

Vol.10 Num,2



RECIBE

Revista electrónica
DE COMPUTACIÓN, INFORMÁTICA, BIOMÉDICA Y ELECTRÓNICA

ISSN 2007-5448



Índice

Computación e Informática

- **Paralelismos entre bases de datos relacionales y no relacionales (un enfoque en seguridad)** **C1**
Washington Chingo Esquivel, Galo López Sevilla
- **Reconocimiento de palabras de la Lengua de Señas Mexicana utilizando información RGB-D** **C2**
Felipe Trujillo Romero, Gibran García Bautista
- **Adaptación al uso tecnología en el ámbito educativo durante la pandemia derivada del COVID-19 en México** **C3**
Adriana Peña Pérez Negrón, Carlo Giovanella, Aurora Espinoza Valdez, Mirna Muñoz, David Bonilla Carranza, Marcello Pasarrelli
- **UML para modelar patrones de contaminación de las Ciudades Inteligentes para la Zona Metropolitana de Guadalajara** **C4**
Martha Patricia Martínez Vargas, Elsa Estrada Guzmán, Roció Maciel Arellano
- **Clasificación de parásitos en imágenes de copro con transferencia de aprendizaje y aumento de datos** **C5**
Miguel Ángel Gutiérrez Velázquez, Mario Ignacio Chacon Murguía, Juan Alberto Ramírez Quintana, Carlos Arzate Quintana, Alma Delia Corral Saenz

Paralelismos entre bases de datos relacionales y no relacionales (un enfoque en seguridad)

Parallelisms between relational and non-relational databases (a security approach)

Washington Chingo Esquivel¹
bryanbyr@gmail.com

Galo López Sevilla¹
glopez@pucesa.edu.ec

¹Pontificia Universidad Católica del Ecuador - Sede Ambato

Resumen: Las bases de datos nacen como una herramienta para el almacenamiento estandarizado en las aplicaciones, con el avance tecnológico surgen diferentes enfoques a la forma de relacionar los datos que se almacenan en las mismas. Comparar dos tecnologías con una misma finalidad, pero con diferente filosofía puede ayudar a comprender de mejor manera para que fue concebida cada una; y tener una mejor perspectiva sobre cómo se pueden complementar y mejorar, a través de distintas prácticas. A pesar de existir diferentes estudios comparativos sobre las bases de datos relacionales y no relacionales, todos se basan en el rendimiento y no en la seguridad. El estudio tiene como objetivo comparar las seguridades en bases de datos relacionales y no relacionales con la finalidad de encontrar diferencias y similitudes entre bases de datos relacionales y no relacionales. La metodología de comparación, se realiza con servidores en la nube aporta un ambiente apropiado para realizar el experimento entre dos bases de datos muy conocidas y de acceso libre PostgreSQL y MongoDB, siendo la primera un tipo de base relacional y la segunda no relacional (NoSql), para generar como resultado una comparativa que se difunda en la comunidad científica.

Palabras Clave: Seguridad de la información; Bases de Datos; Bases de Datos Relacionales; Bases de Datos No Relacionales; PostgreSQL; MongoDB.

Abstract: Databases were born as a tool for standardized storage in applications, with technological advance it arises different approaches of how to relate the data that is stored on them. Comparing two technologies with the same purpose, but with a different philosophy, can help to understand why each one was conceived; and have a better perspective on how they can be complemented and improved, through different practices. Despite there are different comparative studies on relational and non-relational databases, all based on performance and not in security perspective. The study aims to compare the securities in relational and non-relational databases in order to find differences and similarities between relational and non-relational databases. The comparative methodology, which is carried out with servers in the cloud, provides an appropriate environment to carry out the experiment between two well-known and open-access databases PostgreSQL and MongoDB, the first being a type of relational base and the second is an non-relational (NoSql), to generate as a result a comparison that is disseminated in the community scientific.

Keywords: Information Security; Databases; Relational Databases; Non-Relational Databases; PostgreSQL; MongoDB.

1. Introducción

Las bases de datos son parte fundamental de los sistemas informáticos, Coronel et al. (2010) define una base de datos como una estructura informática compartida e integrada que almacena una colección, mientras que Date (2001) considera a las bases de datos como una especie de armario electrónico para archivar; es decir, es un depósito o contenedor de una colección de archivos de datos computarizados, para administrar una base de datos es necesario un sistema gestor de bases de datos o DBMS (por sus siglas en inglés, *Data Base Management System*) para Sánchez Asenjo (2009) es el software que permite a los usuarios procesar, describir, administrar y recuperar los datos almacenados en una base de datos. En estos Sistemas se proporciona un conjunto coordinado de programas, procedimientos y lenguajes que permiten a los distintos usuarios realizar sus tareas habituales con los datos, además de garantizar la seguridad de los mismos.

Existen diferentes tipos de bases de datos que van desde las bases de datos jerárquicas, en red, relacionales y no relacionales. Codd (1970) diseñó el modelo de bases de datos relacionales en el que propone que: Los usuarios interactúen con un modelo relacional de los datos que consiste en una colección de relaciones que varían en el tiempo y cada usuario no necesita saber más acerca de cualquier relación que su nombre junto con los nombres de sus dominios. Durante muchos años las bases de datos relacionales fueron las predilectas por su facilidad de uso, pero en 2005 Google enfrentó un problema con el manejo de grandes cantidades de datos ya que las bases de datos relacionales, eran inadecuadas para hacer frente a los volúmenes y la velocidad de los datos que enfrentaba Google. Para Harrison (2015) los retos que las empresas enfrentan hoy en día con "*big data*" son problemas que Google encontró por primera vez hace casi 20 años. Acorde a Edward y Sabharwal (2015) Carlo Strozzi acuñó el término NoSQL en 1998, al usar este término para identificar su base de datos porque su base de datos no tenía una interfaz SQL por lo que NoSQL es un término general para los almacenes de datos que no siguen los principios de DBMS relacionales. Las bases de datos no relacionales comienzan a tener una gran importancia para el manejo de grandes cantidades de información a partir de los problemas enfrentados por Google.

El DBMS orientado a objetos PostgreSQL (llamado Postgres95) está derivado del paquete Postgres escrito en la universidad de Berkeley. Perkins et al. (2018) detalla las siguientes características: "tiene compatibilidad integrada con Unicode, secuencias, herencia de tablas y sub-selecciones, y es una de las bases de datos relacionales más compatibles con ANSI SQL del mercado", esto sumado a su longevidad le convierte en uno de los DBMS más probados y estables existentes.

MongoDB fue diseñado como una base de datos escalable, el nombre Mongo proviene de "humongous", con el rendimiento y el fácil acceso a los datos como núcleos objetivos de diseño. Para Perkins et al. (2018) MongoDB es una base de datos de documentos, que le permite almacenar objetos anidados a la profundidad que desee, y puede consultar esos datos anidados en un modo *ad hoc*. No aplica ningún esquema (similar a HBase), por lo que los documentos pueden contener campos o tipos que ningún otro documento en la colección contiene.

Las bases de datos en la actualidad están marcadas por estos dos grupos: relacionales y no relacionales (SQL y NoSQL), existen varios estudios que comparan las ventajas y desventajas a nivel de rendimiento entre ambos grupos, sin embargo, no existe la suficiente bibliografía en la que compare las seguridades de las mismas, para esto es necesario tener en cuenta sus filosofías y principios, ACID en el caso de las bases de datos relacionales y BASE en el caso de las bases de datos no relacionales. Para realizar la comparación se ha seleccionado PostgreSQL uno de los DBMS relacionales más extendidos y MongoDB uno de los DBMS no relacionales más usados, para poder hallar los paralelismos de seguridad existentes entre bases de datos relacionales y no relacionales.

Según Coronel et al. (2010) las bases de datos relacionales reflejan los eventos del mundo real a través de transacciones cuyas propiedades son atomicidad, consistencia, aislamiento y durabilidad:

- La atomicidad (*atomicity*) requiere que se completen todas las operaciones (solicitudes SQL) de una transacción; si no, la transacción se cancela.
- La consistencia (*consistence*) indica la permanencia del estado coherente de la base de datos. Una transacción lleva una base de datos de un estado coherente a otro estado coherente; si alguna de las partes de la transacción viola una restricción de integridad, se aborta toda la transacción.

- Aislamiento (*isolation*) significa que los datos utilizados durante la ejecución de una transacción no pueden ser utilizados por una segunda transacción hasta que se complete la primera.
- La durabilidad (*durability*) garantiza que una vez que se realizan (confirman) los cambios de transacción, no se pueden deshacer ni perder, incluso en el caso de una falla del sistema.

Las Bases de Datos no relacionales por su parte tienen los principios BASE que se basan en el Teorema CAP, para Gilbert y Lynch (2012) el teorema presenta tres requerimientos de un sistema distribuido, sólo es posible proporcionar dos al mismo tiempo como se observa en la figura 1. Los tres requerimientos son:

- Consistencia. Todos los nodos ven los mismos datos al mismo tiempo.
- Disponibilidad. El fallo de uno o más nodos no impide a los demás seguir en funcionamiento.
- Tolerancia a las Particiones. El sistema continúa funcionando a pesar de pérdidas en los mensajes.

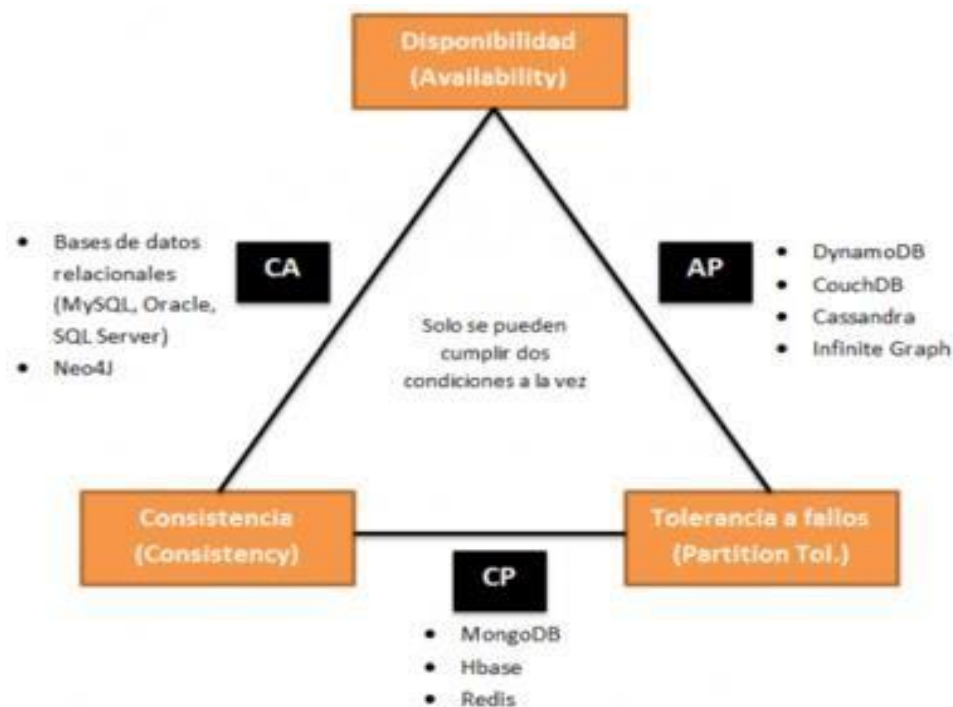


Figura 1. Clasificación de las Bases De Datos según El Teorema De CAP (Rubenfa 2014).

Edward y Sabharwal (2015) establecen las características de BASE que se utiliza en contraposición a ACID:

- *Basically Available* (básicamente disponible) significa que el sistema estará disponible en términos del teorema CAP.
- *Soft State* (estado suave) indica que incluso si no se proporciona ninguna entrada al sistema, el estado cambiará con el tiempo. Esto está de acuerdo con la consistencia eventual.
- *Eventual consistency* (consistencia eventual) significa que el sistema alcanzará consistencia a largo plazo, siempre que no se envíe información al sistema durante ese tiempo.

Con la comprensión de los requerimientos que busca cada tipo de base de datos, se puede identificar las seguridades de cada sistema y el porqué del enfoque de cada una de estas. De acuerdo al teorema CAP, se puede señalar que los dos tipos bases de datos convergen en el requerimiento de consistencia.

Seguridad Informática y Seguridad de la Información

La triada de la seguridad informática tiene como objetivos la confidencialidad, integridad y disponibilidad; la seguridad de la información agrega a estos tres objetivos la autenticación. La confidencialidad se refiere a una característica que asegura que solo los usuarios autorizados

puedan leer la información a la cual tengan acceso, la integridad asegura que la información no ha sido modificada por terceros, la disponibilidad asegura que la información pueda accederse bajo cualquier circunstancia y la autenticación asegura que el usuario sea quien dice ser.

2. Metodología

La comparación de los DBMS se realiza a través del método comparativo, que para Tonon (2011) objetivo la búsqueda de similitudes y disimilitudes en la modalidad descriptiva que no busca evaluar las características a estudiar, para el proceso de análisis se toma como referencia PostgreSQL ya que al iniciar el proceso de conexión al DBMS se encuentra la primera característica que es la autenticación, para después pasar a su contraparte en MongoDB, como siguiente paso se analiza la confidencialidad que asegura la conexión a los DBMS, por último la integridad y disponibilidad se analizan por medio de la concurrencia, por su parte la comparación de vulnerabilidades se realiza a través de una variante de la metodología Open Web Application Security Project (OWASP) en los ítems que se considere necesario como el caso de inyección de consulta (*QUERY Injection*). Se ha seleccionado las últimas versiones disponibles al momento de realizar las pruebas que son PostgreSQL 13.0 y MongoDB 4.4.1.

Autenticación (Control de acceso)

PostgreSQL al ser un DBMS relacional ocupa dos formas de control de acceso, Date (2001) conceptualiza el control discrecional “un usuario específico tendrá generalmente diferentes derechos de acceso (también conocidos como privilegios) sobre diferentes objetos”, también el mismo autor define control obligatorio “ cada objeto de datos está etiquetado con un nivel de clasificación determinado y a cada usuario se le da un nivel de acreditación”, en cuanto a MongoDB el control de acceso de acuerdo a Edward y Sabharwal (2015) se realiza en dos niveles “La autenticación verifica la identidad del usuario y la autorización determina el nivel de acciones que el usuario puede realizar en la base de datos autenticada”.

Las vistas que permiten comprobar estas características en PostgreSQL son `pg_user` y `pg_roles`, para esto existen comandos `\du` y `\dg` que equivalen a consultas SQL, en la figura 2 se realiza la consulta acerca de la información del usuario.

```
postgres=# \du+
***** CONSULTA *****
SELECT username AS role_name,
       CASE
         WHEN usesuper AND usecreatedb THEN
           CAST('superuser, create database' AS pg_catalog.text)
         WHEN usesuper THEN
           CAST('superuser' AS pg_catalog.text)
         WHEN usecreatedb THEN
           CAST('create database' AS pg_catalog.text)
         ELSE
           CAST(' ' AS pg_catalog.text)
       END role_attributes
FROM pg_catalog.pg_user ORDER BY role_name desc;
```

Figura 2. Consulta De Información Acerca Del Usuario.

Tanto `pg_user` como `pg_roles` son parte del esquema `pg_catalog` que contiene los metadatos e información sobre el funcionamiento interno de PostgreSQL por lo que tanto usuarios y roles pueden ser consultados por medio de este esquema, en la figura 3 se indica la consulta para los roles del usuario.

```

postgres=# \dg+
***** CONSULTA *****
SELECT r.rolname, r.rolsuper, r.rolinherit, r.rolcreatorole, r.rolcreatedb,
r.rolcanlogin, r.rolconndef, r.rolvaliduntil, ARRAY(SELECT b.rolname FROM
pg_catalog.pg_auth_members m JOIN pg_catalog.pg_roles b ON (m.roleid = b.oid)
WHERE m.member = r.oid) AS memberof, pg_catalog.shobj_description(r.oid,
'pg_authid') AS description, r.rolreplication FROM pg_catalog.pg_roles r
ORDER BY 1;

```

Figura 3. Consulta de Roles del Usuario.

Las dos consultas dan como resultado una salida similar a la presentada en la figura 4.

List of roles			
Role name	Attributes	Member of	Description
Someone		{}	THIS IS A ROLE COMMENT

Figura 4. Resultado de las consultas.

Como se resultado se obtiene información acerca del nombre de usuario, sus atributos, grupos y roles. MongoDB por su parte ocupa privilegios de acuerdo a cada base de datos, para esto dispone de métodos de manejo de usuarios y roles, para obtener información de los usuarios se ocupa el comando getUser() y para la información del rol del usuario getRole(), estos métodos se ejecutan a través de db que hace referencia a la base de datos actual y se puede comprobar con el siguiente comando observado en la figura 5.

```

use Nombredb
db.getRole("readWrite", { showPrivileges: true })
{
  "role" : "dbAdmin",
  "db" : "Nombre_db",
  "isBuiltin" : true,
  "roles" : [ ],
  "inheritedRoles" : [ ],
  "privileges" : [ {
    "resource" : {
      "db" : "Nombre_db",
      "collection" : ""
    },
    "actions" : [
      "collMod",
      "collStats",
      "compact",
      "convertToCapped",
      "createCollection",
      "createIndex",
      "dbStats",
      "dropCollection",
      "dropDatabase",
      "dropIndex",
      "enableProfiler",
      "indexStats",
      "planCacheIndexFilter",
      "planCacheRead",
      "planCacheWrite",
      "reIndex",
      "renameCollectionSameDB",
      "repairDatabase",
      "storageDetails",
      "validate"
    ]
  } ]
} ]

```

Figura 5. Comando GETROLE().

El resultado del comando es mucho más amplio y se ha recortado a la parte que es de interés para el estudio, se puede observar el nombre del usuario, sobre qué base de datos tiene privilegios, sus privilegios sobre la base de datos y las acciones que puede realizar.

Por lo tanto, en cuanto a autenticación y autorización los dos tipos de DBMS manejan iguales formas de control de acceso de datos, observando que según de The MongoDB Manual estas opciones vienen desactivadas por defecto, y se debe activar mediante la característica “*authorization: enabled*” en la configuración del DBMS.

Confidencialidad (Cifrado de datos)

La forma más aceptada para asegurar la confidencialidad es a través de la encriptación o cifrado de datos, aquí se encuentra un símil entre los dos DMS. PostgreSQL ocupa diferentes niveles de encriptación comenzado con encriptación de clave a través de la configuración de *password_encryption*, encriptación de columnas a través del módulo *pgcrypto*, encriptación de partición de datos, encriptación de datos a través de la red por medio de *hostssl* usando *Secure Socket Layer* (SSL), además de encriptación del lado del cliente. MongoDB por su parte permite encriptación de datos en movimiento a través de *Transport Layer Security* (TLS) y SSL, encriptación de datos en reposo donde de forma nativa se puede encriptar un archivo (desde la versión 3.2), además de encriptación del lado del cliente. Acorde a esto a pesar de ciertas diferencias la confidencialidad es un elemento que los sistemas gestores de bases de datos buscan asegurar.

A continuación, se muestra en la tabla 1 la comparativa de algoritmos de encriptación e intercambio de claves de los sistemas gestores de bases de datos.

Tabla 1. Comparativa entre los algoritmos de encriptación PostgreSQL y MongoDB

PostgreSQL	MongoDB
<ul style="list-style-type: none"> • MD5 • SHA1 • SHA224/256/384/512 • Blowfish • AES • DES/3DES/CAST5 • Raw encryption • PGP • Symmetric encryption • PGP Public-Key encryption 	<ul style="list-style-type: none"> • AES256-GCM • SCRAM-SHA-1* • SCRAM-SHA-256* <p>*Utilizadas para el intercambio de claves por medio de Public Key Infrastructure-PKI</p>

Fuente: elaboración propia, basado en Lockart (1996) y The MongoDB Manual.

El módulo *pgcrypto* se usaría de una manera similar a la de la figura 6.

```
psql CREATE EXTENSION pgcrypto; SELECT crypt('mypass', gen_salt('bf', 4));
crypt
-----
$2a$04$1bfMQDOR6aLyD4q3Kvb8/ujG7ZAkyie4d/s3ABwuZNbzkFFgXtC76
```

Figura 6. Ejemplo Criptografía PostgreSQL.

En MongoDB se puede comprobar la encriptación por medio del comando mostrado en la figura 7.

```

db.system.users.findOne({user:dbAdmin})
{
  "_id": "admin.dbAdmin",
  "userId": "UUID("f325412e-1943-4a47-b637-ecbcafe58cf5)",
  "user": "dbAdmin",
  "db": "admin",
  "credentials": {
    "SCRAM-SHA-1" : {
      "iterationCount": 10000,
      "salt": "jGWNJKQz8UWIdH8GTyRw==",
      "storedKey": "5uuLgYr0zCBcZ0/rgFy0XhWDFQY=",
      "serverKey": "LfQdMqNa68JWESYebyg86G9tueo"
    },
    "SCRAM-SHA-256" : {
      "iterationCount": 15000,
      "salt": "oOyPFEfITiIRvOfvFeuTuXQAe0cFNP60vba14g==",
      "storedKey": "6nwbTJfub5nR4/7+Wgo0kNvWD86uIUDsHe03/tmgY=",
      "serverKey": "Htw2GA/VQwZbFgh5soqAVy0IDy1v7YPDgj0rmfD0/g="
    }
  },
  "roles": [
    {
      "role": "root",
      "db": "admin"
    }
  ]
}

```

Figura 7. Ejemplo Criptografía MongoDB.

Concurrencia

La concurrencia en PostgreSQL se basa en Control de la Concurrencia MultiVersión (por sus siglas en ingles MMVC - *Multi-Version Concurrency Control*), Bernstein y Goodman (1983) establecen el control de concurrencia como la actividad de sincronizar operaciones emitidas al ejecutar concurrentemente programas en una base de datos compartida. La figura 8 muestra como MMVC, evita la sobrecarga de bloqueo, ya que evita que el bloque 2 cree un bloqueo que habría impedido que la sesión 2 lea los datos hasta que se complete la transacción de la sesión 1.

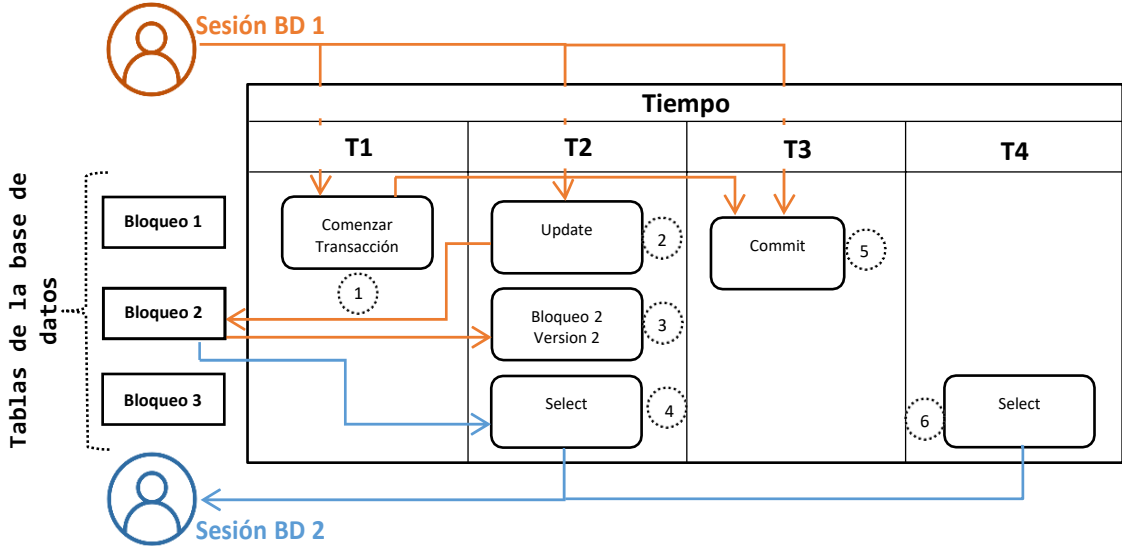


Figura 8. Funcionamiento De MMVC basado en Harrison (2015)

Transacciones Serializables

Una ejecución serial de las transacciones significa que éstas se ejecutan una después de otra, sin cualquier intercalación de operaciones (Ricardo, 2009). Si X y Y son dos transacciones, sólo hay dos posibilidades de ejecución seriales: se termina la transacción X y después se ejecuta Y, o se termina la transacción Y para luego ejecutar X. PostgreSQL maneja la serialización de transacciones a través del aislamiento transaccional por medio de tres acciones no deseadas conocidas como lectura sucia, lectura no repetible y lectura fantasma y cuatro niveles de aislamiento, como se observa en la tabla 2.

Tabla 2. Niveles de aislamiento de PostgreSQL

	Lectura "sucia"	Lectura no repetible	Lectura "fantasma"
Lectura no cursada	Posible	Posible	Posible
Lectura cursada	No Posible	Posible	Posible
Lectura repetible	No Posible	No Posible	Posible
Serializable	No Posible	No Posible	No Posible

Fuente: Elaboración propia basado en Lockhart (1996)

De acuerdo con Lockhart (1996) en el manual de usuario de PostgreSQL las acciones no deseadas se definen como:

- Lecturas sucias. Una transacción lee datos escritos por una transacción no esperada, no cursada.
- Lecturas no repetibles. Una transacción vuelve a leer datos que previamente había leído y encuentra que han sido modificados por una transacción cursada.
- Lectura fantasma. Una transacción vuelve a ejecutar una consulta, devolviendo un conjunto de filas que satisfacen una condición de búsqueda y encuentra que otras filas que satisfacen la condición han sido insertadas por otra transacción cursada.

En cuanto a niveles de aislamiento Lockhart (1996) enumera:

- Nivel de lectura cursada: Nivel de aislamiento por defecto, una consulta a este nivel sólo ve datos cursados antes de que la consulta comenzara. Si una fila devuelta por una consulta mientras se ejecuta una declaración UPDATE (o DELETE, o SELECT FOR UPDATE) está siendo actualizada por una transacción concurrente no cursada, entonces la segunda transacción que intente actualizar esta fila esperará a que la otra transacción se curse o pare. Los resultados de la ejecución de SELECT o INSERT (con una consulta) no se verán afectados por transacciones concurrentes.
- Nivel de aislamiento serializable: Nivel más alto de aislamiento transaccional, una consulta en este nivel sólo ve los datos cursados antes de que la transacción comience y nunca ve ni datos sucios ni los cambios de transacciones concurrentes cursados durante la ejecución de la transacción. La ejecución de una declaración UPDATE (o DELETE, o SELECT FOR UPDATE) está siendo actualizada por una transacción concurrente no cursada, la segunda transacción que trata de actualizar esta fila esperará a que la otra transacción se curse o pare. Los resultados de la ejecución de SELECT o INSERT (con una consulta) no se verán afectados por transacciones concurrentes.

Las transacciones y sus diferentes estados se pueden comprobar a través de la vista pg_stat_activity, como se muestra en la figura 9.

```
select * from pg_stat_activity;
datid|datname|pid|leader_pid|usesysid|username|application_name|
client_addr|client_hostname|client_port|backend_start|xact_start|query_start|state_change|
wait_event_type|wait_event|state|backend_xid|backend_xmin|query|backend_type
-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
Activity|AutoVacuumMain|60|10|10|postgres|autovacuum launcher
-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
Activity|LogicalLauncherMain|62|10|10|postgres|logical replication launcher
-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
```

Figura 9. Consulta De Transacciones Serializables.

Como se observa en la figura la consulta regresa un registro de las transacciones que se realizan desde el inicio del DBMS y tienen los siguientes pid que es identificador de la transacción, usesysid identificador del usuario, username con el nombre del usuario, backend_start que indica cuando realmente se realizó la transacción en caso de haber una serie y tiene las variaciones de milisegundos entre transacciones, además se observa el tipo de evento en espera, el evento, la consulta.

Bloqueos

Los candados (también llamados bloqueos) y las estampas de tiempo son las dos técnicas que generalmente se usan para garantizar la seriability de transacciones concurrentes (Ricardo, 2009). PostgreSQL ofrece varios modos de bloqueo para controlar el acceso concurrente a los datos en tablas. Algunos de estos modos de bloqueo los adquiere PostgreSQL automáticamente antes de la ejecución de una declaración, mientras que otros son proporcionados para ser usados por las aplicaciones.

PostgreSQL presenta bloqueos a nivel de tabla y fila, de acuerdo con Lockhart (1996) los bloqueos a nivel de tabla son:

- AccessShareLock modo de bloqueo adquirido automáticamente sobre tablas que están siendo consultadas.
- RowShareLock adquirido por SELECT FOR UPDATE y LOCK TABLE para declaraciones IN ROW SHARE MODE.
- RowExclusiveLock adquirido por UPDATE, DELETE, INSERT y LOCK TABLE para declaraciones IN ROW EXCLUSIVE MODE.
- ShareLock adquirido por CREATE INDEX y LOCK TABLE para declaraciones IN SHARE MODE.
- ShareRowExclusiveLock adquirido por LOCK TABLE para declaraciones IN SHARE ROW EXCLUSIVE MODE.
- ExclusiveLock adquirido por LOCK TABLE para declaraciones IN EXCLUSIVE MODE.
- AccessExclusiveLock adquirido por ALTER TABLE, DROP TABLE, VACUUM y LOCK TABLE.
- Bloqueos a nivel de fila: se producen cuando campos internos de una fila son actualizados (o borrados o marcados para ser actualizados). SELECT FOR UPDATE modificará las filas seleccionadas marcándolas, de tal modo que se escribirán en el disco. Los bloqueos a nivel de fila no afectan a los datos consultados. Estos son usados para bloquear escrituras a la misma fila únicamente.

Los bloqueos se pueden comprobar por medio de la consulta mostrada en la figura 10.

```
SELECT * from pg_locks pl LEFT JOIN pg_stat_activity psa ON pl.pid=psa.pid;
locktype|database|relation|page|tuple|virtualxid|transactionid|
virtualtransaction|classid|objid|objsubid|pid|mode|granted|
fastpath|datid|datname|pid|leader_pid|usesysid|username|
application_name|client_addr|client_hostname|client_port|backend_start|xact_start|
query_start|state_change|wait_event_type|wait_event|state|backend_xid|backend_xmin|
query|backend_type

relation|13395|12250|--|--|--|--|
--|--|--|3/20|66|AccessShareLock|t|
t|1395|postgres|66|--|10|postgres|
psql|127.0.0.1|--|38346|2020-11-25 16:33:11.130921+00|--|
--|--|--|--|active|--|486|
SELECT * from pg_locks pl LEFT JOIN pg_stat_activity psa ON pl.pid=psa.pid|clientbackend
```

Figura 10. Consulta De Bloqueos En PostgreSQL.

La consulta retorna el tipo de bloqueo, id de la base de datos, la relación en este caso AccessSharelock que se adquiere automáticamente al ser la tabla consultada y como se observa lo que se hizo fue una consulta a pg_locks, además se tienen los mismos campos que en la consulta de transacciones.

Bloqueos en MongoDB

MongoDB no implementa un sistema MVCC y, por lo tanto, la lectura no se puede realizar en un documento que se está actualizando. MongoDB de forma predeterminada proporciona un documento estricto único de consistencia, en una implementación de un solo servidor en una base de datos. Cuando se modifica un documento de MongoDB, se bloquea tanto para lecturas como para escrituras para otras sesiones. En implementaciones de conjuntos de réplicas, es posible configurar algo más cercano a consistencia eventual al permitir que las lecturas se completen en servidores secundarios que pueden contener datos desactualizados.

Acorde a Harrison (2015) la coherencia para documentos individuales se logra en MongoDB mediante el uso de bloqueos. Se utilizan cerraduras para asegurar que dos escrituras no intentan modificar un documento simultáneamente, y también que un lector no verá una Vista inconsistente de los datos.

Para Edward y Sabharwal (2015) en MongoDB, aunque las lecturas se pueden enrutar a nodos secundarios, las escrituras siempre se enrutan al nodo primario, erradicando el escenario en el que dos nodos intentan simultáneamente actualizar el mismo conjunto de datos. El conjunto de datos del nodo principal siempre es coherente. Si las solicitudes de lectura se enrutan al nodo principal, siempre verá los cambios actualizados, que significa que las operaciones de lectura son siempre coherentes con las últimas operaciones de escritura, así se logra un bloqueo a nivel granular.

Sin embargo, si la aplicación ha cambiado la preferencia de lectura para leer desde secundarios, puede haber una probabilidad de que el usuario no vea los últimos cambios o los estados anteriores. Esto se debe a que las escrituras son replicadas asincrónicamente en los secundarios.

El bloqueo granular que tiene cuatro niveles, acorde a Giamas (2019) estos niveles son:

- Global (Instancia de MongoDB): Todas las bases de datos en la instancia son afectadas.
- Base de Datos: Sólo la base de datos donde se aplica el bloqueo es afectada.
- Colección: Sólo la colección donde se aplica el bloqueo es afectada.
- Documento: Sólo el documento donde se aplica el bloqueo es afectado

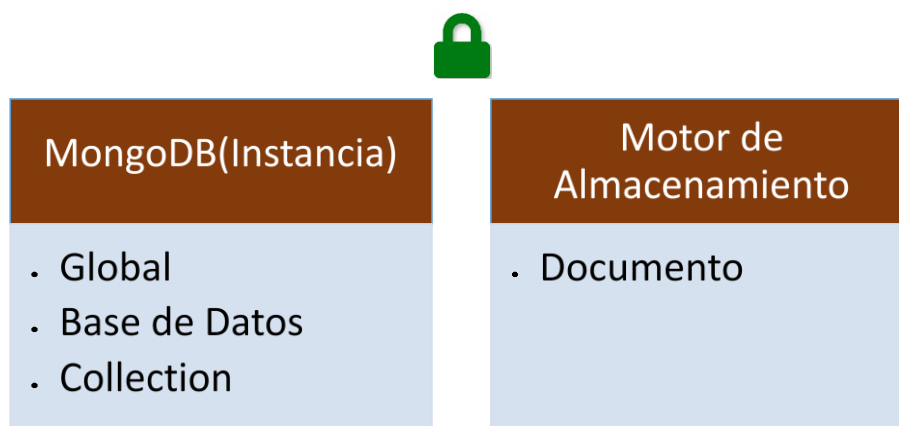


Figura 11. Bloqueos En MongoDB elaboración propia Basado en MongoDB Locks - Shared, Exclusive And Intent Modes (2018).

Además, según *The MongoDB Manual (2020)* existen cuatro modos de bloqueo que son:

- S – *Shared* – Este modo permite que los lectores concurrentes compartan el recurso y se utiliza en operaciones READ.
- X – *Exclusive* – Este modo evita que los lectores concurrentes compartan el recurso y se utiliza en operaciones WRITE.
- IS – *Intent Shared* – Este modo indica que el contenedor del bloqueo podrá leer el recurso a un nivel granular. Si se aplica IS a la base de datos, significa que el contenedor aplicará bloqueo en modo S a nivel de colección o documento.

- *IX – Intent Exclusive* – Este modo indica que el contenedor del bloqueo podrá modificar el recurso a nivel granular. Si se aplica IX a la base de datos, significa que el contenedor aplicará un bloqueo en modo X a nivel de colección o documento.

