información necesaria para ser utilizado en el contexto de animaciones por computadora. Para cumplir este objetivo, se descompone el procedimiento en las siguientes etapas:

1) Contracción de Geometría: Implementación y análisis a fondo del algoritmo planteado por Au et al. [13], con la finalidad de generar el esqueleto curvo partiendo de la geometría inicial.
2) Simplificación: Filtrado de las aristas del esqueleto generado en la etapa previa.
3) Refinamiento: Reubicación de cada nodo del esqueleto curvo al centro local correspondiente en el mallado.
4) Selección: Identificación de los huesos relevantes del modelo que puedan ser empleados como herramienta de animación del mallado original.
5) Asignación de Pesos: Cálculo del grado de influencia de los huesos sobre los vértices del mallado utilizando [22].
6) Exportado: Empaquetado del resultado de las etapas previas en un archivo formato FBX. Este formato define el estándar de representación geométrica soportado por cualquier programa de diseño tridimensional como: Blender, Maya o 3DS.

En la Figura 1 se pueden apreciar todos los pasos aplicados sobre uno de los modelos de prueba *Raptor*.

A. Contracción de Geometría

Inicialmente se contrae el mallado hasta obtener su esqueleto curvo aplicando iterativamente el suavizado laplaciano con restricciones descrito por Au et al. [13], el cual sugiere el uso del operador laplaciano definido en la ecuación (1).

$$L_{ij} = \begin{cases} \omega_{ij} = cot(\alpha_{ij}) + cot(\beta_{ij}) & (i,j) \in E \\ \sum_{(i,k)\in E}^{k} -w_{ik} & i = j \\ 0 & otro \end{cases} \tag{1}$$

Donde según Desbrun et al. [23] α_{ij} y β_{ij} son los ángulos opuestos correspondientes a la arista (i,j), siendo E el conjunto que representa las aristas de la malla.

Aplicar la ecuación de Laplace en repetidas ocasiones causará la pérdida progresiva de características del modelo inicial, para evitar este comportamiento se introducen dos variables adicionales a la ecuación: W_l y W_h que acentúan y relajan el grado de contracción respectivamente. W_l con dirección al operador laplaciano y W_h en dirección a la malla original. Con este planteamiento se define el sistema de ecuaciones (2) como modelo matemático de optimización, que permite contraer la geometría y preservar las características del modelo.

$$\begin{bmatrix} W_l L \\ W_h \end{bmatrix} V' = \begin{bmatrix} 0 \\ W_h V \end{bmatrix} \tag{2}$$

Las matrices W (o matrices de peso) tienen la característica de ser diagonales y contienen los valores de fuerza de contracción y atracción locales a cada uno de los vértices. Es decir que las entradas (i,i) de la matriz, contienen información sobre las fuerzas, basándose únicamente en su posición y en la de sus vecinos directos sin involucrar al resto de vértices.

Al ser un proceso iterativo, requiere de varias ejecuciones para obtener el esqueleto curvo de volumen cero aproximado. Por tanto, W_l y W_h deben actualizarse antes de la siguiente iteración. Au et al. [13] sugieren los siguientes pasos a realizar:

1) Solucionar (2) para V^{t+1}.
2) Actualizar W_l tal que: $W_l^{t+1} = s_l W_l^t$.
3) Actualizar $W_{h,i}$ tal que: $W_{h,i}^{t+1} = W_{h,i}^0 \sqrt{A_i^0 / A_i^t}$.
4) Calcular el nuevo operador laplaciano L^{t+1} con las posiciones de vértices actuales V^{t+1} utilizando (1).

Analizando la matriz L, se observa que contiene una gran cantidad de entradas nulas. Esto se debe a que por definición (1), sus valores se concentran en la diagonal y los índices (i,j) que conforman una arista en el mallado. En otras palabras la forma de la matriz L, es la misma que la matriz de adyacencia del modelo, junto con valores en la diagonal.

En la matriz laplaciana de la Figura 2 se puede observar que los elementos no nulos se encuentran cercanos a la diagonal y que gran parte de la matriz son valores nulos. Es por esto que la estructura de datos utilizada en implementación, debe ser una matriz esparcida. Además, tomando en cuenta que la matriz es del tamaño de la cantidad de vértices en el modelo, utilizar otro tipo de representación no sería aconsejable.

Continuando con el análisis del sistema de ecuaciones (2), definamos (3) y (4):

$$A = \begin{bmatrix} W_L L \\ W_H \end{bmatrix} \tag{3}$$

$$B = \begin{bmatrix} 0 \\ W_H V \end{bmatrix} \tag{4}$$

Estas definiciones permiten construir el sistema de ecuaciones sobredeterminado (tiene más ecuaciones que incógnitas) $Ax = B$. Donde A es una matriz rectangular, cuya forma es descrita por la Figura 3 y B son 3 vectores de lado derecho (*rhs, right hand side*), por lo que se plantea obtener su solución por medio de mínimos cuadrados (5), cuya forma es descrita en la segunda matriz de la Figura 2.

$$A^t A x = A^t B \tag{5}$$

[3]Según Thalmann y Thalmann [21], un esqueleto es una estructura compuesta de nodos y huesos que se asemeja a un esqueleto humano, donde los nodos representan las articulaciones, los huesos las extremidades y a su vez las articulaciones son la intersección de dos extremidades.

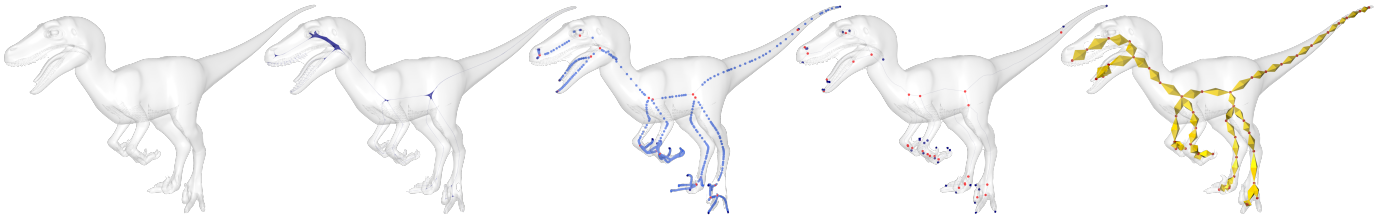

Figura 1: De Izquierda a Derecha cada Etapa para la Generación del Esqueleto: Modelo Inicial, Contracción de Geometría, Simplificación, Refinamiento, Selección

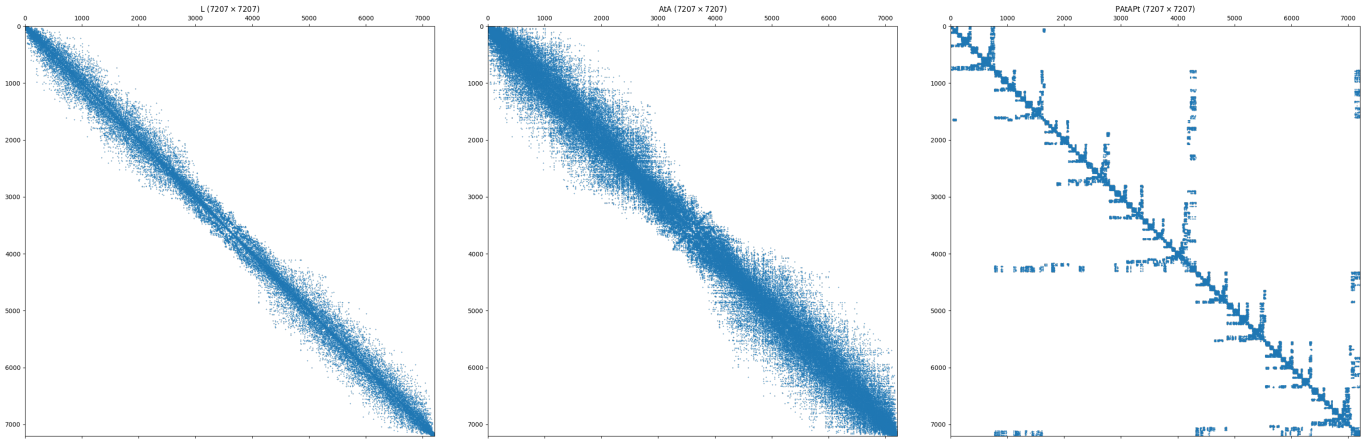

Figura 2: Matrices Involucradas en la Contracción Geométrica. De Izquierda a Derecha las Matrices: Laplaciana, A^tA y PA^tAP^t Usando la Permutación METIS. Todas Generadas para el Modelo *Cat*

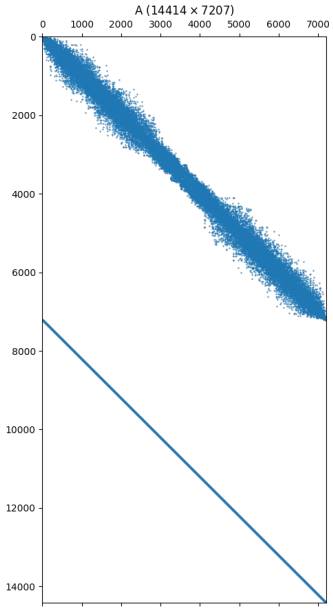

Figura 3: Forma de la Matriz A

Al utilizar el planteamiento de mínimos cuadrados, las condiciones del sistema *(matriz simétrica-positiva-definida)* resultante, son óptimas para aplicar diversos algoritmos de resolución de ecuaciones como: factorización Cholesky, LU o incluso QR. También se pueden utilizar métodos iterativos, en especial los basados en multigrilla [24][25], como lo expone Au et al. [13].

En la propuesta actual se evaluó el rendimiento de diversos métodos para resolver el sistema ecuaciones. Además, los métodos para solucionar (5) pueden ser paralelizados, por lo cual se desarrollaron alternativas que se ejecutan en la tarjeta gráfica, haciendo uso de la librería CUDA de NVIDIA.

Ya que el sistema de ecuaciones a resolver está representado en una matriz esparcida, su acceso es un cuello de botella en el rendimiento del algoritmo. El formato utilizado es CSC (*Compressed Sparse Column*) [26], el cual comprime la matriz ordenando los elementos no nulos de forma contigua en memoria por columnas y, de esta manera, minimiza la cantidad de memoria necesaria para almacenar grandes cantidades de datos.

Igualmente, la forma de acceder a las filas y columnas de la matriz difiere entre los distintos métodos para solucionar el sistema de ecuaciones, por lo cual dichos métodos son empleados en conjunción a una matriz de permutaciones P que permite reordenar la matriz inicial simétricamente (PA^tAP^t). Dicho reordenamiento, permite a los métodos de resolución, trabajar sin necesidad de permutar la matriz original [4], es por esto, que el tipo de ordenamiento juega un factor importante en la eficiencia del algoritmo final y es considerado en las pruebas que se realizan al sistema. Finalmente, la misma permutación aplicada a A debe aplicarse al vector de resultados del sistema. Luego de resolver el sistema, se aplica la permutación inversa para conservar el orden original del vector de resultados.

Existen diversos algoritmos de permutación para la resolución

[4]La permutación en una factorización de matrices, es una operación elemental que permite organizar los elementos de la mejor forma para ejecutar los pasos involucrados en ella.

de (5), se utilizaron los siguientes:

- **AMD** (*Approximate Minimun Degree ordering*) u ordenamiento de Grado Mínimo Aproximado: Pivota los valores concentrándolos bajo o sobre la diagonal [27].
- **RCM** (*Reverse Cuthill-McKee permutation*) o permutación Reversa de Cuthill-McKee: Concentra los elementos no nulos cerca de la diagonal [28].
- **MDQ** (*Minimum Degree by Quotient graph permutation*) o permutación de Grado Mínimo por grafo de Cociente: Variación del método AMD [25], con la misma finalidad [29].
- **METIS**: Permutación de matrices esparcidas por partición de grafos [30].

A nivel de implementación de todo el proceso de contracción, existen detalles que son necesarios aclarar:

1) Los objetos procesados están compuestos por triángulos y la suma de los ángulos internos de un triángulo es de 180 grados. Cuando los triángulos se comprimen para alcanzar un volumen cercano a cero, sus ángulos internos varían radicalmente. Ya que los cálculos involucran el uso de la función cotangente, la cual tiende al infinito negativo y positivo para ángulos cercanos a 0 y 180 grados respectivamente, es necesario limitar los ángulos internos de los triángulos durante el proceso de compresión para mantener la estabilidad del algoritmo.
2) Los parámetros iniciales para las matrices de peso son: $W_h^0 = 1$, $W_l^0 = 10^{-3}\sqrt{A}$ (donde A representa el área inicial promedio de las caras del modelo) y $sl = 2$, todos manteniendo los mismos valores indicados por Au et al. [13]. Cabe destacar que estos parámetros pueden ser modificados, pero estos valores por defecto generan resultados homogéneos independientemente del modelo o de su escala.
3) Las matrices diagonales W_h y W_l incrementan todos sus valores en cada una de las iteraciones aplicadas en la contracción. Por ello, se limita su crecimiento para prevenir un efecto de "sobrecontracción" que pueda afectar la solución de las ecuaciones y eliminar las características principales del objeto. Se proponen valores límites de: $W_l <= 2048$ y $W_h <= 10000$ y, al igual que el caso anterior, estos valores pueden ser modificados, pero los indicados por defecto logran adaptarse bien a todos los modelos de forma homogénea.

B. Simplificación

El proceso previo desplaza los vértices que componen la malla del objeto hasta alcanzar un volumen cercano a cero, pero no modifica bajo ninguna circunstancia su estructura, por ende, se necesita un procedimiento adicional que disminuya el número de vértices y aristas de la geometría resultante hasta conseguir los nodos claves (centrales) del esqueleto. Este proceso se conoce como "Simplificación" y consta de dos partes fundamentales: primero colapsa la mayor cantidad de vértices posibles que mantengan la estructura conexa y luego selecciona vértices y aristas, que bajo condiciones avanzadas preservan los nodos claves. De forma más detallada, los pasos del algoritmo son los siguientes:

1) Realizar una simplificación natural, que puede ser resumida en remover las aristas que sean tan cortas en el mallado, que su ausencia no represente ninguna perdida de características. En líneas generales, se define una distancia mínima constante que una arista debe sobrepasar, para que su eliminación represente un cambio visual notorio.
2) Seguidamente se aplica el algoritmo de simplificación de mallado QEM. Como indica Au et al. [13], QEM es modificado para que opere correctamente sobre vértices y aristas muy cercanos.

