



**Pontificia Universidad
Católica del Ecuador**
Seréis mis testigos

IBARRA

GESTIÓN, INFRAESTRUCTURA Y APLICACIONES EN SISTEMAS DE TECNOLOGÍA DE INFORMACIÓN

Dulce Rivero, Francklin Rivas, Stalin Arciniegas

 **Publicaciones**
CENTRO DE PUBLICACIONES
PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR

CRÉDITOS

Pontificia Universidad Católica del Ecuador - Sede Ibarra Ibarra:
Av. Jorge Guzmán Rueda y Av.
Aurelio Espinosa Pólit. Cdla. "La Victoria"
Teléfono: 06 2615 500 / 06 2615 631
Fax: (593)6- 2615 446 Apartado Postal 10.01.12
Web Site: www.pucesi.edu.ec Email: prorect@pucesi.edu.ec

SELLO EDITORIAL

Centro de Publicaciones PUCE
Web Site: www.edipuce.edu.ec
Quito, Av. 12 de octubre y Robles Apartado N° 17-01-2184
Telf. (5932) 2991 700
E-mail: publicaciones@puce.edu.ec
Primera edición copyright © 2023

Título: Gestión, infraestructura y aplicaciones en Sistemas de Tecnologías de la Información

Editores: Dulce Rivero, Francklin Rivas y Stalin Arciniegas

Revisión de estilo y redacción: Gabriela Garcés

Concepto gráfico y diagramación: Oswaldo Portilla

ISBN: 9789978375648

REVISIÓN DE PARES

Cada capítulo del presente libro fue sometido al debido arbitraje y dictamen de pares evaluadores expertos en el área del conocimiento.

ISBN: 978-9978-375-64-8



9 789978 375648

Prólogo

La transformación digital es un proceso sobre el cual se había estado estudiando, planificando, y discutiendo desde hace tiempo, pero es importante destacar el papel que tuvo la pandemia en su aceleración. En todos los entornos fue necesario realizar dicha transformación, y está aún por evaluarse el impacto definitivo en cada uno de los contextos. Los entornos educativos, industriales, comerciales e inclusive los familiares y cotidianos fueron afectados significativamente por la pandemia y tuvimos que migrar de forma abrupta y adoptar las herramientas tecnológicas como mecanismos de poder sobrellevar las actividades de forma remota, con la ayuda de las plataformas que permitían desde el teletrabajo, educación en línea, adquisición de bienes y servicios, seguimiento epidemiológico, atención médica especializada y de forma no menos importante, el poder tener mecanismos de comunicación con los seres queridos y tener acceso a la información tanto de la evolución de la situación sanitaria como de todos los temas que puedan ser de interés.

Todo lo anterior muestra la importancia que tiene poder estudiar de forma integral y sistemática las tecnologías de información. Por ello, este libro es una herramienta que permitirá a los lectores conocer desde los temas asociados a las plataformas tecnológicas requeridas para una apropiada conexión, la integración de dichas plataformas, el uso de las actuales metodologías y herramientas de desarrollo de software, analítica descriptiva hasta los conceptos avanzados asociados a los sistemas recomendadores y de gestión a través de una eficiente asignación de tareas y recursos.

Los 7 capítulos del libro permitirán tener esa visión integral, imprescindible para conocer de forma global los avances de las tecnologías de información y poder tener los elementos que conlleven a los nuevos estudios y desafíos que los tiempos presentes y futuros nos esperan sobre este apasionante mundo. Los invitamos a disfrutar de los aportes de los diversos autores y deleitarse con las investigaciones de alta calidad realizadas en nuestro entorno universitario.

Los editores

INTRODUCCIÓN

La creciente demanda de personal calificado en el campo de las tecnologías de la información y la comunicación ha ocasionado que la oferta en carreras a nivel técnico y universitario en este campo sea cada vez mayor y con diferentes propuestas de estudios, algunas orientadas a la gestión y al desarrollo de sistema, otras enfocadas al diseño y gestión de infraestructura y aplicaciones en sistemas. Sin embargo, todas estas carreras necesitan formación en las áreas definidas en el campo de las Tecnologías de la Información y Comunicación. Sin embargo, a pesar de las diferencias curriculares, sus planes de estudio deben contener asignaturas de programación, ingeniería de software, inteligencia artificial, matemáticas y redes entre otros. Por ello, este libro surge con el propósito de ofrecer al lector una visión amplia de los campos que se abordan en estas carreras.

El libro está orientado a estudiantes de maestría y de pregrado que se encuentre en niveles superiores de la carrera, el mismo está compuesto por siete capítulos y cada capítulo se divide en dos secciones, en la primera se presentan los conceptos y fundamentos teóricos de un tema específico del área de conocimiento en los que se presentan las principales características, manejo y métodos para abordar problemas en ese campo. En la segunda sección se presenta un caso de estudio en el que se aplican estos conceptos.

Esta obra recopila las aportaciones de un grupo de docentes especializados en las diferentes áreas de conocimiento en el campo de las tecnologías de la información, tales como matemáticas, analítica de datos, gestión de proyectos, bases de datos y comunicaciones. En el primer capítulo se formaliza el problema de asignación de recursos usando teoremas y fundamentos de matemática discreta. Los modelos de recomendación basados en algoritmos de analítica predictiva son presentados en el capítulo 2 mientras que en el capítulo 3 se presentan los conceptos de la analítica descriptiva y su aplicación. En el capítulo cuatro

se formalizan los conceptos relacionados al manejo de grandes volúmenes de datos. El marco de trabajo de desarrollo y puesta en operación de proyectos se presenta en el capítulo cinco y en el seis se presenta como desarrollar una aplicación orientada a servicios. Por último, el capítulo siete aborda aspectos del área de las comunicaciones.

El libro persigue ser un instrumento de enseñanza para reforzar los conocimientos y habilidades en el campo de las tecnologías de manera de complementar su formación, brindando una visión general de alguno de los campos donde se puede desarrollar profesionalmente.

ÍNDICE

Una herramienta computacional para asignaciones de recursos no divisibles	7
Sistemas de Recomendación: Enfoques y Técnicas	53
Analítica Descriptiva. Procesamiento y visualización de datos	79
Big Data: Ecosistema Hadoop	103
La administración de proyectos y el sistema ágil, ¿cuál es la tendencia en la gestión ágil de proyectos de software?	123
Integración de sistemas utilizando Servicio Web SOAP	145
Procedimiento para la instalación y configuración de canales satelitales libres sobre linux	163

Una herramienta computacional para asignaciones de recursos no divisibles

Guido Tapia¹, Franklin Camacho¹, Rigoberto Fonseca-Delgado¹,
Israel Pineda² y Cristopher Zhunio³

guido.tapia@yachaytech.edu.ec, fcamacho@yachaytech.edu.ec,
rfonseca@yachaytech.edu.ec, ipineda@usfq.edu.ec, czhunio@libelulasoft.com

¹ Universidad de Investigación de Tecnología Experimental Yachay, Escuela de Ciencias Matemáticas y Computacionales.

² Universidad San Francisco de Quito, Quito, Ecuador

³ Libélula Soft, Ecuador.

Introducción

Asignar un grupo de recursos a un grupo de agentes es una situación muy común en nuestro día a día. ¿Cómo asignar las tareas del hogar entre los miembros de una familia?, ¿cómo una institución de un gobierno puede distribuir su presupuesto? Sin duda alguna es un problema común, fácil de plantear, pero quizás, complicado de resolver. Este problema, por lo general, tiene tres ingredientes: un conjunto finito de agentes, un grupo limitado de recursos y preferencias individuales. El conglomerado de agentes, dependiendo de la aplicación del problema, representa: individuos, objetos, instituciones del gobierno, entre otros. Los recursos por repartir pueden ser divisibles o indivisibles.

Un recurso divisible puede separarse y se asigna a diferentes agentes; mientras que, un recurso indivisible no. Un ejemplo con recursos divisibles es repartir un pastel entre un cierto número de individuos. Por otro lado, un modelo con recursos indivisibles son los bienes de una herencia: una casa, un vehículo, entre otros.

Las preferencias individuales definen las necesidades, los gustos, los intereses o requerimientos que tienen los agentes sobre los subconjuntos de los insumos. Estas preferencias se establecen desde un punto de vista cualitativo o cuantitativo. En el enfoque cuantitativo, las preferencias individuales son dadas a través de

funciones numéricas conocidas como funciones de utilidad. A partir de las preferencias individuales se busca la satisfacción grupal e individual.

La satisfacción grupal, también conocida como eficiencia u óptima de Pareto, busca establecer cuando una asignación beneficia al grupo, garantizando que no existe otra concesión que mejore a un agente sin perjudicar a otro.

La satisfacción individual, conocida como justicia, busca asignaciones que disminuyan la envidia entre los individuos involucrados. Cuando esta propiedad se alcanza se dice que la asignación es libre de envidia (EF, por sus siglas en inglés); sin embargo, en caso de bienes indivisibles esta propiedad no puede garantizarse. Una versión débil de esta propiedad es la libre de envidia de hasta un recurso (EF1). Caragiannis et al. (2019) demostraron que, bajo funciones de utilidad aditivas, asignaciones que sean EF1 y óptima de Pareto pueden encontrarse; pero, su complejidad computacional es alta (Ramezani & Endriss, 2010).

Las funciones de bienestar social son índices que clasifican a todas las asignaciones, estas tratan de capturar la satisfacción grupal. Si se quieren analizar los criterios de justicia y eficiencia comparando el bienestar social de cada par de asignaciones, es complicado y quizás innecesario debido a que el conjunto de todas las posibles asignaciones puede ser muy grande, ver (de Keijzer, Bouveret, Klos, & Zhang, 2009). Por esta razón, se quiere reducir el tamaño del espacio de búsqueda y considerar el conjunto formado por todas las asignaciones que maximizan algunas de estas funciones de bienestar.

Una herramienta computacional que permita definir agentes, preferencias, utilidades, funciones de bienestar social y además facilite el análisis de la eficiencia y justicia es de gran utilidad en este campo. RAPL es un lenguaje de programación que satisface estos requerimientos, la primera versión fue propuesta en (Pineda, Zhunio, Camacho, & Fonseca-Delgado, 2020).

En este capítulo se desarrollan dos aspectos fundamentales: una revisión teórica del problema de asignación de recursos y una descripción elemental de un lenguaje de programación de dominio específico (RAPL). Este se distribuye en cinco secciones adicionales: la primera (Sección 2) revisa los fundamentos teóricos relacionados con la asignación de recursos. La siguiente sección (Sección 3) presenta un enfoque matricial del problema, el cual es un eslabón entre los fundamentos teóricos y la herramienta computacional. Una sección (Sección 4) que describe el lenguaje de programación RAPL. Finalizando con las conclusiones y la bibliografía.

2. Fundamentos teóricos del problema de asignación de recursos

En esta sección se describen las nociones básicas del problema de asignación de recursos no divisibles.

2.1. Agentes, recursos y asignaciones

Los agentes pueden ser: personas, objetos, máquinas, entre otros. Se denota por N al conjunto finito de todos los agentes, $N = \{1, \dots, n\}$. Los recursos son los aspectos o factores para distribuir y dependen de la naturaleza del problema; pueden estar relacionados con factores humanos, materiales, financieros, técnicos, entre otros. El conjunto finito formado por todos los recursos no divisibles se denota con $M = \{1, \dots, m\}$ (Endriss U. , 2010). El conjunto formado por todos subconjuntos de M se denota por $P(M)$ y una partición de tamaño n de M es una familia $\{A_1, A_2, \dots, A_n\}$ donde $A_1, A_2, \dots, A_n \subseteq M$ tales que $\bigcup_{i=1}^n A_i = M$ y para todo $i, j \in N$ con $i \neq j, A_i \cap A_j = \emptyset$. A continuación, la definición de una asignación:

Definición 1. Una asignación A de los recursos M en los agentes N , o simplemente una asignación, es una partición de M , $\{A_1, A_2, \dots, A_n\}$, representada a través de una n -úpla,

$$A = (A_1, \dots, A_n)$$

donde para cada $i \in N$, A_i son los recursos asignados al agente i . El conjunto formado por todas las asignaciones se denota por N^M .

Obsérvese que en una partición $\{A_1, A_2, \dots, A_n\}$ de M , se garantiza que:

- todos los recursos son repartidos, es decir, $\forall r \in M, \exists i \in N$ tal que $r \in A_i$, y, además
- ningún recurso es repartido dos veces ya que $A_i \cap A_j = \emptyset$.

2.2. Funciones de utilidad

La preferencia individual de un agente i sobre $P(M)$ comúnmente se define como un preorden total. Esto es, una relación binaria que es transitiva y total. Por lo general, las preferencias individuales se establecen a través de funciones numéricas conocidas como funciones de utilidad.

Definición 2. Cualquier función μ de $P(M)$ a los números reales, $\mu: P(M) \rightarrow R$, se conoce como función de utilidad. El número $\mu(R)$ representa la utilidad del conjunto R . Además, una función de utilidad μ se dice que:

- es no negativa si, y sólo si, $\mu(R) \geq 0$ para todo $R \in P(M)$,
- es positiva si, y sólo si, μ es no negativa y para todo $R \in P(M)$ con $R \neq \emptyset$ se tiene que $\mu(R) > 0$,
- es aditiva si, y sólo si, μ es no negativa, $\mu(\emptyset) = 0$, y para todo $R \in P(M)$

$$\mu(R) = \sum_{r \in R} \mu(\{r\})$$

- tiene presupuesto K , con $K \in R$ y $K \geq 0$ si, y sólo si, $\mu(M) = K$

Obsérvese que si μ es aditiva con presupuesto K , entonces

$$\mu(M) = \sum_{r \in M} \mu(\{r\}) = K \tag{1}$$

Lema 1. Dado un agente $i \in N$ y μ_i una función de utilidad asociada a i , entonces la relación \geq_i sobre $P(M)$ dada por: $\forall A, B \in P(M)$,

$$A \geq_i B \Leftrightarrow \mu_i(A) \geq \mu_i(B) \quad (2)$$

es una preferencia individual del agente i .

Demostración. Se quiere demostrar que \geq_i es un preorden total. Sean $A, B, C \in P(M)$. Primero se demuestra la totalidad. Como $\mu_i(A), \mu_i(B) \in R$, por la tricotomía de los números reales, se tiene que $\mu_i(A) \geq \mu_i(B)$ o bien $\mu_i(B) \geq \mu_i(A)$. Luego por la ecuación (2), $A \geq_i B$ o $B \geq_i A$. Así, \geq_i es la total. Para la transitividad supóngase que $A \geq_i B$ y $B \geq_i C$. Por la ecuación (2), $\mu_i(A) \geq \mu_i(B)$ y $\mu_i(B) \geq \mu_i(C)$. Por la transitividad de \geq en R , $\mu_i(A) \geq \mu_i(C)$. Nuevamente por la ecuación (2), $A \geq_i C$. Así, \geq_i es transitiva, se ha demostrado que \geq_i es total y transitiva; luego, \geq_i es un preorden total.

Ejemplo 1. Pablo (1) y María (2) han ganado un concurso y se han hecho acreedores de 4 helados, cada uno de diferente sabor. Los sabores son: chocolate (c), mora (m), vainilla (v), y fresa (f). En este caso, $N=\{1,2\}$ y $M=\{c,m,v,f\}$, existen n^m posibles asignaciones, para este ejemplo $|N^M|=2^4=16$. Supongamos que Pablo, el agente 1, considera la función de utilidad, μ_1 , dada por

$$\mu_1(\{c\})=\mu_1(\{m\})=\mu_1(\{v\})=\mu_1(\{f\})=2.5;$$

es decir, a partir de la ecuación (2)

$$\{c\} \sim_1 \{m\} \sim_1 \{v\} \sim_1 \{f\};$$

y para todo $R \in P(M)$, $\mu_1(R)=\sum_{r \in R} \mu_1(\{r\})$. Es decir, μ_1 es una función de utilidad aditiva. Por otro lado; María, el agente 2, tiene una función de utilidad aditiva, μ_2 , dada por

$$\mu_2(\{c\})=\mu_2(\{m\})=\mu_2(\{v\})=2 \text{ y } \mu_2(\{f\})=4;$$

en este caso;

$$\{f\} >_2 \{m\} \sim_2 \{v\} \sim_2 \{c\}.$$

Obsérvese que, tanto μ_1 como μ_2 tiene como presupuesto $K=10$. Ahora bien, para la asignación $C=(\{c\},\{f,m,v\})$, el agente 1, "evalúa" lo que le asignaron con 2.5; es decir, $\mu_1(C_1)=\mu_1(\{c\})=2.5$. Mientras que él evalúa con 7.5 lo asignado a María, $\mu_1(C_2)=\mu_1(\{f,m,v\})=7.5$. Pablo da más utilidad a lo asignado a María, pues $\mu_1(C_2) > \mu_1(C_1)$.

Eficiencia y justicia

Al momento de repartir m recursos indivisibles entre n agentes se pueden calcular n_m posibles asignaciones. Por supuesto, no todas las asignaciones son buenas, pero ¿cómo se puede determinar esto? En esta sección se estudian propiedades para determinar cuando una asignación es buena. Estas propiedades están relacionadas con la eficiencia y la justicia. Por otro lado, se asume que para todo $i \in N$, las preferencias individuales \geq_i son dadas a través de funciones de utilidad, μ_i . Es decir, \geq_i están dadas por la ecuación (2). Luego, para cada $i \in N$, se hará referencia a la función de utilidad μ_i en lugar de la relación \geq_i .

2.2.1. Eficiencia – óptima de Pareto

La eficiencia de una asignación será medida de acuerdo con el criterio de la eficiencia de Pareto, también conocida como óptimo de Pareto. Este criterio recibe su nombre en memoria de Vilfredo Pareto (1848-1923). Consiste en maximizar una relación de satisfacción social sobre todas las asignaciones. Esta satisfacción se mide comparando dos asignaciones A y B , a partir de las preferencias individuales de cada agente y sus correspondientes evaluaciones, esto es:

12

“ A es socialmente preferida a B , si existe un agente que mejore la utilidad de su valuación en la asignación B , con respecto a lo asignado por A y ningún agente empeora su evaluación en B con respecto a A ”. La definición es la siguiente:

Definición 3. Dados $A, B \in N^M$ y $i, j \in N$. A es Pareto dominada por B si, $\mu_i(A_i) \leq \mu_i(B_i)$ y existe al menos un agente, j , donde $\mu_j(A_j) < \mu_j(B_j)$.

A continuación, se muestra un ejemplo donde una asignación domina a otro en el sentido de Pareto.

Ejemplo 2. Se desean repartir 3 recursos entre 3 agentes; es decir, $M = \{a, b, c\}$ y $N = \{1, 2, 3\}$. Cada agente $i \in N$ considera funciones de

utilidad aditivas μ_i dadas de la siguiente manera

$$u_1(\{a\})=20, u_1(\{b\})=30, u_1(\{c\})=50;$$

$$u_2(\{a\})=15, u_2(\{b\})=50, u_2(\{c\})=35;$$

$$u_3(\{a\})=80, u_3(\{b\})=20, u_3(\{c\})=0.$$

Obsérvese que las funciones de utilidades tienen presupuesto $K=100$ y $\mu_i(\emptyset)=0$ para todo i . Considérese las siguientes asignaciones:

$$A=(A_1, A_2, A_3)=(\{a, b\}, \emptyset, \{c\})$$

$$B=(B_1, B_2, B_3)=(\{c\}, \{b\}, \{a\})$$

Las valuaciones de cada agente sobre las asignaciones A y B están dadas por:

$$u_1(A_1)=u_1(\{a, b\})=u_1(\{a\})+u_1(\{b\})=50, u_1(B_1)=u_1(\{c\})=50;$$

$$u_2(A_2)=u_1(\emptyset)=0, u_2(B_2)=u_2(\{b\})=50;$$

$$u_3(A_3)=u_1(\{c\})=0, u_3(B_3)=u_3(\{a\})=80.$$

El primer agente, en las asignaciones A y B , tiene igual utilidad; pero, en B el segundo y tercer agente mejoran su utilidad. Así, A es Pareto dominada por B . Para obtener la satisfacción social se busca asignaciones que no sean dominadas por ninguna otra asignación, en el sentido de Pareto. A continuación, la definición formal dada en (Endriss, 2010).

Definición 4. Dados $A, B \in N^M$, se dice que B es óptima de Pareto, PO , si no es Pareto dominada por otra asignación A .

Las asignaciones PO también son llamadas eficientes en el sentido de Pareto o simplemente eficientes. Se denota como PO , al conjunto de todas las asignaciones PO . Caragiannis et al. (2019) demostraron que, bajo funciones de utilidad aditivas, siempre existe una asignación PO en N^M ; es decir, $PO \neq \emptyset$. La figura 1 muestra que las asignaciones en N^M pueden agruparse en dos conjuntos. El primero, PO conformado por las asignaciones PO , y el segundo, PO^c con las asignaciones que no son PO . En otras palabras, $N^M = PO \cup PO^c$.

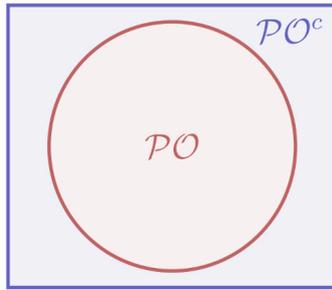


Figura 1. $N^M = PO \cup PO^c$

Las asignaciones PO más fáciles de hallar son aquellas que reparten todos los recursos a un solo agente.

Ejemplo 3. Considérese nuevamente el ejemplo 2, $M = \{a, b, c\}$ y $N = \{1, 2, 3\}$. Existen $3^3 = 27$ posibles asignaciones. Considérese C la asignación que reparte todos los recursos al agente 1; es decir, $C = (M, \emptyset, \emptyset)$. En este caso,

$$\mu_1(C_1) = \mu_1(M) = 100, \mu_2(C_2) = \mu_2(\emptyset) = 0, \text{ y } \mu_3(C_3) = \mu_3(\emptyset) = 0.$$

Como la función de utilidad del agente 1 es positiva, entonces cualquier asignación que mejore al agente 2 o al agente 3, perjudica al agente 1; luego, C es PO . Otra asignación PO es $B = (\{c\}, \{b\}, \{a\})$; una manera de verificar esto, es comprobado que ninguna de las 26 asignaciones domina a B . Como $A = (\{a, b\}, \emptyset, \{c\})$ es Pareto dominada por B , entonces A no es PO .

14

En el ejemplo anterior se observaron las asignaciones B y C que son PO y la asignación A que no es PO . A partir de estas asignaciones que no son PO es posible llegar a asignaciones PO , a través de negociaciones entre agentes (Endriss, Maudet, Sadri, & Toni, 2006).

Pareciera atractivo determinar que una asignación es buena si satisface PO . Sin embargo, esta propiedad no considera la satisfacción individual de los agentes; es decir, es indiferente a la justicia. Como se puede ver en el ejemplo 3, la asignación

C , que es PO , no reparte ningún recurso a los agentes 2 y 3. Si bien esta asignación satisface el criterio de eficiencia, se puede observar que no todos los agentes quedan conformes. En general, determinar cuándo una asignación está en PO es un problema complicado; debido a que se requiere comparar todas las posibles asignaciones. Esto incrementa la complejidad computacional del problema (de Keijzer, Bouveret, Klos, & Zhang, 2009).

Es conveniente señalar dos conclusiones importantes: primero, puede existir más de una asignación que sea óptima de Pareto dentro de un mismo problema. Segundo, este criterio no es suficiente para decidir o etiquetar a una asignación como *buena*. Se necesitan más criterios para clasificar asignaciones. En la siguiente subsección se estudia la propiedad de libre envidia entre agentes.

2.2.2. Justicia - Libre de envidia

Se desea encontrar un criterio que compare las utilidades de los agentes dentro de una misma asignación y evalúe la “satisfacción” de cada uno. Este criterio establece la justicia de la asignación. Ahora bien, la justicia y la eficiencia, PO , serán las condiciones para determinar cuando una asignación es buena.

Como se ha mencionado antes, un criterio relacionado con la justicia es la propiedad libre de envidia (Endriss, 2010). Una asignación se dice que es libre de envidia si todos los agentes valoran igual o mejor al conjunto de recursos asignados, que al conjunto de recursos asignados a los otros agentes. Esto describe la satisfacción de los agentes con la asignación. Desafortunadamente, en el caso de recursos indivisibles esta propiedad no siempre existe. A continuación, la definición formal de una asignación con libre envidia.

Definición 5. Una asignación A es libre de envidia, EF , si para todo $i, j \in N, \mu_i(A_i) \geq \mu_i(A_j)$.

Obsérvese que si existe un agente $i \in N$ tal que $\mu_i(A_i) < \mu_i(A_j)$ para algún $j \in N$, entonces el agente i envidia al agente j . Por otro lado, se denota con EF al conjunto formado por todas las asignaciones EF . En el siguiente ejemplo se estudia un problema donde $EF \neq \emptyset$.

Ejemplo 4. Se desean repartir 3 recursos entre 2 agentes; en este caso, $M = \{a, b, c\}$ y $N = \{1, 2\}$. Supóngase que ambos agentes consideran funciones de utilidad aditivas dadas por:

$$\mu_1(\{a\}) = 20, \mu_1(\{b\}) = 30, \mu_1(\{c\}) = 50;$$

$$\mu_2(\{a\}) = 15, \mu_2(\{b\}) = 50, \mu_2(\{c\}) = 35.$$

Como para todo i, μ_i es aditiva, entonces $\mu_i(\emptyset) = 0$. Considérese las asignaciones:

$$A = (A_1, A_2) = (\{a, c\}, \{b\});$$

$$B = (B_1, B_2) = (\{b\}, \{a, c\})$$

las valuaciones de cada agente sobre A están dadas por:

$$\mu_1(A_1) = \mu_1(\{a, c\}) = \mu_1(\{a\}) + \mu_1(\{c\}) = 20 + 50 = 70, \mu_1(A_2) = \mu_1(\{b\}) = 30;$$

$$\mu_2(A_1) = \mu_2(\{a, c\}) = \mu_2(\{a\}) + \mu_2(\{c\}) = 15 + 35 = 50, \mu_2(A_2) = \mu_2(\{b\}) = 50.$$

Luego,

$$70 = \mu_1(A_1) > \mu_1(A_2) = 30 \text{ y } 50 = \mu_2(A_2) = \mu_2(A_1) = 50$$

ningún agente envidia al otro. Esto es, $A \in EF$. Mientras que las valuaciones sobre B están dadas por:

$$\mu_1(B_1) = 30, \mu_1(B_2) = 70 \text{ y } \mu_2(B_1) = 50, \mu_2(B_2) = 50.$$

16

De manera que,

$$\mu_1(B_1) < \mu_1(B_2) \text{ y } \mu_2(B_2) = \mu_2(B_1)$$

así, el agente 1 envidia al agente 2. De manera que, $B \notin EF$.

Otro ejemplo donde $EF \neq \emptyset$ es el ejemplo 3, donde la asignación $B \in EF$ mientras que $A, C \in EF$ (en ambas asignaciones el agente 2 envidia al menos al agente 1). En dicho ejemplo, la asignación C es eficiente, PO , pero no es justa no es EF . A no es ni eficiente, no PO , ni justa no es EF . Finalmente, la asignación B es eficiente, PO , y justa, EF , se puede concluir que la asignación B es buena.

Desafortunadamente, cuando se consideran recursos no divisibles y utilidades aditivas, existen problemas donde no es posible encontrar alguna asignación EF . Por ejemplo, si $|N| > |M|$, siempre existe un agente que envidia en la repartición; en este caso $EF = \emptyset$. Por este motivo, la propiedad EF no siempre existe y, por tal razón, no es posible establecer la justicia de una asignación. A continuación, se presenta una versión relajada de la propiedad libre de envidia conocida como libre de envidia de hasta un bien (Caragiannis et al., 2019).

Libre de envidia de hasta un bien (EF1)

La propiedad libre de envidia de hasta un recurso es una propiedad que establece cuando la envidia entre los agentes es mínima; en el sentido que, si existe envidia entre un par de agentes, la misma desaparece eliminando un recurso. Vale aclarar que, cuando se dice eliminar un recurso, no se hace referencia a quitar ese recurso del problema; esto se refiere a que se garantiza que los agentes están a un recurso de estar satisfecho con los recursos asignados. A continuación, la definición formal.

Definición 6. Una asignación A se dice que es libre de envidia de hasta un bien, $EF1$, si para todo par de agentes $i, j \in N$, existe $g \in A_j$ tal que $\mu_i(A_i) \geq \mu_i(A_j \setminus \{g\})$.

Se denota por EFO al conjunto formado por todas las asignaciones $EF1$. A diferencia de la libre de envidia, la libre de envidia de hasta un bien siempre existe, ver (Caragiannis, et al. 2019). Es decir, $EFO \neq \emptyset$. Por tal razón, se asume que una asignación es justa si ella está en EFO . En la figura 2 muestra que N^M se clasifica en dos conjuntos: EFO (justas) y EFO^c (no justas), donde EFO^c son todas las asignaciones que no son $EF1$. Es decir, $N^M = EFO \cup EFO^c$.

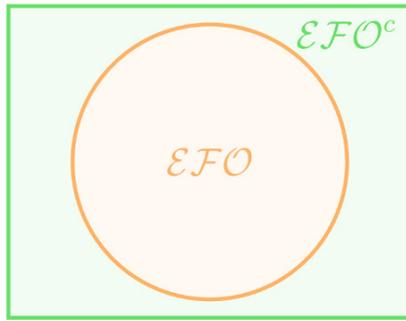


Figura 2. $N^M = EFO \cup EFO^c$.

Claramente, se tiene que $EF \subseteq EFO$.

Lema 2. Supongamos que todos los agentes consideran funciones de utilidad aditivas, sea $A \in N^M$. Si A es libre de envidia, entonces A es libre de envidia de hasta un recurso.

Demostración. Si $EF = \emptyset$, el resultado es directo, $EF \subseteq EFO$. Supongamos que $EF \neq \emptyset$. Sea $A \in EF$, entonces para todo $i, j \in N$ se tiene que $\mu_i(A_i) \geq \mu_i(A_j)$. Si $g \in A_j$, entonces, por la aditividad de μ_i ,

$$\mu_i(A_j) = \sum_{s \in A_j} \mu_i(\{s\}) \geq \sum_{s \in A_j \setminus \{g\}} \mu_i(\{s\}) = \mu_i(A_j \setminus \{g\})$$

así, para cualquier $g \in A_j$

$$\mu_i(A_i) \geq \mu_i(A_j \setminus \{g\})$$

de manera que, $A \in EFO$. Luego, $EF \subseteq EFO$.

A continuación, un ejemplo.

Ejemplo 5. Considérese nuevamente el ejemplo 4, $M = \{a, b, c\}$ y $N = \{1, 2\}$, ambos agentes tienen funciones de utilidad aditivas dadas por:

$$\begin{aligned} \mu_1(\{a\}) &= 20, \mu_1(\{b\}) = 30, \mu_1(\{c\}) = 50; \\ \mu_2(\{a\}) &= 15, \mu_2(\{b\}) = 50, \mu_2(\{c\}) = 35. \end{aligned}$$

Como $A=(A_1,A_2)=(\{a,c\},\{b\})\in EF$; por el Lema 2, $A \in EFO$. Por otro lado, la asignación $B=(B_1,B_2)=(\{b\},\{a,c\})\notin EF$. Veamos que B es EF 1. En efecto, el agente 1 envidia al agente 2 $\mu_1(B_1)=\mu_1(\{b\})=30 < 70=\mu_1(\{a,c\})=\mu_1(B_2)$.

Se debe determinar que recurso en B_2 se puede suprimir para eliminar la envidia. Si a B_2 le quitamos el recurso $\{a\}$, la envidia no desaparece; pues, $\mu_1(B_1)=30$ y $\mu_1(B_2\setminus\{a\})=\mu_1(\{c\})=50$. Ahora bien, si se quita el recurso $\{c\}$, en lugar del recurso $\{a\}$, entonces la envidia desaparece; ya que, $\mu_1(B_1)=30$ y $\mu_1(B_2\setminus\{c\})=\mu_1(\{a\})=20$. Luego, el recurso que genera envidia en la asignación B es $\{c\}$. En el ejemplo 5, se observa dos asignaciones, A,B , que son EF 1. La asignación A es EF y en la repartición no existe un recurso que genere envidia entre los agentes; pero B no es EF y en la repartición existe un recurso que produce envidia en el agente 1. La propiedad EF 1, permite determinar estos *recursos problemáticos*.

Funciones de bienestar social

Se estudiarán algunos métodos para ordenar las asignaciones usando las preferencias individuales de los agentes. Estos métodos consisten en definir relaciones binarias \succsim sobre N^M . Para ello se definen las funciones de bienestar social. Veamos formalmente la definición:

Definición 7. Una función de bienestar sobre N^M es cualquier función $SW:N^M \rightarrow \mathbb{R}$, donde el número $SW(A)$ representa el bienestar social de la asignación A .

Ahora bien, dada una SW se define la relación bienestar social \succsim de la siguiente manera:
para todo par $A,B \in N^M$,

$$A \succsim B \Leftrightarrow SW(A) \geq SW(B) \quad (3)$$

donde $A \succsim B$ es interpretado como "la asignación A tienen mejor o igual bienestar social que la asignación B ". La relación \succsim es un preorden total sobre N^M ; donde $>$ es la parte estricta y

~ es la parte indiferente. En algunos casos se usan subíndices para etiquetar la función de bienestar social que lo define. Es de interés considerar dos SW : el bienestar social utilitario y el bienestar social de Nash (Endriss, 2010).

2.2.3. Bienestar social utilitario

Como se muestra continuación, una bondad es que toda asignación que maximice este bienestar social es PO ; lo malo es que, no siempre son EF 1. A continuación, la definición del bienestar social utilitario:

Definición 8. Para todo $A \in N^M$, el bienestar social utilitario, SW_U , de A está definido como:

$$SW_U(A) = \sum_{i \in N} \mu_i(A_i) \quad (4)$$

donde para todo $i \in N$, μ_i es la función de utilidad.

Se denota por \succsim_U la relación de bienestar social que define a partir de SW_U , según la ecuación (3).

