

**TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES
MEDIANTE IOT PARA LA SOLUCIÓN DE PROBLEMAS, PROCESAMIENTO
DE IMÁGENES EN ARQUITECTURAS MULTIPROCESADOR Y
ESTRATEGIAS DE TOLERANCIA A FALLOS EN HPC**

Information and Communication Technologies through IoT for solving problems, Image Processing in Multiprocessor Architectures and Fault Tolerance strategies in HPC

- ❖ *Osio, Jorge Rafael¹*
- ❖ *Montezanti, Diego Miguel*
- ❖ *Cappelletti, Marcelo Ángel*
- ❖ *Salvatore, Juan Eduardo*

Universidad Nacional Arturo Jauretche

Resumen

El proyecto se desarrolló sobre dos pilares fundamentales, por un lado, se explotaron las tecnologías de internet de las cosas (IoT) para la solución de problemáticas actuales, tales como la eficiencia energética, el control de parámetros en cultivos bajo cubierta y el diseño y desarrollo de laboratorios remotos para potenciar la enseñanza a distancia. Adicionalmente, se continuó con la línea anterior sobre la determinación de eficiencia en el procesamiento de imágenes en sistemas reconfigurables y la tolerancia a fallo en sistemas de cómputo de altas prestaciones (HPC).

Se desarrolló un sistema IoT de control de energía escalable que permite minimizar el consumo en diferentes ambientes, en conjunto con la implementación de un sistema informático que provee información precisa de magnitudes climatológicas, del estado del suelo y del nivel de iluminación en invernaderos. Luego de la implementación se obtuvieron resultados muy satisfactorios respecto a la eficiencia energética, al lograr

¹ josio@unaj.edu.ar

independizar el consumo del factor humano. Además, se implementaron interfaces wifi y GPRS para el monitoreo y control de parámetros desde plataformas IoT.

Respecto al procesamiento de imágenes en sistemas reconfigurables, se determinó la eficiencia respecto a la ejecución secuencial, mediante la implementación de un sistema multicore (computo paralelo), y la combinación de este último con la implementación concurrente del algoritmo, en donde se obtuvieron muy buenos resultados. Por último, se diseñó una metodología que permite detectar fallos en arquitecturas multicore, dando buenos resultados en diferentes pruebas.

Palabras claves: Internet de las cosas, Procesamiento de Imágenes, Sistemas multiprocesador, tolerancia a fallos, Cómputo de altas prestaciones

Abstract

The project was developed on two fundamental pillars, on the one hand, Internet of Things (IoT) technologies were exploited to solve current problems, such as energy efficiency, control of parameters in crops under cover and design and development remote laboratories to enhance distance learning. Additionally, the previous line on the determination of efficiency in image processing over reconfigurable systems and fault tolerance in high-performance computing (HPC) systems was continued.

A scalable energy control IoT system was developed that allows to minimize consumption in different environments, in conjunction with the implementation of a computer system that provides precise information on climatological magnitudes, the state of the soil and the level of illumination in greenhouse. After the implementation, very satisfactory results were obtained regarding energy efficiency, by keeping independence of consumption from the human factor. In addition, Wi-Fi and GPRS interfaces were implemented for the monitoring and control of parameters from IoT platforms.

Regarding the processing of images in reconfigurable systems, the efficiency was determined with respect to the sequential execution, through the implementation of a multicore system (parallel computing), and the combination of the latter with the concurrent implementation of the algorithm, where very good results were obtained.

Finally, a methodology was designed that allows detecting failures in multicore architectures, giving good results in different tests.

Keywords: Internet of things, Image Processing, Multicore Systems, Fault Tolerance, High Performance Computing.

Introducción

Internet de las cosas aplicado a eficiencia energética, monitoreo de cultivo y educación a distancia:

El concepto de internet de las cosas ha adquirido gran relevancia en los últimos años, debido a la posibilidad que ofrece de interconectar objetos entre sí y la conectividad a internet que provee a las redes de objetos [1]. Aunque los sistemas IoT son más que conocidos, su implementación y desarrollo no ha sido completamente explotada en determinados sectores. IoT aplicado a cultivos bajo cubierta es un área en la que la evolución tecnológica nunca ha sido una prioridad, entre los factores que han postergado la investigación en esta área se tiene la falta de capacitación y falta de soporte e interacción con los productores para determinar cuáles son las principales problemáticas que se deben resolver. Para este tipo de ambientes se dispone de un conjunto de protocolos de comunicación e interfaces como GSM/GPRS, Wifi, bluetooth y RF (a 433Mhz) que requieren un sistema de procesamiento basado en Arquitecturas de Sistemas Embebidos como las desarrolladas en [2]. Las investigaciones realizadas mediante IoT se podrían aplicar en el monitoreo de cultivos, información relevante para la toma de decisiones, automatizar el riego, protección de heladas, fertilización e iluminación artificial, entre otras [3].

Otra problemática común en el sector fruti-hortícola, es el acceso al suministro de energía, en cuyo caso requiere de sistemas costosos de energía alternativa. Actualmente, se considera una problemática a tratar de manera urgente el uso eficiente de la energía [4], debido a que se está investigando mucho sobre energías alternativas, pero hay mucho por hacer respecto a la reducción en el consumo de energía en función del factor humano y del control automático de los sistemas eléctricos/electrónicos [5]. En este sentido, se

debe tener en cuenta la necesidad de minimizar el consumo en los sistemas de energías alternativas, principalmente energía fotovoltaica [6], debido a que estos sistemas almacenan la energía en baterías altamente contaminantes y al lograr optimizar el consumo de energía se extendería notablemente la vida útil de la misma. Estos sistemas son de gran interés, principalmente por ser de uso común en los sectores agrícolas, donde se aplicaron las investigaciones realizadas conjuntamente con el proyecto “Estudio de sistemas inteligentes para optimizar el aprovechamiento de la radiación solar en la actividad agroindustrial del territorio de la UNAJ” dirigido por el Dr. Marcelo Cappelletti [7].

Respecto al uso racional y eficiente de la energía, este se ha tornado uno de los asuntos más importantes a considerar a nivel mundial. En particular, en Argentina, la energía es considerada como un área prioritaria de acuerdo con el Plan “Argentina Innovadora 2020”, del Ministerio de Ciencia, Tecnología e Innovación Productiva de la Nación. La demanda de energía ha aumentado considerablemente en los últimos años y de continuar al mismo ritmo de crecimiento, se estima que para el año 2030 habrá un 40% más de consumo que en el último año. En la actualidad, aproximadamente el 78% de la demanda final mundial de energía se satisface con sistemas energéticos basados en recursos fósiles no renovables, como el carbón, el gas natural y el petróleo, que si bien no se agotarán en el corto o mediano plazo se trata de recursos finitos. Poco más del 19% es cubierto con fuentes de energía renovable, mientras que el resto se satisface con energía nuclear. Los edificios residenciales, comerciales y públicos presentan un consumo energético estimado del 30% al 40% de la energía utilizada a nivel mundial. Además, dicho sector contribuye entre el 25% y el 35% de las emisiones de CO₂ mundiales [Datos de la Agencia Internacional de Energía (IEA)]. Para la República Argentina, los edificios residenciales, comerciales y públicos, tienen una participación del 31% en el consumo total de energía, del 55% del consumo total de electricidad y del 50% del consumo total de gas por red. Por lo tanto, en el proyecto consideramos que el estudio de la eficiencia energética se debe hacer de forma general, pensando no solo en sistemas de energía alternativa, sino también en un sistema escalable que se pueda aplicar en instituciones de todo tipo. Para minimizar el consumo, se considera necesario implementar un algoritmo

inteligente que tenga en cuenta todos los factores intervinientes, para esto es necesario diseñar un algoritmo que tenga en cuenta los datos previos para la toma de decisiones.

Respecto a la aplicación de IoT en Laboratorios remotos, teniendo en cuenta el contexto actual, en donde se están explotando las TICs para la enseñanza en aulas virtuales, es indispensable para dar soporte a estas plataformas el diseño de sistemas de control que permitan manipular y relevar el estado de los sistemas en estudio de forma remota. Desde el proyecto se implementó una interfaz IoT de Hardware y Software embebido entre servidor y el laboratorio físico para ser utilizado en el proyecto de la Dra. María Joselevich ([8], [9]).

Procesamiento de imágenes biomédicas mediante técnicas de cómputo paralelo en sistemas multi-core y tolerancia a fallos de dichos sistemas:

La segunda línea de investigación desarrollada en el proyecto se enfoca en los sistemas de procesamiento de altas prestaciones y la búsqueda de eficiencia en el procesamiento de imágenes en sistemas multicore.

En este sentido, los equipos de imágenes médicas cobran un papel progresivamente crítico en el cuidado de la salud. La prevención y el tratamiento se manejan combinando modalidades tales como la tomografía por emisión de positrones (PET), Tomografía computarizada (CT) y equipos de rayos X [10]. Estos requerimientos en sistemas que a su vez deben poseer la capacidad de actualizar continuamente sus características y algoritmos, cuya implementación debe ser flexible y contemplar la modalidad de fusión de métodos, no deja otra alternativa que usar sistemas reconfigurables como las FPGAs. Teniendo en cuenta estos factores, es de gran importancia implementar algoritmos de procesamiento de imágenes que además de posibilitar el realce de las imágenes médicas [11], permitan obtener información adicional posibilitando la extracción de estructuras biológicas de dichas imágenes [12]. Luego en base a los resultados obtenidos, se podrían combinar los métodos de procesamiento implementados para acelerar y mejorar la eficiencia en el diagnóstico médico. La implementación de dichos algoritmos sobre imágenes de mucha resolución requiere grandes capacidades de cálculo, para esto se hace necesaria la utilización de técnicas de paralelismo y concurrencia.

La facilidad de implementar procesadores embebidos de forma rápida, junto con la posibilidad de proveer concurrencia mediante la programación en HW permite combinar esta característica de las FPGAs con el paralelismo, obtenido mediante sistemas multicore, para alcanzar la máxima eficiencia.

Entre las ventajas que proveen las FPGAs actuales para el cómputo paralelo se pueden enumerar:

- SoftCores (o procesadores embebidos) que permiten realizar las tareas de administración de datos de baja tasa. Y la posible conexión de IP-Cores (CoProcesadores) específicos con las siguientes características para realizar el procesamiento duro de señales:
- Capacidad de procesamiento con millones de MACs por segundo (operaciones de Multiplicación / acumulación)
- DSP o Delay Locked Loops (DLL) que permiten la multiplicación o división de la frecuencia de reloj, entre otras tareas.
- Interfaces DRAM / SRAM de alta velocidad y rendimiento.
- Manejo del ancho de banda en señales y buses ahorrando pines I/O.
- Registros de desplazamiento, útiles para buffers de línea o FIFOs.
- RAM distribuida para almacenar coeficientes o pequeñas FIFOs.
- Block RAM con capacidad "true dual-port" para almacenar datos de fotograma, líneas o porciones de imagen, grandes tablas o FIFOs.