En esta propuesta, se realiza un paso previo al utilizado en [13], donde se agrupan todas las aristas del mallado en una cola de prioridad, tomando como valor referencial su longitud y, en cada iteración, se eliminan únicamente aquellas con longitud menor o igual a un δ especificado. Este proceso se repite descartando las aristas que superen ese tamaño hasta que eventualmente la cola quede vacía. El resultado de esta etapa es un mallado menos denso que conserva las características del mallado original. El proceso descrito fue agregado con la finalidad de reducir la entrada que se suministran a la siguiente etapa del algoritmo y mejorar su eficiencia al procesar modelos de gran tamaño.

Con el resultado previo, se hace uso del algoritmo QEM [31] para optimizar la selección de aristas a eliminar hasta conseguir el número esperado de vértices (nodos fundamentales). El proceso, califica cada arista con un peso que depende de la suma de las constantes de "forma" y "muestreo". Cada constante se calcula a partir de las posiciones actuales de los vértices, la longitud de las aristas y el número aristas que comparten un vértice en específico.

1) Costo o Constante de Forma: QEM calcula el costo de forma como la estimación de la distorsión causada por el colapso de una arista, mediante una métrica de error en cada vértice del mallado. La métrica mide las distancias al cuadrado entre un vértice y sus caras asociadas. En el caso específico de contracción de mallado, las caras que lo componen tienen área cero, por ende una métrica basada en caras no es válida para el caso de uso particular y se ajusta, resultando en la siguiente ecuación:

$$K_{ij} = \begin{bmatrix} 0 & -a_z & a_y & -b_x \\ a_z & 0 & -a_x & -b_y \\ -a_y & a_x & 0 & -b_z \end{bmatrix} \tag{6}$$

Donde a es el vector normalizado de la arista (i, j) y $b = a \times v_i$. La métrica de error inicial del vértice i es la suma de todas las distancias cuadradas a sus bordes adyacentes, es decir (7):

$$F_i(p) = p^T \sum_{(i,j)\in E} (K_{ij}^T K_{ij})p = p^T Q_i p \tag{7}$$

$Q_i p$ representa la matriz de error actual para el punto p. Para mantener la forma de la malla intacta durante la simplificación, se seleccionó el colapso de aristas $i \to j$ basado en el costo de forma (8):

$$F_a(i,j) = F_i(v_i) + F_j(v_j) \tag{8}$$

Es decir, el colapso $(i \rightarrow j)$, es aquel con la suma mínima de distancias al cuadrado desde la posición contraída v_j a las aristas adyacentes de los vértices i y j, así como a todos los adyacentes de los vértices que se contrajeron previamente al vértice i o j. Después de un colapso de arista, se actualiza la matriz de error del vértice j como: $Q_j \leftarrow Q_i + Q_j$, de manera que las aristas que estaban asociados al vértice i ahora se asocian al vértice j. Se almacena la matriz de error de cada vértice como una matriz 4×4 y, como en QEM, cada actualización de costos implica solo una adición de matriz.

2) Costo o Constante de Muestreo: El costo de forma conserva la apariencia de la malla original. Sin embargo, lleva a una simplificación excesiva en regiones rectas, produciendo bordes largos en el esqueleto y la pérdida de correspondencia fina entre el esqueleto y la superficie. Por lo tanto, se agrega un término de costo de muestreo que penaliza los colapsos que generan bordes largos. El costo de muestreo mide la distancia total que recorren las aristas adyacentes al vértice de origen, durante el colapso de la arista $(i \rightarrow j)$ (9):

$$F_b(i,j) = |v_i - v_j| \sum_{(i,k)\in E} |v_i - v_k| \tag{9}$$

Donde E es la conectividad perimetral simplificada actual.

3) Costo Total: La función de costo total es una suma ponderada del costo de forma y el costo de muestreo (10):

$$F(i,j) = w_a F_a(i,j) + w_b F_b(i,j) \tag{10}$$

Se utilizan como valores iniciales $w_a = 1.0$ y $w_b = 0.1$, tal como lo hacen Au et al. [13], pero ambas variables pueden ser modificadas directamente en la aplicación. Ya que las matrices de error de forma son acumuladas a medida que el algoritmo itera, el costo de forma tendrá más peso en la selección de aristas a colapsar, lo que prioriza la conservación de la forma en la malla contraída.

El algoritmo es implementado mediante el uso de una cola de prioridad al igual que la etapa previa. La cola, almacena los costos totales para cada arista y se actualiza en cada iteración. En cada paso se remueven los vértices productos de un colapso y se repite el proceso hasta obtener el número de vértices deseado que representan los nodos principales del esqueleto.

C. Refinamiento

De la etapa previa se obtienen un conjunto de nodos que componen el esqueleto curvo de la geometría original, estos nodos al ser producto de la contracción del mallado, se encuentran centrados dentro de la geometría original; sin embargo, debido a las características particulares que pueda tener un objeto, el esqueleto resultante puede no encontrarse correctamente centrado dentro de la geometría.

La etapa de refinamiento, corrige los errores de centrado que pueda tener el esqueleto curvo generado. El proceso de centrado consiste en evaluar la posición original de cada vértice que fue contraído para generar un nodo del esqueleto y reubicar dicho nodo en el centroide de los vértices seleccionados. Este proceso asegura que el esqueleto se encuentre correctamente centrado en las distintas partes que componen el objeto original.

Definiendo Π_k como la región local compuesta por los vértices contraídos para formar el nodo k. Se calcula el desplazamiento necesario para mover el nodo k al centro de la región, como el promedio de las posiciones de los vértices involucrados en la creación del nodo k (11). Donde d_k representa la nueva posición (centroide) para el nodo k del esqueleto.

$$d_k = \frac{\sum_{i\in\Pi_k}(v_i - v_k)}{|\Pi_k|} \tag{11}$$

Es importante destacar que el algoritmo de refinamiento propuesto por Au et al. [13] tiene un problema en algunos modelos donde ciertas regiones locales tienen su centroide fuera del objeto, ocasionando que el esqueleto se ubique fuera de la malla. En la propuesta actual previo a desplazar el esqueleto a su nueva posición, se verifica que su ubicación se encuentre dentro de la malla, lo que corrige el problema anteriormente mencionado.

D. Selección

El esqueleto curvo generado por las etapas previas tiene un alto número de nodos y aristas, ya que su finalidad es adaptarse lo mejor posible al objeto que lo define. Sin embargo, se plantea un proceso de selección, para reducir el número de nodos a solo los más representativos. Para el caso de uso particular de animaciones, se considera que un esqueleto compuesto de 20 a 40 nodos representa una buena base general.

El proceso es similar a la simplificación QEM antes expuesta, la diferencia radica en que las funciones de costo miden el tamaño de cada arista (ver ecuación 12), lo que permite eliminar las menos representativas (las de menor tamaño) hasta alcanzar el número de nodos deseado.

$$F(i,j) = |v_i - v_j| \tag{12}$$

E. Asignación de Pesos

Con el esqueleto simplificado, se procede a agregar la información necesaria que permita a los huesos que lo conformen, alterar la posición del mallado original y por ende animarlo. Específicamente, el "peso", se refiere al grado de influencia que las transformaciones espaciales aplicadas a un hueso, tendrán sobre los vértices que componen el objeto original. La posición final de un vértice viene dada por la suma de las transformaciones espaciales de cada hueso que lo afecten, ponderada por los pesos o influencias de cada hueso sobre el vértice, por ende, la suma de los pesos de todos los huesos que afecten a un vértice nunca excederá uno.

Para el proceso de pesado se utiliza el planteamiento de Baran y Popoviç [22]. Donde se plantea simular un equilibrio de "calor" dentro de cada hueso, de manera que este influya sobre los vértices cercanos. El equilibrio sobre la superficie del hueso i viene dado por (13), que se puede escribir como (14):

$$\frac{\partial w^2}{\partial t} = \Delta w^i + H(p^i - w^i) = 0 \tag{13}$$

$$-\Delta w^i + H w^i = H p^i \tag{14}$$

Donde Δ es la superficie discreta laplaciana, calculada con la fórmula cotangente de Meyer et al. [32], p^i es un vector, donde $p^i_j = 1$ si el hueso más cercano al vértice *j* es *i* y $p^i_j = 0$ de lo contrario.

H es una matriz diagonal, donde H_{ij} es el peso de contribución de calor de el hueso más cercano al vértice j. Sea $d(j)$ la distancia desde el vértice j al hueso más cercano, $H_{jj} = c/d(j)^2$, si el segmento de línea más corto desde el vértice hasta el hueso está contenido en el volumen del modelo y, $H_{jj} = 0$ en caso contrario. Para $c \approx 0.22$, el método genera pesos con transiciones similares a las calculadas al encontrar el equilibrio sobre el volumen. Pinocho[5] usa $c = 1$ (correspondiente al calor anisotrópico de difusión) dado que los resultados lucen más naturales.

Cuando k huesos son equidistantes del vértice j, $p_j = 1/k$ para todos ellos y, $H_{jj} = kc/d(j)^2$.

La ecuación (14) es un sistema de ecuaciones lineal esparcido, donde su lado izquierdo (matriz $-\Delta + H$) no depende de i, siendo i cada uno de los huesos en el esqueleto. Esto permite factorizar el sistema una sola vez y con el resultado calcular el peso de cada hueso. Botsch [33], muestra cómo usar la factorización Cholesky para este tipo de problema. Es importante aclarar que los pesos W^i siempre deben sumar 1 por cada vértice: si sumamos (14) sobre i, se obtiene $(-\Delta + H)\sum_i w^i = H \cdot 1$ que equivale a $\sum_i w^i = 1$.

F. Exportar

Para poder empaquetar las estructuras generadas en un archivo soportado por cualquier programa de animación, es necesario conocer los fundamentos del formato animado. La idea principal es definir cada hueso del esqueleto no como dos vértices y una arista, sino por el contrario de manera analítica, como un vector con origen y dirección [34] (no es necesario incluir la longitud). Además hay que definir una jerarquía de huesos, donde cada uno tiene exactamente un padre, a excepción del hueso raíz. Cabe destacar que el hueso raíz no necesariamente se encuentra siempre dentro del modelo y eso no influye en el formato, recordemos que los huesos son estructuras que no se despliegan en la animación final, sino que simplemente sirven para el análisis de como se debe comportar cada vértice del modelo.

[5]Librería introducida por Baran y Popoviç [22].

Otra característica importante a resaltar, es la relación que existe entre cada hueso. El esqueleto es una estructura jerárquica donde la transformación espacial aplicada a un hueso padre, es heredada por su hijo. De manera que se si se mueve un hueso padre, todos sus hijos se moverán con él.

Las transformaciones de cada hueso son almacenadas en una matriz de 4x4 que contiene las transformaciones afines a aplicar al hueso canónico[6]. Existen 3 tipos importantes de matrices para expresar este modelo, y son los siguientes:

1) Matriz de Modelo: Define una transformación del origen del modelo sobre la posición absoluta del hueso. Se puede determinar a partir de la posición de origen de las aristas (v_i) y su dirección ($|v_j - v_i|$).
2) Matriz de Modelo (inversa): Inversa de la matriz anterior. Se utiliza para definir como se mueve el hueso al origen del modelo.
3) Matriz Local: Describe la posición y transformación de cada hueso, basada en el información de su hueso padre.

La matriz local es la única que se almacena en el formato de salida. Esta se puede calcular en base a la matriz de modelo de cada hueso, partiendo de la ecuación (15):

$$M_{Wk} = M_{Lk-n} * M_{Lk-(n-1)} * ...M_{Lk-1} \tag{15}$$

Donde M_W y M_L son las matrices de modelo y locales respectivamente, esta ecuación puede ser reescrita de la forma (16), para obtener la matriz local.

$$M_{Lk} = M^{-1}_{Wk-n} * M^{-1}_{Wk-(n-1)} * ...M^{-1}_{Wk-1} \tag{16}$$

Los huesos a exportar deben encontrarse, en lo que se que se conoce como la "pose en descanso", lo cual corresponde con la transformación inicial de los huesos que no alteren la posición de los vértices. Con esta información se puede crear un archivo de salida en formato Filmbox (FBX), el cual puede ser usado por diferentes aplicaciones de creación de contenido 3D, concluyendo el proceso de esqueletonización.

IV. Pruebas

Durante las pruebas realizadas al sistema, se experimentó con 8 objetos tridimensionales con distintas características topológicas y de conteo poligonal. Dichas variaciones verifican:

- El crecimiento en el tiempo de computo ante la variación en la cantidad de polígonos del objeto procesado.
- La efectividad del algoritmo ante variaciones en la topología del objeto procesado.
- La robustez del algoritmo para procesar objetos con gran cantidad de vértices y caras.

Por cada etapa del método propuesto, se realizaron pruebas individuales que verifican su correcto funcionamiento. Los objetos tridimensionales utilizados pueden observarse en la Figura 4 y sus conteos de polígonos se indican a continuación:

[6]El hueso canónico es el que se encuentra situado con origen en (0, 0, 0) y tiene una longitud ι de tal forma que el vector que lo define es (ι, 0, 0).