Ejemplo 6. Sean $M=\{a,b,c\}$ y $N=\{1,2\}$ y las funciones de utilidad aditivas dadas como:

$$\begin{aligned} \mu_1(\{a\})=25, \mu_1(\{b\})=35, \mu_1(\{c\})=40; \\ \mu_2(\{a\})=30, \mu_2(\{b\})=35, \mu_2(\{c\})=35. \end{aligned}$$

Considere las asignaciones:

$$A=(\{b\},\{a,c\}), B=(\{c\},\{a,b\}), C=(M,\emptyset), D=(\{a\},\{b,c\}).$$

Note que

$$\begin{aligned} \mu_1(\{a\})=25, \mu_1(\{b\})=35, \mu_1(\{c\})=40; \\ \mu_2(\{a,b\})=65, \mu_2(\{a,c\})=65, \mu_2(\{b,c\})=70. \end{aligned}$$

Luego,

$$\begin{aligned} SW_U(A) &= \mu_1(\{b\}) + \mu_2(\{a,c\}) = 35 + 65 = 100; \\ SW_U(B) &= \mu_1(\{c\}) + \mu_2(\{a,b\}) = 40 + 65 = 105; \\ SW_U(C) &= \mu_1(M) + \mu_2(\emptyset) = 100 + 0 = 100; \end{aligned}$$

$$SW_U(D) = \mu_1(\{a\}) + \mu_2(\{b,c\}) = 25 + 70 = 95.$$

De manera que,

$$SW_U(B) > SW_U(A) = SW_U(C) > SW_U(D)$$

así,

$$B >_U A >_U C >_U D$$

En este ejemplo, se observa que B produce más bienestar social utilitario que A, C y D; mientras que D produce el menor bienestar social de estas asignaciones. Se denota por MSW_U al conjunto de todas las asignaciones que maximizan el bienestar social utilitario; es decir,

$$MSW_U = \{A \in N^M : SW_U(A) \geq SW_U(B), \forall B \in N^M\}$$

Obsérvese que $MSW_U \neq \emptyset$; pues, N^M es finito y para todo $B \in N^M$ existe A tal que $SW_U(A) \geq SW_U(B)$. Luego, N^M se clasifica en las asignaciones que maximizan SW_U y las que no, esto se describe en la siguiente figura:

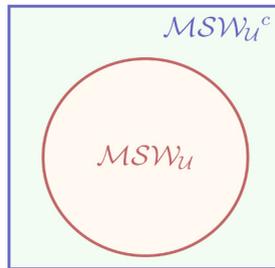


Figura 3. $N^M = MSW_U \cup MSW_U^c$

En la referencia Camacho, Zhunio, Fonseca y Castillo (2020), se demuestra que:

Teorema 1. Bajo funciones de utilidad aditivas, toda asignación en MSW_U es óptima de Pareto.

Encontrar asignaciones en MSW_U que sean EF1, no siempre es posible. Sin embargo, en caso de existir, encontrarlas es un problema NP-Completo (Aziz, Huang, Mattei, & Segal-Halevi, 2021).

2.2.4. Bienestar social de Nash

Al contrario de las asignaciones que maximizan el bienestar social utilitario, toda asignación que maximice el bienestar social de Nash es eficiente y justa.

Definición 9. Para todo $A \in N^M$, el bienestar social de Nash de A es definida como:

$$SW_{Nash}(A) = \prod_{i \in N} \mu_i(A_i)$$

donde para todo $i \in N$, μ_i es una función de utilidad.

La relación de bienestar social que se define, según la ecuación (3), a partir de SW_{Nash} se denota por \succsim_{Nash} .

Ejemplo 7. Sean $M=\{a,b,c\}$ y $N=\{1,2\}$ y las funciones de utilidad aditivas como:

$$\begin{aligned} \mu_1(\{a\})=25, \mu_1(\{b\})=35, \mu_1(\{c\})=40; \\ \mu_2(\{a\})=30, \mu_2(\{b\})=35, \mu_2(\{c\})=35. \end{aligned}$$

Considérese nuevamente las asignaciones

$$A=(\{b\},\{a,c\}), B=(\{c\},\{a,b\}), C=(M,\emptyset), D=(\{a\},\{b,c\})$$

como

$$\begin{aligned} \mu_1(\{a\})=25, \mu_1(\{b\})=35, \mu_1(\{c\})=40; \\ \mu_2(\{a,b\})=65, \mu_2(\{a,c\})=65, \mu_2(\{b,c\})=70 \end{aligned}$$

entonces

$$\begin{aligned} SW_{Nash}(A) &= \mu_1(\{b\}) \cdot \mu_2(\{a,c\}) = 35 \cdot 65 = 2275; \\ SW_{Nash}(B) &= \mu_1(\{c\}) \cdot \mu_2(\{a,b\}) = 40 \cdot 65 = 2600; \\ SW_{Nash}(C) &= \mu_1(M) \cdot \mu_2(\emptyset) = 100 \cdot 0 = 0; \\ SW_{Nash}(D) &= \mu_1(\{a\}) \cdot \mu_2(\{b,c\}) = 25 \cdot 70 = 1750. \end{aligned}$$

De manera que,

$$SW_{Nash}(B) > SW_{Nash}(A) > SW_{Nash}(D) > SW_{Nash}(C)$$

luego,

$$B \succ_{Nash} A \succ_{Nash} D \succ_{Nash} C.$$

En el ejemplo anterior se observa que el orden obtenido de las siguientes asignaciones A, B, C y D , al utilizar el SW_{Nash} , es diferente que el obtenido al utilizar el SW_U .

El conjunto de todas las asignaciones que maximizan el bienestar social de Nash se denota por MSW_N . Es decir:

$$MSW_N = \{A \in N^M : SW_{Nash}(A) \geq SW_{Nash}(B), \forall B \in N^M\}$$

como N^M es finito, entonces $MSW_N \neq \emptyset$. En el ejemplo anterior se observa que $MSW_N = \{B\}$.

Teorema 2. Bajo funciones de utilidad aditivas, una asignación $A \in MSW_N$ es libre de envidia de hasta un recurso y óptima de Pareto (Caragiannis et al., 2019).

Enfoque matricial

Tener una perspectiva matricial del problema de asignación de recursos no divisibles, conducirá a diseñar varios métodos para procesar la información agente-recurso y así presentar los resultados de manera intuitiva.

Con el fin de formalizar el enfoque propuesto, se establecen algunas notaciones; recordemos que $|M|=m$ y $|N|=n$. El conjunto de todas las matrices de tamaño $n \times m$ con entradas en el conjunto K , se denotará por $M_{n \times m}(K)$. Las filas de estas matrices representan a los agentes, y sus columnas representan a los recursos. Entonces, si $A \in M_{n \times m}(K)$, la posición (i,r) , es denotada por $[A]_{ir}$, y se refiere al agente i y al recurso r . Además, $[A]_{i*}$ denota la i -ésima fila de A y $[A]_{*r}$ denota la r -ésima columna de A .

Matriz de asignación

En esta subsección se definen las matrices que representan a todas las asignaciones. Estos tipos de matrices son binarias; por esta razón, se identifica estas matrices como un subconjunto de $M_{n \times m}(B)$ donde $B = \{0, 1\}$.

Definición 10. Sea $A = (A_1, A_2, \dots, A_n) \in N_M$ y $F \in M_{n \times m}(B)$. Se dice que la matriz F representa la asignación A , si

$$[F]_{ir} = \begin{cases} 1, & \text{si } r \in A_i \\ 0, & \text{en otro caso.} \end{cases} \quad (5)$$

Obsérvese que $[F]_{ir} = 1$ es interpretado como: el agente i le corresponde el recurso representado en la columna r . Mientras que $[F]_{ir} = 0$ el agente i no le corresponde el recurso representado en la columna r .

Ejemplo 8. Se desean repartir 4 recursos $M = \{a, b, c, d\}$ entre 3 agentes $N = \{1, 2, 3\}$. Considérese $F \in M_{3 \times 4}(B)$ dada como:

		Recursos			
Agentes		0	1	0	0
		0	0	1	0
		1	0	0	1

Obsérvese que las filas corresponden a los agentes y a las columnas a los recursos. El recurso "a" está representado en la columna 1, el "b" en la 2; "c" y "d" en las columnas 3 y 4, respectivamente. En la fila 1 y columna 2 está 1, $[F]_{12} = 1$, esto se interpreta como: "el agente 1 le corresponde el recurso b"; mientras que, $[F]_{33} = 0$, se interpreta como: "el agente 3 no le corresponde el recurso c". Más aún, la matriz F representa la asignación $A = (\{b\}, \{c\}, \{a, d\})$. Obsérvese que una matriz de asignación es una matriz binaria de 0's y 1's; debido a que, al trabajar con recursos indivisibles cada recurso debe ser asignado en su totalidad a un solo agente y la matriz, que representa una asignación, tendrá un solo 1 por

columna, para fijar ideas, ver matriz F en el ejemplo anterior. En el siguiente lema se demuestra que para cada asignación existe una única matriz que la representa.

Lema 3. Para todo $A \in N_M$, existe una única $F \in M_{n \times m}(B)$ que representa a A .

Demostración. Supongamos que existen dos matrices distintas F y G en $M_{n \times m}(B)$ tales que ambas representan a la asignación A . Como $F \neq G$ existen $r \in M$ e $i \in N$ tales que

$$[F]_{ir} \neq [G]_{ir}$$

luego, una de las dos matrices no representa a A . Cuando se dice que F es una matriz de asignación se refiere a que es la matriz que representa a una asignación. Es decir, F es una matriz de asignación de M recursos en N agentes, si para cada columna r con $1 \leq r \leq m$, existe solamente un agente i con $1 \leq i \leq n$ tal que $[F]_{ir} = 1$ and $[F]_{jr} = 0$ para todo $j \neq i$.

Matriz de valuación

A partir de las funciones de utilidad μ_i que define cada agente, se define la matriz de valuación. Esta matriz contiene los datos originales del problema. Es decir, dentro de un problema de asignación de recursos, la matriz de valuaciones alberga las utilidades que cada agente otorga a cada recurso. Como considerando utilidades aditivas, se denota con R^* al conjunto de los números reales no negativos, $R^* = \{x \in R : x \geq 0\}$. Veamos formalmente la definición de matriz de valuación:

Definición 11. Dada μ_i una función de utilidad de un agente $i \in N$. Una matriz $V \in M_{n \times m}(R^*)$ es una matriz de valuación si $[V]_{ir} = \mu_i(r)$ para todos los agentes $i \in N$ y para todo recurso $r \in M$.

Ejemplo 9. Se requiere repartir 4 recursos entre 3 agentes, $M = \{a, b, c, d\}$ y $N = \{1, 2, 3\}$. Los agentes definen funciones de utilidad aditivas dadas por:

$$\begin{aligned}\mu_1(\{a\})=30, \mu_1(\{b\})=40, \mu_1(\{c\})=20, \mu_1(\{d\})=10; \\ \mu_2(\{a\})=30, \mu_2(\{b\})=10, \mu_2(\{c\})=50, \mu_2(\{d\})=10; \\ \mu_3(\{a\})=30, \mu_3(\{b\})=24, \mu_3(\{c\})=16, \mu_3(\{d\})=30.\end{aligned}$$

La matriz V que se muestra a continuación, en cada fila se distribuye las preferencias

$$V = \begin{pmatrix} 30 & 40 & 20 & 10 \\ 30 & 10 & 50 & 10 \\ 30 & 24 & 16 & 30 \end{pmatrix}$$

numéricas de cada agente. En la posición $[V]_{13}$, representa la utilidad que da la agente 1 al recurso representado en la 3ra columna; es decir, $[V]_{13}=20=\mu_1(\{c\})$.

Cada recurso recibe cierta utilidad por cada agente, la suma de todas esas utilidades la llamaremos la cotización del recurso. En el siguiente lema se demuestra que a partir de la matriz de valuación se calcula el presupuesto de cada agente y la cotización de cada recurso.

Lema 4. Si V es la matriz de valuación de un problema de asignación de recurso, entonces la suma de todas las posiciones de la fila i , se obtiene el presupuesto de la utilidad del agente i ; la suma de todas las posiciones de la columna r , se obtiene que la cotización del recurso representado en la columna r .

26

Demostración. Sea $i \in N$ y propongamos que K_i es el presupuesto de μ_i , entonces

$$\sum_{r=1}^m [V]_{ir} = \sum_{r=1}^m \mu_i(r) = \sum_{r \in M} \mu_i(r) = K_i$$

Por otra parte, dado $r \in M$. Si C_r es la cotización de recurso r , entonces

$$\sum_{i=1}^n [V]_{ir} = \sum_{i=1}^n \mu_i(r) = \sum_{i \in N} \mu_i(r) = C_r.$$

Tanto el presupuesto de cada agente como la cotización de cada recurso se puede hallar usando la norma 1 de \mathbb{R}^k , $|\cdot|_1$; esto es; si $\vec{x} = (x_1, \dots, x_k) \in \mathbb{R}^k$, entonces

$$|\vec{x}|_1 = \sum_{i=1}^k |x_i|.$$

Ejemplo 10. En el Ejemplo 9, todos los agentes tienen igual presupuesto $K=100$, puesm

$$|[V]_{1*}|_1 = \sum_{r=1}^4 [V]_{1r} = 30 + 40 + 20 + 10 = 100;$$

$$|[V]_{2*}|_1 = \sum_{r=1}^4 [V]_{2r} = 30 + 10 + 50 + 10 = 100;$$

$$|[V]_{3*}|_1 = \sum_{r=1}^4 [V]_{3r} = 30 + 24 + 16 + 30 = 100.$$

Mientras que las cotizaciones de cada recurso están dadas por

$$|[V_{*1}]|_1 = \sum_{i=1}^3 [V]_{i1} = 30 + 30 + 30 = 90, |[V_{*2}]|_1 = \sum_{i=1}^3 [V]_{i2} = 40 + 10 + 24 = 74;$$

$$|[V_{*3}]|_1 = \sum_{i=1}^3 [V]_{i3} = 50 + 20 + 16 = 86, |[V_{*4}]|_1 = \sum_{i=1}^3 [V]_{i4} = 10 + 10 + 30 = 50;$$

luego, el recurso "a" es el más cotizado y "d" es el menos cotizado.

Matriz de utilidad

Cada asignación tiene asociada una matriz de utilidad y a partir de esta buscaremos propiedades sobre la asignación. Veamos la definición formal:

Definición 12. Dada la matriz V una matriz de valuación and F una matriz de asignación que representa la asignación A . La matriz de utilidad de F se define como:

$$U_F = V \cdot F^T \quad (6)$$

donde F^T es la matriz transpuesta de F .

La matriz de utilidades U_F es una matriz cuadrada en $M_{n \times m}(\mathbb{R}^*)$ donde la posición (i, j) esta dada por

$$[U_F]_{ij} = \sum_{k=1}^m [V]_{ik} \cdot [F^T]_{kj}$$

por lo tanto, $[U_F]_{ii}$ es la valuación que da el agente i a lo asignado a través de F ; mientras que $[U_F]_{ij}$ es la valuación de da el agente i a lo asignado, a través de F , el agente j . En otras palabras, si F representa la asignación $A=(A_1, \dots, A_n)$ entonces $\forall i, j \in N$
 $[U_F]_{ij} = \mu_i(A_j)$
 donde μ_i es la función de utilidad del agente i .

Ejemplo 11. Considérese $M=\{a,b,c,d\}$ y $N=[1,2,3]$ y la matriz de valuación dada por

$$V = \begin{pmatrix} 30 & 40 & 20 & 10 \\ 30 & 10 & 50 & 10 \\ 30 & 24 & 16 & 30 \end{pmatrix}$$

Sea F la matriz de asignación que representa a la asignación $A=(\{b\}, \{c\}, \{a,d\})$; es decir

$$F = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

La matriz de utilidad de F, U_F , esta dada por

$$U_F = V \cdot F^T = \begin{pmatrix} 30 & 40 & 20 & 10 \\ 30 & 10 & 50 & 10 \\ 30 & 24 & 16 & 30 \end{pmatrix} \cdot \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 40 & 20 & 40 \\ 10 & 50 & 40 \\ 24 & 16 & 60 \end{pmatrix}$$

Ahora bien, $[U_F]_{11}=40$ es la utilidad que le da el agente 1 a lo asignado por medio de F ; $[U_F]_{12}=20$ es la utilidad que da el agente 1 a lo asignado, a través de F , al agente 2; mientras que, $[U_F]_{33}=60$ es la utilidad que da el agente 3 a lo asignado a él.

2.2.5. Matriz de utilidad y justicia

En esta subsección se estudia estas propiedades usando la matriz de utilidad.

Lema 5. Sea $A \in N^M$ y F la matriz de asignación que la representa. Entonces, A es libre de envidia si, y solo si,

$$\forall i, j \in N, [U_F]_{ii} \geq [U_F]_{ij} \quad (7)$$

donde U_F es la correspondiente matriz de utilidad de F .

Demostración. La demostración se obtiene directamente de las definiciones 5 y 12. Sean $i, j \in N$. Supongamos que A es libre de envidia, por la definición 5, $\mu_i(A_i) \geq \mu_i(A_j)$. Pero, F representa a A , entonces por la definición 12, $[U_F]_{ii} \geq [U_F]_{ij}$. Así, se cumple la ecuación (7), como F representa a A , por la definición 12, $\mu_i(A_i) \geq \mu_i(A_j)$. Luego, A es libre de envidia.

Ejemplo 12. La matriz de valuación es V como en el ejemplo 9.

Para la matriz de asignación

$$F = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix} \text{ se tiene que } U_F = \begin{pmatrix} 40 & 20 & 40 \\ 10 & 50 & 40 \\ 24 & 16 & 60 \end{pmatrix}.$$

Obsérvese que en cada fila de la matriz U_F , los elementos de la diagonal son mayores o iguales que las otras entradas. Así, se cumple la ecuación (7) y, como F representa a la asignación $A = (\{b\}, \{c\}, \{a, d\})$, $A \in EF$.

Lema 6. Sean $A \in N^M$ y F la matriz de asignación que la representa. Entonces, A es libre de envidia de hasta un recurso si, y solo si,

$$\forall i, j \in N, \exists r \in M \text{ tal que } [F]_{jr} = 1 \text{ \& } [U_F]_{ii} \geq [U_F]_{ij} - [V]_{ir} \quad (8)$$

donde U_F es la correspondiente matriz de utilidad de F .

Ejemplo 13. Considérese nuevamente V como en el ejemplo 9. Veamos la propiedad $EF1$ en las asignaciones representadas por las siguientes matrices de asignación

$$F = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix} \quad G = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad H = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

las matrices de utilidades correspondientes son:

$$U_F = \begin{pmatrix} 40 & 20 & 40 \\ 10 & 50 & 40 \\ 24 & 16 & 60 \end{pmatrix} \quad U_G = \begin{pmatrix} 30 & 50 & 20 \\ 30 & 20 & 50 \\ 30 & 54 & 16 \end{pmatrix} \quad U_H = \begin{pmatrix} 30 & 60 & 10 \\ 30 & 60 & 10 \\ 30 & 40 & 30 \end{pmatrix}$$

como la asignación representada por F está en EF , por el lema 2, esta asignación es EFO . Las asignaciones representadas por G y H no están en EF . Sin embargo, la asignación H está en EFO , mientras que la de G no está. En efecto, en U_F se puede apreciar que, los agentes 1 y 3 envidian al agente 2; mientras que, el agente 2 no posee envidia.

$$30 = [U_H]_{11} < [U_H]_{12} = 60 \quad \& \quad 30 = [U_H]_{33} < [U_H]_{32} = 40$$

El agente 2 tiene asignados los recursos $\{b,c\}$. Para eliminar la envidia del agente 1, considérese el recurso "b", descrito en la segunda columna de V ; es decir, $[V]_{12} = 40$, así

$$30 = [U_H]_{11} > [U_H]_{12} - [V]_{12} = 60 - 40 = 20$$

para $i = 1$ y $j = 2$ tomando el recurso "b", $r = 2$, se cumple la ecuación (8). De igual manera, para eliminar la envidia del agente $i = 3$ con el agente $j = 3$, se considera nuevamente el recurso "b" y se tiene que

$$30 = [U_H]_{33} < [U_H]_{32} - [V]_{32} = 40 - 24 = 16$$

así, se cumple la ecuación (8). Luego, H está en EFO .

Para la matriz de asignación G , existe envidia entre todos los agentes; esto es, en cada fila, existe una entrada que es mayor que el elemento de la diagonal. En particular, el agente 3 envidia a los agentes 1 y 2. El agente 2 recibió $\{b,c\}$, representados en las columnas 2 y 3 de V respectivamente. Como

$$30 = \min \{ [U_G]_{32} - [V]_{32}, [U_G]_{33} - [V]_{33} \} > [U_G]_{33} = 16$$

Luego, para $i = 3$ y $j = 2$, la ecuación (8) no se cumple. Así, la asignación representada por G no está en EFO .

A partir de la diagonal de la matriz de utilidad asociada a una asignación, se puede determinar si esta es *EF*. En caso de no serlo, se encuentran los agentes (filas) que generan envidia y sobre estos agentes se verifica que se cumpla la ecuación (8). En la próxima parte, se estudia la versión matricial de las propiedades de Pareto dominante y eficiente.

Matriz de utilidad y eficiencia

A continuación, se estudia la versión matricial de las propiedades de Pareto dominante y óptima de Pareto. En el siguiente lema se observa como usando las matrices de utilidades, es posible determinar si existe la propiedad de Pareto dominancia entre ellas; esto es:

Lema 7. Sean $A, B \in N^M$ y F, G dos matrices de asignación que representan a A y B , respectivamente. Entonces A es Pareto dominada por B si, y solo si, para todo i , $[U_F]_{ii} \leq [U_G]_{ii}$ y existe al menor un agente j donde $[U_F]_{jj} < [U_G]_{jj}$.

Ejemplo 14. Considérese 3 recurso y 3 agentes, $M=\{a,b,c\}$ y $N=\{1,2,3\}$. Supongamos que la matriz de valuación está dada por

$$V = \begin{pmatrix} 20 & 30 & 50 \\ 15 & 50 & 35 \\ 80 & 20 & 50 \end{pmatrix}$$

Considérese las matrices de asignación

$$F = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{ y } G = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

con matrices de utilidades

$$U_F = \begin{pmatrix} 50 & 0 & 50 \\ 65 & 0 & 35 \\ 100 & 0 & 50 \end{pmatrix} \text{ y } U_G = \begin{pmatrix} 50 & 30 & 20 \\ 35 & 50 & 15 \\ 50 & 20 & 80 \end{pmatrix}$$

como

$[U_G]_{11}=50=[U_F]_{11}; [U_G]_{22}=50>0=[U_F]_{22}; [U_G]_{33}=80>0=[U_F]_{33}$
entonces G domina a F .

A continuación, se enuncia el criterio de óptima de Pareto desde un punto de vista matricial.

La demostración se obtiene de las definiciones 4 y 12.

Lema 8. Si F es la matriz de asignación que representa a A , entonces A es óptima de Pareto si y solo si

$\forall F' \in M_{n \times m}(B) [\exists i \in N, [U_{F'}]_{ii} > [U_F]_{ii}] \implies [\exists j \in N, [U_F]_{jj} < [U_{F'}]_{jj}]$
Determinar si una asignación es PO usando el lema 8 se necesita compararla con todas las posibles asignaciones matriciales; dependiendo del problema pudiera ser muy complicado. Sin embargo, como se observó en el teorema 1, toda asignación que maximiza el bienestar social utilitario es PO . En la próxima sección se estudia cómo encontrar el bienestar social utilitario de una asignación usando las matrices de utilidad correspondiente.

Matrices de utilidad y bienestar social utilitario

32

En esta subsección se establece la relación que existe entre la matriz de utilidad y bienestar social utilitario. Además, se demuestra que toda asignación que maximiza SW_U es óptima de Pareto.

Lema 9. Dada $A \in N^M$. Si F es una matriz de asignación que representa la asignación A , entonces el bienestar social utilitario de A es dado por

$$SW_U(A) = \text{traz}(U_F) \quad (9)$$

donde U_F es la matriz de utilidad de F .

Demostración. Sea F la matriz de asignación que representa a $A=(A_1, \dots, A_n)$. Por las definiciones 8 y 10, se tiene que

$$SW_U(A) = \sum_{i \in N} \mu(A_i) = \sum_{i \in N} [U_F]_{ii} = \text{traz}(U_F)$$

Ejemplo 15. Considérese 3 recursos y 3 agentes, $M=\{a,b,c\}$ y $N=\{1,2,3\}$, la matriz de utilidad está dada como en el ejemplo 14, para las matrices de asignación

$$F = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{ y } G = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

Las matrices de utilidades son

$$U_F = \begin{pmatrix} 50 & 0 & 50 \\ 65 & 0 & 35 \\ 100 & 0 & 50 \end{pmatrix} \text{ y } U_G = \begin{pmatrix} 50 & 30 & 20 \\ 35 & 50 & 15 \\ 50 & 20 & 80 \end{pmatrix}$$

Si A y B están representadas por F y G , respectivamente, entonces

$$SW_U(A) = \text{traz}(U_F) = 50 + 0 + 50 = 100 \text{ y } SW_U(B) = \text{traz}(U_G) = 50 + 50 + 80 = 180$$

Luego $B \succ_U A$.

Teorema 3. Sea A una asignación. Si $A \in MSW_U$, entonces $A \in PO$.

Demostración. Dadas $A \in MSW_U$ y F la matriz de asignación que representa a A . En búsqueda de una contradicción, supóngase que $A \notin PO$, así existe $B \in M^N$ tal que A es Pareto dominada por B . Entonces, para todo $i \in N$, $[U_F]_{ii} > [U_G]_{ii}$ y existe $j \in N$ tal que $[U_F]_{jj} > [U_G]_{jj}$, donde G representa a B . Dado que la utilidad de los agentes es aditiva y no negativa

$$SW_U(B) = \text{traz}(G) = \sum_{k=1}^n [U_G]_{kk} > \sum_{k=1}^n [U_F]_{kk} = \text{traz}(F) = SW_U(A)$$

lo cual es una contradicción pues $A \in MSW_U$. Así, $A \in PO$.

Matrices de utilidad y bienestar social de Nash

La función de bienestar social Nash también puede hallarse desde la diagonal de la matriz de utilidad; esto es:

Lema 10. Si F es una matriz de asignación que representa la asignación A , entonces el bienestar social de Nash de A está dado por

$$SW_{Nash}(A) = \text{prod}(U_F) = \prod_{i \in N} [U_F]_{ii} \quad (10)$$

donde U_F es la matriz de utilidad de F .

Demostración. Dada F la matriz de asignación que representa a $A=(A_1, \dots, A_m)$. Por las definiciones 8 y 10, tenemos

$$SW_{Nash}(A) = \prod_{i \in N} \mu(A_i) = \prod_{i \in N} [U_F]_{ii} = \text{traz}(U_F).$$

Una vez obtenida la matriz de utilidades se puede calcular directamente el SW_{Nash} . Para esto se deben multiplicar los elementos de la diagonal principal de la matriz de utilidades. En el ejemplo siguiente se explica el cálculo de esta función desde un punto de vista matricial.

34

Ejemplo 16. Considérese $V = \begin{pmatrix} 20 & 30 & 50 \\ 15 & 50 & 35 \\ 80 & 20 & 50 \end{pmatrix}$, las matrices F y G representan a las asignaciones $A=(a,b,\emptyset,\{c\})$ y $B=(c,b,\{a\})$; respectivamente, tienen matrices de utilidades

$$U_F = \begin{pmatrix} 50 & 0 & 50 \\ 65 & 0 & 35 \\ 100 & 0 & 50 \end{pmatrix} \text{ y } U_G = \begin{pmatrix} 50 & 30 & 20 \\ 35 & 50 & 15 \\ 50 & 20 & 80 \end{pmatrix}$$

entonces

$$SW_{Nash}(A) = \text{prod}(U_F) = 50 \cdot 0 \cdot 50 = 0$$

y

$$SW_{Nash}(B) = \text{prod}(U_G) = 50 \cdot 50 \cdot 80 = 200000$$

Matrices de transición y asignaciones transitorias

Se empieza definiendo la matriz de transición asociada a la matriz de valuación y, a partir de esta, se define las asignaciones transitorias.

Definición 13. Dada la matriz $T \in M_{n \times m}(B)$. T es una matriz de transición si $[T]_{ir}=1$ si $[V]_{ir} \in \max \{[V]_{kr}; 1 \leq k \leq n\}$ y $[T]_{ir}=0$ caso contrario. Una asignación es transitoria si cada recurso es asignado al agente que maximiza la utilidad.

La matriz de transición T es una matriz binaria. Esta matriz se construye en base a la matriz de valuación V ; esto es, $[T]_{ir}$ es 1, si $[V]_{ir}$ es un valor máximo de los valores de la columna r y $[T]_{ir}$ es 0 en cualquier otra posición de la columna. La información que brinda la matriz de transición se interpreta como: qué agente ofrece más valuación por qué recurso. Veamos el siguiente ejemplo:

Ejemplo 17. Considérese V la matriz de valuación como en el ejemplo 9 es la matriz de valuación de un problema de asignación de 4 recursos con 3 agentes. La matriz de transición T se construye a partir de la matriz de valuación V . Esto es, para el recurso representado en la columna 1, todos los agentes lo maximizan, para todo $i=1,2,3$, $[T]_{i1}=1$; para el recurso en la columna 2 lo maximiza los agentes 2 y 3, $[T]_{21}=1=[T]_{22}$; mientras que, la columna 3, solo el segundo agente lo maximiza, $[T]_{23}=1$; para la columna, solo el tercer agente lo maximiza, $[T]_{34}=1$. Así,

$$T = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

Usando la matriz de transición se definen asignaciones matriciales particulares; esto es, cualquier matriz de asignación F que sea construida a partir de una matriz transitoria T , se le llama asignación transitoria.

Definición 14. Una matriz de asignación F es transitoria si,
 $\forall r \in M, \exists i \in N$, tal que $([F]_{ir}=1 \Rightarrow [T]_{ir}=1)$ (11)
 donde T es la matriz de transición.

Ejemplo 18. Dada la matriz de valuación V como en el ejemplo 9, la matriz de transición está dada por

$$T = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

luego las matrices

$$F = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad G = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad H = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

son matrices de asignaciones transitorias.

Teorema 4. (Camacho, Zhunio, Fonseca, & Castillo, 2020) Dada una asignación A y F una matriz de asignación que representa a A . Entonces, F es transitoria si, y solo si, $A \in MSW_U$.

Corolario 1. Sea $A \in N^M$ y T la matriz de asignación que representa a A . Si T es transitoria, entonces A es óptima de Pareto.

Una herramienta computacional

Esta sección presenta un nuevo lenguaje de dominio específico (DSL, por sus siglas en inglés) para resolver problemas de asignación de bienes indivisibles, con base en el enfoque matricial propuesto en la sección anterior. En la literatura actual, no existe un DSL enfocado a resolver este tipo de problemas.

Una herramienta de este tipo es importante porque acelerará el desarrollo y análisis de ejemplos además de motivar la investigación en problemas de asignación de recursos. Esta sección comienza con una revisión de los conceptos básicos

relacionados con los lenguajes de programación y luego explica el proceso general seguido por el DSL propuesto, incluyendo ejemplos de uso.

Lenguajes de programación

Un lenguaje de programación permite a los humanos escribir instrucciones que pueden ser ejecutadas por una computadora para resolver problemas específicos. Este conjunto de instrucciones se conoce como código fuente o simplemente código. Con la evolución de la tecnología, varios lenguajes de programación se han desarrollado; los más populares hoy en día son C, Java y Python. Estos lenguajes tienen un diseño de propósito general; esto significa que se utilizan para resolver cualquier problema que pueda ser computado. Sin embargo, el tiempo necesario para aprender y utilizar correctamente estos lenguajes es inmenso y lidiar con nuevos dominios requiere desarrollar muchas rutinas.

Un lenguaje de dominio específico (DSL) es un lenguaje de programación especializado cuyos comandos han sido diseñados para respaldar la resolución de un problema o conjunto de necesidades dentro de un cierto dominio. Un DSL debe mantener un compromiso entre usabilidad y eficiencia; debe ser fácilmente entendido por los humanos y debe ser ejecutable por una computadora. En este sentido, la eficacia de un DSL depende de la claridad con la que defina y aborde problemas del dominio de destino. Este enfoque específico hace que las DSL sean fáciles de aprender, usar y mantener (Fowler & Parsons, 2010).

El lenguaje de programación interactúa con el sistema operativo (SO); existen dos tipos de interacciones: directa e indirecta. La interacción directa requiere un proceso de compilación. El compilador transforma el código fuente en un conjunto de instrucciones, para que el sistema operativo las ejecute directamente. Por otro lado, un lenguaje de programación

interpretado utiliza un programa llamado intérprete que se encarga de traducir cada instrucción en el momento que se ejecuta.

Programas compilados

El ciclo de vida de un programa compilado comienza desde el archivo de código fuente. El compilador crea un programa objetivo. Luego, el usuario ejecuta el programa objetivo directamente sobre el sistema operativo. La Figura 4 muestra este ciclo de vida.

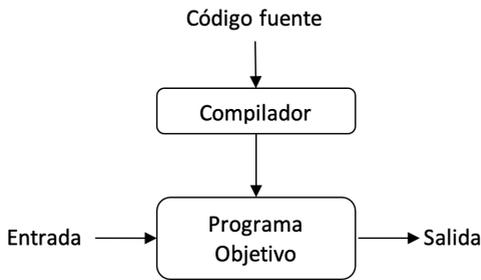


Figura 4. Ciclo de vida de un programa compilado, imagen basada en (Fowler & Parsons, 2010).

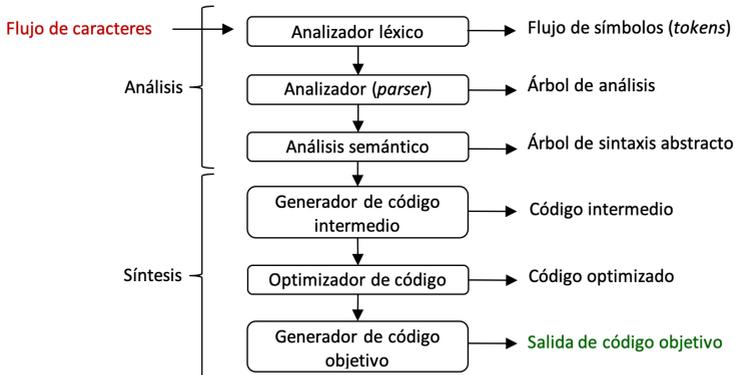


Figura 5. Etapas de compilación, imagen basada en (Fowler & Parsons, 2010).