Con las mejoras constantes que aporta la evolución de la tecnología sobre las FPGAs pueden lograrse diseños de gran magnitud, a tal punto que la tendencia actual es implementar microprocesadores de propósito general, conjuntamente con todo el hardware de propósito específico que requiere la aplicación, dentro de una FPGA.

La utilización de técnicas de paralelismo requiere la implementación de múltiples procesadores en una FPGA. La complejidad incorporada al implementar sistemas multicores, incrementa la vulnerabilidad a los fallos transitorios y estos fallos podrían corromper los resultados de las aplicaciones [13].

El aumento en la escala de integración, con el objetivo de mejorar las prestaciones en los procesadores, y el crecimiento del tamaño de los sistemas de cómputo, han producido que la confiabilidad se haya vuelto un aspecto crítico. En particular, la creciente vulnerabilidad a los fallos transitorios se ha vuelto altamente relevante, a causa de la capacidad que poseen de alterar los resultados de las aplicaciones.

El impacto de los fallos transitorios aumenta notoriamente en el contexto del Cómputo de Altas Prestaciones, debido a que el Tiempo Medio Entre Fallos (MTBF) del sistema disminuye al incrementarse el número de procesadores. En un escenario típico, en el cual cientos o miles de núcleos de procesamiento trabajan en conjunto para ejecutar aplicaciones paralelas, la incidencia de los fallos transitorios crece en el caso de que las ejecuciones tengan una elevada duración, debido a que el tiempo de cómputo y los recursos utilizados que se desperdician resultan mayores [14]. Por lo tanto, el alto costo (en términos temporales y de utilización de recursos) que implica volver a lanzar la ejecución de una aplicación desde el comienzo, en caso de que un fallo transitorio produzca la finalización de la aplicación con resultados incorrectos, justifica la necesidad de desarrollar estrategias específicas para mejorar la confiabilidad y robustez en sistemas de múltiples procesadores [14]. En particular, el foco está puesto en lograr detección y recuperación automática de los fallos silenciosos, que no son detectados por ninguna capa del software del sistema, por lo que, sin producir una finalización abrupta, son capaces de corromper los resultados de la ejecución.

En este sentido se ha considerado indispensable investigar la tolerancia a fallos en sistemas con gran poder de cómputo (HPC) para luego desarrollar un método que permita minimizar los efectos de los fallos transitorios [15].

Objetivos

- Fortalecer la actividad de investigación y vinculación en el área de las TIC que contribuyan a remediar las problemáticas existentes dentro del territorio de influencia de la UNAJ en las áreas de medio ambiente, salud, medio productivo y educación.
- Promover la generación de conocimiento teórico y aplicado, el desarrollo de instrumentos que puedan aplicarse en las materias de grado y la formación de recursos humanos en investigación de posgrado.

- Desarrollar un sistema autónomo y escalable que permita minimizar el consumo de energía conjuntamente con la implementación de un sistema informático que provea información precisa de magnitudes climatológicas, del estado del suelo y del nivel de iluminación para cultivos bajo cubierta.
- Proveer soluciones de software embebido para la implementación de laboratorios remotos que permitan fortalecer la educación a distancia.
- Evaluar los diferentes métodos de ejecución de algoritmos de procesamiento de imágenes médicas sobre dispositivos FPGAs e implementar un método que contenga las ventajas de los anteriores.
- Desarrollar una metodología que permita la detección y recuperación de fallos transitorios en arquitecturas multicore (sistemas de múltiples procesadores), que afectan especialmente la ejecución de aplicaciones paralelas de cómputo intensivo

Hipótesis

- En la actualidad, se considera que un sistema de iluminación es eficiente energéticamente cuando permite adaptar el nivel de iluminación en función de la variación de la luz solar teniendo en cuenta la detección o no de presencia. En esta propuesta se considera que las soluciones existentes no han sido suficientemente desarrolladas, ya que para un uso eficiente de la energía se deben tener en cuenta varios factores y combinar una amplia variedad de métodos para lograr un algoritmo potente que permita optimizar el consumo.
- La principal hipótesis que justifica la necesidad de soluciones tecnológicas en el sector fruti-hortícola es la falta de interacción con los productores para determinar cuáles son las verdaderas problemáticas en la región. La mayoría de los problemas se podrían resolver mediante una red IoT y la interacción con tecnologías como las aplicaciones móviles.
- El contexto actual de pandemia hizo que se dé un gran paso en la educación a distancia, esta nueva realidad nos encontró desprovistos de herramientas que permitan realizar experiencias de laboratorio reales de manera remota. La

aplicación de IoT en la implementación de laboratorios remotos, permitirá resolver un gran problema en las asignaturas cuyo aprendizaje se enfoca en las actividades de laboratorio.

- La hipótesis para lograr eficiencia en el procesamiento de imágenes médicas se basa en aprovechar los beneficios de las técnicas de concurrencia y paralelismo. Para esto se debe encontrar una relación de compromiso entre concurrencia y paralelismo para maximizar la eficiencia.
- La ejecución de programas que procesan grandes cantidades de datos en sistemas de procesamiento paralelo tiene fallos cuando el consumo se incrementa drásticamente por la exhaustiva utilización de los recursos. En base a esto y a las características de dichos fallos, es indispensable poder detectarlos y recuperarse de ellos. Cabe destacar que esta problemática es cada vez más común, debido a las nuevas tecnologías que implementan inteligencia artificial y procesamiento grandes cantidades de datos.

Metodología:

Internet de las cosas aplicado a eficiencia energética, monitoreo y seguimiento de cultivos y educación a distancia

Para optimizar el consumo eficiente de energía, primero se deben tener en cuenta todos los factores de consumo, entre ellos el factor humano, el factor ambiental y el destino de cada uno de los recursos. Con un sistema diseñado a medida se puede lograr optimizar el consumo al máximo mediante la implementación de un algoritmo de decisión, que tenga la capacidad de adaptarse a las características del usuario, que pueda interpretar los parámetros de entrada y generar una salida eficiente. Para tener en cuenta todos estos factores se debe desarrollar un sistema complejo que tenga la capacidad de combinar todos estos aspectos de manera óptima y eficiente ([16] y [17]).

El sistema está formado por sensores (de temperatura, de iluminación el LDR, de movimiento el PIR y para la detección de personas la barrera infrarroja), actuadores (para las luminarias y la ventilación), el microcontrolador y la interfaz de comunicación

wifi (integrada en el sistema de procesamiento NODEMCU basado en el microcontrolador ESP8266). Por ser un sistema modularizable, éste se puede personalizar en función de las prestaciones deseadas y el ambiente de funcionamiento, es decir, que de acuerdo con las características de cada sector se pueden agregar o no sensores dependiendo del lugar de utilización. El prototipo inicial fue implementado en la oficina 215 de la UNAJ, y está encargado de automatizar el consumo de energía proveniente de la red eléctrica. En el lugar se instaló un medidor de consumo comercial GF-18WHM, similar al que provee la empresa proveedora del servicio eléctrico [18].

Para lograr un sistema escalable fue necesario implementar un servidor con una base de datos organizada por sector. Para almacenar la información obtenida de los sensores y registrar las acciones realizadas por el sistema embebido (encendido o apagado de luces) se utilizó MongoDB que es un sistema de base de datos NoSQL. En lugar de guardar los datos en tablas como se hace en las bases de datos relacionales, MongoDB guarda registros de datos como documentos BSON en colecciones. BSON se extiende del modelo JSON para proveer más tipos de datos. Las colecciones son análogas a las tablas en bases de datos relacionales. Por defecto, los documentos en una sola colección no necesitan tener el mismo conjunto de campos y el tipo de datos para un campo puede diferir de un documento a otro dentro de una misma colección. En las versiones más nuevas de MongoDB se pueden definir reglas de validación de documentos de una colección para las operaciones Insert y Update. En el trabajo desarrollado la estructura de los documentos se definió en el servidor web a través de la herramienta Mongoose. Se levantó un único servidor de base datos en el cual se crearon las siguientes colecciones:

- mediciones: almacena los valores medidos por los sensores
- nodos: guarda la información de las placas arduino utilizadas en el proyecto para identificarlas unívocamente
- sectores: almacena los sectores o espacios del edificio donde se aplica el sistema de control de energía
- sensores: almacena los tipos de sensores utilizados
- unidades: almacena las distintas unidades de medición

- acciones: cada documento almacena una acción realizada (encendido o apagado de luces), la fecha y hora en la que se efectúa y el microcontrolador que determina la acción.

Para el desarrollo del servidor web se utilizó Node.js, Express y Mongoose. Este se basa en un servicio que permite acceder a recursos mediante URIs y que soporta distintos tipos de peticiones HTTP para las URIs dadas. Los recursos disponibles son las mediciones, nodos, sensores, sectores, unidades y acciones. Este servicio permite que las aplicaciones intercambien datos: el NODEMCU envía los datos medidos por los sensores al servicio web, el servicio almacena la información en la base de datos MongoDB a través de la herramienta Mongoose ODM, y una vez ejecutada la acción sobre la base de datos, el servicio retorna al NODEMCU una respuesta satisfactoria o de error en formato JSON. El servidor web se estructuró principalmente en los directorios “models”, “controllers”, y “routes” y un archivo llamado “app.js”. La carpeta “models” contiene la estructura de los documentos de las distintas colecciones. Por cada colección se definió un archivo .js, donde se especifican los campos y los tipos de datos de sus documentos. Por ejemplo, la estructura de un documento Medición y de un documento Acción. En el archivo “app.js” se definió la conexión a la base de datos y una secuencia de middlewares por los que pasa la petición HTTP recibida. En base al tipo de recurso que el cliente solicita se utiliza el router correspondiente. En la carpeta “routes” se definió un router por cada recurso disponible en un archivo separado. Cada router redirige las peticiones HTTP a los correspondientes controllers. La carpeta “controllers” contiene los controladores de cada recurso y cada controlador contiene las funciones a ejecutar para atender las solicitudes, las cuales mantienen la lógica de negocio, realizan operaciones sobre la base de datos y retornan un documento JSON. La conexión a la base de datos se realizó a través de Mongoose. El framework Express se utilizó para definir los middlewares y las rutas por las que pasan las peticiones [19].

Para la aplicación se alquiló un servidor privado virtual (VPS) al proveedor Digital Ocean para levantar el servidor web en la nube. El servidor alquilado cuenta con las siguientes características:

- Sistema operativo: Ubuntu 16.04
- Disco rígido de estado sólido: 25 GB
- RAM: 1 GB
- Ubicación del servidor: Nueva York.