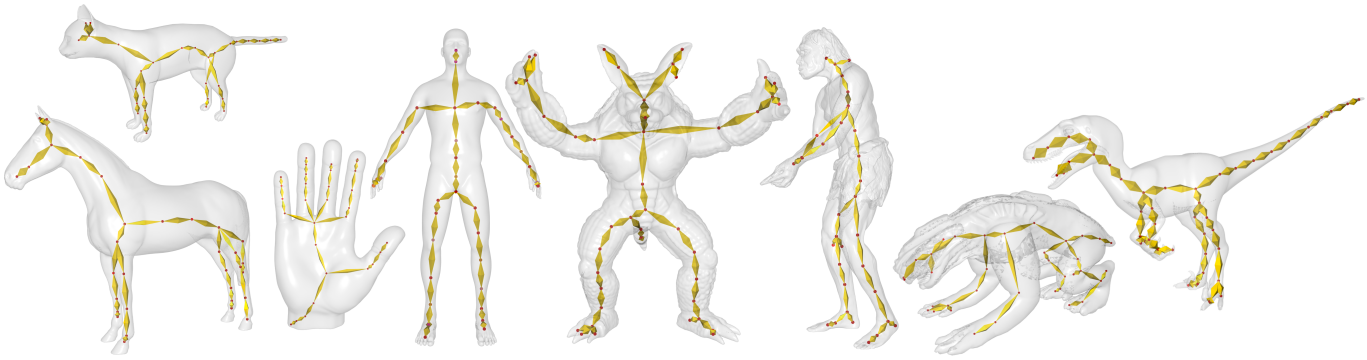

Figura 4: Modelos Utilizados en las Pruebas

- *Cat*: 7207 vértices y 14410 caras.
- *Man*: 24461 vértices y 48918 caras.
- *Hand*: 66848 vértices y 133692 caras.
- *Horse*: 120660 vértices y 241316 caras.
- *Armadillo*: 172974 vértices y 345944 caras.
- *Alien*: 373036 vértices y 745408 caras.
- *Raptor*: 595462 vértices y 1190656 caras.
- *Homoheidelbergensis* 1088639 vértices y 2176968 caras (escultura capturada como nube de puntos).

El ambiente en el que se realizaron las pruebas es el siguiente:

1) Procesador: AMD de 8 núcleos.
2) Memoria RAM: 16GB.
3) Tarjeta Gráfica: NVIDIA GTX 1050ti de 4GB VRAM con 768 CUDA cores.
4) Sistema Operativo: Windows 10.

En cada prueba, se realizaron todas las posibles permutaciones entre el método utilizado para resolver el sistema de ecuaciones y la permutación aplicada a la matriz que lo define. Estas pruebas tienen como intención determinar la mejor combinación entre ambos factores. De igual manera, se evaluó el desempeño del algoritmo tanto en GPU como en CPU.

Las factorizaciones de matrices para los métodos de resolución de ecuaciones directos planteadas, son:

- Cholesky.
- LU.
- QR.

Y las permutaciones ejecutadas sobre cada una de ellas son:

- AMD: Implementada con la librería Eigen.
- SymAMD: Implementada con el API de CUDA.
- MDQ: Implementada con el API de CUDA.
- RCM: Implementada con el API de CUDA.
- METIS: Implementada con el API de CUDA.

Es importante aclarar, que pese a que las pruebas se realizaron tanto en GPU como en CPU, los algoritmos para generar las matrices de permutaciones residen en CPU (inclusive en la implementación en CUDA). Igualmente, el procesamiento de la matriz de permutaciones (P) se realiza una única vez, ya que a pesar de que la contracción del mallado necesite varias iteraciones, la matriz P siempre será la misma, dado a que esta depende únicamente de la estructura de la matriz (su forma) y no de los valores per sé.

Tabla I: Tiempos: Solucionador-Permutaciones

	AMD	S.AMD	MDQ	RCM	METIS
Cholesky	0.5371s	0.5065s	0.839s	0.504s	0.4963s
LU	2.29s	7.331s	1.984s	7.238s	2.407s
QR	11.46s	19.24s	2.55s	19.13s	9.77s

En el Tabla I se pueden observar los resultados promediados en CPU y GPU para el modelo *Cat*, en donde se evidencia que la mejor combinación es la permutación METIS con la factorización Cholesky. Adicionalmente, se puede apreciar en la Figura 5 la solución obtenida. A partir de este punto, el resto de las pruebas con los métodos de solución de ecuaciones directos serán expresados bajo la combinación METIS-Cholesky (ver última columna, Figura 2).

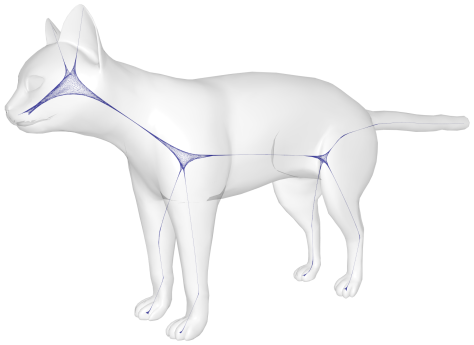

Figura 5: Contracción de Geometría Aplicada Sobre el Modelo *Cat*

La Tabla II compara los tiempos obtenidos por los solucionadores directos y los iterativos basados en multigrilla (AMG [24][25] implementado en CPU en este caso). En estos resultados se observa claramente como a medida que se incrementa el número de polígonos a procesar, el tiempo de procesamiento de los algoritmos incrementa. Además se puede concluir que los métodos directos son más rápidos que los basados en multigrillas, siendo la solución en CPU del método

Tabla II: Métodos Directos (CPU, GPU) vs. AMG

	CPU	GPU	AMG
Cat	0.272s	0.713s	0.323s
Man	1.903s	2.914s	2.043s
Hand	9.272s	10.395s	12.209s
Horse	43.745s	25.948s	70.576s
Armadillo	94.158s	60.654s	157.747s
Alien	96.080s	83.394s	169.855s
Raptor	91.341s	108.729s	148.99s
Homoheidel	749.701s	238.427s	1294.5s

directo 1.7 veces más rápida que la iterativa para el modelo más pesado y la solución en GPU 5.4 veces más rápida.

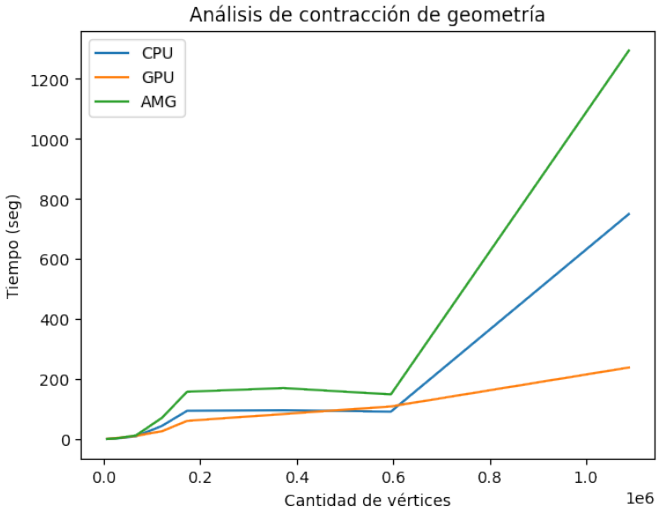

Figura 6: Comparación CPU, GPU, AMG

De igual manera, se evidencia que cuando se procesan modelos de bajo número de polígonos, el costo asociado a la creación del contexto para ejecutar el algoritmo en GPU supera el tiempo de procesamiento del mallado, por lo que las variantes en CPU resultan más eficientes. Sin embargo, al crecer el conteo de polígonos, el tiempo de creación de contexto se vuelve irrelevante y la implementación en GPU exhibe un crecimiento lineal a diferencia del exponencial observado en las vertientes en CPU (ver Figura 6).

Es importante mencionar que la contracción de mallado depende del laplaciano como operador principal, por ende los tiempos se comportan diferentes si la geometría y la topología del modelos son diferentes; por ejemplo: resolver la contracción para una esfera es mucho más rápido que resolverla para un gato o un humano, incluso si la cantidad de vértices es la misma. Otro punto importante es que mientras se tomen en cuenta detalles de implementación como la cotangente en infinito negativo y positivo, la solución de la contracción de geometría es insensible a la escala de la malla.

Finalmente, cabe destacar que el método utilizado no tiene limitantes en cuanto a tamaño del problema, funciona correctamente incluso para casos de prueba con un alto conteo de polígonos involucrados. La única limitante va directamente relacionada con la capacidad del hardware utilizado.

A. *Operador Laplaciano*

Se realizaron pruebas visuales sobre el operador laplaciano que demuestran como evoluciona la curvatura en cada una de las iteraciones del algoritmo hasta que se acerca a los valores máximos y mínimos de la cotangente. Esto fue implementado con la misma metodología de visualización que posee MATLAB en su función *jet*, donde se codifica la información en colores. En la Figura 7 el color azul representa los valores mínimos (menos curvatura) de la cotangente mientras que el rojo el máximo (curvatura pronunciada). A medida que los triángulos del modelo se comprimen convergiendo al esqueleto curvo, la representación de colores se torna azul oscuro, indicando que no hay curvatura y que efectivamente los vértices y las aristas se encuentran tan cerca que el volumen tiende a cero.

B. *Simplificación*

Del proceso de simplificación se pueden extraer resultados parciales que permiten evaluar el funcionamiento del algoritmo. Se puede visualizar el mapeo inducido *esqueleto-malla* representado en la Figura 8 mediante secciones alternando entre el color blanco y cyan. Este mapeo se refiere a los vértices absorbidos por cada nodo durante el proceso de simplificación y su función es confirmar que la etapa se completó de forma correcta ya que cada nodo incluye únicamente a los vértices cercanos a él, generando el efecto que se aprecia en dicha Figura 8.

C. *Refinamiento*

Por otro lado, el refinamiento también genera resultados intermedios tales como la segmentación del mallado, la cual es de utilidad en diversos campos de la computación gráfica. La forma de representarla es identificando con diferentes colores cada uno de los segmentos que se desprenden del modelo inicial, tal como se puede apreciar en la Figura 9.

D. *Selección*

Si bien el proceso de selección (ver Figura 10) no genera resultados parciales, modifica la estructura del esqueleto y por ende los resultados parciales de las etapas previas, disminuyendo el número de secciones en el mapeo inducido *esqueleto-malla* debido a que son menos nodos. El caso análogo sucede con los segmentos generados en el refinamiento, ahora abarcan más espacio y hay menos segmentos. En la Figura 11 se puede apreciar la evolución de los resultados obtenidos luego de aplicar la etapa de selección, a la izquierda se puede observar el aumento del tamaño de las secciones del mapeo inducido sobre el modelo *Man* (comparar con Figura 8), a la derecha se observa el incremento de la dimensión de los segmentos sobre el modelo *Armadillo*, a su vez se pude notar como la cola y las extremidades del modelo tienen menos secciones y como las mismas abarcan más área (comparar con Figura 9).

E. *Asignación de Pesos*

Se verificó que cada vértice tuviese un peso asignado acorde al hueso más cercano. En la Figura 12, se selecciona un hueso

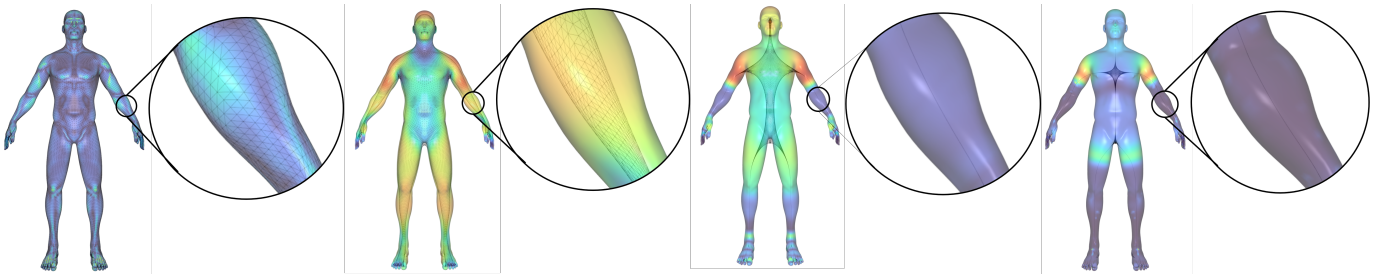

Figura 7: Evolución del Operador Laplaciano sobre el Modelo *Man* en cada Iteración de la Contracción

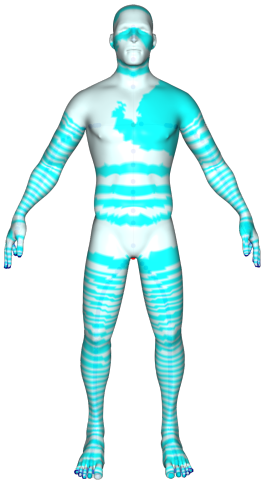

Figura 8: Mapeo Inducido *Esqueleto-Malla*, Representado por las Secciones Alternadas de Colores Blanco y Cyan, Asociadas a cada Nodo que las Contiene. Sobre el Modelo *Man*

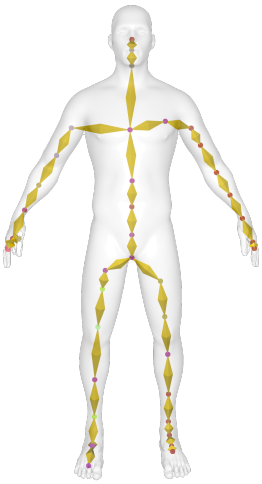

Figura 10: Proceso de Selección. Cada Hueso se Representa con Dos Rombos en Color Amarillo y cada Nodo como Punto Rojo. Sobre el Modelo *Man*

Figura 9: Segmentación sobre el Modelo *Armadillo*

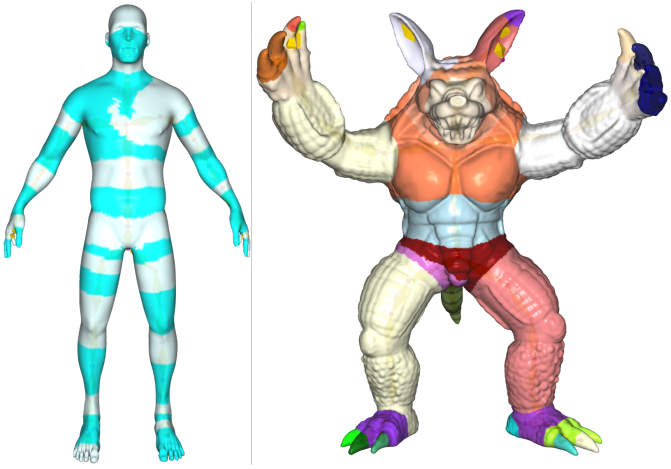

Figura 11: Resultados Posterior a Aplicar la Etapa de Selección (Comparar con Figuras 8 y 9)

y el color azul indica que los vértices no son influenciados por este, mientras que el color rojo expresa un alto grado de influencia.