Las etapas de síntesis comienzan creando el código intermedio, que es un diagrama de flujo con pseudo instrucciones. Los últimos pasos de compilación corresponden a mejoras de código, tanto dependientes como independientes de la arquitectura del ordenador utilizado. Finalmente, el compilador devuelve un archivo ejecutable y desaparece sin participar en la ejecución del programa en ningún momento.

Programas interpretados

Como se muestra en la Figura 6, un lenguaje de programación interpretado utiliza un programa llamado intérprete; que analiza y ejecuta el código fuente sin generar un archivo ejecutable. Por lo tanto, el intérprete permanece activo durante toda la ejecución del programa.

Los ejemplos más famosos de lenguajes interpretados son Python y Ruby.

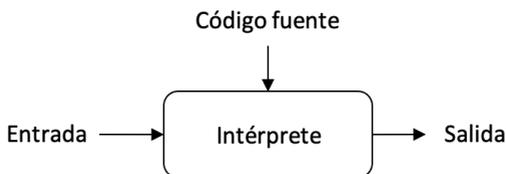


Figura 6. Ciclo de vida de un programa interpretado, imagen basada en (Fowler & Parsons, 2010).

El proceso de interpretación comparte las mismas etapas de análisis que el proceso de compilación, ver Figura 6 Si el árbol de sintaxis abstracta se devuelve sin errores, se llama al intérprete para ejecutar las instrucciones y mostrar los resultados del programa. Luego, a diferencia del compilador, el intérprete visita cada nodo del AST, realiza las operaciones descritas en ese nodo y devuelve los resultados. En otras palabras, el intérprete está presente durante todo el proceso de ejecución del programa.

En términos de ventajas y desventajas, un programa compilado tendrá un mejor tiempo de ejecución que uno interpretado, porque el intérprete realiza las etapas de análisis cada vez que se ejecuta el programa. El archivo ejecutable devuelto por el proceso de compilación depende del hardware y el sistema operativo. Para ejecutar el mismo archivo ejecutable en otro sistema operativo, el archivo de código fuente debe volver a compilarse en el mejor de los casos; de lo contrario, es necesario utilizar bibliotecas específicas del sistema operativo de destino.

Un programa interpretado ejecuta directamente el código fuente, entonces el único requisito es tener instalado el mismo intérprete. Adicionalmente el programador puede ejecutar línea a línea el código fuente, y también puede cambiar los valores de las variables en tiempo de ejecución gracias al intérprete. Esta es una propiedad útil en el desarrollo de nuevos ejemplos de un problema objetivo y también para buscar errores lógicos en el programa.

Para facilitar el desarrollo de ejemplos de asignación de recursos, un DSL interpretado será una valiosa herramienta computacional en esta área de investigación.

DSL propuesto

40

Esta sección propone un nuevo DSL, llamado “lenguaje de programación de asignación de recursos”, RAPL por su nombre en inglés, para representar y analizar problemas de asignación de recursos indivisibles, siguiendo el enfoque matricial desarrollado en la sección anterior. Este incluye la funcionalidad mínima de un DSL completo.

Los problemas de asignación de recursos pueden representarse apropiadamente utilizando matrices. El enfoque matricial para este tipo de problemas es flexible y tiene propiedades matemáticas interesantes, que pueden explotarse para encontrar

soluciones óptimas siguiendo criterios de eficiencia y justicia. Sin embargo, la programación de tales problemas utilizando un lenguaje de programación de propósito general puede incluir un nivel innecesario de complejidad.

Este nuevo DSL permite al usuario declarar las preferencias de los agentes sobre los recursos utilizando matrices. El lenguaje también puede manipular esos elementos con las operaciones adecuadas al problema de asignación de recursos. El DSL propuesto puede medir criterios de eficiencia como optimalidad de Pareto, medir criterios de justicia como libre de envidia y representar resultados utilizando matrices. Esperamos que la facilidad de uso de este DSL pueda motivar más investigaciones sobre este tema.

RAPL es un lenguaje de programación interpretado, de alto nivel y de dominio específico desarrollado en Python. Dado que RAPL es un lenguaje interpretado, sus instrucciones se ejecutan directamente sin una compilación previa del programa. La ventaja de esta característica es que los programas escritos en RAPL serán más fáciles de ejecutar. RAPL también se escribe dinámicamente e incluye inferencia de tipo, lo que significa que el programador no tiene que especificar los tipos de objetos dentro del código. El paradigma de programación soportado por RAPL es declarativo. Este paradigma fue elegido en el diseño porque se espera que la mayor parte del código para este DSL sean operaciones lógico-matemáticas. RAPL sigue los pasos de procesamiento estándar de un intérprete, la figura 7 muestra el flujo general del proceso que toma el programa escrito en RAPL, desde el flujo de caracteres hasta su ejecución.

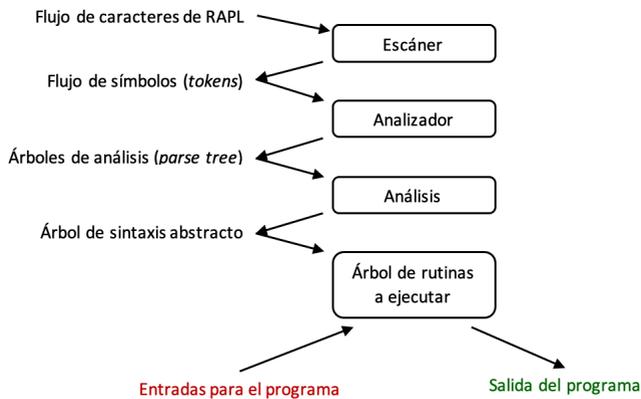


Figura 7. Proceso de Interpretación de RAPL

RAPL escáner

Un programa RAPL debe escribirse en una interfaz de línea de comandos (CLI, por sus siglas en inglés). Esto significa que el código a interpretar se introduce en un shell línea por línea. El escáner toma una línea de código como una secuencia de caracteres y lo convierte en tokens. Un objeto token consta del tipo de token, un valor, una posición inicial y otra final. El escáner ignora espacios vacíos y pestañas. Luego, dependiendo de los caracteres encontrados en la línea de código introducida, se crea un tipo de token. La lista completa de tipos de tokens se puede encontrar en el listado 1. Hay tres grupos de constantes que utiliza el escáner para clasificar los tokens DIGITS, CHARACTERS y CHARACTERS_DIGITS. Si el carácter actual es un dígito, el escáner devuelve un tipo de token numérico. Si el carácter actual es una letra, el escáner comprueba si es una palabra clave o un identificador y crea el tipo de token respectivo. La lista completa de palabras clave admitidas se presentan en el lista 2. Un error de carácter ilegal es devuelto por el intérprete si encuentra un error durante este paso.

Lista 1. Tokens del lenguaje de programación RAPL.

TT INT = 'INT'	TT PO = 'PO'
TT FLOAT = 'FLOAT'	TT LPAREN = 'LPAREN'
TT IDENTIFIER = 'IDENTIFIER'	TT RPAREN = 'RPAREN'
TT KEYWORD = 'KEYWORD'	TT LSQUARE = 'LSQUARE'
TT PLUS = 'PLUS'	TT RSQUARE = 'RSQUARE'
TT MINUS = 'MINUS'	TT LCURLY = 'LCURLY'
TT MUL = 'MUL'	TT RCURLY = 'RCURLY'
TT DIV = 'DIV'	TT EE = 'EE'
TT POW = 'POW'	TT NE = 'NE'
TT EQ = 'EQ'	TT LT = 'LT'
TT AT = 'AT'	TT GT = 'GT'
TT PER = 'PER'	TT LTE = 'LTE'
TT UTIL = 'UTIL'	TT GTE = 'GTE'
TT NASH = 'NASH'	TT COMMA = 'COMMA'
TT EVAL = 'EVAL'	TT ARROW = 'ARROW'
	TT EOF = 'EOF'

Lista 2. Palabras clave del lenguaje de programación RAPL.

1. 'VAR'	'THEN'
'AND'	'ELIF'
'OR'	'ELSE'
'NOT'	'FOR'
'IF'	'TO'
	'STEP'

Análisis sintáctico y semántico del RAPL

Una vez que el escáner devuelve la lista de tokens, el analizador (parser) crea el árbol de sintaxis abstracta. Dependiendo de los tokens encontrados, el analizador crea un tipo de nodo que determina la secuencia correcta de tipos de tokens. Si hay un error, se devuelve un mensaje de error de sintaxis o se devuelve

un error de carácter esperado. Las preferencias de interpretación dentro de los nodos se determinan en función de la gramática utilizada en el diseño del lenguaje.

Gramática de RAPL

La gramática del lenguaje se presenta a continuación y muestra la gramática completa de RAPL utilizando el formulario extendido Backus-Naur (EBNF). RAPL tiene cuatro tipos de objetos hasta ahora: Valor, Número, Lista, Matriz. Un número puede ser entero o flotante. Una lista es una secuencia de números separados por comas ‘,’ y entre corchetes. Una matriz es una secuencia de listas, separadas por comas ‘,’ y entre llaves.

```

<expr> ::= <KEYWORD:VAR> <IDENTIFIER> <EQ> <expr>
| <comp-expr> (((<KEYWORD:AND>)|<KEYWORD:OR>)|
  <comp-expr>)*

<comp-expr> ::= <NOT> <comp-expr>
| <arith-expr> (((<EE>)|<LT>)|<GT>)|<LTE>)|<GTE>)|
  <arith-expr>)*

<arith-expr> ::= <term> (((<PLUS>)|<MINUS>)| <term>)*

<term> ::= <factor> (((<MUL>)|<DIV>)| <factor>)*

<factor> ::= <INT>)|<FLOAT>
| <power>

<power> ::= <atom>)|<POW> <factor>

<atom> ::= (<PLUS>)|<MINUS> <factor>
| <LPAREN> <expr> <RPAREN>
| <matrix-expr>
| <list-expr>
| <if-expr>
| <for-expr>
| <while-expr>

<list-expr> ::= <LSQUARE> ((<expr> <COMMA> <expr>)*)?
  <RSQUARE>

<matrix-expr> ::= <LSQUARE> ((<list-expr>)+ <RSQUARE>

<if-expr> ::= <KEYWORD:IF> <expr> <KEYWORD:THEN>
  <expr> ((<KEYWORD:ELIF> <expr> <KEYWORD:THEN> <expr>)* (<KEYWORD:ELSE>
  <expr>)?

<for-expr> ::= <KEYWORD:FOR> <IDENTIFIER> <EQ> <expr>
  <KEYWORD:TO> <expr> ((<KEYWORD:STEP> <expr>)? <KEYWORD:THEN> <expr>

<while-expr> ::= <KEYWORD:WHILE> <expr> <KEYWORD:THEN>
  <expr>

```

Operadores de RAPL

RAPL admite operaciones binarias matemáticas básicas. Los operadores binarios para dos NÚMEROS son: +, -, /, *, ^. Estos operadores representan suma, resta, división, multiplicación y exponenciación respectivamente. Los paréntesis también son compatibles y pueden ser utilizados para cambiar la prioridad en los cálculos. Los operadores unitarios también son compatibles con RAPL. Gracias a esto, expresiones como -5 tienen sentido. Otros operadores unitarios también están disponibles para operaciones con matrices.

RAPL admite (1) operadores de comparación, como >, <, >=, <= y ==, (2) operadores lógicos, como AND, OR y NOT y (3) declaraciones condicionales y repetitivas, como aquellos declarados con las palabras clave IF, ELIF, ELSE, y FOR. Para el propósito de problemas de asignación de recursos, en RAPL, las operaciones matriciales que el DSL soporta son:

Operaciones matriciales unitarias

- / para el valor del índice de la matriz,
- @ para la matriz transpuesta,
- % para obtener la matriz de transición a partir de la matriz de valuación,
- # para obtener el bienestar social utilitario a partir de la matriz de utilidad,
- ~ para obtener el bienestar social de Nash a partir de la matriz de utilidad.

Operaciones matriciales binarias

- * para la multiplicación de matrices estándar,
- + para la suma de matrices estándar,
- - para la resta de matrices estándar,
- : para obtener la matriz utilitaria a partir de la matriz de valuación y la de asignación.

Obtención del Árbol de Sintaxis Abstracta (AST)

Para construir el AST, el analizador toma la lista de tokens y la lee para identificar si la secuencia corresponde a un nodo específico. El analizador es una clase de Python con diferentes métodos como `matrix_expr`, `list_expr`, `if_expr`, `for_expr`, `while_expr`, `atom`, `power`, `factor`, `term`, `arith_expr`, `comp_expr` y `expr`, que el analizador utiliza para identificar nodos dentro de una secuencia de tokens. Hay diferentes nodos en RAPL: `NumberNode`, `ListNode`, `MatrixNode`, `VarAccessNode`, `VarAssignNode`, `BinOpNode`, `UnaryOpNode`, `IfNode`, `ForNode` y `WhileNode`. Cada nodo es una clase de Python que almacena su posición inicial, posición final y sus métodos correspondientes en el nodo.

Entonces el analizador utiliza sus métodos implementados para validar que el flujo de tokens sigue una secuencia específica comprensible, si es así, el método devuelve un nodo basado en esa secuencia. Si no hay errores durante este proceso, el emparejador devuelve el AST para que sea interpretado. En la Figura 8, la imagen (a) muestra una lista de tokens y la imagen (b) muestra una lista de tokens seguido de un AST, que consiste en un solo `BinOpNode`.

Intérprete de RAPL

46

Si el analizador devuelve el AST sin errores, se llama al intérprete para ejecutar las operaciones y mostrar los resultados. El AST es un mapa de nodos que el intérprete debe visitar en un determinado orden. Luego, el intérprete visita un nodo e intenta devolver el resultado deseado de cada nodo. El intérprete devuelve un error en tiempo de ejecución si hay un error durante esta etapa. Por ejemplo, si el intérprete visita un nodo de operación binaria, se espera que tenga un nodo izquierdo, un nodo operador y un nodo derecho. Una vez que el AST completo ha sido visitado por el intérprete sin errores, el resultado se muestra en la pantalla.

```

RAPL > {[1,2],[3,4]}*{[4,5],[5,6,]}
[LCURLY, LSQUARE, TT_INT:1, COMMA, TT_INT:2, RSQUARE, COMMA, LSQUARE, TT_INT:3, COMMA, TT_INT:4, RSQUARE, R
CURLY, MUL, LCURLY, LSQUARE, TT_INT:4, COMMA, TT_INT:5, RSQUARE, COMMA, LSQUARE, TT_INT:5, COMMA, TT_INT:6,
COMMA, RSQUARE, RCURLY, EOF]
None
Illegal Syntax: Expected 'VAR', int, float, identifier, '+', '-' or '('
File <stdin>, line 1
RAPL >

```

(a) Error del analizador

```

RAPL > _
[]
Illegal Character: '_'File <stdin>, line 1
RAPL >

```

(b) Error de escáner

```

RAPL > 1/0
[TT_INT:1, DIV, TT_INT, EOF]
(TT_INT:1, DIV, TT_INT)
Traceback (most recent call last):
  File <stdin>, line 1, in <program>
Runtime Error: Division by zero
RAPL >

```

(c) Error del interprete

Figura 8. Errores detectados en el proceso de interpretación por RAPL.

Ejemplos usando RAPL

Se utilizará RAPL para analizar algunos de los problemas estudiados en la sección anterior. Se utilizarán algunas líneas de código para obtener las matrices propuestas en el presente trabajo con el fin de realizar un análisis del criterio de justicia y eficiencia de una asignación.

Ejemplo 19. Las figuras 9, 10 y 11, muestran cómo realizar el análisis usando la matriz de valuación:

$$V = \begin{pmatrix} 500 & 200 & 50 & 50 & 200 \\ 500 & 100 & 50 & 100 & 250 \\ 500 & 200 & 25 & 100 & 175 \end{pmatrix}$$

Con la ayuda de RAPL, se analizarán dos asignaciones: la matriz

F que representa una asignación que maximiza el bienestar social utilitario, y la matriz G que representa una asignación que maximiza el bienestar social de Nash.

Primero, se define la matriz de valuación V, que almacena las valuaciones que cada agente da a cada recurso en un problema de asignación de recursos entre 5 recursos y 3 agentes. Luego, con la ayuda del operador %, se obtiene la matriz de transición T a partir de V.

```
RAPL > VAR V = {[500, 200, 50, 50, 200],[500, 100, 50, 100, 250],[500, 200, 25, 100, 175]}
{[500 200 50 50 200]
 [500 100 50 100 250]
 [500 200 25 100 175]
 }
RAPL > %V
{[1 1 1 0 0]
 [1 0 1 1 1]
 [1 1 0 1 0]
 }
```

Figura 9. Declaración de la matriz de valuación y matriz de transición usando el operador %.

A continuación, se definen las matrices transitorias F y G. Las variables F y G se reemplazan luego por la respectiva matriz de asignación transpuesta.

```
RAPL > VAR F = {[1, 1, 1, 0, 0],[0, 0, 0, 1, 1],[0, 0, 0, 0, 0]}
{[1 1 1 0 0]
 [0 0 0 1 1]
 [0 0 0 0 0]
 }
RAPL > VAR G = {[1, 0, 0, 0, 0],[0, 0, 1, 0, 1],[0, 1, 0, 1, 0]}
{[1 0 0 0 0]
 [0 0 1 0 1]
 [0 1 0 1 0]
 }
RAPL > VAR F = @F
{[1 0 0]
 [1 0 0]
 [1 0 0]
 [0 1 0]
 [0 1 0]
 }
RAPL > VAR G = @G
{[1 0 0]
 [0 0 1]
 [0 1 0]
 [0 0 1]
 [0 1 0]
 }
```

Figura 10. Calcular matrices de utilidad multiplicando las correspondientes matrices de valuación y asignación.

Alternativamente, como se muestra en la figura 4.9, las matrices de utilidad U_F y U_G se pueden calcular directamente con el operador ‘:’.

```
RAPL > VAR UF = V:F
{[500 500 500 200 200]
 [500 500 500 100 100]
 [500 500 500 200 200]
 }
RAPL > VAR UG = V:G
{[500 50 200 50 200]
 [500 50 100 50 100]
 [500 25 200 25 200]
 }
RAPL > █
```

Figura 11. Uso del operador ‘:’.

Finalmente, y como se muestra en la figura 12, una vez obtenidas las matrices de utilidad U_F y U_G se procede a calcular los valores del bienestar social utilitario con el operador ‘#’ y los valores de bienestar social de Nash con el operador ‘~’.

```
}
RAPL > # UF
1100
RAPL > ~ UF
0
RAPL > # UG
1100
RAPL > ~ UG
45000000
RAPL > █
```

Figura 12. Calcular SW_U y SW_{Nash} usando los operadores ‘#’ y ‘~’.

Los resultados muestran que RAPL es capaz de describir el problema de asignación de manera muy compacta con una sintaxis clara. Es importante señalar que el cálculo de las asignaciones se realiza con pocas líneas de código. Así se evita el desorden que los lenguajes de programación de propósito general producen cuando intentan representar y resolver este tipo de problemas. Por lo tanto, esta herramienta ayudará en el desarrollo de esta área de investigación.

3. Conclusiones

En este trabajo se presentó un enfoque matricial del problema de asignación de recursos indivisibles. Se mostró el enfoque tradicional que se basa en una perspectiva conjuntista; explicando los aspectos más importantes de este campo y el problema existente de encontrar asignaciones eficientes y justas cuando se consideran recursos indivisibles y utilidades aditivas.

Estos aspectos se presentaron desde enfoque matricial. Esta nueva visión facilita el análisis para identificar las propiedades de eficiencia y justicia en las asignaciones.

A partir de las matrices presentadas se exponen aspectos interesantes. Esto es: de la matriz de asignación se describe la repartición de los recursos de una asignación; a partir de la matriz de valuación: se describe gran parte de los datos iniciales de un problema de asignación, se definen la matriz de transición y, como consecuencia, las asignaciones transitorias; mientras que, de la matriz de utilidad de una asignación: se obtiene el valor del bienestar social utilitario, el valor del bienestar social de Nash y se establece la existencia, o no, de envidia entre un par de agentes.

Por otra parte, el enfoque matricial propuesto motivó al desarrollo de un lenguaje de dominio específico, llamado RAPL. Este se encuentra en sus inicios, pero ya puede realizar las operaciones matriciales básicas necesarias para obtener todas las matrices definidas en este trabajo, con pocas líneas de código. La descripción de RAPL incluyó detalles del intérprete, gramática, operadores y los diferentes tipos de errores. Los ejemplos propuestos mostraron la sintaxis sucinta y limpia de RAPL y sus operadores especializados. Es importante destacar que RAPL permite representar y resolver cualquier problema de asignación de recursos. Esta herramienta computacional facilitará el estudio de nuevos ejemplos y propiedades de interés en la investigación del problema de asignación de recursos. En trabajos futuros, es de interés seguir desarrollando este lenguaje. Para finalizar,

es importante resaltar que resultados presentados en este capítulo ya fueron publicados en (Camacho, Zhunio, Fonseca, & Castillo, 2020) y (Pineda, Zhunio, Camacho, & Fonseca-Delgado, 2020).

Bibliografía

- Aziz, H., Huang, X., Mattei, N., & Segal-Halevi, E. (2021). Computing Welfare-Maximizing Fair Allocations of Indivisible Goods. *Computing Welfare-Maximizing Fair Allocations of Indivisible Goods*.
- Camacho, F., Zhunio, C., Fonseca, R., & Castillo, Z. (2020). Asignación de recursos con eficiencia y justicia débil en el bienestar social utilitario. *Revista Ibérica de Sistemas e Tecnologías de Informação*, 583–595.
- Caragiannis, I., Kurokawa, D., Moulin, H., Procaccia, A. D., Shah, N., & Wang, J. (2019). The unreasonable fairness of maximum Nash welfare. *ACM Transactions on Economics and Computation (TEAC)*, 7, 1–32.
- de Keijzer, B., Bouveret, S., Klos, T., & Zhang, Y. (2009). On the Complexity of Efficiency and Envy-Freeness in Fair Division of Indivisible Goods with Additive Preferences. En F. Rossi, & A. Tsoukias (Ed.), *Algorithmic Decision Theory* (págs. 98–110). Berlin: Springer Berlin Heidelberg.
- Endriss, U. (2010). *Lecture Notes on Fair Division*. CoRR, abs/1806.04234. Obtenido de <http://arxiv.org/abs/1806.04234>
- Endriss, U., Maudet, N., Sadri, F., & Toni, F. (2006). Negotiating Socially Optimal Allocations of Resources. *J. Artif. Intell. Res.*, 25, 315–348. doi:10.1613/jair.1870
- Fowler, M., & Parsons, R. (2010). *Domain Specific Languages*. Addison-Wesley Professional.
- Pineda, I., Zhunio, C., Camacho, F., & Fonseca-Delgado, R. (2020). RAPL: A Domain Specific Language for Resource Allocation of Indivisible Goods. En G. Rodríguez Morales, E. R. Fonseca C., J. P. Salgado, P. Pérez-Gosende, M. Orellana Cordero, & S. Berrezueta (Ed.), *Information and Communication Technologies*. 1307, págs. 479–492. Cham: Springer International Publishing. doi:10.1007/978-3-030-62833-8_35

Ramezani, S., & Endriss, U. (2010). Nash Social Welfare in Multi-agent Resource Allocation. En E. David, E. Gerding, D. Sarne, & O. Shehory (Ed.), *Agent-Mediated Electronic Commerce. Designing Trading Strategies and Mechanisms for Electronic Markets* (págs. 117–131). Berlin: Springer Berlin Heidelberg.

Sistemas de Recomendación: Enfoques y Técnicas

Dulce Milagro Rivero Albarrán

dmrivero@pucesi.edu.ec

Pontificia Universidad Católica del Ecuador - Sede Ibarra

Escuela de Sistemas

Ecuador

1. Introducción

Hoy en día se cuenta con gran cantidad de repositorios de datos de usuarios (patrones de búsqueda, de navegación en la web, gustos de compra) o medidas físicas registradas por algún instrumento de medida (temperatura, humedad, tiempo). Por ello, en la actualidad el problema no es encontrar datos, el problema es qué hacer con esos datos, cómo extraer información relevante de los datos, información que tenga una utilidad para algún usuario. La extracción de información importante es justamente el problema que aborda la minería de datos.

La minería de datos intenta descubrir patrones de comportamiento a partir del procesamiento de grandes volúmenes de datos, integrando métodos de inteligencia artificial, aprendizaje automático, estadística y sistemas de bases de datos.

El desarrollo de aplicaciones que hacen uso de la minería de datos ha permitido el desarrollo de aplicaciones como son los sistemas de reconocimiento de imágenes (Moreano, et al. 2017, Wu, et al. 2015, He, et al 2016), sistemas de detección de fraudes (Douglas, Levesque 2008, Raj, Portia 2011), sistemas de apoyo a los recursos humanos (Jantan, Hamdan, Othman 2010) y sistemas de recomendaciones. Estos últimos proporcionan recomendaciones a los usuarios en función de sus gustos personales. Ejemplos de estas aplicaciones son la recomendación de libros, CD y otros productos en Amazon.com (Linden, Smith 2003), aplicaciones de recomendación de gobierno electrónico, de comercio y compra

electrónica, de enseñanza, por citar algunas (Lu, Wu, Mao, Wang & Zhang 2015).

En este capítulo se describen los conceptos asociados a los sistemas de recomendación; en la primera sección se presentan los enfoques, métricas, técnicas y algoritmos utilizados por los sistemas de recomendación y en la segunda sección se describe un ejemplo de un sistema de recomendación desarrollado para recomendar objetos de aprendizajes a estudiantes.

2. Sistemas de recomendación

Los sistemas de recomendación se encuentran presentes cada vez más en nuestras actividades diarias, son ubicuos. Los encontramos en las aplicaciones de comercio electrónico cuando: le proponen a un usuario artículos que le pueden interesar, revisan noticias, presentan la publicidad en línea que sugieren a los usuarios contenidos que coincidan con sus preferencias. Se puede decir que hoy en día los sistemas de recomendación son inevitables en nuestro quehacer cotidiano. De forma general, un sistema de recomendación es un algoritmo que muestra a los usuarios elementos de interés, ya sean películas para ver, libros para leer, productos para comprar o cualquier otro elemento dependiendo de la empresa u organización. Si se tiene una base de datos que contiene la información de estos elementos, el sistema de recomendación mostraría el número máximo de elementos de esta base de datos que se pueden recomendar al usuario.

El problema de la recomendación se reduce a estimar las calificaciones de los elementos que no han sido vistos por un usuario con el fin de presentarle a éste los elementos que, de acuerdo con su gusto, tiene la mayor calificación. Esta estimación generalmente se basa en las calificaciones otorgadas por el usuario a otros elementos y por las estimaciones otorgadas por otros usuarios a los elementos.

Formalmente, un problema de recomendación se puede formular de la siguiente manera: Sea C el conjunto de todos los usuarios y S el conjunto de todos los elementos posibles que se pueden recomendar, tales como libros, películas o restaurantes. Sea u una función que almacena la utilidad (valoraciones) de los elementos s para el usuario c , es decir,

$$u: C \times S \rightarrow R$$

donde, R es un conjunto totalmente ordenado de valoraciones (enteros no negativos o números reales dentro de un cierto rango), tal como se muestra en la Tabla 1, que es una matriz de valoraciones de películas, los datos fueron extraídos de las bases de datos proporcionada en el portal del grupo Lens. (<http://www.grouplens.org>, 2021).

El sistema de recomendación quiere seleccionar para un usuario $c \in C$ el conjunto de elementos $s \in S$ que maximice la función de utilidad, es decir:

Tabla 1.

Matriz de valoraciones usuario-elementos para un sistema de recomendación de películas

Elemento \ Usuario	Toy Story	Sabrina	Jumanji	Heat	Drácula	Two Much	Fargo	Aristocats	...	Edie
1		3	1		4	3	5	5	...	5
2	4	1	3		5		2		...	2
3	2	1		5			1	2	...	
4	3		2			5	4			5

Nota. (<http://www.grouplens.org>, 2021)

Tanto C como S pueden ser muy grandes, probablemente miles de elementos y millones de usuarios, por lo tanto, el espacio de búsqueda ($C \times S$) es muy grande también. La función de utilidad, por lo general es una valoración en alguna escala, por ejemplo, la escala de Likert (Matas, 2018), que indica el gusto de un usuario por

un elemento particular. La premisa en la que se basa los sistemas de recomendación es que usuarios con gustos similares, en el futuro tendrán preferencias similares. En la Tabla 1 se muestra la matriz usuario-elemento, en ella las valoraciones se encuentran especificadas en una escala de 1 al 5, donde 1 significa que no le gusto y 5 le gustó mucho, los espacios en blanco se refieren a las películas que no han sido valoradas por el usuario.

El objetivo del sistema de recomendación es estimar (predecir) las calificaciones de las combinaciones de elementos/usuarios no calificados y emitir las recomendaciones apropiadas usando estas predicciones. Luego de calcular las valoraciones desconocidas, se seleccionan las N valoraciones más altas.

Las nuevas estimaciones se pueden realizar de diversas formas, utilizando aprendizaje automático, heurísticas o teoría de aproximaciones. Además, las recomendaciones se pueden realizar aplicando alguno de los siguientes métodos:

- Basado en contenido: donde se recomiendan elementos similares a los que el usuario seleccionó en el pasado
- Colaborativo: donde se recomiendan elementos que usuarios con gustos o preferencias similares seleccionaron en el pasado
- Híbrido: se combina los resultados obtenidos en los métodos basados en contenidos y colaborativo

56 2.1. Método basado en contenido

Este método tiene su origen en los sistemas de recuperación de información y en los sistemas de filtrado de información, en la actualidad se utilizan para recomendar elementos que contengan información de tipo texto como: documentos, sitios web, noticias. Este método requiere información del producto, por ejemplo, para la recomendación de películas se necesitan datos como el nombre del director, géneros de la película (drama, comedia, ...), actores, año, por nombrar algunos. Estos datos son llamados etiquetas o *tag* (t).

En el método basado en contenido la función de utilidad $u(c,s)$ del elemento s para el usuario c se calcula a partir de las valoraciones que el usuario c le ha asignado a los s' elementos que son “similares” al elemento s . En el ejemplo de las películas, el método identificaría un grado de similitud entre las etiquetas comunes (actores, directores, géneros, año, entre otros) que el usuario c ha asignado una valoración alta y recomendará las películas que tengan un alto grado de similitud con las preferencias del usuario.

El método define un perfil para cada usuario de forma independiente, con información sobre sus gustos, preferencias y/o necesidades. Esta información puede obtenerse de forma explícita, a través de cuestionarios en la que el usuario le asigna una valoración a la etiqueta o de forma implícita, observando el comportamiento del usuario en el pasado.

Para que este método funcione de forma adecuada es necesario tener un buen número de elementos valorados por el usuario, esto con el fin de poder identificar claramente las preferencias de dicho usuario. Por lo tanto, este método, para nuevos usuarios, o usuarios con muy pocas valoraciones, no producirá buenas recomendaciones.

Formalmente, se define *ElementProfile*(s) como el perfil de cada elemento, éste contiene el conjunto de etiquetas que caracteriza al elemento s y es utilizado para el cálculo de la recomendación. Se define la ponderación w_{ij} como la importancia de la etiqueta k_j en el elemento d_i . Esta ponderación se calcula de diversas maneras, una de las más utilizadas de la propuesta por Salton (1989) la cual se basa en la frecuencia inversa TF-IDF, para el cálculo de esta ponderación se define a N como el total de elementos a recomendar a un usuario y k_j es la etiqueta j -ésima del documento la cual aparece n_i veces los documentos. Además, se asume que $f_{i,j}$ es el número de veces que la etiqueta k_j aparece en el d_i , entonces

la frecuencia TF_{ij} de la etiqueta k_i en el documento d_j es

$$TF_{ij} = \frac{f_{z,j} k_i}{Numd_j} \quad (E.1)$$

Donde el máximo se calcula sobre las frecuencias $f_{z,j}$ de todas las etiquetas k_z que aparecen en el documento d_j . Para el cálculo de las ponderaciones de cada etiqueta (w_{ij}), normalmente se utiliza la frecuencia inversa del término junto a la frecuencia inversa del documento (IDF) que se calcula $IDF_i = \log$ de la cantidad total de documentos d_j dividido por la suma de todas las veces que un k_z se repite, entonces

$$w_{ij} = TF_{ij} * IDF_i \quad (Ec. 2)$$

Una vez se calculen las ponderaciones de cada etiqueta se tiene:

$$ElementProfile(s) = (w_{1,j}, w_{2,j}, w_{3,j}, \dots, w_{k,j})$$

El sistema de recomendación comparará los elementos a recomendar con los elementos que ya han sido valorados por los usuarios y seleccionará aquellos elementos cuyas valoraciones sean similares.

Por otro lado, se define *ContentBasedProfile(c)* como el perfil que describe las preferencias del usuario. Este vector se calcula a partir de los elementos que han sido valorados por el usuario. Se define el peso w_{ci} como la importancia que tiene para el usuario c la etiqueta i . Por tanto, se tiene

$$ContentBasedProfile(c) = (w_{c,1}, w_{c,2}, w_{c,3}, \dots, w_{c,k})$$

$w_{c,k}$ puede ser calculado de maneras diferentes, por ejemplo, el promedio de los valores individuales de cada elemento o utilizando clasificadores Bayesiano. Finalmente, La función de utilidad $u(c,s)$ se define como:

$$u(c,s) = score(ContentBasedProfile(c), ElementProfile(s))$$

Por lo general, esta función de utilidad es calculada usando una heurística de puntuación definida en términos de vectores y como la medida de similitud del coseno, finalmente se tiene

$$u(c,s) = \frac{\sum_{i=1}^k w_{i,c} * w_{i,s}}{\sum_{i=1}^k w_{i,c}^2 * \sum_{i=1}^k w_{i,s}^2} \quad (\text{Ec. 3})$$

Donde K es el total de etiquetas o *tag*.