En cuanto a las soluciones tecnológicas para el monitoreo y control de cultivos, se requiere de la disponibilidad de datos tanto del clima como del suelo, la iluminación y el agua. El desarrollo del sistema informático consiste en un algoritmo de software con soporte para diferentes aplicaciones que, en base a la información proporcionada por un conjunto de sensores, emite mensajes de alertas a los usuarios y es capaz de automatizar acciones. Para que la propuesta sea aplicable en cualquier lugar, se planteó la utilización de una interfaz GSM/GPRS (mediante un módulo sim800) que permite transmitir información mediante las antenas celulares. El diseño del sistema local a utilizar en huertas a campo abierto y bajo cubierta, está formado por un sistema de procesamiento (basado en tecnologías de microcontroladores ATmega328 y ATmega2560), al cual se conectan los sensores y el dispositivo de comunicación GSM/GPRS. El nodo central del sistema se compone del bloque procesamiento formado por el microcontrolador ATmega 2560 y los sensores (de humedad, humedad del suelo, temperatura, iluminación, movimiento y barrera infrarroja), actuadores (para el encendido de la luminaria y riego) y la interfaz de comunicación (módulo sim800). El sistema es escalable y puede contener nodos simples que contienen solos sensores y se comunican con el Gateway mediante el protocolo LoRa a 533Mhz.

Se realizaron tres implementaciones de la interfaz web y el control de parámetros. Una de las implementaciones se realizó sobre una raspberry en lenguaje Python, mediante un sistema de tres hilos que se encargan de la lectura de los sensores, la actualización de la plataforma web y el control de la iluminación [20]. La segunda también se implementó en raspberry, pero con una interfaz web más profesional [21]. La tercera se realizó en un sistema embebido basado en el lpc1769, que contiene un servidor web embebido y los sensores. A diferencia de la raspberry el servidor es más limitado y permite muy pocas conexiones al a vez [22].

Respecto al laboratorio remoto, entre las estrategias posibles, se eligió estudiar la incidencia de la incorporación de laboratorios remotos (LR) en las prácticas de enseñanza de la materia Física 1. El trabajo que se presenta se dirige en particular al estudio de las ondas mecánicas tomando en cuenta los antecedentes del tratamiento de este tema a partir del análisis de ondas sonoras, ondas en una cuerda, entre otras. Los laboratorios remotos, se presentan actualmente como una opción que despierta un gran interés [9]. La implementación del laboratorio remoto consta de las siguientes etapas:

- El servidor de enlace fue implementado mediante la plataforma educativa moodle, en donde se configuró un plugin para la solicitud de turnos y la gestión de acceso al laboratorio.
- El servidor multimedia se implementó en una minicomputadora (raspberry pi), donde se configuró Motion como un servidor streaming para el acceso a las imágenes y Apache como servidor web, donde se muestran las opciones de encendido y configuración del laboratorio.
- Se utilizó el motor de base de datos phpMyAdmin que permite manejar la administración de MySQL para el almacenamiento de los datos generados en cada experiencia.

Además, el sistema de control del laboratorio físico basado en Raspberry, se implementó configurando los pines I/O de la placa para el control del sistema generador de onda (un pin para el encendido y apagado del sistema, el protocolo SPI para el control de amplitud mediante un potenciómetro digital y una salida PWM para el control de la frecuencia). El laboratorio se visualiza mediante una plataforma que permite seleccionar la amplitud, la frecuencia y visualizar la cámara [9].

Procesamiento de imágenes biomédicas mediante técnicas de procesamiento paralelo en sistemas multi-core y tolerancia a fallos

Los pasos a realizar en la determinación de la eficiencia en el procesamiento de imágenes sobre dispositivos FPGA son:

- Implementación de un sistema multiprocesador en Dispositivos Lógicos Programables (FPGAs).
- Análisis y determinación del desempeño logrado en el procesamiento de imágenes mediante la combinación de cómputo paralelo y concurrencia.

En la implementación del sistema se utilizó una tecnología de cálculo basada en dispositivos FPGAs, la cual permite la implementación de varios procesadores en un único chip. La implementación de paralelismo en plataformas FPGAs consiste en el uso de procesadores embebidos para ejecutar aplicaciones y en la utilización de las características que provee la lógica programable para manejar las porciones de código que se ejecutan concurrentemente [23].

Para una implementación eficiente, se propone dividir el algoritmo en dos partes. La primera parte consiste en aprovechar las características propias de las FPGAs logrando que la parte no común a todos los algoritmos se implemente de manera concurrente mediante bloques aritméticos de HW disponibles. La segunda parte requiere la utilización de un sistema de múltiples procesadores, implementado dentro de la FPGA, que permita ejecutar las partes comunes de los algoritmos de manera indistinta. Para esta etapa se propone el uso de arquitecturas de múltiples procesadores. Para fundamentar las mejoras en la ejecución de la parte no común, se medirá el nivel de concurrencia logrado. Respecto a las mejoras en la ejecución de la segunda parte de los algoritmos se utilizarán las leyes de Amdahl y Gustafson. Dichas leyes permitirán determinar la mejora esperada para contrastar con los resultados obtenidos en la implementación y sacar las conclusiones pertinentes [24]. En la Figura siguiente se muestra un diagrama en bloques del sistema implementado en la FPGA.

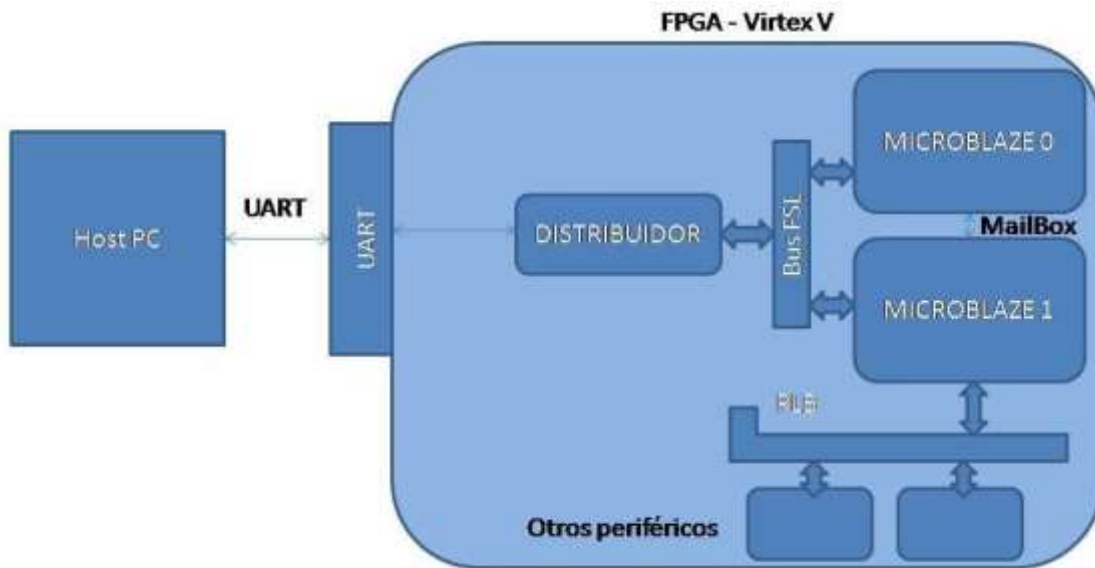


Figura 1. Conexiones del sistema completo

Respecto a la tolerancia a fallos:

- Diseño e implementación de una estrategia que permite al sistema detectar y recuperarse automáticamente de los errores producidos por fallos transitorios.
- La implementación está basada en replicación de software y es totalmente distribuida; está diseñada para operar en entornos de clusters de multicores donde se ejecutan aplicaciones paralelas científicas de paso de mensajes.

Para la detección de fallos en los sistemas paralelos se propone una metodología distribuida basada en software, diseñada específicamente para aplicaciones paralelas científicas que utilizan paso de mensajes, capaz de proveer tolerancia a fallos transitorios que, de otra forma, producirían ejecuciones incorrectas. La estrategia de detección se basa en validar los contenidos de los mensajes que se van a enviar y comparar los resultados finales. Para lograr proveer tolerancia completa a los fallos se buscará restaurar el sistema a un estado consistente, previo a la ocurrencia del fallo [23].

Resultados

Internet de las cosas aplicado a eficiencia energética, monitoreo de cultivo y educación a distancia

En el sistema de eficiencia energética se detecta presencia en la sala mediante la barrera infrarroja y el sensor de movimiento. El uso de la barrera se hace indispensable, debido a que el sensor de movimiento puede detectar fuentes de calor que no necesariamente son personas, es por eso que para asegurar presencia se detecta el ingreso de individuos mediante la barrera. Como no hay forma de diferenciar si la persona egresa o ingresa, para mantener la luz encendida se debe detectar movimiento en un lapso máximo de 3 minutos. La información relevante se almacena en el servidor mediante la interfaz wifi. Para esto se debe enviar una secuencia de comandos al módulo wifi que posibilite transmitir el paquete de datos de los sensores al servidor. Si se produce algún error en la secuencia se deberá iniciar la misma desde el principio. El reenvío de comandos se puede realizar tres veces, si a la tercera vez no hay respuesta del módulo wifi, se produce un TIMEOUT. Actualmente, el sistema se encuentra relevando datos de intensidad de luz, presencia y tiempo de encendido de la iluminación de forma automática. Para determinar la eficiencia fue necesario medir el consumo en la misma sala, durante un periodo de tiempo considerable con y sin el uso del sistema para luego, determinar el ahorro real de energía. Los tiempos de medición son de 30 días, para determinar el consumo en la misma época del año. Esta información es relevante para determinar el ahorro de energía que se pueda lograr con el sistema propuesto en oficinas laborales [18].