F. Tiempos Totales

Finalmente, se presenta los tiempos individuales por etapa del algoritmo propuesto. En el Tabla III se puede observar que la etapa de contracción involucra aproximadamente el 77% del total.

V. Conclusión y Trabajos Futuros

En el presente trabajo, se desarrolló un método de esqueletonización basado en geometría, tomando como referente el propuesto por Au et al. [13]. Mediante las pruebas realizadas, se puede concluir que el método propuesto es 5.4 veces más rápido que la referencia.

Adicionalmente, las pruebas indican que los métodos de solución directa de sistemas de ecuaciones son más eficientes que los iterativos y, que la permutación METIS con la factorización

Tabla III: Tiempos por Etapa

Modelo	Contracción	Simplificación	Refinamiento	Selección	Peso	Exportación	Total
Cat	0.71379s	0.10455s	0.0017008s	0.0028183s	0.10477s	0.085525s	1.0132s
Man	2.9148s	0.4708s	0.010105s	0.016621s	0.22983s	0.22872s	3.8709s
Hand	10.396s	1.4438s	0.022026s	0.043721s	0.78632s	0.59756s	13.289s
Horse	25.948s	2.7979s	0.038827s	0.07831s	2.023s	1.075s	31.961s
Armadillo	60.654s	4.847s	0.0507s	0.12393s	2.7777s	1.628s	70.082s
Alien	83.395s	11.914s	0.10427s	0.47783s	5.3554s	2.5349s	103.78s
Raptor	108.73s	20.05s	0.1892s	0.61818s	7.7117s	4.3389s	141.64s
Homoheidel	238.43s	34.006s	0.39226s	0.9673s	24.722	12.042s	310.56s

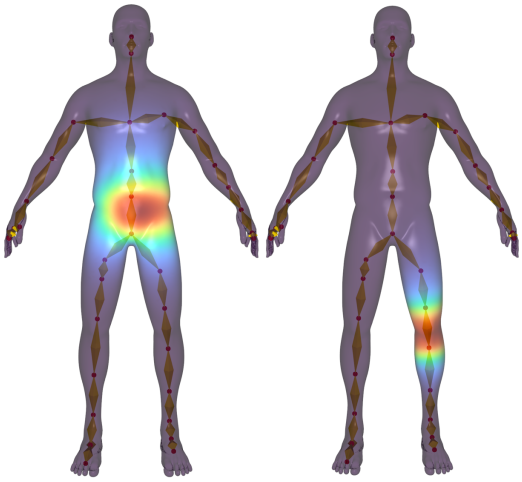

Figura 12: Pesos Asignados

Cholesky es la óptima para el caso particular de contracción de geometría.

Finalmente, el esqueleto curvo generado se adaptó para ser utilizado en animaciones por computador, permitiendo incluso exportar el resultado del algoritmo en un formato estándar soportado por todas las herramientas de animación tridimensional.

Como trabajo futuro, se propone expandir los casos de uso a los que puede ser adaptado el esqueleto generado por el algoritmo, a fin de abarcar más allá de la industria del entretenimiento.

REFERENCIAS

[1] T. He, L. Hong, D. Chen, and Z. Liang, *Reliable Path for Virtual Endoscopy: Ensuring Complete Examination of Human Organs*, IEEE Transactions on Visualization and Computer Graphics, vol. 7, pp. 333–342, 2001.

[2] L. Hong, S. Muraki, A. Kaufman, D. Bartz, and T. He, *Virtual Voyage: Interactive Navigation in the Human Colon*, in Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH), pp. 27–34, Los Angeles, CA, USA, 1997.

[3] D. Perchet, C. Fetita, and F. Preteux, *Advanced Navigation Tools for Virtual Bronchoscopy*, Image Processing: Algorithms and Systems III, vol. 5298, pp. 147–158, 2004.

[4] D. Bartz, W. Straßer, M. Skalej, and D. Welte, *Interactive Exploration of Extra- and Intracranial Blood Vessels (Case Study)*, in Proceedings of the Conference on Visualization 1999: Celebrating Ten Years, pp. 389–392, San Francisco, CA, USA, 1999.

[5] S. Pizer, D. Fritsch, P. Yushkevich, V. Johnson, and E. Chaney, *Segmentation, Registration and Measurement of Shape Variation Via Image Object Shape*, IEEE Transactions on Medical Imaging, vol. 18, pp. 851–865, 1999.

[6] S. Pizer, G. Gerig, S. Joshi, and S. Aylward, *Multiscale Medial Shape-Based Analysis of Image Objects*, Proceedings of the IEEE, vol. 91, pp. 1670–1679, 2003.

[7] M. Straka, M. Cervenansky, A. La cruz, A. Kochl, M. Sramek, E. Gröller, and D. Fleischmann, *The VesselGlyph: Focus & Context Visualization in CT-Angiography*, Visualization, pp. 385–392, 2004.

[8] C. Menier, E. Boyer, and B. Raffin, *3D Skeleton-Based Body Pose Recovery*, Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06), pp. 389–396, Chapel Hill, NC, USA, 2006.

[9] S. Aylward, J. Jomier, S. Weeks, and E. Bullitt, *Registration and Analysis of Vascular Images*, International Journal of Computer Vision, vol. 55, pp. 123–138, 2003.

[10] R. Blanding, G. Turkiyyah, D. Storti, and M. Ganter, *Skeleton-Based Three-Dimensional Geometric Morphing*, Computational Geometry, vol. 15, pp. 129–148, 2000.

[11] J. Bloomenthal, *Medial-Based Vertex Deformation*, in Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 147–151, San Antonio, TX, USA, 2002.

[12] A. Vasilakis, *Skeleton-Based Rigid Skinning for Character Animation*, International Conference on Computer Graphics Theory and Applications, Funchal, MD, Portugal, 2008.

[13] O. Au, C. Tai, H. Chu, D. Cohen-Or, and T. Lee, *Skeleton Extraction by Mesh Contraction*, ACM Transactions on Graphics, vol. 27, 2008.

[14] H. Blum, *Biological Shape and Visual Science (Part I)*, Journal of Theoretical Biology, vol. 38, 1973.

[15] N. Cornea, D. Silver, and P. Min, *Curve-Skeleton Properties, Applications, and Algorithms*, IEEE Transactions on Visualization and Computer Graphics, vol. 13, no. 3, pp. 530–548, 2007.

[16] G. Bertrand and Z. Aktouf, *Three-Dimensional Thinning Algorithm using Subfields*, in Proceedings on Vision Geometry III, vol. 2356, Boston, MA, USA, 1995.

[17] D. Morgenthaler, *Three-Dimensional Simple Points: Serial Erosion, Parallel Thinning, and Skeletonization*, Technical Report: Computer Science Center, University of Maryland, 1981.

[18] T. Lee, R. Kashyap, and C. Chu, *Building Skeleton Models via 3-D Medial Surface/Axis Thinning Algorithms*, CVGIP: Graphical Models and Image Processing, vol. 56, no. 6, pp. 462–478, 1994.

[19] K. Palágyi, *A 3D Fully Parallel Surface-Thinning Algorithm*, Theoretical Computer Science, vol. 406, no. 1, pp. 119–135, 2008.

[20] G. Borgefors, I Nyström, and G. Sanniti Di Baja, *Computing Skeletons in Three Dimensions*, Pattern Recognition, vol. 32, no. 7, pp. 1225–1236, 1999.

[21] N. Thalmann and D. Thalmann, *Computer Animation: Theory and Practice*, Computer Science Workbench, 1985.

[22] I. Baran and J. Popović, *Automatic Rigging and Animation of 3D Characters*, ACM Transactions on Graphics, vol. 26, no. 3, pp. 72–80, 2007.

[23] M. Desbrun, M. Meyer, P. Schröder, and A. Barr, *Implicit Fairing of Irregular Meshes Using Diffusion and Curvature Flow*, in Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, pp. 317–324, New York, NY, USA, 1999.

[24] B. Metsch, *Algebraic Multigrid (AMG) for Saddle Point Systems*, 2013.

[25] D. Demidov, *AMGCL—A C++ Library for Efficient Solution of Large Sparse Linear Systems*, Software Impacts, vol. 6, 2020.

[26] Y. Saad, *SPARSKIT: a Basic Tool Kit for Sparse Matrix Computations - Version 2*, 1994.

[27] P. Amestoy, T. Davis, and I. Duff, *An Approximate Minimum Degree Ordering Algorithm*, SIAM Journal on Matrix Analysis and Applications, vol. 17, no. 4, pp. 886–905, 1996.

[28] E. Cuthill and J. McKee, *Reducing the Bandwidth of Sparse Symmetric Matrices*, in Proceedings of the 1969 24th National Conference, pp. 157–172, New York, NY, USA, 1969.

[29] A. George and J. Liu, *A Fast Implementation of the Minimum Degree Algorithm using Quotient Graphs*, ACM Transactions on Mathematical Software (TOMS), vol. 6, no. 3, pp. 337–358, 1980.

[30] G. Karypis and V. Kumar, *METIS – Unstructured Graph Partitioning and Sparse Matrix Ordering System, Version 2.0*, 1995.

[31] M. Garland and P. Heckbert, *Surface Simplification Using Quadric Error Metrics*, ACM Siggraph, 1997.

[32] M. Meyer, M. Desbrun, P. Schröder, and A. Barr, *Discrete Differential-Geometry Operators for Triangulated 2-Manifolds*, Visualization and Mathematics III, pp. 35–57, 2003.

[33] M. Botsch, D. Bommes, and L. Kobbelt, *Efficient Linear System Solvers for Mesh Processing*, Mathematics of Surfaces XI, pp. 62–83, 2005.

[34] S. Marschner and P. Shirley, *Fundamentals of Computer Graphics*, 2018.

Revista Venezolana de Computación
ISSN: 2244-7040
http://www.svc.net.ve/revecom
Vol. 8, No. 1, pp. 37-46, Junio 2021
Fecha recepción: 22/03/2021, Fecha aceptación: 15/08/2021

Despliegue de un Clúster HPC Utilizando Contenedores Docker

José Francisco Hernández[1], Gianfranco Alfredo Verrocchi[1], Yudith Cardinale[2,3], Ana Aguilera[4], Irvin Dongo[3,5]
kikohernandez92@gmail.com, gianfrancoverrocchi@gmail.com, ycardinale@usb.ve, ana.aguilera@uv.cl, ifdongo@ucsp.edu.pe

[1] Escuela de Informática, Universidad Católica Andrés Bello, Caracas, Venezuela
[2] Departamento de Computación, Universidad Simón Bolívar, Caracas, Venezuela
[3] Electrical and Electronics Engineering Department, Universidad Católica San Pablo, Arequipa, Peru
[4] Escuela de Ingeniería Informática, Universidad de Valparaíso, Valparaíso, Chile
[5] Univ. Bordeaux, ESTIA Institute of Technology, Bidart, France

Resumen: En ambientes de computación de alto rendimiento (HPC por sus siglas en inglés de *High Performance Computing*), una de las tareas más difíciles, tanto para los administradores de sistemas como para los usuarios finales, es preparar el ambiente de trabajo (e.g., configuración del clúster) y resolver las dependencias de software. En este contexto, este trabajo propone una solución de software, basada en la moderna tecnología de contenedores Docker, para construir un clúster HPC virtual con computadores personales de manera simple y automática, particularmente para trabajar bajo el paradigma de paso de mensajes con el estándar MPI (*Message Passing Interface*). La solución de software propuesta, se despliega de forma automatizada en virtualmente cualquier ambiente que soporte la instalación de la herramienta Docker, permite cubrir múltiples casos de uso y provee flexibilidad en cuanto a requerimientos de software, hardware y disponibilidad de equipos de cómputo. Para demostrar la flexibilidad y conveniencia de la solución propuesta, en este trabajo presentamos dos versiones de instalación: uno con el sistema operativo Alpine y librería de comunicación MPICH, y el segundo utiliza el sistema operativo Ubuntu y la librería de comunicación OpenMPI. Para evaluar la efectividad de la solución propuesta, se realizan pruebas con ambas versiones en dos ambientes de prueba: usando una red local (con tres computadores conectados con Ethernet) y usando cinco nodos virtualizados. Los resultados demuestran que la solución desarrollada puede ser desplegada eficientemente en distintos ambientes (real y virtualizado), lo que la hace ideal para fines académicos.

Palabras Clave: Cluster HPC; Contenedores; Docker; Instalación Automática; MPI.

Abstract: On High Performance Computing (HPC) environments, one of the hardest tasks for system administrators and developers, is to prepare their working environment (e.g., building and configuring a cluster) and resolve software dependencies. In this context, this work proposes a software solution based on the modern Docker container technology, to build a virtual HPC cluster in a simple and automated way using personal computers, particularly to work with the MPI (Message Passing Interface) standard. Our proposed software solution, can be deployed in an automated way in any environment that supports the Docker run time installation. It allows to cover multiple use cases and provides flexibility in terms of software requirements, hardware, and computer equipment availability. To demonstrate the flexibility and convenience of the proposed solution, we develop two versions of possible installation: one with Alpine operating system and MPICH and the second one with Ubuntu operating system and OpenMPI. To evaluate the effectiveness of the proposed solution, we test both versions in two different scenarios: one built with a local network (with three computers connect through Ethernet) and the other one with five nodes in virtual machines. Results demonstrate that the developed solution can be deployed efficiently across different environments (real and virtualized), which makes it ideal for academic purposes.

Keywords: HPC Cluster; Containers; Docker; Automatic Installation; MPI.