El cálculo de similitudes del coseno se ha utilizado en sistemas de recuperación de información. Sin embargo, este método puede utilizar los algoritmos de aprendizaje automático, tales como árboles de decisión, redes neuronales o agrupamiento (*clustering*), con el fin de clasificar los elementos relevantes o no relevantes. El modelo que se obtiene, a partir del entrenamiento del algoritmo con los elementos valorados previamente, podrá predecir la utilidad de un elemento. Por ejemplo, si se utiliza el clasificador de Naive Bayes (Mitchell 1997), asumiendo independencia entre sus etiquetas, se puede calcular la probabilidad que un determinado elemento s_j pertenezca a una clase C_i de la siguiente manera:

$$P(C_i/s_j) = P(C_i/ k_{1j} \& k_{2j} \& k_{3j} \& \dots \& k_{mj}) \quad (\text{Ec. 4})$$

Donde las probabilidades $P(C_i)$ y $\prod P(k_{xj} | C_i)$ son calculadas durante el entrenamiento y cada nuevo elemento s' será asignada a la clase donde $P(C_i/s')$ tenga la mayor probabilidad.

Para utilizar un algoritmo de agrupamiento (*k-means* o *k-means spectral*) se debe calcular a qué clase pertenece el elemento a recomendar para luego verificar si esta es la clase a la que pertenece el usuario.

2.2. Método colaborativo

Para generar nuevas recomendaciones en este método se utiliza las interacciones que se tiene registradas de los usuarios y los elementos. Estas interacciones se almacenan en una matriz

usuario-elemento, como la mostrada en la Tabla 1. Este método se basa en el supuesto que los usuarios tienden a seleccionar elementos similares a los que en el pasado usuarios similares a este han seleccionado. Para ello, calcula la utilidad de un elemento s para un usuario particular c , usando las valoraciones de elementos que previamente han realizado otros usuarios.

Formalmente, $u(c,s)$ se calcula a partir de las utilidades $u(c_j,s)$ asignadas a los elementos s por los usuarios c_j pertenecientes a C que son “similares” al usuario c . Por ejemplo, en la Tabla 1, se desea recomendar películas al usuario con $id=4$, para ello, primero será necesario encontrar el usuario más similar a él, y luego se elaborará una lista de recomendaciones con las películas que un usuario similar a él ha dado una valoración alta y que aún no han sido valoradas por el usuario con $id=4$.

Los algoritmos para la recomendación colaborativa pueden ser agrupados en dos grandes clases:

- Algoritmos basados en memoria: llamados así porque requieren mantener disponibles (en memoria) las valoraciones realizadas por todos los usuarios para encontrar una nueva recomendación a un usuario c . Trabajan directamente con la matriz usuario-elementos.
- Algoritmos basados en modelos. Se asume que existe un modelo subyacente que explica las interacciones usuario-elemento. El objetivo es encontrar este modelo para hacer nuevas predicciones.

2.2.1. Algoritmo basado en memoria

Son por lo general heurísticas que hacen predicciones de calificación basadas en la colección completa de elementos previamente calificados por los usuarios. Se buscan similitudes entre usuario-usuario y elemento-elemento utilizando sólo la información de la matriz usuario-elemento.

2.2.1.1. Similitud Usuario-Usuario

Este método se centra en las preferencias del usuario, ya que representa a los usuarios en función de la valoración de los elementos y evalúa las distancias entre los usuarios. Es decir, la valoración desconocida $v_{c,s}$ del usuario c al elemento s generalmente se calcula como un agregado de las calificaciones de otros usuarios c' similares al usuario c que han calificado el mismo elemento s .

Los pasos por seguir en este método son los siguientes:

- i. Seleccionar la fila que representa al usuario c al que se le desea realizar la recomendación
- ii. Calcular, utilizando alguno de los cálculos de semejanza, la similitud entre el usuario c y todos los demás usuarios.
 - i. La similitud permitirá obtener los usuarios con interacciones similares en los mismos elementos.
- iii. Obtener los k vecinos más cercanos a nuestro usuario
- iv. Sugerir los elementos más populares entre ellos, es decir, que tengas mayor valoración.

2.2.1.2. Similitud elemento-elemento

Este método busca el conjunto de elementos que son similares a los que el usuario ya interactuó y les dio una alta calificación. Se considera que dos elementos son similares si la mayoría de los usuarios que han interactuado con ambos lo hicieron de manera similar. Este método está centrado en el elemento, ya que representa elementos basados en las interacciones que los usuarios tuvieron con ellos y evalúa las distancias entre esos elementos.

El método realiza los siguientes pasos:

- i. Seleccionar el elemento s que el usuario c le asignó la mayor valoración $v_{c,s}$.
- ii. Seleccionar la columna k (columna en la matriz de

interacción) que contiene el elemento preferido (s con mayor valoración) por el usuario c .

- iii. Calcular similitudes entre la columna k que contiene el elemento preferido y todos los demás elementos.
- iv. Mantener los k vecinos más cercanos al mejor elemento seleccionando los nuevos elementos que el usuario de interés no ha evaluado y recomendar estos elementos.

2.2.2. Funciones para el cálculo de similitudes

Para la valoración de un elemento por lo general se utiliza una función de agregación simple puede ser el promedio.

$$v_{c,x} = \frac{1}{N} \sum_{c' \in C'} v_{c',s} \quad (\text{Ec. 5})$$

Donde C' es el conjunto de usuarios similares a c . Para el cálculo de los usuarios similares a c se utiliza la lista de los elementos que otros usuarios han valorado $s_y = \{s \in S \mid v_{y,s} \neq 0\}$, y la lista de los elementos que el usuario c ha valorado $s_x = \{s \in S \mid v_{x,s} \neq 0\}$. Se designa a $s_{x,y}$ como el conjunto de elementos evaluados por ambos usuario, es decir $s_{x,y} = \{s \in S \mid s \in s_y \cap s_x\}$.

62

Para encontrar el conjunto de usuarios (c') o elementos (s') similares se usan varios enfoques. El primero es directo, utilizando la distancia euclidiana para encontrar los K vecinos más cercanos (KNN de sus siglas en inglés). Para ello se calcula la distancia de los elementos de $s_{x,y}$, es decir:

$$d(x,y) = \sqrt{\sum_{i=1}^n (v_{x,s} - v_{y,s})^2} \quad (\text{Ec. 6})$$

Para convertir la función de distancia $d(x,y)$ se utiliza la siguiente función de similitud

$$s(x,y) = \frac{1}{1+d(x,y)} \quad (\text{Ec. 7})$$

Para encontrar los K vecinos más cercanos se debe calcular s_{xy} para cada elemento del conjunto de usuarios C y seleccionar las K distancias más pequeñas. Esta función de similitud, aunque es sencilla de calcular cuando el número de elementos comunes es grande, tiende a distorsionar los resultados. Por ejemplo, si la diferencia entre todos los elementos es 1, con dos elementos el valor es $= 1.42$ pero, si son cuatro elementos la distancia aumenta a $= 2$

Otras funciones que se utilizan para encontrar $sim(c,c')$ son el coeficiente de correlación de Pearson y similitud del coseno. La Correlación de Pearson mide la similitud entre dos usuarios de la siguiente manera:

$$sim(x, y) = \frac{\sum_{s \in S_{x,y}} (v_{x,s} - \bar{v}_x)(v_{y,s} - \bar{v}_y)}{\sqrt{\sum_{i=1}^n (v_{x,s} - \bar{v}_x)^2} * \sqrt{\sum_{i=1}^n (v_{y,s} - \bar{v}_y)^2}} \quad (\text{Ec. 8})$$

Donde, si $0 < sim(x,y) \leq 1$ significa que existe una correlación positiva, si $sim(x,y) = 0$ no hay una relación lineal y si $-1 < sim(x,y) < 0$ indica que existe una correlación negativa.

La segunda función, la similitud del coseno es:

$$sim(x, y) = \frac{\sum_{s \in S_{x,y}} (v_{x,s}) * (v_{y,s})}{\sqrt{\sum_{s \in S_{x,y}} v_{x,s}^2} * \sqrt{\sum_{s \in S_{x,y}} v_{y,s}^2}} \quad (\text{Ec. 9})$$

La selección del enfoque para el cálculo puede variar, diferentes sistemas de recomendación pueden utilizar distintos enfoques para el cálculo de semejanzas entre usuarios y las valoraciones de los elementos.

2.2.3. Algoritmo basado en modelo

Los algoritmos basados en modelos utilizan la matriz usuario-elemento para aprender un modelo, el cual posteriormente

se usa para predecir las clasificaciones y poder realizar las recomendaciones. Un modelo es una abstracción que se obtiene a partir de la matriz usuario-elemento, la cual se utiliza para entrenar el algoritmo de aprendizaje. Por tanto, la matriz no se necesita almacenar, bastará almacenar las descripción o parámetros de los grupos obtenidos durante el entrenamiento (el modelo) para realizar las recomendaciones nuevas. Las clases representan los grupos de usuarios-elementos con gustos similares. Para encontrar un modelo que realice predicciones más precisas es necesario que la matriz registre muchas interacciones entre los usuarios y los elementos, si se tiene un conjunto fijo de usuarios y elementos, las nuevas interacciones registradas a lo largo del tiempo aportarán nueva información y harán que el sistema sea cada vez más eficaz.

Diferentes algoritmos de aprendizaje automático han sido utilizados en la construcción de sistemas de recomendación basados en modelos, algunos son: algoritmo de agrupamiento (agrupamiento *K-means*), redes Bayesianas, redes neuronales y modelos probabilísticos complejos como son el uso de cadenas de Markov.

La selección del algoritmo de aprendizaje depende del sistema a desarrollar. Sin embargo, se puede decir que, desde el punto de vista de utilidad, en ciertos entornos restringidos los algoritmos de filtrado colaborativo simples son casi tan efectivos como los mejores algoritmos posibles.

Dentro de los sistemas simples se proponen: los modelos de agrupamiento y las redes Bayesianas. En el primer modelo, los usuarios con ideas afines se agrupan en clases. El número de clases y los parámetros del modelo se aprenden de los datos. En el segundo modelo, cada elemento se representa como un nodo en la red Bayesiana, donde el estado del nodo guarda el conjunto de valoraciones del elemento. La estructura de la red y los valores de las probabilidades condicionadas ($Pr(r_{c,s} = i | r_{c,s'}, s' \in S)$) se calculan

a partir de los datos de entrenamiento, donde la probabilidad condicional se usa para calcular $r_{c,s}$ se utiliza la ecuación:

$$r_{c,s} = E(r_{c,s}) = \sum_{i=1}^n i * Pr \quad (\text{Ec. 10})$$

Una limitación de los métodos colaborativos es que ubica a un usuario en un único grupo (clúster), siendo útil para varios sistemas de recomendación ubicar al usuario en diferentes categorías, por ejemplo, categorías asociadas a su actividad de trabajo y categorías asociadas a sus actividades de recreación.

2.2.4. Problemas de los métodos de recomendación

El método colaborativo es probablemente el más exitoso en los sistemas de recomendación. sin embargo, sufre de tres problemas: los datos esparcidos (pocos elementos evaluados), el ingreso de un nuevo usuario y el ingreso de un nuevo elemento (arranque en frío).

El problema de los datos esparcidos se presenta cuando la mayoría de los usuarios sólo han calificado una pequeña parte de todos los elementos, es decir, la cantidad de valoraciones registradas es muy pequeña en comparación con la cantidad de valoraciones que deben predecirse, por lo que es difícil encontrar usuarios con valoraciones similares. Por ejemplo, en el sistema de recomendación puede haber muchas películas que han sido valoradas por pocas personas, por tanto, éstas se recomendarían muy raramente, aun cuando estos pocos usuarios les hayan otorgado calificaciones altas. Además, para el usuario cuyos gustos son diferentes en comparación con el resto de la población, probablemente será difícil encontrar otros usuarios similares, lo que generará recomendaciones poco precisas.

Para tratar el problema de los datos esparcidos se proponen dos alternativas

- i. Trabajar con perfiles de usuarios para calcular los usuarios similares, por ejemplo, el uso de edad, género, educación, actividad económica para el sistema de recomendación de películas.

Usar la técnica de reducción de dimensionalidad, Descomposición de Valores Singulares (SVD de sus siglas en inglés), para reducir la dimensionalidad de las matrices de calificaciones dispersas. SVD es un método bien conocido para la factorización de matrices, propuesto en los métodos colaborativos basado en modelo. Este método se describe en el siguiente apartado.

Otros problemas que presentan ambos métodos de recomendación son: 1) el problema de nuevos usuarios, éste se debe a que, para poder hacer recomendaciones precisas, se debe conocer al menos algunas valoraciones que el usuario haya asignado a un conjunto de elementos; 2) el problema del nuevo elemento, ya que éste no se puede recomendar antes de que un usuario lo haya calificado. Entonces, hasta que el nuevo ítem no sea valorado por algunos usuarios éste no será recomendado. Los métodos basados en contenidos sufren menos de estos problemas, ya que en ellos la búsqueda de nuevas recomendaciones se realizan a partir de los elementos o usuarios similares. Solo los nuevos usuarios o elementos con características nunca vistas presentarán este problema, pero, cuando el sistema tenga tiempo operando esta situación es muy poco probable que se presente.

Existen varias estrategias para abordar la problemática de los nuevos usuarios y elementos, ellas son:

- i. Estrategia aleatoria: recomendar elementos aleatorios a nuevos usuarios o elementos a usuarios seleccionados de forma aleatoria
- ii. Estrategia de máxima expectativa: recomendar elementos populares a nuevos usuarios o nuevos elementos a los usuarios más activos
- iii. Estrategia exploratoria: recomendar un conjunto de

- varios elementos a nuevos usuarios o un nuevo elemento a un conjunto de varios usuarios
- iv. Utilizar un método no colaborativo para un nuevo usuario o del elemento.

2.2.5. Método para la reducción de dimensionalidad

El método SVD busca extraer la información más importante de la matriz de datos original de tamaño p . Para ello, reduce la dimensionalidad de la matriz esparcida usuario-elemento en la nueva matriz de dimensionalidad q ($q < p$) que debe tener una pérdida mínima de información al proyectar los n individuos sobre el espacio de dimensión q . La matriz reducida representa el grado de afinidad de un usuario con las características y el grado de afinidad de un elemento a estas.

Al considerar sólo las características importantes en lugar de todas las valoraciones el tiempo para los cálculos en el sistema de recomendación se reduce. Además, se reducen los problemas relacionados con la dispersión de la matriz. Por ejemplo, en el sistema de recomendación de películas existen algunas características que describen y diferencian muy bien las películas y estas funciones también se pueden utilizar para describir las preferencias del usuario.

Este método, en lugar de aprender el modelo a partir de los datos de la matriz usuario-elemento, deja al sistema descubrir por él mismo las funciones que son útiles para diferenciar los elementos y que él realice las representaciones tanto de los usuarios como de los elementos. La idea de este método es que tanto los usuarios cercanos en términos de preferencias como los elementos cercanos en términos de características terminan teniendo representaciones cercanas en el espacio latente.

El método SVD es un método de factorización de matrices que descompone la matriz $A =$ usuario-elemento (de tamaño n usuarios \times m elementos) en otras matrices U, E tal que:

$$SVD(A) = U * E^t \quad (\text{Ec. 11})$$

El objetivo es encontrar las matrices U y E tal que el producto punto de $U^t E$ se aproxime a la matriz original usuario-elemento, utilizando el método de mínimos cuadrados, se quiere minimizar el error de reconstrucción de la matriz

$$(X, Y) = \operatorname{argmin} (\sum_{x,y \in E} (X_i Y_j^t - A_{i,j})^2) \quad (\text{Ec. 12})$$

Donde E el conjunto de pares (i, j) tal que A_{ij} está definido. Finalmente, añadiendo regularización y dividiendo por 2 se tiene

$$(X, Y) = \operatorname{argmin} \left(\frac{1}{2} \sum_{x,y \in E} (X_i Y_j^t - A_{i,j})^2 + \frac{\lambda}{2} (\sum_{i,k} X_{i,k}^2 + \sum_{i,k} Y_{i,k}^2) \right) \quad (\text{Ec. 13})$$

Los valores de X, Y se obtienen siguiendo el método de optimización del gradiente descendiente. Una vez que se ha factorizado la matriz, se tiene menos información para hacer una nueva recomendación, simplemente se multiplica un vector de usuario por cualquier vector de ítem para estimar la calificación correspondiente.

2.3. Método basado en conocimiento

Los sistemas de recomendación basados en conocimiento intentan resolver dos de los problemas que presentan los métodos colaborativo y basado en contenido, específicamente el problema del nuevo elemento y el elemento con pocas valoraciones. Se utilizan cuando: el cliente desea establecer alguna restricción o reglas de preferencia; es difícil obtener valoraciones de un producto o cuando las valoraciones de un producto cambian muy rápidamente.

El razonamiento basado en casos es una expresión común de la técnica de recomendación de sistemas basados en conocimiento (KB, de sus siglas en inglés). En estos sistemas los elementos se representan como casos y las recomendaciones se generan a partir de los casos más similares a la consulta o perfil del usuario. Asimismo, las ontologías han sido utilizadas para la representación del conocimiento, representando los conceptos del dominio y las relaciones entre los conceptos y a partir de ésta se calcula la similitud semántica entre los elementos es calculada.

También se encuentran los sistemas basados en reglas. En estos sistemas se definen un conjunto de reglas que son utilizadas para hacer las recomendaciones, las reglas son de la forma $X \Rightarrow Y$. Los algoritmos de recomendación basados en reglas utilizan dos umbrales:

- i. De soporte mínimo (S): establece la probabilidad de que una entrada satisfaga tanto X como Y
- ii. De confianza mínima: se define como la probabilidad de que una entrada satisfaga Y cuando satisface X .

Por tanto, el algoritmo debe encontrar todas las reglas de asociación que satisfagan un mínimo de restricciones de soporte y confianza, con los parámetros especificados por el usuario. Por lo tanto, el usuario debe tener cierta experiencia para encontrar el soporte adecuado y la configuración de confianza para lograr las mejores reglas. Lo que significa que el usuario debe tener cierta experiencia para establecer los adecuados valores de estos umbrales a fin de encontrar las mejores reglas.

2.4. Método híbrido

El método híbrido intenta resolver algunos de los problemas citados en los párrafos anteriores. Para ello, este método propone combinar ambos métodos, basado en contenido y colaborativo. Diferentes maneras de combinar estos métodos se

han propuesto, estos son:

- Combinación en paralelo: Implementar de forma independiente cada método (basado en contenido, colaborativo, basado en conocimiento) y mezclar, utilizando alguna estrategia (mezcla lineal simple, más votados, etc), los resultados obtenidos por cada método.
- Incorporar algunas características del método basado en contenido dentro del colaborativos. Por ejemplo, el uso de los perfiles de usuarios que se mencionó anteriormente.
- Incorporar algunas características del método colaborativo dentro del método basado en contenido, esta forma utiliza alguna técnica de reducción de dimensionalidad en un grupo de perfiles basados en contenido
- Construir un método unificado que integre las características más importantes de cada método. Bajo esta forma se han propuesto: un sistema clasificador basado en reglas, un sistema que utiliza modelo de regresión bayesianos de efectos mixtos que emplean métodos de Monte Carlo en cadena de Markov para la estimación y predicción de parámetros.

Estos sistemas híbridos unificados pueden ser extendidos, por ejemplo, agregando una base de conocimientos. Para ello, utiliza el conocimiento que se tiene del usuario y de los productos de manera de hacer más personales las recomendaciones.

3. Un sistema recomendador de películas

Con el objeto de describir cómo se usan los métodos descritos en la sección anterior, se presenta un ejemplo de cómo calcular los usuarios o películas similares para recomendar nuevas películas a un usuario en particular. El ejemplo utiliza el conjunto de datos disponibles <http://www.grouplens.org/> específicamente *ml-latest-small* y el lenguaje que se usa para implementar los algoritmos para el cálculo de similitudes es Python

Los datos se almacenan en 3 tablas: *tag.csv*, *movies.csv* y *rating.csv*. La tabla *rating* contiene las valoraciones (1-5) que un usuario (identificado por su *id*) asigna a película (identificada por un *id*), tal como se muestra es la Tabla 2.

Tabla 2: Ejemplo del archivo *rating.csv*

userId	movieId	rating	Timestamp
1	1	4.0	964982703
2	89774	5.0	1445715189
3	31	0.5	1306463578

La tabla *movies* relaciona el *id* de la película, con su nombre y géneros, tales como: *action*, *adventure*, *animation*, *comedy*, *crime*, *drama*, etc. (ver Tabla 3).

Tabla 3: Ejemplo del archivo *movies.csv*

movieId	Title	genres
1	Toy Story (1995)	Adventure Animation Children Comedy Fantasy
2	Jumanji (1995)	Adventure Children Fantasy
3	Grumpier Old Men (1995)	Comedy Romance

La tabla *tag* relaciona el *id* del usuario con el *id* de la película y las etiquetas que cada usuario asigna a la película (ver Tabla 4).

Tabla 4: Ejemplo del archivo *tag.csv*

userId	movieId	Tag	timestamp
2	60756	Funny	1445714994
2	106782	Leonardo DiCaprio	1445715051
2	106782	Martin Scorsese	1445715056
7	48516	way too long	1169687325
18	431	Al Pacino	1462138765
18	431	Gangster	1462138749
18	431	Mafia	1462138755

En el ejemplo se muestran esquemas simplificados de las tablas. Así, la Tabla 5 presenta la matriz usuario-película.



Tabla 5: Matriz usuario/película

userid – movieid	47	52	590	1101	2020	2414	3210	4006
17		3.0	1.0		4.0	3.0	5.0	
120	4.0	1.0	3.0		5.0			2.0
232	2.0	1.0		5.0				1.0
329	3.0		2.0			5.0		4.0
461			5.0			3.0		4.0

Los datos de la Tabla 5, para ser manipulados en el programa, se almacenan como datos tipo diccionario de la siguiente forma:

valoraciones = { 17: {52:3, 590:1, 2020:4, 2414:3, 3210:5},
 120: {47:4, 52:1, 590:3, 2020:5, 4006:2},
 232: {47:2, 52:1, 1110:5, 4006:1},
 329: {47:3, 590:2, 2014:5, 4006:4},
 461: {590:5, 2014:3, 4006:4}}

3.1. Recomendación de OA utilizando el método basado en contenido

Para la búsqueda de las películas a recomendar utilizando el método basado en contenido se seleccionaron un conjunto de los tags asignados por para crear la matriz usuarios-value-tags (*us_tag*), donde [*us_tag*] = número de veces que el usuario ha indicado ese *tag* en la evaluación de la película.

Posteriormente se creó la matriz de similitud usuario elementos utilizando las Ec. 1,2,3 y se seleccionaron los K ($K=1$) elementos más similares al usuario c .

A partir de ello, se generó una lista de películas con las N películas que los usuarios c' han valorado y el usuario c no ha valorado.

Además, se utilizó el algoritmo de K -means para encontrar agrupamientos de usuarios con gustos similares en sus etiquetas (*tag*). Encontrado los grupos, se procedió a crear las listas de

películas valoradas por todos los usuarios pertenecientes a cada grupo. Posteriormente, para recomendar una película a un usuario se:

- Identificó a qué grupo pertenece el nuevo usuario
- Seleccionó el número de Películas a recomendar

Para esta aplicación se asume que el usuario nuevo al menos ha valorado algunas películas y ha asignado algunas etiquetas. De lo contrario, se tenía que definir un perfil de usuario que facilitara la búsqueda o utilizar algunas de las estrategias propuestas en la sección 1.2.3 para abordar este tipo de problema.

3.2. Recomendación de OA utilizando el método colaborativo basado en memoria

Para la búsqueda de las películas a recomendar utilizando el método basado en memoria, primero se debe encontrar el o los usuarios similares al usuario *c*. En este ejemplo, dado el tamaño de la Tabla 5 sólo se busca un único usuario similar. Si se desea hacer una recomendación al usuario con *id=329*, se utiliza la función de similitud (Ec. 7) basada en la distancia euclidiana.

Para el cálculo de la similitud se utiliza la Correlación de Pearson, los resultados se muestran en la Tabla 6

Tabla 6: Matriz de similitud usando la Correlación de Pearson

userid	17	120	232	329	461
17	1.0	0.33	0	0	0
120	0.33	1.0	0.95	-0,5	1
232	0	0.95	1.0	-1.0	0
329	0	-0.5	-1.0	1.0	-0.98
461	0	1	0	-0.98	1.0

Se puede observar que el usuario que tiene el mayor valor de similitud al usuario 232 es el usuario con 120 (similitud = 0.95),

por tanto, se le pueden recomendar las películas que el usuario 120 le ha dado una alta valoración (≥ 3) y que el usuario 329 todavía no ha valorado (movieid = 590, 2020).

3.3. Recomendación de OA utilizando el método colaborativo basado en modelo

En este caso se desea agrupar los clientes teniendo en cuenta la similitud de los ítems previamente valorados (vistos) por los usuarios. Se debe encontrar el modelo que reúne a los usuarios similares en K grupos. En este ejemplo, para encontrar el modelo se usa el algoritmo *k-means* (apartado 2.2.3). Seleccionado el número óptimo de grupos se procede a almacenar el modelo aprendido (centroides de cada grupo). Posteriormente, se identificarán los usuarios que pertenecen a cada grupo y se almacenarán en un vector las películas vistas por cada usuario, registrando las frecuencias y el promedio de sus valoraciones (Ec. 1). Finalmente, se ordena el vector de acuerdo con la frecuencia y la valoración para recomendar la N películas más vistas y mejor valoradas por ese grupo.

Para realizar una recomendación a un usuario se debe:

- i. Encontrar el grupo al cual él pertenece
- ii. Seleccionar del vector de películas las N películas que el usuario aún no ha valorado (visto)

Los códigos del ejemplo se encuentran en:

<https://github.com/dulcemilagro/AlgoritmosRecomendadores>

4. Resumen

Los sistemas de recomendación son cada día más ubicuos, ofreciendo información a los usuarios sobre aquellos elementos que en los que éstos han demostrado algún interés. Los Sistemas de recomendación se utilizan para agregar nuevas funcionalidades a los sistemas, las cuales

sugieren a los usuarios que en algún momento han ingresado al sistema productos similares a los que han venido consultando o seleccionado.

Los sistemas de recomendación presentados en este capítulo utilizan algoritmos de agrupamiento. Existen tres métodos de recomendación: 1) método basado en contenidos, que para el agrupamiento define un conjunto de etiquetas que describen los elementos; 2) método basado, estos se basan en las interacciones que se tiene de los usuarios y los elementos; los métodos basados en memoria son simples de implementar, pero requieren de suficiente información. 3) métodos híbridos, que son una mezcla de los métodos anteriores

Bibliografía

- Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, 17(6), (pp. 734-749).
- Barry Smyth. (2007). Case-based recommendation. *The adaptive web*, (pp. 342-376).
- Benítez, R., Escudero, G., Kanaan, S., & Rodó, D. M. (2014). *Inteligencia artificial avanzada*. Editorial UOC.
- Berrío Galindo, Á. (2020). *Sistemas de recomendación*.
- Charu C Aggarwal. (2016). *Recommender Systems*. Springer.
- Douglas, D. J., & Levesque, M. (2008). U.S. Patent No. 7, (pp. 366-339). Washington, DC: U.S. Patent and Trademark Office.
- Galán, S. (2007). *Filtrado colaborativo y sistemas de recomendación*. Madrid: Universidad Carlos III de Madrid.
- García García, P. P. (2013). *Reconocimiento de imágenes utilizando redes neuronales artificiales*.
- González, O. E., & Jacques, S. M. (2017). Estado del arte en los sistemas de recomendación. *Res. Comput. Sci.*, 135, 25-40.
- Linden, G., Smith, B., & York, J. (2003). Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1), (pp. 76-80).

- Lu, J., Wu, D., Mao, M., Wang, W., & Zhang, G. (2015). Recommender system application developments: a survey. *Decision Support Systems*, 74, (pp. 12-32).
- Jantan, H., Hamdan, A. R., & Othman, Z. A. (2010). Intelligent techniques for decision support systems in human resource management. *Decision Support Systems*, (pp. 261-276).
- Jiménez, A. Murillo, A., Piza, E., Villalobos, M. y Trejos, J. (2010). Interviewees, Reducción de la dimensionalidad en análisis de datos. Recuperado el 12 de febrero de 2021 de http://www.stat.rice.edu/~jrojo/PASI/lectures/Costa%20rica/2b_Ejemplos_ACP.pdf
- Harper, F. M., & Konstan, J. A. (2015). The movielens datasets: History and context. *ACM transactions on interactive intelligent systems (TIIS)*, 5(4), (pp. 1-19).
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).
- Matas, Antonio. (2018). Diseño del formato de escalas tipo Likert: un estado de la cuestión. *Revista electrónica de investigación educativa*, 20(1), 38-47. Recuperado en 10 de febrero de 2021, de http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1607-40412018000100038&lng=es&tlng=es
- Markus Zanker, Markus Jessenitschnig, and Wolfgang Schmid (2010). Preference reasoning with soft constraints in constraint-based recommender systems. *Constraints*, 15(4), (pp. 574-595).
- Mitchell, Tom M. (1997). *Machine Learning*. McGraw-Hill Science/Engineering/Math.
- Morales, C. A. R. (2018). Algoritmo SVD aplicado a los sistemas de recomendación en el comercio. *Tecnología Investigación y Academia*, 6(1), (pp. 18-27).
- Moreano, J. A. C., Pulloquina, R. H. M., Lagla, G. A. F., Chisag, J. C. C., & Pico, O. A. G. (2017). Reconocimiento facial con base en imágenes. *Revista Boletín Redipe*, 6(5), (pp. 143-151).
- <http://www.grouplens.org>. (1997). *movieLens*. Minnesota,

- Estados Unidos. Recuperado el 2020, de <https://license.umn.edu/product/movieLens-database>
- Quiroz Martínez, Miguel Ángel, Palacios Bravo, Johnny Giuseppe, Gómez Ríos, Mónica Daniela, & Leyva Vázquez, Maikel Yelandi. (2020). Modelo de recomendación basado en conocimiento para el desarrollo del pensamiento del trabajo con objetos de aprendizaje. *Conrado*, 16(75), (pp.111-116). Epub 02 de agosto de 2020. Recuperado en 19 de febrero de 2021, de http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1990-86442020000400111&lng=es&tlng=es.
- Pérez, A.A., & Close, J.G. (2015). *Sistemas de recomendación. Robots Autónomos: Navegación*.
- Raj, S. B. E., & Portia, A. A. (2011, March). Analysis on credit card fraud detection methods. In 2011 International Conference on Computer, Communication and Electrical Technology (ICCCET) (pp. 152-156). IEEE.
- Ricci, F., Rokach, L., & Shapira, B. (2011). Introduction to recommender systems handbook. In *Recommender systems handbook* (pp. 1-35). Springer, Boston, MA.
- Robin Burke. Hybrid recommender systems: Survey and experiments. (2002). *User modeling and user-adapted interaction*, 12(4), (pp. 331-370).
- Salton G. (1989). *Automatic Text Processing*. Addison-Wesley.
- Torres, N. (2015). *Sistema de recomendación basados en métodos de filtrado colaborativo*. Santiago de Chile: Universidad Técnica Federico Santa María.
- Wu, R., Yan, S., Shan, Y., Dang, Q., & Sun, G. (2015). Deep image: Scaling up image recognition. *arXiv preprint arXiv:1501.02876*, 7(8).



Analítica Descriptiva. Procesamiento y visualización de datos

Autores: Laura Guerra Torrealba, Mairett Rodríguez-Balza
Correos: lrguerra@pucesi.edu.ec mairett.balza@gmail.com
Universidad: Pontificia Universidad Católica del Ecuador sede
Ibarra, Universidad Técnica del Norte
Escuela de Ingeniería, Facultad de Educación, Ciencia y
Tecnología.
Ecuador

Introducción

La transformación digital que está viviendo la sociedad actualmente ha conducido a la cuarta revolución industrial denominada Industria 4.0 (Bär, Herbert-Hansen. & Khalid, 2018), donde la interconexión y la comunicación entre humanos, humanos -máquinas y máquinas – máquinas es digital y ubicua. Esta interrelación entre diversos agentes y mediante diferentes canales, genera datos con características específicas (volumen, variedad, velocidad, veracidad, valor) que deben ser analizados para producir información y conocimiento que puedan apoyar la toma de decisiones inteligentes, conduciendo a una mejora de la productividad, flexibilidad y adaptabilidad de las organizaciones para el bien de la sociedad. Ese conocimiento se obtiene a partir de la analítica de datos que reúne un grupo de técnicas y software especializado en el estudio de los datos, donde la estadística tiene un peso considerable (Eggert y Alberts, 2020; Frazzetto et al. 2019). El objetivo de este trabajo es introducir al lector en la aplicación de la estadística descriptiva para realizar un análisis de datos y sus representaciones gráficas con el fin de conocer la gestión de un proceso. Es importante resaltar que existen varios programas que permiten el análisis estadístico, sin embargo, en este documento se utiliza Excel porque el problema de aplicación a resolver representa una situación sencilla, además que es una herramienta accesible desde los equipos de la universidad y que los estudiantes ya la conocen. Excel ofrece la posibilidad de hacer cálculos estadísticos de diferentes maneras, directamente con las fórmulas, con el complemento de análisis

de datos (para ampliar las funcionalidades de Excel respecto a análisis estadísticos), con tablas dinámicas, entre otras. Para lo que concierne a este estudio, la aplicación de los conceptos de la estadística descriptiva como principio de la analítica de datos se realiza a través del uso de las tablas dinámicas.

Conceptos Teóricos

La estadística descriptiva es la rama de la estadística enfocada al estudio de los datos obtenidos de una población o muestra con el propósito de organizar, resumir, describir y analizar cómo se ha ejecutado un evento determinado. Es importante diferenciar entre datos, variables y sus clasificaciones porque, de acuerdo con ello, se decide el análisis estadístico que se debe realizar.

1. Variable

Una variable es un atributo o característica que puede cambiar con el tiempo, entre las personas o por algún otro efecto sobre la unidad experimental. El término de unidad experimental está referido a la persona, evento u objeto en el que se mide una variable. Las variables se clasifican como variables cualitativas y variables cuantitativas.

1.1. Variables cualitativas

Están relacionadas con cualidades o categorías que se presentan en la unidad experimental y se desglosan en:

- Variables nominales: cuando la cualidad no se puede jerarquizar
- Variables ordinales: cuando el tipo de cualidad se puede organizar de acuerdo con una jerarquía

1.2 Variables cuantitativas

Por su parte, las variables cuantitativas representan una característica numérica de la unidad experimental. Se derivan dos clasificaciones adicionales:

- Variables discretas: toman valores finitos y todos deben ser números enteros
- Variables continuas: puede tomar infinitos valores que no son necesariamente números enteros.