Durante el desarrollo del proyecto surgieron varios problemas con la utilización del módulo Wifi ESP8266. En un principio, para lograr la comunicación entre el módulo Wifi y el Arduino, se utilizó la librería “SoftwareSerial” que usa software para replicar la funcionalidad del RX, TX en otros pines digitales diferentes a los del módulo UART, que se usa para test. Esta librería resultó ser bastante inestable por lo que se decidió aprovechar las características propias del Arduino Mega, y utilizar uno de los tres puertos seriales adicionales que posee. Una vez establecida la comunicación entre el módulo Wifi y el microcontrolador ATMEGA, se procedió a realizar pruebas enviando datos del Arduino al servidor Web. Para ello se debieron mandar una secuencia de comandos AT al módulo para

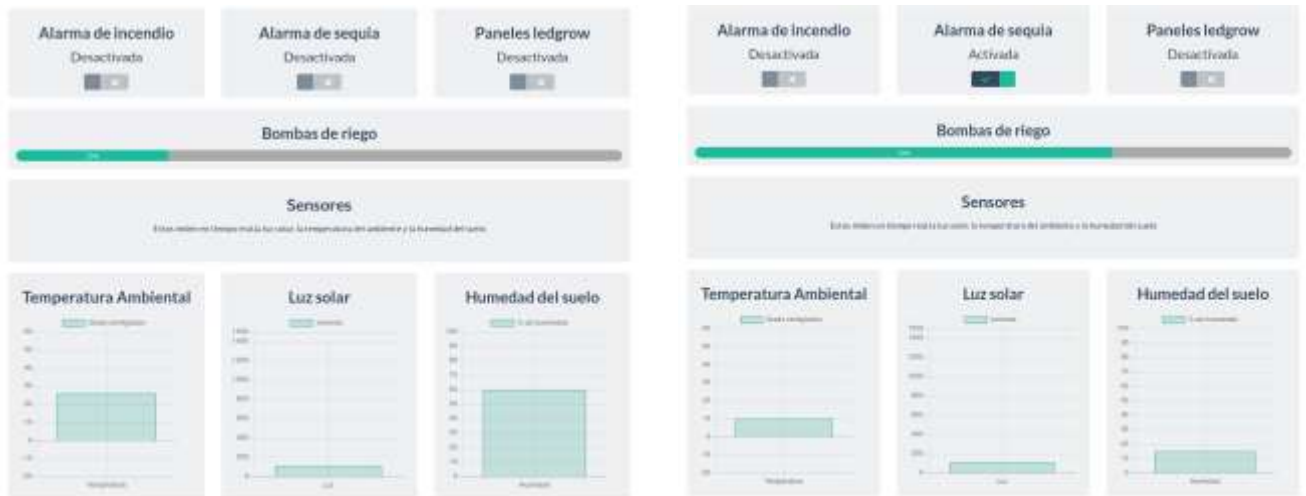
que se conectara a la red local, estableciera una conexión TCP con el servidor y luego enviara el mensaje. Luego de realizar una gran cantidad de pruebas se llegó a la conclusión de que el módulo es bastante inestable ya que en determinados momentos funciona correctamente, pero luego deja de funcionar entrando en un estado de error constante en el que es necesario hacer un Reset, o simplemente se resetea o se desconecta solo. En función de la inestabilidad del módulo Wifi utilizado, debimos optar por la migración del sistema a un NODEMCU, que permite programar directamente el microcontrolador (esp8266) que contiene el módulo wifi, sin la necesidad del microcontrolador de Arduino y de los comandos AT. De esta forma, el NODEMCU manda la información medida por los sensores al servidor para mantener un registro y en caso de error reintenta enviarlos un número determinado de veces. Durante las pruebas también se detectó que el sensor de movimiento emite falsas señales de detección si se encuentra muy próximo a la lámpara, ya que detecta como un movimiento el cambio de temperatura emitido por la luz. Por lo que se debió disponer el sensor un poco más alejado de la ubicación de la lámpara.

Las pruebas realizadas permitieron verificar el correcto funcionamiento del sistema, encendiendo la iluminación sólo cuando la luz natural es insuficiente y si se detecta presencia en la sala mediante la barrera infrarroja y el sensor de movimiento. La última etapa consistió en medir el consumo en un ambiente determinado durante un periodo de tiempo considerable, con y sin el uso del sistema desarrollado. Esto permite obtener información relevante del funcionamiento del sistema para corroborar que el mismo contribuye al ahorro del consumo energético [19].

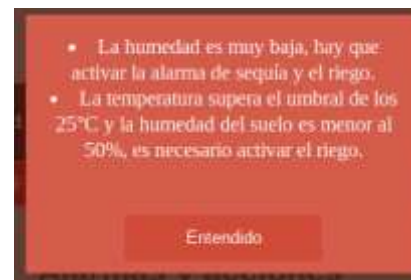
En el sistema de monitoreo y control del cultivo, se implementó un sistema de telemetría mediante la interfaz GSM/GPRS con el módulo SIM800, que permite supervisar el estado del cultivo (humedad del suelo, iluminación, humedad ambiente y temperatura), emitir alarmas por sobre temperatura e incendio y accionar el riego automático desde una aplicación móvil, el cual fue probado en campo y se obtuvieron excelentes resultados. Durante las pruebas se manifestaron varios inconvenientes provenientes de problemas en la señal provista por la empresa de telefonía y de las características de la antena del módulo. Estos inconvenientes impiden que se consiga una buena tasa de transferencia con la plataforma de acceso remoto (dashboard) y posibilitan una posible pérdida de datos. Entre las soluciones se propone el reenvío de los datos cuando no se recibe confirmación y el

cambio de antena por una con mejores prestaciones. Además, se implementaron dos plataformas de control diferentes sobre una minicomputadora raspberry, en la Figura 2 (a) se muestra una aplicación más vistosa, pero menos eficiente y en la Figura 2(b) se muestra una aplicación implementada con tres hilos que ejecutan tres tareas diferentes de forma concurrente (la actualización web, la lectura de sensores y el control de actuadores).

Un tema que se abordó en el diseño de este sistema es la utilización de energía alternativa, principalmente la energía solar, que es habitual por el hecho de que el tendido eléctrico en las zonas agrícolas es poco común debido a las grandes extensiones que se deben cubrir. En este sentido, contemplar la alimentación del sistema con energía alternativa provee flexibilidad y versatilidad, es por eso que se realizaron estudios relacionados con el aprovechamiento de la energía solar, mediante un análisis y predicción de la radiación solar y el dimensionado del sistema de energía solar (paneles, regulador de carga, inversor y sistema de baterías) ([26]y[27]).



(a) Primera implementación del sistema, a la izquierda en estado normal y a la derecha en estado de sequía



(b) Implementación con 3 hilos, menos elegante pero más eficiente

Figura 2. Implementaciones del sistema de monitoreo y control de cultivos bajo cubierta y a cielo abierto

Respecto a la implementación de una plataforma para IoT basada en el procesador Cortex M3 de 32 bits y el interfaz wifi mediante el módulo ESP8266, se obtuvieron resultados aceptables [2]. Durante las pruebas realizadas se obtuvieron excelentes tiempos en la interacción entre servidor (PC Intel core I7) y la plataforma de lectura de sensores y control de actuadores mediante el protocolo TCP. Entre los aspectos negativos, el módulo ESP01 es inestable, cuando se incrementa el consumo es común que se reinicie o quede en un estado inactivo que requiere el reinicio. Para minimizar este efecto fue necesario agregar un filtro capacitivo que minimice la caída de tensión durante los picos de consumo.

Respecto al laboratorio remoto diseñado para la materia física 1, se realizó una prueba piloto y se evaluaron las diferentes etapas del sistema [9].

Actualmente el proyecto se encuentra en la etapa de testing, en donde se ha logrado la implementación completa del sistema con algunas pruebas de funcionamiento que permitieron verificar la correcta ejecución del acceso de usuarios, solicitud de turnos, acceso a la plataforma donde se pueden ejecutar los comandos de configuración de forma remota (selección de frecuencia, fase y obtención de imágenes) y el almacenamiento de datos del experimento en la nube (ver Figura 3).

La siguiente instancia consiste en resolver los aspectos de seguridad para poder realizar la primera prueba piloto con una comisión de alumnos, en donde se podrán recabar datos sobre las dificultades surgidas, los aportes de este al tema de estudio y las posibles modificaciones a futuro [9].

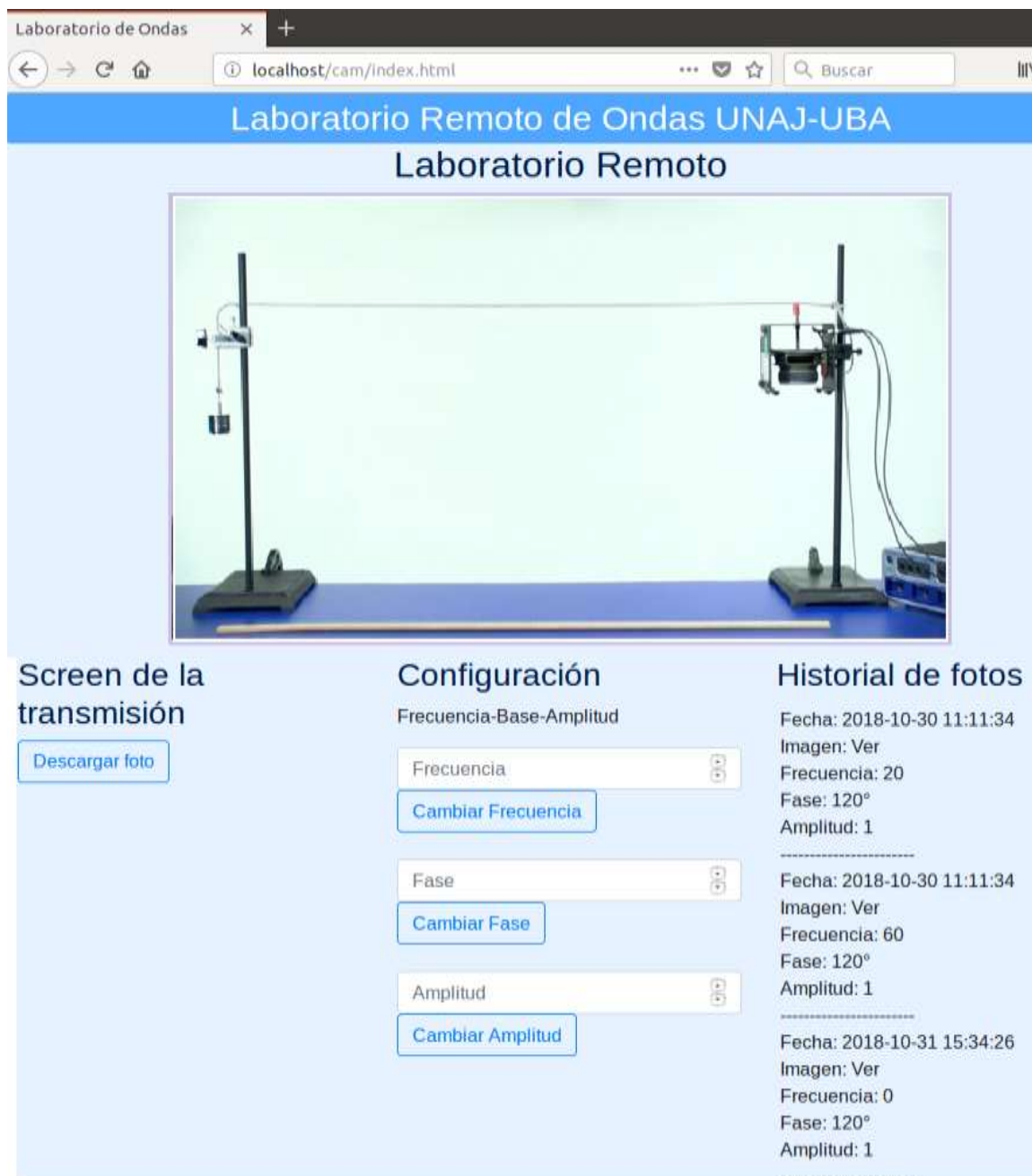


Figura 3. Entorno de control y visualización del laboratorio de ondas

Procesamiento de imágenes biomédicas mediante técnicas de procesamiento paralelo en sistemas multi-core y tolerancia a fallos

Respecto a la determinación de eficiencia en el procesamiento de imágenes sobre dispositivos FPGAs, Con una arquitectura constituida por varios procesadores implementados en una misma FPGA (microblaze), se buscó lograr eficiencia mediante la

combinación de técnicas de paralelismo y la concurrencia lograda en VHDL. Para las pruebas se emplearon algoritmos de procesamiento de imágenes basados en operadores de ventana, debido a que una parte del procesamiento es común para todos, por lo que puede paralelizarse entre todos los procesadores, mientras que el resto puede implementarse de forma concurrente mediante un co-procesador. La implementación de diferentes algoritmos permitió medir performance en la ejecución usando el sistema multiprocesador, y sacar conclusiones a partir de las características de cada algoritmo.