Estos modos de bloqueo se pueden aplicar sobre otro ya existente, con ciertas restricciones entre cada modo como se muestra en la tabla 3.

tabla 3. Modos de bloqueo MongoDB.

		Modo de bloqueo ya existente en un recurso			
Nuevo modo de bloqueo		S	IS	IX	X
	S		SI	NO	NO
	IS	SI		SI	NO
	IX	NO	SI		NO
	X	NO	NO	NO	

Fuente: elaboración propia, basado en Giamas (2019)

Una forma de comprobar los bloqueos en MongoDB es a través del siguiente comando de la figura 12.

```

db
{
  "lo
  "ns":
  "op": "que
  "opid": 13491
  "query": {"test":
  "secs_running": 15,
  "waitingForLock": true
}

```

Figura 12. Comando para verificar Bloqueos MongoDB.

Como se observa la base de datos se encuentra esperando el bloqueo esto es más frecuente cuando se trabaja con multiusuarios. PostgreSQL basa la concurrencia en transacciones serializables y los bloqueos basados en MMVC, mientras MongoDB basa la concurrencia en bloqueos.

Así como es posibles hallar paralelismos en las seguridades de los dos tipos de DBMS, también es posible encontrar vulnerabilidades en las que los dos DBMS coinciden independientemente de su finalidad.

Vulnerabilidades

OWASP TOP 10 presenta las vulnerabilidades que más afectan a aplicaciones web, pero estas se pueden tomar como marcadores para otras aplicaciones como el primer ítem de control A1:2017 Inyección, luego A6:2017 Configuración de Seguridad Incorrecta, a pesar de no ser vulnerabilidades directamente relacionadas a los DBMS, sino que se presentan con el uso de los mismos, por lo cual es necesario detallar su funcionamiento.

Inyección

De acuerdo a The OWASP Foundation (2017) las fallas de inyección, como SQL, NoSQL, OS o LDAP ocurren cuando se envían datos no confiables a un intérprete, como parte de un comando o consulta. Los datos dañinos del atacante pueden engañar al intérprete para que ejecute comandos involuntarios o acceda a los datos sin la debida autorización.

Para Nwankwo (2020) las bases de datos NoSQL sufren ataques de inyección de consultas al igual que las bases de datos SQL, a pesar de las modificaciones a los protocolos a través de una serie de técnicas de bajo nivel, los riesgos de inyección, la gestión del control de acceso inadecuado y la exposición de la red insegura siguen siendo relativamente altos.

Por lo general la Inyección se presenta a través del cliente por falta de revisión de parámetros o de validación de consultas, como se observa en la figura 13.

Escenario #1: la aplicación utiliza datos no confiables en la construcción del siguiente comando SQL vulnerable:

```
String query = "SELECT * FROM accounts WHERE custID=" +  
request.getParameter("id") + "";
```

Escenario #2: la confianza total de una aplicación en su *framework* puede resultar en consultas que aún son vulnerables a inyección, por ejemplo, *Hibernate Query Language (HQL)*:

```
Query HQLQuery = session.createQuery("FROM accounts WHERE  
custID=" + request.getParameter("id") + "");
```

En ambos casos, al atacante puede modificar el parámetro "id" en su navegador para enviar: ' or '1'=1. Por ejemplo:

```
http://example.com/app/accountView?id=' or '1'=1
```

Esto cambia el significado de ambas consultas, devolviendo todos los registros de la tabla "accounts". Ataques más peligrosos podrían modificar los datos o incluso invocar procedimientos almacenados.

Figura 13. Ejemplos de Inyección (The OWASP Foundation 2017).

Configuración de Seguridad Incorrecta

Para The OWASP Foundation (2017) la configuración de seguridad incorrecta es un problema muy común y se debe en parte a establecer la configuración de forma manual, *ad hoc* o por omisión (o directamente por la falta de configuración). Esta vulnerabilidad está directamente relacionada a los administradores de los DBMS y como se menciona anteriormente se debe principalmente al desconocimiento por parte de los mismos acerca de prestaciones que estas tecnologías ponen a su disposición.

3. Resultados

Una vez detalladas las características de seguridad comunes entre los dos DBMS, se conceptualiza cada uno de los mecanismos por los cuales aseguran estas características, además de ciertas vulnerabilidades comunes, se obtiene como resultado la tabla 4 que presenta los paralelismos de seguridad entre bases de datos relacionales y no relacionales que al igual que el desarrollo del estudio toma como estructura la autenticación, confidencialidad, concurrencia y por último la vulnerabilidades que afectan a los dos DBMS.

TABLA 4. Paralelismos entre Bases de Datos Relacionales y No Relaciones

DBMS Característica	POSTGRESQL	MONGODB
Autenticación	<ul style="list-style-type: none"> • Control de Acceso Obligatorio • Control de Acceso Discrecional 	<ul style="list-style-type: none"> • Autenticación • Autorización * Desactivadas por defecto
Confidencialidad	<ul style="list-style-type: none"> • Encriptación de clave • Encriptación de columnas • Encriptación de partición de datos • Encriptación de datos a través de la red • Encriptación del lado del cliente 	<ul style="list-style-type: none"> • Encriptación de datos en movimiento • Encriptación de datos en reposo • Encriptación del lado del cliente * De forma nativa a nivel de archivos desde la versión 3.2.
Concurrencia	<ul style="list-style-type: none"> • <i>Multi-Version Concurrency Control</i> Transacciones Serializables Bloqueos 	<ul style="list-style-type: none"> • Bloqueos
Vulnerabilidades	Errores de configuración Inyección de consultas	

4. Conclusiones

A pesar de la diferencia de filosofías y de objetivos de los dos sistemas gestores de bases de datos, se halla que los dos DBMS aseguran la autenticación, confidencialidad y concurrencia que incluye integridad y disponibilidad, características propias de la seguridad de sistemas informáticos.

La autenticación se logra a través de dos mecanismos en los dos DBMS y a pesar de que los nombres varían son bastante similares, sin embargo, la autenticación esta deshabilitada por defecto en MongoDB ya que no es parte de sus objetivos. La confidencialidad se asegura por medio de la encriptación y a pesar de estar mucho más desarrollada en el DBMS relacional, el DBMS no relacional la ha puesto especial interés en los últimos años. Por su parte la concurrencia que asegura la disponibilidad e integridad evidencia que parte de MMVC está presente en los dos DBMS, que es el punto en cual los dos tipos de DBMS convergen en el teorema de CAP, así se puede concluir que los paralelismos de seguridad entre bases de datos relacionales y no relacionales existen y ayudan comprender de mejor manera el funcionamiento de cada tipo de DBMS siempre teniendo en cuenta sus filosofías.

Por ultimo las vulnerabilidades que afectan a los dos tipos de DBMS son más de índole administrativa y por parte del cliente, lo que deja en claro que es necesario sin importar el tipo de DBMS leer sus manuales, entender su finalidad e investigar las prestaciones y características de seguridad que cada uno de estos ofrece.

Referencias Bibliográficas

Bernstein, P. A., & Goodman, N. (1983). Multiversion concurrency control—Theory and algorithms.

ACM Transactions on Database Systems, 8(4), 465-483.

<https://doi.org/10.1145/319996.319998>

Codd, E. F. (1970). *A Relational Model of Data for Large Shared Data Banks*. 13(6), 11.

- Coronel, C., Morris, S., & Rob, P. (2010). *Bases de datos [recurso electrónico]: Datos, implementación y administración*. (9.^a ed.). Cengage Learning Editores, S.A. de C.V.
- Date, C. J. (2001). *Introducción a los sistemas de bases de datos*. Pearson Educación.
- Edward, S. G., & Sabharwal, N. (2015). *Practical MongoDB*. Apress. <https://doi.org/10.1007/978-1-4842-0647-8>
- Giamas, A. (2019). *Mastering MongoDB 4.x: Expert techniques to run high-volume and fault-tolerant database solutions using MongoDB 4.x, 2nd Edition*. Packt Publishing Ltd.
- Gilbert, S., & Lynch, N. (2012). Perspectives on the CAP Theorem. *Computer*, 45(2), 30-36.
<https://doi.org/10.1109/MC.2011.389>
- Harrison, G. (2015). *Next Generation Databases*. Apress. <https://doi.org/10.1007/978-1-4842-1329-2>
- Lockhart, T. (Ed.). (1996). *Manual del usuario de PostgreSQL*.
- MongoDB Locks—Shared, Exclusive and Intent Modes. (2018). *TutorialKart*.
<https://www.tutorialkart.com/mongodb/mongodb-locks/>
- Nwankwo, W. (2020). A Review of Critical Security Challenges in SQL-based and NoSQL Systems from 2010 to 2019. *International Journal of Advanced Trends in Computer Science and Engineering*, 9(2), 2029-2035. <https://doi.org/10.30534/ijatcse/2020/174922020>
- Perkins, L., Redmond, E., & Wilson, J. R. (2018). *Seven Databases in Seven Weeks*. Pragmatic Programmers.
- Ricardo, C. M. (2009). *Bases de datos*. The McGraw-Hil.
- rubenfa. (2014, enero 28). *NoSQL: Clasificación de las bases de datos según el teorema CAP*. Genbeta. <https://www.genbeta.com/desarrollo/nosql-clasificacion-de-las-bases-de-datos-segun-el-teorema-cap>
- Sánchez Asenjo, J. (2009). *Sistemas Gestores de Bases de Datos (Autoedición)*.
- The MongoDB 4.4 Manual—MongoDB Manual*. (s. f.).
<https://github.com/Mongodb/Docs/blob/master/Source/Index.Txt>. Recuperado 11 de diciembre de 2020, de <https://docs.mongodb.com/manual/>
- The OWASP Foundation. (2017). *OWASP Top 10—2017 Los diez riesgos más críticos en Aplicaciones Web*. The OWASP Foundation. <https://wiki.owasp.org/images/5/5e/OWASP-Top-10-2017-es.pdf>

Tonon, G. (2011). *La Utilización Del Método Comparativo En Estudios Cualitativos En Ciencia Política Y Ciencias Sociales: Diseño y desarrollo de una tesis doctoral.*
<https://dialnet.unirioja.es/descarga/articulo/3702607.pdf>



Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-CompartirIgual 2.5 México.

Reconocimiento de palabras de la Lengua de Señas Mexicana utilizando información RGB-D

Mexican Sign Language word recognition using RGB-D information

Felipe Trujillo-Romero¹
fdj.trujillo@ugto.mx

Gibrán García-Bautista²
ybrran@gmail.com

¹División de Ingenierías, Campus Irapuato-Salamanca, Universidad de Guanajuato, Guanajuato, México.

²Intabi Company, Orizaba, Veracruz, México.

Resumen. La Lengua de Señas es el principal método alternativo de comunicación entre personas con discapacidad en el habla o en la escucha. Sin embargo, la mayoría de la población que no padece esta discapacidad no la comprende. Esto hace que la comunicación de las personas signantes con su entorno social sea casi imposible. En este trabajo se presenta un avance hacia la construcción de un sistema que pueda traducir palabras de la Lengua de Señas Mexicana mediante el reconocimiento de estas a partir de la trayectoria 3D del movimiento de las manos de signantes usando un sensor Kinect. Se construyó un corpus de 53 palabras considerando solo palabras pertenecientes a once campos semánticos. Con el objetivo de eliminar posibles inconsistencias y ruidos en el patrón extraído se usó la adición de puntos intermedios y el algoritmo KNN fue usado para el filtrado. Además, el método descriptor empleado divide el patrón en dos secciones de acuerdo con la cúspide de su trayectoria y mediante la media aritmética se obtienen las posiciones 3D representativas de ambas secciones. Del patrón general, se obtienen también su anchura, altura, profundidad y orientación. Para la clasificación de las palabras del corpus se usa una Red Neuronal Artificial de tipo Perceptrón Multi Capa. Esta red fue entrenada con el algoritmo de *Backpropagation* y para la validación del sistema reconocedor se realizó utilizando el método *K-Fold Cross Validation*. El porcentaje de precisión media alcanzado por esta implementación fue del 93.46%.

Palabras clave: reconocimiento de patrones, corpus de la lengua de señas, redes neuronales, Lengua de Señas Mexicana.

Abstract. Sign Language is the primary alternative method of communication between people with hearing or speech impairment. However, most of the population that does not suffer from this disability cannot understand or interact with them. Consequently, communication of the signatories with their social environment becomes almost impossible. This paper presents progress towards constructing a system to translate words from the Mexican Sign Language into text, by the signatory's hands motion recognition from a 3D trajectory using a Kinect sensor. A corpus of 53 words was built and considered only words belonging to eleven semantic fields. Intermediate points were added, the KNN algorithm was used for filtering to eliminate possible inconsistencies and noise in the extracted pattern. In addition, the descriptor method used divides the pattern into two sections according to the cusp of its trajectory, and the representative 3D positions of both sections are obtained utilizing the arithmetic mean. From the general pattern, its width, height, depth, and orientation are also obtained. For the classification of the words in the corpus, an Artificial Neural Network of the Multi-Layer Perceptron type is used. This network was trained with the Backpropagation algorithm, and for the validation of the recognizing system, it was performed using the K-Fold Cross Validation method. The percentage of mean precision achieved by this implementation was 93.46%.

Keywords: pattern recognition, sign language corpus, neural networks, Mexican Sign Language.

1. Introducción

De acuerdo con el Censo de Población y Vivienda (2020) realizado por el Instituto Nacional de Estadística y Geografía (INEGI), en México hay 6,179,890 personas con algún tipo de discapacidad, lo que representa 4.9 % de la población total del país. De estos el 22% sufre alguna discapacidad auditiva mientras que el 15% tiene algún problema del habla. Por lo que se estima que una población de alrededor un millón de personas tendría alguna discapacidad auditiva o del habla. Cabe mencionar que una persona puede tener una de las discapacidades o ambas.

La Lengua de Señas (LS) es la principal forma de comunicación de las personas que padecen alguna discapacidad auditiva o en el habla. Sin embargo, esta lengua de señas es desconocida por la mayoría de la población que no padece de alguna de estas discapacidades, incluso por muchos que poseen este tipo de discapacidad. Además de que no existe una LS universal y en un mismo país una misma LS puede variar de una región a otra. Por esa razón diversos investigadores de diversas áreas han tratado de dar solución a esta problemática.

Por ejemplo, en el área computacional el Reconocimiento de la Lengua de Señas (RLS) se ha convertido en un importante estudio de interés que requiere la Interacción Humano Computador (IHC). Sin embargo, lograr obtener un sistema de RLS es una tarea muy compleja debido a que implica tres aspectos fundamentales, que son: 1) el reconocimiento de la mano, 2) de los dedos y 3) de las expresiones faciales. Además, los sistemas de RLS se desarrollan en diversos campos tal como la visión computacional y el reconocimiento de patrones. Podemos decir que existen dos acercamientos principales para tratar con la implementación de un RLS: 1) basado en hardware tal como los guantes de datos y 2) basado en visión computacional. El primero de estos requieren que el signante porte un accesorio como un guante o una pulsera que permita transmitir los movimientos hacia el sistema de procesamiento que los interpretará y dará la salida correspondiente. En el caso de los sistemas implementados usando visión por computadora, se conocen como métodos no invasivos ya que no requieren que el signante utilice algún hardware en especial. Sino que se usa una cámara para capturar la información que será procesada por el sistema interprete. Dicha cámara normalmente en el sistema de procesamiento.

Por lo tanto, el reconocimiento basado en visión computacional es un enfoque más práctico y económico comparado con el basado en hardware, pero también más desafiante. Los principales desafíos son, principalmente, debidos al ruido que causan las condiciones de iluminación del entorno y sobre todo variabilidad en la representación de los símbolos de la LSM por los signantes. Esta es la razón por lo que los dispositivos de captura a utilizar representan un factor importante para alcanzar buena precisión en la implementación de un sistema de RLS.

En los últimos años, el desarrollo de sistemas de RLS basado en sensores RGB-D se ha generalizado. Lo anterior debido a que la información 3D proporcionada por los sensores de profundidad mejora la interactividad, así como la comodidad para el usuario. También se debe tener en cuenta que los precios son cada vez más asequibles para este tipo de dispositivos, como el sensor Kinect (2021) de Microsoft.

El procesamiento de datos tridimensional tiene varias ventajas sobre la información 2D que capturan las cámaras comunes. De inicio el uso de la información 3D permite estimar con mayor exactitud la posición y la orientación del objeto con respecto al sensor de captura. Otro punto importante es que el reconocimiento de objetos 3D tiende a ser más robusto en escenas con ruido. Por ejemplo, donde los objetos que se encuentran al frente obstruyen a los que se encuentran en el fondo. Cabe resaltar el hecho de que el sensor Kinect ha sido usado por una gran cantidad de investigadores en sus trabajos de RLS. Esto debido en gran medida a que el sensor Kinect cuenta con cámara de color, sensor de profundidad, un arreglo de micrófonos y es capaz de ajustarse verticalmente para una mejor captura de la información. Además, posee algoritmos integrados los cuales son capaces de calcular y proveer la posición de hasta 20 articulaciones del cuerpo humano. Para poder acceder a los algoritmos antes mencionados es necesario utilizar el Microsoft Kinect Software Development Kit (SDK, 2021) que Microsoft ofrece para programar su dispositivo. Se debe mencionar que este sensor es relativamente accesible para cualquier investigador e incluso estudiantes que deseen trabajar con este tipo de dispositivos.

Los trabajos realizados para el RLS se pueden clasificar básicamente en 3 enfoques: reconocimiento del alfabeto y números, reconocimiento de palabras y reconocimiento de oraciones. Ejemplos de trabajos donde se realiza el reconocimiento alfanumérico del LS se tiene el desarrollado por Jiménez et al. (2017), quienes usaron características de tipo Harr 3D aplicado en la LSM. Por su parte Pérez et al. (2017) emplearon Momentos de Hu y Lógica difusa para generar un sistema que fuera capaz de reconocer los símbolos estáticos de la LSM. Otro trabajo sobre el reconocimiento de símbolos estáticos lo realizó Carmona-Arrollo et al. (2021), solo que ellos usaron las características invariantes de momentos afines en 3D, como la pose, la posición y la forma entre las manos de los signantes para realizar el reconocimiento. Lahamy y Lichti (2012) también usaron características invariantes, pero en este trabajo fue en dos dimensiones. En el artículo desarrollado por Agarwal y Thakur encontramos un ejemplo en el cual solo se utilizan los símbolos en LS correspondientes a los dígitos. Finalmente, en este rubro, se comentará el trabajo realizado por Luis-Pérez et al., (2011), en el cual se reconocieron los 27 símbolos del alfabeto de LSM pero en este caso se realizó el control de un robot móvil orientado hacia la robótica de servicio.

En el área del reconocimiento de palabras comencemos citando el trabajo de Estrivero-Chavez et al. (2019), quienes usando el sensor Leap Motion (2021) lograron reconocer, además del alfabeto, una decena de palabras en LSM. Por su parte Garcí-Bautista et al. (2017) desarrollaron un sistema que usando un Kinect y el algoritmo de Dynamic Time Warping lograron reconocer 20 palabras de LSM. Tazhigaliyeva et al. (2017) aumentaron a 33 los símbolos que un sistema automático era capaz de reconocer, solo que esta vez fue para el alfabeto Cirílico.

Respecto al reconocimiento de estructuras semánticas más complejas como frases y desde luego sistemas que puedan traducir del LSM a texto o a voz se pueden comentar los trabajos siguientes. Uno de estos desarrollos lo encontramos en Hazari et al. (2017) en cuyo trabajo diseñaron un sistema basa en un kinect para traducción del lenguaje de señas Americano (ASL). Por su parte Chai et al. Implementaron un sistema para la traducción del Lenguaje de Señas Chino (CSL) también usando como dispositivo de captura un Kinect. Ghotkar y Kharate (2015) realizaron un sistema que reconocía el lenguaje de señas Indio (ISL) además de interpretar frases en dicho lenguaje. En el caso de la LSM mencionamos los trabajos de García-Bautista et al. (2016) y Sosa-Jiménez et al. (2017) quienes, cada quien por su parte, realizaron implementaciones orientadas al reconocimiento de palabras y frases simples en LSM orientado hacia el desarrollo de un traductor de LSM a voz o texto.

Como se puede imaginar por los trabajos antes mencionados, el reconocimiento de palabras de la LS implica la combinación de gestos manuales y no manuales involucrando movimiento. Para ello es necesario contar con los métodos que nos permitan realizar dicho reconocimiento. Por esa razón en la literatura podemos encontrar diversos métodos de reconocimiento de patrones que son utilizados para el RLS, como es el caso de las redes neuronales artificiales. Por ejemplo, Molchanov et al. (2015), implementaron un Sistema de reconocimiento de gestos de la mano usando redes neuronales convolucionales. Otro trabajo es el de Oyewole et al. (2018), quienes realizaron un sistema para ayudar a la comunicación entre personas oyentes y aquellos con alguna discapacidad auditiva. Un robot tutor de reconocimiento de señas orientado a la asistencia de niños con discapacidad auditiva o del habla fue desarrollado por Gürpınar et al. (2020). Jiang et al. (2021) haciendo uso de redes neuronales reconocen el esqueleto y hacen el reconocimiento del ASL. Por otro lado Zhang et al. (2019) mediante el uso de redes neuronales desarrollan un sistema en tiempo real para el reconocimiento de la lengua de señas China. Finalmente se comenta el trabajo de Fregos et al. (2021) en cuyo trabajo se usa una red neuronal convolucional y optimización de sistemas de partículas (PSO) tanto para el reconocimiento de la lengua de señas Mexicana como Americana.

Debido a lo comentado en los párrafos anteriores es la motivación que nos mueve para que en este trabajo se presente el reconocimiento de 53 palabras de la LSM a partir del seguimiento de la trayectoria tridimensional de las manos captadas por el sensor Kinect y una red neuronal artificial de tipo Perceptrón Multi Capa y *Backpropagation* como algoritmo de aprendizaje.

Las contribuciones de este trabajo son la que se enumeran a continuación:

1. Creación de un mini-Corpus de la LSM
2. Desarrollo de un sistema que reconoce palabras en LSM.
3. Plantear las bases para la construcción tanto de un corpus de LSM más completo como para un sistema de intérprete y traducción de LSM a voz o a texto.

La estructura del artículo es la siguiente. En la sección 2 se describe a grandes rasgos, lo referente a la lengua de señas mexicana, las características del sensor Kinect, la red neuronal empleada, así como la base de datos usada en este trabajo. Posteriormente, la implementación del sistema de reconocimiento del mini corpus de LSM se presenta en la sección 3. Los resultados obtenidos al realizar las pruebas de *cross-reference* usando 5-folds se muestran en la sección 4. Finalmente, la sección 5 contiene las conclusiones de este trabajo, así como la posible continuación del mismo.

2. Métodos y materiales

En esta sección se abordará la descripción tanto de los algoritmos como de las diferentes herramientas empleadas en la realización de este trabajo. Se iniciará con una descripción referente a la Lengua de Señas Mexicana para situar el contexto. Para continuar con las características del sensor usado para capturar y crear la base de datos usada. Posteriormente, se hablará de la red neuronal que se empleo para realizar el sistema de reconocimiento para finalizar con la mención del Corpus de LSM se creo para realizar el presente trabajo.

2.1 Lengua de Señas Mexicana

La Lengua de Señas Mexicana, se deriva del antiguo lenguaje de signos francés traído a México en 1869. Para ese tiempo, en México ya existían personas sordas que usaban señas para comunicarse, estas señas se fueron incorporando al nuevo lenguaje y se complementaron ampliamente con el sistema francés (Calvo, 2004). Como la mayoría de las lenguas de señas la LSM está compuesta de la dactilología e ideogramas (Serafín y González, 2011).

El alfabeto de la LSM es el que se muestra en la Figura 1 y está formado tanto por señas estáticas como por señas dinámicas. Las señas dinámicas son aquellas cuya trayectoria de movimiento es representada por una flecha color rojo (ver Figura 1). En la ejecución del alfabeto de la LSM se hace uso de una mano base y una mano dominante. Normalmente la mano dominante es la izquierda para personas zurdas y la mano derecha para personas diestras.

La LSM, corresponde al ISO 639-2 sgn-MX (Código estándar Mex-Esp, 2021). El ISO-639-2, es el estándar internacional para los códigos de idioma, cuyo objetivo es establecer los códigos reconocidos internacionalmente para la representación de lenguajes o familias de lenguajes.

Hay que tener en cuenta que como lengua de comunicación la LSM es un lenguaje completo y distinto. Es distinto de otros lenguajes de señas tales como el Lenguaje de Señas Americano (ASL) y distinto del idioma Español (la lengua oral oficial en México).

Se debe de considerar que también existen diferencias en las diferentes lenguas de señas, aunque sea el mismo idioma oral de referencia. Es decir que la lengua de señas no es la misma en todos los países donde se hable español. Por lo tanto, existirá una lengua de señas para México, otra para España, Colombia, Perú, Chile, y así una lengua de señas diferente para país de habla hispana.

También se debe de observar que una cantidad significativa de personas sordas son mayormente monolingües en la LSM. Esto significa que ni el ASL y ni el español son adecuados para una completa comunicación entre la comunidad de sordos de México en ninguna forma, sea por video, por escrito, o por contacto personal.

Otra cosa importante es que si bien es cierto que hay diferencias significativas de la lengua de señas en los países donde se habla español, también hay variaciones dentro de la misma lengua de señas. Algunas variaciones son debidas a expresiones regionales y otras que se deben a las distintas clases de religión, a la edad e inclusive al nivel de educación.

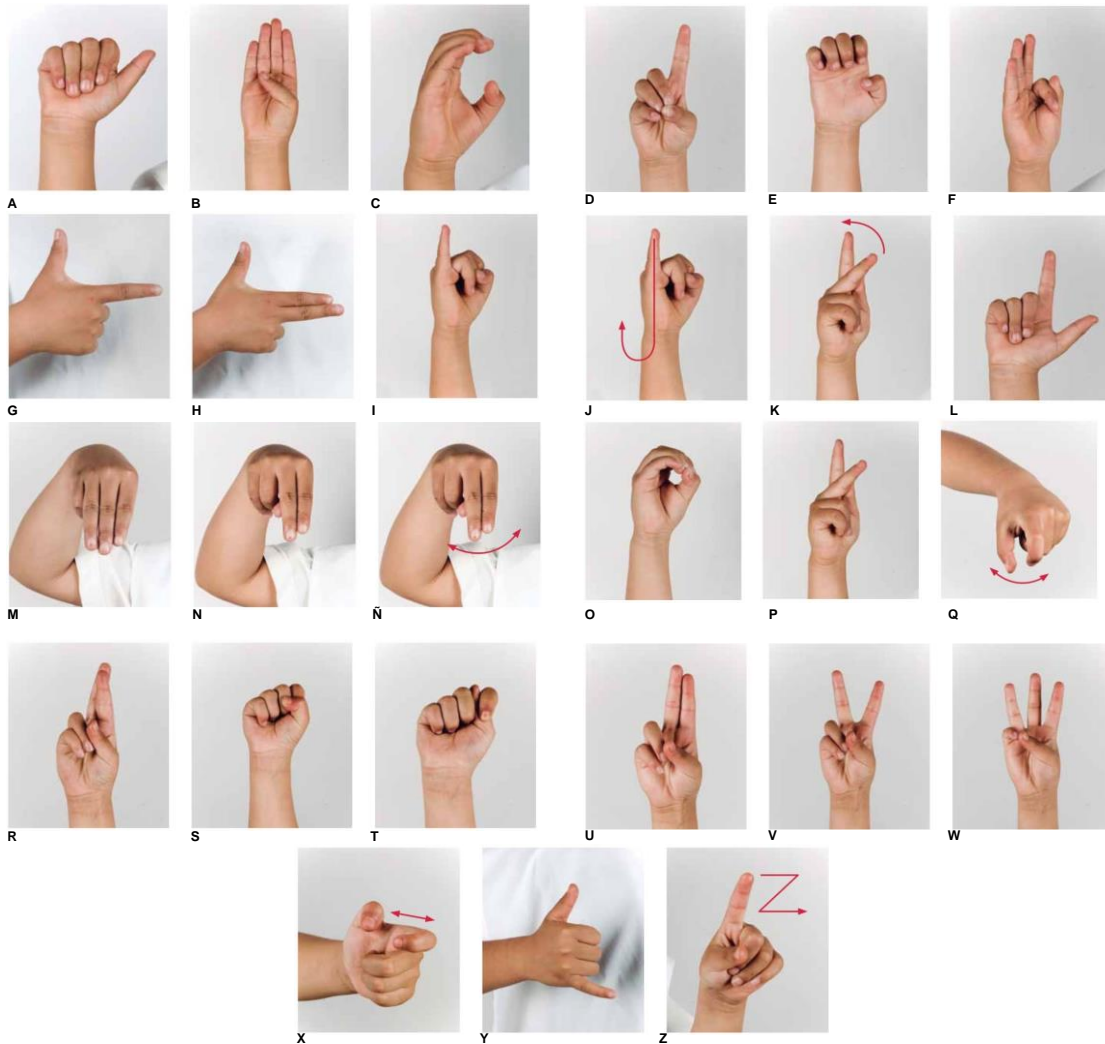


Figura 1 Alfabeto de la LSM (Serafín y González, 2011).

Estas variaciones necesitan ser examinadas más detalladamente lo cual está fuera del alcance de este trabajo. Sin embargo, la mayor parte de la evidencia apunta a un alto grado de similitud, no muy diferente de las variaciones regionales encontradas en las lenguas habladas como el Inglés Americano o el Español Mexicano (Serafín y González, 2011).

2.2. Sensor RGB-D

Un sensor RGB-D, como su nombre lo dice nos permite capturar la información fotométrica en el esquema RGB. Además de poder tener los datos correspondientes a la profundidad con la que se encuentran los objetos en una escena. El sensor Kinect es un dispositivo que pertenece a este tipo de sensores RGB-D. El sensor Kinect fue creado por Microsoft principalmente para ser usado para la consola de video juegos Xbox. Sin embargo, la comunidad científica encontró en él una herramienta de bajo coste para ser usada en diversos proyectos y aplicaciones (Zhang, 2012).

Este sensor al estar compuesto por cámaras y sensores permiten el reconocimiento de movimientos corporales y gestos para facilitar la interacción humano computadora a través de la interfaz natural de usuario (Natural User Interface, NUI) provista por el fabricante.

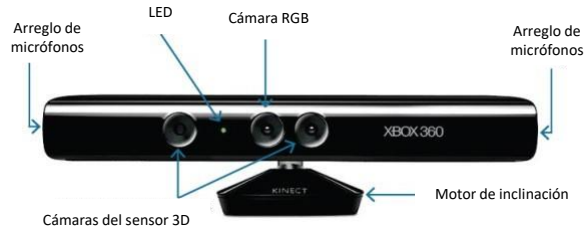


Figura 2 Componentes del dispositivo Kinect™ para Xbox.

Un dispositivo Kinect está integrado por los siguientes componentes: sensor de profundidad, cámara RGB, array de micrófonos y sensor de infrarrojos, que le permiten capturar el esqueleto humano, reconocerlo y posicionarlo en el plano, (Ver figura 2). Actualmente el Kinect cuenta con dos versiones, pero para este trabajo se utilizará la primera versión (V1). En la tabla 1 se pueden observar las características de la versión 1 del sensor Kinect.

Tabla 1 Características del Kinect (Dal Mutto et al., 2012).

Funciones	Kinect V1
Cámara de color	640 x 480 @30fps
Cámara de profundidad	320 x 240
Distancia máxima de profundidad	Aproximado de 4.5 mts
Distancia mínima de profundidad	40 cm en modo cercano
Campo de visión horizontal	57 grados
Campo de visión vertical	43 grados
Motor de inclinación	SI
Articulaciones definidas del esqueleto	20 articulaciones
Seguimiento completo del Esqueleto	2
Estándar USB	2.0
S.O soportados	Windows 7, 8 y 10

La captura del dispositivo Kinect inicia cuando las cámaras tanto la RGB como la de infrarrojos captan la información presente en la escena, por ejemplo, a una persona que se encuentra enfrente del dispositivo como en la Figura 3.a. En esta Figura 3.a se puede observar la información RGB del individuo presente. Al mismo tiempo y por medio de un láser se mide la distancia a la cual se encuentra el sujeto. Esto mediante el láser, el cual vuelve al lugar de origen de manera continua como el sonar de un submarino. Como se comentó, el láser solo capta la profundidad a la que se encuentra el sujeto, mientras que la cámara de infrarrojos capta los puntos que se proyectan sobre él. Lo anterior se puede observar en la figura 3.b. A partir de los datos obtenidos por el sensor infrarrojo es posible generar nubes de puntos o mapas de voxels.

Estos datos constituyen una imagen 3D de la zona en una escala de grises. La combinación de ambas cámaras hace que el software obtenga una escena donde excluye todo lo que no sea el cuerpo, y obtiene la imagen o mapa de profundidad como el que se puede apreciar en la figura 3.c. Este mapa de profundidad es útil ya que es el elemento indispensable para la orientación y la captación de movimiento del cuerpo humano. Y es partir de este que se puede extraer el esqueleto de la persona presente en la escena como se ve en la Figura 3.d.

Este mecanismo tiene unos resultados excelentes en todo tipo de condiciones de iluminación, incluida la oscuridad total y con la misma precisión en usuarios de tez oscura. Para la identificación, el Kinect viene embebido con un sistema de reconocimiento de bípedos llamado *BodyIndex*. Este sistema evalúa varios factores como el movimiento de brazos, el rango de altura, posición de la cabeza, etc. Una vez identificado el cuerpo, el programa crea un objeto virtual independiente de la imagen.



Figura 3. Imágenes capturadas por los sensores del Kinect™.

Para poder acceder a los recursos del sensor Kinect, Microsoft proporciona el kit de desarrollo de software para Windows. Este SDK es un conjunto de controladores, de interfaces de programación de aplicaciones (APIs), interfaces de dispositivo, documentos de instalación y manuales de uso que permite la creación de aplicaciones comerciales y no comerciales para Windows (Kinect SDK, 2021).

2.3. Perceptrón multicapa

Las redes neuronales artificiales (RNA), al margen de “parecerse” a las neuronas biológicas que se encuentran en el cerebro humano, presentan una serie de características propias del mismo. Como, por ejemplo, las RNA son capaces de aprender de la experiencia, generalizan de ejemplos previos a ejemplos nuevos y abstraen las características principales de una serie de datos.

Lo que se considera la primera red neuronal y que se le llamó “Perceptrón” fue desarrollado por Rosenblatt (1957). Este primer perceptrón era capaz de aprender algo y también era robusto, ya que su comportamiento solo se veía afectado si resultaban dañados los componentes que lo conformaban. Este perceptrón es solo una neurona, por lo que su capacidad para aprender y resolver problemas está limitada. Por esa razón se comenzó a desarrollar otros tipos de arquitecturas de redes, pero teniendo como base al perceptrón.

El primer desarrollo para entrenamiento de una red neuronal multicapa fue descrito por Werbos (1974), donde se presentaba un algoritmo de entrenamiento para redes neuronales de contexto general. Sin embargo, fue hasta mediados de los años 80’s, cuando Rumelhart et al. (1986) retomaron la investigación y comenzó a popularizarla en diferentes publicaciones, hasta lo que hoy se conoce como algoritmo *backpropagation*.