2. Datos

Son los valores que toman las variables en un estudio determinado. Pueden ser números, hechos, imágenes, etc. Se denominan univariados cuando están relacionados con una sola variable o multivariados cuando son datos referidos a múltiples variables. Adicionalmente, los datos también se distinguen según el tipo de variables con la que están asociados: datos cualitativos o categóricos o datos cuantitativos

3. Tablas de frecuencias

Al estudiar un experimento, los datos que se van recolectando están desorganizados y se les llama datos en bruto. Es conveniente realizar ciertos ajustes para limpiarlos (escribirlos igual), organizarlos y resumirlos. De manera de responder dos interrogantes básicas: ¿Cuáles son los valores que ha tomado la variable? ¿Cuántas observaciones se obtienen para cada valor de la variable, es decir con qué frecuencia se observa cada valor de la variable?

Una de las formas de presentación de los datos organizados es mediante una tabla de frecuencias, que consiste en una tabla donde se representan los valores o clases de la variable junto con sus frecuencias. Si se trabajan con datos cuantitativos, frecuentemente es conveniente distribuir los datos en clases o categorías en donde se muestra el número de observaciones obtenidas en cada clase; este número de observaciones por clase se denomina la frecuencia de la clase.

Hay que tomar en cuenta que al construir las clases se deben cumplir dos principios: exhaustividad y exclusión (Mendenhall, Beaver y Beaver,2006). Esto significa que una observación debe

pertenecer a una sola clase y todas las observaciones deben estar repartidas entre las clases construidas. Una tabla de distribución de frecuencias de datos agrupados, es una tabla donde se representan las clases conjuntamente con sus frecuencias. Se puede medir la frecuencia de una variable durante un intervalo dado, en tres formas diferentes:

- frecuencia absoluta o frecuencia (f): número de observaciones para cada valor de la variable
- frecuencia relativa (fr): se utiliza para mostrar la fracción del total de observaciones para cada valor de la variable. Se calcula como el número de observaciones para cada valor de la variable dividido entre el número total de observaciones. Por ejemplo, si con n representamos el número total de observaciones en el conjunto, se puede hallar la frecuencia relativa

$$Frecuencia\ relativa = \frac{Frecuencia}{n}$$

la frecuencia relativa generalmente se expresa en porcentaje $100(fr)$

- Frecuencia acumulada (F): suma de las frecuencias del valor de la variable analizada más las frecuencias de los valores que la preceden en la tabla de frecuencia.

Para los valores de frecuencia calculados se cumple que la suma de las frecuencias es siempre n , la suma de las frecuencias relativas es 1 y la suma de los porcentajes es 100%.

4. Distribución de frecuencias acumuladas

Muestra el número de observaciones por debajo de valores dados (Lind, Marchal, y Wathen, 2012). Es similar a la distribución de frecuencias solo que aquí, para cada clase, se suma la frecuencia de la clase con las frecuencias de todas las clases que la anteceden.

De igual manera, *La frecuencia acumulada relativa o frecuencia acumulada porcentual* es la frecuencia acumulada dividida entre la suma de todas las frecuencias (frecuencia total). Por lo tanto, la última clase debe tener como frecuencia acumulada el valor total de observaciones. La tabla que representa esta información se denomina tabla de distribución de frecuencias acumuladas .

Las clases o categorías se pueden construir a juicio del investigador o siguiendo una estructura más formal. En el último caso, se deben determinar los siguientes conceptos:

- Rango: se calcula como la diferencia entre el valor máximo y el valor mínimo del conjunto de datos

$$\text{Rango} = R = \text{Valor máximo} - \text{Valor mínimo}$$

- Número de clases (k): sea n el número total de datos, se puede determinar de varias maneras, pero siempre se debe aproximar al entero más cercano:

$$a) 2^k > n \quad b) k = 1 + 3,22 \log \log (n)$$

- Intervalos de clase: representan los límites para los valores de la variable comprendidos en la clase. El primer intervalo debe contener el valor más bajo x_1 hasta, *pero no incluyendo* x_2 , para el segundo intervalo desde x_2 hasta, pero no incluyendo x_3 , y así sucesivamente. Usando el método de inclusión izquierda. Los valores de x_1, x_2, \dots, x_2 son llamados fronteras de la clase.
- Tamaño o amplitud de la clase: es la diferencia entre sus fronteras superior e inferior. Algunas veces, estos valores son constantes para todas las clases, pero no es restrictivo. Si se desea establecer las clases con la misma amplitud, se debe asegurar que todas las observaciones pertenezcan a una clase, lo cual se puede trabajar a través de la fórmula de amplitud:

$$\text{amplitud de clase} = \frac{\text{Rango}}{\text{número de clases}}$$

Luego se establecen los intervalos de clase, tal como se muestra en la Tabla 1:

Tabla 1. Frecuencias por clase

	LI_i =Límite inferior del intervalo i	LS_i = Límite superior del intervalo i
1er intervalo	LI_1 =valor mínimo (*)	LI_1 + amplitud
2do intervalo	LS_1	LS_1 + amplitud
3er intervalo	LS_2	LS_2 + amplitud

Si se desea redondear la amplitud de la clase hacia el entero superior con el fin de obtener un tamaño conveniente de clase, se cubre un rango más amplio que el necesario, por lo que es adecuado distribuir el excedente en los extremos del intervalo total o de las dos colas.

- Marca de la clase: es el punto medio del intervalo de cada clase y se calcula con la semisuma de los valores frontera de la clase.

5. Representación gráfica de datos cualitativos

Una variable cualitativa está asociada a datos cualitativos. Para representar este tipo de variable, se utilizan principalmente, los gráficos de barra y los gráficos de pastel o sectores. Dichas representaciones se realizan fácilmente en Excel:

5.1 Gráficos de barras

Este gráfico se genera a partir de la tabla de frecuencias, obteniendo una figura conformada por rectángulos que pueden ser verticales u horizontales. Si se consideran los valores obtenidos de la variable en el eje horizontal y las frecuencias absolutas o relativas en el eje vertical, se obtendrá un gráfico de barras verticales. Se debe tener en cuenta que las barras no se deben presentar adyacentes unas a otras, pues los valores de la variable cualitativa son características cualitativas. Con este tipo de representación, se desea destacar la frecuencia de cada categoría.

5.2 Gráficos de torta, pastel o sectores

Se utilizan con la intención de mostrar la relación que existe entre las frecuencias de cada categoría con el total de observaciones,

es decir que se grafican son los porcentajes de las frecuencias relativas. La idea intuitiva es repartir el pastel en porciones (sectores circulares) de tamaño que representan cada categoría. Como el pastel tiene forma de círculo y un círculo tiene 360°, se calcula el ángulo de cada sector circular de la siguiente manera:

$$\text{ángulo} = \text{frecuencia relativa} * 360^\circ$$

De esta manera, el ángulo del sector circular es proporcional al porcentaje de la frecuencia relativa de la categoría. Sobre este tipo de gráficos, Machado y Rodríguez (2019), opinan: “es recomendable su uso para variables con pocas categorías. En caso contrario, pierde efectividad de representación ya que puede resultar confusa su percepción e interpretación” (p.83).

6. Representación gráfica de datos cuantitativos

Los gráficos de barra y de pastel mencionados en 5.1 y 5.2, también se pueden utilizar con datos cuantitativos, especialmente cuando se relacionan una variable cualitativa con una cuantitativa o cuando se segmentan los datos. Por ejemplo, si se tienen los datos de ingreso mensual de los pobladores de la provincia de Imbabura, se puede realizar un gráfico de barras para representar los ingresos para cada grupo de la población (jóvenes, adultos, adultos mayores) o por sectores (Ibarra, Atuntaqui, Otavalo). El gráfico de pastel mostraría la proporción de los ingresos de cada grupo o sector con respecto al total de ingresos de la población. Existen otros tipos de gráficos para los datos cuantitativos, que son de interés:

6.1 Gráficos de líneas – series de tiempo:

Cuando los datos que se desean graficar tienen que ver con valores de la variable y el tiempo de ocurrencia de dichos valores, es conveniente utilizar un gráfico de línea o series de tiempo. En este tipo de representación se suele colocar la variable tiempo en el eje horizontal para que se facilite la visualización de la

tendencia de los datos en un futuro cercano. La información obtenida de este tipo de gráficos permite realizar inferencias o predicciones que facilitan la toma de decisiones.

6.2 Histogramas de frecuencia

Al trabajar con datos cuantitativos provenientes de variables continuas, el gráfico que se utiliza se denomina histograma de frecuencias o de frecuencias relativas, según sea el caso. Los histogramas son muy similares a los gráficos de barra, con la salvedad, de que en el eje horizontal se colocan los intervalos de clase y la altura de cada rectángulo vendrá dada por la frecuencia de la clase (o frecuencia relativa si es el caso). Otra particularidad de esta representación es que los extremos de las barras o rectángulos deben estar solapados, para indicar la continuidad de la variable.

6.3. Polígonos de frecuencia

Son gráficos de línea, donde en el eje horizontal se colocan las marcas de la clase (punto medio de la clase) y en el eje vertical, las frecuencias de la clase, generando los puntos en el plano (x =marca, y =frecuencia) para cada clase. Luego se unen todos estos puntos con segmentos de rectas y se construye el polígono de frecuencia. Tanto el histograma como el polígono de frecuencias reflejan información sobre el conjunto de datos: valor máximo, valor mínimo, rango de concentración de datos, etc. Sin embargo, con el polígono se pueden comparar, de manera más simple, dos distribuciones de frecuencias correspondientes a datos distintos.

6.4 Gráficas de las distribuciones de frecuencias relativas

Tienen la misma forma que los histogramas y polígonos de frecuencia. Al tener estos últimos, se sustituye en el eje vertical, las frecuencias por las frecuencias relativas (o por el porcentaje de frecuencia relativa), manteniendo en el eje horizontal con las mismas características. y conservando la gráfica exactamente igual. A las gráficas que se obtienen se les llama *histogramas*

de frecuencias relativas (o histogramas porcentuales) y polígonos de frecuencias relativas (o polígonos porcentuales), respectivamente.

6.5 Gráficas de distribuciones de frecuencias acumuladas o acumuladas relativas

Este tipo de gráficas son llamadas ojivas; para construirlas se utilizan los histogramas o polígonos de frecuencia mencionados previamente, con la novedad que en el eje vertical se colocan los valores de frecuencia acumulada para cada clase (frecuencia relativa acumulada). El resultado será la gráfica de una función creciente.

7. Datos bivariados. Regresión lineal

Cuando en la misma unidad experimental se observan los valores de dos variables, los datos obtenidos se llaman datos bivariados. Estas observaciones pueden realizarse en forma independiente o determinando la relación que hay entre las variables durante cada observación. Los gráficos de las dos variables sean discretas o continuas, permiten analizar la relación entre ellas. La selección del tipo de gráfico dependerá de la naturaleza de los datos.

Una de las representaciones más importantes para datos bivariados de variables cuantitativas son los gráficos de dispersión, que consisten en ubicar cada pareja de datos sobre el plano cartesiano, considerando una de las variables como la variable x , y la otra variable como la variable y . Mediante esta gráfica se puede determinar el comportamiento en conjunto de las dos variables, al responder a interrogantes como las siguientes:

- ¿Qué tipo de modelo se muestra? ¿lineal, curvo? ¿Existe alguna tendencia?
- ¿Cuán fuerte es el modelo? ¿Todos los puntos se ajustan al modelo?
- ¿Existen puntos atípicos o que se comportan distintos a los demás?

Una de las relaciones más comunes entre datos cuantitativos bivariados es la relación lineal, que implica la posibilidad de trazar una línea recta con puntos de datos hacia arriba y abajo, a la misma distancia fija de la recta. Este tipo de modelo se llama regresión lineal (Walpole, 2012). El modelo de regresión lineal exhibe una medida numérica que representa su fortaleza y que está relacionada con las desviaciones estándar de cada una de las variables (s_x , s_y) y la covarianza entre ellas (s_{xy}). La mayoría de las calculadoras científicas y muchos programas informáticos permiten su cálculo, sin embargo, se comparte su fórmula:

$$\text{Coeficiente de correlación} \quad r = \frac{s_{xy}}{s_x s_y}$$

El valor de este coeficiente estará siempre entre -1 y 1. Mientras más cercano se encuentre a uno de los valores extremos, más fuerte será la relación entre las variables. Por lo tanto, se tienen los siguientes casos:

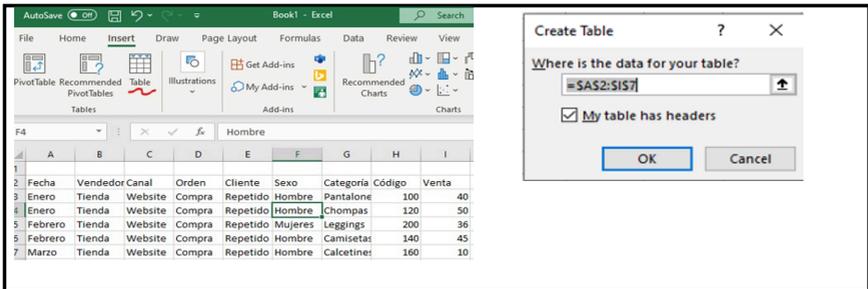
- Sí $0 < r \leq 1$, el modelo es positivo. A medida que aumenta una de las variables, la otra también aumenta. Algunos autores, plantean que la correlación positiva puede dividirse:
positiva perfecta $r = 1$, positiva fuerte, $0,75 \leq r < 1$, positiva débil, $0 < r < 0,75$
- Sí $-1 \leq r < 0$, el modelo es negativo. Al aumentar una de las variables, la otra variable disminuye. De forma análoga, se puede indicar que la correlación negativa será:
negativa perfecta $r = -1$, negativa fuerte, $-1 < r \leq -0,75$, negativa débil, $-0,75 < r < 0$
- $r = 0$, no existe relación entre las variables

8. Herramienta Excel – tablas dinámicas

Cuando los datos obtenidos reúnen especificaciones de diferentes variables, Excel ofrece la alternativa de trabajar con tablas dinámicas para simplificar el análisis, por lo que se presenta un breve resumen sobre los aspectos claves para ejecutar dichas tablas:

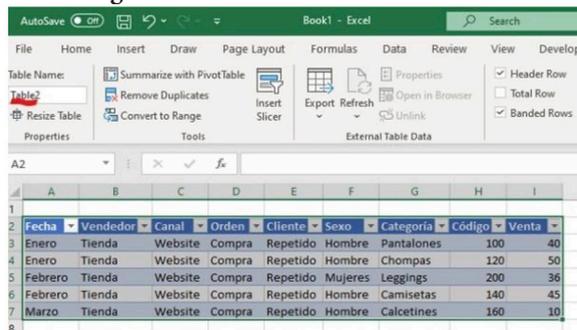
- Lo primero que se debe hacer es llevar los datos a la hoja de Excel.
- Base de datos principal: se asigna a este conjunto de datos como la base de datos con la que se va a trabajar, para lo cual hay que ubicarse en cualquier celda de los datos, seleccionar del menú principal la opción insertar tabla, allí se mostrará el rango de datos y debe asegurarse de indicarle al sistema que la tabla tiene encabezado (Figura 1).

Figura 1. Tabla de datos



El sistema ejecutará la acción, modificará el diseño de la tabla y le asignará un nombre. Puede personalizar la tabla, pero debe estar pendiente que el nombre asignado a esta tabla, se usará para asociar los datos con las posteriores subtablas o tablas dinámicas (Figura 2).

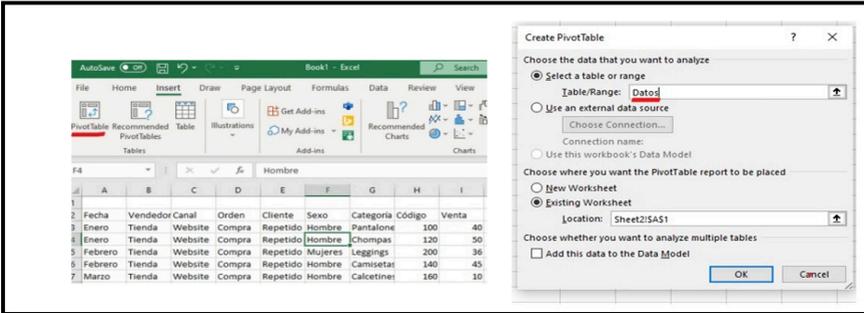
Figura 2. Personalizar tabla de datos



- Tabla dinámica: es una tabla generada de la base de datos. Para construir tablas dinámicas se activa el icono de insertar en el menú principal y luego el de "tabla dinámica o pivot

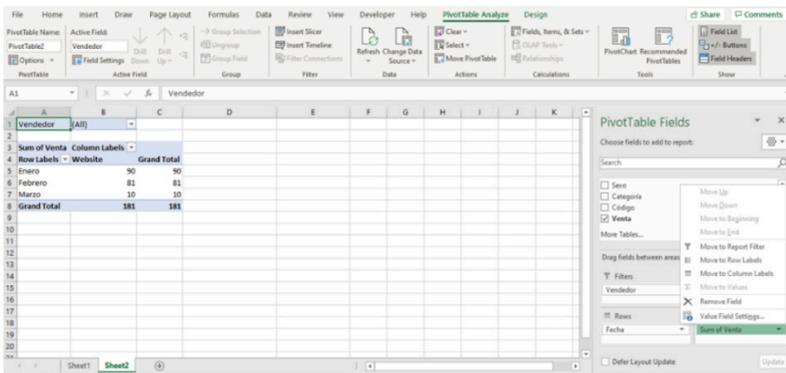
table”. Se debe indicar de donde se tomarán los datos; definir un rango o el nombre de la tabla principal o base de datos y en donde se van a imprimir los resultados (Figura 3).

Figura 3. Tabla dinámica



A la izquierda de la pantalla, aparece el menú de campos de la tabla dinámica. Se seleccionan los campos que se desean ingresar como filas o columnas de la tabla, así como la configuración de cada campo (suma, conteo, promedio, máximo, mínimo, etc.) (Figura 4).

Figura 4. Campos de la tabla dinámica



Se pueden realizar tantas tablas dinámicas como convenga para responder las preguntas de un problema. Para realizar los gráficos estadísticos, se ubica el curso en cualquier celda de la tabla dinámica, se activa el comando insertar del menú principal y se selecciona el gráfico.

Ejercicio de aplicación

La empresa Ecuatextil presenta sus datos de comercialización y ventas del periodo enero -abril 202x , de acuerdo con los atributos mostrados en la Tabla 2. En el anexo A se ubican los datos completos para el desarrollo del ejercicio.

El gerente le pide a Usted, información sobre (1) ¿cuál de los canales ha estado más activo con respecto a las unidades vendidas?, (2) ¿cómo fue la distribución entre los nuevos clientes y los tradicionales y hacia qué tipo de mercado se enfocan (damas o caballeros)? También le interesa saber (3) ¿cuál es la cantidad de unidades devueltas mensualmente por categoría de producto y por vendedor?, (4) ¿qué rango de precios de los blazers, corresponden el 86% de las ventas de los mismos?, (5) ¿cuál es la tendencia de las ventas para el mes de mayo. Finalmente, está convencido en realizar una promoción de calcetines, porque presume que al vender más calcetines también vende más shorts (6) ¿será adecuada esta promoción?

Tabla 2. Formato de Datos de Ventas Ecuatextil

Mes de la venta	Vendedor	Canal de la venta	Tipo de orden	Cliente	Sexo asociado a la mercancía	Categoría	Código	Ventas
Enero	Tienda	Tienda	Compra	Nuevo	Hom-	Pan-	Aso-	En
Febrero	Ana	Website	Devolu-	Repeti-	bre	talones	ciado	uni-
Marzo	María	Insta-	ción	do	Mujer	Chompas	a cada	dades
Abril	José	gram				Leggings	cate-	
	Luis	Face-				Shorts	goría	
	Sandra	book				Calce-		
		Youtube				tines		
						Camise-		
						tas		
						Blazers		

Solución

Los datos proporcionados por la empresa se ubican en una hoja de Excel y se definen como la tabla de base de datos identificándola como Datos. A partir de allí, se generan subtablas,

conformadas como tablas dinámicas para responder cada una de las interrogantes planteadas.

- El canal de comercialización representa una variable cualitativa nominal. Se establece la tabla dinámica colocando los campos de canal (filas) y de Ventas en la casilla de valor. Tome en cuenta, que se especifica que va a contar las unidades vendidas y no el monto total de las ventas. Instagram es el canal más activo para las ventas. (Figura 5).
- El tipo de clientes y el género al cual va dirigido el producto, son variables cualitativas. En la tabla dinámica se coloca el campo sexo en las columnas, el campo cliente en filas y en valor se establece el conteo de las ventas, en unidades (Figura 6). Analizando los datos en porcentaje, el 35% son clientes nuevos y se produjeron 54% pedidos de artículos de damas.

Figura 5. Tabla dinámica, respuesta 1

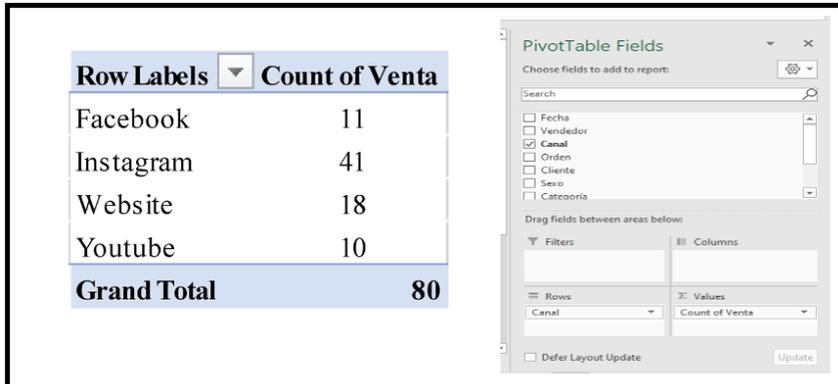
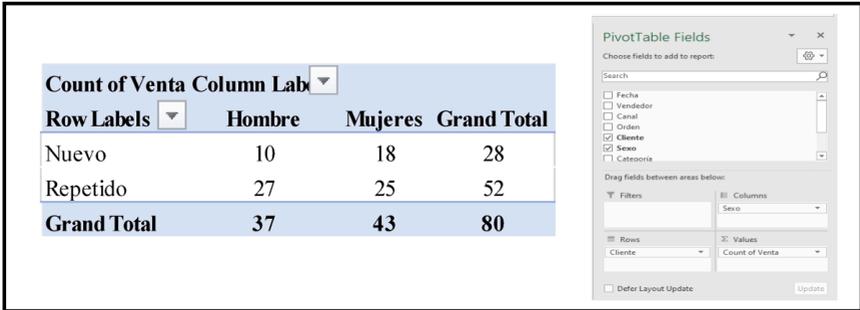
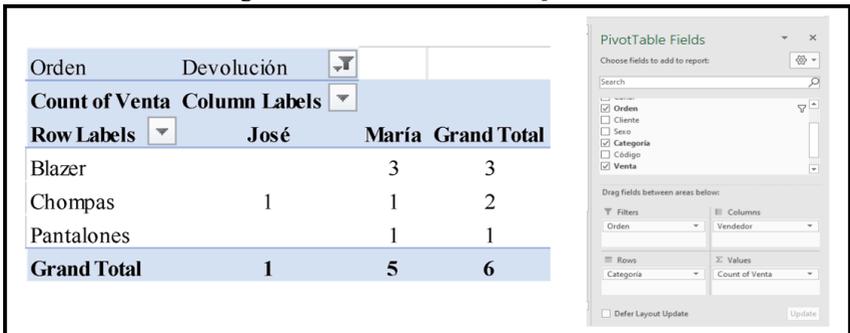


Figura 6. Tabla dinámica, respuesta 2 (unidades vendidas)



- Las devoluciones mensuales (medidas en unidades) es una variable cuantitativa. Para contabilizarlas, se disponen los campos de categoría como filas, de vendedor en columnas y se filtran los datos por tipo de orden igual a devolución (Figura 7). Se observa que el 83% de estas operaciones (5/6*100) corresponden a la vendedora María y se concentran en chompas, blazer y pantalones.

Figura 7. Tabla dinámica, respuesta 3



- Para responder la pregunta 4 se deben calcular las frecuencias acumuladas de las unidades vendidas por rango de precios. Se establece la tabla dinámica con ventas en filas y en la sección de valor (contar), filtrado por categoría (blazer). Se obtienen así, los valores máximos, mínimo de los precios y la cantidad total de unidades vendidas. Se calculan las clases y luego se

posiciona el cursor en la tabla dinámica, se activa el botón izquierdo, luego agrupar, para formar las clases y determinar las frecuencias de cada clase (Figura 8). Se determina que el 86% de las ventas corresponden s a blazer de precios menores a 74\$.

Figura 8. Tabla dinámica, respuesta 4

Categoría	Blazer
34	1
42	2
50	3
55	1
65	5
82	1
100	1
Grand Total	14

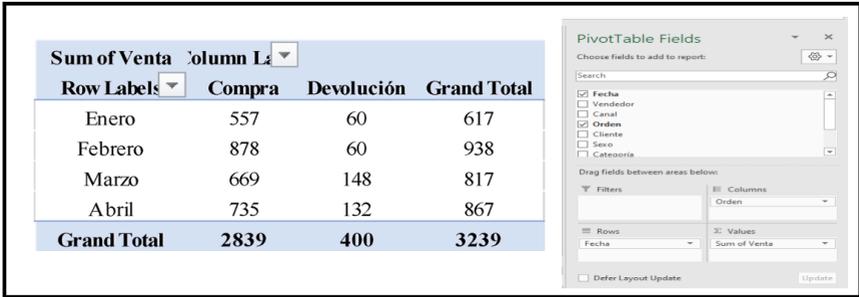
Categoría	Blazer
32-46	3
46-60	4
60-74	5
74-88	1
88-102	1
Grand Total	14

Clase	LI	LS	f	F	%F
1	32	46	3	3	21
2	46	60	4	7	50
3	60	74	5	12	86
4	74	88	1	13	93
5	88	102	1	14	100

n	14
Valor máximo	100
Valor mínimo	34
Rango	66
k	4,690532
k*	5
Amplitud	13,2
Amplitud*	14
Diferencia	4

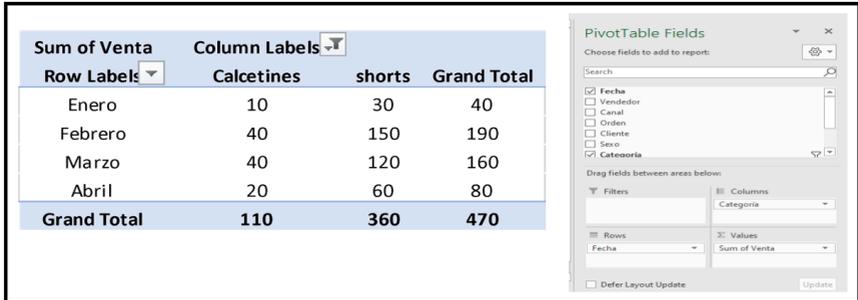
- La tendencia de las ventas se analiza mediante una serie de tiempo generada a partir de una tabla dinámica considerando los campos de fecha en las filas, orden en columnas y de venta en la sección de valor, manteniendo este último campo definido como suma de ventas (Figura 9). Los resultados no muestran una tendencia contundente del comportamiento de las ventas, más bien es importante atender las razones de las devoluciones, que se han incrementado respecto a los primeros meses.

Figura 9. Tabla dinámica, respuesta 5



- Para determinar la conveniencia de una promoción entre dos productos de la empresa, se estudia la relación que existe entre ellos a través de la regresión lineal. Se extraen los datos que interesan mediante una tabla dinámica, donde el campo fecha está en las filas, el campo categoría o producto en las columnas, y suma de ventas en la casilla de valor. (Figura 10).

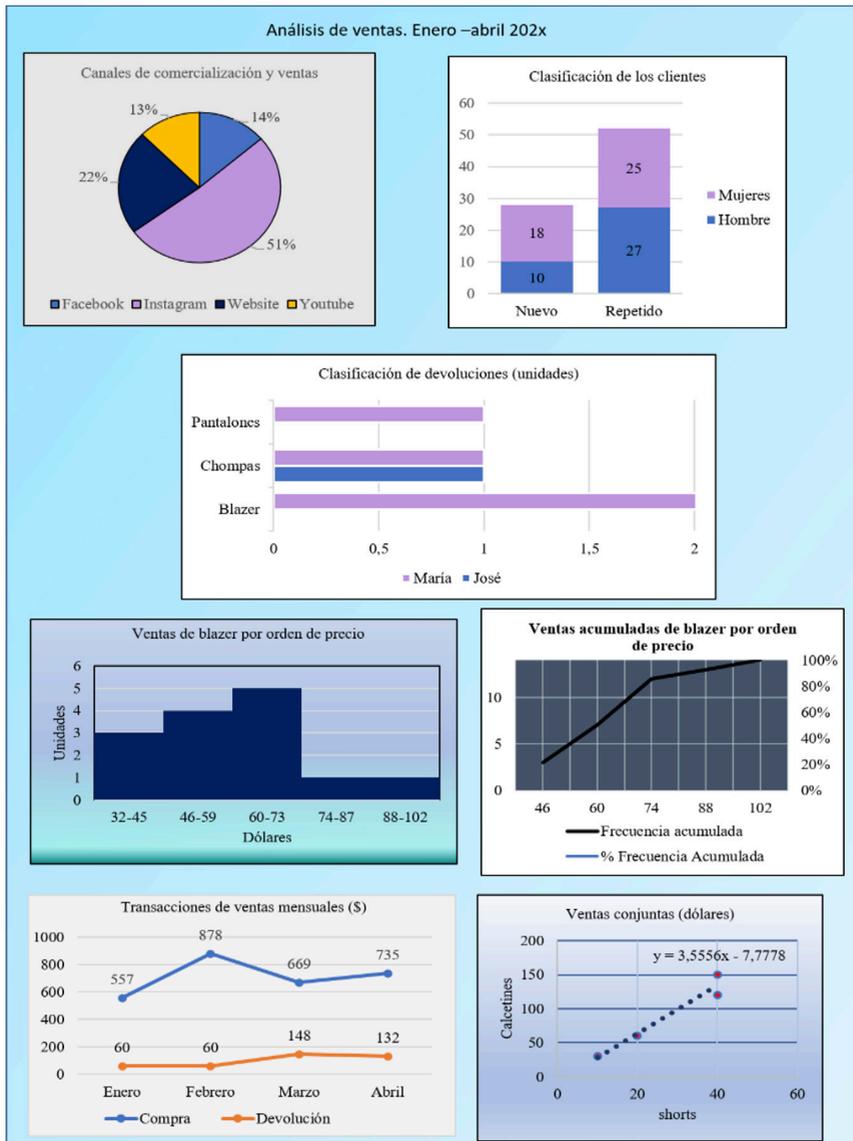
Figura 10. Tabla dinámica, respuesta 6



Se calcula el coeficiente de correlación entre las dos variables utilizando la fórmula de Excel correlación (rango variable 1, rango variable 2) el cual resultó de 0,97, evidenciando una asociación lineal positiva, por lo que se afirma que, si conviene una promoción entre estos dos productos, pues al incrementar las ventas de uno de ellos, también aumentan las ventas del otro. Al hacer el gráfico de dispersión se puede determinar la recta que define la relación entre las variables. Toda la información obtenida del análisis estadístico de los datos puede ser resumida

en un informe gráfico o tablero gerencial donde se presente la gestión de la empresa en el periodo de estudio, tal cual se muestra en la Figura 11.

Figura 11. Resultados de Ventas



Consideraciones finales

El mundo de hoy se caracteriza por la digitalización y la interconectividad, dando gran relevancia a los espacios físicos-virtuales, donde se almacenan datos de diversas estructuras y de diferentes fuentes, los cuales son capturados mediante artefactos automáticamente, por ejemplo, con sensores (Yan, 2021). El uso de la estadística para analizar datos proviene de hace muchos años, sin embargo, la evolución de la data al incrementarse su velocidad de reproducción y complejidad y a la necesidad de las organizaciones de tener respuestas en el momento requerido, ha repercutido en que, realizar estudios estadísticos con los métodos tradicionales es casi imposible. Es aquí, donde la tecnología ha beneficiado todo el proceso, pues interviene en la captura, limpieza, almacenamiento, análisis (aprendizaje automático, minería de datos, procesamiento de análisis en línea *OLAP*, entre otros) y finalmente en las predicciones y recomendaciones. Pero no hay que olvidar que la raíz del análisis de los datos proviene de conceptos de estadística, siendo la estadística descriptiva la base de la analítica descriptiva, que de acuerdo con Frazzetto (2019) sirve para responder ¿Qué pasa en la organización?, el resto de esta disciplina contribuye con el desarrollo de los modelos predictivos (¿Qué va a pasar?) y modelos prescriptivos (¿Cómo va a pasar?).