Se han implementados varios algoritmos en sistemas multicore con buenos resultados en el procesamiento paralelo. En el sistema, uno de los cores se encarga de recibir los datos y almacenarlos en memoria compartida (DDR2-SDRAM de alta velocidad y con múltiples puertos) para el posterior procesamiento. El procesador máster es el encargado de coordinar la lectura y procesamiento de los datos mediante pasaje de mensajes.

Tomando la imagen como una matriz de píxeles, los tiempos de lectura, escritura y procesamiento se midieron para la mínima cantidad de filas, lo que permitió sacar conclusiones respecto a los tiempos individuales por fila y de la imagen completa. Estos resultados permiten observar el beneficio obtenido al implementar un sistema multiprocesador, que usa memoria compartida con múltiples puertos, respecto a uno simple procesador, ya que al poseer casi todo el código paralelizable la mejora es prácticamente el doble [29]. El análisis que se puede hacer con relación a los tiempos de los distintos algoritmos es que a medida que la complejidad del algoritmo se incrementa, el tiempo de lectura/escritura de memoria compartida se vuelve despreciable respecto al de procesamiento [15].

En la Tabla 1 se muestran los tiempos de procesamiento con dos procesadores respecto al uso de un solo procesador.

Tabla 1. Tiempos de procesamiento de una fila usando uno y dos procesadores

Algoritmo	Un procesador (useg)	Dos procesadores (useg)
Filtro de mediana	3715	1870
Filtro de media	215	120
Filtro de sobel	6055	3040
Erosión/dilatación	3715	1870

Para evaluar el rendimiento conseguido por los distintos sistemas que se probaron, se ejecutaron los algoritmos de la tabla, midiendo el tiempo de ejecución sobre cada uno [30]. En cada caso evaluado, se ejecutaron las aplicaciones primero sobre un procesador y luego sobre dos procesadores con y sin coprocesador. Luego, para determinar el rendimiento se utilizaron las siguientes métricas:

- *SpeedUp*: representa la aceleración del sistema a la hora de ejecutar una aplicación respecto al sistema con sólo un procesador, esto es el cociente entre el tiempo que tarda el programa en ejecutarse sobre un solo procesador y el tiempo que tarda en n procesadores [30].

$$Speedup = \frac{T_1}{T_n}$$

- *Eficiencia relativa al número de procesadores*: representa el grado de aprovechamiento de los procesadores del sistema a la hora de ejecutar una aplicación paralela. Se define como el cociente entre el speedup y el número de procesadores.

$$E_p = \frac{S_p}{P}$$

- *Eficiencia relativa al área*: representa el grado de aprovechamiento de los recursos de la FPGA al ejecutar una aplicación paralela utilizando varios procesadores. Se define como la división entre el speedup y el cociente de las áreas del sistema ocupadas por un procesador y por n procesadores.

$$C_a = \frac{A_1}{A_n} \quad E_a = \frac{S_n}{C_a}$$

- *Speedup*

Se podría haber utilizado la ley formal de amdahl, que depende del porcentaje de código paralelizable y del número de procesadores, pero no es una buena métrica al momento de tener en cuenta los coprocesadores principalmente porque una parte del algoritmo se ejecuta en lenguaje C y otra en VHDL. En la Tabla 2 se muestran los valores de speedup para un sistema de dos procesadores con y sin coprocesador, se puede destacar que la mejora lograda para los algoritmos de mayor costo computacional es notable, esto se debe a que los tiempos consumidos en el movimiento de datos son despreciables respecto a los tiempos de cálculo. En cambio, en la ejecución del algoritmo de media no se logra ningún beneficio, debido a que los tiempos de cálculo son mucho menores y comparables con los tiempos de movimiento de datos.

Tabla 2. Cálculo de Speedup sin y con coprocesador

Algoritmo	Speedup sin coprocesador	Speedup con coprocesador
Filtro de mediana	1,987	4,07
Filtro de media	1,791	1,65
Filtro de sobel	1,991	3,089
Erosión/dilatación	1,987	4,07

Como se esperaba el gráfico de speedup de la figura 4 muestra un comportamiento bastante lineal, teniendo en cuenta que las pruebas realizadas incluyen uno, dos procesadores y dos procesadores más el coprocesador [31]. El único caso en donde no se observa mejora aparente, es el cálculo del algoritmo de media que tiene características muy diferentes al resto de los algoritmos.

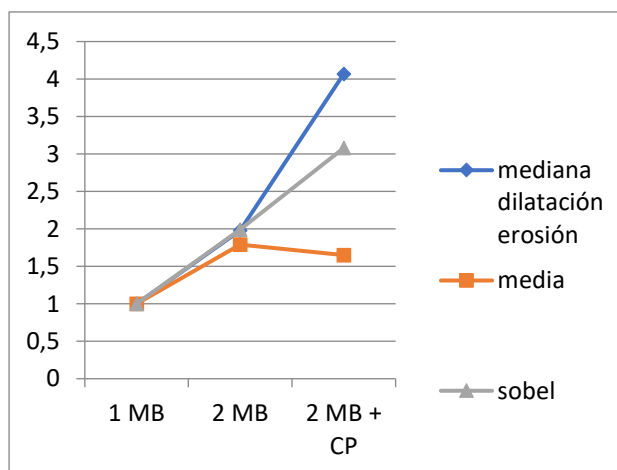


Figura 4. Speedup vs número de procesadores

- *Eficiencia relativa al número de procesadores*

Se observa en la Tabla 3 que la misma se incrementa a casi el doble al agregar a los procesadores un acelerador de hardware (coprocesador).

Tabla 3. Eficiencia relativa al número de procesadores

Algoritmo	$E_{p\text{sin}}$ coprocesa dor	E_p con coprocesad or
Filtro de mediana	0,9935	2,035
Filtro de media	0,8955	0,825
Filtro de sobel	0,955	1,5445
Erosión/dilatación	0,9935	2,035

- *Eficiencia relativa al área*

Primero se debe calcular el cociente de las áreas del sistema con un procesador y con dos procesadores. Para el cálculo del coeficiente C_a se tuvieron en cuenta los datos detallados en las tablas 1,2 y 3; de donde se desprende que dos procesadores ocupan el 34% de los bloques slice, un procesador el 15,78%, el filtro de ordenamiento en VHDL el 4,9%, la convolución el 2,23% y el promedio se considera despreciable. En conclusión, la mejora en la eficiencia relativa al área que se observa en la Tabla 4 se debe a que el área ocupada por los coprocesadores es despreciable respecto a la utilizada por cada procesador, sumado a la mejora lograda en el cálculo de speedup al usar los coprocesadores.

Tabla 4. Eficiencia relativa al área

Algoritmo	E_A	E_A con coprocesador
Filtro de mediana	0,9216	1,6507
Filtro de media	0,8307	0,7653
Filtro de sobel	0,9234	1,3468
Erosión/dilatación	0,9216	1,6507

Entre los resultados se debe destacar la mejora que se logra (speedup) al realizar el procesamiento mediante el agregado de un bloque de HW coprocesador, pero también es importante el análisis de la imagen para verificar la efectividad del algoritmos y su correcta ejecución.

En cuanto al tipo de imágenes a procesar, este método se puede aplicar en la ejecución de algoritmos basados en operadores de ventana, sobre cualquier tipo de imágenes digitales. Se decidió aplicar a imágenes médicas principalmente por el gran caudal de datos de imágenes que deben procesar los equipos médicos.

Resultados del procesamiento

En la Figura siguiente se puede observar el efecto de aplicar el filtro de media sobre imágenes médicas de 128x128 píxeles. En este caso se procesó una imagen radiográfica y una tomográfica con ruido sal y pimienta, donde se observa que el efecto del ruido es minimizado, debido a que el promedio en una ventana permite suavizar imágenes con irregularidades.

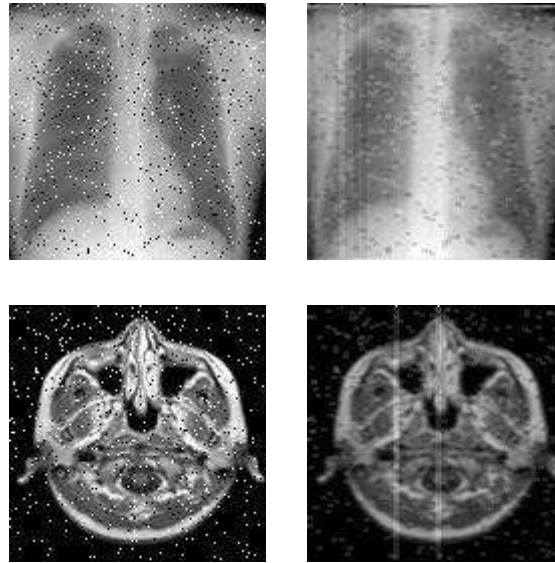


Figura 5. Procesamiento de una imagen radiográfica con el filtro de media

En la Figura siguiente se observan los resultados de aplicar el filtro de mediana a las imágenes de 128x128. Aquí se observa la eficiencia de este método para eliminar el ruido de tipo sal y pimienta. Al realizar el ordenamiento puede suceder que el píxel central sea un valor extremo y por eso no se elimina completamente el ruido. Una posible solución podría ser realizar el procesamiento con una ventana de 5x5 en lugar de usar una de 3x3.