El objetivo de entrenar de una red neuronal es conseguir que una aplicación determinada, para un conjunto de entradas, produzca un conjunto de salidas deseadas. El proceso de entrenamiento consiste del ajuste en los pesos de las interconexiones según un procedimiento predeterminado. En cada iteración, los pesos convergen gradualmente hacia los valores que hacen que cada entrada produzca la salida deseada. Los algoritmos de entrenamiento se pueden clasificar en dos grupos: supervisados y no supervisados. Los primeros consisten en presentar un vector de entrada a la red, calcular la salida de la misma, compararla con la salida deseada, y el error o diferencia resultante se utiliza para realimentar la red y cambiar los pesos de acuerdo con un algoritmo que tiende a minimizar el error. Los segundos modifican los pesos de la red de forma que produzcan vectores de salida consistentes, extraen las propiedades estadísticas del conjunto de vectores de entrenamiento y agrupa en clases los vectores similares.

El método de entrenamiento *backpropagation* (entrenamiento hacia atrás), es un sistema de entrenamiento con capas ocultas perfeccionado en la década de los 80's (Rumelhart et al., 1986). A grandes rasgos, el sistema de entrenamiento mediante *backpropagation* consiste en:

1. Inicializar los pesos de la red de manera aleatoria.
2. Introducir datos de entrada de entre los que se van a usar para el entrenamiento.
3. Dejar que la red genere un vector de datos de salida (propagación hacia adelante).
4. Comparar la salida de la red, con la salida deseada.
5. La diferencia entre la salida generada y la deseada (denominada error) se usa para ajustar los pesos en la capa de salida.
6. El error se propaga hacia atrás (*backpropagation*), hacia la capa de neuronas anterior, y se usa para ajustar los pesos de esa capa.
7. Se continúa propagando el error hacia atrás y ajustando los pesos hasta que se alcance la capa de entrada.

2.3 Corpus de la Lengua de Señas Mexicana

Un corpus es una base de datos que contiene la información necesaria para poder desarrollar herramientas o aplicaciones de reconocimiento de la lengua de señas. El corpus es, entonces, el conjunto de muestras de diferentes palabras en lengua de señas que se procesará y servirá de entrada para el sistema de reconocimiento. Actualmente existen muchos corpus de la LS tal como el Corpus de la Lengua de Señas Británica (BSL) (BSL Corpus, 2021), el Corpus de la Lengua de Señas Alemana (GSL) (GSL Corpus, 2021), el Corpus de la Lengua de Señas Española (LSE Corpus, 2021), el Corpus de la Lengua de Señas Peruana (LSP Corpus, 2021) entre otros. En México existen algunos diccionarios representativos del LSM y estos son el DIELSEME (Calvo, 2004) y Manos con Voz (Serafín y González, 2011). Sin embargo, a pesar de contar con una base rica en información, los diccionarios existentes en México para la LSM no constituyen un corpus y carecen de suficiente información para trabajar en el reconocimiento de la LSM. Cabe mencionar que hasta el presente día en México no existe de manera formal y oficial un Corpus de la Lengua de Señas Mexicana (CLSM).

Por esta razón se procedió a generar la construcción de un pequeño CLSM con el objetivo de tener los datos necesarios para poder lograr un sistema de reconocimiento de un grupo de palabras en LSM específicas.

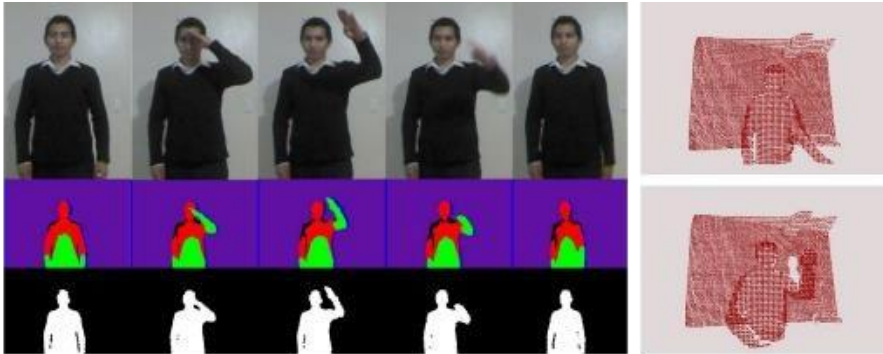


Figura 4 Muestras de color, profundidad y nubes de puntos contenidas en el CLSM.

Para iniciar con la obtención de este corpus, se recolectaron un total de 1590 muestras de 53 palabras interpretadas por 30 personas que utilizan la LSM para comunicarse. Del total de participantes 16 fueron mujeres y 14 hombres, con rango de edad entre 12 y 51 años. En la figura 4, se muestra un ejemplo de la palabra HOLA en LSM que está contenida en el corpus. Los datos que se almacena en este corpus a partir de las palabras de la LSM capturadas contienen la información de profundidad, color, las nubes de puntos, así como las posiciones de 14 articulaciones del esqueleto humano que nos proporciona el sensor Kinect.

En la Tabla 2, se puede observar los 11 campos semánticos seleccionados, así como las palabras en cada uno de ellos, considerando la variabilidad en el movimiento de ambas manos. Además, en la Tabla 2, se presenta el número de muestras de cada seña, las manos que se utilizan para realizar cada seña y como dato adicional una abreviación que se utilizará para referirnos a una seña en particular.

Tabla 2. Palabras que conforman el CLSM utilizado.

Campo Semántico	Palabra	Abreviación	Manos implicadas	No. Muestras
Saludos	Hola	HLA	DER	30
	Adiós	ADS	DER	30
	Gracias	GRA	DER	30
	Nos vemos	NVE	DER	30
	¡Buenos días!	BNS	IZQ	30
Familia	Mamá	MAM	DER	30
	Papá	PAP	DER	30
	Hijo	HJO	DER	30
	Hija	HJA	DER	30
	Familia	FAM	IZQ, DER	30
Pronombres	Yo	YO	DER	30
	Tú	TU	DER	30
	Él/Ella	EL	DER	30
	Nosotros	NOS	DER	30
	Ellos	ELL	DER	30
Lugares	Calle	CAL	IZQ, DER	30
	Edificio	EDI	IZQ, DER	30
	Casa	CAS	IZQ, DER	30
	Baño	BAN	DER	30
	Escuela	ESC	DER	30
Alimentos	Agua	AGU	DER	30
	Fruta	FRU	IZQ, DER	30
	Huevo	HUE	IZQ, DER	30
	Café	CAF	IZQ, DER	30
	Galleta	GAL	DER	30
Preguntas	¿Cómo estás?	CME	IZQ, DER	30
	¿Qué haces?	QHA	DER	30
	¿Cuál?	CUA	IZQ, DER	30
Cocina	Comida	COM	IZQ, DER	30
	Cuchara	CUC	IZQ, DER	30
	Mesa	MES	IZQ, DER	30
	Plato	PLA	IZQ, DER	30
	Silla	SIL	IZQ, DER	30
Colores	Blanco	BLA	IZQ, DER	30
	Negro	NEG	IZQ, DER	30
	Amarillo	AMA	IZQ, DER	30
	Azul	AZU	IZQ, DER	30
	Rojo	ROJ	DER	30
Estados de Ánimo	Aburrido	ABU	IZQ, DER	30
	Enojado	ENO	DER	30
	Feliz	FEL	DER	30
	Miedo	MIE	DER	30
	Triste	TRI	DER	30
Medios de Transporte	Avión	AVI	IZQ, DER	30
	Automóvil	AUT	IZQ, DER	30
	Taxi	TAX	IZQ, DER	30
	Autobús	AUB	IZQ, DER	30
	Motocicleta	MOT	IZQ, DER	30
Misceláneos	Bien	BIE	DER	30
	Mal	MAL	DER	30
	Trabajo	TRA	IZQ, DER	30
	Problemas	PRO	IZQ, DER	30
	Abrazos	ABR	IZQ, DER	30

3. Implementación del sistema de reconocimiento

El proceso realizado en este trabajo se muestra en la Figura 5, y comienza con la captura de los datos que conformarán el corpus de la LSM. Posteriormente, el patrón de la trayectoria de la mano es reconstruido y pre-procesado. A partir del nuevo patrón se genera el vector de características que se introduce a la red neuronal. Finalmente el método *K-Fold Cross Validation* es usado para el entrenamiento y la validación de la red neuronal.

Para poder realizar el sistema de reconocimiento usando el corpus de LSM generado se requiere de procesar previamente los datos de este para extraer las características que nos interesan y que van a servir para el entrenamiento del sistema y la subsecuente evaluación del reconocedor de palabras de LSM. De esta manera entonces vamos a revisar la implementación del sistema de reconocimiento de LSM empezando por el acondicionamiento de los datos del corpus.

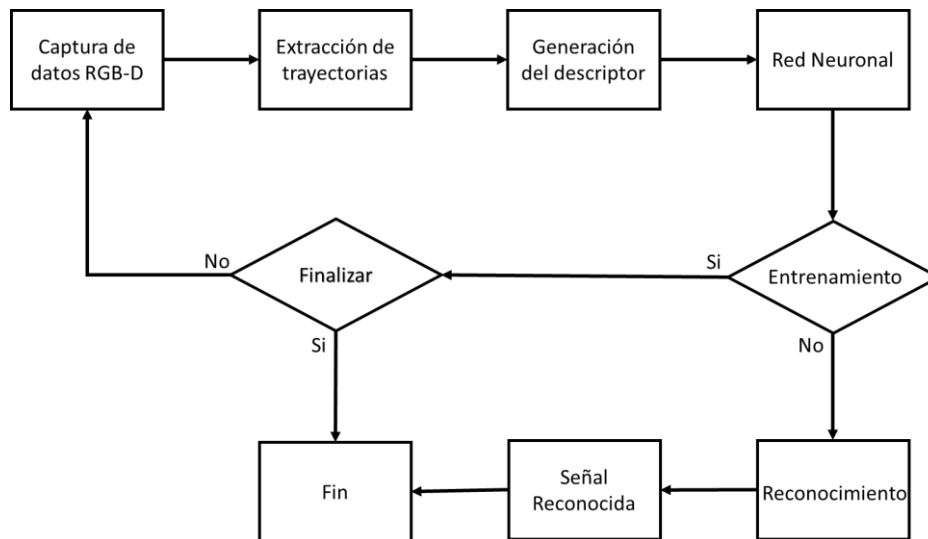


Figura 5 Proceso del sistema reconocedor de palabras de la LSM.

3.1 Acondicionamiento del corpus de LSM

En este paso se hará uso de un filtro pasa-banda con la finalidad de eliminar toda información que no pertenezca al signante. Además, la aplicación de este filtro pasa-banda nos permitirá realizar una reducción del tamaño de los datos lo cual hará que se procesen más rápido.

El funcionamiento de este filtro consiste en remover todos los puntos que se encuentren fuera de un rango dado para una dimensión específica. Considerando que todas las muestras fueron capturadas en un entorno controlado con respecto a la distancia entre el signante y el Kinect de alrededor de 1.7 metros con la posición del usuario se encuentra centrada respecto al sensor. Además, el sensor está colocado a una altura de 1.3 metros. Por tal razón se consideró factible aplicar este filtro en base a un umbral de 1.9 metros respecto al eje central del sensor.

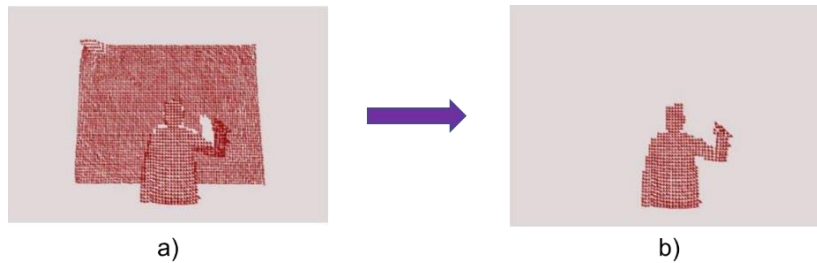


Figura 6. a) Datos de entrada, b) datos filtrados

La figura 6a) muestra la nube de puntos que se obtiene directamente del sensor Kinect mediante el uso del SDK que proporciona Microsoft. Mientras que en la Figura 6b) se puede observar la nube de puntos ya filtrada.

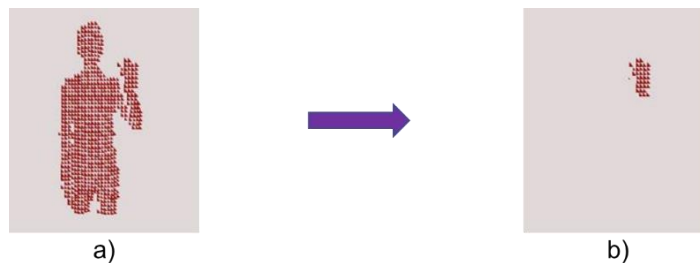


Figura 7. Segmentación de la mano

Continuando con el procesamiento de la información, lo que nos interesa es localizar las manos del signante para poder seguir su trayectoria. Por lo tanto, el movimiento de la mano durante la ejecución de una palabra de la LSM involucra una trayectoria en un plano tridimensional. Así que es necesario realizar una segunda etapa de filtrado, pero esta vez para reconocer y segmentar la mano del signante con la finalidad de seguir y trazar solo la trayectoria de esta parte del cuerpo. La figura 7a) muestra la nube de puntos después de quitar la información que no pertenece al signante. A partir de esta información se segmenta la mano para poder seguir su trayectoria. La figura 7b) se puede observar la mano segmentada de la nube de puntos de la figura 6a).

Una vez ubicada y segmentada la mano haremos el seguimiento con ayuda del sensor Kinect, el cual nos permite almacenar esta trayectoria como un conjunto ordenado de posiciones 3D también llamados puntos (ver Figura 8a). Tanto el número de puntos como la precisión en el cálculo de las posiciones dadas por el Kinect SDK están directamente ligados a la velocidad con la que un gesto es ejecutado frente al sensor y a la respuesta de este a los cambios en la entrada. Cabe mencionar que un error en la precisión del cálculo de una posición puede generar un punto fuera de contexto y ocasionar ruido.

3.2 Reconstrucción del nuevo patrón

Por lo tanto, dada la variabilidad en el número de puntos de una muestra a otra, la posición de captura respecto al sensor y el posible ruido existente es necesario generalizar y simplificar el patrón de la trayectoria de la mano. Para ello se realizó una etapa de reconstrucción del patrón de puntos, en el cual se incluyeron las fases de adición de nuevos puntos intermedios y filtrado de puntos.

La adición de nuevos puntos intermedios busca dar mayor consistencia al patrón de la trayectoria al completar espacios vacíos formados por la rápida ejecución del signante y evitar que el método de filtrado elimine algún dato importante. Decimos que un punto nuevo está conformado de la siguiente manera:

$$p_{nuevo} = (x_{nuevo}, y_{nuevo}, z_{nuevo}) \quad (1)$$

Donde:

x_{nuevo} es la nueva posición en el eje de la abscisa,

y_{nuevo} es la nueva posición en el eje de la ordenada y

z_{nuevo} es la nueva posición en el eje de la cota.

El cálculo de una nueva posición para cada eje es dado por la ecuación (2), considerando al patrón de la trayectoria como un arreglo ordenado donde:

pos_{nueva} = el valor de la nueva posición,

pos_{actual} = es el valor de la posición actual y

$pos_{destino}$ = es el valor de la posición siguiente.

$$pos_{nueva} = \min(pos_{actual}, pos_{destino}) + \frac{\sqrt{\frac{pos_{actual} - pos_{destino}}{2}}}{2} \quad (2)$$

Este método es utilizado para calcular las nuevas posiciones de x, y, z . Este proceso fue iterado 5 veces para dar un aspecto uniforme en el espacio entre cada punto.

Con la finalidad de eliminar aquellos puntos que se encuentran fuera de contexto y que pueden ocasionar ruido se aplica el algoritmo de K-vecinos más cercanos (KNN). Utilizando el SDK del Kinect se convirtieron las posiciones dadas por el Kinect a posiciones en píxeles de una imagen de 640x480 píxeles. De esta forma el algoritmo KNN buscó los puntos que en un espacio de $K=4$ no tuvieran ninguna conexión y de esta forma proceder a su eliminación.

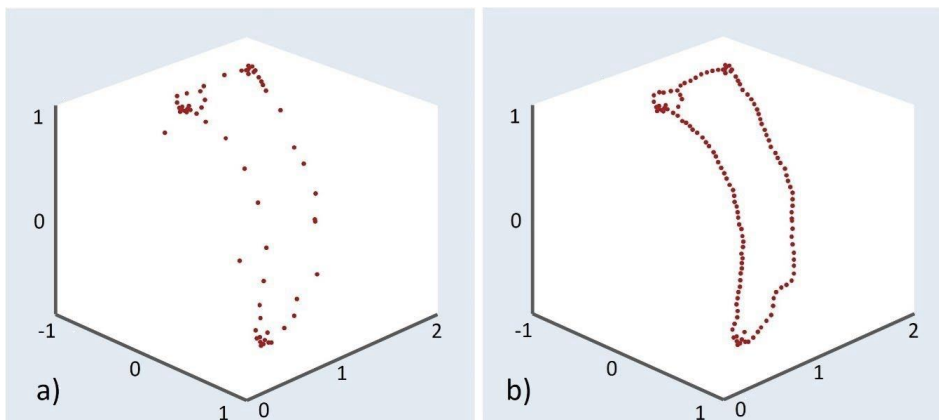


Figura 8. a) Patrón de puntos de la palabra **Hola** y b) nuevo patrón generado: filtrado y adición.

A partir del nuevo patrón reconstruido se realizó el cálculo de la Distancia Euclidiana usando la ecuación 3, en cada punto respecto al punto inicial y el etiquetamiento de este valor a su punto correspondiente. En la figura 9a) se puede ver las distancias calculadas para un punto en particular.

Decimos que $P = \{p_1, \dots, p_n\}$ es el conjunto de posiciones de la trayectoria de la mano ejecutadas en un plano tridimensional. Sea q la posición inicial de la mano. Para cada posición p_{ii} se calcula la distancia Euclidiana respecto a q .

$$d = \sqrt{(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2} \quad (3)$$

Posteriormente, se realizó el redondeo de cada distancia a su valor entero más cercano para poder después reducir el tamaño del patrón de acuerdo a la eliminación de los números repetidos secuencialmente dejando un solo valor (véase Figura 9b).

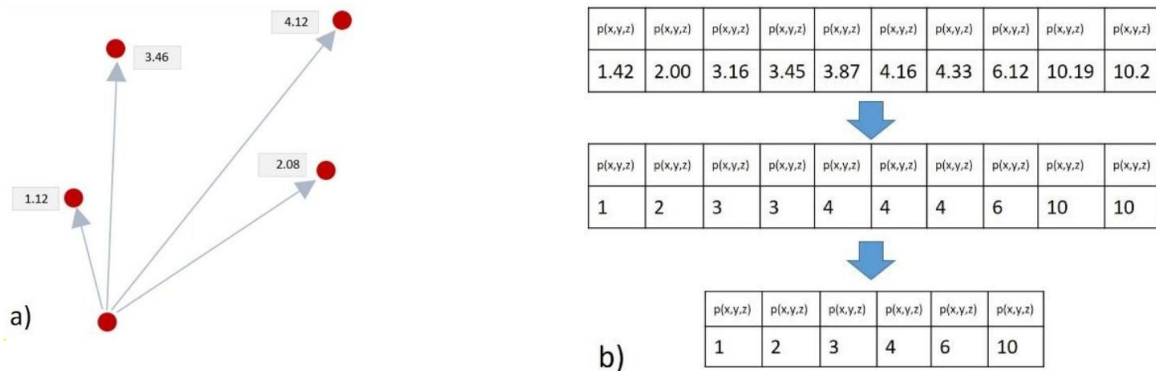


Figura 9. a) Distancia Euclidiana de cada punto respecto a su punto inicial y b) redondeo y reducción del patrón de puntos.

A partir de los datos reducidos se describe el patrón de movimiento de acuerdo a su geometría y a su trayectoria. Para describir al patrón de acuerdo a su geometría se consideró la anchura (4), altura (5), profundidad (6) y orientación del patrón (7).

$$anchura = x_{max} - x_{mddn} \quad (4)$$

$$altura = y_{max} - y_{mddn} \quad (5)$$

$$profundidad = z_{max} - z_{mddn} \quad (6)$$

$$\vec{m} = \sqrt{(x_{max} - x_{mddn})^2 + (y_{max} - y_{mddn})^2 + (z_{max} - z_{mddn})^2} \quad (7)$$

Donde:

- $x_{max} =$
- $y_{max} =$
- $z_{max} =$
- $x_{min} =$
- $y_{min} =$
- $z_{min} =$

Se calculan estos valores tanto para la mano izquierda como para la mano derecha.

De acuerdo a la observación hecha durante la realización de este trabajo se pudo constatar que las palabras que hemos usado y además un gran número de palabras de la LSM culminan en la posición más alta de su ejecución para después volver a la posición inicial velozmente. Por lo tanto se realizó la descripción de la trayectoria de la mano basada en la división del patrón en dos secciones respecto a la distancia Euclidiana máxima alcanzada. En la figura 10, se puede observar el procedimiento mediante el cual se obtiene la media aritmética (7) de las distancias Euclidianas y de las profundidades por cada sección y de todo el patrón. Este procedimiento se repite para el patrón de la mano izquierda y derecha.

$$\mu = \frac{1}{n} \sum_{d=1}^n d_i = \frac{d_1 + d_2 + \dots + d_n}{n} \quad (8)$$

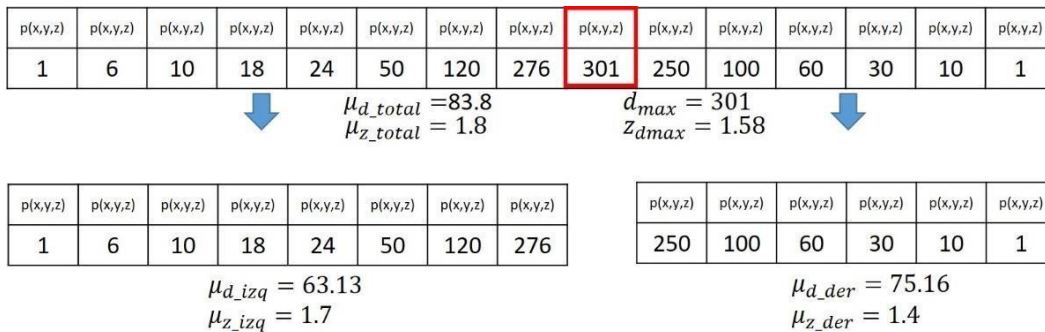


Figura 10. Representación del procedimiento para dividir el patrón y obtener 8 elementos característicos.

Finalmente, se crea un vector de 12 elementos mediante la concatenación de las características de altura, anchura, profundidad, ángulo, μ_{d_total} , μ_{z_total} , μ_{d_der} y μ_{z_der} . Entonces se forma un vector de 12 elementos por cada mano para generar un arreglo con un total de 24 características. Este arreglo es el que sirve de entrada a la red neuronal tanto para el aprendizaje de las señas del corpus como para su posterior reconocimiento de estas.

3.3. Sistema de Reconocimiento

Como se ha comentado en párrafos anteriores, el método de reconocimiento presentado en este trabajo utiliza una red neuronal de tipo perceptrón multicapa y *Backpropagation* como método de entrenamiento.

Si bien es cierto que se usó *Backpropagation* para el entrenamiento de la red, no fue el algoritmo base el utilizado sino una de las variantes que existen para este algoritmo. En este caso se utilizó la variante que hace uso de la función *Levenberg-Marquardt* para dicho entrenamiento. Además de la función del Gradiente Descendiente para el ajuste de los pesos de la red y el error cuadrático medio para evaluar su desempeño.

La ventaja de usar dicha variante de *Levenberg-Marquardt* es, principalmente, una convergencia más rápida del algoritmo de entrenamiento. Sin embargo, esta rapidez en la convergencia tiene un alto costo computacional ya que se requieren más recursos para su implementación. Como por ejemplo se requiere más memoria y la capacidad de cómputo del sistema en el cual se realiza el entrenamiento.

Tabla 3. Parámetros de la Red neuronal.

Parámetro	Valor
Tipo de red	MLP
Algoritmo de Entrenamiento	Backpropagation
Variante	<i>Levenberg-Marquardt</i>
Num. Capas ocultas	3
Razón de aprendizaje,	0.2
Num. Épocas Máximo	10000
Error mínimo	1e-10

La arquitectura de esta red neuronal consiste en una capa de entrada, 3 capas intermedias con 10, 30, y 10 neuronas respectivamente y una capa de salida. Además de la cantidad de capas y de neuronas por capa se consideran otros parámetros que se presentan en la tabla 3.

4. Resultados

En esta sección se presentan los resultados obtenidos a partir de las pruebas realizadas. Una vez validado el algoritmo de entrenamiento se procedió a entrenar el sistema de reconocimiento mediante el método *KFold Cross Validation (KFCR)*. Este método de validación cruzada permite evaluar el desempeño y la robustez del sistema de reconocimiento implementado. Lo anterior al evaluar el sistema mediante conjuntos de entrenamiento tomados al azar y validando con las imágenes restantes.

Entonces, si se considera que solo se cuenta con 30 muestras de cada palabra se debe de asignar un valor de *kfold* máximo. En este caso se asignó un valor máximo de $K = 6$, dado que el número de muestras por palabra es de 30. Esto significa que se realizaron 6 pruebas diferentes de K_1 hasta K_6 y en cada una de ellas se tomaron 5 imágenes de las 30 muestras de cada palabra en el corpus. Cada grupo de 5 imágenes era diferente para cada *kfold*. Por lo tanto, en cada iteración se tomaron 265 muestras considerando 5 de cada palabra. Estas muestras fueron separadas para la validación y las otras 1,325 muestras (25 de cada palabra) se utilizaron para el entrenamiento.

Tabla 3. Precisión obtenida en las pruebas realizadas

Fold	Precisión
K_1	94.34%
K_2	93.96%
K_3	98.87%
K_4	90.19%
K_5	92.83%
K_6	90.57%

En la gráfica de la Figura 11, se muestran los valores de reconocimiento promedio obtenidos después de la ejecución de cada uno de los 6 experimentos uno para cada uno de los 6-folds. En esta Figura 11, cada barra representa el porcentaje promedio obtenido para cada una de las palabras de LSM evaluadas.

La tasa de reconocimiento obtenida durante cada iteración o para cada *kfold* se muestra en la tabla 3, aquí es posible observar que los porcentajes de reconocimiento obtenido para cada *kfold* no es muy diferente y que oscilan entre 90% y 99%. Al calcular la desviación estándar a partir de los resultados generados al evaluar cada *kfold* se obtiene una desviación de 0.03153, lo cual es una dispersión de los datos muy pequeña. Esto representa que el sistema de reconocimiento implementado es robusto ya que no hay gran variación entre los resultados obtenidos en cada uno de los experimentos realizados. Cabe resaltar el hecho de que en cada experimento se realizó la evaluación del sistema con imágenes diferentes. Es decir, en cada experimento se usaron imágenes de prueba diferente del anterior.

La precisión del clasificador en cada *kfold* es determinada por la ecuación 9, donde:

TP son los verdaderos positivos y

FP son los falsos positivos.

Los verdaderos positivos son los patrones clasificados correctamente y los falsos positivos los patrones clasificados de manera incorrecta. Finalmente se obtuvo una precisión media del 93.46% la cual fue determinada por la ecuación 10.

$$precddsdonK (\%) = \frac{TP}{(TP + FP)} * 100 \quad (9)$$

$$precddsdon meddda(\%) = \frac{1}{n} \sum_{i=1}^n K_i = \frac{K_1 + K_2 + \dots + K_n}{n} \quad (10)$$

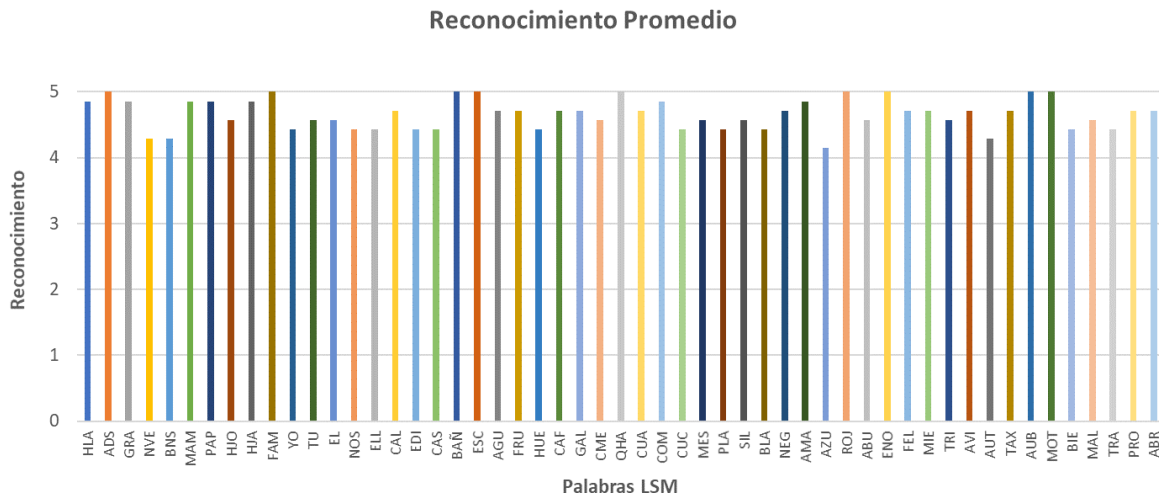
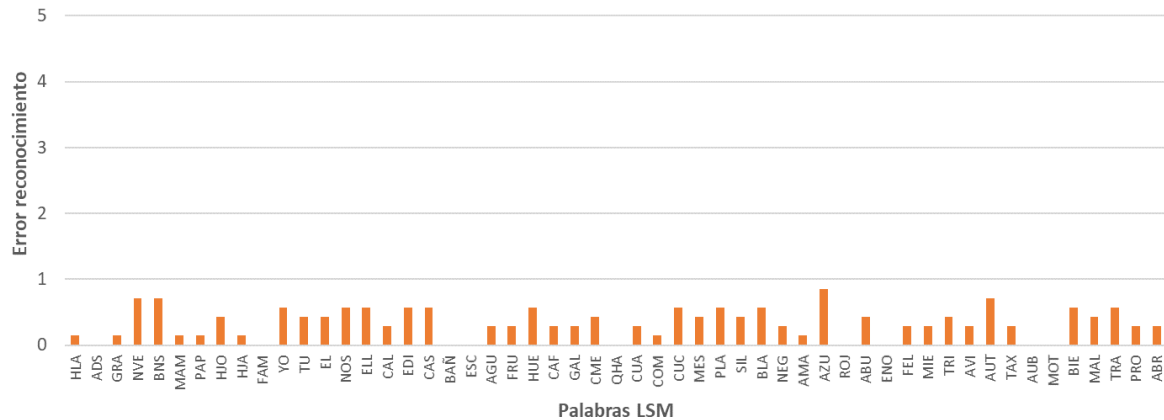


Figura 11. Matrices de confusión de cada iteración.

En la Figura 12, se observa el error promedio resultante de la diferencia del valor esperado y el valor obtenido por el sistema de reconocimiento en cada *kfold*. En esta gráfica se puede apreciar de mejor manera que la palabra que más se confunde al momento de realizar el reconocimiento es AZUL la cual se identifica como AZU en la gráfica. En el otro lado estaría que el sistema reconoce sin problemas 9 palabras, las cuales son: ADIOS, FAMILIA, BAÑO, ESCUELA, ¿QUÉ HACES?, ROJO, ENOJADO, AUTOBUS y MOTOCICLETA. Identificadas en la gráfica como: ADS, FAM, BAÑ, ESC, QHA, ROJ, ENO, AUB y MOT, respectivamente.

Error promedio



5.- Conclusiones y Perspectivas

De acuerdo con los resultados obtenidos en este trabajo se concluye que este modelo tiene como principal ventaja la descripción de la trayectoria del movimiento de la mano a partir de un procedimiento simple y rápido.

La precisión media alcanzada del 93.46% muestra que el modelo presenta una buena eficiencia al reconocer las palabras de la LSM aprendidas con el método de redes neuronales. Además, al existir un alto número de patrones diferentes, se considera que este modelo es capaz de reconocer un número mucho mayor de palabras y puede ser utilizado para el reconocimiento de gestos en otras áreas tal es el caso de la robótica de servicio, en donde se controlaría un robot mediante el lenguaje natural de la lengua de señas.

En trabajos futuros se plantea ampliar tanto el número de palabras como la cantidad de muestras del corpus que nos permita contar con más información para continuar realizando experimentos en el reconocimiento de la LSM. Además, se pretende implementar este enfoque en tiempo real para validar su desempeño y dar cavidad a pruebas con fines educativos y experimentales en instituciones de apoyo a personas con discapacidad en la escucha o en el habla.

Finalmente, se puede afirmar que este trabajo representa una buena aproximación para la construcción de un traductor de la LSM. Por lo que también se propone en trabajos futuros realizar el procesamiento de todos los datos que se extraen mediante el sensor Kinect. Es decir utilizar las nubes de puntos, imágenes de color y profundidad que se encuentran en el corpus creado para robustecer este trabajo.