Referencias bibliográficas

- A2 Capacitación: Excel. (2021). *Mini curso de Business Intelligence en Excel - Tablas Dinámicas, Gráficas y Dashboards - Parte 1*. [Video] Youtube. <https://www.youtube.com/watch?v=TgcGa0cnIpg>
- Bär, K., Herbert-Hansen, Z.N.L. & Khalid, W. (2018). *Production Engineering. Considering Industry 4.0 aspects in the supply chain for an SME*. Volumen (12). Consultado el 10 de febrero del 2021 en <https://doi.org/10.1007/s11740-018-0851-y>
- Eggert, M., Alberts, J. (2020). *Bus Res. Frontiers of business intelligence and analytics 3.0: a taxonomy-based literature review and research agenda*. Volumen (13). Consultado el

15 de febrero del 2021 en <https://doi.org/10.1007/s40685-020-00108-y>

- Frazzetto, D., Nielsen, T.D., Pedersen, T.B. et al. (2019). *The VLDB Journal. Prescriptive analytics: a survey of emerging trends and technologies*. Volumen (28). Consultado el 18 de febrero del 2021 en <https://doi.org/10.1007/s00778-019-00539-y>
- Lind, D., Marchal, W. y Wathen, S. (2012). *Estadística aplicada a los negocios y la economía*. 15a. ed. México: McGraw-Hill
- Machado, W. y Rodríguez Balza, M. (2019). *Criterios para la selección y aplicación de herramientas estadísticas en la investigación educativa Delectus. Revista de investigación y capacitación continua*. Volumen (2). Consultado el 10 de enero del 2021 en <https://doi.org/10.36996/delectus.v2i1.8>
- Mendenhall, W., Beaver, R. y Beaver, B. (2006). *Introducción a la probabilidad y estadística*. 13a ed. Mexico: Cengage Learning.
- Walpole, Ronald E. (2012). *Probabilidad y estadística para ingeniería y ciencias*. 9a. ed. México: Pearson Educación.
- Yang, F., Gu, S. (2021). *Complex & Intelligent Systems. Industry 4.0, a revolution that requires technology and national strategies*. Volumen (1). <https://doi.org/10.1007/s40747-020-00267-9>

Anexo A. Disponible en: https://docs.google.com/spreadsheets/d/1SeITwonkxMcbu6b8zGdu_ympL2jKNA3X/edit?usp=sharing&ouid=102475912454646469429&rtpof=true&sd=true

Datos de Ventas Ecuatextil

Fecha	Vendedor	Canal	Orden	Cliente	Sexo	Categoría	Código	Venta
Enero	Tienda	Website	Compra	Repetido	Hombre	Pantalones	100	40
Enero	Tienda	Website	Compra	Repetido	Hombre	Chompas	120	50
Febrero	Tienda	Website	Compra	Repetido	Mujeres	Leggings	200	36
Febrero	Tienda	Website	Compra	Repetido	Hombre	Camisetas	140	45

Marzo	Tienda	Website	Compra	Repetido	Hombre	Calcetines	160	10
Marzo	Tienda	Website	Compra	Repetido	Mujeres	shorts	220	30
Marzo	Tienda	Website	Compra	Repetido	Mujeres	shorts	220	30
Abril	Tienda	Website	Compra	Repetido	Hombre	Pantalones	101	48
Abril	Tienda	Website	Compra	Repetido	Hombre	Pantalones	100	40
Febrero	Ana	Facebook	Compra	Repetido	Hombre	Pantalones	101	48
Febrero	Ana	Facebook	Compra	Repetido	Mujeres	shorts	220	30
Febrero	Ana	Facebook	Compra	Repetido	Hombre	Calcetines	160	10
Febrero	Ana	Facebook	Compra	Repetido	Hombre	Chompas	121	60
Febrero	Ana	Facebook	Compra	Repetido	Mujeres	Blazer	284	65
Febrero	Ana	Facebook	Compra	Repetido	Hombre	Camisetas	141	60
Febrero	Ana	Facebook	Compra	Repetido	Hombre	Pantalones	100	40
Febrero	Ana	Facebook	Compra	Repetido	Mujeres	Shorts	220	30
Marzo	Ana	Facebook	Compra	Repetido	Hombre	Pantalones	100	40
Marzo	Ana	Facebook	Compra	Repetido	Mujeres	Shorts	220	30
Marzo	Ana	Facebook	Compra	Repetido	Mujeres	Blazer	282	50
Enero	María	Instagram	Compra	Nuevo	Mujeres	Blusas	262	30
Marzo	María	Instagram	Compra	Nuevo	Mujeres	shorts	220	30
Marzo	María	Instagram	Devolución	Nuevo	Mujeres	Blazer	286	100
Abril	María	Instagram	Compra	Nuevo	Mujeres	Pantalones	241	60
Febrero	María	Instagram	Compra	Nuevo	Mujeres	Blazer	284	65
Abril	María	Instagram	Compra	Nuevo	Mujeres	Blusas	263	30
Marzo	María	Instagram	Compra	Nuevo	Hombre	Camisetas	140	45
Marzo	María	Instagram	Compra	Nuevo	Hombre	Calcetines	160	10
Marzo	María	Instagram	Devolución	Nuevo	Hombre	Pantalones	101	48
Marzo	María	Instagram	Compra	Nuevo	Mujeres	Blazer	283	55
Abril	María	Instagram	Compra	Nuevo	Mujeres	Blazer	284	65
Abril	María	Instagram	Compra	Nuevo	Mujeres	Blusas	261	28
Abril	María	Instagram	Devolución	Nuevo	Mujeres	Blazer	285	82

Abril	María	Instagram	Compra	Nuevo	Mujeres	Pantalones	240	46
Enero	José	Youtube	Compra	Repetido	Hombre	Camisetas	142	50
Enero	José	Youtube	Compra	Repetido	Mujeres	Blazer	282	50
Enero	José	Youtube	Compra	Repetido	Hombre	Pantalones	101	48
Marzo	José	Youtube	Compra	Repetido	Mujeres	Blusas	262	30
Marzo	José	Youtube	Compra	Repetido	Mujeres	Leggings	201	40
Marzo	José	Youtube	Compra	Repetido	Hombre	Chompas	121	60
Marzo	José	Youtube	Compra	Repetido	Hombre	Pantalones	101	48
Abril	José	Youtube	Compra	Repetido	Mujeres	Blazer	281	42
Febrero	José	Youtube	Devolución	Repetido	Hombre	Chompas	121	60
Febrero	José	Youtube	Compra	Repetido	Mujeres	shorts	220	30
Abril	Luis	Instagram	Compra	Nuevo	Mujeres	Blazer	284	65
Abril	Luis	Instagram	Compra	Nuevo	Hombre	Calcetines	160	10
Abril	María	Instagram	Compra	Repetido	Mujeres	Leggings	200	36
Abril	María	Instagram	Compra	Repetido	Mujeres	shorts	220	30
Abril	Tienda	Website	Compra	Repetido	Hombre	Camisetas	140	45
Febrero	Sandra	Instagram	Compra	Repetido	Mujeres	Blazer	283	65
Febrero	Sandra	Instagram	Compra	Repetido	Mujeres	Shorts	220	30
Enero	Sandra	Instagram	Compra	Repetido	Hombre	Calcetines	160	10
Enero	Sandra	Instagram	Compra	Repetido	Hombre	Pantalones	101	48
Marzo	Sandra	Instagram	Compra	Repetido	Hombre	Calcetines	160	10
Abril	Sandra	Instagram	Compra	Repetido	Mujeres	Leggings	202	54
Enero	Sandra	Instagram	Compra	Repetido	Hombre	Camisetas	140	45
Marzo	Sandra	Instagram	Compra	Repetido	Hombre	Chompas	122	55
Marzo	Sandra	Instagram	Compra	Repetido	Mujeres	Leggings	200	36
Febrero	Sandra	Instagram	Compra	Repetido	Mujeres	shorts	220	30

Abril	San-dra	Instagram	Compra	Repetido	Mujeres	Shorts	220	30
Febrero	María	Instagram	Compra	Nuevo	Hombre	Calcetines	160	10
Febrero	María	Instagram	Compra	Nuevo	Mujeres	Pantalones	240	46
Enero	María	Instagram	Compra	Nuevo	Mujeres	Pantalones	240	46
Enero	María	Instagram	Compra	Nuevo	Hombre	Pantalones	100	40
Enero	María	Instagram	Devolución	Nuevo	Hombre	Chompas	121	60
Febrero	Luis	Instagram	Compra	Nuevo	Mujeres	Leggings	201	40
Marzo	Luis	Instagram	Compra	Nuevo	Hombre	Calcetines	160	10
Marzo	Luis	Instagram	Compra	Nuevo	Hombre	Camisetas	142	50
Abril	María	Instagram	Devolución	Nuevo	Mujeres	Blazer	282	50
Abril	Luis	Instagram	Compra	Nuevo	Hombre	Calcetines	160	10
Abril	Luis	Instagram	Compra	Nuevo	Mujeres	Leggings	202	54
Abril	Luis	Instagram	Compra	Nuevo	Mujeres	Blazer	281	42
Enero	Tienda	Website	Compra	Repetido	Mujeres	shorts	220	30
Enero	Tienda	Website	Compra	Repetido	Mujeres	Blusas	264	30
Enero	Tienda	Website	Compra	Repetido	Hombre	Pantalones	100	40
Febrero	Tienda	Website	Compra	Repetido	Hombre	Calcetines	160	10
Febrero	Tienda	Website	Compra	Repetido	Hombre	Chompas	121	60
Febrero	Tienda	Website	Compra	Repetido	Mujeres	Blazer	280	34
Febrero	Tienda	Website	Compra	Repetido	Mujeres	Blusas	260	24
Febrero	Tienda	Website	Compra	Repetido	Hombre	Calcetines	160	10

Nota: Adaptado de :Mini curso de Business Intelligence en Excel - Tablas Dinámicas, Gráficas y Dashboards – Parte 1, A2 Capacitación: Excel. <https://www.youtube.com/watch?v=TgcGa0cnlpg>



Big Data: Ecosistema Hadoop

Stalin Marcelo Arciniegas Aguirre

smarciniegas@pucesi.edu.ec

Pontificia Universidad Católica del Ecuador - Sede Ibarra

Escuela de Ingeniería

Ecuador

Introducción

Hoy en día el crecimiento de datos es realmente abrumador. Es así como, en el 2012 la empresa IBM publicó una infografía que se basa en la IDC (International Data Corporation) (IBM, 2012), donde revela los datos que inundan la web, en la cual se indica que diariamente se manda 294 millones de correos, se suben 100 terabytes de datos a Facebook y que se generan aproximadamente 5 exabytes de datos cada dos días. La misma IDC en el 2018 publicó en su revista “The Digitization of the World” que para el 2025 la cantidad de datos en el mundo será de aproximadamente 175 ZB (Zeta byte), razón por la cual es necesario contar con herramientas tecnológicas que capturen, almacenen y analicen esta información. En este capítulo vamos a conocer las principales herramientas dentro del ecosistema Hadoop, así como sus principales características y configuraciones. Hadoop es una estructura de código abierto que permite la integración de herramientas, con la finalidad de almacenar y procesar datos de enorme capacidad y de cualquier tipo.

103

Big Data

Big Data es un término que se usa cuando tenemos una cantidad considerable de datos que supera notablemente los almacenamientos convencionales en bases de datos, pero el término no solo hace referencia al tamaño de la información, sino también a la variedad del contenido y la velocidad con la cual se genera y se analizan, es por lo que (Gartner, 2012) definió las 3 V del Big Data. Volumen, Variedad y Velocidad.

a)

a) **Volumen.** Hace referencia a la cantidad de datos que se genera diariamente, hoy en día más del 80% de fuente de información almacenada, proviene de los datos no estructurados como son la voz, imagen, video etc.

b) **Variedad.** Se trata de las diferentes fuentes que pueden aportar datos y estos pueden ser estructurados, semiestructurados y no estructurados, con la finalidad combinarse para dar valor a los datos, estas pueden provenir de redes sociales, datos públicos, bases de datos, entre otras. Su principal objetivo es tener un data lake o lago de datos disponibles para ser analizados.

c) **Velocidad.** Es la velocidad con la cual se generan, almacenan y se analizan los datos, hoy en día existen muchas herramientas que pueden procesar esta información en tiempo real.

Hoy en día la necesidad de las empresas ha hecho que se integren 4 V, una de ellas es la **Veracidad** de los datos la que hace referencia a que tan integro y fiable son los datos adquiridos, la otra es **Valor** que hace referencia al beneficio para las empresas, por otro lado tenemos las **Viabilidad de los datos** que permite determinar si los datos obtenidos tendrán impacto en los proyectos, y finalmente la **Visualización de datos** que nos hace más entendible los datos resultados obtenidos.

Este conjunto de características del Big Data hace que se tengan varios beneficios en diferentes áreas que se detallan a continuación:

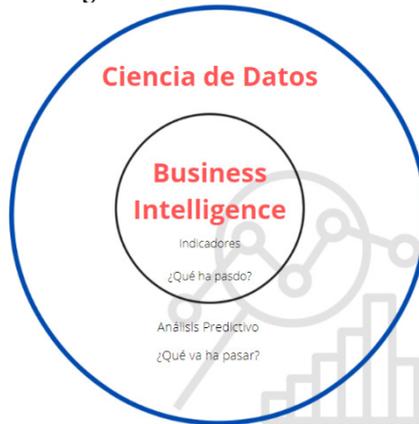
1. Mejora en la eficiencia operativa. En esta área se encuentran aplicaciones relacionadas con el tiempo real, data streaming, procesamiento de imagen y video, data lakes, entre otras.
2. Generación de nuevo ingresos. En esta parte, se encuentran aplicaciones como visión 360 del cliente, motores de recomendación, análisis predictivo, etc.
3. Prevención y predicción del fraude. Las aplicaciones de esta área están relacionadas con la seguridad del cliente, prevención de mora.

Por lo tanto, el objetivo principal que persigue el Big Data es sacar el mejor provecho a los datos, que se consiguen con la aplicación de ciencia de datos, los nuevos perfiles profesionales como es el Arquitecto de Big Data, Ingeniero de Big Data y Científico de Datos y las arquitecturas tecnológicas que hoy en día están a la disposición.

1. Ciencia de Datos

La ciencia de datos es un campo interdisciplinar que tiene como objetivo sacar el mejor provecho a los datos por medio de la analítica, en este campo interviene la estadística para resumir y modelar, la minería de datos, aprendizaje automático y el análisis predictivo. (Briega, 2016).

Figura 1: Ciencia de Datos

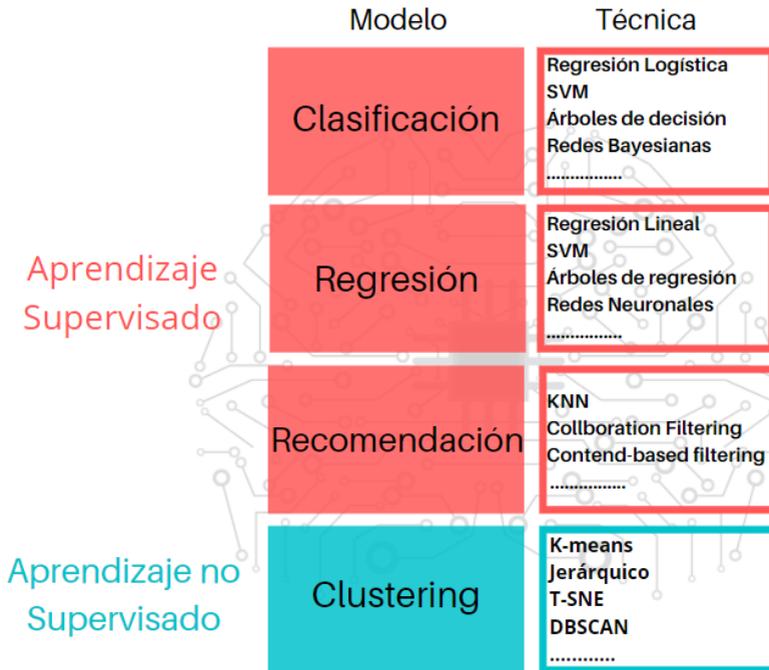


Hace algunos años se tomaba como referencia el *Business Intelligence* para la toma de decisiones, ya que aportaba con indicadores que podían medir los procesos, sin embargo, siempre se estaba tomando como referencia lo que ya pasó. Es por ello, que muchas empresas invierten que proyectos en Big Data, porque se realiza un análisis predictivo, por lo tanto, va a hacer referencia a lo que sucederá, como se observa en la figura 1.

La parte medular de la ciencia de datos son los algoritmos de Inteligencia Artificial que manejan datos, ellos forman parte de la Minería de Datos, y tienen la finalidad de explorar y sacarles el máximo valor. Dentro de este campo, tenemos 2 tipos de aprendizaje. El supervisado y no supervisado, en la figura 2 se indica a detalle los modelos y técnicas usadas en la minería de datos.

El aprendizaje supervisado utiliza datos para predecir posibles valores con la información disponible, dentro de este tipo de aprendizaje, tenemos varios modelos, como son los de clasificación, regresión y recomendación. La característica principal que tienen estos modelos es que son entrenados previamente con el objetivo de ajustarlos.

Figura 2: Desglose de la Minería de Datos

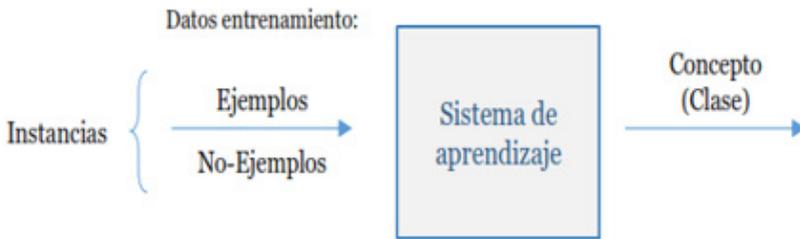


Los elementos que interviene dentro del aprendizaje supervisado son los Datos entrenamiento, las instancias y el concepto, en primer lugar, los datos de entrenamiento son aquellos permiten ajustar los parámetros de los algoritmos y esta divididos en ejemplos, que

son aquellos que sirven para entrenar, y los no ejemplos los que sirven para valorar la eficiencia de los algoritmos. Por lo general, la proporción es de 70% para datos de entrenamiento y 30% para datos de prueba.

A continuación, en la figura 3 se detalla en proceso de entrenamiento y cada una de las características, en primer lugar, las instancias se refieren a cada una de las filas donde se encuentran los ejemplos y no ejemplos, y finalmente la clase o concepto se hace referencia al valor a predecir. Por ejemplo, un conjunto de síntomas como el dolor de cabeza, resfriado, malestar serían los datos de entrenamiento y la clase es la enfermedad que puede ser una gripe.

Figura 3: Proceso de entrenamiento



Modelo de Clasificación

Los modelos de clasificación o discriminación buscan predecir una clase utilizando diferentes técnicas basándose en las probabilidades, como la regresión logística, árboles de decisión, redes bayesianas, máquinas de soporte vectorial (SVM) entre otros. La principal característica de estas técnicas es que las instancias son de tipo categórica, es decir valores no numéricos y la clase también.

Modelo de Regresión

A diferencia del modelo de clasificación, este tipo de modelo utiliza valores numéricos en sus instancias, así como en la clase. Permitiendo así, una aproximación numérica de la variable

a predecir. Las principales técnicas son árboles de regresión, máquinas de soporte vectorial, regresión lineal y redes neuronales.

Modelo de Recomendación

Los modelos de recomendación están enfocados en dar a escoger al usuario, lo que le podría gustar de un amplio contenido que se tiene hoy en internet. En donde más se utiliza este tipo de algoritmos es en las compras como Amazon, en las plataformas de video como YouTube etc.

Finalmente, el aprendizaje no supervisado es cuando se desconoces la clase y lo que se busca son patrones que ayuden agrupar los datos, para posteriormente modelarlos. Las técnicas más utilizadas son K-means, Jerárquico, Tscan. Debido a la gran cantidad de datos muchas de las veces se aplican el aprendizaje no supervisado y luego el aprendizaje supervisado.

Como comentario final, se puede decir que ninguno de los algoritmos mostrados tiene una eficiencia del 100%, y que ni uno es mejor que otro. Lo que tenemos que hacer es probar cada uno de ellos y verificar cual se acopla mejor a los datos. De esa manera podemos decidir con cual algoritmo implementar la solución. Además, si es un sistema continuo de ingreso de datos, se tiene que realizar calibraciones periódicas para ajustar los modelos.

2. Arquitecturas tecnológicas

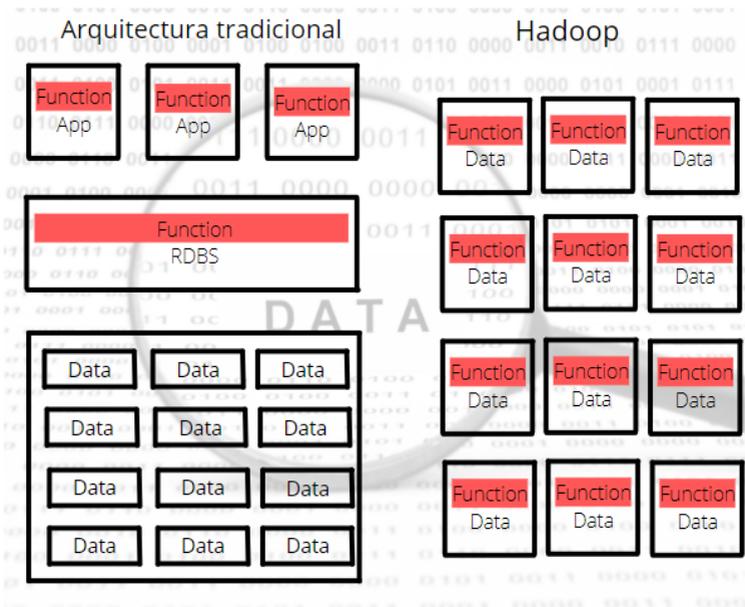
Hadoop es un framework para el almacenamiento y procesamiento de información en forma distribuida desarrollado por Apache, por lo tanto, es de código abierto. Hadoop esta formado por dos partes, la primera es el sistema de almacenamiento distribuido HDFS (Hadoop Distribuid File System), y Map Reduce para el procesamiento distribuido.

Como antecedente, Hadoop fue creado en el 2006 por Doug Cutting y Mike Cafarella, luego de que Google publicara un

artículo científico sobre el manejo de datos distribuido GFS (Google File System). En el cual, se propone una arquitectura que dé respuesta al creciente almacenamiento de datos. El nombre Hadoop se origina de un elefante amarillo que tenía como juguete la hija de Doug Cutting.

La arquitectura tradicional para el almacenamiento de datos está compuesta por aplicaciones que interactúan con los usuarios, luego existe un cuello de botella que es el administrado de la base de datos (RDBS), el cual conoce la ubicación de los datos. En cambio, la arquitectura Hadoop está compuesta por bloques de 64 MB configurable, eliminándose el RDBS como se observa en la figura 4. De esta manera, se agiliza el acceso a los datos y se dinamiza el procesamiento, ya que múltiples usuarios pueden acceder a la información almacenada. Además, agiliza el trabajo de los algoritmos de minería de datos, porque acceden varias veces a los bloques mientras se los está entrenando.

Figura 4: Arquitectura tradicional vs Hadoop



Ecosistema Hadoop

El ecosistema Hadoop es el conjunto de varias herramientas que hacen posible la integración de estas, para que trabajen en conjunto. Hoy en día existen distribuciones donde están las herramientas instaladas y listas para funcionar, las distribuciones más importantes son Cloudera, Hortonworks, MapR, Amazon, entre otras.

A continuación, vamos a describir los componentes básicas que conforman el ecosistema Hadoop:

Hardware. Es donde se alojarán todas las herramientas y será la base para el correcto desempeño.

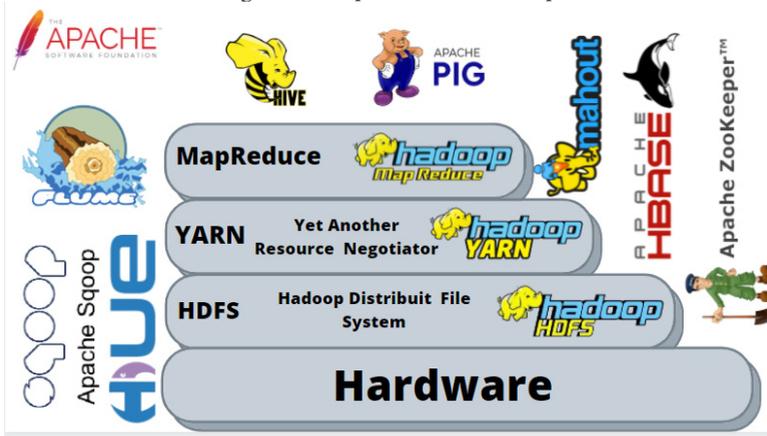
HDFS. Hadoop Distributed File System se encarga del almacenamiento en bloques, y su estructura es tolerante a fallos ya que trabaja en paralelo y se encarga automáticamente de la replicación de los datos.

YARN. Es el que permite realizar multitareas entre el procesamiento y el acceso a los datos. Es fundamental para que varias aplicaciones puedan utilizar los datos de los bloques y pueden realizar las tareas de Map Reduce

Map Reduce: Es el paradigma de programación en paralelo, que ayuda a sacar el mejor rendimiento a los bloques HDFS, y cumple dos tareas fundamentales, la de mapear los datos y luego reducirlos.

Además, como el ecosistema de Hadoop es un proyecto de código abierto existen muchas herramientas que pueden ser desarrolladas y acopladas a este ecosistema como se muestra en la figura 5. A continuación, vamos a nombrar las principales herramientas que conforman este ecosistema desde la distribución Cloudera.

Figura 5: Arquitectura Hadoop



HIVE: Es una herramienta desarrollada por Facebook para permitir interactuar con el ecosistema Hadoop, utilizando el lenguaje de programación SQL. Eso facilita para que profesionales en las bases de datos tradicionales puedan adaptarse rápidamente. Un aspecto importante es que HIVE-SQL es una emulación del lenguaje SQL, por lo tanto, existen varias funciones que no se pueden utilizar.

SQOOP: Permite transferir grandes volúmenes de datos desde una estructura SQL al ecosistema Hadoop y viceversa. Esto vuelve más eficiente la transferencia de datos y permite el desarrollo de *Data Where House* o *Datalakes*, ya que se puede utilizar desde diferentes fuentes de datos.

FLUME: Es un ingestador de datos que utiliza como principal fuente las redes sociales y los logs que se generan a cada momento, permitiendo un flujo de datos estable, desde las diferentes fuentes hacia el ecosistema Hadoop.

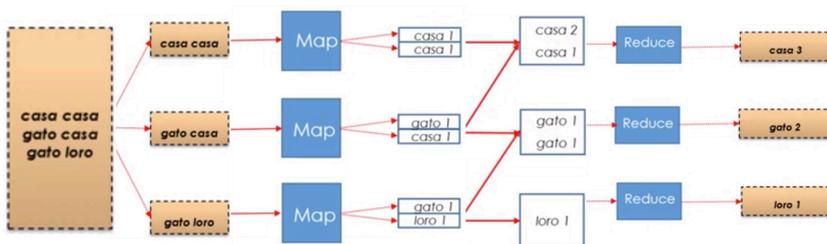
ZOOKEEPER: Es un coordinador de eventos, que permite a las aplicaciones acceder al entorno Hadoop de forma confiable. Ya que permite un consenso distribuido que organiza a los múltiples procesos para que no vayan a todos a la vez al mismo nodo.

MAHOUT: Es un proyecto que permite crear algoritmos de minería de datos, donde están incluidos los modelos básicos como son recomendación, clasificación y clustering. Como forma parte del proyecto Apache, facilita la integración con todo el Ecosistemas y las diferentes herramientas.

1.3 Map reduce

Map reduce es un paradigma de programación que explota el paralelismo en el análisis y procesamiento de grandes conjuntos de datos, en primer lugar, realiza un mapeo y luego la reducción. En la primera fase se divide por filas extraídas de cada bloque de datos, posteriormente estos pasan por un proceso de mapeo, en el cual se asigna una clave y un valor a cada dato (k, v) y devuelve tuplas de clave valor (kn,vm), a continuación, estos se agrupan por clave. Finalmente, la fase de reduce entrega los valores correspondientes a cada grupo de datos como se muestra en la Figura 6. Un aspecto importante, es que cada proceso se lo realiza en paralelo y de forma simultánea, reduciendo considerablemente los tiempos de proceso.

Figura 6: Proceso Map Reduce



Apache Spark es un proyecto de código abierto desarrollado por Apache, que tiene como objetivo optimizar los procesos de Big Data. Esto lo consigue al realizar las tareas en microlotes y no en lotes como lo hace Hadoop, además, un factor importante es que estas tareas se realizan en la memoria RAM, acelerando 100 veces más el procesamiento. Sin embargo, Spark se apoya en el sistema de ficheros HDFS del ecosistema Hadoop para sus operaciones.

En conclusión, el uso de ecosistema Hadoop con sus diversas herramientas, han permitido procesar datos muy grandes y rápidamente, siendo un complemento perfecto para los diversos algoritmos de minería de datos, para que finalmente las empresas brinden nuevos y mejores servicios a sus usuarios.

2. Ejemplo Práctico

Este trabajo busca familiarizar con el paradigma de programación MapReduce y que ganar confianza con algunas de las herramientas que forman parte del ecosistema de Hadoop. Para ello, te proporcionaremos un conjunto de datos y la descripción de un problema que debes resolver.

La resolución consta de dos partes:

- Una resolución conceptual empleando MapReduce, que no es necesario programar.
- Una resolución empleando Hive, en la que debemos escribir las consultas necesarias para obtener una solución.

Problema

Para esta actividad vamos a hacer uso de un dataset que formó parte de la competición **KDD Cup en el año 2012**.

En concreto, puedes descargar este dataset, así como ver una descripción exhaustiva de los ficheros que lo componen y su estructura en la siguiente dirección: <http://www.kddcup2012.org/c/kddcup2012-track2>.

De todos los ficheros disponibles, solo vamos a emplear dos para esta actividad: `training.txt` y `userid_profile.txt`.

El problema que debemos resolver es el siguiente: queremos saber cómo los diferentes perfiles de usuario (entendiendo por perfil de usuario un determinado sexo y grupo de edad)

interactúa con la publicidad que se le muestra. Para acotar la actividad, nos limitaremos a calcular el CTR (click-through rate) asociado a cada perfil para todos los anuncios mostrados.

La siguiente información puede ser relevante para entender mejor cómo abordar el problema:

- El CTR (click-through rate) es la relación entre anuncios mostrados y el número de clics realizados, y se calcula por tanto como $CTR = Clicks / Impressions$.
- El fichero training.txt contiene, entre otros campos, un identificador de usuario (UserID), el número de impresiones de un anuncio concreto en una determinada sesión (Impression) y el número de veces que el usuario hizo clic en ese anuncio durante esa sesión (Click). Los demás campos podemos obviarlos, pues contienen información del anuncio que es irrelevante para nuestro propósito.
- El fichero userid_profile.txt contiene el identificador del usuario (UserID) junto con su sexo (gender) y su rango de edad (age).
- La salida que se busca es una lista de perfiles de usuario junto con su CTR medio para todos los anuncios. El formato, por tanto, será similar al siguiente (los CTR mostrados no tienen por qué corresponderse con los reales), como se observa en tabla 1.

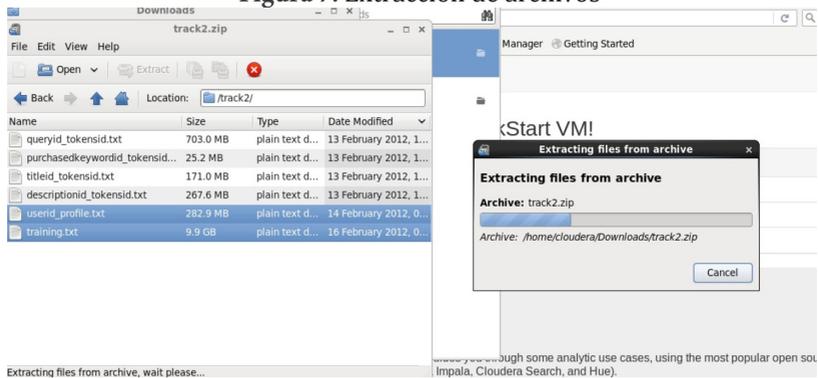
Tabla 1: Ejemplo CTR

Genero	Edad	CTR
Male	(0, 12)	0.74
Female	(0, 12)	0.79
Male	(12, 18)	0.68

Pasos previos a realizar la práctica

1. Descargamos VMware, además de Cloudera, e inicializamos.
2. Descarga del dataset requerido en formato .txt, guardamos en el entorno de Cloudera en el la carpeta home y ejecutamos los comandos para verificar la operacionalidad, para esto debemos registrarnos previamente en kaggle, ya que nos genera un token de acceso para poder clonar dicho repositorio como se observa en la figura 7.

Figura 7: Extracción de archivos



3. Creamos la carpeta `/user/pucesi` en el entorno hadoop `# hdfs dfs -mkdir /user/pucesi`
4. Creamos dentro de la carpeta “pucesi” una subcarpeta llamada “mapreduce”
5. Verificar que los servicios estén iniciados por medio de la terminal, ejecutando los comandos para verificar.

hadoop versión:

`hadoop 2.6.0 -cdh5.13.0`

hdfs dfsadmin report:

verificación de nodos

Creamos la carpeta “trb”:

`hdfs dfs -mkdir -p/user/trb`

6. Nota: en el caso de que salga el presente error:
mkdir: Permission denied: user=cloudera, access= WRITE, inode="/user" : hdfs:supergroup:drwxr-xr-x

Es porque no se tiene permisos para editar las carpetas y se debe ingresar el siguiente código dentro de HDFS Service Advance:

```
<property>  
<name>dfs.permissions</name>  
<value>>false</value>  
</property>
```

6. Desde hive creamos las tablas “training” y “userid_profile” que almacenarán el contenido de los archivos txt:

Se sugiere tener levantado los siguientes servicios:

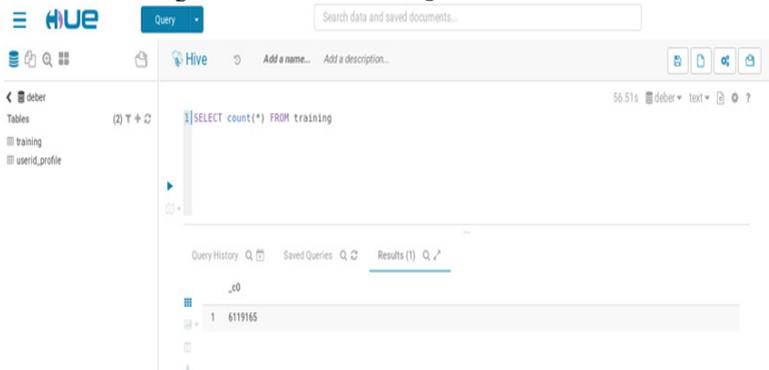
- HDFS
- Hive
- YARN
- ZooKeeper

7. Una vez creado las tablas, se puede revisar la DATA usando el comando:

```
SELECT count(*) FROM training
```

Se visualiza tabla training con la cantidad de líneas de datos 6' 119.165, a través de la Figura 8.