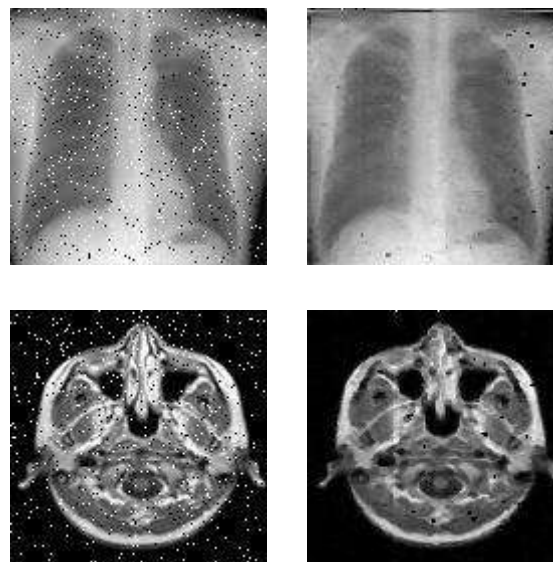


Figura 6. Procesamiento de una imagen radiográfica con el filtro de mediana

En la Figura siguiente se aplicó el filtro de sobel para obtener los bordes de los objetos de la imagen que son un conjunto de embriones [31]. En este caso lo que interesa es demarcar los bordes de los objetos de un color diferente para poder individualizarlos y determinar su tamaño en función de la cantidad de píxeles o sus formas.

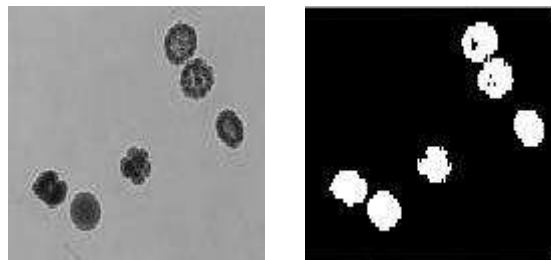


Figura 7. Procesamiento de una imagen de embriones con el filtro de sobel

Por último, en la Figura siguiente se observa el efecto del algoritmo de dilatación, que es muy utilizado luego de aplicar algún algoritmo que reduce el tamaño de los objetos, por ejemplo, para separar estructuras. El objetivo de este algoritmo es devolver el tamaño original para poder determinar su tamaño correctamente.

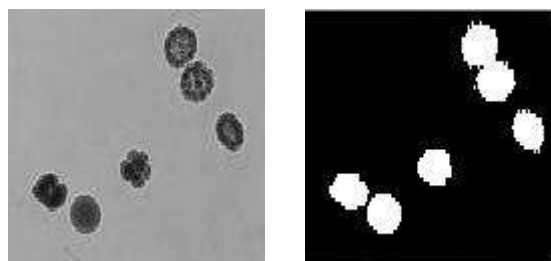


Figura 8. Procesamiento de una imagen de embriones con el método de dilatación

Tolerancia a Fallos

En cuanto a la tolerancia a fallos, se ha propuesto una metodología distribuida basada en replicación de software, diseñada específicamente para aplicaciones paralelas científicas de paso de mensajes, capaz de protegerlas de fallos transitorios que producirían ejecuciones incorrectas. Bajo la premisa de que, en este tipo de aplicaciones, la mayor parte de los datos relevantes para el resultado son transmitidos entre procesos, la estrategia de detección se

basa en validar los contenidos de los mensajes que se van a enviar y comparar los resultados finales, obteniendo un compromiso entre un alto nivel de cobertura frente a fallos y la introducción de un bajo overhead temporal y sobrecarga de operaciones, debido a que no se realiza trabajo para detectar fallos que no afectan a los resultados.

Para lograr este objetivo, se ha integrado la detección con un mecanismo basado en múltiples checkpoints coordinados en capa de sistema, contruidos con la librería DMTCP (que proporcionan cobertura en el caso de que un checkpoint resulte afectado por un fallo, imposibilitando la recuperación desde él), o la utilización de un único checkpoint no coordinado de capa de aplicación (construido ad-hoc), que puede ser verificado para asegurar la integridad de sus datos.

Para una aplicación de prueba, se ha desarrollado un modelo formal de verificación, que permite constatar la eficacia del mecanismo de recuperación. Además, se implementó una experimentación basada en casos de inyección controlada de fallos que valide las predicciones dadas por el modelo mencionado. Se buscó además caracterizar el comportamiento temporal de cada nivel de solución (sólo detección, recuperación basada en múltiples checkpoints de capa de sistema o recuperación basada en un único checkpoint de capa de aplicación) y verificar el comportamiento pronosticado.

El uso de estas estrategias posibilita prescindir de la utilización de redundancia triple con votación para detectar y recuperar de fallos transitorios [15].

Mediciones de overhead y resultados experimentales

Para cuantificar la incidencia de las distintas alternativas de SEDAR y su relación con las características de las aplicaciones, se han llevado a cabo algunas pruebas a pequeña escala, destinadas a describir los comportamientos temporales y los overheads introducidos. En este sentido, hemos comparado los distintos modos de operación mencionados previamente, tanto en ausencia como en presencia de fallos, en distintos escenarios de intervalos de checkpointing y momentos de inyección. De esta forma, se aporta información en la búsqueda de criterios que permitan alcanzar la configuración óptima de la herramienta según las características particulares del sistema.

Los experimentos diseñados están destinados a determinar la incidencia de la incorporación de SEDAR a la ejecución de las aplicaciones paralelas MPI, mostrando

además la eficacia de los mecanismos de detección y de recuperación automática, y sus tiempos de respuesta, en distintos escenarios de un entorno pequeño, de forma de realizar una prueba de concepto. Con este fin, adjuntamos SEDAR al código de las aplicaciones para integrar su funcionalidad. SEDAR se utilizó tanto en las variantes de sólo detección y relanzamiento automático (SEDAR-det en las Tablas 5 y 6) y de recuperación basada en checkpoints periódicos de nivel de sistema (SEDAR-rec en las Tablas 5 y 6)

En ambas aplicaciones, para los experimentos, se utilizaron matrices cuadradas de 8192 x 8192 elementos. Los experimentos consisten en medir los tiempos de ejecución de ambas variantes mencionadas de SEDAR, en ausencia de fallos en una primera instancia. En el caso de la estrategia de detección y relanzamiento automático, el objetivo es determinar el overhead introducido respecto de: (1) una ejecución sin ninguna protección y (2) una ejecución protegida manualmente por medio de ejecución simultánea de dos instancias independientes. En tanto, en el caso del mecanismo de recuperación automática, se buscó establecer la relación entre el overhead y el intervalo de checkpointing t_i , midiéndolo para tres valores distintos: $t_i = 60s$, $t_i = 90s$, $t_i = 120s$. En el caso de la multiplicación de matrices, para las duraciones de ejecución planteadas, la cantidad de checkpoints almacenados fue $N_{ckpt} = 6$, $N_{ckpt} = 4$ y $N_{ckpt} = 3$, respectivamente. Estos intervalos fueron definidos, tomando en cuenta la breve duración total de estas ejecuciones para prueba de conceptos, buscando ofrecer distintas alternativas de compromiso entre la protección brindada y el overhead introducido, frente a distintos valores posibles del MT BE.

Nuevamente, los experimentos de este pequeño conjunto se llevaron a cabo utilizando dos nodos de un cluster. Los resultados de las pruebas realizadas se resumen en las Tablas 5 y 6. En todos los casos se utilizaron 8 procesos, y los tiempos obtenidos son el promedio de 5 repeticiones de cada experimento particular. En ambas tablas, el tiempo de la fila 1 es el de la ejecución de la aplicación MPI pura, con la particularidad de que el mapping se realizó con 4 procesos en cada nodo, aun quedando cores libres, de forma de forzar la utilización de la red, para que la comparación con SEDAR (que replica los procesos y utiliza todos los cores) fuese más justa. En la fila 2 de ambas tablas, se muestra el tiempo de ejecutar el esquema de detección manual consistente en dos instancias independientes simultáneas de la aplicación. La diferencia de tiempos con la ejecución original se debe a la competencia por los recursos entre los procesos de las dos instancias. Si ocurre un fallo en la ejecución

de dos instancias simultáneas, (para la estrategia de la fila 1 pasará desapercibido), se requiere una tercera ejecución. Para realizarla, sin modificar el mapping, se ejecutaría una sola instancia de desempate, igual a la de la primera fila. Por lo tanto, el tiempo de ejecución total sería la suma del baseline (fila 2) con el valor de la fila 1 (es decir, sin competencia por los recursos). Se debe aclarar que esto sólo tiene en cuenta los tiempos de ejecución puros, sin contemplar el tiempo de comparación offline de los resultados. Además, debe notarse que el tiempo total es independiente del instante de ocurrencia del fallo, debido a que, en cualquier caso, la ejecución debe concluir para poder verificar la validez de los resultados. En la fila 3 de ambas tablas se muestra el tiempo cuando se incorpora el mecanismo de detección de SEDAR. Se puede observar que el overhead introducido (debido a la duplicación de procesos, sincronización entre réplicas, comparación y copia de los contenidos de los mensajes y validación final de los resultados) es extremadamente bajo (menor al 0.3 % para la multiplicación de matrices). En tanto, en caso de fallo, el tiempo total, que proviene de ejecutar hasta el momento de la detección, parar y relanzar desde el comienzo, es aproximadamente el doble del de la ejecución sin fallos para la multiplicación de matrices, y representa un overhead aproximado menor al 7 %, respecto de realizar una tercera ejecución (compárense los valores de las filas 2 y 3 de la Tabla 6, con fallos). Sin embargo, para cualquier aplicación, SEDAR-det ofrece la ventaja de incluir los tiempos de comparación de resultados y de relanzamiento automático, que en el caso del baseline deberían ser llevados a cabo por el usuario una vez terminado el procesamiento (offline) [32].