Referencias

- INEGI Censo población 2020. (2021). Población. Discapacidad. Cuentame.inegi.org.mx. Recuperado en 5 de mayo de 2021, de: <http://www.cuentame.inegi.org.mx/poblacion/discapacidad.aspx?tema=P>.
- Sensor Kinect. (2021). Microsoft Kinect for Windows Specs y Prices. CNET. Retrieved 3 June 2021, from <https://www.cnet.com/products/microsoft-kinect-for-windows/>.
- Microsoft Kinect SDK. (2021). Download Kinect for Windows SDK v1.0 from Official Microsoft Download Center. Microsoft.com. Retrieved 8 July 2021, from <https://www.microsoft.com/en-us/download/details.aspx?id=28782>.
- Jimenez, J., Martin, A., Uc, V. y Espinosa, A. (2017). Mexican Sign Language Alphanumeric Gestures Recognition using 3D Haar-like Features. *IEEE Latin America Transactions*, 15(10), 2000–2005. <https://doi.org/10.1109/TLA.2017.8071247>
- Pérez, L. M., Rosales, A. J., Gallegos, F. J., y Barba, A. V. (2017). LSM static signs recognition using image processing, 14th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE 2017), pp. 1-5, doi: 10.1109/ICEEE.2017.8108885.
- Carmona-Arroyo, G., Rios-Figueroa, H. V., y Avendaño-Garrido, M. L. (2021). Mexican Sign-Language Static-Alphabet Recognition Using 3D Affine Invariants. In M. Malarvel, S. R. Nayak, P. K. Pattnaik, y S. N. Panda (Eds.), *Machine Vision Inspection Systems, Volume 2* (1st ed., pp. 171–192). Wiley. <https://doi.org/10.1002/9781119786122.ch9>
- Lahamy, H., y Lichti, D. (2012). Towards Real-Time and Rotation-Invariant American Sign Language Alphabet Recognition Using a Range Camera. *Sensors*, 12(11), 14416–14441. MDPI AG. Retrieved from <http://dx.doi.org/10.3390/s121114416>
- Agarwal, A. y Thakur, M. K. (2013). Sign language recognition using Microsoft Kinect. 2013 Sixth International Conference on Contemporary Computing (IC3), 181–185. <https://doi.org/10.1109/IC3.2013.6612186>
- Luis-Pérez, F. E., Trujillo-Romero, F., y Martínez-Velazco, W. (2011). Control of a Service Robot Using the Mexican Sign Language. *Advances In Soft Computing*, 419-430. https://doi.org/10.1007/978-3-642-25330-0_37
- Estrivero-Chavez, C., Contreras-Teran, M., Miranda-Hernandez, J., Cardenas-Cornejo, J., Ibarra-Manzano, M., y Almanza-Ojeda, D. (2019). Toward a Mexican Sign Language System using Human Computer Interface. 2019 International Conference On Mechatronics, Electronics And Automotive Engineering (ICMEAE). <https://doi.org/10.1109/icmeae.2019.00010>
- LeapMotion. (2021). LeapMotion Datasheet. Ultraleap.com. Retrieved 5 June 2021, from https://www.ultraleap.com/datasheets/Leap_Motion_Controller_Datasheet.pdf.
- Garcia-Bautista, G., Trujillo-Romero, F., y Caballero-Morales, S. (2017). Mexican sign language recognition using kinect and data time warping algorithm. 2017 International Conference On Electronics, Communications And Computers (CONIELECOMP). <https://doi.org/10.1109/conielecomp.2017.7891832>
- Tazhigaliyeva, N., Kalidolda, N., Imashev, A., Islam, S., Aitpayev, K., Parisi, G., y Sandygulova, A. (2017). Cyrillic manual alphabet recognition in RGB and RGB-D data for sign language interpreting robotic system (SLIRS). 2017 IEEE International Conference On Robotics And Automation (ICRA). <https://doi.org/10.1109/icra.2017.7989526>
- Hazari, S., Asaduzzaman, Alam, L., y Goni, N. (2017). Designing a sign language translation system using kinect motion sensor device. 2017 International Conference On Electrical, Computer And Communication Engineering (ECCE). <https://doi.org/10.1109/ecace.2017.7912929>

- Chai, X., Li, G., Lin, Y., Xu, Z., Tang, Y., Chen, X., y Zhou, M. (2013, April). Sign language recognition and translation with kinect. In IEEE Conf. on AFGR (Vol. 655, p. 4).
- Ghotkar, A., y Kharate, G. (2015). Dynamic Hand Gesture Recognition for Sign Words and Novel Sentence Interpretation Algorithm for Indian Sign Language Using Microsoft Kinect Sensor. *Journal of Pattern Recognition Research*, 10(1), 24–38. <https://doi.org/10.13176/11.626>
- Garcia-Bautista, G., Trujillo-Romero, F., y Diaz-Gonzalez, G. (2016). Advances to the development of a basic Mexican sign-to-speech and text language translator (A. G. Tescher, Ed.; p. 99713E). <https://doi.org/10.1117/12.2238281>
- Sosa-Jimenez, C., Rios-Figueroa, H., Rechy-Ramirez, E., Marin-Hernandez, A., y Gonzalez-Cosio, A. (2017). Real-time Mexican Sign Language recognition. 2017 IEEE International Autumn Meeting On Power, Electronics And Computing (ROPEC). <https://doi.org/10.1109/ropec.2017.8261606>
- Molchanov, P., Gupta, S., Kim, K. y Kautz, J. (2015). Hand gesture recognition with 3D convolutional neural networks. 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 1–7. <https://doi.org/10.1109/CVPRW.2015.7301342>
- Oyewole, O. G., Nicholas, G., Oludele, A., y Samuel, O. (2018). Bridging communication gap among people with hearing impairment: An application of image processing and artificial neural network. *International Journal of Information and Communication Sciences*, 3(1), 11.
- Gürpınar, C., Uluer, P., Akalın, N. et al. Sign Recognition System for an Assistive Robot Sign Tutor for Children. *Int J of Soc Robotics* 12, 355–369 (2020). <https://doi.org/10.1007/s12369-019-00609-9>
- Zhang, Z., Su, Z., y Yang, G. (2019). Real-Time Chinese Sign Language Recognition Based on Artificial Neural Networks*. 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO), 1413–1417. <https://doi.org/10.1109/ROBIO49542.2019.8961641>
- Jiang, S., Sun, B., Wang, L., Bai, Y., Li, K., y Fu, Y. (2021). Skeleton aware multi-modal sign language recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 3413-3423).
- Fregoso, J., Gonzalez, C. I., y Martinez, G. E. (2021). Optimization of Convolutional Neural Networks Architectures Using PSO for Sign Language Recognition. *Axioms*, 10(3), 139. MDPI AG. Retrieved from <http://dx.doi.org/10.3390/axioms10030139>
- Calvo, M.T. (2004). *Diccionario Español - Lengua de Señas Mexicana (DIELSEME): estudio introductorio*. Dirección de Educación Especial: México.
- Serafín de Fleischmann, M., González Pérez, R. (2011). *Manos con voz, Diccionario de Lenguaje de Señas Mexicana*. Primera edición, Libre Acceso, A.C., ISBN 978-607-9134-01-3
- Código estándar Mex-Esp. (2021). ISO 639 — Language codes. ISO. Retrieved 14 Mayo 2021, from <https://www.iso.org/iso-639-language-codes.html>.
- Zhang, Z. (2012). Microsoft Kinect Sensor and Its Effect. *IEEE MultiMedia*, 19(2), 4–10. <https://doi.org/10.1109/MMUL.2012.24>
- Dal Mutto, C., Zanuttigh, P., y Cortelazzo, G. (2012). *Time-of-flight cameras and Microsoft Kinect*. Springer.
- Rossenblatt, F. (1957). *The perceptron, a perceiving and recognizing automation*. Cornell Aeronautical Laboratory. Report No. 85-460-1.
- Werbos, P. (1974). *Beyond Regression: New tools for prediction and analysis in the behavioral sciences* (Ph.D). Harvard University.

- Rumelhart, D. E., Hinton, G., y Williams, R. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088), 533-536. <https://doi.org/10.1038/323533a0>
- BSL Corpus. (2021). Home | BSL Corpus Project. British Sign Language Corpus Project. Retrieved 13 March 2021, from <http://www.bslcorpusproject.org/>.
- GSL Corpus. (2021). German Sign Language Korpus. Retrieved 14 March 2021, from: <http://www.sign-lang.uni-hamburg.de/dgs-korpus/index.php/welcome.html>
- LSE Corpus. (2021). Corpus de la lengua de signos española. *Corpuslse.es*. Retrieved 14 July 2021, from <https://www.corpuslse.es/>.
- LSP Corpus. (2021). Repositorio Digital de la Lengua de Señas Peruana - Grupo Señas Gramaticales. Grupo Señas Gramaticales. Retrieved 10 March 2021, from <https://investigacion.pucp.edu.pe/grupos/senasgramaticales/proyecto/repositorio-digital-de-la-lengua-de-senas-peruana/>.



Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-CompartirIgual 2.5 México.

Adaptación al uso tecnología en el ámbito educativo durante la pandemia derivada del COVID-19 en México

Use adaptation for educational technology during the pandemic of the COVID-19 in Mexico

Adriana Peña Pérez Negrón¹
adriana.pena@cucei.udg.mx

Carlo Giovannella²
mifav@roma2.infn.it

Aurora Espinoza-Valdez¹
aurora.espinoza@academicos.udg.mx

Mirna Muñoz⁴
mirna.munoz@cimat.mx

David Bonilla Carranza¹
jose.bcarranza@academicos.udg.mx

Marcello Passarelli³
passarelli@itd.cnr.it

¹Universidad de Guadalajara CUCEI, México.

² ASLERD, Dept. SPFS, University of Rome Tor Vergata, Rome, Italia.

³Istituto per le Tecnologie Didattiche, Consiglio Nazionale delle Ricerche.

⁴Centro de Investigación en Matemáticas, A.C. Unidad Zacatecas, México.

Resumen

La pandemia por el COVID-19 tomó al mundo por sorpresa. En México, al igual que en muchas partes del mundo, millones de estudiantes cambiaron al aprendizaje en línea de un día para otro. En este documento se presentan los resultados de una encuesta aplicada a estudiantes universitarios con la finalidad de entender tanto su adaptación como el involucramiento en las tecnologías para la educación en línea. Participaron 1297 alumnos de diversas partes de la República mexicana. Los principales resultados indican que, la actitud de los usuarios de cara al futuro hacia las tecnologías, su uso y aprendizaje, se ve influida por la sustentabilidad percibida como resultado de innovación en contraste con su preservación. Otros resultados muestran que, aun cuando la adaptación a la tecnología ha sido favorable, los estudiantes extrañan el salón de clase.

Palabras clave: tecnología para trabajo a distancia, adaptación a la tecnología, educación en línea,

Abstract

The COVID-19 pandemic took the world off guard. In Mexico, as in many other parts of the world, millions of students switched from the classroom to distance learning from one day to the other. In this paper are presented results from a survey applied to Mexican university professors and students, aimed to understand their adaptation and their involvement in the technologies to support online education. One hundred and eighty-seven teachers and 1297 students from diverse parts of the Mexican republic participated. Main results indicated that, the users' attitude towards the future use and learning of the technologies, is influenced by the technology sustainability perceived as a result of its innovation in contrast with its preservation. Other results show that, even they have been adapted to the technology, they miss classrooms.

Keywords: distance work technology, technology adaptation, e-learning.

1. Introducción

La enfermedad COVID-19 ocasionada por el coronavirus SARS-COV2 paralizó el mundo. Esta enfermedad tuvo su origen en diciembre del 2019 en Wuhan, Hubei, China, por ese entonces la Comisión Municipal de Salud de Wuhan notificó sobre un conglomerado de casos de neumonía en la ciudad, que posteriormente se relacionarían con el nuevo coronavirus. La Organización Mundial de la Salud (OMS) declara la pandemia del COVID-19 el día 11 de marzo del 2020 (OMS, 2020). En México, el 27 de febrero del 2020 se confirma el primer caso. En los primeros días de marzo el gobierno mexicano implementa programas de prevención que contemplan medidas de aislamiento y distanciamiento social, el lavado de manos, y en algunos estados el uso de cubre-bocas. En el mismo mes de marzo, el día 17, se anuncia el cierre de toda actividad no esencial, y para el 27 de marzo oficialmente se cierran también las escuelas (Martínez y Gamboa, 2020; Villasís-Keever y García-Bolaños, 2020). En algunos estados como Guanajuato, Jalisco y Yucatán el cierre de las escuelas fue de la mano con las medidas de aislamiento, lo cual es secundado por algunos otros estados.

La pandemia ocasionó en México, al igual que en muchas partes del mundo, que millones de alumnos se vieran obligados a continuar sus estudios de manera remota; este cambio tuvo que hacerse de un día para otro y sin preparación previa, por lo que profesores y alumnos fueron improvisando sobre la marcha. En este contexto, el uso de herramientas tecnológicas fue clave para la transición.

De acuerdo con un reporte presentado por la Secretaria de Educación Pública (SEP) mexicana, con base en información reportada por las universidades estatales públicas, las acciones frente a la pandemia de estas entidades educativas incluyeron el diseño de aplicaciones digitales y materiales informativos, diseño e implementación de campañas de concientización, producción de equipo y materiales sanitarios, servicios de asesoría sanitaria, psicológica, nutricional y jurídica, y capacitación a trabajadores de la salud (SEP, 2020). La mayoría de las Instituciones de Educación Superior (IES), adoptaron diferentes medidas para continuar con la educación en modalidad a distancia, pero es innegable que habrá un efecto negativo en el aprendizaje de los alumnos, aunque paradójicamente también se esperan impactos positivos en cuanto a tecnología, ya que el uso de tecnologías digitales en la educación seguramente se quedará para formar parte de contenidos, estructura y formas (Malo Álvarez, Maldonado-Maldonado, Gacel-Ávila, y Marmolejo, 2020), este impacto positivo muy probablemente alcanzará también otros ámbitos especialmente si se trata de jóvenes universitarios que pronto se integrarán a la industria.

Una gran cantidad de información se ha publicado en referencia al impacto del COVID-19. En referencia a educación superior, en (DAAD, 2020) se puede encontrar una lista de estudios con un apartado dedicado a Latinoamérica. Los análisis y estudios de esta época, representan evidencia que en el futuro permitirá identificar la universalidad y las peculiaridades de cada contexto sobre el uso de la tecnología (Giovannella y Passarelli, 2020).

En el espacio universitario en México, Maldonado Gómez et al. (2020) presentaron un análisis comparativo del impacto de la virtualidad en el desarrollo del proceso de enseñanza entre universidades de México y Argentina mediante un cuestionario aplicado a 206 profesores y estudiantes de ambos países, de los que el 12% fueron maestros. El documento no especifica cuántos maestros de la muestra pertenecen a cada país. Entre los resultados, se detectó que el 65% de los maestros mexicanos estuvo de acuerdo en la infraestructura y acompañamiento institucional del acceso a recursos tecnológicos, el 41% dijo tener problemas de adaptación y el 65% consideró como

incorrecto el proceso de aprendizaje. En términos generales, los números en las encuestas son similares para ambos países. De acuerdo con los autores tanto para México como para Argentina se encontró la necesidad de incrementar el factor humano y buscar alternativas para trabajar con el estrés y la angustia generados por las clases a distancia.

Sánchez Mendiola et al. (2020), llevaron a cabo una encuesta para maestros de bachillerato, licenciatura y posgrado de la Universidad Nacional Autónoma de México con 383 respuestas, de las cuales el 73.4% fue de maestros universitarios. En las respuestas se encontró que poco más del 60% de los profesores está de acuerdo con el tipo de apoyo institucional recibido. Entre los principales problemas reportados están los relacionados con la logística como manejo de tiempo, horarios de clase, espacio físico para trabajar; relacionados con la tecnología como el acceso a Internet, disposición de equipo de cómputo, conocimiento de plataformas educativas; relacionados con la pedagogía, conocimiento de herramientas para educación a distancia, envío de tareas y actividades, por ejemplo; y socio-afectivos, que representan aspectos emocionales, en ese orden de frecuencia.

Aunque de momento es imposible conocer el impacto y las implicaciones que la pandemia del COVID-19 tendrá sobre la aplicación y aceptación de la tecnología para el aprendizaje a distancia, muchas de ellas con fundamento en las diferencias culturales y sociales de la comunidad académica, es innegable que seguramente también habrá un impacto positivo en cuanto a la aceptación y uso de la tecnología (Martínez y Gamboa, 2020; Giovannella, et al., 2021).

En este documento se presentan el análisis y los resultados de un cuestionario desarrollado por la Asociación para Ecosistemas de Aprendizaje Inteligente y Desarrollo Regional (ASLERD, por sus siglas en inglés), cuyo objetivo principal es auxiliar escuelas y ecosistemas de aprendizaje para promover sistemas inteligentes de aprendizaje a fin de conseguir el bienestar de las personas involucradas en el proceso de enseñanza. ASLERD es una asociación europea (<http://www.aslerd.org>). El cuestionario fue traducido del inglés al español y adaptado para México.

El cuestionario tiene la finalidad de entender la actitud, en el ámbito universitario, hacia las tecnologías que pueden soportar el aprendizaje, permitiéndonos entender primero la preparación previa para el cambio de clases presenciales a clases en línea, así como el apoyo institucional recibido para la transición, y finalmente la percepción de cara al futuro en la integración y uso de tecnologías digitales.

El estudio fue lanzado a nivel nacional para profesores en el mes de septiembre del 2020 y posteriormente para alumnos en el mes de octubre del 2020, aquí se presentan los resultados de este último. En el mes de septiembre y octubre las universidades ya habían cerrado el primer ciclo escolar del año y en su mayoría tenían alrededor de un mes de haber iniciado el segundo ciclo escolar; esto incluye un periodo vacacional en el que muchos profesores tuvieron oportunidad de capacitarse en el uso de tecnologías.

1.1 Antecedentes

El conocido modelo de aceptación de la tecnología TAM (por sus siglas en inglés, *Technology Acceptance Model*) propuesto por Davis (1989), establece que lo que lleva a la utilización de un sistema tecnológico está determinado por ciertas variables externas que tienen un impacto tanto en la percepción de utilidad como en la percepción de la facilidad de uso por parte de los posibles usuarios. Salloum, et al. (2019), estudiaron los factores externos directamente relacionados con la tecnología para la educación en línea mediante una revisión literaria. De acuerdo con Salloum et al. (2019) las variables externas del modelo de Davis (1989) se pueden dividir en características del sistema y factores individuales, en la Figura 8 se muestra el TAM adaptado para educación en línea, en el que básicamente se distinguen las variables externas dividiéndolas en características del sistema y factores individuales.

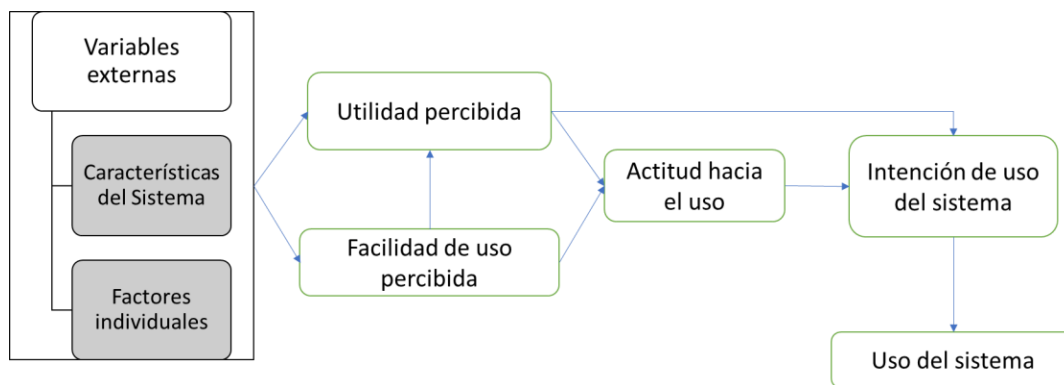


Figura 8: TAM para educación en línea (Salloum, et al. 2019)

El TAM está encaminado a entender la intención de uso de un sistema, sin embargo, en el caso de la pandemia, el usuario se vio obligado a hacer uso de la tecnología, aun cuando, por ejemplo, no considerara que era fácil de usar. De tal forma que, basado en el TAM y para establecer un modelo en ecosistemas para determinar la actitud de los usuarios a involucrarse con la tecnología de cara al futuro, se llevó a cabo una encuesta a profesores universitarios en tres países, Italia, México e Iraq (Giovannella, et al., 2021). Derivado de la similitud entre las redes de causalidad de los tres países se propone el modelo MAETI (por sus siglas en inglés *Model for Attitude to get Engaged in Technological Innovation*).

El modelo MAETI sugiere que la percepción del entorno que regularmente influyen en las intenciones futuras del usuario para el uso de la tecnología, no necesariamente debe pasar por la percepción de facilidad y utilidad de uso, ya que, al aumentar la dimensión y complejidad del sistema, estas variables tienden a perder su poder de mediación. En contraste, la madurez de la tecnología, y la disparidad entre la innovación y la preservación, tienen influencia sobre la percepción de la tecnología en cuanto a su sustentabilidad y viabilidad, lo que influye directamente sobre las intenciones futuras de uso y aprendizaje hacia las nuevas tecnologías.

A diferencia de los profesores, la actitud de los estudiantes para involucrarse con las tecnologías para el aprendizaje, la comunicación mediada por computadora y el uso de herramientas de trabajo a distancia, pudo verse influenciada no solo por características de la tecnología, sino por el papel que los profesores y la universidad jugaron para su implementación. Por otro lado, se esperaría que su actitud hacia el uso de la tecnología, previo a la pandemia, fuese menos conservador que el de los profesores. Con la finalidad establecer los factores que pudieran presentar influencia en la actitud de los estudiantes universitarios hacia las tecnologías de innovación, se realizó una encuesta que presenta además datos importantes sobre el estado general en cuanto a capacidad tecnológica de los estudiantes universitarios en México.

2. Método

2.1 Cuestionario

El cuestionario fue desarrollado con la herramienta Formularios Google™ y consta de tres secciones principales, en el Apéndice A se encuentran las preguntas numeradas, en lo sucesivo el número de pregunta al que corresponden los resultados o la variable asociada por primera vez, se muestra en corchetes cuadrados precedido con la letra P.

En el cuestionario, en un apartado inicial se informa a los participantes sobre el uso de los datos y se pide su autorización para la utilización en este estudio de los mismos. Luego, la primera sección contiene preguntas sobre antecedentes socio-biográficos, los datos solicitados son: género, edad, tipo de universidad (pública o privada), área de estudio, estado de la República en el que estudian y el nombre de la universidad a la que pertenecen.

La segunda sección consta de 38 preguntas, 21 de opción múltiple, y 17 abiertas en las que se piden comentarios sobre algunas de las preguntas de opción múltiple. En esta sección las preguntas van encaminadas a comprender la respuesta a la pandemia y las condiciones de operación a seis meses del cierre de las escuelas.

La tercera sección consta de 34 preguntas, 16 de opción múltiple y 18 preguntas abiertas en las que se piden comentarios sobre las preguntas de opción múltiple y dos preguntas más, una acerca de dificultades para contestar el cuestionario y la otra sobre algún consejo para sus compañeros. La finalidad de esta sección fue recabar información sobre cambios en la opinión sobre la tecnología para la enseñanza en línea y expectativas futuras al respecto. Todas las preguntas son opcionales, no se obliga a responder ninguna de ellas para continuar con el cuestionario, excepto la inicial de aceptación del uso de los datos para este análisis.

2.2 Participantes

El cuestionario fue socializado por los autores mediante correos electrónicos a colegas y haciendo uso de redes sociales. Este cuestionario estuvo abierto durante un mes, del 1 al 31 de octubre del 2020. Se recibieron 1310 respuestas de las cuales 13 fueron descartadas por estar vacías o porque el encuestado contestó no estar de acuerdo con el uso de los datos. De las 1297 respuestas validas, 55% de los encuestados son hombres (714), 44% mujeres (575) y 1% prefirió no contestar (8) [P2]. Las edades de los estudiantes [P3] son: 1 alumno mayor de 41 años, 2% en el rango de entre 31-44 años, 2% en el rango de entre 27-30 años, 22% en el rango de entre 22-26 años, y la mayoría (74%) fueron estudiantes menores a 22 años.

En su gran mayoría (99%), los encuestados provienen de una institución pública, solo 16 alumnos provienen de instituciones privadas [P4]. De manera muy similar, solo 21 estudiantes son de posgrado (9 de maestría y 12 otro), y la mayoría corresponde al nivel universitario de licenciatura (99%) [P5]. En su primer año de estudios están el 36% de los participantes, segundo año 14%, tercer año 15%, cuarto año 9%, quinto año 12%, y el 13% contestó 'otro' [P6]. Los participantes pertenecen a 14 áreas de estudio [P7] de la siguiente forma: Actuaría (5), Administración (83), Ciencias Políticas (19), Ciencias Sociales (119), Computación e Informática (366), Contabilidad (11), Derecho (147), Economía (138), Estadística (16), Humanidades (12), Ingenierías (357), Mercadotecnia (6), Ciencias Básicas (16) y Medicina o Salud (2).

Las respuestas provienen de 14 estados de la República [P8]: Aguascalientes, Coahuila, Estado de México, Guerrero, Jalisco, Mazatlán, Puebla, San Luis Potosí, Sinaloa, Sonora, Veracruz, Yucatán y Zacatecas; entre los que destacan el Estado de México con 62%, Jalisco 14%, y Zacatecas 10%. En referencia a la fatiga, 63 encuestados abandonaron la encuesta en algún punto o contestaron menos de la mitad de las preguntas.

3. Resultados

El análisis de los resultados se divide en tres partes. Primero se hace un análisis descriptivo de las respuestas de opción múltiple, excepto las de valores en escala numérica. En la segunda parte se hace un análisis de las respuestas de valores numéricos en cuanto a preparación para el cambio de presencial a virtual, y el cambio de percepción en cuanto a tecnologías, y las expectativas respecto al futuro. La tercera parte contiene una descripción de algunas respuestas abiertas seleccionadas.

3.1 Respuestas de opción múltiple e indicadores opuestos

Esta sección comprende preguntas con respuesta en las que el participante seleccionó su respuesta o respuestas de entre varias opciones.

3.1.1 Sección 2 condiciones de respuesta ante la pandemia

Respecto a la conexión a internet, el dispositivo que más utilizan los encuestados para tomar clase es el celular (75.2%), seguido de las computadoras portátiles (55.3%), la computadora de escritorio (26.9%) y el de menor uso es la tableta (2.5%) [P11]. El tipo de conexión a Internet más utilizada es la Fibra óptica (45.3%), seguida del celular (24.4%) [P12].

En cuanto a las dificultades para tomar clases a distancia [P13], la más frecuentemente seleccionada es la mala conexión a internet (68.8%). Además, el 47.4% de los encuestados manifestó tener problemas para concentrarse, el 32.2% extraña el pizarrón físico, y el 31.6% tuvo dificultades relacionados con la cámara (e.g., mostrar un vídeo o hablar frente a la cámara), principalmente. En referencia al tiempo para acostumbrarse a la nueva modalidad [P20] para el 12% fue de menos de una semana, para el 30.4% entre 1 y 2 semanas, para el 28% más de dos semanas, y el 29.2% contestó que todavía no se acostumbra. El tiempo que dedican a tomar clases [P26] es de más de 4 horas para el 62.6%, y entre 2 y 4 horas para el 28.7%, los demás participantes pasan menos tiempo en clases. En referencia a todas sus actividades escolares [P27] (clases, tareas, reuniones, etc.), el 21.5% dijo estar más de 8 horas al día en línea, 27% entre 6 y 8 horas, 26% entre 4 y 6 horas, y 17.9% entre 2 y 4 horas.

Los encuestados consideran que entre los propósitos del uso de las tecnologías [P39] están principalmente el de dejar lecciones interactivas (70.5%) y compartir contenido (62.3%). El método de evaluación [P41] más utilizado es el de hacer examen en línea (73.2%), y el menos utilizado es el de hacer una entrevista en línea (solo el 10%). Por último, un poco más de la mitad de los encuestados (55.1%) siente que sus relaciones con personas de la institución han ido en decremento, mientras el resto opina que se han incrementado y mejorado [P46].

3.1.2 Sección 3 cambios de opinión sobre tecnología y enseñanza en línea

Las preguntas de esta sección tienen como parte del texto: “Después de esta experiencia...” para contextualizar el cambio de opinión.

Aquí se encontró que la utilidad de la tecnología para apoyar una educación de alta calidad se percibe principalmente [P52] como para compartir contenido (65.2%), asignar tareas/ejercicio para desarrollarse de forma asíncrona (58.4%), y la producción de contenido (49.8%). De acuerdo con las respuestas de los encuestados, la tecnología educativa permite mejorar principalmente [P54] el aprendizaje autónomo y auto-regulado (58.6%), y el desarrollo de una identidad digital (47.5%).

Las competencias digitales de los encuestados han mejorado principalmente [P61] en la descarga, organización e intercambio de contenido digital (55.1%), en el uso de herramientas para la comunicación digital (51.8%), y en el procesamiento y visualización de información (45%). De las diferentes prácticas relacionadas con actividades educativas, lecciones, ejercicios/prácticas, revisiones, exámenes escritos, y exámenes orales, se les preguntó si es mejor hacerlo de manera presencial o virtual [P67]. Las respuestas se pueden observar en el gráfico de la Figura 1, en donde podemos ver que la preferencia es mucho mayor para realizar todas estas actividades de manera presencial.

En particular, después de esta experiencia crees que es mejor realizar

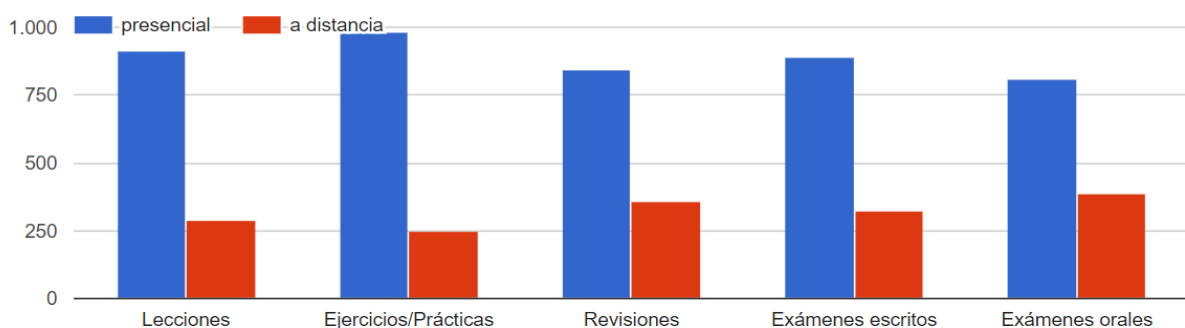


Figura 1: Frecuencia en la preferencia de modalidad por tipo de actividad

De cara al futuro se les preguntó cómo preferirían continuar sus clases, el 69.5% seleccionó el formato presencial frente al 23% de manera mixta o semipresencial, y solo el 7.4% preferiría continuar de manera totalmente virtual [P64].

3.2 Respuestas de opción múltiple en escala numérica

Las preguntas con escala numérica del -5 al + 5 manejan indicadores opuestos. Por ejemplo, para la pregunta en cuanto a la organización del tiempo y actividades (OTA), en el extremo izquierdo el valor es ‘mucho peor (-5)’ y en el derecho es ‘mucho mejor (+5)’. En las actividades educativas (AELD, AETI, AEAS y AEIC) se indican los dos extremos en la descripción. La Tabla 1, presenta las medidas centrales de las respuestas a estas preguntas. Llama la atención que la OTA, es en general mucho mejor (3.00 de media, aunque con una desviación de un poco más de 3 puntos). En cuanto a las actividades educativas tenemos ligeramente más debate que lecciones, son más interactivas que solamente transmisivas, se realizan más actividades síncronas que asíncronas, y el trabajo es colaborativo más que individual.

Tabla 1. *Respuestas Sección 2 (Condiciones de respuesta para el cambio), preguntas con indicadores opuestos*

Variable	μ [LIC, LSC]	σ
OTA – Organización de tiempo y actividades (-5 a +5) [P30]	3.00 [2.84,3.16]	3.01
AELD - Actividad educativa: lección-debate (-5 a +5) [P35]	0.17 [0.02,0.31]	2.61
AETI Actividad educativa: transmisora-interactiva (-5 a +5) [P36]	0.23 [0.08,0.37]	2.65
AEAS - Actividad educativa: asíncrona-síncrona (-5 a +5) [P37]	0.60 [0.47,0.73]	2.35
AEIC - Actividad educativa: individual-colaborativa (-5 a +5) [P38]	0.40 [0.25,0.55]	2.78

μ = valor medio, LIC límite inferior de confianza, LSC límite superior de confianza, σ desviación estándar

En la siguiente Tabla 2 podemos observar las medidas centrales de resultados de la sección 2, para las respuestas con escala del 1 al 10 (excepto la variable ICT, incremento en la carga de trabajo, que maneja porcentajes del 0% con incrementos del 10%). En la primera columna están las variables relacionada a las preguntas, en la segunda columna es la media y entre paréntesis cuadrados el límite inferior y superior con 95% de confianza, la tercera columna muestra la desviación estándar.

De acuerdo con los datos de la Tabla 2, las universidades fueron calificadas con 6.77 en promedio en cuanto a su preparación para el cambio (PU), números muy parecidos a los datos de preparación tecnológica (6.77) y pedagógica (6.91) de los maestros (PTM y PPM). La tecnología utilizada para el cambio parece apropiada (TAEE), con 7.45 en promedio. Así mismo, podemos observar que el incremento en la carga de trabajo (ICT) es bastante significativo (57%).

Tabla 2. *Respuestas Sección 2 (Condiciones de respuesta para el cambio), preguntas con valores numéricos*

Variable	μ [LIC, LSC]	σ
EPED - Experiencia previa en educación a distancia (1 a 10) [P10]	6.31 [6.18,6.43]	2.29
PU - Preparación de la universidad para cambiar a educación en línea (1 a 10) [P16]	6.77 [6.64,6.89]	2.22
TAEE - Tecnología adecuada para el entorno en línea (1 a 10) [P18]	7.45 [7.35,7.55]	1.85
PTM - Preparación tecnológica de los maestros (1 a 10) [P22]	6.77 [6.65,6.88]	2.12
PPM - Preparación pedagógica de los maestros (1 a 10) [P24]	6.91 [6.80,7.02]	2.01
ICT - Incremento en la carga de trabajo (% con base 0) [P28]	0.57 [0.56,0.59]	0.26
RDSC - Reproducción de las dinámicas del salón de clase (1 a 10) [P32]	5.81 [5.68,5.93]	2.32

μ = valor medio, LIC límite inferior de confianza, LSC límite superior de confianza, σ desviación estándar

La siguiente Tabla 3, presenta los resultados de la tercera sección del cuestionario, dirigida a entender el cambio de opinión en ciertos temas. Sobresale la alta calificación que se le da a extrañar las aulas físicas o clases presenciales (EAF, 8.56). Podemos ver también que los encuestados han incrementado sus competencias tecnológicas educativas (IIAT, 6.96) y su interés por estas tecnologías (ICTE, 7.38). Finalmente, también puntúan alto el grado de madurez digital de su universidad (GMEU, 7.36).

Tabla 3. *Respuestas Sección 3 (Cambio de opinión sobre tecnología y educación en línea), preguntas con valores numéricos*

Variable	μ [LIC, LSC]	σ
FEL - Facilidad de hacer educación en línea (1 al 10) [P48]	6.09 [5.96,6.21]	2.26
UEL - Utilidad de hacer educación en línea (1 a 10) [P49]	6.19 [6.06,6.32]	2.36
SEE - Sustentabilidad y viabilidad de la educación en línea (1 a 10) [P50]	5.89 [5.75,6.02]	2.38
CPEE - Cambio en la percepción sobre su experiencia educativa (1 a 10) [P56]	6.19 [6.05,6.32]	2.41
IIAT - Incremento en el interés en aprender tecnologías (1 a 10) [P58]	7.38 [7.24,7.52]	2.46
ICTE - Incremento en las competencias tecnológicas educativas (1 a 10) [P60]	6.96 [6.84,7.08]	2.17
ITFR - Intención de trabajar en el futuro de manera remota (1 a 10) [P69]	6.89 [6.75,7.02]	2.29
UEEU - Uso de educación en línea de la universidad (1 a 10) [P71]	6.33 [6.20,6.46]	2.29
UELE - Uso de la educación en línea de las escuelas (1 a 10) [P73]	6.15 [6.02,6.28]	2.35
EAF - Extrañar clases presenciales (1 a 10) [P75]	8.56 [8.44,8.68]	2.13
GMEU - Grado de madurez digital de la educación de la universidad (1 a 10) [P77]	7.36 [7.26,7.47]	1.88

μ = valor medio, LIC límite inferior de confianza, LSC límite superior de confianza, σ desviación estándar

3.2.1 Correlación entre variables

Los coeficientes de correlación Pearson entre variables más significativas encontradas (arriba de 0.50) son: PU con TAEE (.68), PTPM (.65) y PPM (.67) lo que significa que la preparación de la universidad para el cambio está relacionada con el uso de una tecnología adecuada para el entorno en línea, y la preparación pedagógica y tecnológica de los maestros. A su vez TAEE está relacionado con PTM (.59), PPM (.62) y la percepción del grado de madurez de la universidad, GMEU (.56). La percepción del grado de madurez de la universidad tiene a su vez correlación con PU (.58), TAEE (.56), PTM (.52) y PPM (.55). Asimismo, PTM y PPM tienen una fuerte correlación entre ellos (.78). La facilidad de hacer educación en línea (FEL) está relacionado con su utilidad (UEL, .56) y en el mismo grado con su sustentabilidad y viabilidad (SEE, .56). UELE, la percepción de la utilidad de la educación en línea de las escuelas, está fuertemente correlacionado con SEE (.73). UELE también tiene correlación con UEEU (.55) la utilidad de la educación en línea para la universidad. ICTE y IIAT tiene una correlación de .60, lo que significa que el incremento en el interés por aprender tecnologías educativas se relaciona con el aprendizaje que se ha tenido de dichas tecnologías. Por último, ITFR, la intención de trabajar en el futuro de manera remota presenta correlación con SEE (.51), UEEU (.62) y UELE (.58).