I. Introducción

La computación de alto rendimiento (HPC por sus siglas en inglés de *High Performance Computing*) implica el uso de plataformas multi-procesadores, particularmente clústers de computadores, y de software especializado como librerías de paso de mensajes (e.g., MPI, OpenMP) y manejadores de colas (e.g., Torque, Slurm). En ambientes HPC, por un lado, una de las tareas más difíciles es preparar el ambiente de trabajo (e.g.,

construcción y configuración del clúster) y resolver las dependencias de software. Generalmente encontramos conflictos en las versiones del software, así como con las características del sistema relacionadas con, por ejemplo, el sistema operativo, librerías, kernel; lo que conlleva a la instalación, en los recursos computacionales, de múltiples versiones de software y librerías especializadas. En consecuencia, se incrementa la complejidad de las tareas de los administradores del sistema y la inconformidad de los usuarios del ambiente de HPC, al tener que modificar y recompilar sus aplicaciones para que se ejecute en el ambiente HPC disponible.

Por otro lado, es común en un ambiente de clúster HPC que los usuarios y desarrolladores pretendan disponer, de manera exclusiva, de los recursos computacionales. Esto representa limitaciones, si las aplicaciones se ejecutan por mucho tiempo, privando al resto de los usuarios de su uso. Herramientas, como el planificador de tareas Slurm, evitan que procesos mal programados utilicen los recursos indefinidamente. Sin embargo, aún con estos mecanismos de control, es posible que estos procesos erráticos afecten el estado del clúster, impidiendo que futuros procesos funcionen con normalidad y generan la necesidad de auditar el sistema e incluso de causar interrupciones en el servicio.

La reciente tecnología de contenedores, y en particular, de contenedores Docker, se presenta como una solución potencial para simplificar el despliegue y administración del hardware y software requeridos para construir un clúster HPC. Docker es una plataforma amigable y muy popular, para ejecutar y administrar contenedores Linux. De hecho, actualmente, la vasta mayoría de herramientas provistas en contenedores, están empaquetadas en imágenes Docker. La naturaleza declarativa de la herramienta Docker y el aislamiento de procesos mediante el uso de contenedores `lxc`[1], permite entonces definir un ambiente de clúster que funcione de manera ininterrumpida, posea alta disponibilidad, y más importante aún, que permita que el estado del clúster sea replicable e independiente del ambiente en el que es desplegado. Esto permite que un ambiente de clúster pueda coexistir a la par de otras instancias del mismo con otras herramientas o configuraciones especializadas. Otro aspecto importante es que esta automatización abstrae las dificultades y necesidad de conocimiento técnico y costos de hardware requerido para implementar un ambiente de HPC.

Dados esos beneficios, recientemente existen varios trabajos que proponen el uso de contenedores Docker para el despliegue y configuración de clústers HPC [1]–[8]. Sin embargo, estas soluciones presentan limitaciones por ser de ámbito de aplicación específico y ofrecer una integración muy pobre de diversas tecnologías de HPC.

Con la intención de sobreponer tales limitaciones, en este trabajo de investigación proponemos una solución de software, basada en contenedores Docker, para construir un clúster HPC virtual con computadores personales. La solución propuesta ofrece varios ambientes de desarrollo y prueba de programas que se desenvuelven en el ámbito de la computación distribuida, computación paralela y computación de alto rendimiento, particularmente bajo el paradigma de pase de mensajes con el estándar MPI (*Message Passing Interface*). La solución propuesta permite desplegar un ambiente de clúster HPC a usuarios o desarrolladores que no cuenten con los recursos, tiempo o experiencia para conceptualizar, implementar y mantener un ambiente de clúster para el desarrollo de aplicativos que aprovechen un ambiente con estas características.

Nuestra solución extiende el trabajo propuesto en [5] y modifica e integra las soluciones parciales disponibles en [9], [10]; aprovechando la característica de auto-escalamiento de los servicios de descubrimiento y la herramienta de virtualización liviana de Docker. La solución propuesta se despliega de forma automatizada en virtualmente cualquier ambiente que soporte la instalación de la herramienta Docker. Esta solución está compuesta de varias imágenes, cuyo alcance cubre múltiples casos de uso y provee flexibilidad en cuanto a requerimientos de hardware, disponibilidad de equipos y necesidades del ambiente de desarrollo. En este trabajo presentamos el proceso detallado de construcción de los contenedores y las diferentes alternativas desarrolladas e integradas, así como las nuevas posibilidades que la solución resultante permite. Se especifican las consideraciones que se deben tomar al crear una imagen Docker y los pasos necesarios para desplegar un clúster HPC con nodos MPI, sobre un grupo de computadores personales. Nuestro objetivo es proveer la estrategia metodológica detrás de nuestra solución, de manera que otros interesados puedan adaptarla a diferentes requerimientos del sistema. Las variantes que definen los ambientes que ofrece esta solución forman parte de un mismo proyecto dentro de un repositorio de Docker Hub [11], en donde cada variante (imagen de Docker y código fuente) tiene un conjunto de herramientas completamente diferente una de la otra. Sin embargo, la arquitectura de despliegue se mantiene: un nodo maestro y N nodos esclavos.

Para evaluar el desempeño de la solución propuesta, realizamos una serie de pruebas de *benchmarking* comparando las versiones desarrolladas en diferentes ambientes: usando una red local (con tres computadores conectados con Ethernet) y usando cinco nodos virtualizados. Los resultados demuestran que la solución desarrollada puede ser desplegada eficientemente en distintos ambientes (real y virtualizado), lo que la hace ideal para fines académicos.

II. Computación de Alto Rendimiento y Contenedores: Preliminares

El término *High Performance Computing* (HPC), deriva de la utilización de equipos de computación y de procesamiento de datos en paralelo para resolver problemas computacionales complejos. Hoy día, se utiliza para resolver problemas avanzados, investigación científica, realizar predicciones del clima a través de modelado, simulación y análisis de los datos. Los clústers HPC son comúnmente denominados *Supercomputadores*, dada la enorme capacidad de procesamiento que poseen.

[1] LXC es una interfaz de espacio de usuario para las características de contenerización del kernel de Linux. A través de una poderosa API y herramientas simples, permite a los usuarios de Linux crear y gestionar fácilmente contenedores de sistemas o de aplicaciones.

En el mundo de HPC, uno de los paradigmas más usados para la implementación de las aplicaciones distribuidas y paralelas es el de pase de mensajes. En este contexto, el estándar MPI se destaca como una de las tecnologías más comúnmente usadas para software científico, generalmente escrito en C/C++ y Fortran. MPI está diseñado para soportar el modelo de programación paralelo SPMD (*Single Process Multiple Data*), bajo el paradigma maestro-esclavo: un único programa se ejecuta en todos los nodos, cada uno procesando diferentes datos (nodos esclavos), bajo la coordinación de un coordinador (nodo maestro).

Sin embargo, desplegar un clúster MPI para HPC puede resultar una tarea retadora, demandante de tiempo y dedicación, por parte de los administradores de sistemas, para satisfacer las necesidades de los usuarios, muchas veces requiriendo distintas versiones de las diversas librerías existentes. De aquí, la necesidad de proveer soluciones capaces de abstraer la dificultad de tales tareas de administración, de manera automática, simple y eficiente. La reciente tecnología de contenedores, y en particular de Docker, se presenta como una alternativa atractiva. Docker permite crear contenedores Linux para encapsular aplicaciones y sus respectivas dependencias para poder ejecutarlas, aliviando al usuario final de dichas tareas, y en este caso a los administradores de sistemas de HPC. Docker además permite "orquestar" (i.e., controlar la ejecución de un conjunto de contenedores con un fin específico) aplicaciones seccionadas en microservicios a través de su herramienta `Swarm`, que crea al momento de su ejecución un clúster de alta disponibilidad, para mantener en línea la aplicación en todo momento. `Swarm` también permite la adición y sustracción de nodos, sin perder su estado, siempre y cuando exista un quorum de $(n/2) + 1$ nodos[2] (siendo "n" la cantidad de computadores pertenecientes al `Swarm`) que estén declarados como *Managers* del clúster de alta disponibilidad.

III. Trabajos Relacionados

Actualmente, existen trabajos que exploran las posibilidades que introduce la herramienta Docker en el campo de la computación de alto rendimiento. En [4] los autores presentan un estudio comparativo entre el uso de máquinas virtuales y contenedores Docker para desplegar aplicaciones HPC, usando *benchmarks* bien conocidos. Sus resultados confirman las ventajas y beneficios que representan ambas técnicas en el mundo de HPC. Otros trabajos se concentran en mostrar cómo se puede integrar ambientes de MPI en clúster HPC a través de contenedores Docker [7].

Otros estudios más relacionados a este trabajo proponen soluciones para desplegar clúster de HPC con contenedores Docker. El trabajo presentado en [1] describe una solución con contenedores Docker para encapsular el ambiente de los nodos de un clúster, que al igual que nuestra solución, ofrece auto escalamiento de nodos y permite que varias instancias del clúster pueden coexistir. Sin embargo, difiere de nuestra solución en los siguientes aspectos: (i) fue ejecutado y concebido para hardware de clúster, en lugar de computadres personales; (ii) utiliza la red bridge; (iii) utiliza cónsul, un servicio distribuido de descubrimiento de servicios, para mantener la lista de nodos activos/conocidos en el archivo de host de mpi; el servicio está presente fuera de la solución, es decir, no está dentro de los contenedores; (iv) no posee soporte de almacenamiento centralizado; y (v) no utiliza un orquestador de contenedores, ni servicios, no está integrado ni automatizado el proceso de despliegue del clúster; sólo está automatizado el proceso de construcción del ambiente.

En [2] se realiza un análisis investigativo y exhaustivo sobre las implicaciones y decisiones técnicas, que conlleva desplegar un ambiente de HPC basado en contenedores de Docker, sin embargo plantean la implementación del concepto en un trabajo posterior [12]. Similar a nuestra propuesta, este trabajo, utiliza los contenedores de Docker para encapsular el ambiente de los nodos y un *script* de inicialización de los nodos. Las principales diferencias son: (i) la solución fue ejecutada y concebida para hardware de clúster; (ii) presenta un análisis más profundo sobre el impacto de Docker sobre el rendimiento en comparación con el hypervisor `kvm` y concluye que el *overhead* entre un ambiente con contenedores y uno real, es mínimo; (iii) soporta dos arquitecturas para el manejo de los procesos de MPI, una con un contenedor por nodo y otra con un contenedor por proceso (similar a nuestra solución propuesta); (iv) usan el *benchmark* de la librería LINPACK [13]; (v) no utiliza un orquestador de contenedores, ni servicios, no está integrado ni automatizado el lanzamiento, sólo la construcción del ambiente; y (vi) no posee soporte de almacenamiento centralizado.

Los autores del trabajo presentado en [3], a diferencia del presentado en [2], determinaron en sus estudios (y en base a otros estudios previos referenciados en su trabajo) que sí existe un cuello de botella al ejecutar trabajos de MPI en ambientes de múltiples contenedores. Por lo que decidieron estudiar y diagnosticar las causas de este *overhead*, para luego desarrollar una librería que permita a los procesos de MPI comunicarse a través de memoria compartida, en lugar de por mensajes de red. Ese proyecto, al igual que los anteriores mencionados y el nuestro, utiliza contenedores Docker para encapsular el ambiente de los nodos. Las principales diferencias con nuestra solución son: (i) fue ejecutado y concebido para hardware de clúster, especialmente para ambientes en la nube; (ii) usa *Runtime Privilege*[3], lo que permite darle accesos al contenedor a todos los dispositivos y procesos del host; esto podría utilizarse en la solución propuesta en este trabajo, para darle acceso al contenedor (un nodo MPI como el maestro) al demonio de docker (encargado de desplegar la solución); un beneficio de ésto, sería que le otorgaría al maestro la capacidad de desplegar, apagar o destruir nodos del clúster sin tener que salir de la sesión o de la instancia de clúster; en el caso del trabajo presentado en [3], se utiliza esta característica para darle acceso a los contenedores a los dispositivos infiniband, esto para obtener las capacidades de

[2]Documentación sobre Algoritmo de consenso `Raft`: https://docs.docker.com/engine/swarm/raft

[3]Documentación sobre Runtime Privilege: https://docs.docker.com/engine/reference/run/#runtime-privilege-and-linux-capabilities

acceso a memoria compartida mencionadas anteriormente.

Un trabajo reciente en el contexto de procesamiento de imágenes de teledetección [14] utiliza la tecnología Docker para implementar un entorno HPC, necesitando solamente 5 minutos en la configuración de 16 servidores (*clusters*) en comparación de un día de trabajo en una configuración manual. La arquitectura consiste en un administrador de flujo de trabajo, un administrador de recursos y un calendarizador. En [15], los autores proponen Sarus, una herramienta para sistemas HPC que crea una instancia de contenedores con muchas funciones a partir de imágenes de Docker. Administra completamente el ciclo de vida del contenedor al extraer imágenes de los registros, administrar el almacenamiento de imágenes, crear el sistema de archivos raíz del contenedor y configurar *hooks* y espacios de nombres.

El proyecto presentado en [5] trata sobre una imagen basada en alpine con la implementación de MPICH, llamada `alpine-mpich`. El estudio de la arquitectura de esta solución permitió desarrollar más a fondo la imagen generada en el presente trabajo, llamada `ubuntu-openmpi`. En nuestra propuesta se utilizan algunas aspectos de [5] tales como el auto escalamiento de nodos, el uso de un *script* de inicialización de los nodos, la posibilidad de la coexistencia de instancias del clúster, el uso de contenedores Docker para encapsular el ambiente de los nodos, uso de un orquestador de contenedores, servicios, y la integración y automatización del lanzamiento y la construcción del ambiente. Así mismo, difiere en los siguientes aspectos: (i) no posee soporte de almacenamiento centralizado, de manera de que cuando se requiera ejecutar nuevos *scripts* se debe volver a construir la imagen de la solución; (ii) no posee soporte para ejecución de *scripts* en python; (iii) posee solamente un ambiente de desarrollo de aplicaciones distribuidas (alpine linux con la librería MPICH). Mientras que la solución aquí propuesta, además de ofrecer ese mismo ambiente, ofrece una versión basada en Ubuntu con la librería OpenMPI.