Tabla 5. Comparación de los tiempos de las distintas estrategias de SEDAR, con diferentes casos de inyección e intervalos de checkpoint, para la multiplicación de matrices

	Estrategia	Tiempo [s] según caso de inyección			
		Sin fallo	Al comienzo	Intermedio	Al final
1	Sin SEDAR - 1 instancia	329,57	-		
2	Sin SEDAR - 2 instancias simultáneas	373,16	702,73		
3	SEDAR - det	374,19	745,06	748,82	747,62
4	SEDAR - rec ($t_c = 60s$); $N_{check} = 6$	440,70	2066,23	1000,12	477,35
5	SEDAR - rec ($t_c = 90s$); $N_{check} = 4$	418,91	1569,85	822,93	459,63
6	SEDAR - rec ($t_c = 120s$); $N_{check} = 3$	407,06	1303,28	895,41	454,22

Tabla 6. Comparación de los tiempos de las distintas estrategias de SEDAR, con diferentes casos de inyección e intervalos de checkpoint, para la aplicación de Jacobi

	Estrategia	Tiempo [s] según caso de inyección			
		Sin fallo	Al comienzo	Intermedio	Al final
1	Sin SEDAR - 1 instancia	495,7	-		
2	Sin SEDAR - 2 instancias simultáneas	560,4	1056,1		
3	SEDAR - det	562,5	619,9	844,6	1069,7
4	SEDAR - rec ($t_i = 60s$); $N_{cápu} = 9$	594,7	662,6	637,7	622,9
5	SEDAR - rec ($t_i = 90s$); $N_{cápu} = 6$	583,2	640,1	597,5	656,6
6	SEDAR - rec ($t_i = 120s$); $N_{cápu} = 4$	576,2	633,2	620,5	605,3

En primer lugar, se analiza lo que ocurre con la multiplicación de matrices. Si se observa la fila 3 de la Tabla 5, se ve que el tiempo total de ejecución es prácticamente insensible al momento de inyección del fallo. Desde el punto de vista del funcionamiento de SEDAR-det, este es el peor escenario posible, ya que todos los fallos inyectados son detectados casi sobre el final de la ejecución. Esto se debe al comportamiento particular de la aplicación de prueba, y su interacción con el mecanismo de detección. Como se ha explicado, la multiplicación de matrices es fuertemente limitada por cómputo, y las fases de comunicación son breves y están ubicadas al comienzo y al final. Esto produce que, para los tres instantes de inyección que hemos seleccionado, la detección se produzca siempre durante la última comunicación colectiva (SEDAR_Gather). Como consecuencia, cuando el fallo se inyecta sobre el final la latencia de detección es baja; pero cuando el fallo se inyecta antes de comenzar la fase de cómputo, permanece latente por mucho tiempo antes de ser detectado.

Por lo tanto, el tiempo total de ejecución va creciendo a medida que el fallo tarda más en ser detectado, ya que se debe sumar el tiempo de una ejecución completa (fijo) a la fracción que haya llegado a realizarse de la ejecución original (variable). Cuanto más tempranamente puedan ser detectados los fallos (menor latencia), mejor será el desempeño de la estrategia de detección y relanzamiento.

Recomendaciones

- Respecto al sistema de eficiencia energética escalable, se consiguió diseñar un sistema realmente escalable, en donde cada nodo puede ser, por ejemplo, cada una de las aulas y oficinas de la Universidad. Se pudo comprobar, que la reducción del consumo lograda es notable respecto al uso habitual de la energía eléctrica sin sistema de control, es por esto que se puede considerar que un sistema de iluminación es eficiente energéticamente cuando permite adaptar el

nivel de iluminación en función de la variación de la luz solar y de la detección o no de presencia. En función de los resultados se considera que las soluciones existentes no han sido suficientemente desarrolladas, ya que para un uso eficiente de la energía se deben tener en cuenta varios métodos y factores para lograr un algoritmo potente que permita optimizar el consumo manteniendo el confort en el usuario. La crisis energética global se podrá reducir explotando este tipo de sistemas y la generación de energía no convencional.

- En función de las soluciones tecnológicas desarrolladas para el sector fruti-hortícola, aunque todavía no se pudieron realizar las pruebas en campo, la posibilidad de acceder a los parámetros de forma remota y del control automático de variables críticas como la humedad del suelo y la iluminación, simplifica mucho la labor del productor y provee flexibilidad para la realización de otras tareas. Teniendo en cuenta la posibilidad de acceder a los parámetros desde una aplicación móvil, se simplifica muchísimo la capacitación del usuario, debido a que en la actualidad todo el mundo tiene un Smartphone.

- El contexto actual de pandemia hizo que se dé un gran paso en la educación a distancia, esta línea de investigación tiene un gran potencial, debido a que claramente la tendencia actual es pasar a una modalidad semi presencial. Esta tendencia proveerá flexibilidad dando paso a una nueva alternativa de acceso a la educación para el que se ve imposibilitado por las distancias. Aunque el Laboratorio de ondas fue implementado y se encuentra funcional, a causa de la cuarentena no se pudo poner en funcionamiento para que lo usen las comisiones de física, pero ha abierto una puerta para el desarrollo de nuevos laboratorios en otras materias. Además, nos permitió formar parte de la red Nacional de laboratorios remotos, posibilitando a la Universidad utilizar herramientas desarrolladas por otras Universidades y viceversa.

- Respecto a la obtención de eficiencia en el procesamiento de imágenes médicas, se obtuvieron importantes beneficios de las técnicas de concurrencia y paralelismo. Se lograron reducir notablemente los tiempos de ejecución de los diferentes algoritmos de procesamiento de imágenes, quedando en

evidencia como se puede incrementar la eficiencia con el aumento de procesadores, principalmente por las características paralelizables de los datos de imágenes (ver [31]).

- En relación con la tolerancia a Fallos, la ejecución de programas que procesan grandes cantidades de datos en sistemas de procesamiento de altas prestaciones tiene fallos cuando el consumo se incrementa drásticamente por la exhaustiva utilización de los recursos. En base a esto y a las características de dichos fallos, se desarrolló la línea que derivó en estudios de doctorado. Cabe destacar que esta problemática es cada vez más común, debido las nuevas tecnologías que implementan inteligencia artificial y procesamiento de grandes cantidades de datos, por lo que el software de detección diseñado, aunque es puramente experimental será una pieza fundamental para el monitoreo y corrección de fallos en los sistemas de procesamiento actuales (ver [32]).

Conclusiones

Internet de las cosas aplicado a eficiencia energética, monitoreo y seguimiento de cultivos y educación a distancia

Los sensores utilizados en el sistema de eficiencia energética escalable entregan señales digitales y analógicas en un determinado rango, éstos fueron acondicionados para ser utilizados por el sistema y almacenados en la base de datos. Con respecto al sensor de iluminación, éste debe ser calibrado en función del lugar donde se instale el sistema, teniendo en cuenta las condiciones de uso de la iluminación (ejemplo, para lectura). Con el uso de las librerías que provee el microcontrolador, se facilitó el desarrollo del firmware del sistema de control, logrando muy buenos resultados en las pruebas de funcionamiento. La utilización de una base de datos disponible en un servidor comercial provee al sistema de respaldo e integridad de la información recabada, que podrá usarse para conocer los hábitos del usuario, para tener una referencia histórica del consumo y conocer el ahorro de energía que se logró a lo largo del tiempo. Finalmente, se puede concluir que se logró un sistema robusto capaz de determinar presencia en la sala/oficina independientemente del sensor de movimiento. Esto evita que se active el sistema innecesariamente y permite

optimizar el consumo en función de los hábitos del usuario [18]. El proyecto garantiza el uso eficiente del sistema de iluminación ya que permite adaptar el nivel de luz del ambiente en función de la variación de la luz solar y en función de la presencia o ausencia de personas.

Si bien el sistema es funcional en cuanto a la gestión del uso energético, con el esp01 no se logró una conectividad eficaz estable. Por lo tanto, fue necesario utilizar una placa con módulo Wifi integrado en el microcontrolador (kit NODE MCU) para garantizar la estabilidad de la conexión con el servidor.

Este proyecto asegura el uso eficiente del sistema de iluminación de manera automatizada en base a las mediciones en tiempo real realizadas por los sensores. La gestión de la luz se basa únicamente en el encendido o apagado de luces, pero la eficiencia energética puede mejorarse variando las intensidades de iluminación mediante la utilización de un dimmer.

Este trabajo se puede ampliar de diversas maneras:

- Variar las intensidades de luces para los distintos ambientes del edificio, por ejemplo, en la biblioteca de una universidad se necesita una buena iluminación por lo que se deberían encender las luces al 100% pero en los pasillos podría prenderse la luz a un 50%
- Utilizar varios sensores de iluminación dispuestos en distintos puntos de un ambiente y sectorizar al mismo, es decir, en un ambiente grande los lugares próximos a una ventana pueden contar con luz solar pero los más alejados no, por lo que se podrían encender las luces sólo de los sectores del ambiente que no posean luz solar
- Que el encendido o apagado de luces se determine del lado del servidor a partir de un algoritmo inteligente que analizando los datos almacenados pueda detectar cuales son los horarios más concurridos o en los que se ausenta la luz solar, y que a su vez el microcontrolador tenga la capacidad de gestionar por sí mismo el uso del sistema de iluminación en caso de un fallo de conexión con el servidor [19].

Respecto al monitoreo de cultivos bajo cubierta y a cielo abierto, se puede concluir que pese a los problemas de conectividad mediante las antenas celulares, al tratarse de parámetros lentos (cambian lentamente a lo largo del tiempo), el sistema responde correctamente y permite almacenar una gran cantidad de valores. Actualmente, las pruebas fueron realizadas en una dashboard gratuita en los servidores de thingspeak. Pero se plantea

continuar con la línea en el siguiente proyecto con la idea de desarrollar nuestro propio servidor con herramientas de monitoreo y control personalizadas. De las tres aplicaciones implementadas del nodo principal, se concluye que el servidor web implementado en raspberry resultó muy eficiente no solo para el almacenamiento de los datos, sino también para la actualización de la aplicación web. Una de las aplicaciones en Python mediante tres procesos (para lectura de sensores, actualización web y control de parámetros) resultó ideal para no sobrecargar al servidor cuando llegan una gran cantidad de conexiones a la vez.

En cuanto al diseño del Laboratorio remoto, se implementó todo el sistema de control del HW (amplitud y frecuencia de la onda) y el manejo de la cámara desde una minicomputadora raspberry pi. Se implementó y probó correctamente la plataforma de acceso remoto y el sistema de asignación de turnos. El sistema desarrollado, es el comienzo de la implementación de una amplia variedad de laboratorios remotos, que contribuirá a la mejora de la enseñanza en tiempos de pandemia. Se supone que el uso de LR en el diseño y el desarrollo de actividades didácticas promoverá el enriquecimiento de las prácticas de enseñanza en los docentes. Se espera además que la disponibilidad de LR en el marco de la propuesta de enseñanza que se diseñó fortalezca la autonomía de los estudiantes respecto a la apropiación del conocimiento. Esperamos que estos resultados impacten en los estudiantes de manera de aumentar su aprendizaje acerca de los contenidos que se abordan [9].

Procesamiento de imágenes biomédicas mediante técnicas de procesamiento paralelo en sistemas multi-core y tolerancia a fallos

En cuando al procesamiento de imágenes sobre dispositivos FPGAs, el tiempo de procesamiento se disminuye casi al 50% al utilizar dos procesadores. Se han obtenido tiempos muy buenos en los filtros de mediana y de media, pero este se duplica al ejecutar algoritmos con mucho cálculo matemático como el algoritmo de sobel. Adicionalmente, se logró mejorar aún más el tiempo de procesamiento ejecutando la parte no común de los algoritmos en un co-procesador (bloque de HW implementado en VHDL), esto permitió

ejecutar las operaciones matemáticas de los algoritmos de forma concurrente en lugar de hacerlo de forma secuencial dentro de cada procesador.