Cabe mencionar que no se encontraron correlaciones fuertes con el perfil de los encuestados (genero, edad, año que cursan y área de estudio) y el resto de las variables.

3.2.2 Redes causales

Para entender la causalidad entre variables, se realizó una estructura causal basada en el concepto de Pearl de d-separación (Hayduk et al., 2003), proceso que permite establece la direccionalidad de la correlación entre variables; es importante mencionar que dicho proceso permite un análisis puramente exploratorio (Kalisch et al., 2012), de manera que su uso debe reducirse a una interpretación tentativa. El algoritmo utilizado es el denominado PC basado en pruebas estadísticas de independencia condicional para identificar las relaciones de dependencia de las variables del modelo, a fin de obtener un grafo causal, el nivel de significancia para la prueba de independencia condicional utilizado fue $\alpha=0.01$.

Veamos primero la correlación entre variables separado por sección. En la Figura 2 se muestran las correlaciones entre variables, el grosor de la línea indica la fuerza de dicha correlación, en este caso todas las correlaciones son positivas. Las correlaciones más significativas son PU (preparación de la universidad) con TAEE (tecnologías adecuadas para el entorno en línea), y PPM (preparación pedagógica de los maestros) con PTM (preparación tecnológica de los maestros). PU a su vez presenta correlación con PPM y PPM, y en menor grado con EPED (experiencia previa en educación en línea) y RDSC (reproducción de las dinámicas del salón de clase). RDSC además presenta correlación con OTA (organización de tiempo y actividades), y es la única variable que tiene alguna correlación, aunque débil, con ICT (el incremento en la carga de trabajo).

En la Figura 3 se muestra la red causal para las variables de la sección 2, el grafo con direccionalidad. Podemos observar que al parecer PU tiene influencia sobre EPED y RDSC. RDSC se ve a su vez influenciado por varias variables (PU, OTA, EPED, PPM y PTM).

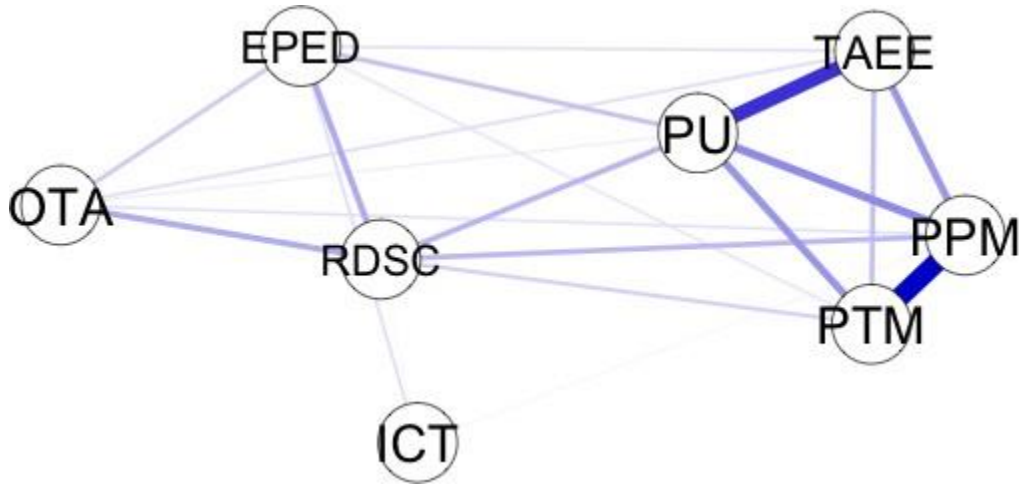


Figura 3: Correlaciones entre variables

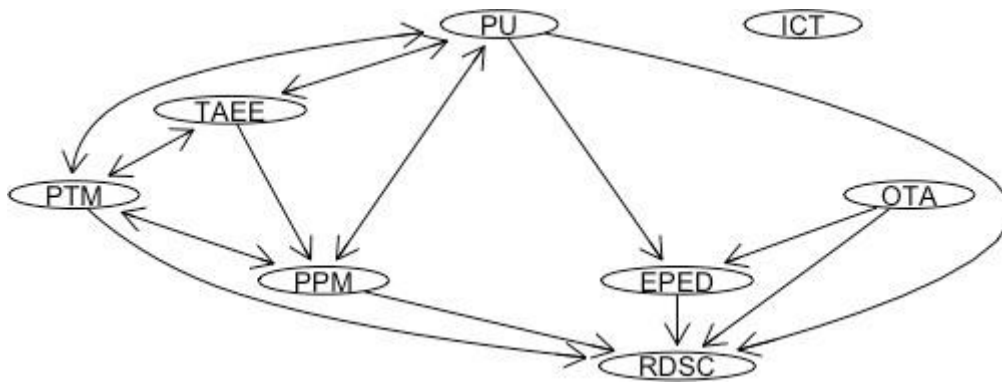


Figura 3: Red causal variables sección 2

En la Figura 4 se muestran las correlaciones entre variables de la sección 3. Una línea azul indica correlación positiva, mientras que una línea de color naranja indica correlación negativa. En este caso la variable UELE (uso de la educación en línea de las escuelas) no se incluyó ya que no pertenece al ecosistema universitario, de igual forma la variable EAF (extrañar clases presenciales) no fue incluida para tratar dar un enfoque de cara al futuro. Con respecto a la variable CFE (preferencia de modalidad para el futuro) [P64], como podemos observar en el Apéndice A, las posibles respuestas a esta pregunta son: presencial, a distancia, y mixto (*blended* o semipresencial). El tratamiento que se hizo a esta variable fue dicotómico en relación a la opción semipresencial, con la finalidad de entender qué variables tienen injerencia en la preferencia de continuar los estudios en el futuro con esta modalidad.

En esta Figura 4 podemos observar una fuerte correlación entre SEE (sostenibilidad y viabilidad de la educación en línea) y UEL (utilidad de hacer educación en línea), ITFR (intención de trabajar en el futuro de manera remota) y UEEU (el uso de educación en línea de la universidad), e ICTE (incremento en las competencias tecnológicas educativas) con IIAT (incremento en el interés de aprender tecnologías). Otras correlaciones fuertes son SEE con UEEU y FEL (facilidad de hacer educación en línea); UEL con FEL, e ICTE con CPEE (cambio en la percepción sobre la experiencia educativa) y CPEE con IIAT. También podemos observar una correlación negativa entre CFE y GMEU (la preferencia de continuar de manera semipresencial con el grado de madurez digital de la educación de la universidad). En la Figura 5 tenemos la red causal de las variables de la sección 3 del cuestionario. En esta figura llama la atención el número de variables que tienen influencia sobre GMEU (i.e. FEL, UEL, ICTE, IIAT e ITFR), y sobre UEEU (CFE, SEE, CPEE, UEL, ITFR e IIAT). Podemos ver que la preferencia de continuar con la modalidad semipresencial (CFE) tiene influencia sobre la SEE (sustentabilidad y viabilidad en la educación en línea) y UEEU (uso de la educación en línea de la universidad). ICTE (el incremento en las competencias tecnológicas educativas) tiene influencia sobre GMEU, IIAT e ITFR.

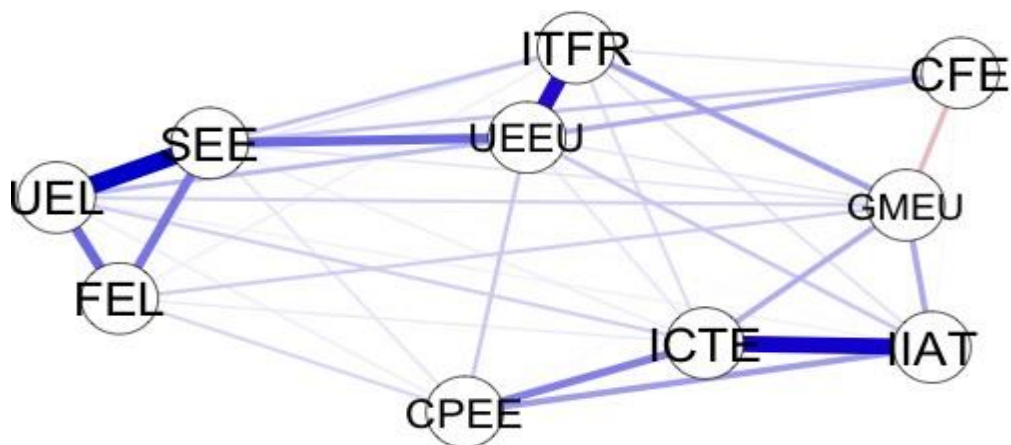


Figura 4: Correlaciones sección 3

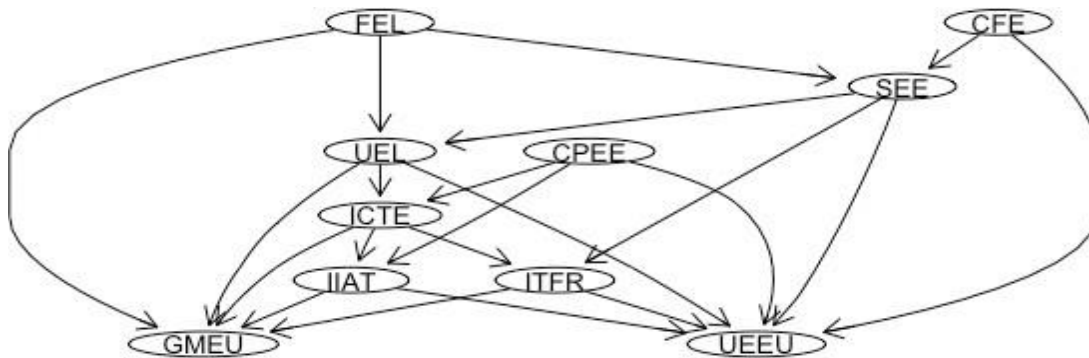


Figura 5: Red causal sección 3

Por último, en la Figura 6 se unieron las variables de la sección 2 y 3. En esta Figura 6 se marcó con un punto azul las variables de la sección 3 para diferenciarlas de la sección 2. Las correlaciones que llaman la atención es la de la variable OTA (organización de tiempos y actividades) que tiene correlación con varias variables de la sección 3 (UEL, SEE, UEEU e ITFR). CFE (preferencia por continuar de manera semipresencial en el futuro) tiene correlación negativa con la actual RDSC (reproducción de la dinámica de clases), y positiva con las variables EPED (experiencia previa en educación a distancia) e ICT (incremento en la carga de trabajo). GMEU tiene correlación con TAAE, PU, PPM, RDSC de la sección 2.

La Figura 7 muestra la red causal de las variables de las dos secciones, aquí podemos observar la interacción de direccionalidad entre secciones en, por ejemplo, la variable OTA que al parecer tiene influencia sobre ITFR y UEEU. También GMEU sobre PU y PPM, RDSC tiene influencia sobre GMEU, SEE y UEEU, y EPED sobre UEEU e ICTE.

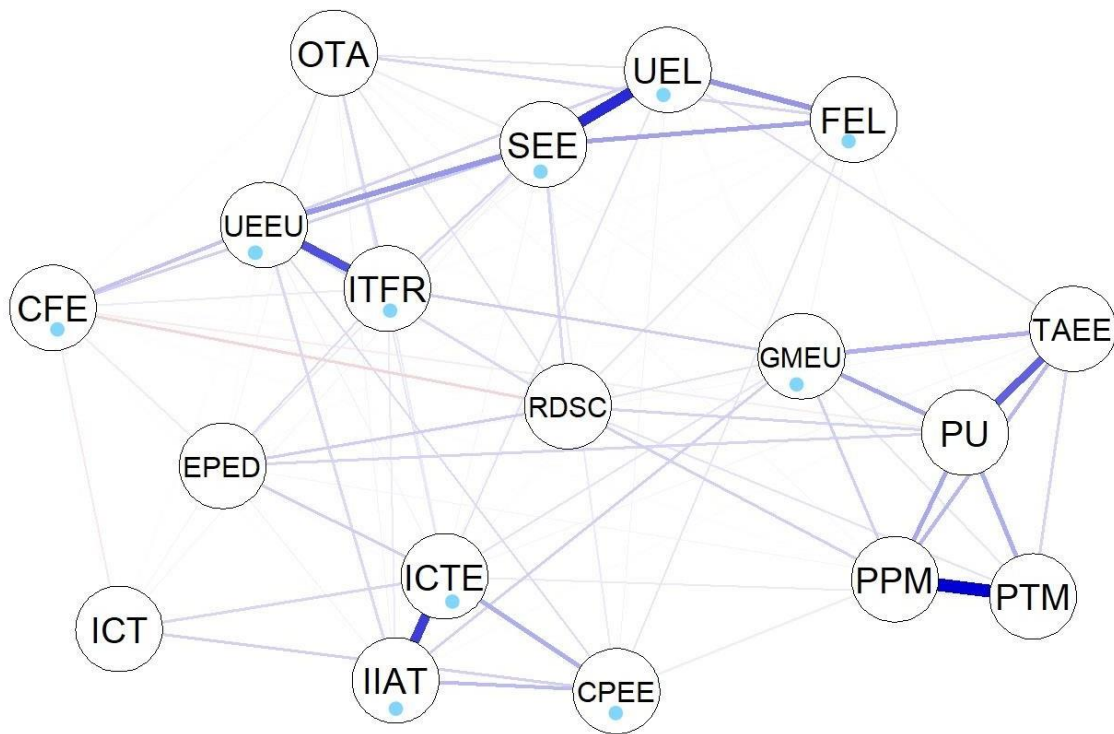


Figura 6: Correlación secciones 2 y 3

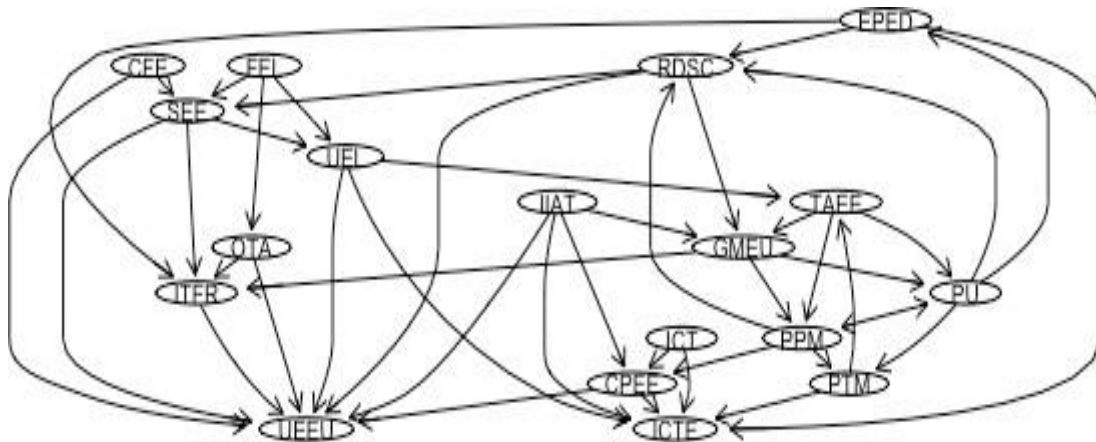


Figura 7: Red causal sección 2 y 3

3.3 Respuestas abiertas

En esta sección se describen algunas de las respuestas abiertas más significativas, es importante destacar que la variabilidad en dichas respuestas hace difícil establecer más similitudes. Los alumnos que seleccionaron clases semipresenciales, en su mayoría proponen llevar la teoría en línea, y las prácticas y laboratorios de manera presencial.

Entre los motivos para la organización del tiempo y actividades (OTA), que como se mencionó tuvo una puntuación alta, encontramos entre las respuestas [P31] que, para varios encuestados, no tener que trasladarse les ahorra tiempo que les ayuda en la organización de sus actividades, además algunos de ellos están utilizando plataformas digitales para organizarse. Sin embargo, hay varias quejas sobre la mala conectividad del internet, además las actividades de la casa que no siempre se compaginan con las actividades escolares, y aunado a esto hay problemas económicos que han provocado que algunos alumnos tengan que salir a buscar trabajo.

Los alumnos extrañan las clases presenciales (EAF) [P76] principalmente por la convivencia con sus amigos, el poder salir de casa, y para algunos de ellos es difícil el aprendizaje en línea.

Sobre la pregunta de cómo mejorar el aprendizaje en línea [P80], los encuestados sienten que tienen demasiada tarea, piden mejores plataformas, capacitación para los profesores, y clases más cortas, principalmente.

La última pregunta del cuestionario es ¿Qué consejos le darías a tus compañeros con respecto a la educación a distancia? [P81], aquí sobresalen respuestas de ánimo, echarle ganas, no desespérer, adaptarse, paciencia y no rendirse.

3.4 Limitaciones del estudio

A pesar de que el cuestionario se lanzó a nivel nacional, las respuestas representan apenas una pequeña parte de la diversidad socio-económica del país (ver sección 2.2 Participantes). Se podría decir que la región de estudio comprende apenas una sección del centro-occidente del país, y debemos considerar que las condiciones en esta parte son muy diferentes, por ejemplo, de la región sur en donde la pobreza tiene un fuerte impacto negativo tanto en educación como en tecnología (Martínez, y Gamboa, 2020).

Otro sesgo en los datos es el que se refiere al tipo de universidades (99% de los encuestados proviene de universidades públicas), además de que los posgrados prácticamente quedaron fuera del estudio (99% de los encuestados perteneces a la licenciatura).

Sin embargo, aun cuando es importante tener en cuenta estas consideraciones, el propósito del estudio mantiene su validez para entender el cambio de percepción sobre el uso de tecnologías para la educación.

3.5 Actitud de aceptación de tecnología

Haciendo un comparativo con el TAM, si consideramos la variable UEEU, es decir la respuesta de la pregunta sobre qué tanto las universidades deberían utilizar la educación en línea, como la actitud hacia el uso de la tecnología para la educación y de acuerdo con la red causal, podemos observar la gran cantidad de variables que la afectan (CPEE, IIAT, RDSC, UEL, OTA, ITFR, SEE, CFE); entre ellas está la pregunta directamente relacionada con el TAM sobre la percepción de utilidad (UEL). En cuanto a la variable relativa a la percepción de la facilidad (FEL), está no presenta relación directa con UEL, sin embargo, FEL si presenta influencia sobre UEL y SEE (la percepción de sostenibilidad y viabilidad de la educación en línea).

De igual forma la intención de uso se puede relacionar con la variable ITFR (intención de utilizar en el futuro las tecnologías de educación en línea). Esta variable se ve afectada por OTA, CFE y GMEU, que se refieren a la organización del tiempo, la preferencia por educación semipresencial y el grado de madurez de la universidad, en este caso OTA puede considerarse un factor individual, mientras que GMEU puede verse como una característica del sistema.

En resumen, podemos decir que el modelo TAM presenta variantes, aunque en términos generales aplica para determinar la intención de uso de las tecnologías. No obstante, debemos considerar que en este caso no estamos hablando de la intención de uso de una nueva tecnología desde el punto de vista individual, sino de todo un ecosistema que sustenta la educación en línea, que, por otra parte, debido a la pandemia ha tenido que implementarse aun cuando no hubiese la suficiente aceptación por parte de las instituciones y los usuarios.

De acuerdo con el MAETI, la madurez de la tecnología, y el contraste de fuerzas entre la innovación y la preservación, tienen influencia sobre la percepción de la tecnología en cuanto a su sustentabilidad viabilidad, lo que influye directamente sobre las intenciones futuras a nivel individual hacia las nuevas tecnologías. Lo que podemos ver reflejado en las respuestas de la encuesta mediante la influencia que tienen las variables SEE y GMEU sobre ITFR (ver Figura 7). Con lo que podemos concluir que profesores y alumnos consideran la madurez y la sustentabilidad de la tecnología como elementos que influyen directamente en las intenciones futuras de uso y capacitación de los usuarios.

De la variable CFE (preferencia por el modelo semipresencial), se esperaba encontrar causalidad. Sin embargo, por el contrario, esta preferencia tiene causalidad sobre SEE y UEEU, es decir, sobre la sustentabilidad y viabilidad de la educación en línea, aunado a recomendar la utilización de la educación a distancia por parte de la universidad.

Conclusiones

La pandemia ocasionada por el COVID-19 nos recluyó en casa, lo que tuvo un impacto directo en el uso de tecnologías para la educación. Estamos en una etapa muy temprana para conocer las consecuencias de cambiar del sistema presencial al entorno virtual de un día para otro, por lo que los estudios realizados en estos momentos son muy importantes para ayudar a comprender las consecuencias de este abrupto cambio.

En este documento se presenta el análisis de un cuestionario lanzado a nivel nacional a estudiantes universitarios en México, para el que recibimos más de mil respuestas de estudiantes de diferentes estados de la República. Como resultado del análisis de las respuestas se puede resaltar que:

- Los encuestados extrañan las clases presenciales, siendo la convivencia con sus compañeros uno de los principales motivos,
- A varios de los encuestados les es difícil comprender las lecciones mediante clases en entorno virtual.
- Los problemas socio-económicos han impactado en la calidad de la educación en línea.
- Se observa una mejora en cuanto a sus habilidades para manejar la tecnología.
- Evitar los traslados ha permitido que manejen mejor la organización de sus tiempos.
- La preparación de las universidades y sus profesores fue apropiada para realizar el cambio
- El grado de madurez de las universidades alcanzado por el cambio es adecuado.
- La aceptación de las tecnologías para la educación está relacionada con la percepción del usuario sobre su perspectiva de uso, así como su sustentabilidad y viabilidad. En un contexto en el que la madurez de dicha tecnología y el contraste entre preservación e innovación presentan fuerte influencia sobre dicha aceptación.

En México se tiene la expectativa de iniciar el siguiente semestre, alrededor de los meses de agosto y septiembre, de manera presencial. Se estudia la posibilidad de manejar algunas clases de manera virtual aprovechando el conocimiento adquirido y los materiales realizados en este periodo. Esto permitiría en un momento aumentar la matrícula que actualmente está sobrepasada por la demanda en varias carreras universitarias. Los efectos de la pandemia han sido devastadores para la sociedad, en contraste se ha impulsado el uso de las tecnologías. Entender los factores que promueven una actitud de los usuarios a involucrarse en su uso y el interés por mejorar sus habilidades, ayudará a consolidar su utilización.

Referencias

DAAD. (2020). Info & Services for Higher Education Institutions. Disponible en <https://www.daad.de/en/information-services-for-higher-education-institutions/centre-of-competence/covid-19-impact-on-international-higher-education-studies-and-forecasts/>

Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS quarterly*, 319-340.

Giovannella, C., y Passarelli, M. (2020). The effects of the Covid-19 pandemic seen through the lens of the Italian university teachers and the comparison with school teachers' perspective. *Interaction Design & Architecture(s) Journal, IxD&A*, 46, 2020, 120-136.

Giovannella, C., Passarelli, M., Alkhafaji, A. S., & Peña Pérez Negrón, A. (2021), A comparative study on the effects of the COVID-19 pandemic on three different national university learning ecosystems as bases to derive a Model for the Attitude to get Engaged in Technological Innovation (MAETI). *Interaction Design and Architecture(s) Journal - IxD&A*, N.47, 2020-21, pp. 167 – 190.

Hayduk, L., Cummings, G., Stratkotter, R., Nimmo, M., Grygoryev, K., Dosman, D., ... y Boadu, K. (2003). Pearl's D-separation: One more step into causal thinking. *Structural Equation Modeling*, 10(2), 289-311. https://doi.org/10.1207/S15328007SEM1002_8

Kalisch, M., Mächler, M., Colombo, D., Maathuis, M. H., y Bühlmann, P. (2012). Causal inference using graphical models with the R package pcalg. *Journal of Statistical Software*, 47(11), 1-26. <https://doi.org/10.3929/ethz-b-000049768>

Maldonado Gómez, G. M., de los Ángeles Miró, M., Stratta, A. E., Mendoza, A. B. M. A. B., & Zingaretti, L. (2020). LA educación superior en tiempos de Covid-19: análisis comparativo México-Argentina. *Revista de Investigación en Gestión Industrial, Ambiental, Seguridad y Salud en el Trabajo-GISST*, 2(2), 35-60.

Malo Álvarez, S., Maldonado-Maldonado, A., Gacel-Ávila, J., y Marmolejo, F. (2020). Impacto del COVID-19 en la educación superior en México. *Revista de Educación Superior en América Latina*, (8).

Martínez, Y. M., y Gamboa, L. A. A. (2020). Covid-19, Pobreza y Educación en Chiapas: Análisis a los Programas Educativos Emergentes. *Revista Internacional de Educación para la Justicia Social*, 9(3), 61-82.

Organización Mundial de la Salud. (2020). COVID-19: cronología de la actuación de la OMS. Disponible en <https://www.who.int/es/news/item/27-04-2020-who-timeline---covid-19>

Salloum, S. A., Alhamad, A. Q. M., Al-Emran, M., Monem, A. A., y Shaalan, K. (2019). Exploring students' acceptance of e-learning through the development of a comprehensive technology acceptance model. *IEEE Access*, 7, 128445-128462.

Sánchez Mendiola, M., Hernández, A., Torres, R., Carrasco, M., Romo, A., Mario, A., y Cazales, V. (2020). Retos educativos durante la pandemia de COVID-19: una encuesta a profesores de la

UNAM. Revista Digital Universitaria, 21(3). Disponible en <https://www.revista.unam.mx/wp-content/uploads/a12.pdf>

Secretaría de Educación Pública. (2020). Acciones de las Instituciones de Educación Superior durante la emergencia sanitaria. Disponible en <https://educacionsuperiordurantedecovid.anuies.mx/>

Villasís-Keever, M. Á., & García-Bolaños, C. (2020). <https://www.revista.unam.mx/wp-content/uploads/a12.pdf>. Revista Mexicana de Pediatría, 87(2), 43-45. Disponible en <https://www.revista.unam.mx/wp-content/uploads/a12.pdf>

Apéndice A

Yo y el aprendizaje a distancia en tiempos del Covid-19- cuestionario para estudiantes universitarios

Debido al encierro y el cierre de las universidades en los últimos meses, esta encuesta tiene como objetivo obtener tu opinión sobre algunos aspectos del aprendizaje a distancia.

La encuesta fue diseñada por ASLERD (the Association for Smart Learning Ecosystems and Regional Development) y es llevada a cabo por profesores de la Universidad de Guadalajara y el CIMAT, Zacatecas.

Agradecemos tu tiempo. En la medida de lo posible incluye tus comentarios explicando las preguntas de opción múltiple y de escala numérica.

Los resultados de esta encuesta serán públicos y ofrecerán una reflexión profunda sobre el tema de aprendizaje a distancia.

El cuestionario está compuesto de 4 secciones:

- 1) Información sobre el uso de los datos recolectados y aceptación
- 2) Preguntas generales
- 3) Actividades a distancia actuales
- 4) Tu opinión sobre el aprendizaje a distancia

El cuestionario toma 10 a 15 minutos si solo contestas las preguntas de opción múltiple, listados y escalas; y unos 20 a 30 minutos contestando las preguntas abiertas.

1) Este cuestionario es anónimo, no se piden personales. La información será utilizada únicamente con fines de investigación académica. Debes aceptar para continuar:

1. sí / no

2) Preguntas generales

2. Género

F / M / Prefiero no contestar

3. Edad

menor de 22 años / entre 22-26 años / entre 27-30 años / entre 31- 40 años / mayor de 41 años

4. Tipo de universidad en la que estudias

pública / privada / pública abierta / privada abierta

5. Nivel de estudios

universitario / maestría / doctorado / otro

6. Año que cursas

primero / segundo / tercero / cuarto / quinto / otro

7. Área a la que pertenecen tus estudios

Agricultura / Arquitectura y Diseño / Economía / Farmacia / Leyes / Ingeniería / Humanidades / Educación / Medicina o Salud / Veterinaria / Psicología / Matemáticas, Física, Química y Ciencias Naturales / Ciencias Políticas / Ciencias Sociales / Estadística / Computación, Informática / Otro: _____

8. Estado _____

9. Nombre de tu Universidad _____

3) Actividades actuales de aprendizaje a distancia

10. En escala de 1 (extremadamente limitado) a 10 (muy extenso), ¿Cómo evalúas tu experiencia previa (antes de la pandemia) en cuanto a aprendizaje a distancia?
11. ¿En qué dispositivos llevas a cabo tus actividades de educación a distancia (se pueden marcar varias)?
 - *Celular*
 - *Tablet*
 - *Computadora*
 - *Laptop*
12. ¿Cómo te conectas a Internet?

Fibra óptica ultra rápida / Fibra óptica / ADSL / Celular / Conexión satelital / Otro

13. Si has tenido dificultades para llevar a cabo las actividades de aprendizaje a distancia, ¿cuáles fueron las causas (se pueden marcar varias opciones)? *Selecciona todos los que correspondan.*
 - ninguna o poca disponibilidad de dispositivos (incluido si tienes que compartir con otros) dispositivos viejos u obsoletos
 - mala conexión (ancho de banda limitado) contrato con límite de datos
 - el ambiente familiar no es apropiado mis habilidades digitales son limitadas
 - uso de entornos virtuales/aplicaciones que desconozco uso de muchos entornos/aplicaciones
 - procedimientos muy complicados para participar en las actividades educativas extraño el pizarrón
 - ninguna o poca asistencia técnica (por parte de la universidad)
 - leyes de privacidad (por ejemplo, Ley Federal de Protección de Datos) dificultad en encontrar el material que tengo que estudiar
 - dificultad para comunicarme con el profesor
 - limitaciones para expresarme a través de canales en línea dificultades para concentrarme
 - problemas relacionados con la cámara (por ejemplo, mostrar un vídeo, hablar frente a la cámara)
 - otro
14. Si contestaste "otro" indica por favor qué dificultades adicionales has tenido, gracias.
15. Si contestaste " mis habilidades digitales son limitadas", por favor indica a qué habilidades te refieres, gracias.
16. En una escala del 1 (muy poco) al 10 (completamente), ¿Qué tan preparada estaba tu UNIVERSIDAD para reaccionar a la transferencia de actividades de educación en línea?
17. Si deseas explicar tu evaluación numérica hazlo en la siguiente línea
18. ¿Qué tanto las tecnologías (entornos virtuales) que utilizó la universidad fueron apropiados para enfrentar las necesidades del proceso de educación en línea? 1 (muy poco) a 10 (completamente)
19. Si deseas explicar tu evaluación numérica, hazlo en las siguientes líneas.
20. ¿Cuánto tiempo te tomó acostumbrarte al aprendizaje a distancia?
menos de 1 semana / entre 1 y 2 semanas / más de 2 semanas / todavía no me acostumbro
21. Si tu respuesta fue "todavía no me acostumbro", por favor indica las razones. Gracias
22. En tu opinión, ¿Qué tan preparados estaban tus MAESTROS, desde el punto de vista tecnológico, para transferir sus actividades educativas de presencial a distancia? 1 (muy poco) a 10 (completamente)
23. Si deseas explicar tu evaluación numérica, hazlo en las siguientes líneas.
24. ¿Hasta qué punto los MAESTROS de tu universidad, desde el punto de vista de educación, estaban preparados para transferir las actividades de presencial a distancia? 1 (muy poco) a 10 (completamente)
25. Si deseas explicar tu evaluación numérica, hazlo en las siguientes líneas.
26. En promedio ¿cuánto tiempo al día debes estar en línea tomando CLASES de aprendizaje a distancia?
menos de 30 minutos / entre 30 minutos y una hora / entre 1 y 2 horas / entre 2 y 4 horas / más de 4 horas al día
27. En promedio ¿cuánto tiempo al día estás en línea haciendo TODAS las actividades relacionadas con aprendizaje a distancia?
menos de 30 minutos / entre 30 minutos y 1 hora / entre 1 y 2 horas / entre 2 y 4 horas / entre 4 y 6 horas / entre 6 y 8 horas / más de 8 horas al día
28. ¿Crees que las actividades de aprendizaje a distancia han incrementado el trabajo que le dedicas a tu educación? Si es así ¿en qué porcentaje?
0% / 10% / 20% / 30% / 40% / 50% / 60% / 70% / 80% / 90% / 100%
29. Por favor explica por qué.
30. ¿Has podido organizar tu tiempo y actividades mejor o peor que antes? ¿Qué tanto en una escala de -5 (mucho peor) a + 5 5 (mucho mejor)?