Se puede observar que la mayoría de los trabajos considerados se enfocan en la optimización, puesta en producción y el rendimiento del clúster. Esto es deseable, sin embargo, todos excepto el trabajo presentado en [5], requieren de hardware especializado para su utilización. De allí el valor de nuestra solución, ya que es más accesible y sencilla de utilizar, en equipos de cómputo de uso cotidiano.

IV. Arquitectura de la Solución Propuesta

La utilización de la herramienta Docker permite diseñar un clúster con una arquitectura distinta a la de un clúster tradicional, como puede observarse en la Figura 1. Además, ofrece una manera más ordenada de declarar los componentes de la arquitectura. El *script* de lanzamiento utiliza el archivo de despliegue para desplegar cualquier versión del clúster, en nuestro caso `alpine-mpich` o `ubuntu-openmpi`, como lo muestra la Figura 2. A continuación se describe brevemente los principales elementos de la arquitectura de un clúster HPC, desplegado en contenedores Docker.

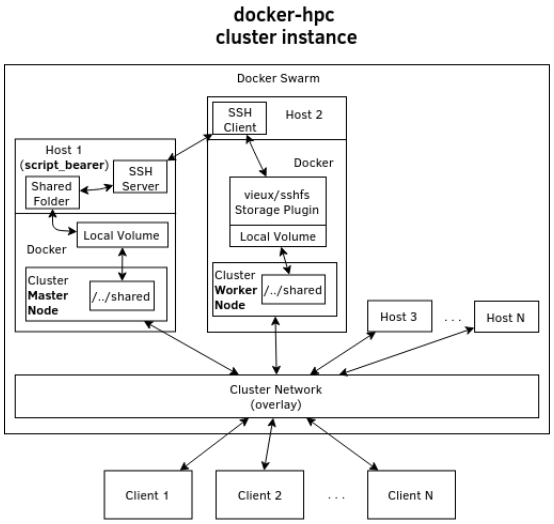

Figure 1: Despliegue de una Instancia del Clúster a lo Largo del Swarm

- **Contenedores, servicios y `stacks`:** Cada nodo es encapsulado en un contenedor que contiene todas las herramientas necesarias para su funcionamiento, dándole autonomía y aislamiento. Estos contenedores pueden pertenecer a dos servicios, uno que despliega el nodo maestro y otro que despliega un nodo esclavo junto con sus réplicas. A su vez, estos servicios forman parte de un `Stack`. Un `stack` es un grupo de servicios interrelacionados que pueden compartir dependencias o pueden ser orquestados para ser escalados en conjunto[4]. Ambos servicios pueden ser desplegados automáticamente utilizando un archivo de configuración `yaml` en donde éstos son definidos. Esto permite además que los servicios de los esclavos puedan ser replicados a voluntad, agregando o disminuyendo recursos al clúster.

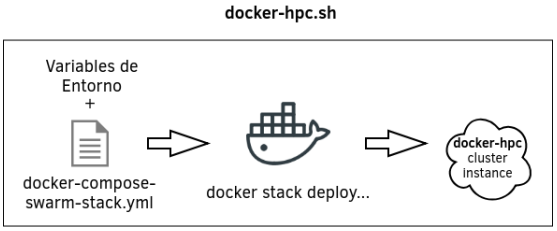

Figure 2: Estructura del *Script de* Lanzamiento del Clúster

- **Comunicación:** La comunicación entre estos nodos se realiza a través de una red virtual llamada `red overlay`, nativa de la herramienta Docker, que segmenta la red de manera lógica. Sin embargo, ésta no necesariamente tiene que ser local.
- **Almacenamiento:** Es deseable que en un ambiente de clúster exista un almacenamiento común para todos los nodos, de manera que los archivos fuente a ser ejecutados de manera paralela, estén disponibles para cada uno de ellos. Dado que los contenedores se ejecutan en diferentes computadores en un Swarm, la capacidad de Docker de

[4]Documentación sobre Stack en Docker: https://docs.docker.com/get-started/part5

montar `volúmenes` (directorios locales del host que son montados en una ruta dentro del contenedor) no podría usarse como almacenamiento común. Cada nodo tendría una versión local del punto de montaje diferente.

La centralización del almacenamiento en un Swarm no es una característica o funcionalidad nativa de Docker. El desarrollo de dicha característica se encuentra fuera del alcance de este trabajo. Existen herramientas que implementan este requerimiento y pueden instalarse en forma de *plugins* de Docker, y dan soporte especializado a diferentes plataformas como Amazon Web Services u otros programas de orquestación de contenedores como Kubernetes[5]. Estos *plugins*, en la mayoría de los casos, requieren una instalación y configuración que no es amigable para usuarios inexpertos, o requiere de pasos extra que aumentan la complejidad de la instalación. Para que esta solución sea verdaderamente simple de utilizar y portable, se eligió un complemento muy simple de instalar y configurar: `docker-volume-sshfs` [16]. Se encarga de montar un directorio en un servidor remoto a través de la herramienta `sshfs`, y lo hace disponible como un volumen local para el uso de los contenedores (nodos trabajadores).

Otro beneficio de utilizar este *plugin* es que es compatible con el ambiente `Playwithdocker`, un *sandbox* temporal para la ejecución de proyectos que utilizan la herramienta Docker y su orquestador Swarm. Permitiendo desplegar la solución de este proyecto en un par de minutos. Es importante acotar que cuando el Swarm posee un solo nodo, el *script* de lanzamiento le indica a los nodos del clúster que utilicen un volumen local, ya que no tiene sentido montar de manera remota un directorio local.

A continuación se describen las diferentes versiones disponibles en la solución desarrollada.

V. Imágenes Docker Desarrolladas: Variantes de Clústers HPC

En este trabajo desarrollamos dos variantes de clústers HPC, en cuanto a sistema operativo base y librería de paso de mensajes MPI implementada: `alpine-mpich` y `ubuntu-openmpi`. Es importante destacar que en un proyecto de Docker pueden existir diferentes etiquetas o `tags` que permiten diferenciar las versiones entre sí, de acuerdo a criterios definidos por los desarrolladores como: tipos de herramientas presentes en la imagen, configuración, tamaño, entre otros. Para diferenciar las imágenes, se definió una nomenclatura basada en los componentes y herramientas más importantes en cada una. La nomenclatura es la siguiente:

xxx-yyy[-ppp][-zzz], donde:

- **xxx**: describe el sistema operativo base.
- **yyy**: indica la principal herramienta o ambiente de trabajo que se ejecutará al desplegar el clúster.
- **ppp**, **zzz**: representa la versión o acabado de la imagen (por ejemplo: *latest, base, onbuild*).

[5]Documentación sobre plugins en Docker: https://docs.docker.com/engine/extend/legacy_plugins/#volume-plugins

Los parámetros encerrados entre corchetes son opcionales. Por ejemplo, la imagen con el sistema operativo Ubuntu, con las herramientas `OpenMPI` y `Slurm` instaladas, se representaría como **ubuntu-openmpi-slurm-latest**, según esta nomenclatura. A continuación describimos las variantes que forman parte del los de ambientes disponibles en nuestra solución.

A. Versión alpine-mpich

Este ambiente despliega un clúster con el sistema operativo Alpine Linux[6], una distribución de Linux muy popular entre los desarrolladores de imágenes para Docker [17], [18] y los que necesitan una utilización óptima de recursos con un mínimo impacto en la seguridad y espacio en disco [19].

Sobre esta distribución se realiza la instalación de la implementación del estándar MPI, MPICH en la versión 3.2. La imagen correspondiente a esta versión, se basa en la propuesta en el trabajo presentado en [5] y disponible en [20]. A la versión disponible en [20], se le hicieron adaptaciones y extensiones para agregar funcionalidades y solucionar algunos problemas presentados en el despliegue (e.g., el clúster no queda en el estado deseado, ya que el nodo maestro del clúster es incapaz de comunicarse con los nodos esclavos). Las modificaciones realizadas se detallan a continuación.

- Solución de problemas de despliegue. Aunque la imagen disponible en [20] estaba correctamente armada, al desplegar el proyecto de acuerdo a la documentación, el nodo maestro del clúster no era capaz de comunicarse con los esclavos. Este problema fue ocasionado porque el *script* que mantiene actualizado el archivo de hosts, utilizaba la herramienta `nslookup` para mapear los esclavos existentes en la red (*overlay network*). De acuerdo a la documentación de Docker[7], para poder determinar cuáles son los hosts activos dentro de una *overlay network*, se tiene que consultar al dominio `tasks.`⟨`nombre del servicio`⟩. Sin embargo, ésto solo funcionó para un máximo de 10 hosts. Al escalar el servicio por encima de este número, el resto de los nodos no podía ser encontrado. Según algunas publicaciones, `nslookup` es una herramienta considerada antigua y en algunos casos obsoleta[8,9]. En esos trabajos se recomienda que se suspenda su uso y en su lugar se utilice la herramienta `dig`, que no posee la limitación antes mencionada. Luego de esta modificación la instancia del clúster fue capaz de reconocer todos los nodos de la red.
- Soporte de ejecución de *scripts* de python. Se instalaron las versiones 2.7 y 3 del *runtime* de Python, junto con su manejador de paquetes `pip` [21]. Finalmente, utilizando `pip`, se instaló el módulo para ejecución de tareas en paralelo `mpi4py` [22].

[6]https://alpinelinux.org

[7]https://docs.docker.com/network/overlay/#container-discovery

[8]https://www.linuxquestions.org/questions/linux-networking-3/why-nslookup-is-deprecated-122337

[9]http://blog.smalleycreative.com/linux/nslookup-is-dead-long-live-dig-and-host

B. Versión ubuntu-openmpi

A diferencia del ambiente anterior, en éste despliega un clúster con Ubuntu, un sistema operativo más comercial que Alpine Linux. En esta versión se encuentra instalada la implementación OpenMPI. La imagen utilizada de base para esta versión, forma parte del proyecto `Dispel4Py` disponible en [10]. Las adaptaciones y modificaciones realizadas a ese proyecto se describen a continuación.

- Reducción del tamaño en disco de la imagen original. Se eliminaron diversos paquetes que no son necesarios para la implementación correspondiente en este trabajo, permitiendo la creación de una imagen más ligera y en consecuencia, más fácil de descargar en los nodos a utilizar.
- Corrección en la exclusión e inclusión de interfaces de red para OpenMPI. La configuración de OpenMPI en los nodos carecía de parámetros de configuración de la capa de envío de datos de mensajes MPI (BTL por las siglas en inglés de *Byte Transfer Layer*) para interfaces TCP, causando *hangups* (interrupción de envío de mensajes) en MPI, al no "saber" qué interfaz utilizar para poder enviar y recibir los mensajes en paralelo. También se realizó la configuración en todos los nodos, de la interfaz a utilizar para los mensajes de control *Out-of-band* (OOB).
- Limitación de los procesos multi-hilo: Se estableció la configuración necesaria para que OpenMPI no trabaje en modo multi-hilo y permita dedicación exclusiva de sus nodos.
- Igual a lo corregido en la imagen de Alpine Linux, los nodos no eran capaces de interconectarse por una falla en la resolución de nombres y hosts. La falla se corrigió de la misma manera que para la imagen de `alpine-mpich`.
- Luego de estudiar el funcionamiento de la imagen de `alpine-mpich`, se desarrollaron una serie de *scripts* que le permiten a la instancia de esta imagen, mantener actualizado el archivo de *hosts* a medida que los nodos se incorporan o son eliminados del clúster.

Luego de haber culminado con el desarrollo de las imágenes, para realizar pruebas y comprobar el correcto funcionamiento del clúster desplegado, se utilizaron dos programas sencillos de prueba: `mpi-hello-world.py` y `ring.py`. Estos programas se ejecutaron para probar la conectividad de los nodos esclavos con el maestro (`mpi-hello-world.py`) y para probar el correcto funcionamiento del envío y recepción de mensajes (en `ring.py` los nodos se comunican siguiendo una topología lógica de anillo). Dichos programas resultaron en ejecuciones exitosas, lo que permitió proseguir al despliegue de la solución y ejecución de pruebas sintéticas para evaluar el rendimiento, cuyos resultados se encuentran en la Sección VII.

VI. Despliegue de la Solución

En esta sección se describen los pasos a seguir para que un usuario pueda instalar, desplegar y utilizar un clúster en sus equipos personales. Antes de realizar el despliegue de alguna de las versiones disponibles en nuestra solución, es necesario instalar una versión de Docker superior a 1.13 (preferiblemente 18.06 en adelante). Si se quiere aprovechar el poder de cómputo de varios hosts, el programa debe estar presente en todos ellos, para aprovechar el orquestador de contenedores Docker Swarm[10]. Adicionalmente se requerirá que al menos uno de los nodos cuente con un servidor `ssh` activo. Finalmente se deben seguir los pasos que se muestran a continuación para asegurarse de que el *plugin* de almacenamiento está instalado y funciona correctamente.

A. Preparación del Ambiente

Para explicar el proceso de despliegue se usarán dos máquinas: `bob` y `alice`. Ambas corriendo el sistema operativo Linux. `bob` albergará un contenedor con el nodo maestro del clúster (y es el que debe tener un servidor `ssh` activo de manera nativa) y `alice` hospedará los contenedores ejecutándose como nodos esclavos (debe poder tener acceso como cliente al servidor `ssh`, esto es necesario para que el *plugin* funcione correctamente). En la Figura 3 se ejemplifica este proceso. `bob` posee un archivo `cureofcancer.c`, al cual `alice` quiere tener acceso para utilizarlo dentro de sus contenedores, además desea que los cambios sean persistentes. Utilizando el *plugin* de almacenamiento podrá hacer disponible ese directorio remoto a otros contenedores locales, debido a que este crea un volumen local de `docker` que mantiene los cambios en la computadora de `bob` luego de que los contenedores de `alice` hayan sido destruidos.