Figura 8: Cantidad de registros desde HUE



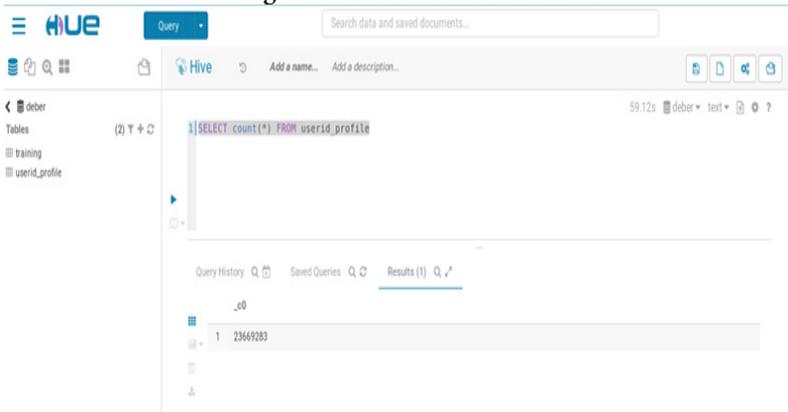
The screenshot displays the HUE web interface. At the top, there is a search bar and navigation icons. Below that, the Hive console is visible with the query `SELECT count(*) FROM training` entered. The results section shows a table with one row and one column, containing the value `6119165`. The interface also includes a sidebar with a table of contents and a bottom section for query history and saved queries.

Para visualizar en la figura 9 el número de filas de datos de la tabla utilizamos el comando:

```
SELECT count(*) FROM userid_profile
```

Total de registros: 23'669.283.

Figura 9: Extracción de archivos



The screenshot shows the Hive web interface. The top navigation bar includes the Hive logo, a search bar, and utility icons. The left sidebar displays a tree view of tables: 'deber', 'training', and 'userid_profile'. The main area contains a SQL query editor with the text: `1 | SELECT count(*) FROM userid_profile`. Below the editor, the 'Results (1)' tab is active, showing a single row of data: `1 | 23669283`. The interface also shows 'Query History' and 'Saved Queries' tabs.

- El ejercicio solicita el CTR (click-through rate) es la relación entre anuncios mostrados y el número de clics realizados, y se calcula por tanto como $CTR = Clicks / Impressions$.

En referencia a esta consulta se utilizó el comando SQL, limitando el número de registros indicado en la Figura 10.

Figura 10: Consulta SQL

The screenshot shows the Hue web interface. At the top, there is a search bar and a 'Query' dropdown menu. Below the search bar, there are navigation icons and a 'Hive' header with 'Add a name...' and 'Add a description...' options. On the left side, there is a sidebar with a 'Tables' section showing a tree view of the database structure, including a 'training' table with various fields like 'id_user', 'impression', 'click', and 'userid_profile'. The main area displays a SQL query: `SELECT (click/impression) from training limit 10;_`. Below the query editor, there is a 'Query History' section and a 'Results (20)' section. The results are displayed in a table with 5 rows and 1 column labeled '_c0'. The values in the table are: 4.2981186814246446e+18, 4.8605714994285804e+18, NULL, NULL, and 3.2847602447996047e+18.

	_c0
1	4.2981186814246446e+18
2	4.8605714994285804e+18
3	NULL
4	NULL
5	3.2847602447996047e+18

El fichero training.txt contiene, entre otros campos, un identificador de usuario (UserID), el número de impresiones de un anuncio concreto en una determinada sesión (Impression) y el número de veces que el usuario hizo clic en ese anuncio durante esa sesión (Click). Los demás campos podemos obviarlos, pues contienen información del anuncio que es irrelevante para nuestro propósito.

- El fichero userid_profile.txt contiene el identificador del usuario (UserID) junto con su sexo (gender) y su rango de edad (age).

- La salida que se busca es una lista de perfiles de usuario junto con su CTR medio para todos los anuncios. El formato, por tanto, será similar al siguiente (los CTR mostrados no tienen por qué corresponderse con los reales):

Los datos obtenidos de la consulta se visualizan en la Figura 11

Figura 11: CTR final

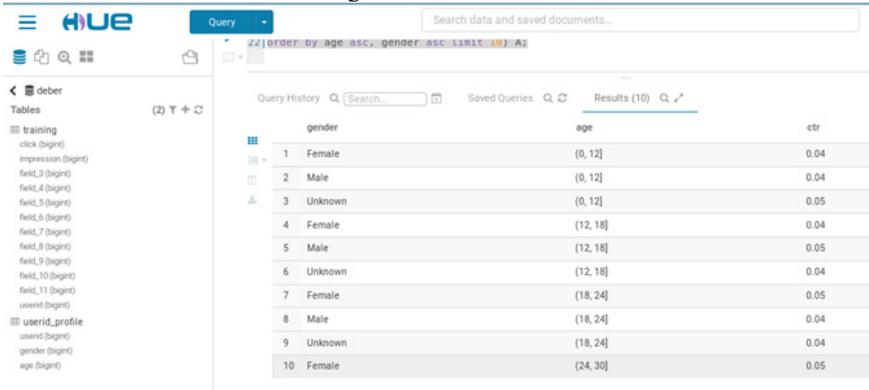


Figura 12: CTR Final

Respuesta de la información solicitada:			
	GENERO	EDAD	CTR
1	Female	(0, 12]	0.04
2	Male	(0, 12]	0.04
3	Unknown	(0, 12]	0.05
4	Female	(12, 18]	0.04
5	Male	(12, 18]	0.05
6	Unknown	(12, 18]	0.04
7	Female	(18, 24]	0.05
8	Male	(18, 24]	0.04
9	Unknown	(18, 24]	0.04
10	Female	(24, 30]	0.05

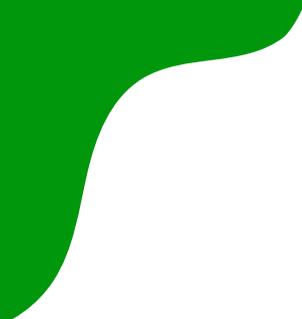
El procesamiento masivo de información utilizando el ecosistema distribuido Hadoop a través del paradigma de programación Map-Reduce permite aumentar la velocidad en la obtención de resultados, sin recurrir al aumento de las capacidades del Hardware. De esta forma se puede ahorrar recursos y tener resultados en un menor tiempo. Además, es ideal para trabajar con algoritmos de Inteligencia de Artificial ya que se necesita muchas interacciones en los entrenamientos y una limitación son las capacidades de hardware.

Otra de las ventajas es que las herramientas como HIVE permiten que se adapte el lenguaje SQL a esta nueva forma de trabajo, y de esta manera sea más rápida la adaptación de las profesiones en esta rama.

Bibliografía

- Briega, R. E. (2016). IAAR. Obtenido de <https://iaarbook.github.io/>
- Garcia, J. (2018). Ciancia de Datos . Colombia : Alfaohmega.
- Gartner. (19 de 11 de 2012). techtarget. Obtenido de <https://searchdatacenter.techtarget.com/es/noticias/2240171952/Reporte-de-Gartner-analiza-big-data-alrededor-de-tecnologia-de-datos>
- IBM. (2012). Obtenido de <https://ticsyformacion.com/2013/01/24/datos-sobre-ibm-en-2012-infografia-infographic/>
- CDH.(2014) Cloudera. Disponible en: <http://www.cloudera.com/content/cloudera/en/products-ndservices/cdh.html> .
- Dean, J. y Ghemawat, S. (2004). MapReduce: Simplified Data Processing On Large Clusters. In Proceedings of the 6th Symposium on Operating System Design and Implementation. Driving Marketing Effectiveness by Managing the Flood of Big Data. (2012). IBM.
- Disponible en: <http://www.ibmbigdatahub.com/infographic/flood-big-data> .
- Ghemawat, S., Gobioff, H. y Leung, S. (2003). The Google File System. In Proceedings of the 19th ACM Symposium on Operating Systems Principles. Under the Hood: Hadoop Distributed Filesystem reliability with Namenode and Avatarnode. (2012). Disponible en: <https://www.facebook.com/notes/facebookengineering/under-the-hood-hadoop-distributed-file-system-reliability-withnamenode-and-avata/10150888759153920> .
- Echeverry, G (). Ecosistema Hadoop. Universidad Autónoma de Occidente.
- Gates, A. (2011). Programming Pig. Massachusetts: O'Reilly.
- Giacomelli, P. (2013). Apache Mahout Cookbook. Birmingham: Packt Publishing

- Lee, D. (2013). Apache Hive Essentials How-to. Birmingham: Packt Publishing.
- Miner, D. y Shook, A. (2012). MapReduce Design Patterns: Building Effective Algorithms and Analytics for Hadoop and Other Systems. Massachusetts: O'Reilly.
- Perera, S. (2013). MapReduce Patterns: Hadoop Essentials How-to. Birmingham: Packt Publishing.
- Perera, S. y Gunarathne, T. (2013). Hadoop MapReduce Cookbook. Birmingham: Packt Publishing.
- Tannir, K. (2014). Optimizing Hadoop for MapReduce. Birmingham: Packt Publishing.



La administración de proyectos y el sistema ágil, ¿cuál es la tendencia en la gestión ágil de proyectos de software?

Pusdá Chulde Segundo Eliceo

sepusda@pucesi.edu.ec

Pontificia Universidad Católica del Ecuador Sede Ibarra
Escuela de Ingeniería – Ecuador

1. Introducción

Dentro de la industria del desarrollo de software para que un producto cumpla con los estándares mínimos de calidad debe contemplar la aplicación de metodologías ágiles tanto en el ciclo de desarrollo como en la administración del proyecto, ahora, la pregunta clave en la industria es: cómo funcionan las metodologías ágiles, cómo se aplican y si puede realmente ayudar al equipo en la mejora continua de forma más eficiente y rápida en el desarrollo y ejecución de las actividades o responsabilidades. El pensamiento, filosofía o cultura de trabajo de DevOps, juntamente con las metodologías ágiles es un proceso mediante el que un equipo puede administrar un proyecto dividiéndolo en varias etapas, este proceso involucra una colaboración constante con las partes interesadas o *stakeholders* del proyecto, lo que conlleva a una mejora continua por medio de una comunicación efectiva y cooperación del equipo en cada etapa del proyecto.

Uno de los objetivos las metodologías ágiles es fomentar la responsabilidad del equipo de desarrollo a lo largo del proceso y ejecución del proyecto y no solo al final. Básicamente, los proyectos ágiles deben exhibir valores centrales y comportamientos de confianza, flexibilidad, empoderamiento y colaboración de todos los *stakeholders*. (Gutiérrez, Pérez de Celis, & Cossío, 2011)

Teniendo en cuenta la parte introductoria, este capítulo tiene la finalidad de realizar un aporte en el ámbito académico y a la industria de desarrollo de software. En el aspecto académico se desea aportar sobre la implementación de metodologías ágiles

en la gestión de proyectos de software, que permita realizar una adecuación o adaptación de nuevas filosofías de trabajo y gestión conjuntamente con las metodologías más usadas, tal como son DevOps y SCRUM, esto da paso a una mejora en la industria del software ya que permite hacer más eficiente y sencilla la puesta en práctica de las metodologías con la parte teórica. Finalmente, se desea brindar las bases para que el trabajo en equipo sea más eficiente, que mejore las relaciones interpersonales y profesionales lo que conlleva al cumplimiento de objetivos en común.

2. La Ingeniería de Software y la Gestión De Proyectos

2.1 Gestión de proyectos de software

El software es un producto que está inmerso en múltiples cambios o adaptaciones, sobre todo en el proceso de desarrollo, es por eso que la gestión de un proyecto no se la realiza de la misma forma que la creación de un producto físico que sigue las recomendaciones de las normativas o metodologías de procesos industriales. El punto de partida para una gestión eficiente es tener definido el alcance, recursos y presupuestos del proyecto, seguidamente definir estrategias para el proceso de desarrollo que vaya conjuntamente con las etapas del ciclo de vida del software para de esta manera analizar y optar de las formas de trabajar en equipo que vayan enfocados en lograr y cumplir objetivos en común.

Con el crecimiento y evolución de las diferentes metodologías conjuntamente con la complejidad que implica el desarrollo de un software empezando desde su análisis hasta la implementación, la etapa de diseño, construcción, gestión y operación han tenido muchos tropiezos, es por eso que se ha dado paso a la evolución e implementación de diferentes metodologías que adaptan el desarrollo por partes o iteraciones que vayan de la mano de la construcción de prototipos incrementales que sean a posterior sencillos de dar mantenimiento y soporte a los usuarios.

La fase de diseño, codificación y operativa han sido procesos que se gestionan por separado, por lo que se ha evidenciado que no ha generado los resultados esperados, esto ha sido el constructo teórico de la ingeniería y las metodologías clásicas, es por eso que, con DevOps se desarrolla cada fase del proceso de software conjuntamente con el equipo de operaciones, estas desventajas las menciona (Garzas, 2011), en su artículo de metodologías clásicas y las desventajas que tienen al desarrollar productos de software de acuerdo a los estándares actuales de la industria del software, las desventajas que presentan las tecnologías tradicionales son:

- Tener una fase de diseño muy separada de la codificación.
- Que la codificación no comienza hasta que el diseño no esté culminado.
- Que el diseño concluya con unos planos precisos que guíen totalmente la construcción.
- Que una vez que se hace un diseño este no se modifique.
- Falta de comunicación con el equipo de operaciones.

2.2. El Manifiesto Ágil

El manifiesto ágil hace referencia a los “Métodos Ágiles” que han dado el inicio a las metodologías que han sido alternativas para el desarrollo de proyectos de software de una manera eficiente y optimizando los procesos, frente a las metodologías tradicionales y estándares, por ejemplo: La metodología Rational Unified Process, por las siglas se la representa como (RUP), este tipo de metodologías se las considera excesivamente pesadas y rígidas por las normas, procedimientos y documentación que se deben aplicar y generar al momento de implementarlas en el desarrollo. (Amorim & Reis, 2020)

Según (Mitre, Ortega, & Lemus, 2014), los lineamientos que definen al manifiesto ágil son:

- Individuos e interacciones sobre procesos y herramientas
- Software funcionando sobre documentación extensiva
- Colaboración con el cliente sobre negociación contractual

- Respuesta ante el cambio sobre seguimiento de un plan

Dentro de estos cuatro valores se han definido los principios o valores que describen de mejor manera al manifiesto ágil, se describe la información citada desde (Corona, 2016):

- Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.
- Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.
- Los proyectos se desarrollan en torno a individuos motivados
- Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.

El manifiesto ágil ha dado apertura a las metodologías de desarrollo ágil, que se basan en ciclos de desarrollo cortos con objetivos y entregables específicos, así también los principios más relevantes son:

- El producto de software evoluciona conforme los ciclos avanzan.
- Los grupos de trabajo son multidisciplinarios y por lo general auto gestionados.

Los elementos inmersos en el desarrollo ágil de un producto de software son:

- Control de versiones: Diversas versiones de código, por ejemplo: Apache Subversion, mercurial, Github, GitLab, SourceForge.
- Gestión de entornos: Se refiere a los diferentes entornos que se tiene entre desarrollador y el cliente, tenga en cuenta que el entorno de desarrollo no es el mismo que

va a utilizar o manipular el cliente. Se recomienda utilizar herramientas que permiten virtualizar para simular de un entorno de desarrollo – pruebas – producción, por ejemplo: heroku, OpenShift, Docker Cloud, Dokku, Zeit, Elastic Beanstalk, Google app engine.

- **Automatización:** Este concepto hace referencia a la integración automática con otras infraestructuras, se recomienda las siguientes herramientas: Salt, chef, Puppet, Ansible, AnsibleTower, Terraform.
- **Pruebas y QA (Quality Assurance):** ¿Qué probar o validar?: Código (pruebas unitarias), integración, funcionalidad, el desarrollo ágil permite integrar estos diferentes tipos de pruebas que son parte importante para el aseguramiento de la calidad de un producto de software. QA surge como un compromiso de la calidad en los productos digitales. Se trata de un perfil que facilita la realización de pruebas de testeo para comprobar el funcionamiento del software.
- **Comunicación:** Uno de los conceptos clave en DevOps, ya que, si no sabe comunicar los errores, los requerimientos, o si no existe una estrategia adecuada de comunicación con el equipo no se van a lograr los resultados esperados.
- **La integración continua:** Se implementa dentro de los proyectos de software para unificar la parte de los desarrolladores con la infraestructura generada por el área de operaciones, lo cual ayuda a mejorar en tiempo de entrega del producto desarrollado.
- **Herramientas:** Slack, Stack Overflow, Jenkins, Travis, Buildbot, Selenium, Bugzilla.

2.3. DevOps

DevOps es un marco de trabajo y una filosofía en constante evolución que promueve un mejor desarrollo de aplicaciones en menos tiempo y la rápida publicación de nuevas funcionalidades

de software o productos para los clientes, estas actividades se las conoce como la automatización y la entrega continua del software. En las últimas dos décadas, el escenario de la tecnología ha cambiado radicalmente, las plataformas, las arquitecturas, el software y los métodos de comunicación también han experimentado cambios extremos, de tal magnitud que el mundo de la tecnología es totalmente irreconocible comparado con las generaciones anteriores, es por eso que la innovación tecnológica y el proceso de desarrollo también ha experimentado cambios significativos (MicroFocus, 2015). Con DevOps se promueve una comunicación continua más fluida, la colaboración, la integración, la visibilidad y la transparencia entre equipos de desarrollo de aplicaciones (Dev) y sus homólogos en los procesos de operaciones (Ops), (Mamani, Del Pino, & Gonzales, 2020).

Esta relación estrecha entre «Dev» y «Ops» se extiende a cada una de las fases del ciclo de vida de DevOps: desde la planificación inicial del software a las fases de codificación, compilación, pruebas y publicación y en la puesta en marcha, las operaciones y la supervisión continua. Esta relación impulsa un bucle de retroalimentación continua con los clientes sobre las mejoras, el desarrollo, las pruebas y la puesta en marcha. Uno de los resultados de todos estos esfuerzos puede ser la publicación continua y más rápida de las adiciones y los cambios que se necesitan en las funciones (Mamani, Del Pino, & Gonzales, 2020).

Algunas personas agrupan los objetivos de DevOps en cuatro categorías: cultura, automatización, medición y uso compartido (CAMS, por sus siglas en inglés), ya que DevOps es un paradigma que utiliza un conjunto de principios enfocado en la entrega e integración continua de software, esto implica una nueva cultura para desarrollar y desplegar software en contextos altamente colaborativos y ágiles, este paradigma involucra a todos los actores como desarrolladores, profesionales de TI, usuarios, stakeholders orientado a reducir la brecha que existe entre el desarrollo y operaciones (Dittrich, Nørbjerg, Tell, & Bendix, 2018).

En la figura 1 se muestra el proceso de interacción y trabajo al proceso con DevOps, teniendo como ejes centrales al área de desarrollo y operaciones, es decir que los equipos de desarrollo y de operaciones (o sistemas) eliminan el trabajo “en silos” y comienzan a trabajar de una manera colaborativa y bidireccional.

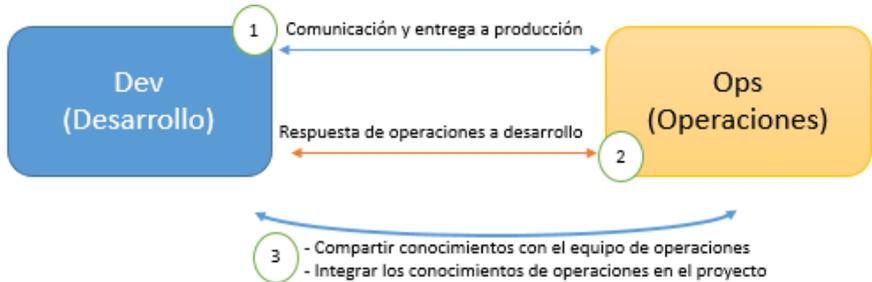


Figura 1: Áreas de relación DevOps

Fuente: (Jiménez Marco, 2016)

Según lo mencionado por (Mamani, Del Pino, & Gonzales, 2020), DevOps, la metodología que está cambiando la forma en desarrollar el software, no solo a nivel tecnológico, sino también empresarial, es por eso que aquellas organizaciones que hagan el esfuerzo por adoptar DevOps obtendrán una ventaja competitiva sobre las demás.

Otra característica de DevOps que se menciona en el portal de (Xeridia, 2017) es que DevOps establece una “intersección” entre Desarrollo, Operaciones y Calidad, pero no se rige por un marco estándar de prácticas, sino que permite una interpretación mucho más flexible en la medida en que cada organización quiera llevarlo a la práctica, según su estructura y circunstancias.

En la figura 2 se muestra el proceso de intersección de las áreas que comprende a un proceso completo de DevOps.

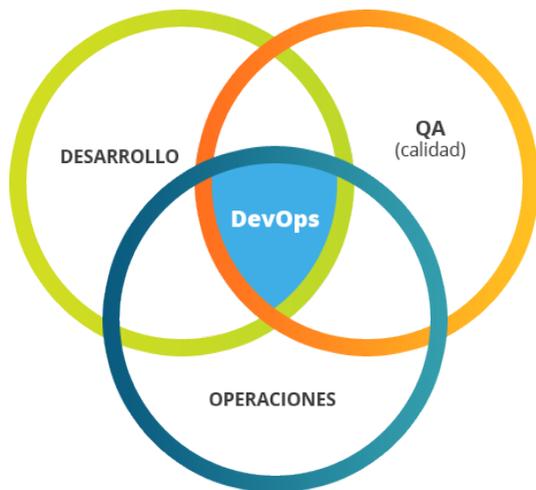


Figura 2: Intersección entre Desarrollo, Operaciones y Calidad

Fuente: (Xeridia, 2017)

2.3.1. Objetivos de DevOps

Los objetivos que describe a DevOps son los siguientes:

- a. DevOps desde sus inicios ha brindado lineamientos que permiten generar productos de software de una manera rápida, automatizada, eficiente y económica.
- b. Minimizar el riesgo de los cambios que se producen en las entregas y dar así un mayor valor tanto a los clientes como al propio negocio.

2.3.2. La cadena o ciclo de vida de DevOps

DevOps influye en el ciclo de vida de las aplicaciones a lo largo de las fases de planeamiento, desarrollo, entrega y uso. Cada fase depende de las demás y las fases no son específicas de un rol. En una auténtica cultura de DevOps, todos los roles están implicados de algún modo en todas las fases.

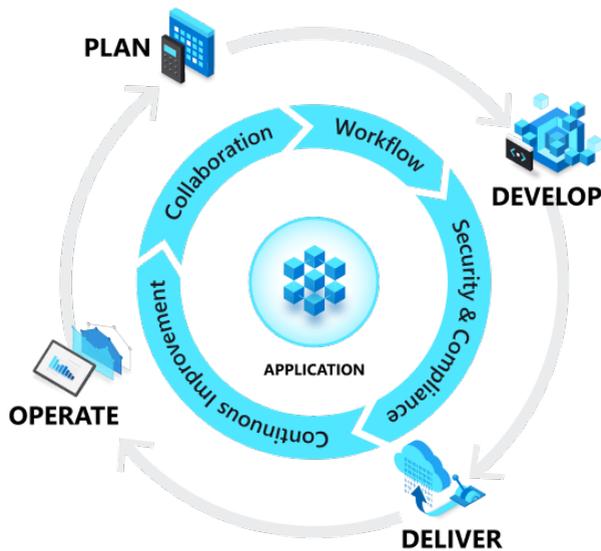


Figura 3: El ciclo de DevOps

Fuente: (Quijano, 2018)

A continuación, se describe cada una de las etapas descritas en la figura 3, información tomada de (Guijarro, Caparrós, & Cubero, 2019):

a) Plan

En la fase de planeamiento, los equipos de DevOps conciben, definen y describen las características y la funcionalidad de las aplicaciones y los sistemas que van a crear.

b) Desarrollar

La fase de desarrollo incluye todos los aspectos de la programación (escritura, pruebas, revisión e integración del código por parte de los miembros del equipo) y la compilación de ese código en artefactos de compilación que se pueden implementar en varios entornos.

c) Entregar

La entrega es el proceso de implementar aplicaciones en entornos de producción de un modo constante y confiable. La fase de entrega incluye también la implementación y la configuración de la infraestructura básica totalmente gobernada que constituye esos entornos.

d) Funcionamiento

La fase de uso implica mantener y supervisar las aplicaciones, así como solucionar los posibles problemas en los entornos de producción.

2.3.3. Beneficios de DevOps

1. Incremento de la velocidad de entregas y despliegues en un ambiente de producción.
2. Disminución de errores en el ciclo de desarrollo a través de la automatización.
3. Mejorar el control del ciclo de desarrollo de un producto.
4. Mejorar la confiabilidad del producto entregado.
5. Mejorar la comunicación de las diferentes áreas que están involucradas en la construcción y entrega del producto.
6. Escalabilidad de las aplicaciones y recursos.

2.3.4. ¿Cuál es la promesa de DevOps?

- Autoservicio: empieza a trabajar de manera rápida
- Integración continua: significa que después del ciclo, podemos seguir haciendo crecer de manera modular hasta el momento de entregar
- Colaboración, la más importante de DevOps, retroalimentación del equipo, desarrolladores, pruebas, el cliente.

2.3.5. Qué motiva a las empresas a emprender iniciativas DevOps

El crecimiento tecnológico y la nueva forma de implementar procesos de desarrollo que mejoren la eficiencia de los sistemas, ha dado paso para que DevOps ocupe un lugar distintivo en muchos de los proyectos a corto y medio plazo de muchas de las organizaciones TI a nivel mundial, en la figura 4, se muestra los resultados obtenidos del estudio realizado por (Vanson Bourne,

2016) a varias empresas de TI y varios negocios o sectores de España, entre ellos: servicios financieros, sanidad, industria, telecomunicaciones, distribución, medios de comunicación y entretenimiento en seis países europeos (Alemania, España, Francia, Italia, Reino Unido y Suiza) y otros siete países en todo el mundo (Estados Unidos, Canadá, Brasil, Australia, China, India y Japón) (Jiménez, 2016).

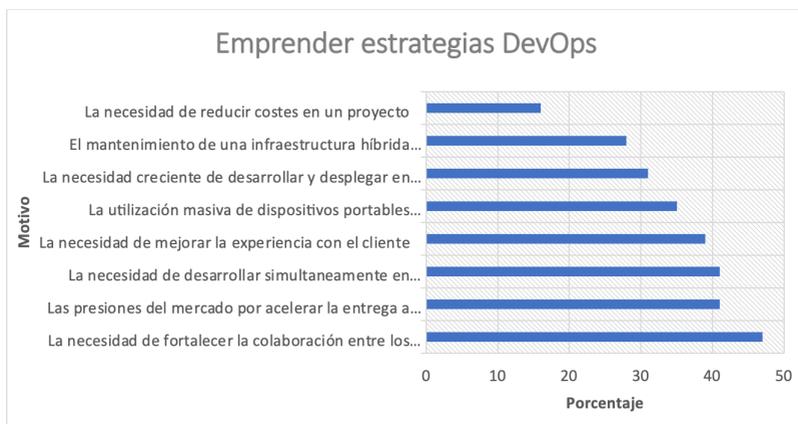


Figura 4: Razones para aplicar DevOps
Fuente: Encuesta de Vanson Bourne, citado por (Jiménez Marco, 2016, pág. 91)

También es interesante observar que la necesidad de reducir los costes no se encuentra entre las principales razones de las organizaciones al adoptar una estrategia DevOps, aunque a menudo llevar a cabo estas prácticas evita subir costes relacionados con la inversión en infraestructura de prueba y detección de defectos de software.

La información de la figura 4 muestra que las nuevas tendencias que desestabilizan las organizaciones TI, y estos motivos no son ajenos a la realidad del entorno de Ecuador y la región, la necesidad de mejorar las prácticas relacionadas con el desarrollo de aplicaciones, control de calidad y gestión de la entrega continua están en aumento.

2.3.6. Los cinco principales obstáculos de DevOps

Al aplicar metodologías, ya sean ágiles o tradicionales, siempre traen ventajas y desventajas, es por eso que en este apartado se realiza un análisis de los inconvenientes u obstáculos que se pueden presentar al momento de implementar DevOps en la ejecución de cualquier proyecto.

Los gerentes y directores de proyectos tecnológicos deben asignar parte del presupuesto a DevOps, existen algunas razones que bien podrían frenar su iniciativa, así lo refleja el estudio realizado por (Vanson Bourne, 2016) a varias empresas de TI, donde revela que hay serias dificultades en el inicio del despliegue DevOps, la mayoría de estas trabas recaen en las personas y la ejecución de los procesos más que en las tecnologías adoptadas conjuntamente con las herramientas. Esto es un problema claramente definido no sólo de los partidarios de DevOps, sino también en el mercado global que va dirigido a encontrar soluciones para el despliegue sencillo y a contribuir al éxito de una estrategia DevOps, información citada del trabajo de (Jiménez, 2016).

Tabla 1: Cinco principales obstáculos en un proyecto DevOps

Desventaja u Obstáculo	Porcentaje
Complejidad del proyecto: muchas personas y departamentos implicados con interdependencias	35%
Discordancia de roles y de responsabilidades entre desarrollo y operaciones	28%
Problemas de seguridad o de conformidad	25%
Mala comprensión de diversas fases del ciclo de desarrollo y de las personas responsables de cada etapa	24%
Falta de presupuesto o de claridad sobre la inversión que se quiere hacer	24%

Fuente: (Jiménez Marco, 2016, pág. 94)

2.3.7. ¿Cómo medir el éxito de una estrategia DevOps?

Los resultados para medir la aplicación de la cultura DevOps se pueden generar a partir de factores externos por ejemplo lo más útil es tomar como fuente la mejora de la experiencia con el cliente, también se pueden aplicar varias medidas con base en factores internos o externos, los factores internos pueden ser: Costes, factores de eficacia, ROI, entre otros; los externos se consideran: servicio ofrecido, entrega al cliente, experiencia con el cliente por medio de “Tickets de clientes”, es considerada el mejor y el peor indicador de los problemas de la aplicación son los tickets de soporte al cliente. En la figura 5 se muestran los resultados tomados de referencia del estudio realizado por Vanson Bourne en el año 2016 sobre los factores por los que los consumidores no tienen sus expectativas satisfechas de acuerdo a los requerimientos iniciales.



Figura 5: Como medir el éxito de aplicar DevOps
Fuente: (Jiménez, 2016)

2.3.8. Técnicas de aplicación de DevOps

DevOps desde su origen se ha entendido no únicamente como un conjunto de herramientas o procesos, sino como una cultura organizativa. Dentro de esta nueva dinámica de trabajo, se trata de poner énfasis en algunos aspectos del trabajo diario, mejorar en la documentación de procesos, buena intercomunicación y colaboración entre equipos, reparto de responsabilidades, aprender de errores pasados y estar abiertos a nuevas ideas. El conocimiento relacionado con técnicas organizativas y herramientas útiles que facilitan la adopción de estas prácticas está directamente correlacionado con metodologías de desarrollo

de software ágil, por ejemplo: Scrum, Kanban, entre otras.

a) Scrum

Define la forma en la que los miembros de un equipo deben colaborar para conseguir entre todos acelerar los proyectos de desarrollo y control de calidad. Las prácticas de Scrum incluyen flujos de trabajo principales y terminología específica (sprints, bloques de tiempo, scrum diario [reunión]), y roles designados (Scrum Master, propietario del producto [product owner]).

Para la implementación de SCRUM deben llevarse a cabo bajo los valores y principios ágiles que permita responder a los desafíos que el sistema requiere dado que es un entorno cambiante y el desarrollo es muy rápido (Pardo, Chilito, Viveros, & Pino, 2019), las principales características que se llevan a cabo con este tipo de metodología son:

- Personas e interacciones sobre procesos y herramientas,
- Entrega de software funcional sobre documentación comprensiva,
- Colaboración con el cliente sobre contratos negociados,
- Responder a los cambios sobre seguir un plan específico.

b) Kanban

Kanban se originó a partir de las eficiencias que se alcanzaron en la fábrica de Toyota. Kanban prescribe que el estado «en curso» (WIP, del inglés «work in progress») de un proyecto de software debe controlarse en un tablero Kanban.

c) Agile

Los anteriores métodos de desarrollo de software Agile siguen teniendo una gran influencia en las herramientas y las prácticas de DevOps. Muchos de estos métodos, incluidos Scrum y Kanban, han incorporado elementos de la programación ágil. Algunas de estas prácticas están asociadas a una mayor capacidad de respuesta a los continuos cambios en requisitos y necesidades, los requisitos de

documentación en forma de casos prácticos, la realización de reuniones diarias para ponerse al día y la incorporación de comunicación continua con los clientes para conocer sus opiniones.

La metodología que se aplique, la recomendación que brinda (Jiménez, 2016) sobre los aspectos clave para aplicar una estrategia DevOps deben ser:

- Nombrar a un director de proyecto DevOps a nivel ejecutivo
- Reuniones y checkpoints de los miembros de cada campo de DevOps.
- Definir una lista de las habilidades necesarias.
- Optimizar los procesos para incorporar las aportaciones en el desarrollo, ensayo / control de calidad y producción.
- Establecer un presupuesto para la contratación de recursos especializados y la tecnología.
- Identifique las aplicaciones perjudiciales (p. 96).

La organización para la aplicación de la cultura DevOps se puede realizar por medio de dos grandes ámbitos, el de desarrollo de software y el de operaciones. A su vez, estos dos ámbitos tienen sus propios departamentos, los cuales pueden variar en función de las necesidades de cada empresa. Algunos de ellos pueden aparecer fusionados o en ocasiones prescindir de ellos dado que el volumen de negocio no lo requiere. Un ejemplo de los distintos departamentos que puede haber en una empresa es que se muestra en la Figura 6:

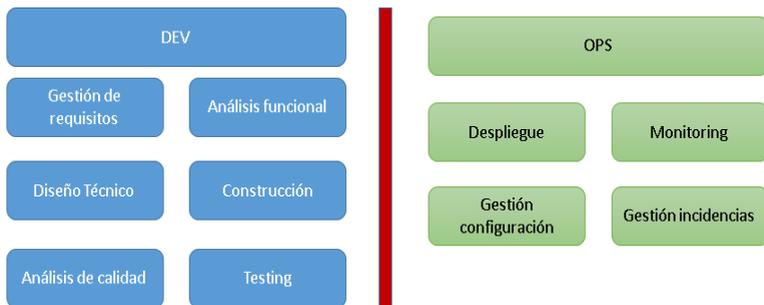


Figura 6: Áreas principales de los departamentos de Desarrollo y Operaciones

Fuente: (Mamani, Del Pino, & Gonzales, 2020) CASO DE ESTUDIO

a) Preliminares

Para la aplicación de caso de estudio se detalla la información inicial:

- Nombre de la empresa: IT-Empresarial
- Descripción de la empresa: Es una compañía proveedora de desarrollo ágil de software, que cuenta con varias líneas de negocio, entre las principales: Servicios de Desarrollo de Software, Aplicaciones Web, Soporte y Mantenimiento, Consultoría y capacitaciones, CLOUD, Procesos de Desarrollo de Software, Gestión de Reglas de Negocio.
- Personal
 - Desarrollo: 15 personas.
 - Operaciones: 5 personas.
 - Administrativo: 3 personas.

b) Proceso por automatizar

La automatización de inicio a fin se trabajó sobre uno de los productos de software propios de la empresa llamado “*checkone*”, en el cual el personal debe registrar las horas y actividades trabajadas durante el día para actividad asignada.