31. Si deseas explicar tu evaluación numérica, hazlo en las siguientes líneas.
32. En tu opinión ¿qué tanto las actividades en línea reproducen la dinámica del salón de clases? (10 completamente, 1 para nada)
33. Si tu respuesta previa es menos de 6 ¿hasta qué punto piensas que esto es positivo y por qué?
34. ¿Existen actividades educativas, entre las que has tenido en este periodo, que consideres que no se pueden reproducir de manera presencial?
35. ¿En dónde colocarías las actividades educativas que atiendes en este periodo, en una escala entre dos extremos lecciones y debates? (-5 solo lecciones a +5 solo debates)
36. ¿En dónde colocarías las actividades educativas que atiendes en este periodo, en una escala entre dos extremos transmisiva e interactivas? (-5 solo transmisiva a +5 solo interactiva)
37. ¿En dónde colocarías las actividades educativas que atiendes en este periodo, en una escala entre dos extremos asíncrona e síncrona? (-5 solo asíncrona a +5 solo síncrona)
38. ¿En dónde colocarías las actividades educativas que atiendes en este periodo, en una escala entre dos extremos individual y colaborativa? (-5 solo individual a +5 solo colaborativa)
39. En las actividades educativas de este periodo, ¿cuáles son los propósitos para el uso de tecnologías? (se pueden seleccionar varios) Selecciona todos los que correspondan.
 - Producción de contenido
 - Compartir contenido
 - Dejar lecciones transmisiones (videoclips, películas, lecciones no interactivas)
 - Dejar lecciones interactivas (por ejemplo, webinars)
 - Asignar tareas/ejercicio para desarrollarse de forma asíncrona (por ejemplo, en casa)
 - Llevar a cabo ejercicios de manera síncrona (por ejemplo, en línea)
 - Comunicación síncrona y asíncrona con los maestros
 - Comunicación síncrona y asíncrona con personas interesadas localmente
 - Personalización/adaptación
 - Diversificación educacional y metodológica
 - Colaboración y trabajo en equipo Socialización
 - Planeación de procesos y actividades educativas
 - Manejo de procesos y actividades educativas
 - Evaluación
 - Auto-evaluación
 - Otra
40. Si tu respuesta fue “Otra”, por favor indica para que otros propósitos ha sido usada la tecnología.
41. ¿Qué métodos de evaluación se han adoptado? (se pueden seleccionar varios) Selecciona todos los que correspondan.
 - entrevista en línea
 - examen en línea
 - evaluación individual
 - para hacerse en casa trabajo individual
 - para presentarse en línea
 - trabajo en equipo para presentarse en línea
 - otro
42. Si tu respuesta fue “Otro”, por favor indica qué otros métodos han sido usados.
43. ¿Qué problemas encontraste durante las evaluaciones en línea y que se podría hacer para solucionarlos?
44. Describe por favor como ha cambiado la relación con tus maestros (si ha ido a mejor o a peor)
45. Por favor describe cómo ha cambiado la relación con tus compañeros (si ha ido a mejor o a peor)
46. En relación con otras partes interesadas locales, las relaciones se han *incrementado y mejorado en cuanto a número y calidad / decrementado*
47. ¿Podrías explicar por qué?

4) Tus opiniones sobre el actual aprendizaje a distancia

48. ¿Qué tan fácil es hacer aprendizaje a distancia? (1 nada, 10 completamente)
49. ¿Qué tan útil es hacer aprendizaje a distancia? (1 nada, 10 completamente)
50. ¿Qué tanto crees que la educación a distancia es sostenible y viable? (1 nada, 10 completamente)
51. Por favor explica tu evaluación numérica de la pregunta anterior

52. Después de esta experiencia, ¿para qué consideras son útiles las tecnologías con la finalidad de proveer una experiencia de educación de alta calidad? (es posible varias opciones) Selecciona todos los que correspondan.
- Producción de contenido
 - Compartir contenido
 - Dejar lecciones transmisiones (videoclips, películas, lecciones no interactivas)
 - Dejar lecciones interactivas (por ejemplo, webinars)
 - Asignar tareas/ejercicio para desarrollarse de forma asíncrona (por ejemplo, en casa)
 - Llevar a cabo ejercicios de manera síncrona (por ejemplo, en línea)
 - Comunicación síncrona y asíncrona con los maestros
 - Comunicación síncrona y asíncrona con personas interesadas localmente
 - Personalización/adaptación
 - Diversificación educacional y metodológica
 - Colaboración y trabajo en equipo
 - Socialización
 - Planeación de procesos y actividades educativas
 - Manejo de procesos y actividades educativas
 - Evaluación
 - Auto-evaluación
 - Otra
53. Si tu respuesta fue "Otra", por favor indica para que otros propósitos ha sido usada la tecnología.
54. Después de esta experiencia, en tu opinión la tecnología educativa permite mejorar: Selecciona todos los que correspondan.
- Eficacia del aprendizaje
 - Eficiencia del aprendizaje
 - Calidad en la experiencia de aprendizaje
 - Aprendizaje autónomo y auto-regulado
 - Desarrollo de sentido/pertenencia de comunidad
 - Desarrollo de interacciones
 - Desarrollo de una identidad digital
 - Diseño y manejo del proceso de aprendizaje
 - Otro
55. Si tu respuesta fue "Otro", por favor indica en qué otro aspecto la tecnología puede mejorar.
56. Después de esta experiencia, ¿qué tanto ha cambiado tu idea de sobre la experiencia educacional? (1 nada, 10 completamente)
57. ¿Podrías explicar cómo? Gracias
58. Después de esta experiencia, ¿qué tanto se ha incrementado tu interés en aprender tecnologías? (1 nada, 10 completamente)
59. ¿Podrías explicar por favor por qué?
60. Después de esta experiencia, ¿qué tanto se han incrementado tus competencias en tecnología educacional? (1 nada, 10 completamente)
61. ¿En qué área se han incrementado? (se pueden seleccionar varias) Selecciona todos los que correspondan.
- Gráficas digitales
 - Edición de vídeo digital
 - Edición de fotografía digital
 - Edición de sonido digital
 - Diseño e impresión 3D
 - Producción de contenido fuera de línea
 - Producción de contenido en línea
 - Búsqueda y filtros críticos de contenido digital
 - Descarga, organización e intercambio de contenido digital
 - Procesamiento y visualización de información Herramientas para comunicación digital
 - Entornos de trabajo en equipo
 - Productividad personal fuera de línea (por ejemplo, aplicaciones de oficina)

- Productividad personal en línea (por ejemplo, herramientas Google)
- Manejo de espacios personales digitales (entornos sociales, blogs, etc.)
- Entornos de clase virtual
- Herramientas digitales de evaluación
- E-portafolio
- Información de seguridad
- Ética digital
- Detección de plagio
- Manejo de procesos y diseño
- Tecnología de asistencia (por ejemplo, para discapacidades)
- Representación de conocimiento (por ejemplo, mapas, gráficos, etc.)
- Herramientas de juegos o gamificación
- Manejo y uso de hardware básico

62. Después de esta experiencia, ¿cuál es tu opinión sobre la fortaleza el aprendizaje a distancia?
63. Después de esta experiencia, ¿cuál es tu opinión sobre las debilidades del aprendizaje a distancia?
64. Después de esta experiencia, ¿en qué forma preferirías continuar con tus actividades educativas universitarias?

presencial / a distancia / mezclado (blended)

65. Si seleccionaste “mezclado”, ¿cómo imaginas la organización física de los espacios dedicados al aprendizaje?
66. Si seleccionaste “mezclado”, ¿cómo imaginas la organización física de los espacios dedicados al aprendizaje?
67. En particular, después de esta experiencia crees que es mejor realizar (Lecciones, Ejercicios/Prácticas, Revisiones, Exámenes escritos, Exámenes orales)

presencial /a distancia

68. Por favor explica las razones de tus elecciones
69. Después de esta experiencia, ¿qué tanto estás dispuesto a considerar trabajar en el futuro de manera remota? (1 nada, 10 totalmente)
70. ¿Deseas explicar el valor numérico de la pregunta anterior?
71. En tu opinión, ¿hasta qué punto deberían las universidades utilizar actividades de educación en línea? (1 nada, 10 completamente)
72. ¿Deseas explicar el valor numérico de la pregunta anterior?
73. En tu opinión, ¿hasta qué punto deberían las escuelas en general deberían utilizar actividades de educación en línea? (1 nada, 10 completamente)
74. ¿Deseas explicar el valor numérico de la pregunta anterior?
75. ¿Qué tanto extrañas tus clases presenciales? (1 nada, 10 completamente)
76. ¿Deseas explicar el valor numérico de la pregunta anterior?
77. En general, en una escala del 1 al 10, ¿cómo calificas el nivel de madurez digital de tu Universidad (infraestructura + equipo + habilidades + manejo + visión)?
78. ¿Deseas explicar el valor numérico de la pregunta anterior?
79. ¿Tuviste problemas, no relacionados con el cuestionario, que quieras comentar?
80. En tu opinión, ¿cómo se podría mejorar el proceso de aprendizaje en línea de tu Universidad?
81. ¿Qué consejo les darías a tus compañeros de clase sobre el aprendizaje a distancia?



Adriana Peña Pérez Negrón es doctora en Ingeniería de la Informática de la Universidad Politécnica de Madrid, España. Realizó una estancia investigación en la University of Salford, Reino Unido para la mención de “Doctor Europeo” Actualmente es profesor investigador en el Departamento de Ciencias Computacionales del CUCEI de la Universidad de Guadalajara, México. Es fundadora y actual Editor en jefe de la revista ReCIBE de la Universidad de Guadalajara. Es responsable del Cuerpo Académico Informática Aplicada del Programa para el Desarrollo Profesional Docente. Sus principales líneas de investigación son la Realidad Virtual y el estudio del comportamiento en medios digitales.



Carlo Giovannella se graduó en Física, trabajo durante mucho tiempo en el estado sólido físico como experto en sistemas complejos. Hoy día se considera un ‘Diseñador para la experiencia’: experto en la mejora de tecnología para el aprendizaje, diseño de la interacción, comunicación mediada por computadora, diseño y manejo de procesos, procesos y producción de innovación, monitoreo de ecosistemas y mercadotecnia. Desde 2015 es presidente de ASLERD. Fue director científico del Área de Industrias Creativas del Consorzio Roma Ricerche del 2013 al 2016. Es miembro del Departamento de Historia, Patrimonio Cultural, Educación y Sociedad de la University of Rome Tor Vergata, en el que es presidente del ISIM_garage (Interfaces and Multimodal Interactive Systems), un laboratorio de investigación dedicado al diseño y desarrollo de TEL-environment, herramientas y métodos para educación, diseño de interacción y más.



Aurora Espinoza Valdez es doctora en Ciencias Aplicadas del Instituto Potosino de Investigación Científica y Tecnológica, México. Actualmente es profesor investigador en el Departamento de Ciencias Computacionales del CUCEI de la Universidad de Guadalajara, México. Tiene el reconocimiento por el Sistema Nacional de Investigadores como Investigador Nivel I. Sus principales líneas de investigación son Biomatemáticas y aplicaciones de teoría de grafos.



Mirna Muñoz es Doctora en Informática por la Facultad de Informática de la Universidad Politécnica de Madrid, estancia posdoctoral en la Universidad Carlos III de Madrid. Actualmente es investigadora titular A del CIMAT- Unidad Zacatecas en Ingeniería de Software, tiene el reconocimiento por Sistema Nacional de Investigadores como Investigador Nivel I. Ha participado en la traducción oficial del libro CMMI-DEV v1.2 y 1.3, así como al perfil entrada ISO/IEC 29110 que desde mayo ha sido publicada como norma mexicana. Ha publicado diversos artículos técnicos en journals, capítulos Spriger link y el libro IGI global, así como en congresos nacionales e internacionales de alto impacto en su área



David Bonilla Carranza es doctorando del programa de Electrónica y Computación del CUCEI de la Universidad de Guadalajara. Hizo su maestría en Tecnologías de la Información en la Universidad de Guadalajara CUCEA. Actualmente es profesor del Departamento de Ciencias Computacionales de la Universidad de Guadalajara. Es miembro fundador de COPSIJAL (Colegio de Profesionistas en Sistemas de Informática de Jalisco). Es cofundador de la plataforma INNOVUS WEB para el desarrollo de sistemas web. Fundador de la comunidad Taban.mx dedicada a la enseñanza y creación de video juegos. Su principal interés de investigación es la mejora de procesos del software a través de video juegos.



Marcello Passarelli es doctor en Psicología, Antropología y Ciencias Cognitivas. Es investigador en ITD-CNR desde el 2017. Sus intereses de investigación incluyen medidas implícitas, modelado estadístico y conocimiento social, con fuerte enfoque cuantitativo. Durante su estancia en ITD-CNR, trabajó en el proyecto H2020 Gaming Horizons, enfocado en el impacto social de los vídeo juegos y la gamificación en el proyecto Erasmus+ CODUR para el aseguramiento de la calidad del e-learning en instituciones, en el proyecto Erasmus+ ENhANCE que diseñó un curriculum europeo de comunidades y familias de enfermeras.



Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-CompartirIgual 2.5 México.

UML to model pollution patterns of the Smart Cities for Metropolitan Zone of Guadalajara

UML para modelar patrones de contaminación de las Ciudades Inteligentes para la Zona Metropolitana de Guadalajara

Martha Patricia Martínez Vargas
martha.mvargas@academicos.udg.mx¹

Elsa Estrada Guzmán
elsa.estrada@academicos.udg.mx²

Roció Maciel Arellano
ma.maciel@academicos.udg.mx³

¹ Universidad de Guadalajara – CUCEA – Departamento de Sistemas de Información

² Universidad de Guadalajara – CUCEI – Departamento de Ciencias Computacionales

³ Universidad de Guadalajara – CUCEA – Centro de Innovación en Ciudades Inteligentes

Abstract: Pollution in dense populations such as the Metropolitan Zone of Guadalajara grows exponentially, affecting the health of citizens and reducing their Quality of Life. One of the main research challenges pursued with the study of Smart Cities is environmental pollution to improve the well-being of citizens and to protect natural areas. Therefore, it is urgent the development of Information Technology to reduce this problem by the scientific analysis of data, to classify the zones with greater contamination. Currently, these data are captured by constant monitoring stations, generating a large volume of information representing a challenge for the classification process. This work proposes a Unified Modeling Language (UML) model for the automatic execution of a classification algorithm using Support Vector Machine implementing libraries in python for parallel processing. As a result of this research, a UML model to identify patterns of pollution in Smart Cities, for example, the Metropolitan Zone of Guadalajara.

Keywords: Data Mining, UML, Support Vector Machines, Parallel Support Vector Machine Libraries, Internet of Things, Smart Cities.

Resumen: La contaminación en poblaciones densas como la Zona Metropolitana de Guadalajara crece exponencialmente, afectando la salud de los ciudadanos y reduciendo su Calidad de Vida. Uno de los principales retos de investigación que se persiguen con el estudio de las Smart Cities es la contaminación ambiental para mejorar el bienestar de los ciudadanos y proteger los espacios naturales. Por lo tanto, es urgente el desarrollo de Tecnologías de la Información para reducir este problema mediante el análisis científico de datos, para clasificar las zonas con mayor contaminación. Actualmente, estos datos son capturados por estaciones de monitoreo constante, generando un gran volumen de información que representa un desafío para el proceso de clasificación. Este trabajo propone un modelo UML (Lenguaje Unificado de Modelado) para la ejecución automática de un algoritmo de clasificación utilizando Máquinas de Vectores de Soporte con la implementación de librerías en python para procesamiento paralelo. Como resultado de esta investigación, se desarrolló un modelo UML para identificar patrones de contaminación en Ciudades Inteligentes, por ejemplo, la Zona Metropolitana de Guadalajara.

Palabras Clave: Minería de Datos, UML, Máquinas de Vectores de Soporte, Librerías en paralelo de Máquinas de Vectores de Soporte, Internet de las Cosas, Ciudades Inteligentes.

1. Introduction

With the emergence of the concept of Smart Cities, importance is given to the Quality of Life of citizens, promoting the paradigm of the use of Information Technology (IT) for environmental monitoring, scientific analysis, and with-it pollution conditions in urban areas. Guadalajara has an atmospheric monitoring system, and the information has been stored for 20 years. The stored contamination variables vary in the files, and in some cases, the data are missing or out of range. In addition, there is no automatic data analysis model that allows continuing the relocation of monitoring devices for the city in constant urban growth. Data Mining has provided techniques that are useful for the development of solutions, appropriate for the processing of information volumes with an exponential tendency, generating the Big Data phenomenon, such as Support Vector Machine (SVM) techniques used for classification.

The objective of this research is to propose a UML model by the design of an SVM parallelized algorithm with the cloud importing python libraries for the automatic classification of the elements of contamination data that yield the areas with the highest pollution index thus obtaining the position of the installation of new environmental sensors to install. The research of a UML method was carried out through the study of air quality models, the Internet of Things (IoT), and mining techniques. This article is part of a project where at this stage only the UML model is presented, which serves as the basis for the future execution of the proposed algorithms.

The organization of the paper is as follows. Section 2 presents an outline of the UML model. Section 3 gives an overview of the Smart Cities concepts and Quality of Life, pollution in big cities, the Mundial variables of quality of the environment and how is the monitoring environment of Smart Cities. Section 4 describes the problem of identifying pollution patrons in georeferenced maps, the static monitoring stations, the need for the identification of the best location of sensors for dynamic monitoring, also it is shown the challenges to managing the Big Data to analyze pollution variables. Section 5 presents the methodology and explains the implementation in detail of the UML model proposed. Also, it is presented data analysis techniques to identify patrons, SVM, parallelization with Python libraries to import, and the data variables to the UML Model. Finally, the parallel-proposed UML model. In Section 7, the conclusions.

2. Unified Modeling Language models to design technological proposals

Xinping *et al.* [1], proposed a Unified Modeling Language (UML) model to model the technology of virtual assembly systems to ensure the validity of the model system proposed. The model building of state diagram, use case diagram, sequence diagram, and deployment diagram is completed on the system, providing evaluation criteria and uncertain factors not only can be fully described by the system modeling method but also the effective solution for the control system can be acquired. They proposed a UML model to model virtual assembly technology that reduces operating costs to provide solutions to the problems of product assembly.

The University of Quebec [2], presented a model-driven development method that automatically generates detailed UML domain models from business process models expressed in the Business Process Model and Notation (BPMN). Their approach was based on business patterns consisting of three transformations: 1) extraction of the process value chain, 2) construction of the UML domain models, 3) elaboration of the UML models. They tested their approach on a set of business processes from the Enterprise Resource Planning literature. The results showed computational feasibility and the conceptual soundness of the approach.

Hongjie [3], proposed a model of a P2P file shared system based on UML. The P2P model has the function of finding and locating servers. The system adopted the centralized peer-to-peer network structure, with the central server initializing the work of mutual communication, category searching, and management services between the two end-users. The UML design stage consisted of a static model designed with a class diagram and a dynamic model designed in phase diagram and sequence diagram. The UML class diagram described the static relations among different classes as the structure of the information and the behavior of the system.

Martinez *et al.* [4], promoted the use of UML for modeling the extraction process of Big Data specifically. This work is an approach for modeling the Extract Transform Load (ETL) process in a data warehouse, for depicting the data conversion and the loader process involved in the extracted task. The use of UML was at a conceptual level modeled on five new stereotypes with UML diagrams. The benefits that UML diagrams brought to this system were the flexibility of replacing or modifying any unit according to the designer's requirements.

Besnard *et al.* [5], presented a UML model implementation by verifying and executing a robot soccer player of the MDETools'19 challenge problem with Embedded/Experimental Model Interpreter (EMI). EMI is a model interpreter that can be used to execute, simulate, and formally verify UML models on the host or embedded targets. This approach is integrated into (i) designing a UML model for the challenge problem, (ii) applying formal verification using model-checking on the design model, and (iii) executing this model to participate in the challenge.

Guerriero *et al.* [6], proposed the use of the UML model to model developing streaming applications. They presented StreamGem App, a model-driven tool aiming at simplifying the development of distributed streaming applications. The UML model allowed to model streaming-specific concepts through UML Class Diagrams and a model-to-text transformation to automatically generate the application code starting from the UML models.

Bagherzadeh *et al.* [7], presented a tool that supports live modeling of UML for Real-Time (UML-RT) that is a language for the modeling of soft real-time systems. They developed a conceptual framework for live modeling in the context of model execution by code generation that is independent of live programming services.

Besnard *et al.* [8], developed a Model-Driven Reverse Engineering (MDRE) framework named Source to Model Framework (Src2MoF). This framework was developed to generate UML structural (class) and behavioral (activity) diagrams from the Java source code. In this project, the UML model was a central part of the design phase to verify the design of the system early in the development process.

Huixia and Ke [9], proposed a modeling method of fault diagnostic system software based on UML for the standard IEEE1232 which is also named AI-ESTATE. The AI-ESTATE services are portability of diagnostic knowledge and enable the consistent exchange and integration of diagnostic capability. This work analyzed the function of AI-ESTATE reasoner services in a fault diagnostic system software. The UML class diagrams, and sequence diagrams of the fault diagnostic system management services are set up. A modeling method for AI-ESTATE diagnostic services with UML is better to embody the AI-ESTATE architecture and construct the foundation for implementation of a fault diagnostic software system.

Tudoroiu *et al.* [10], presented a proposal of an intelligent neural-fuzzy modeling strategy based on UML diagrams and Petri nets models of the pollution sources impact on the air quality along the Romanian coast of the Black Sea, especially in the Constanta vicinity. They monitor the physical and chemical parameters of air quality, such as temperature, wind speed, Carbon Dioxide (CO₂), methane (CH₄), Nitrogen Oxide, ozone, water vapors concentrations. This work is focused on determining the possibility of applying Fuzzy UML and Petri Nets architectures for simulation and prediction of the performance of air quality of the Constanta Black Sea resort city environment.

In general, the presented works implement the UML model to modeling systems proposal and the validation of the prototypes. But in other areas as a UML model to model the technology of virtual assembly systems, UML domain models from business process models, a P2P file shared system based on UML, UML for modeling the extraction process of Big Data, a UML model implementation by verifying and executing a robot soccer player, a UML model to model developing streaming applications, a tool that supports live modeling of UML for Real-Time, a Model-Driven Reverse Engineering framework, a modeling method of fault diagnostic system software based on UML, and an intelligent neural-fuzzy modeling strategy based on UML diagrams and Petri nets models of the pollution sources impact on the air quality. Our proposal is focused on a UML model by the design of an SVM parallelized algorithm with the cloud importing of Python libraries for the automatic classification of the elements of contamination data that yield the areas with the highest pollution index according to an established variable and thus obtain the position of the installation of new environmental sensors to install.

3. Smart Cities and Quality of Life

Townsend [11], refers to a Smart City "Smart Cities" as one that solves problems by monitoring events and resources using sensors installed throughout the city. Generating, in this activity, large volumes of data, which when subjected to their analysis it is possible to observe patterns that allow visualizing alternative solutions. Smart Cities assume the challenge of generating strategies to solve problems of sustainability and Quality of Life, through IT.

In a press release [12] of 2015, the Organization for Economic Co-operation and Development (OECD) and the National Institute of Statistics and Geography (INEGI) published a report on the most relevant indicators that influence the well-being of Mexicans Quality of Life. These were grouped into 12 dimensions, one of them being the environment. One of the challenges of environmental monitoring is to guarantee the Quality of Life of citizens. Air pollution records for the year 2012, in cities and rural areas, has caused 3 million premature deaths each year throughout the year, due to exposure to small particles of 10 microns in diameter (PM₁₀), which can lead to heart disease, cancer, and pneumonia. Most sources of air pollution are out of the control of people and require action by cities as well as national bodies in sectors such as construction, agriculture, transport, etc.

3.1 Mundial Variables of Quality of Environment

In [13] mentions the guidelines that apply worldwide and are based on expert assessment of scientific evidence concerning, Particles (PM₁₀), ozone (O₃), nitrogen dioxide (NO₂), and sulfur dioxide (SO₂). Particles that cause the most health damage are those with 10 microns in diameter if they are lodged inside the lungs. Measurements of air quality warn of annual concentrations of PM₁₀ particles per cubic meter (m³) of air. O₃ at ground level is one of the main components of the toxic mist. The highest ozone levels are recorded in periods of sunny weather. Nitrogen dioxide in concentrations greater than 200 mg/m³ is a toxic gas that mainly affects the airways. The principal sources of this gas are combustion processes. SO₂, colorless gas with penetrating odor generated from fossil fuel combustion (coal and petroleum). In combination with water, it is transformed into sulfuric acid, being the main component of acid rain that causes deforestation.

3.2 Monitoring Environment of Smart Cities

The development of IoT: sensor development, infrastructure, communication networks, analytics, and advanced technology, allows the construction of data collection networks in cities to obtain information on air quality and subsequently raise awareness among citizens on pollution reduction, thus IoT can help to maintain better air quality for all citizens. Within a city, one of the most complicated parameters is air quality. The IoT is translated from a change from reactive systems to proactive systems and it has contributed to inserting objects into a dynamic world of Information Technology. The citizen-led Oxford Flood Network has developed a system of sensors to gather data on water levels from stream and groundwater sources and anticipate flood incidents. The system relies on sensors that use ultrasonic rangefinders, positioned above local waterways, to measure changes in the water level and indicate flood risks [14]. Some companies use sensors to determine when a car is parked in each spot. This data can be sent directly to drivers, helping to guide them more quickly to an available location. The companies can use the data to adjust pricing based on demand [14].

Towards the challenges of urbanization, the technology offers a solution to optimize processes and services in the cities in what is called the implantation of an Intelligent City. Sensor networks, data repositories, and data analysis are current processes that must be managed as part of the city's intelligent infrastructure, where complexity, resilience, and interoperability are important issues to consider. The information of all processes in a city must be connected through a sensor network to communicate and manage the control and optimization of the water supply. For example, if a house breaks and there is not a sufficient water supply, it is necessary to restore the service as soon as possible. In doing so, it should be known how much water should be allocated to water a garden in a portion of the city, which areas need to be watered first among other things [15]. One of the most important challenges is to ensure the good quality of the air; in this respect, the data obtained through a network of sensors begin to play an important role in measuring and managing air quality. By installing a citywide data collection network, cities can implement smart actions after data analysis to help ensure cleaner air for all [16]. IoT signaling and sending data to each other consume a lot of power and CPU. As mentioned in [17], methods for monitoring air pollution made use of complex, expensive, and fixed equipment in one place. This leads to limitations on IT collecting data and access to data. This process is changing with the use of IoT, using low-cost sensors, easy to install, and obtaining data in real-time. This provides valuable tools to perform data analysis, providing air quality monitoring information to citizens. In [18] proposed to locate or relocate sensors through hot zones that are a perimeter with significant data related to the observed phenomenon. These works propound the installation of alternatively fixed sensors determined by the objects of study in contrast to our proposal with a different approach where the most appropriate location is not known, or the translocation of the sensors determined by the UML model applying SVM algorithms.

4. The Problem of Identification of Pollution Patrons in Georeferenced Maps

This work identified three problems in the identification of pollution patrons in georeferenced maps as the identification of the best locations of sensors, the management of Big Data to analyze pollution variables, and the identification of patrons of behavior in georeferenced maps through data years.

4.1 The Static Monitoring Stations

The Metropolitan Zone of Guadalajara (MZG) is formed by eight municipalities of the Jalisco state as Guadalajara, Zapopan, San Pedro Tlaquepaque, Tlajomulco de Zuñiga, Tonalá, El Salto, Ixtlahuacán de Los Membrillos, and Juanacatlán [19]. With a total of territorial superficies 2,384.66 Km² and a total population of 4,796,603 estimations of the 2015 year from INEGI. MZG continuously registers periods of unhealthy levels of air quality [20]. In the MZG exist eight static monitoring stations located in Las Águilas, Atemajac, Centro, Loma Dorada, Miravalle, Oblatos, Tlaquepaque, and Vallarta. Integrating the Automatic Network of Atmospheric Monitoring of the MZG of the state government of Jalisco. These monitoring stations determine the zones of greatest pollution with measurements every 10 minutes 24 hours a day, and 365 days a year [21] with static monitoring stations. But the MZG has a constant increase in pollutants, therefore, predominant the identification of zones with a greater increase in pollution. By identifying the causes of pollution more accurately better actions could be taken to reduce the agents of contamination consequently the Quality of Life.

4.2 The Need to Identify the Best Location of Sensors for a Dynamic Monitoring

Currently, there is no classification of the most heavily polluted areas of MZG, and the geographic direction of the impact of pollution is unknown. Likewise, few Information Technology tools collaborate in the analysis of information to support the classification of territorial areas with greater pollution. For this reason, it is required the proposed UML model by the design of an SVM parallelized algorithm with the cloud importing Python libraries. This implementation of Data Mining techniques allows the management of large volumes of records for georeferenced analysis by which environmental behavior patterns are identified in the MZG. With this analysis, it is possible to identify the coordinates of greater risk of contamination to establish a georeferenced map of the location of sensors, thus achieving a better targeting of the factors that generate pollution. So far, the monitoring stations are not enough to make a thorough evaluation of the phenomenon. The correct location of the sensors has been an obstacle to starting the projects of installation of the infrastructure of IoT for the environment level at MZG. The sensors should not be fixed due to the constant changes of a city, for example, increased traffic, urbanization, etc.

For this reason, it is foreseen to require the analysis of the data of the same sensors installed providing their new relocation, operating based on the coordinates suggested by the patterns identified by the sensors themselves. The idea with the UML model presented is that in the areas with the greatest contamination the location of new sensors is determined. This would imply locating more sensors to determine with greater precision factors that generate contamination. Figure 1 presents an example of the usefulness of the proposed UML model and the results of the application, the Monitoring Stations of MZG are represented by green, yellow, and red circles, representing the location of the current quality monitoring sensors. The color of the circles is determined by the quality of the air, green circles represent satisfactory levels of air quality, yellow circles represent a warning, and red circles represent poor air quality. In the red circles, there is a lower quality of the air near them; when a low-quality variation in the air is detected, the location of new sensors will be installed and represented by blue circles to identify with more precision the factors that produce pollution, and these new locations will be calculated by the UML proposed model.

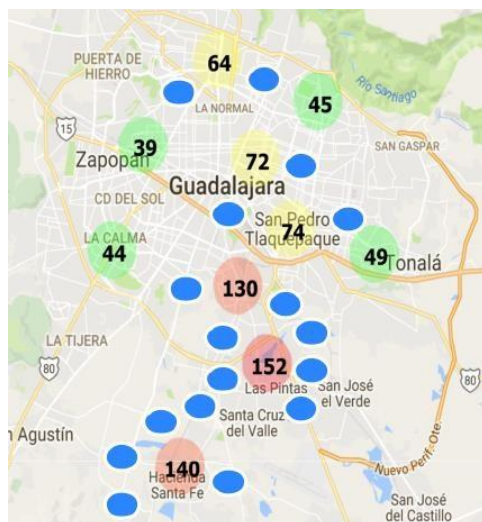


Figure 1. Monitoring Stations of MZG⁴.

4.3 The Challenges to Manage the Big Data to Analyze Pollution Variables

The management and analysis of large amounts of information present several challenges such as data cleaning. The data obtained by the monitoring stations show heterogeneity, making the standardization process difficult to analyze. The data were obtained from the Jalisco Atmospheric Monitoring System (SIMAJ). This data represents around 20 years of monitoring from 1996 to 2015, starting with 8 monitoring stations and in the 2012 year, 2 additional stations were added. The records presented were from every hour for each station. The challenges of information integration are mainly data cleansing, like the loss or absence of information, out of range, blanks, text instead of numbers. Another problem of the data is that there does not exist a homogeneous format of the acquisition and the storage of the information, in this way increasing the challenge of integrating the information. Until 2011, the data from all stations were presented in an Excel file. Subsequently, the data was generated in an Excel file for the monitoring of each station. This means contemplating various processing designs with a variety of formats. As an example, some records do not have the same 24-hour format.

Another difference presented was different variables have been handled per monitoring station. The World Health Organization (WHO) proposed an environmental quality model. The model defines which variables should be measured and in which ranges air quality should be measured. On the other hand, the data do not have standardized variables nor the number of variables, this complicates further the cleaning of the data.

5. Methodology

Literature concerning the pollution problem of the city of Guadalajara, national, and international models of air quality were initially reviewed. Likewise, the official SIMAJ Web site was visited to study the atmospheric monitoring system currently used for the application of contingency plans, which includes the infrastructure for daily operation and data management. This led to the verification of their files that are available and open at the site already mentioned above, totaling 878,988 records. In the future, it is expected an increased emission of catches multiplying the files by the proliferation of sensors. Therefore, more sensors and more files will be generated.

⁴ Source: <http://siga.jalisco.gob.mx/aire/>

Then, classification strategies were investigated for high and low volumes of data, specifically dedicated to the delimitation of georeferenced borders to be able to discover zones of atmospheric contingency. This led to the selection and study of SVM for its benefits in solving these problems from its parallel approach, which takes advantage of using the python language and its NumPy, SciPy, MPI4py, and Sklearn.SVM libraries combined with associated technology such as NVIDIA Graphics Processing Unit (GPU) microprocessors and INTEL as an operating platform.

Some references for the modeling specification of infrastructures specifically for the location of IoT sensors in geographic territories were scrutinized. The objective of this research is to generate a proposal through the design of a UML model for the automatic and dynamic relocation of constant monitoring sensors, using the results of patterns found by the design of an SVM algorithm, and in turn analyze the data coming from the stations (or previous sensors), to improve the atmospheric monitoring schemes for impact on the form of prevention to the citizen and on the detection of contingencies that force the changes of policies that regulate habits of not contaminate and care for the environment.

5.1 Data Analysis to Identify Patrons

In principle, Data Mining is a process of pattern discovery [22]. This premise was raised in this way because of its analogy with mining since a series of stages are carried out to achieve the economic exploitation of minerals such as extraction, smelting, processing, refinement, and transport. Mainly, the handling of large volumes of geological material to be revised. This need also appears in data analysis, where filtering or cleaning, sorting, and presentation of the data becomes more vital, and is better known as Extraction, Transform, and Load (ETL). The IoT boom has accelerated research on the analysis of high volumes of information caused by the high speed of information capture by the monitoring sensors, a factor that has influenced the appearance of the term Big Data [23]. Different studies have been proposed to solve problems concerning these topics, in Table 1. Data Analysis Techniques are presented. Several of the techniques are parallelized and implemented by GPU, the GPU technology is required to speed up the operation of applications, and it is an open-source architecture. The parallel SVM algorithm was recommended in this model as the analysis technique. It reduces the training time significantly compared with the Sequential Minimal Optimization (SMO) algorithm while maintaining a high level of accuracy in classifications [29]. Godoy [30] presented a literature review of scientific journals published in the years 2010 and 2011, with the aim of identifying the main machine learning techniques utilized for text mining. The main technique used by the studied author was SVM with 22.72% of incidents. SVM is a method that achieves high degrees of correct classification in various types of applications.

Table 1. Data Analysis Technique

Data analysis techniques	Target	Open-source
Implementation of Genetic algorithm training of artificial neural networks which achieves an order-of-magnitude speedup over a sequential algorithm under Compute Unified Device Architecture (CUDA) as a platform. Use GPU architecture.	CUDA-accelerated genetic feedforward-ANN training for Data Mining [24]. (Parallelized, GPU)	Yes
R-CUDA implements a Bayesian multilevel model. R-CUDA is only configured for Linux systems with Nvidia CUDA support.	The volume of the data to be analyzed for magnetic resonance imaging. (Parallelized, GPU) [25].	Yes
Classification: K-Nearest Neighbors (KNN), Decision Trees, SVM, Neural Networks, Gradient Boosted Decision Trees (GDBT) Clustering: K-Means, dendrogram, Graph Mining.	An overview of the most prominent tasks and methods that form the basis of Data Mining. (No Parallelized, No GPU) [26].	Yes
Locality Sensitive Hash with Nearest Neighbor Search (LSH-NN).	c-approximate r-nearest neighbor the search algorithm for CUDA. (Parallelized, GPU) [27].	Yes
SVM.	Hyperspectral Image Classification. Implements OpenMP and CUDA. (Parallelized, GPU) [28].	Yes
Parallel SVM.	A Resource Aware MapReduce for Large Scale Image Classifications. Implements OpenMP and CUDA. (Parallelized, GPU) [29].	Yes

5.2 Support Vector Machines

SVM is a set of algorithms of supervised learning models with associated learning that analyze data used for classification and regression analysis. Development by Vladimir Vapnik and his work team in the AT&T Laboratory in 1979 [31]. The SVM algorithm consists of two phases: training and test. In the training phase, the data are used to build a math model to classify. A set of training data is used consisting of pairs; one represents the correct inputs and the other the correct outputs. In this way, the system learns the correct output mapping for each input pattern presented [30]. In the test phase, the math model is used to predict a new class and classify points [32]. SVM techniques are applied in diverse tasks of classification such as image recognition, bioinformatics, and text processing [33]. SVM also gives good generalization performance of problems such as handwritten character recognition, face detection, pedestrian detection, and text categorization [34].