En este trabajo se demuestra la potencialidad de las plataformas FPGAs para implementar sistemas multi-core. La flexibilidad de estos sistemas permitió evaluar diferentes diseños en la búsqueda de eficiencia al momento de procesar algoritmos de imágenes con características específicas. Mediante los resultados obtenidos se demostró que el sistema diseñado permite procesar de manera eficiente los algoritmos de procesamiento, aprovechando las características paralelizables que proveen los datos de este tipo de algoritmos. Los tiempos obtenidos en la ejecución de los filtros usando uno y dos procesadores fueron muy satisfactorios, debido a que cada procesador accede a datos ubicados en diferentes partes de la memoria simultáneamente, gracias a la memoria compartida con múltiples puertos [29].

En cuanto a la tolerancia a los fallos transitorios se ha diseñado e implementado SEDAR, una metodología que permite detectar los fallos transitorios y recuperar automáticamente las ejecuciones, aumentando la fiabilidad y la robustez en sistemas en los que se ejecutan aplicaciones paralelas determinísticas de paso de mensajes, de una manera agnóstica a los algoritmos a los que protege [28]. Aquí se aplica una estrategia de detección basada en la replicación de procesos y el monitoreo de las comunicaciones. Además, se desarrolló un mecanismo de recuperación basado en el almacenamiento de un conjunto de checkpoints distribuidos de capa de sistema. La recuperación es automática y se realiza regresando atrás al checkpoint correspondiente. Para la evaluación, se ha desarrollado un conjunto de casos de prueba. Además, se ha diseñado una estrategia de recuperación alternativa, para el caso en el que se cuente con checkpointing de capa de aplicación. En este caso, es posible almacenar sólo el último checkpoint luego de validarlo [25].

Referencias bibliográficas

- [1] Leila Fatmasari Rahman, "Choosing your IoT Programming Framework: Architectural Aspects", 2016 IEEE 4th International Conference on Future Internet of Things and Cloud
- [2] Osio J., C. Aquarone, E. Hromek, J. Salvatore, "Plataforma de desarrollo pasra IoT", IV conaiisi, 2017

- [3] A. J. Anaya, D. Peluffo, “Sistema de Riego Basado En La Internet De Las Cosas (IoT)”, Fica, 2016.
- [4] Eficiencia energética en los edificios, José María Fernández Salgado, AMV ediciones, 2011
- [5] Domotica e Inmotica - Viviendas y Edificios Inteligentes, 2. Ed. (Spanish Edition), Cristóbal Romero, Francisco Vazquez, editorial alfaomega, 2008
- [6] Energía Solar Fotovoltaica 2ª Edición, Miguel Aparicio, Edit. marcombo, 2009
- [7] LUCAS OLIVERA, JULISSA ATIA, LEONARDO AMET, JORGE OSIO, MARTÍN MORALES, MARCELO CAPPELLETTI. **MÉTODO BASADO EN ALGORITMOS GENÉTICOS Y EL MODELO DE HOVEL PARA LA EXTRACCIÓN DE PARÁMETROS DE LA EFICIENCIA CUÁNTICA EXTERNA EN CELDAS SOLARES.** XLI Reunión de Trabajo de la Asociación Argentina de Energías Renovables y Medio Ambiente, Argentina, 2018. ISBN: 978-987-29873-1-2
- [8] M A. Revuelta, Laboratorio remoto en un entorno virtual de enseñanza aprendizaje, FI- UNLP, 2016
- [9] MARÍA JOSELEVICH, ROBERTO E. ALONSO, PABLO GONZÁLEZ CASCO, MARTÍN MORALES, JORGE OSIO, ALEJANDRA SERIAL. **Proyecto de laboratorios remotos para la enseñanza de la física.** Argentina. San Juan. 2019. Libro. Artículo Completo. Workshop. XXI Workshop de Investigadores en Ciencias de la Computación.
- [10] John L. Semmlow: *Biosignal and Biomedical Image Processing*, Marcel Dekker, New Jersey, 2004.
- [11] Dougherty, E.: *An Introduction to Morphological Image Processing*, SPIE, Bellingham, WA, 1992.
- [12] AndraKa Consulting Group, Inc.: “Digital Signal Processing for FPGAs,” Seminar Notes, 1999.
- [13] D. Montezanti, A. De Giusti, M. Naiouf, J. Villamayor, D. Rexachs, E. Luque, “A Methodology for Soft Errors Detection and Automatic Recovery”, in Proceedings of the 15th International Conference on High Performance Computing & Simulation (HPCS). ISBN: 978-1-5386-3250-5/17. IEEE, 2017, pp. 434

- [14] F. Cappello, A. Geist, W. Gropp, S. Kale, B. Kramer, and M. Snir, "Toward exascale resilience: 2014 update," Supercomputing frontiers and innovations, vol. 1, no. 1, pp. 5–28, 2014
- [15] J. Osio, D Montezanti, E. Kunysz, Morales M., "Análisis de eficiencia y tolerancia a fallo en Arquitecturas Multiprocesador para aplicaciones de procesamiento de datos ", UNNE, Corrientes, WICC 2018. ISBN: 978-987-3619-27-4
- [16] J. Osio, J. Salvatore , D. Alonso , V. Guarepi , M. Cappelletti , M. Joselevich , M. Morales, "**Tecnologías de la información y las comunicaciones mediante IoT para la solución de problemas en el medio socio productivo**", UNNE, Ciudad de Corrientes, WICC 2018. ISBN: 978-987-3619-27-4
- [17] J. Osio, M. Cappelletti , G. Suárez, L. Navarro, F. Ayala, J. Salvatore , D. Alonso ,D. Encinas, M. Morales, "**Diseño de aplicaciones de IoT para la solución de problemas en el medio socio productivo**", UNSJ, San Juan, WICC 2019.
- [18] Gabriela Suarez, Jorge Osio, Marcelo Cappelletti. Diseño de un sistema de eficiencia energética escalable. 2018. Libro. Artículo Completo. Congreso. IX Congreso de Microelectrónica Aplicada.
- [19] Suarez Gabriela, Osio Jorge, Cappelletti Marcelo, "**Diseño de un sistema de control de energía**". 6to CONGRESO NACIONAL DE INGENIERÍA EN INFORMÁTICA/SISTEMAS DE INFORMACIÓN. 2018. ISSN: 2347-0372
- [20] Hromek Erik, Gómez Mauro, Olivera Lucas, Salina Mauro, Osio Jorge R, Cappelletti Marcelo, Morales D. Martín. **Sistema de control y monitoreo remoto de variables en aplicaciones agroindustriales**. Argentina. San Justo. 2019. Libro. Artículo Completo. Congreso. 7° Congreso Nacional de Ingeniería Informática/Sistemas de Información.
- [21] Nicole Denon, Matías Benary, Santiago Doti, Jorge Osio. **Desarrollo de un software para la automatización y control de un invernadero a través de una Raspberry Pi 3**. San Justo. 2019. Libro. Poster. Congreso. 7° Congreso Nacional de Ingeniería Informática/Sistemas de Información.
- [22] Leonardo Cabral, Gonzalo Negro , Emanuel Vigil , Elina Lo , Facundo Fain , Julissa Atia , Jorge Osio , Marcelo Cappelletti. **Sistema de seguimiento y control de**

parámetros para cultivos intensivos bajo cubierta. San Martín. 2019. Libro. Artículo Completo. Congreso. X Congreso de Microelectrónica Aplicada.

[23] J. Osio, D Montezanti, E. Kunysz, Morales M., “Determinación de la eficiencia y Estrategias de Tolerancia a Fallos en Arquitecturas Multiprocesador para aplicaciones de procesamiento de datos”, UNSJ, San Juan, WICC 2019.

[24] Kunysz E, Osio J., Rapallini J., Morales D.M., *Análisis de Eficiencia en Sistemas de Cómputo de Alta Performance Reconfigurables*. Revista Tecnología y Ciencia. Argentina. 2018. ISSN: 1666-6933

[25] JORGE R. OSIO, MARCELO CAPPELLETTI, MAURO SALINA, MAURO GOMEZ, LEONEL NAVARRO, JUAN. E SALVATORE, DANIEL ALONSO, DIEGO ENCINAS, MARTÍN MORALES. *Tecnologías de la información y las comunicaciones mediante IoT aplicadas a soluciones en el medio productivo y medioambiental*. Argentina. Santa Cruz. 2020. Libro. Artículo Completo. Workshop. XXII Workshop de Investigadores en Ciencias de la Computación.

[26] LUCAS OLIVERA, JULISSA ATÍA, JORGE OSIO, MARTIN MORALES, MARCELO CAPPELLETTI. **Estimación de la radiación solar global diaria a través de modelos de redes neuronales artificiales**. Argentina. San Francisco, Córdoba. 2020. Libro. Artículo Completo. Congreso. 8° Congreso Nacional de Ingeniería Informática/Sistemas de Información (CoNaIISI 2020).

[27] LUCAS OLIVERA, JULISSA ATIA, LEONARDO AMET, JORGE OSIO, MARTIN MORALES, MARCELO CAPPELLETTI. **Uso de redes neuronales artificiales para la estimación de la radiación solar horaria bajo diferentes condiciones de cielo**. Revista Avances en Energías Renovables y Medio Ambiente. Vol. 24, pp 232-243 (2020) ISSN 2314-1433 ASADES 2020 EDICIÓN ESPECIAL.

[28] JORGE R. OSIO; DIEGO MONTEZANTI; MARCELO CAPPELLETTI; EDUARDO KUNYSZ; MARTIMORALES. **Determinación de la eficiencia en el procesamiento sobre Arquitecturas Multiprocesador y Estrategias de Tolerancia a Fallos en HPC**. Argentina. Santa Cruz. 2020. Libro. Artículo Completo. Workshop. XXII Workshop de Investigadores en Ciencias de la Computación.

[29] *Jorge Osio, Walter Aroztegui, Antonio Quijano, Jose Rapallini, “Determinación de eficiencia en la ejecución de algoritmos de procesamiento de imágenes con múltiples*

procesadores en FPGA”, CASE 2019, Facultad de Ciencias Exactas Ingeniería y Agrimensura de la Universidad Nacional de Rosario, Santa Fe, Argentina.

[30] William Stallings, Organización y Arquitectura de Computadores, Editorial Prentice Hall, 8va edición, 2010.

[31] Osio Jorge Rafael, “Procesamiento digital de imágenes médicas sobre plataformas FPGAs”, Tesis de Maestría, 2019.

[32] Montezanti Diego Miguel, “SEDAR: Detección y Recuperación Automática de Fallos Transitorios en Sistemas de Cómputo de Altas Prestaciones”, Tesis de Doctorado, 2020.