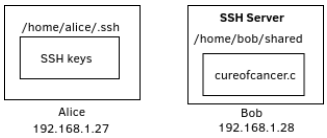

Figure 3: Elementos Presentes en los *Hosts* para Instalación del *Plugin*

En la máquina cliente (`alice`) se deberán ejecutar los siguientes comandos:

1) `ssh-keygen`
2) `ssh-copy-id bob@192.168.1.28`
3) `docker plugin install --grant-all-permissions vieux/sshfs sshkey.source=/home/alice/.ssh`
4) `docker volume create -d vieux/sshfs -o sshcmd=bob@192.168.1.28:/home/bob/shared sshvolume`
5) `docker run -it -v sshvolume:/brilliant busybox ls /brilliant`

La secuencia de comandos anterior, en el caso de las máquinas `bob` y `alice`, deberá generar la siguiente salida (obviando la salida de los comandos de instalación):

```
alice@client /home/alice $

cureofcancer.c
```

[10]https://docs.docker.com/get-started/part4/#prerequisites

B. Pasos para Levantar el Clúster HPC

Continuamos utilizando como referencia los computadores de `bob` y `alice`.

1) Iniciar el Swarm en el host que posee el servidor `ssh` activo (`bob`) con la orden `docker swarm init`; si éste posee más de una interfaz de red deberá especificar cuál desea utilizar mediante el parámetro `--advertise-addr string`, en donde `string` deberá ser la dirección `ip` o el nombre de la interfaz. Si se desea utilizar los recursos de varios equipos, cada uno deberá unirse al Swarm previamente creado utilizando el token generado por `bob` en este paso (ver Figura 4).
2) Clonar el código del repositorio que contiene las versiones de la solución, disponible en [11] en el nodo que posee el servidor `ssh` activo (`bob`). El *script* de lanzamiento permite elegir y lanzar posteriormente alguna de esas versiones (ver Figura 5).

```
[alice@client ~]$ docker swarm join --token SWMTKN-1-3hdjmpj14cfucusaolgz6o
6uvnz4qv6holfegq3ynvtq6srhnx-7uhxmapabglv4ni47656yfj3m 192.168.1.28:2377
This node joined a swarm as a worker.
[alice@client ~]$
```

Figure 4: Agregando Nodos al Swarm

3) Como se detalla en la documentación del repositorio de este proyecto [23], el *script* `docker-hpc.sh` sin parámetros, desplegará un clúster con el ambiente Alpine Linux y la implementación del estándar MPI, MPICH, provisto de un maestro y cuatro nodos esclavos (Figura 5). En caso de que se quiera modificar las imágenes que componen a los ambientes de la solución, será necesario modificar el usuario y repositorio en el archivo de configuración `docker-hpc.conf` y proveer los parámetros para construir y subir la imagen al nuevo repositorio de DockerHub.
   Al finalizar la descarga o construcción y publicación de la imagen, el *script* esperará a que el nodo maestro esté disponible (Figura 6) y se autenticará automáticamente utilizando las llaves SSH presentes en el directorio de cada variante (Figura 7). Si se desea lanzar el clúster con la versión de `ubuntu-openmpi`, se debe especificar la bandera `-i` seguida del nombre de la versión a utilizar, en ese caso `ubuntu-openmpi` (ver Figura 8).
4) Finalmente, si se desea eliminar la instancia del clúster antes creada, se pasa el parámetro `-d` al *script* de lanzamiento, seguido del nombre de la imagen utilizada para crear el ambiente. Si no se provee ningún parámetro, por defecto se utiliza `alpine-mpich`.
   Si se desea aumentar o reducir el tamaño del clúster, tan soló es necesario proveer la bandera `-s` seguida del número de trabajadores deseado como lo muestra la Figura 9. Esto también se puede conseguir en el paso 3 utilizando la bandera `-n` junto con un número de réplicas.

```
[bob@server docker-hpc-cluster]$ ./docker-hpc.sh
                    Developed by:
          Gianfranco Verroccht & José Hernández
===> Importing Variables from docker-hpc.conf
===> No parameters specified
===> Staring cluster using the default parameters...
===> Environment variables:
IMAGE_NAME=gianv9/docker-hpc:alpine-mpich-latest
STACK_TAG=alpine-mpich
REPLICAS=4
The current dir is /home/bob/docker-hpc-cluster/alpine-mpich/cluster
This node ip is 192.168.1.28
```

Figure 5: Ejecutando el *Script* de Lanzamiento del Clúster

```
This node ip is 192.168.1.28
This user is bob@

===> Tagging node (lkfejr23pb5ljac2cdg53s83p) as 'script_bearer=true' before deployment
lkfejr23pb5ljac2cdg53s83p

===> Node tagged:
map[script_bearer:true]

 ------------------------------------------------------------------
|            ##        .                  ##        .             |
|      ## ## ##       ==               ## ## ##       ==           |
|   ## ## ## ##      ===             ## ## ## ##      ===         |
|  /"""""""""""""""\___/ ===       /"""""""""""""""\___/ ===      |
|~~~{~~ ~~~~ ~~~ ~~~~ ~~ ~ /  ===-~~~{~~ ~~~~ ~~~ ~~~~ ~~ ~ /  ===-~~~|
|     \_____ o          __/            \_____ o          __/       |
|       \    \        __/               \    \        __/         |
|        \____\______/                   \____\______/             |
|          M A S T E R                     W O R K E R x 4         |
 ------------------------------------------------------------------

===> Waiting for previous alpine-mpich cluster to die...
Removing network alpine-mpich_default

===> Waiting for previous cluster networks to die...
........
===> Deploying the cluster stack alpine-mpich with 4 workers...
Creating network alpine-mpich_default
Creating service alpine-mpich_master
Creating service alpine-mpich_worker

===> Waiting for the master node to spawn...
===> Press CTRL-C if automatic login does not occur
===> Then login by using 'docker-hpc.sh -l alpine-mpich'
```

Figure 6: Estado del Despliegue del Clúster

```
===> Cluster Master Node is Ready

Welcome to Alpine MPICH Cluster!

Try this to see all host names in the cluster:
$ mpirun hostname

You don't need to provide host file to MPI because the default host file
of connected nodes in the cluster is automatically updated at /etc/opt/hosts

To obtain hostfile manually and provide to MPI by yourself:
$ get_hosts > hosts
$ mpirun -f hosts hostname

For more information: https://github.com/NLKNguyen/alpine-mpich/
1cfa9b80cc48:/project$
```

Figure 7: Instancia de Clúster `alpine-mpich` Desplegada Exitosamente

VII. Resultados Experimentales

Para la evaluación de nuestra propuesta, se desplegó una instancia de cada versión de la solución, en dos ambientes: un ambiente con una red local (tres nodos heterogéneos pertenecientes a un Swarm clúster interconectados vía cables Ethernet) y un ambiente remoto con máquinas virtualizadas utilizando el ambiente *Play With Docker*[11] (cinco nodos homogéneos).

Los tres computadores usados para el ambiente con la red local son `Link`, `Falco` y `Wolf`. Los tres son *Managers* del Swarm debido a que es requerido por Docker para su correcto funcionamiento. En el ambiente remoto con máquinas virtuales, se usó una plantilla pre-existente de tres *Managers* y dos trabajadores para el `Swarm`.

En la Figura 10 se muestra las especificaciones de los equipos utilizados para desplegar la instancia local del clúster (primera fila y segunda fila con primera columna), junto con las especificaciones de las instancias de *Play With Docker* (segunda fila con segunda columna).

En cada versión de la solución se utilizó la herramienta de pruebas sintéticas OSU Microbenchmarks, *benchmark* de MPI que mide el desempeño de varias operaciones [24]:

[11] https://labs.play-with-docker.com – *Play With Docker* es una plataforma en línea, que ofrece de forma gratuita contenedores trabajando como instancias de máquinas virtuales.

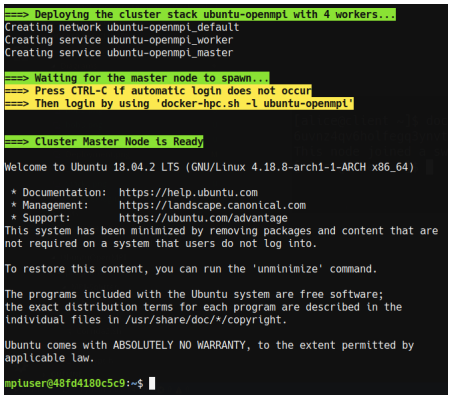

Figure 8: Instancia del Clúster `ubuntu-openmpi` Desplegada Exitosamente

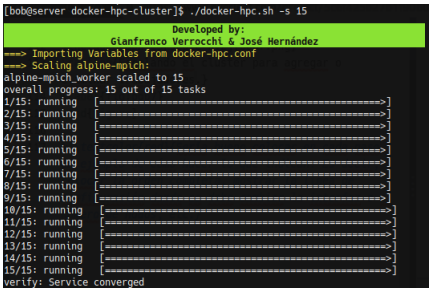

Figure 9: Escalando el Clúster para Agregar o Eliminar Trabajadores

- *collective*: Pruebas sintéticas de funciones colectivas de MPI.
- *one-sided*: Pruebas sintéticas de funciones unilaterales y bidireccionales de MPI.
- *point-to-point*: Pruebas sintéticas de funciones punto a punto de MPI.
- *startup*: Pruebas de arranque y chequeo de librerías.

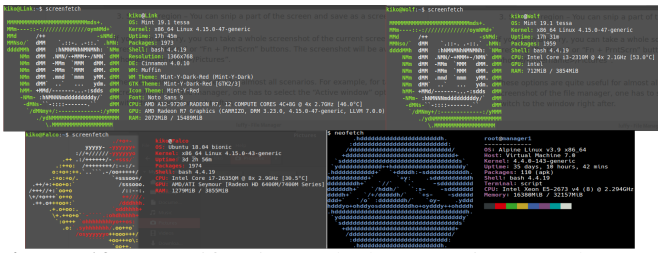

Figure 10: Especificaciones de las Máquinas Usadas para el Ambiente de Red Local: `Link`, `Falco` y `Wolf`, y el Ambiente Remoto: `Play With Docker`

Para el lanzamiento, en ambos casos el clúster que fue desplegado usando el *script* de la Figura 2, y los pasos descritos en la Sección VI. El clúster desplegado está compuesto de cinco contenedores: un coordinador y cuatro esclavos de MPI.

Las Figuras 11 a 21 muestran de manera gráfica los resultados de la ejecución de las pruebas más significativas (aquellas que prueban funciones referenciales de MPI como *scatter* y *broadcast* o que presentan diferencias significativas en cuanto a latencia entre las ejecuciones en los dos diferentes ambientes), usando la versión `alpine-mpich`. No se muestran los resultados de la versión `ubuntu-openmpi` dado que son muy similares. Los tiempos y el ancho de banda están expresados en base logarítmica para apreciar mejor los resultados.

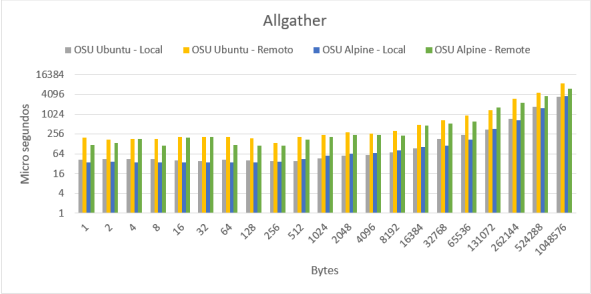

Figure 11: Prueba *allgather* con OSU Microbenchmarks

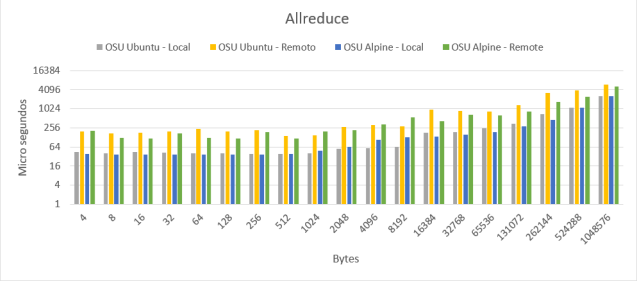

Figure 12: Prueba *allreduce* con OSU Microbenchmarks

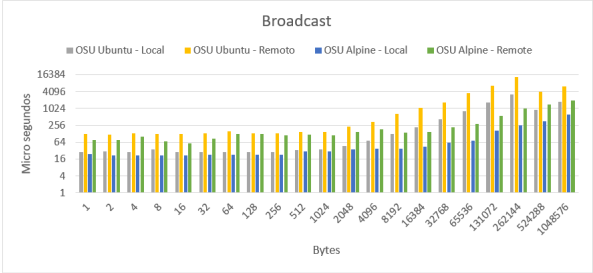

Figure 13: Prueba *broadcast* con OSU Microbenchmarks

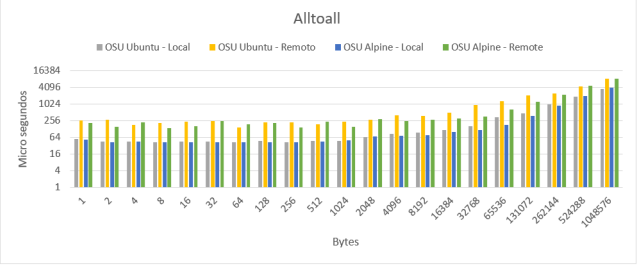

Figure 14: Prueba *alltoall* con OSU Microbenchmarks

Los resultados detallados de éstas y el resto de las pruebas, así como los obtenidos en `ubuntu-openmpi` de la ejecución de OSU Microbenchmarks y otro *benchmark*, Intel MPI Benchmarks)[12], que por incompatibilidades no se puedo ejecutar en

[12] https://software.intel.com/content/www/us/en/develop/articles/intel-mpi-benchmarks.html