6.2 Parallel Support Vector Machines with Python

Python is recommended as a programming language in the UML proposed model since it is supported by an extremely large variety of packages. Besides being an open-source language, Python includes an Integrated Development Environment (IDE) that allows fast prototyping and profiling projects [35] for the source code editor. Python provides build automation tools and a debugger. Another technology recommended in the UML model is the implementation of Python parallelism through CUDA technology. CUDA is a parallel computing platform and programming model that enables dramatic increases in computing performance by harnessing the power of GPU to deliver the benefits of GPU acceleration. CUDA Python implements the NumbaPro Python compiler that integrates the Anaconda Accelerate package. The benefits of these technologies are the speed of a compiled language targeting CPU and GPU [36].

The UML model also integrates some libraries that ease in python the analysis of data. The libraries are NumbaPro, NumPy, SciPy, Mpi4py, and Sklearn.SVM. Python as an interpreted language is considered too slow for high-performance computing, but with the NumbaPro library, the performance has increased for data analytics; moreover, CUDA Python implements the NumbaPro because the Python compiler is part of the Anaconda Accelerate package that brings rapid interactive development and the speed of a compiled language [37] [35]. NumPy is the fundamental package for scientific computing in Python. This library provides an N-dimensional array object, and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulations, sorting, selection, input/output data, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation [38]. SciPy library is a fundamental library for scientific computing, it provides user-friendly and efficient numerical routines such as routines for numerical integration and optimization [39]. The Mpi4py library supports communication and provides Python bindings for the Message Passing Interface [40]. The Sklearn.SVM library includes the GridSearchCV object to parallelize jobs and finds the best kernel function that approaches the hyperplane that classifies the data.

5.3 Data Variables to the UML Model Proposed

The files produced by the monitoring stations had several variables of contamination but for this study, there were some relevant variables such as ID, DATE, TIME, PM₁₀, LATITUDE, LONGITUDE, and STATION. These variables are the entry to the SVM to classify the data in a range of contamination and determine the zones with higher contamination.

5.4 The Parallel Proposed Model

As a result of the analysis of the problem, a model is proposed to implement an automated system that carries out the activities of data analysis through SVM to identify the areas that require the placement of new sensors for better territorial monitoring. The model is described by the architectural, behavioral, and deployment view through UML class diagrams and activity diagrams. Figure 2 Block diagram air pollution monitoring system with IoT shows the deployment view of the automated system of air pollution monitoring using the IoT. In conjunction with the detection of data in the current stations, integrating an open architecture with heterogeneous devices constitutes a modular system. In the monitoring stations, there will be a network of sensors will be integrated with a microcontroller under an embedded software platform, the sensors capture the quality of the air produced in real-time, the data in continuous sensing are obtained by the microcontroller and sent to the server where there would be a database that would store the collected data. In this UML model, the first action is the control of data cleaning, then the data analysis with vector machines and the derivation of processes in parallel to the working modes of the cloud; finally, the UML model returns the best kernel function (radial basis or polynomial function).

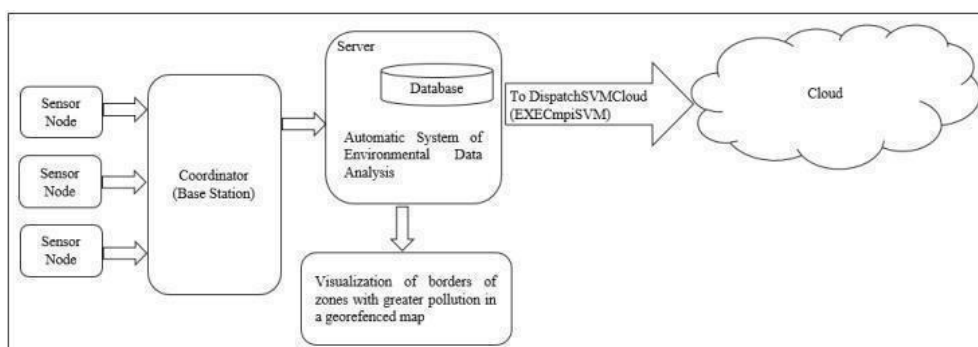


Figure 2. Block diagram air pollution monitoring system with IoT.

To automate the process of data analysis of the environment, it was observed that a data analysis algorithm for Smart Cities metrics operates as a domain coupled to an information context or workspace and processes in the form of elements of control, which are: configuration for storage, filtering (or cleaning) and form of data extraction, the configuration of SVM environment variables for their cloud parallelization and visualization. See figure 3, the class diagram that describes the architecture of the proposed model to Smart Cities automated system for analysis of environmental data.

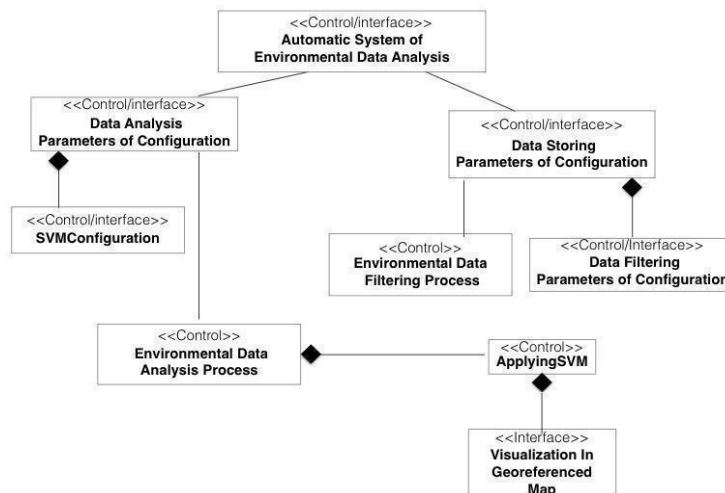


Figure 3. Class diagram describing the architecture of the proposed UML model in Smart Cities to automate the analysis of environmental data.

In this class diagram, it is presented the labeled elements <<Control/Interface>>. The <<Control/Interface>> indicates that the methods of the class are dedicated to the capture of configuration variables from a Graphical User Interface to its storage control. The Interface class only displays the results of the analysis, the control class is used to trigger internal process methods when it is called. The Automatic System class of Environmental Data Analysis is the main class, it displays a menu that contains two options:

A) Updating the data analysis parameters of the environment or Data Analysis Parameters of Configuration class. The class consists of attributes that define the variables for the determination of ranges of values for the measurement of air quality, information of sensors or machines for processing in the cloud, etc. This in turn has a composite association represented by a diamond that means a strong membership of the SVM configuration class. It has a second association with the Environmental Data Analysis Process class to trigger it according to the value of the frequency parameter, which in turn triggers ApplyingSVM which requires the list of machines available for parallel execution and calls the Visualization in class Georeferenced Map for displaying the data.

B) Updating the parameters for storage named Data Storing. Parameters of Configuration associated with the filtering and storage configuration classes. It relates to the filtering process to trigger it according to the value of the cleaning frequency parameter. The internal design for the Classes Automatic System of Environmental Data Analysis, Data Analysis Parameters of Configuration and Data Storing Parameters of Configuration are shown in figure 4 Class design for the main class, the design of the classes Environmental Data Analysis Process, SVMConfiguration, ApplyingSVM, and Visualization in Georeferenced Map. In figure 5 Class design for classes involved in scientific data analysis, figure 6 Class design for the classes involved in data cleaning for analysis prepared for the Environmental Data Classes Filtering Process and Data Filtering Parameters of Configuration.

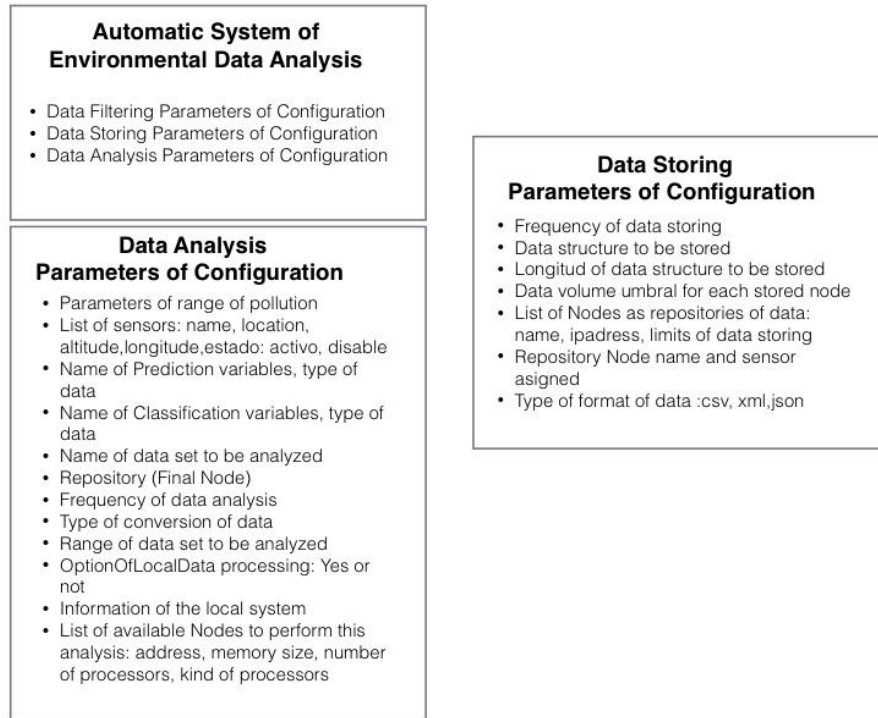


Figure 4. Class design for the main class.

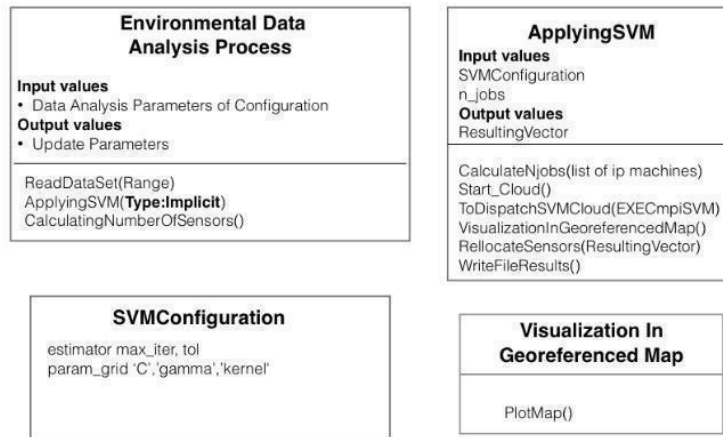


Figure 5. Class design for classes involved in scientific data analysis.

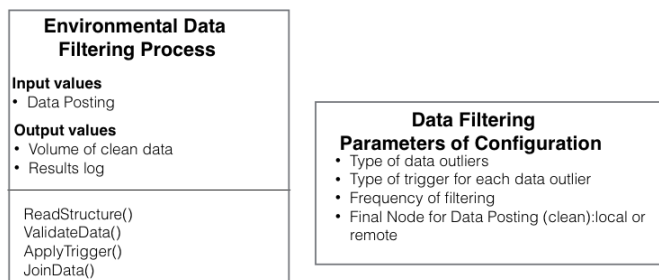


Figure 6. Class design for the classes involved in data cleaning for analysis preparation.

In the behavioral view, it is indispensable to show the algorithm EXECmpiSVM triggered by the ToDispatchSVMCloud method belonging to the ApplyingSVM class. figure 7 shows the Diagram of the transition of activities of the behavioral view of the algorithm parallelized for SVM for data analysis of the environment.

The design of the proposed UML model requires the Python libraries Mpi4py, Numpy, and Sklearn.SVM. In the configuration, the parameters, param_grid, and the data set of the environment data are obtained at runtime from the environment of the instantiated configuration objects explained above. The call to the GridSearchCV must be designed in parallel according to the number of concurrent jobs that must be distributed in the cloud to get the best kernel function that fits the data for the training process and the number of sensors. Process rank 0, is the principal of the concurrent process as the following instruction `MPI.COMM_WORLD.Get_rank()==0` in this case, it is proposed that it be designed to collect the results of the threads (example `rank==1` radial basis, `rank==2` polynomial kernel function, etc.). Finally, the proposed UML model uses several functions to Support Vector Machines referring to the divided zones to locate the territorial routes of installation of the new sensors for the IoT.

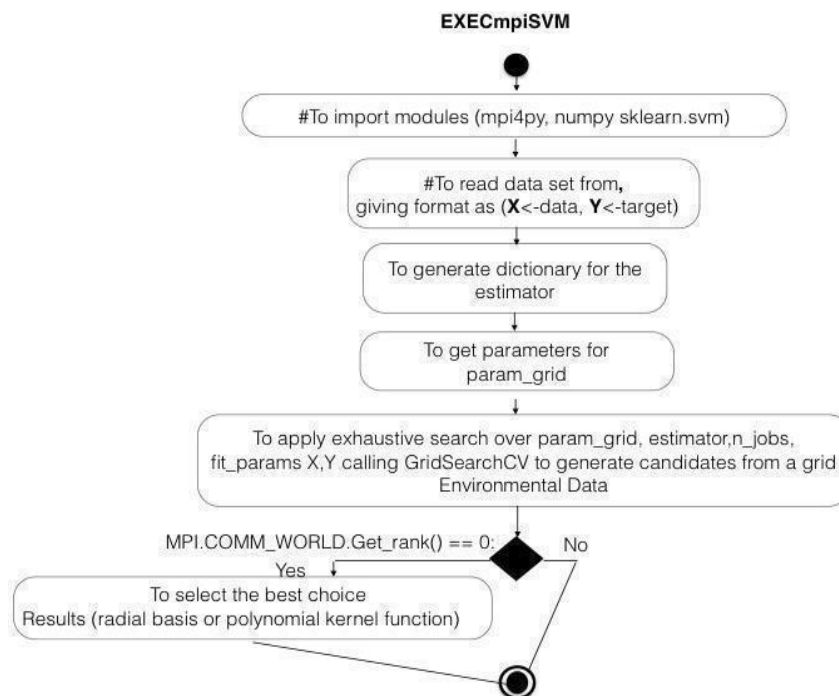


Figure 7. Diagram of the transition of activities of the behavioral view of the algorithm parallelized for SVM for data analysis of the environment.

7. Conclusions

In the proposed UML model, the automation of scientific analysis of environmental data, it is important to have a robust, scalable, interoperable, and modular infrastructure, therefore the Python libraries to be used for the implementation of the parallelized SVM algorithm must conform to the open paradigm source, ensuring that the algorithm must be flexible to extend parallel to several machines whenever it is desired to increase the processing capacity of the cloud, given the indeterminate increase in the volume of data generated year by year by the workstations and due to the expected technological phenomenon of IoT.

It is necessary for the data storage and cleaning protocol to exist a set of configurable classes so that the monitoring variable can be selected according to the emanation of captured contamination, as well as to specify the destination variable of the data in its state original and clean, in addition to its format since over time files evolve to improve their manipulation like csv, xml or json. It is recommended that in the training of data, import Python libraries that facilitate the parallel search implementation of the best kernel function for data classification.

References

- [1] X. Yu, P. Gu, X. Liu, L. Sun, D. Qian, and Y. Guo, "Modeling technology of virtual assembly system based on UML," 2017 International conference of Electronics, Communication and Aerospace Technology (ICECA), 2017, pp. 722-726, DOI: 10.1109/ICECA.2017.8212761.
- [2] A. Leshob, "Towards a Business-Pattern Approach for UML Models Derivation from Business Process Models", 2016 IEEE 13th International Conference on e-Business Engineering (ICEBE), 2016, pp. 244-249, DOI: 10.1109/ICEBE.2016.049.
- [3] Hongjie Tang, "The design model of P2P file sharing system based on UML," 2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), 2016, pp. 1257-1260, DOI: 10.1109/IMCEC.2016.7867413.
- [4] D. Martinez-Mosquera, S. Luján-Mora and H. Recalde, "Conceptual Modeling of Big Data Extract Processes with UML," 2017 International Conference on Information Systems and Computer Science (INCISCOS), 2017, pp. 207-211, DOI: 10.1109/INCISCOS.2017.18.
- [5] V. Besnard, C. Teodorov, F. Jouault, M. Brun and P. Dhaussy, "A Model Checkable UML Soccer Player," 2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C), 2019, pp. 211-220, DOI: 10.1109/MODELS-C.2019.00035.
- [6] M. Guerriero, A. Nesta and E. Di Nitto, "StreamGen: A UML-Based Tool for Developing Streaming Applications," 2018 IEEE/ACM 10th International Workshop on Modeling in Software Engineering (MiSE), 2018, pp. 57-58.
- [7] M. Bagherzadeh, K. Jahed, B. Combemale, and J. Dingel, "Live-UMLRT: A Tool for Live Modeling of UML-RT Models," 2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C), 2019, pp. 743-747, DOI: 10.1109/MODELS-C.2019.00115.
- [8] V. Besnard, C. Teodorov, F. Jouault, M. Brun and P. Dhaussy, "A Model Checkable UML Soccer Player," 2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C), 2019, pp. 211-220, DOI: 10.1109/MODELS-C.2019.00035.
- [9] J. Huixia and L. Ke, "Modeling for AI-ESTATE diagnostic services with UML," 2017 2nd International Conference on Frontiers of Sensors Technologies (ICFST), 2017, pp. 382-386, DOI: 10.1109/ICFST.2017.8210541.
- [10] E. Tudoroiu, G. Neacsu, A. Astilean, M. Zoltan, T. Letia and N. Tudoroiu, "Fuzzy UML and Petri nets modeling investigations on the pollution impact on the air quality in the vicinity of the Black Sea Constanta Romanian resort", 2011 Federated Conference on Computer Science and Information Systems (FedCSIS), 2011, pp. 763-766.
- [11] Townsend, A. M., (2003). "Smart Cities", First. New York, pp. 111–114.
- [12] Dario, C. G. M., (2015), "La Ocde y El Inegi Presentan Los Resultados más Destacados y el Sitio Web del Proyecto: Midiendo El Bienestar en Las Entidades Federativas," pp. 1–2.
- [13] "OMS | Calidad del aire ambiente (exterior) y salud," WHO, (2016).
- [14] "The Urban Internet of Things | Data-Smart City Solutions". (2015) Retrieved from <http://datasmart.ash.harvard.edu/news/article/the-urban-internet-of-things-727>.
- [15] Mora, O. B., & Larios, V. M., (2015) "Urban Operating System For Sensor Networks Management in Smart Cities", pp. 1–4.
- [16] "Monitoring Air Quality and the Impacts of Pollution | Data-Smart City Solutions", (2015) Retrieved from <http://datasmart.ash.harvard.edu/news/article/monitoring-air-quality-and-the-impacts-of-pollution-679>.

- [17] Snyder, E. G., Watkins, T. H., Solomon, P. A., Thoma, E. D., Williams, R. W. Hagler, Preuss, P. W., (2013), "The Changing Paradigm of Air Pollution Monitoring", *Environ. Sci. Technol.*, pp. 11369-11377.
- [18] E. Estrada, M. Martínez, J. Gómez, A. Peña, G. Lara, R. Maciel. "Smart Cities Big Data Algorithms for Sensors Location". *Applied Sciences*, 2019, pp. 1-14, <https://doi.org/10.3390/app9194196>.
- [19] Programa para Mejorar la Calidad del Aire Jalisco (2011-2020), Secretaría de Medio Ambiente para el Desarrollo Sustentable del estado de Jalisco.
- [20] Mendoza A., & García, M. (2018). "Aplicación de un modelo de calidad del aire de segunda generación a la Zona Metropolitana de Guadalajara, México", pp. 1-13.
- [21] Ramírez, H., Andrade, M., González, M., & Celis, A. (2006). "Contaminantes atmosféricos y su correlación con infecciones agudas de las vías respiratorias en niños de Guadalajara, Jalisco", *Salud Pública Méx*; Vol. 48(5):385-394.
- [22] Coenen, F., (2011). "Data Mining: past, present and future," *The Knowledge Engineering Review*, Vol. 26:1, 25–29. & Cambridge University Press, DOI:10.1017/S0269888910000378
- [23] Aggarwal, C., Ashish, N., & Sheth, A., (2013). "The Internet of Things: A Survey from the Data-Centric Perspective". In C. C. Aggarwal (Ed.), *Managing and Mining Sensor Data*, pp. 383–428. Boston, MA: Springer US.
- [24] Patulea, C., Peace, R. & Green, J. (2010). "CUDA-accelerated genetic feedforward-ANN training for Data Mining," *J. Phys.: Conf. Ser.*, vol. 256, pp. 012014–9.
- [25] Pebesma, E., Bivand, R., & P. J., n.d., Ribeiro, "cudaBayesreg: Parallel Implementation of a Bayesian Multilevel Model for fMRI Data Analysis".
- [26] Gullo, F. (2015). "From Patterns in Data to Knowledge Discovery: What Data Mining Can Do," *Physics Procedia*, vol. 62, pp. 18–22.
- [27] Carraher, L. A., Wilsey, P. A., & Annexstein, F. S. (2013). "A GPGPU Algorithm for c-Approximate r-Nearest Neighbor Search in High Dimensions", presented at the 2013 IEEE International Symposium on Parallel & Distributed Processing, Workshops, and Ph.D. Forum (IPDPSW), pp. 2079–2088.
- [28] Tan, K. Zhang, J., Du, Q., & Wang, X. (2016). "GPU Parallel Implementation of Support Vector Machines for Hyperspectral Image Classification", *IEEE J. Sel. Top. Appl. Earth Observations Remote Sensing*, vol. 8, no. 10, pp. 4647–4656.
- [29] Guo, W., Alham, N. K., Liu, Y., Li, M., & Qi, M. (2015). "A Resource Aware MapReduce Based Parallel SVM for Large Scale Image Classifications", *Neural Processing Letters*, vol. 44, no. 1, pp. 161–184.
- [30] Godoy-Viera, A. (2015). "Técnicas de aprendizaje de máquina utilizada en minería de datos", *Investigación Bibliotecológica*, vol. 31, no. 71, January/april, 2017, México, ISSN: 0187-358.
- [31] Vapnik, V. (1982). "Estimation of Dependences Based on Empirical Data", Springer-Verlag.
- [32] Russell S., y Norvig P. (2003). "Artificial Intelligence: A Modern Approach", Prentice-Hall.
- [33] Catanzaro, B., Catanzaro, B., Keutzer, K., & Keutzer, K., 2008, "Fast Support Vector Machine Training and Classification on Graphics Processors". *Machine Learning*, pp. 104–111.
- [34] Platt, J. C., 1998. "Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines". *Advances in Kernel Methods*, pp. 185–208.
- [35] Dogaru, R., & Dogaru, I. (2015). "A Low-Cost High-Performance Computing Platform for Cellular Nonlinear Networks using Python for CUDA". 20th International Conference on Control Systems and Science, pp. 593–598, DOI: [10.1109/CSCS.2015.36](https://doi.org/10.1109/CSCS.2015.36).
- [36] CUDA a parallel computing platform definition, 2020. Retrieved from <https://blogs.nvidia.com/blog/2012/09/10/what-is-cuda-2/>
- [37] GPU Accelerated Computing with Python, 2020. Retrieved from <https://developer.nvidia.com/how-to-cuda-python>.
- [38] NumPy Python Library Definition, 2020. Retrieved from <https://docs.scipy.org/doc/numpy/user/whatisnumpy.html>
- [39] SciPy Python Library Definition, 2020. Retrieved from <http://scipy.org/scipylib/index.html>.
- [40] Mpi4py Python Library Definition, 2021. Retrieved from <https://pypi.org/project/mpi4py/>.

Biographical Notes



Martha Patricia Martínez Vargas has a Ph.D. in Information Technology from the University of Guadalajara, Mexico. Currently, she is a full-time professor in the Systems Department and a member of the Academic research group UDG-CA-931 of the University Center for Administrative Economic Sciences. Her main lines of research are data analysis and education.



Elsa Estrada Guzmán obtained a doctorate in Information Technology in 2018. Currently, she teaches courses in the Master of Information Systems at the University of Guadalajara. Her main lines of research are Data Analysis using Machine Learning, on Smart Cities issues, as well as on Software Engineering for the development of applications for event monitoring and decision making.



Roció Maciel Arellano is a Research Professor at the Department of Information Systems of the CUCEA University of Guadalajara (UDG). She works as a Researcher and coordinator of Special Projects of the Center for Innovation in Smart Cities of the UDG. Among his areas of research interest from the perspective of Smart Cities are Smart People strategies in virtual or online education, the application of technology for the inclusion of people with different abilities and the development of user experience on technological platforms. Likewise, she has organized different congresses, diplomas, workshops and has collaborated in the design of undergraduate and postgraduate educational programs oriented to Information Technologies. Additionally, she has indexed scientific publications, patents, and has given lectures and participated in national and international panels.



Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-CompartirIgual 2.5 México.

Clasificación de parásitos en imágenes de copro con transferencia de aprendizaje y aumento de datos

Parasite classification in copro images with transfer learning and data augmentation

Miguel Ángel Gutiérrez Velázquez¹
m19061419@chihuahua.tecnm.mx

Mario Ignacio Chacon-Murguia¹
mario.cm@chihuahua.tecnm.mx

Juan Alberto Ramirez-Quintana¹
juan.rq@chihuahua.tecnm.mx

Carlos Arzate Quintana²
carzate@uach.mx

Alma Delia Corral-Saenz¹
alma.cs@chihuahua.tecnm.mx

¹ Tecnológico Nacional de México / I.T. Chihuahua, Chihuahua, Chihuahua, México.

² Universidad Autónoma de Chihuahua, Chihuahua, Chihuahua, México.

Resumen

Los seres humanos pueden albergar parásitos, por lo tanto, es fundamental una detección temprana para prevenir enfermedades. Los parásitos pueden observarse a través de imágenes microscópicas, con lo cual, la visión por computadora se muestra como un enfoque que pueda ayudar a la detección y clasificación de parásitos en imágenes digitales. Los modelos de aprendizaje profundo han mostrado un desempeño formidable en la clasificación de imágenes, debido a esto, en este artículo se presentan varios clasificadores profundos multiclase para reconocer 8 clases: 7 tipos de parásitos y la clase no parásitos. Los clasificadores diseñados utilizan transferencia de aprendizaje basada en la arquitectura AlexNet modificada. Al contar con una cantidad de muestras escasa de imágenes de parásitos, se realizó también un aumento de datos, tanto con el método tradicional como mediante la generación de imágenes con una red generativa adversaria (GAN) diseñada para ese propósito. El clasificador con mejor desempeño presentó una exactitud del 99.94%, 98.97% y 98.18% en el conjunto de entrenamiento, validación y de prueba, respectivamente.

Palabras clave: Clasificación de parásitos, aumento de datos, transferencia de aprendizaje, GAN, AlexNet.

Resume

Humans can harbor parasites; hence, it is fundamental an early detection to prevent diseases. Parasites can be observed in microscopic images, and computer vision may be a helpful approach to detect and classify those parasites in digital images. Deep learning models have shown to have a high performance in image classification. Therefore, this article presents various multi-class deep learning classifiers to recognize 8 classes: 7 types of parasites and non-parasite class. The designed classifiers are based on transfer learning from an AlexNet modified architecture. By having a reduce amount of parasite images samples, a data augmentation was done, employing traditional methods and images generation with an adversarial neural network (GAN) designed for this purpose. The classifier with best performance presented a 99.94%, 98.97% and 98.18% accuracy in the training, validation, and testing sets, respectively.

Keywords: Parasite classification, data augmentation, transfer learning, GAN, AlexNet.

1. Introducción

La visión por computadora es un campo científico cuyo objetivo es extraer información de imágenes digitales para su posterior análisis. Para efectuar tal extracción, se emplean algoritmos que realizan operaciones en las imágenes con el fin de mejorar la información pictórica para la interpretación humana, o para tareas como almacenamiento, transmisión y extracción de información en las imágenes. La visión por computadora aparece en innumerables campos, como: agricultura (Song C. *et al*, 2020), automatización (Chain E. *et al*, 2021), seguridad (Prakash R. *et al*, 2020), medicina (Meda K. *et al*, 2021), entre otros. La integración de la visión por computadora en el área de la medicina ayuda a los expertos a realizar los diagnósticos y tratamientos médicos de una manera más eficiente mediante el análisis de imágenes médicas. Las imágenes médicas predominantes son las obtenidas mediante rayos X, tomografía computarizada (CT), imagen de resonancia magnética (MRI), tomografía de emisión de positrones (PET) y ultrasonido. Además de los métodos anteriores, existen imágenes obtenidas a partir de un microscopio. Las imágenes conseguidas con un microscopio amplifican objetos diminutos, permitiendo observar lo que no es posible de otra manera. Gracias a esto, se puede analizar y examinar la composición, la estructura y las características de células y tejidos de seres vivos, así como la detección y clasificación de parásitos que habitan en el interior de humanos y animales.

El 2 de marzo del 2020, la Organización Mundial de la Salud (OMS) publicó una noticia acerca de la helmintiasis, enfermedad de parásitos (OMS, 2020). Se menciona la cifra escandalosa de 1500 millones de personas infectadas en el mundo. Esto aun cuando la OMS, en el 2001, solicitó a los países afrontar esta enfermedad. Dado que las cifras proporcionadas por la OMS son alarmantes y exponen un gran riesgo e impacto en la salud (para una sola enfermedad intestinal), por lo que es menester el desarrollo de tecnologías que detecten eficientemente parásitos intestinales. Una forma es mediante un sistema de análisis de imágenes digitales de muestras de copro. En esta investigación, se utilizan imágenes de parásitos y de no parásitos para clasificar el tipo de parásito presente en dichas imágenes o, por el contrario, clasificar el objeto que aparece en la imagen como no parásito. La clasificación de parásitos en imágenes de muestras de copro se realiza con una red neuronal profunda modificada de la red Alexnet, haciendo una transferencia de aprendizaje. Además, debido a que el número de muestras de imágenes de copro era reducido, se realizó un aumento de datos en el conjunto de entrenamiento mediante técnicas clásicas y con una red generativa adversaria, GAN, diseñada para ese propósito. Las contribuciones de este trabajo son: diseño de clasificadores mediante técnica de transferencia de aprendizaje, donde el clasificador con mejor desempeño presentó una exactitud del 98.18% en el conjunto de prueba, y el aumento de datos: una demostración de que las imágenes generadas con la GAN incrementan el desempeño del clasificar profundo.

El artículo está organizado como sigue: en la sección 2 se muestran trabajos relacionados, en la sección 3 se presenta el modelo de GAN diseñado para aumento de datos. la técnica de transferencia de aprendizaje utilizada y el clasificador, en la sección **Error! Reference source not found.** se describen los experimentos realizados y se analizan los resultados obtenidos. Finalmente, en la sección 1 se comentan las conclusiones del trabajo.

2. Trabajos relacionados

Existe en la literatura algunos trabajos relacionados al tema de análisis de imágenes médicas. En esta sección se presentan trabajos donde se realiza una clasificación de parásitos mediante clasificadores profundos. Por ejemplo, en (Shorten C. *et al*, 2021; Suriva *et al*, 2019; Shah *et al*, 2020) se reportan trabajos donde se realizan clasificadores binarios utilizando redes neuronales convolucionales (CNN). En (Shorten C. *et al*, 2021) se clasifica malaria en imágenes de células rojas. Las imágenes de entrada a la CNN son de 125x125x3, la CNN tiene 5 capas de convolución (con 32, 64, 128, 256 y 300 filtros respectivamente), 5 capas de max-pooling y 2 capas totalmente conectadas. El conjunto de imágenes utilizado es del conjunto de datos NIH Malaria, donde se tienen 13779 imágenes con parásitos y 13779 sin parásitos. Este conjunto de datos fue aumentado mediante técnicas clásicas hasta obtener 173700 imágenes. El clasificador tuvo una precisión de 96.33%. En (Suriva . *et al*, 2019) se realiza una clasificación de imágenes de células rojas infectadas debido a presencia de parásitos. Primeramente, la imagen es procesada para tener un tamaño de 128x128. Luego se entrena la red diseñada. La cual tiene 19 capas de convolución para la extracción de características, 6 capas de max

pooling para reducir el número de cálculos computacionales, 3 capas de normalización por lotes para normalizar las características, 4 capas totalmente conectadas y una capa plana. El conjunto de datos consiste en 26188 imágenes, 13105 pertenecen a la clase de imágenes parasitadas y 13083 a imágenes no parasitadas. La red tuvo un 98.9% de precisión. En (Shah . *et. al*, 2020) se identifica el parásito *Plasmodium* en imágenes de sangre. La CNN propuesta consta de 3 capas de convolución con 32, 64 y 128 filtros, respectivamente. Cada filtro tiene un tamaño de 3x3 y la función de activación es la ReLU. El algoritmo de aprendizaje es el optimizador Adam. Además, el modelo propuesto tiene 3 capas de pooling, 3 capas totalmente conectadas y la salida de la red está dada por la función sigmoideal. El conjunto de entrenamiento consiste en 8000 imágenes, mientras que el conjunto de prueba contiene 2000 imágenes. Para cuantificar la funcionalidad del modelo, se tienen 17460 imágenes de prueba, con 8760 imágenes con la presencia de parásito y 8700 sin presencia de parásito. La precisión del modelo en el conjunto de prueba mencionado fue de 95%.

Por otro lado, los enfoques basados en la utilización de redes profundas pre entrenadas se presentan en (Huq y Pervin, 2020; Nakasi *et. al*, 2021; Var y Boray, 2018; Bharadwaj y Sujitha, 2019; Chakradeo *et. al*, 2021). En (Huq y Pervin, 2020) se implementó la red VGG16 para identificar imágenes de células con parásitos y sin parásitos. Se utilizó un conjunto de datos disponible en National Institute of Health, que tiene 27558 imágenes disponibles. De esas imágenes, 13779 son imágenes sin parásitos y 13779 son imágenes infectadas. El 60% de las imágenes fueron utilizadas para el entrenamiento, el 10% para la validación y el 30% para la prueba. El desempeño fue de 95.96%. En (Nakasi *et. al*, 2021) se clasifican parásitos intestinales en muestras de heces. El conjunto de datos consta de 3327 imágenes con 11141 parásitos, este conjunto fue aumentado mediante técnicas clásicas para obtener 7 imágenes a partir de una sola. Se utilizaron las redes AlexNet y GoogleNet para la clasificación. Ambas redes se entrenaron durante 10 épocas con el optimizador Adam, con un factor de aprendizaje de 0.001 y un momentum de 0.9. La métrica de evaluación fue el área bajo la curva (AUC) de la región de convergencia (ROC). Utilizando la red GoogleNet se obtuvo un AUC de 0.99, mientras que con la AlexNet el AUC fue 1.00. En (Var y Boray, 2018) se clasifican parásitos en imágenes de muestras de sangre. Se aplicó transferencia de aprendizaje utilizando la red CNN pre entrenada VGG19. La red se entrenó durante 20 épocas para 5 clases: 1428 imágenes para la clase *Plasmodium Vivax*, 1425 para la clase *Plasmodium Ovale*, 1446 para la clase *Plasmodium Falciparum*, 1450 para la clase *Plasmodium Malariae* y 1440 para la clase de no parásitos. El modelo alcanza una precisión de 80%, 83%, 86% y 75% de precisión en 19 imágenes de prueba. En (Bharadwaj y Sujitha, 2019) se clasifican imágenes de células infectadas debido a presencia de parásitos implementando transferencia de aprendizaje. El conjunto de datos contiene 27558 imágenes de células infectadas y no infectadas, dicho conjunto de datos está disponible en el sitio web de National Library of Medicine. La red pre entrenada utilizada es la ResNet50. Se utilizó el optimizador gradiente descendiente estocástico (SGD) y la función de pérdida es la entropía cruzada. La red se entrenó durante 9 épocas. La precisión del entrenamiento fue de 95.91% y de la validación 95.4%. En (Chakradeo *et. al*, 2021) realiza una clasificación binaria de imágenes de células infectadas y no infectadas. Se tiene un conjunto de datos con 12058 imágenes infectadas y 14142 imágenes no infectadas. Luego, mediante aumento de datos se alcanzan las 22046 imágenes para cada clase para el entrenamiento. La red reportada en (Chakradeo *et. al*, 2021) utilizada está basada en la red VGG; la arquitectura de la red implementada en (Chakradeo *et. al*, 2021) tiene 6 capas de convolución, las capas 1 y 2 tienen 64 filtros, la 3 y 4 tienen 128 filtros y la 5 y 6 tienen 256 filtros. La función de activación es la Leaky ReLU, hay tres capas de max pooling con dropout y dos capas totalmente conectadas. La exactitud fue de 98.2%.

Al analizar los trabajos anteriores se observa que en la mayoría de los casos fue necesario contar con un gran número de muestras iniciales, con lo cual surge la necesidad de investigar si la clasificación de parásitos es posible cuando la cantidad de muestras disponibles es reducida. Por lo tanto, esta investigación tiene como fin diseñar un clasificador de parásitos intestinales con alto desempeño utilizando una red profunda partiendo de una muestra reducida de muestras, 358 muestras de imágenes de parásitos (estas 358 muestras se definirán como imágenes de parásitos originales, *IPO*) y 845 muestras negativas (no parásitos), este conjunto de muestras negativos se define como *NP*.

3. Aumento de datos, transferencia de aprendizaje y clasificadores

Esta sección presenta el aumento de datos mediante técnicas clásicas y mediante la generación de imágenes con una GAN. Posteriormente, se describe el concepto de transferencia de aprendizaje y el diseño de la red pre entrenada que se utiliza en esta investigación.

3.1 Aumento de datos

El aumento de datos tradicional está basado en transformar una imagen para obtener una nueva imagen que aporte en las muestras de entrenamiento y prueba para modelos de aprendizaje de maquina o profundo. Las transformaciones pueden ser: rotación, reflejo, adición de ruido, modificación del brillo, corrección gamma, borrado aleatorio, cambio en el espacio de color, etcétera (Shorten y Khoshgoftaar, 2019). En la Figura 1 se muestran algunas de estas transformaciones.

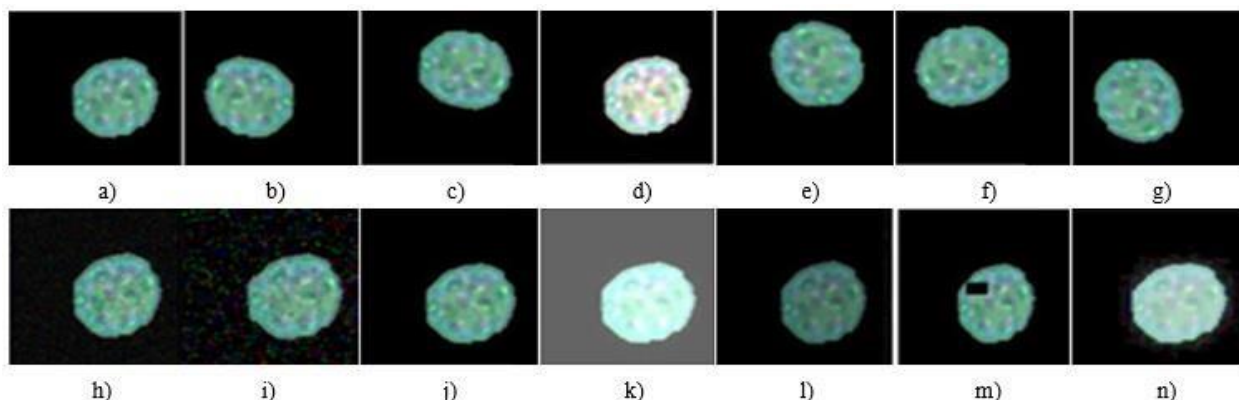
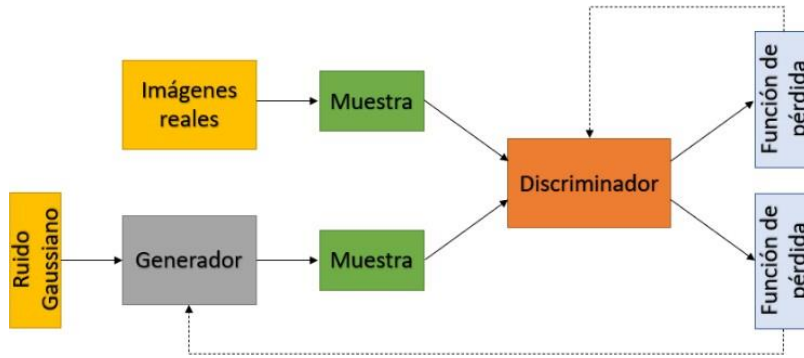


Figura 1. Aumento de datos mediante transformaciones, a) Imagen original, b) Imagen reflejada en el eje y, c) Imagen reflejada en el eje x, d) Aumento de contraste, e) Imagen rotada 90°, f) Imagen rotada 180°, g) Imagen rotada 270°, h) Imagen con ruido gaussiano, i) Imagen con ruido sal y pimienta, j) Imagen con ruido Poisson, k) Aumento de brillo, l) Disminución de brillo, m) Borrado aleatorio, n) Corrección gamma.

Adicional a estas transformaciones, existen paradigmas de aumento de datos que generan nuevas imágenes, por ejemplo, la GAN. La GAN es un paradigma basado en los métodos generativos propuesto por Goodfellow *et. al* (Goodfellow *et. al*, 2014). Mientras que los modelos discriminativos asocian una entrada a unas coordenadas en un hiperplano y de esa manera le asignan la etiqueta correspondiente, los modelos generativos intentan producir una salida que se parezca a la entrada.

En la Figura 2 se observa la arquitectura general de una GAN, donde existen dos clasificadores: el generador y el discriminador. El discriminador tiene dos entradas: las imágenes reales y las imágenes generadas por el generador. En cambio, el generador tiene como entrada ruido gaussiano. El discriminador aprende de las imágenes reales y de las imágenes creadas por el generador, de tal modo que clasifica una imagen como real o artificial. Por otra parte, el generador intenta engañar al discriminador creando imágenes parecidas al conjunto de muestras reales. Basado en este modelo de GAN, se presenta en este trabajo la red GAN profunda (DCGAN) (Radford y Metz, 2015), cuya arquitectura se muestra en la Tabla 1. La DCGAN se entrenó 7 veces para generar artificialmente imágenes de los parásitos estudiados en este trabajo. Estos 7 entrenamientos se deben a que se tienen 7 clases de parásitos, y cada entrenamiento corresponde a la generación de imágenes de un tipo específico de parásito. Para cada entrenamiento, las imágenes de cada parásito fueron aumentadas con métodos tradicionales, ya que esto mejora el resultado de las imágenes generadas, visualmente hablando.



Generador			Discriminador		
Capa	Tipo de capa	Dimensión de la activación	Capa	Tipo de capa	Dimensión de la activación
1	Ruido gaussiano de entrada	1x1x100	1	Conjunto de entrenamiento	64x64x3
2	Proyección y cambio de tamaño	4x4x512	2	Dropout	64x64x3
3	Convolución traspuesta	8x8x384	3	Convolución	32x32x64
4	Normalización por lotes	8x8x384	4	Leaky ReLU	32x32x64
5	ReLU	8x8x384	5	Convolución	16x16x128
6	Convolución traspuesta	16x16x192	6	Normalización por lotes	16x16x128
7	Normalización por lotes	16x16x192	7	Leaky ReLU	16x16x128
8	ReLU	16x16x192	8	Convolución	8x8x256
9	Convolución traspuesta	32x32x96	9	Normalización por lotes	8x8x256
10	Normalización por lotes	32x32x96	10	Leaky ReLU	8x8x256
11	ReLU	32x32x96	11	Convolución	4x4x512
12	Convolución traspuesta	64x64x3	12	Normalización por lotes	4x4x512

Figura 2. Modelo general de una red GAN.

13	Tanh	64x64x3	13	Leaky ReLU	4x4x512
			14	Convolución	1x1x1

Tabla 1. Arquitectura del generador y discriminador de la DCGAN.

A continuación, se presentan la cantidad de muestras de imágenes de parásitos de entrenamiento usadas para diseñar la DCGAN para cada tipo de parásito: 1548 *Blastocystis hominis* (BH), 984 *Entamoeba coli* (EC), 672 *Chilomastix Mesnili* (CM), 1440 *Entamoeba histolytica* (EHY), 384 *Giardia Lamblia* (GL), 251 *Endolimax nana* (EN) y 2881 de *Enteromonas hominis* (EHO). Algunas imágenes generadas con la DCGAN diseñada para cada tipo de parásito se muestran en la Figura 3. La DCGAN fue entrenada durante 1000 épocas con un factor de aprendizaje de 0.0002 y un valor de decaimiento de 0.7 (valores típicos), el tamaño del mini-batch es de 64. La cantidad de imágenes generadas con la DCGAN es: 218 BH, 475 EC, 157 CM, 204 EHY, 158 GL, 466 EN y 1014 EHO, para un total de 2692 imágenes generadas con la DCGAN.

3.2 Transferencia de aprendizaje

El escenario ideal para que una red profunda aprenda es aquel donde hay decenas de miles imágenes de entrenamiento etiquetadas, que tienen la misma distribución que las imágenes de prueba. Sin embargo, generar esta cantidad de imágenes en ocasiones no es viable por el costo, trabajo y tiempo necesarios para obtener este tipo de información y realizar el entrenamiento correspondiente.

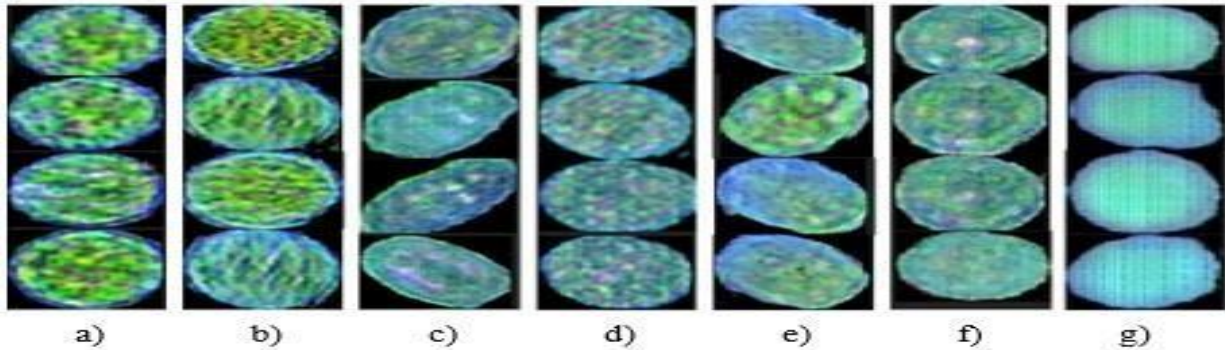


Figura 3. Algunas imágenes generadas con la GAN, a) *Blastocystis hominis*, b) *Entamoeba coli*, c) *Chilomastix mesnili*, d) *Entamoeba histolytica*, e) *Giardia lamblia*, f) *Endolimax nana*, g) *Enteromonas hominis*.

Para dar solución a este problema, existe la transferencia de aprendizaje. La transferencia de aprendizaje es utilizada para mejorar el aprendizaje de un dominio al transferir información de otro dominio relacionado (Weiss *et. al*, 2016). El aprender a transferir es el resultado de la generalización de la experiencia. Es decir, es posible realizar una transferencia de conocimiento de una situación a otra, siempre y cuando se generalice la experiencia.

El requisito de la transferencia es que exista una conexión entre las dos actividades de aprendizaje. La transferencia de aprendizaje se inspira en la habilidad de los seres humanos de transferir conocimientos a través de dominios, de un dominio fuente a un dominio objetivo. Donde el dominio fuente es el dominio del que ya se tiene conocimiento, y el dominio objetivo es el que se quiere aprender a partir de lo que se sabe sobre el dominio fuente.

De esta manera, esta forma de aprendizaje tiene como objetivo aprovechar el conocimiento de un dominio relacionado para mejorar el aprendizaje o minimizar el número de ejemplos etiquetados requeridos en otro dominio de prueba (Zhuang *et. al*, 2020) En este trabajo, el dominio fuente es lo que aprendió la red Alexnet durante el entrenamiento con millones de imágenes correspondientes a diversas clases, y el dominio objetivo son las imágenes de parásitos y no parásitos Para entender la transferencia de aprendizaje se requieren varias definiciones (Pan y Yang, 2009).

- Dominio: Un dominio D está formado de dos partes: el espacio de características χ y su distribución marginal $P(X)$.

$$D = \{\chi, P(X)\} \quad (1)$$

donde X representa el conjunto de instancias

$$X = \{x | x_i \in \chi, i = 1, \dots, n\} \quad (2)$$

- Tarea: Una tarea τ consiste en un espacio de etiqueta Y y una función de decisión f que se espera sea aprendida desde los datos de muestra

$$\tau = \{Y | f\} \quad (3)$$

Algunos modelos de aprendizaje dan como salida la distribución condicional predicha de los ejemplos

$$f(x_j) = \{P(y_k | x_j) | y_k \in Y, k = 1, \dots, |Y|\} \quad (4)$$

En la práctica, el dominio se observa por un número de instancias x con o sin la información etiquetada (en este trabajo, al ser aprendizaje supervisado, se tiene la información etiquetada). El dominio fuente D_S correspondiente a la tarea fuente τ_S se expresa como la tupla ejemplo-etiqueta

$$D_S = \{(x, y) | x_i \in \mathcal{X}^S, y_i \in Y^S, i = 1, \dots, n^S\} \quad (5)$$

- **Transferencia de aprendizaje:** Dado una o más observaciones correspondientes a $m^S \in \mathbb{N}^+$ dominios y tareas $\{(D_{S_i} | \tau_{S_i}) | i = 1, \dots, m^S\}$ y alguna o algunas observaciones $m^T \in \mathbb{N}^+$ sobre el dominio y tarea de prueba $\{(D_j | \tau_T) | j = 1, \dots, m^T\}$, donde m^S es la cantidad de dominios fuente y m^T la cantidad de tareas, la transferencia aprendizaje utiliza el conocimiento obtenido del dominio fuente al mejorar el desempeño de las funciones de decisión aprendidas f^j ($j = 1, \dots, m^T$) en el dominio de prueba. En este trabajo la cantidad de tareas son 8, ya que se clasificarán 7 clases de parásitos y la clase de no parásitos.

La transferencia de aprendizaje que se implementó fue con la red Alexnet, la cual es una CNN profunda utilizada en el concurso de ImageNet LSVRC en 2012 para clasificar imágenes entre 1000 clases. La red tiene 60 millones de parámetros y 650000 neuronas, 5 capas convolucionales, max pooling y tres capas totalmente conectadas (Krizhevsky *et. al*, 2012). AlexNet se entrenó con 1.2 millones de imágenes, con 50,000 imágenes de validación y 150,000 imágenes de prueba.

El conocimiento de la red son los valores de los filtros. Esto es, cada filtro activará ciertas regiones de la imagen de entrada, generando las características que analizará la red. Por ejemplo, la Figura 4a muestra una imagen de un parásito y en la Figura 4b se observan algunas activaciones en la primera capa de convolución de la red AlexNet. Estas activaciones en las capas de convolución son conocimiento sobre ciertas características que aprendió la red durante el entrenamiento, y ese conocimiento se utiliza para reconocer ciertas características de las imágenes de parásitos de entrada que no estaban en el conjunto de entrenamiento con el cual la red fue entrenada.

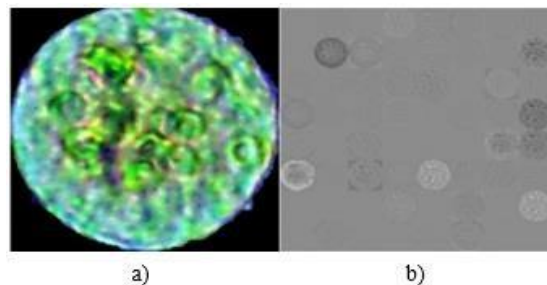


Figura 4. a) Imagen de entrada a la red AlexNet, b) Algunas características de la primera capa de convolución de la red AlexNet.

3.3 Clasificadores

Para el diseño de los clasificadores multiclase se hace una transferencia de aprendizaje de una red profunda ya entrenada (AlexNet). Para ello, se modificó la última capa de la red AlexNet, correspondiente a una capa totalmente conectada, para que sea de tamaño 8, pues ahora solo clasificarán 8 clases y no mil, para la cual fue diseñada originalmente. La Figura 5 muestra la arquitectura de la red que se entrenó con el gradiente estocástico descendente con momentum, con una tasa de aprendizaje inicial de 0.001. El total de épocas es de 20 con un mini lote de 128. Lo único que cambia en cada clasificador (antes de ser entrenados) es el conjunto de entrenamiento.

4. Aumento de datos, transferencia de aprendizaje y clasificadores

En esta sección se explican los experimentos realizados y sus resultados. Los diversos experimentos tienen como fin el ver cómo cambia el desempeño del clasificador profundo al utilizar distintas muestras de entrenamiento. Todos los clasificadores utilizaron la arquitectura mostrada en la Figura 5.

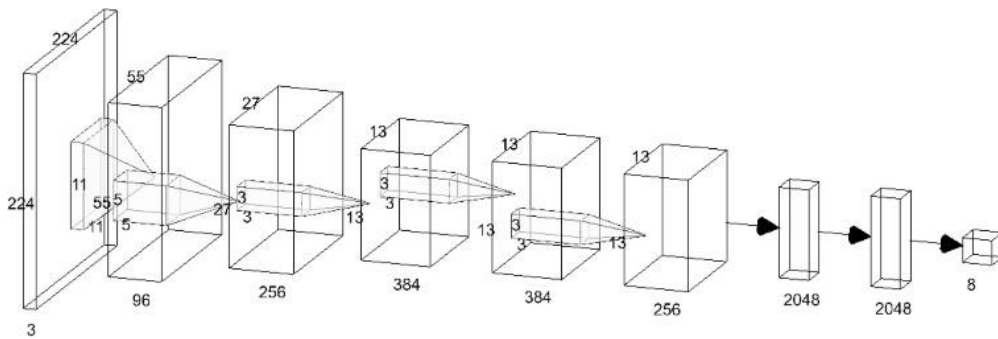


Figura 5. Arquitectura de la red para los clasificadores diseñados.

Las 358 muestras de parásitos reales están divididas de la siguiente manera: 72 *BH*, 41 *EC*, 28 *CM*, 60 *EHY*, 16 *GL*, 21 *EN* y 120 *EHO*.

4.1 Clasificadores multiclase

Clasificador 1, C1, se entrenó la red con las 358 muestras de parásitos reales y las 845 imágenes de *NP*. Las muestras fueron divididas en 70% para el entrenamiento, 15% para la validación y 15% para la prueba para los 4 clasificadores diseñados. El desempeño de C1 se muestra en la Tabla 2. Clasificador 2, C2, se entrenó la red con las 358 imágenes de parásitos reales más las 2692 imágenes generadas con la DCGAN y las 845 imágenes de *NP*. Los resultados de C2 se muestran en la Tabla 3. Clasificador 3, C3, se hizo un aumento de datos con técnicas clásicas a las 358 muestras de parásitos. El aumento de datos consistió en reflejar la imagen en el eje x y en el eje y, adición de ruido gaussiano, ruido sal y pimienta y ruido de poisson, y rotación de 90°, 180° y 270°. El conjunto de entrenamiento de C3 consiste en las muestras: 1548 *BH*, 984 *EC*, 672 *CM*, 1440 *EHY*, 384 *GL*, 251 *EN*, 2881 *EHO* y las 845 imágenes de *NP*. El desempeño de C3 se presenta en la Tabla 4. Clasificador 4, C4, este clasificador tiene como conjunto de entrenamiento el mismo que C3, más las 2692 imágenes generadas con la DCGAN. En la Tabla 5 se muestra el desempeño de C4

Conjunto	Exactitud
Entrenamiento	100%
Validación	92.22%
Prueba	91.67%

Tabla 2. Resultados del clasificador C1

Conjunto	Exactitud
Entrenamiento	99.89%
Validación	96.06%
Prueba	95.88%

Tabla 3. Resultados del clasificador C2.

Conjunto	Exactitud
Entrenamiento	100%
Validación	98.07%
Prueba	98.08%

Tabla 4. Resultados del clasificador C3.

Conjunto	Exactitud
Entrenamiento	99.94%
Validación	98.97%
Prueba	98.18%

Tabla 5. Resultados del clasificador C4.

El primer clasificador, C1, obtuvo un desempeño del 91.67% en el conjunto de prueba. El clasificador, C2, tuvo un desempeño superior a C1, del 91.67% al 95.88%. El clasificador C3 tuvo como conjunto de entrenamiento a las imágenes originales con aumento de datos tradicional. En la Tabla 4 se observa que su desempeño fue del 98.08% en el conjunto de prueba. Tal conjunto contiene imágenes reales, por lo que podría compararse con C1 directamente. Este experimento muestra que el desempeño con aumento de datos tradicional es mucho mejor que el obtenido utilizando solamente las muestras originales. Por último, se entrenó el clasificador C4.

El conjunto de entrenamiento de C4 fue una combinación de las muestras originales, muestras generadas con la DCGAN y muestras obtenidas con aumento de datos tradicional. En la Tabla 5 se observa un ligero incremento en el desempeño en relación con C3: de 98.08% a 98.18%. Con estos resultados se comprueba que el aumento de datos mejora el desempeño del clasificador.

En la Figura 6, Figura 7, Figura 8 y Figura 9 se muestra las matrices de confusión para C1, C2, C3 y C4 para el conjunto de prueba que le pertenece a cada clasificador. Las filas corresponden a las etiquetadas reales y las columnas a las salidas del clasificador. La penúltima fila muestra los porcentajes de clasificaciones correctas e incorrectas para cada clase real. La columna de la derecha muestra los porcentajes de las clasificaciones correctas e incorrectas para cada clase predicha por el clasificador.

En la diagonal de la matriz de confusión (las casillas verdes) se muestran las clasificaciones correctas para cada clase, y los elementos fuera de la diagonal en la matriz de confusión (casillas rojas) están las clasificaciones incorrectas.

BH	8	1	0	1	1	0	0	0	72.7%
CM	1	2	0	0	1	0	0	0	50%
EN	0	1	2	0	0	0	0	0	66.7%
EC	0	0	0	6	0	0	0	0	100%
EHY	2	0	0	0	7	0	0	0	77.8%
EHO	0	0	0	0	0	15	0	3	83.3%
GL	0	0	0	0	0	0	1	1	50%
NP	1	1	0	0	0	1	0	124	97.6%
	66.7%	40%	100%	85.7%	77.8%	93.8%	100%	96.9%	91.7%
	BH	CM	EN	EC	EHY	EHO	GL	NP	

Figura 6. Matriz de confusión de C1 en el conjunto de prueba de C1.

BH	39	1	0	0	1	0	0	2	90.7%
CM	3	22	1	1	1	0	0	0	78.6%
EN	0	1	72	0	0	0	0	0	98.6%
EC	2	0	0	75	0	0	0	0	97.4%
EHY	0	0	0	1	38	0	1	0	95%
EHO	0	0	0	0	0	170	0	0	100%
GL	0	0	0	1	1	1	22	1	84.6%
NP	0	1	0	0	0	2	1	123	96.9%
	88.6	88%	98.6%	96.2%	92.7%	98.3%	91.7%	97.6%	99.2%
	BH	CM	EN	EC	EHY	EHO	GL	NP	

Figura 7. Matriz de confusión de C2 en el conjunto de prueba de C2.

BH	225	0	0	0	6	0	0	1	97%
CM	0	97	0	0	4	0	0	0	96%
EN	0	1	36	0	1	0	0	0	94.7%
EC	0	0	0	148	0	0	0	0	100%
EHY	7	0	0	3	206	0	0	0	95.4%
EHO	0	0	0	0	0	431	0	1	99.8%
GL	0	1	0	0	0	0	57	0	98.3%
NP	0	0	0	0	0	0	1	126	99.2%
	97%	98%	100%	98%	94.9%	100%	98.3%	98.4%	99.2%
	BH	CM	EN	EC	EHY	EHO	GL	NP	

Figura 8. Matriz de confusión de C3 en el conjunto de prueba de C3.

BH	264	0	0	1	0	0	0	0	99.6%
CM	3	108	10	0	1	2	0	0	87.1%
EN	0	0	103	0	4	0	1	0	95.4%
EC	0	0	0	219	0	0	0	0	100%
EHY	2	0	0	2	243	0	0	0	98.4%
EHO	0	0	0	0	0	584	0	0	100%
GL	0	0	0	1	0	0	79	1	97.5%
NP	0	0	0	0	0	4	0	123	96.9%
	98.1%	100%	91.2%	98.2%	98%	99%	98.8%	99.2%	99.2%
	BH	CM	EN	EC	EHY	EHO	GL	NP	

Figura 9. Matriz de confusión de C4 en el conjunto de prueba de C4.

Finalmente, se realiza un experimento, E1, que tiene como objetivo evaluar de mejor manera el desempeño de C2. Pues C2 fue entrenado con un alto porcentaje de imágenes generadas con la DCGAN, y en el conjunto de prueba también hay un alto porcentaje de imágenes generadas con la DCGAN en relación con las imágenes reales. E1 consiste, entonces, en clasificar el conjunto de imágenes utilizadas en C3 (puras imágenes reales con aumento de datos) sin las IPO (imágenes de parásitos originales). Es decir, clasificar solamente las imágenes obtenidas con el aumento de datos tradicional. El resultado de C2 en la clasificación del conjunto de prueba descrito anteriormente se muestra en la 0. En la 0 se observa que el desempeño de C2 al clasificar imágenes reales es muy bajo. Pero hay clases como *BH*, *EHY* y *EHO* que tuvieron buen desempeño (88.6%, 90.4% y 99.2% respectivamente). Con lo cual se puede concluir que las imágenes generadas de ciertos parásitos son mejores que para otro tipo de parásito. Como se describió en la sección 3.1, en el entrenamiento de la DCGAN se utilizaron más imágenes de *BH*, *EHY* y *EHO* que de los otros 4 tipos de parásitos. Es decir, quizá si se generan más imágenes con la DCGAN utilizando un conjunto pequeño de imágenes, el desempeño de un clasificador entrenado con muestras generadas por esa DCGAN resultaría mejor al clasificar imágenes reales.

BH	838	0	321	294	7	0	4	12	56.8%
CM	2	294	280	1	28	3	17	19	45.7%
EN	0	10	209	0	11	0	0	0	90.9%
EC	41	0	1	898	3	0	0	0	95.2%
EHY	65	21	442	214	64	0	11	13	44.5%
EHO	0	23	505	0	0	2055	4	173	74.5%
GL	0	50	122	11	15	9	101	60	27.4%
NP	0	1	0	0	1	4	1	836	99.2%
	88.6%	73.7%	11.1%	63.3%	90.4%	99.2%	73.2%	75.2%	67.6%
	BH	CM	EN	EC	EHY	EHO	GL	NP	

Figura 10. Matriz de confusión del experimento E1.

1. Conclusiones

En este trabajo se presentaron varios clasificadores profundos que fueron entrenados utilizando diversos conjuntos de entrenamiento. Con lo cual, no solamente se presenta un clasificador profundo con un alto desempeño, sino que diversos experimentos para saber cómo varía el desempeño si se modifica el conjunto de entrenamiento. Tales conjuntos de entrenamiento consisten en alguna combinación de *IPO* e *NP*, imágenes con aumento de datos tradicional e imágenes generadas con la DCGAN. El clasificador C1, cuyo conjunto de muestras únicamente presenta imágenes de parásitos reales y las imágenes de *NP* obtuvo un desempeño del 91.67% en el conjunto de prueba. El clasificador C3 tiene como muestras las mismas imágenes de parásitos reales que C1, pero fueron incrementadas mediante aumento de datos, y las imágenes de *NP*. El desempeño de C3 fue del 98.08% en el conjunto de prueba. Por último, el clasificador C4 contiene como muestras las mismas que C3, pero con la adición de las imágenes de parásitos generadas con la DCGAN. Este clasificador presentó una exactitud del 98.18% en el conjunto de prueba.

Estos resultados son importantes para demostrar que el aumento de datos mejora notablemente la clasificación multiclase de imágenes de parásitos, incluso sin tener miles de imágenes de muestra, como en otros trabajos. Además, se observó que un clasificador entrenado con imágenes generadas con la DCGAN, en combinación con imágenes reales, obtiene un mejor desempeño que sin ellas. En cambio, diversas investigaciones se ven necesarias: primeramente, entrenar un clasificador con un alto porcentaje de imágenes generadas con la DCGAN y que tenga un buen desempeño al clasificar imágenes reales, ya que en el experimento E1, se observa que C2 no tiene un buen desempeño en la clasificación de imágenes de parásitos reales para todos los tipos de parásitos; también se presenta la necesidad de investigar qué características aprende el modelo profundo a partir del aumento de datos tradicional y DCGAN. Como se comprobó en este trabajo, el aumento de datos tradicional y DCGAN incrementa el desempeño de un clasificador profundo. Se intuye, por lo tanto, que una red profunda obtiene más información y encuentra más características cuando se tiene aumento de datos, sería muy interesante y útil saber cuáles son, es decir: qué características se obtienen de las imágenes de parásitos con ciertas modificaciones, que no se tienen en las imágenes de parásitos originales.

Agradecimientos

Este proyecto se realizó bajo el apoyo del Tecnológico Nacional de México con número 10071.21-P.

REFERENCIAS

- Bharadwaj A. y Sujitha D. (2019). Transfer Learning with ResNet-50 for Malaria Cell-Image Classification. *2019 International Conference on Communication and Signal Processing (ICCSP)*, 946-949. DOI: 10.1109/ICCSP.2019.8697909
- Chain E., Fang W., Miang Y y Tian J., (2021). Computer vision approaches for detecting missing barricades. *Automation in Construction*, 131, 103852. <https://doi.org/10.1016/j.autcon.2021.103862>
- Chakradeo K., Delves M. y Titarenko S., (2019). Malaria Parasite Detection Using Deep Learning Methods. *2019 International Conference on Communication and Signal Processing (ICCSP)*, 15(2), 175-182. <https://doi.org/10.1016/j.jns.2017.08.946>
- Goodfellow I., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S, Courville A. y Bengio Y., (2014). Generative adversarial networks. *Advances in neural information processing systems*, 27, 2672–2680.
- Huq A. y Pervin T., (2020). Robust Deep Neural Network Model for Identification of Malaria Parasites in Cell Images. *2020 IEEE Region 10 Symposium (TENSYMP)*, 1456-1459, <https://doi.org/10.1109/TENSYMP50017.2020.9230832>.
- Krizhevsky A. Sutskever I. y Hinton G., (2012). ImageNet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 1097-1105.
- Meda K., Milla S. y Rostad B., (2021). Artificial intelligence research within reach: an object detection model to identify rickets on pediatric wrist radiographs, *Pediatric Radiology*, 51, 782-791. <https://doi.org/10.1007/s00247-020-04895-8>
- Nakasi R., Alija E. y Nakatumba J., (2021). A Poster on Intestinal Parasite Detection in Stool Sample Using AlexNet and GoogleNet Architectures, *ACM SIGCAS Conference on Computing and Sustainable Societies (COMPASS)*, 389-395. <https://doi.org/10.1145/3460112.3472309>
- OMS (2 de marzo 2020). Helminthiasis transmitidas por el suelo. <https://www.who.int/es/news-room/fact-sheets/detail/soil-transmitted-helminth-infections>
- Pan S. y Yang Q., (2009). A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*. 22(10), 1345-1359. <https://doi.org/10.1109/TKDE.2009.191>
- Prakash R. y Chithaluru P., (2020). Active Security by Implementing Intrusion Detection and Facial Recognition. *Nanoelectronics, Circuits and Communication Systems*, 692, 1-7. https://doi.org/10.1007/978-981-15-7486-3_1
- Radford A. y Metz L., (2015). Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv Preprint*.
- Shah D., Kawale K., Shah M., Randive S. y Mapari R., (2020). Malaria Parasite Detection Using Deep Learning. *2020 4th Proceedings of the International Conference on Intelligent Computing and Control Systems (ICICCS)*, 984-988. <https://doi.org/10.1109/ICICCS48265.2020.9121073>
- Shorten C. y Khoshgoftaar T., (2019). A survey on Data Augmentation for Deep Learning. *Journal of Big Data*, 6(60), 1-48. <https://doi.org/10.1186/s40537-019-0197-0>
- Shorten C., Maqsood A., Shahid M., Hassan M. y Grzegorzec M., (2021). Deep Malaria Parasite Detection in Thin Blood Smear Microscopic Images. *Applied Sciences*, 11(5), 2284. <https://doi.org/10.3390/app11052284>
- Song C., Wang C. y Yang Y., (2020). Automatic Detection and Image Recognition of Precision Agriculture for Citrus Diseases. *2020 IEEE Eurasia Conference on OIT, Communication and Engineering*

(ECICE), 187-190. <https://doi.org/10.1109/ECICE50847.2020.9301932>

Suriva M., Chandran V. y Sumithra M., (2019). Enhanced deep convolutional neural network for malaria parasite classification. *International Journal of Computers and Applications*, 1-10. <https://doi.org/10.1080/1206212X.2019.1672277>

Var E. y Boray F., (2018). Malaria Parasite Detection with Deep Transfer Learning. *2018 3rd International conference on computer science and engineering (UBMK)*, 298-302. <https://doi.org/10.1109/UBMK.2018.8566549>

Weiss K., Khoshgoftaar T. y Wang D., (2016). A survey of transfer learning. *Journal of Big Data*. 3(9), 1-40. <https://doi.org/10.1186/s40537-016-0043-6>

Zhuang F., Qi Z., Duan K., Xi D., Zhu Y., Zhu H., Xiong H y He Q., (2020), A Comprehensive Survey on Transfer Learning. *Proceedings of the IEEE*, 109(1), 43-76. <https://doi.org/10.1109/JPROC.2020.3004555>



Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-CompartirIgual 2.5 México.