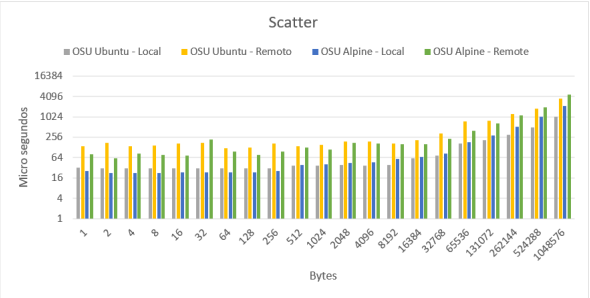

Figure 15: Prueba *scatter* con OSU Microbenchmarks

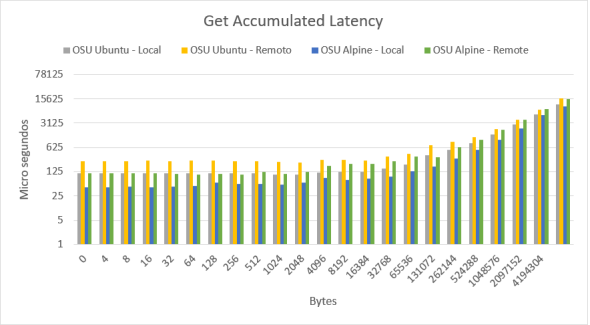

Figure 16: Prueba *get accumulated latency* con OSU Microbenchmarks

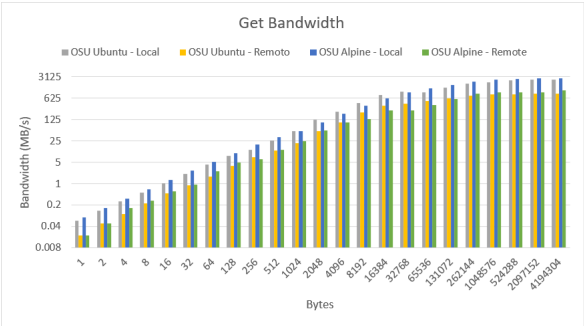

Figure 17: Prueba *get bandwidth* con OSU Microbenchmarks

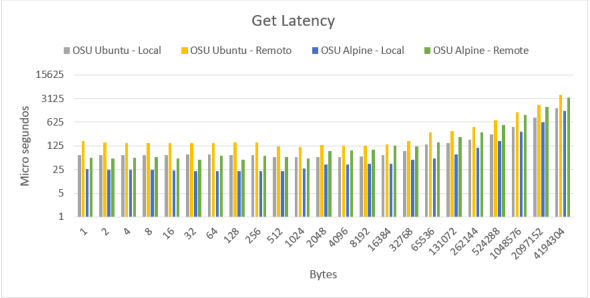

Figure 18: Prueba *get latency* con OSU Microbenchmarks

la versión `alpine-mpich`, se encuentran disponibles en el repositorio de este proyecto [23].

Los OSU Microbenchmarks fueron lo suficientemente ligeros para ejecutarlos sin problema a través de ambos ambientes de prueba y con ambas versiones de la solución. Los resultados muestran que para el ambiente de red local hay menos latencia y mejor desempeño de las operaciones MPI, en particular para cargas bajas. Esto se debe principalmente a que el ambiente `PlayWithDocker` utiliza una arquitectura `Docker In Docker`, que permite ejecutar una instancia de Docker dentro de un contenedor, en este caso para dar el efecto de tener múltiples máquinas virtuales. Se presume que ésto contribuyó al aumento de la latencia. Sin embargo, el ambiente remoto virtualizado permitió localizar mejor los recursos dada la homogeneidad de los nodos del clúster deplegado, mientras que la distribución de las tareas en los nodos heterogéneos de la red local no fue balanceada, dada la diferencia en la cantidad de núcleos de los procesadores utilizados. Adicionalmente, es importante enfatizar que los resultados obtenidos fueron similares para ambas versiones disponibles en nuestra solución: `ubuntu-openmpi` y `alpine-mpich`.

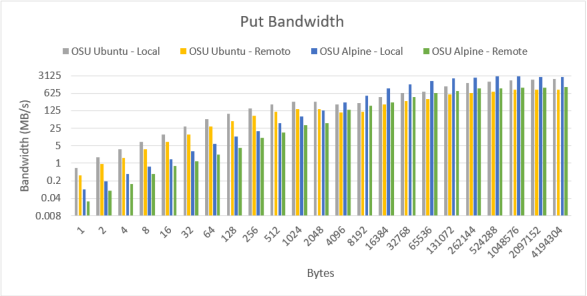

Figure 19: Prueba *put bandwidth* con OSU Microbenchmarks

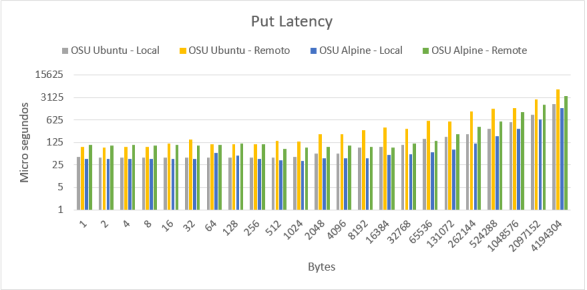

Figure 20: Prueba *put latency* con OSU Microbenchmarks

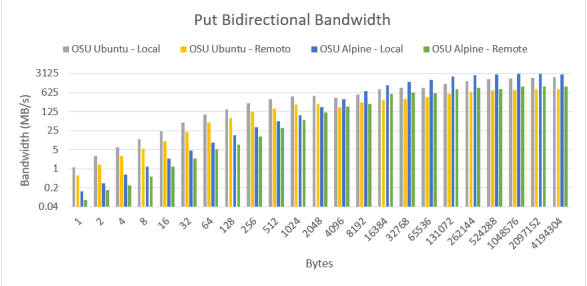

Figure 21: Prueba *bidirectional bandwidth* con OSU Microbenchmarks

Estos resultados demuestran que nuestra solución puede realizar trabajos pesados en ambos tipos de ambiente, sin embargo es más recomendable utilizar el ambiente remoto virtualizado para cargas pesadas debido a su homogeneidad, balance de carga, mientras que el ambiente local es ideal para cargas ligeras puesto que no tiene la carga extra de la virtualización. Para casi todos los comandos mpi, Ubuntu y Alpine Remotos son más lentos que Ubuntu y Alpine locales, como es de esperarse. Sin embargo, para unos pocos comandos resulta

lo inverso. Estos resultados es necesario verificarlos con más pruebas para identificar las razones de este comportamiento, que en principio pueden ser causados por accesos locales a discos que resultan más lentos que el acceso a la red.

Aún cuando la solución que proponemos puede ser usada en cualquier hardware disponible o incluso en un ambiente virtualizado, su uso está orientado a fines académicos más que para ambientes de producción reales. En este sentido, nuestra solución de despliegue automático de un cluster HPC basada en contenedores Docker, no pretende competir con un ambiente de clúster tradicional HPC en términos de velocidad o capacidad de procesamiento, dada la naturaleza del hardware sobre el cual fue probado y para el cual fue concebido (computadores personales interconectados principalmente a través de cables Ethernet). Sin embargo, su arquitectura le otorga las siguientes características (la mayoría no disponibles en un ambiente de clúster tradicional), que en un ambiente académico son de mucho beneficio:

- **Mayor accesibilidad:** Facilita el aprendizaje de conceptos de computación distribuida a estudiantes e investigadores que generalmente, no tienen acceso a un ambiente de este tipo.
- **Ambiente de pruebas portable (*sandbox*):** Debido a que Docker soporta la mayoría de los sistemas operativos, es posible ejecutar una o más instancias del clúster en cualquiera de ellos. Permitiendo a los desarrolladores ejecutar sus programas independientemente de la plataforma en la que se encuentren. Esto es cierto siempre y cuando sus computadores tengan un sistema operativo de 64 bits, bien sea Linux (soporte nativo de contenedores `lxc`) o posean soporte de virtualización en Windows o Mac (`docker-toolbox` y `docker-for-mac`, respectivamente).
- **Maleable, dinámico:** Modificar la configuración de lanzamiento (*script* de lanzamiento y archivo `yaml` (estándar de serialización de datos amigable), disponible para todos los lenguajes de programación o las herramientas y configuraciones del sistema (`Dockerfile`), bastará para modificar la arquitectura y ambiente del clúster. Además, la herramienta de orquestación de contenedores, Docker Swarm, se encarga de utilizar los recursos disponibles, incluso si varias en el tiempo (se agregan o se quitan computadores), manteniendo consistente el estado del clúster.

VIII. Conclusiones y Trabajo Futuro

Es este trabajo presentamos una solución de software basada en contenedores Docker para el despliegue automático de clústers HPC. Se generaron dos imágenes posibles de despliegue: `alpine-mpich` o `ubuntu-openmpi`. Presentamos la estrategia que usamos para generar las imágenes, cuyo enfoque puede ser usado para incorporar otras imágenes con diferentes sistemas operativos y librerías de comunicación. Ambas versiones se evaluaron en dos escenario diferentes, con una red local y con nodos virtualizados en un servidor remoto. Luego de estudiar los resultados obtenidos, se puede apreciar que la solución desarrollada es más recomendable para entornos académicos que para entornos investigativos, sin embargo es de gran utilidad para ambos casos. Existe el caso borde de poseer una arquitectura física lo suficientemente potente y homogénea, donde sea posible realizar el despliegue y cumplir con los requerimientos de cargas mucho más pesadas fácilmente. En ese caso se podría alcanzar un nivel de ambiente de producción, integrando la librería [3].

Actualmente trabajamos en la integración de la herramienta `slurm`, para generar una nueva versión. También se esta investigando sobre el despliegue de un registro local con las imágenes del clúster, además de la inclusión una interfáz web de usuario para gestión de instancias del clúster.

References

[1] H.-E. Yu and W. Huang, "*Building a Virtual HPC Cluster with Auto Scaling by the Docker*," CoRR: arXiv preprint arXiv:1509.08231, vol. abs/1509.08231, pp. 1–4, 2015.

[2] J. Higgins, V. Holmes, and C. Venters, "*Orchestrating Docker Containers in the HPC Environment*," in International Conference on High Performance Computing, pp. 506–513, Springer, 2015.

[3] J. Zhang, X. Lu, and D. K. Panda, "*High Performance MPI Library for Container-based HPC Cloud on InfiniBand Clusters*," in 2016 45th International Conference on Parallel Processing (ICPP 2016), pp. 268–277, IEEE, 2016.

[4] M. T. Chung, N. Quang-Hung, M.-T. Nguyen, and N. Thoai, "*Using Docker in High Performance Computing Applications*," in 2016 IEEE Sixth International Conference on Communications and Electronics (ICCE 2016), pp. 52–57, IEEE, 2016.

[5] N. Nguyen and D. Bein, "*Distributed MPI Cluster with Docker Swarm Mode*," in 2017 IEEE 7th annual Computing and Communication Workshop and Conference (CCWC 2017), pp. 1–7, IEEE, 2017.

[6] A. Azab, "*Enabling Docker Containers for High-performance and Many-task Computing*," in 2017 IEEE International Conference on Cloud Engineering (IC2E 2017), pp. 279–285, IEEE, 2017.

[7] M. de Bayser and R. Cerqueira, "*Integrating MPI with Docker for HPC*," in 2017 IEEE International Conference on Cloud Engineering (IC2E 2017), pp. 259–265, IEEE, 2017.

[8] N. Zhou, Y. Georgiou, M. Pospieszny, L. Zhong, H. Zhou, C. Niethammer, B. Pejak, O. Marko, and D. Hoppe, "*Container Orchestration on HPC Systems through Kubernetes*," *Cloud Computing*, vol. 10, pp. 1–14, Dec. 2021.

[9] N. Nguyen, "*Alpine MPICH*," May 2018.

[10] R. Filgueira, "*docker.openmpi*," Aug. 2015.

[11] G. A. Verrocchi Carias and J. F. Hernández Balestrini, "*docker-HPC-cluster*," Feb. 2019.

[12] J. Higgins, V. Holmes, and C. Venters, "*VCC: A Framework for Building Containerized Reproducible Cluster Software Environments*," The Journal of Open Source Software, vol. 2, 03 2017.

[13] A. Petitet, R. C. Whaley, J. Dongarra, and A. Cleary, "*HPL - A Portable Implementation of the High-performance Linpack Benchmark for Distributed-memory Computers*," Feb. 2018.

[14] F. Hu and J. Zhang, "*A Docker-based HPC Architecture for RS Imagery Processing*," in IOP Conference Series: Earth and Environmental Science, vol. 509, p. 012026, IOP Publishing, 2020.

[15] L. Benedicic, F. A. Cruz, A. Madonna, and K. Mariotti, "*Sarus: Highly Scalable Docker Containers for HPC Systems*," in International Conference on High Performance Computing, pp. 46–60, Springer, 2019.

[16] V. Vieux, "*sshFS Docker Volume Plugin*."

[17] B. Christner, "*Docker Official Images are Moving to Alpine Linux*," May 2016.

[18] S. Bhartiya, "*Meet Alpine Linux, Docker's Distribution of Choice for Containers - The New Stack*," Mar. 2017.

[19] N. Janetakis, "*Benchmarking Debian vs Alpine as a Base Docker Image*," Oct. 2018.

[20] N. Nguyen, "*Alpine-MPICH*," 2017.

[21] P. Gedam, "*pip - The Python Package Installer*," Feb. 2019.

[22] L. Dalcin, "MPI for Python," Feb. 2019.

[23] G. A. V. Carias and J. F. H. Balestrini, "*Project Info*," Mar. 2019.

[24] D. K. D. Panda, "*MVAPICH: MPI over InfiniBand, Omni-Path, Ethernet/iWARP, and RoCE*," Mar. 2019.

Índice de Autores

A

Aguilera Ana 37

B

Berenguer Moisés 25

C

Cardinale Yudith 37

D

Dongo Irvin 37

H

Hernández José 37
Hernández Walter 1

M

Moreno Francisco 1, 25

T

Teruel Alejandro 13

V

Verrocchi Gianfranco 37

REVECOM

Sociedad Venezolana de Computación

La Sociedad Venezolana de Computación está comprometida con el impulso de una nueva generación académica y profesional en nuestra área de saber para el desarrollo del país.

Los conceptos y puntos de vista expresados en los trabajos publicados en este libro representan las opiniones personales de los autores y no reflejan el juicio de los editores o de la Sociedad Venezolana de Computación.

ISSN: 2244-7040

9 772244 704006

www.svc.net.ve/revecom