Para realizar el ejemplo de aplicación, se presenta algunas imágenes de varios Sprint que la empresa asigna a cada miembro de equipo.



Figura 5: Portada de la aplicación CheckOne
Fuente: (IT-Empresarial, 2021)

EQUIPO: ANTS - JANETH MARISOL ORTEGA POZO

Selecciona Estado del Sprint | Configurar Equipo | Configurar Actividades Estandar

LISTA DE SPRINTS

Estado	Título	Mes	Inicio	Fin	Horas	Avance Sprint %	Avance Equipo %
En Proceso	SPRINT 2 - FEBRERO 2021	FEBRERO	2021/02/12	2021/03/14	120	49.09%	48.46%
Finalizado	SPRINT 1 - FEBRERO 2021	FEBRERO	2021/01/29	2021/02/11	80	97.96%	96.13%
Finalizado	SPRINT 2 - ENERO 2021	ENERO	2021/01/15	2021/01/28	80	97.65%	96.83%
Finalizado	SPRINT 1 - ENERO 2021	ENERO	2021/01/14	2021/01/14	72	95.90%	100.00%
Finalizado	SPRINT 2 - DICIEMBRE	DICIEMBRE	2020/12/11	2020/12/31	120	92.25%	96.31%

Figura 6: Backlog
Fuente: (IT-Empresarial, 2021)

En la figura 6 se muestra la lista de sprints programados por cada mes, dentro de cada uno se asigna o programa actividades a los miembros del equipo del desarrollo, para el ejemplo se captura pantalla del sprint en proceso “sprint 2-febrero 2021”, ver figura 7.

ADMINISTRACIÓN DE ACTIVIDADES - SPRINT 2 - FEBRERO 2021 [120.00 HORAS]

Crear Nueva Actividad | Seguimiento | Volver | Recargar | Todos

AVANCE DE EQUIPO: ANTS

PROGRAMADOR	H. ASIGNADAS	H. PLANIFICADAS	H. REALES	H. NO PLANIFICADAS	H. EXTERNAS	AVANCE
JENNY PATRICIA MORALES MALDONADO	120h	120h	87h 30min	1h 30min	2h	54.67%
MARCURIZ DE LOURDES ACOSTA YERONI	120h	120h	88h 30min	0h	2h	48.56%
MERY ELIZABETH MESAANDRANGO	120h	120h	74h	0h	0h	41.16%
RESULTADOS	360h	360h	250h	1h 30min	5h	48.46%

ACTIVIDADES NO PLANIFICADAS Y EXTERNAS

ACTIVIDADES PLANIFICADAS

OPCIONES	Programador	OBJ.	E	Tipo de Entreg.	Aplicación	Título	H.A.	H.P.	H.R.	Avanc.	Estado
	JENNY PATRICIA...	S	0	Informe lider de...	PlanningPOG	Reunión: HJ Información Usuarios FROGM V1.0	2h	2h	1h 30min	60%	Aprobada
	JENNY PATRICIA...	S	0	Especificación f...	FarmaPos	Correcciones Esp.: Normativa ARCSA_Cambios Dev. Programadas	5h	5h	2h 30min	36%	Aprobada
	JENNY PATRICIA...	No	0	General	General	Feriado de Carnaval	16h	16h	16h	100%	Aprobada
	JENNY PATRICIA...	S	0	En producción	Sistema 1930	Pruebas Funcionales Cupones Sistema 1930	29h	29h	46h	94%	Aprobada
	JENNY PATRICIA...	S	0	Especificación f...	Minimarket	Especificación Cambios a Sistema Minimarket	25h	25h	10h	39%	Aprobada
	JENNY PATRICIA...	No	0	General	Minimarket	Seguimiento puesta en producción Sistema Minimarket FarmaPos	2h	2h	3h	40%	Aprobada
	JENNY PATRICIA...	No	0	General	No aplica	Actividades Generales	8h	8h	7h 30min	95%	Aprobada
	JENNY PATRICIA...	No	0	General	FarmaPos	Revisión Funcionalidad Versión Cupones	1h	1h	1h	100%	Aprobada
	JENNY PATRICIA...	S	0	Informe lider de...	FarmaPos	Especificación Generación de Guía de Remisión para devolución de P...	31h	31h	0h	0%	Aprobada
	MARCURIZ DE L...	No	0	General	No aplica	Actividades generales	8h	8h	7h	85%	Aprobada
	MARCURIZ DE L...	No	0	General	No aplica	Actualización de proyectos Analista QA y seguimiento	1h	1h	0h	0%	Aprobada
	MARCURIZ DE L...	No	0	General	General	Feriado carnaval	16h	16h	16h	100%	Aprobada

Página 1 de 1 | Mostrando 1 - 27 de 27

ACTIVIDADES SUSPENDIDAS

Figura 7: Actividades asignadas al equipo
Fuente: (IT-Empresarial, 2021)

En la figura 7 se muestra el cronograma de las actividades asignadas a los miembros del equipo, la asignación se la realiza por horas, cada actividad máximo 8 horas, conforme van cumpliendo las actividades va incrementando el avance del porcentaje en cada sprint.

Finalmente, en la figura 8 se muestra la funcionalidad de control que le brinda al Scrum Master o líder de equipo, la herramienta presenta la opción de mirar el gráfico estadístico para realizar un mejor seguimiento al cumplimiento de actividades por todos los miembros del equipo.

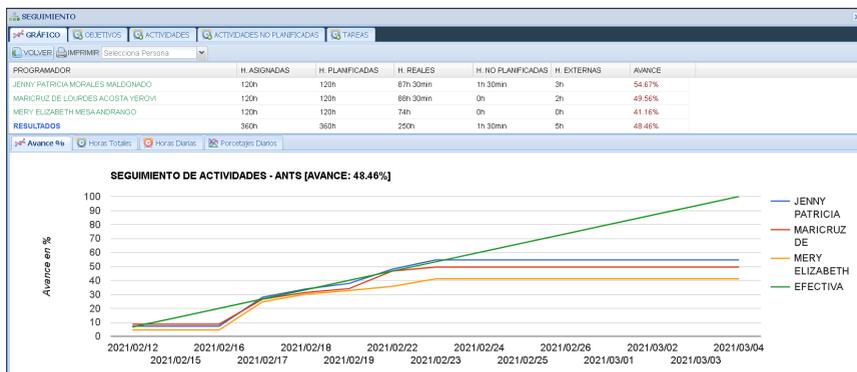


Figura 8: Seguimiento al cumplimiento de las actividades

Fuente: (IT-Empresarial, 2021)

El estado actual DevOps en los procesos ha permitido que el personal de desarrollo y de operaciones brinden oportunidad de estar abiertos al cambio, a aprender nuevas tecnologías, a iniciar cambios culturales y lo más importante de todo, conscientes de la necesidad de empezar a adoptar DevOps como parte de la transformación tecnológica del mercado y de la empresa.

Los procesos más importantes que se han adaptado DevOps es el análisis de requerimientos y pruebas, ya que conjuntamente se trabaja en equipo el personal de desarrollo, test con operaciones del negocio, de esta manera han generado excelentes y viables

planes de pruebas de código continuo que brinden una respuesta rápida y continua sobre el nivel de riesgo empresarial, se han adoptado las herramientas: JUnit + RobotFramework.

c) Conclusión del estudio

- Durante el desarrollo del estudio, surge la necesidad de comprender el rol que realiza el equipo de Quality Assurance (QA), ya que los departamentos de calidad también han sufrido una transformación, expertos afirman que poco a poco no van a ser necesarios. Tradicionalmente la calidad siempre se encargaba de encontrar los errores de un producto ya en funcionamiento, mientras que, integrando DevOps en el proceso de desarrollo, el objetivo no es encontrarlos, sino prevenirlos, sacando hacia delante nuevas releases si todo está funcionando o haciendo retroceder el proyecto si se encuentran fallos.
- La validación del método de adopción de DevOps fue realizada con la ayuda del equipo de TI de la empresa, la cual demostró que con tan solo una pequeña implementación técnica del proceso de DevOps los resultados fueron notorios y se evidenciaron principalmente en el ahorro en tiempo invertido en hacer tareas repetitivas de forma manual.
- Como trabajo futuro se plantea la aplicación de los procesos restantes, de igual manera la implementación de una cadena de herramientas más robustas y maduras, en la industria existen algunas herramientas como Docker, Puppet, Jenkins o AWS Lambda, están asociadas a DevOps que es difícil no considerar una u otra para acelerar los procesos de desarrollo, finalmente, la generación sistémica de documentación para asegurar que todos los componentes de la propuesta estén continuamente actualizados y sean consistentes.

Bibliografía

- Amorim, H., & Reis Grazia, M. (07 de marzo de 2020). Trabajo y Metodologías ágiles. *Controversias y Concurrencias Latinoamericanas*, 11(20), 209-221. Recuperado el 10 de enero de 2022, de <https://www.redalyc.org/journal/5886/588663787012/html/>
- Corona, B. &.-M. (2016). Estado de arte sobre métodos de evaluación de metodologías ágiles en las pymes. *ReCIBE. Revista electrónica de Computación, Informática, Biomédica y Electrónica*, 5(1). Recuperado el 17 de febrero de 2021, de <https://www.redalyc.org/articulo.oa?id=5122/512252666004>
- Dittrich, Y., Nørbjerg, J., Tell, P., & Bendix, L. (2018). Researching cooperation and communication in continuous software engineering. *International Workshop on Cooperative and Human Aspects of Software Engineering*, 87-90. doi:10.1145/3195836.3195856
- Garzas, J. (7 de febrero de 2011). *JavierGarzas*. Recuperado el 12 de febrero de 2021, de <https://www.javiargarzas.com/2011/02/diferencias-software-fabricacion-tradicional-1.html#:~:text=Construir%20software%20no%20es%20igual,proceso%20de%20construcci%C3%B3n%20sea%20diferente.>
- Guijarro Olivares, J., Caparrós Ramírez, J., & Cubero Luque, L. (2019). *DevOps y seguridad cloud*. Editorial UOC. Recuperado el 22 de julio de 2021
- Gutiérrez Martínez , M. T., Pérez de Celis Herrero, C., & Cossío Aguilar, G. (2011). Aplicación de las metodologías ágiles en el proceso de producción de piezas de arte de nuevos medios: Bio-lencia como caso de estudio. *JISTEM Revista de Gestão da Tecnologia e Sistemas de Informação*, 407-424.
- IT-Empresarial. (2021). *It-Empresarial*.
- Jiménez Marco, G. (2016). *DevOps, la nueva tendencia en el desarrollo de sistemas TI, un caso práctico en el análisis de incidencias de software*. Barcelona.

- Mamani Rodríguez, Z., Del Pino Rodríguez, L., & Gonzales Suárez, J. C. (2020). Arquitectura basada en Microservicios y DevOps para una ingeniería de software continua. *Industrial Data*, 141-149. Recuperado el 16 de julio de 2021, de <https://www.redalyc.org/articulo.oa?id=81665362014>
- Mamani Rodríguez, Z., Del Pino Rodríguez, L., & Gonzales Suárez, J. C. (julio de 2020). Arquitectura basada en Microservicios y DevOps para una ingeniería de software continua. *Industrial Data*, 23(2). doi:<https://doi.org/10.15381/idata.v23i2.17278>
- MicroFocus. (15 de noviembre de 2015). *Ejemplos prácticos de DevOps para empresas de mainframe*. Obtenido de https://www.microfocus.com/es-es/media/white-paper/real_world_devops_for_mainframe_enterprises_wp_es.pdf
- Mitre Hernández, H., Ortega, M. E., & Lemus, O. C. (2014). Estimación y control de costos en métodos ágiles para desarrollo de software: un caso de estudio. *Ingeniería. Investigación y Tecnología*, 403-418. Recuperado el 15 de julio de 2021, de <https://www.redalyc.org/articulo.oa?id=40431741007>
- Pardo Calvache, C., Chilito-Gómez, P., Viveros Meneses, D., & Pino, F. (2019). Scrum+: A scaled Scrum for the agile global software development project management with multiple models. *Revista Facultad de Ingeniería Universidad de Antioquia*(93), 105-116. doi:10.17533//udea.redin.20190519
- Quijano, J. (21 de junio de 2018). *Genbeta Dev*. Recuperado el 25 de julio de 2021, de <https://www.genbeta.com/desarrollo/el-ciclo-de-devops-una-guia-para-iniciarse-en-las-fases-que-lo-componen>
- Xeridia. (30 de enero de 2017). *Xeridia*. Recuperado el 25 de julio de 2021, de <https://www.xeridia.com/blog/sabes-realmente-que-es-devops>



Integración de sistemas utilizando Servicio Web SOAP

Santiago Damián Quishpe Morales

squishpe@pucesi.edu.ec

Pontificia Universidad Católica del Ecuador Sede Ibarra

Escuela de Ingeniería

Ecuador

1. Introducción

La globalización ha obligado a las organizaciones a ser más competitivas, dada la necesidad de posicionarse en ese mercado globalizado, este fenómeno ha hecho que necesiten revisar y actualizar sus estrategias y procesos de negocio de forma continua; estrategias que se apoyan en el conjunto de datos e información para el manejo de proceso.

En este capítulo el capítulo contempla algunos conceptos de Servicios Web, SOAP, que es un componente de software que se comunica con otras aplicaciones codificando los mensajes en XML y enviando estos mensajes a través de protocolos estándares de Internet que ayudan a mejorar la flexibilidad de los procesos empresariales mediante la integración con aplicaciones.

Los Servicios Web reflejan el enfoque de Arquitectura Orientada a Servicios (SOA) para la programación. Los Servicios Web ofrecen interoperatividad. Los Servicios Web se basan en tecnologías de transporte existentes, como por ejemplo HTTP y técnicas de codificación de datos estándar como XML (Extensible Markup Language) para invocar la implementación.

Los componentes clave de los Servicios son el Lenguaje de descripción de Servicios Web (WSDL), es el archivo basado en XML que describe el Servicio Web. La solicitud de Servicios Web utiliza este archivo para enlazar con el Servicio.

2. Servicio Web SOAP con SQL Server

2.1. Servicio Web

Un Servicio Web o Web Service es un método de comunicación que maneja una colección de protocolos abiertos que se utilizan para intercambiar los datos entre los sistemas. Para el manejo de los Servicios Web con los diferentes lenguajes de programación que ayudan a intercambiar información a través de una red como se muestra en la (Figura 1).

Estructura de un sitio web:

- Servicio de directorio
- Descripción e Integración o UDDI (Universal Description, Discovery and Integration)
- Descripción de Servicios Web o WSDL (Web Services Description Language)
- Descripción de los Servicios Web
- Formatos de transmisión



Figura 5.1 Infraestructura de los servicios Web
[www.msdn/microsoft.com]

Figura 1: Estructura de Servicio Web

Fuente: (Gallegos, 2013)

La infraestructura del Servicio Web que consta de los siguientes elementos:

- Servicio de directorio. Proporcionan una localización centralizada de un servicio Web.

- Servicio de localización. Localizan uno o más documentos que describen un servicio.
- Descripción de los Servicios Web. Describe las interacciones soportadas por un Servicio Web.
- Formatos de transmisión. Para permitir una comunicación lo más universal posible.

Diagrama del proceso de recreación de Servicio:

Para el proceso de recreación de un Servicio Web se lo maneja

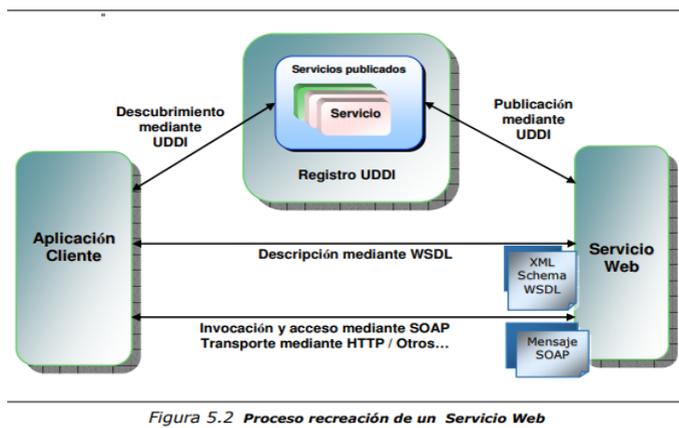


Figura 5.2 Proceso recreación de un Servicio Web

Figura 2: Proceso de recreación de Servicio Web

Fuente: (Gallegos, 2013)

La pila de protocolos de Servicios Web está formado por un directorio para la búsqueda y ubicación del Servicio Web; un mecanismo de descubrimiento para localizar Servicios Web XML, una descripción de servicio para definir cómo se deben utilizar esos servicios y formatos de conexión estándar para la comunicación (Figura 2).

2.2. Tecnología Servicios Web

Los Servicios Web están construidos con varias tecnologías que trabajan conjuntamente con los estándares que están

emergiendo para asegurar la seguridad y operatividad, de modo de hacer realidad que el uso combinado de varios Servicios Web, independiente de la o las empresas que los proveen, esté garantizado (Saffirio C., 2016).

2.2.1. XML

Abreviación de Extensible Markup Language. El XML es una especificación desarrollada por W3C. Permite a los desarrolladores crear sus propios tags, que les permiten habilitar definiciones, transmisiones, validaciones e interpretación de los datos entre aplicaciones y entre organizaciones (Saffirio C., 2016).

2.2.2. SOAP – Servicios Web

Es un sistema software que permite manejar la integración e interoperabilidad entre máquinas con una conexión de red. Para el manejo de una interfaz descrita en un lenguaje que puede ser procesado por una máquina, específicamente el Lenguaje de Descripción de Servicios Web WSDL (de sus siglas del inglés *Web Services Description Language*). También permite interactuar con los Servicios Web a través de mensajes SOAP, que son mensajes especiales que se manejan en diferentes equipos a través de una red.

Con el desarrollo de aplicaciones integradas con los diferentes Lenguajes y Arquitecturas Orientadas a Servicios, Servicios Web, Integración de Aplicaciones empresariales (EAI), que permiten una mejor integración de los procesos de negocios de la organización. Con el manejo del Bus de Servicios de Empresa (BSE) extrae las mejores características de estas tecnologías (Chapell, 2004).

Los BSE son enfoques de integración que permite un bajo acoplamiento entre sus aplicaciones y una red de integración altamente distribuida. Es una plataforma de integración basada en estándares que combina mensajería, Servicios Web, transformación de datos y enrutamiento inteligente para

conexiones confiables y coordina la interacción de un gran número de aplicaciones diferentes de una empresa extendida, constituida por la organización y los interesados en el negocio. La integración de Servicios Corporativos trata de resolver el problema que surge cuando los Servicios Web dentro de la organización se multiplican Ildapena (2009):

- Identificar los mensajes y las rutas entre los Servicios
- Permitir el flujo de mensajes a través de diferentes protocolos de transporte (HTTP, FTP, SMTP)
- Transformar los formatos de los mensajes entre el solicitante y el Servicio
- Proporcionar robustez y seguridad de las comunicaciones
- Proporcionar enrutamiento inteligente y ubicación independiente de la transformación

Con la Arquitectura Orientada a Servicios (AOS) conocida como SOA que maneja la arquitectura de integración y con el manejo de aplicaciones con sus componentes que permiten tratar con Servicios; esta arquitectura permite que una colección de sistemas distribuidos y aplicaciones complejas.

Los trabajos en el área de desarrollo de aplicaciones basadas en Servicios son amplios, por citar algunos trabajos se tiene (Calvo, Gracia, Bayo, 2014) presentan la arquitectura de una aplicación web para el análisis y diseño de estructuras utilizando Servicios Web en el diseño de sus 4 componentes de su sistema. Pernalet, López, Montaña y Miguel (2007) presentan la arquitectura de un sistema de enseñanza utilizando un EBS para la integración de sus componentes, Huanzhuo, Shuai (2008) exponen presentan un modelo conceptual basado en SOA y BSE para auditoría continuas. Por otro lado Echeverría, Astudillo y Estrada (2008) han investigado en los modelos de calidad (ESB-QM) con el objetivo de proveer un lenguaje que haga posible tratar a los EBS como productos estándares.

Propiedades de SOAP:

- Es un protocolo ligero, sencillo de implementar
- Se usa para la comunicación entre aplicaciones.
- No está asociado con ningún lenguaje de programación.
- Está diseñado para comunicarse vía HTTP

2.2.3. SQL Server

SQL Server es un sistema de gestión de bases de datos relacionales de Microsoft que permite el manejo de los datos a nivel empresarial, maneja T-SQL (Transact -SQL), que maneja extensiones de programación de Sybase y Microsoft, esta base de datos maneja control de transacciones y con variables declaradas.

2.2.4. ASP.NET

ASP.NET es un modelo de desarrollo Web creado por Microsoft que permite el desarrollo de sitios y aplicaciones web, ASP.NET forma parte de .NET Framework que contiene las librerías necesarias para la codificación, con este modelo se puede utilizar en distintos lenguajes de programación como por ejemplo lenguaje C# (C Sharp).

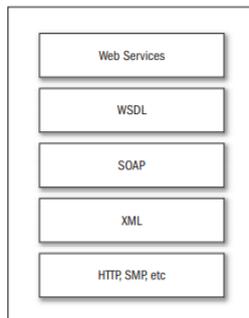


Figura 3: Protocolo y Servicio Web

Fuente: (Quishpe, 2021)

2.2.5. Servicio Web SOAP con SQL Server y ASP

Un Servicio Web es un componente software accesible a través de protocolos estándares de Internet, se maneja con el núcleo

de la plataforma .NET y de la visión del software como Servicio (en contraposición a la visión tradicional del software como producto).

Las aplicaciones web, en las que los usuarios interactúan con las máquinas a través de navegadores web, se pretende que el uso de XML y SOAP que facilite el desarrollo con el manejo de la integración y la interoperabilidad con los sistemas distribuidos. Los Servicios Web (XML y SOAP) están diseñados para facilitar la comunicación entre distintos sistemas y aplicaciones sin la intervención de operadores humanos.

- Seguridad en Servicios Web SOAP
- Creando nuestro Web Service SOAP
- Los Modelos de Datos
- Construyendo el Servicio Web
- Accediendo al Servicio Eb

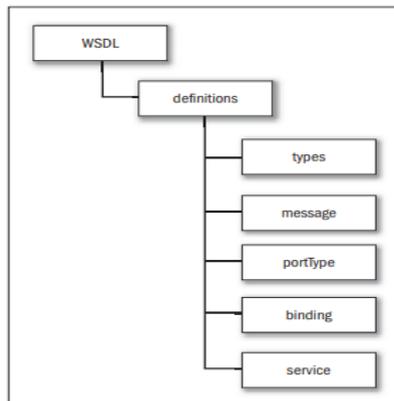


Figura 4: Componentes de WSDL
Fuente: (Quishpe, 2021)

2.2.6. SOAP

Es el protocolo de comunicación para los Servicios Web, en el que maneja la integración de datos para ser procesada:

- Envoltorio SOAP
- Encabezado SOAP
- Cuerpo SOAP

Los mensajes SOAP son XML, un mensaje SOAP puede contener datos jerárquicos (Figura 5).

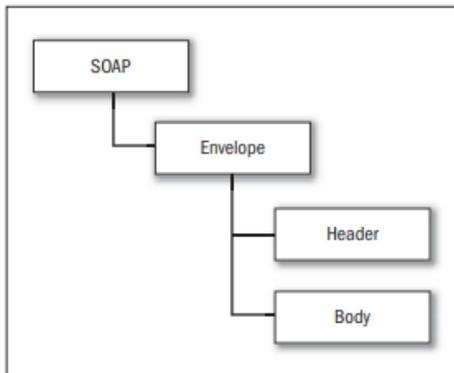


Figura 5: Elementos de SOAP

Fuente: (Quishpe, 2021)

3. Aplicación de los Servicios Web

3.1. SQL Server

- Acceso a SQL Server (Figura 6)

Se ingresa al servidor de SQL con las credenciales de inicio de sesión.

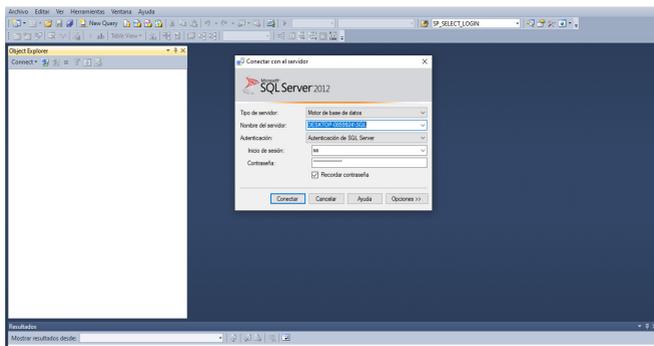


Figura 6: Pantalla de SQL SERVER

Fuente: (Quishpe, 2021)

- Acceso a SQL Server
- Se ingresa al servidor de SQL para crear la base de datos
- Creación de la base datos llamada clientes (Figura 7)

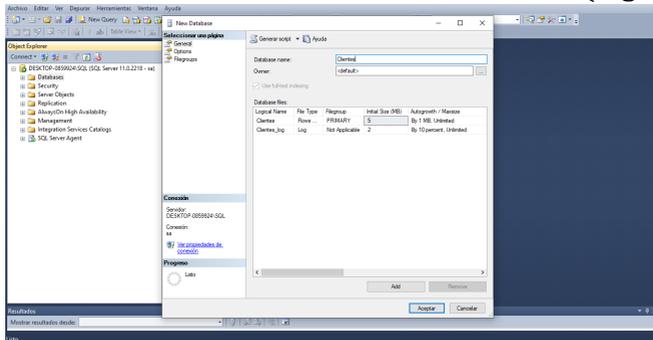


Figura 7: Creación de base de datos

Fuente: (Quishpe, 2021)

- Creación la tabla clientes, con el nombre clientes (Figura 9)

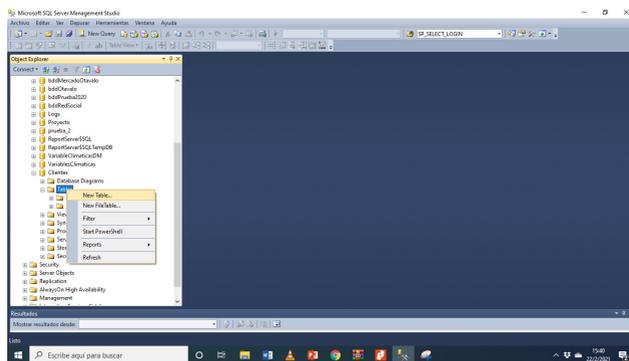


Figura 9: Creación de una tabla

Fuente: (Quishpe, 2021)

- Creación de las propiedades de la tabla clientes, que tiene los siguientes atributos y los tipos de datos (Figura 10).

cli_id int,
 cli_apellidos varchar (50),
 cli_nombres varchar (50),
 cli_direccion varchar (50),
 cli_telefono varchar (20),
 cli_estado, int IDENTITY(1,1)

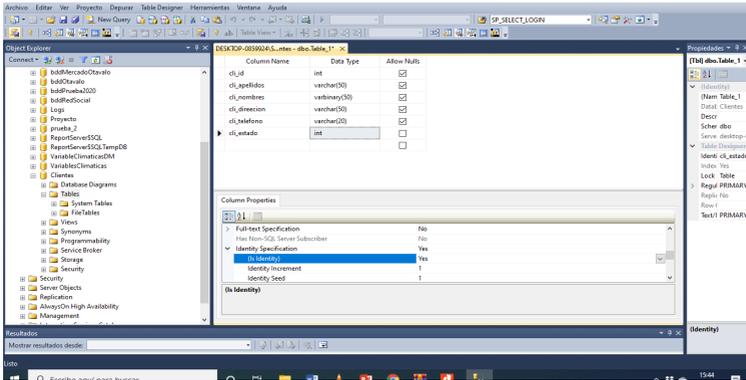
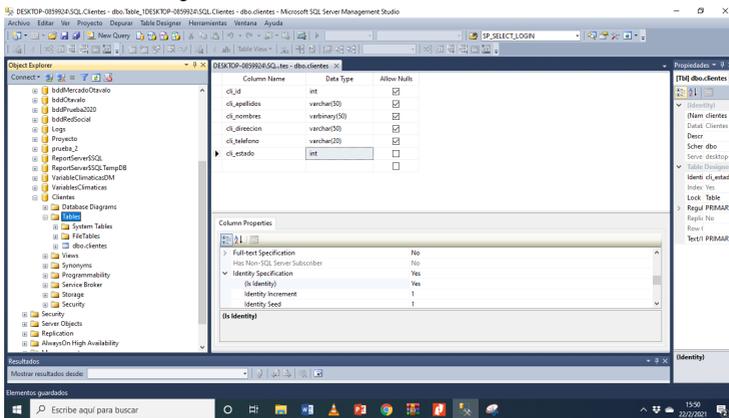


Figura 10: Creación de una tabla
Fuente: (Quishpe, 2021)

- **Tabla**
Tabla de SQL Server



3.2. Creación de Servicio Web

Para la creación del Servicio Web, se debe ingresar con un nuevo proyecto de Visual Studio de tipo Servicio Web, con el que aparece la siguiente pantalla:

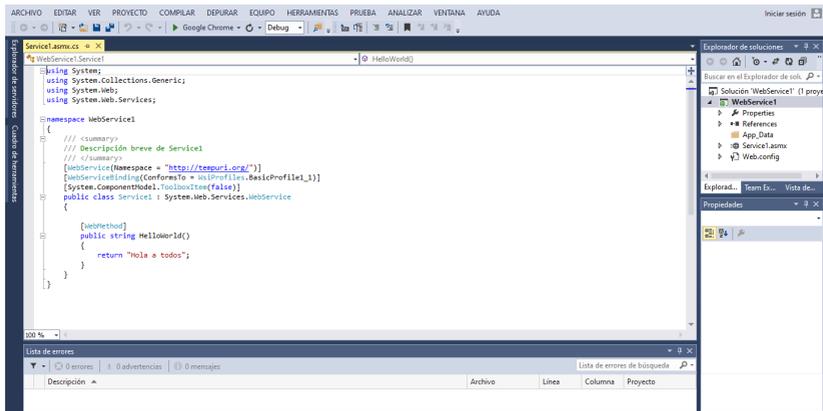


Figura 12: Creación del Servicio

Fuente: (Quishpe, 2021)

Luego se debe crear el método para la construcción del Servicio, en el que tiene la cadena de conexión con la base de datos SQL Server, con el siguiente detalle:

- La cadena de conexión
- La cadena de conexión debe ser mediante la autenticación de SQL Server, para indicar el usuario y la contraseña.
- Datos del servidor (ver Tabla 1)
-

Tabla 1: Datos del Servidor

Valor	Descripción
Local	Servidor que quiere acceder, es preferible usar (local).
Nombre de equipo	El nombre del equipo en el que se está usando la aplicación.
Nombre DNS	Un nombre DNS, es decir, un nombre que después “resolverá” una dirección IP.
Dirección IP	Una dirección IP en la que está el servidor de SQL
Dirección IP, Puerto	La dirección IP del servidor de IP al que se accederá por el puerto indicado.
Servidor SQL / Instancia	Instancia de servidor de SQL Server.
Base de Datos	El nombre de la base de datos de SQL.
Usuario	El nombre del usuario de la base de Datos.
Contraseña	La contraseña del usuario.

Nota: elaboración propia.

3.3. Cadena de conexión

Para conectar a una base de datos de SQL Server se utiliza la siguiente cadena de conexión.

- Data data source = ServidorSQL; initial catalog = BaseDatos; user id = Usuario; password = Contraseña
- Conectar con ADO.NET, mediante los objetos Command para acceder a un procedimiento almacenado, SqlDataAdapter, SqlCommand.
- Creación de una tabla

```
CREATE TABLE [dbo].[clientes](
    [cli_id] [int] NULL,
    [cli_apellidos] [varchar](50) NULL,
    [cli_nombres] [varbinary](50) NULL,
    [cli_direccion] [varchar](50) NULL,
    [cli_telefono] [varchar](20) NULL,
    [cli_estado] [int] IDENTITY(1,1) NOT NULL
) ON [PRIMARY]
GO
```

- Procedimientos almacenados

```
USE [Clientes]
GO
/***** Object: StoredProcedure [dbo].[sp_
seleccionar_clientes] **/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[sp_seleccionar_clientes]
as
begin
    Select *from clientes;
end
```

- Función de Servicio Web

Se crea el método del Servicio Web (Figura 13)

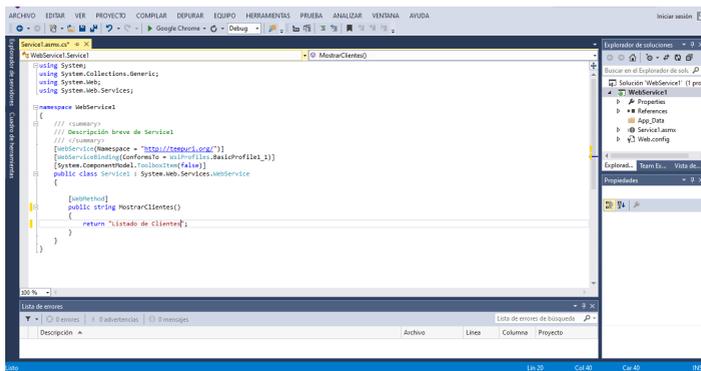


Figura 13: Método de servicio Web
Fuente: (Quishpe, 2021)

- Ejecutar el Servicio Web
- Se ejecuta el método con las diferentes funciones creadas
- Invocar el método del Servicio Web
- Se invoca el método creado (Figura 14)

Service1

Haga clic [aquí](#) para obtener una lista completa de operaciones.

MostrarClientes

Prueba

Haga clic en el botón 'Invocar', para probar la operación utilizando el protocolo HTTP POST.

SOAP 1.1

A continuación se muestra un ejemplo de solicitud y respuesta para SOAP 1.1. Es necesario reemplazar los marcadores de posición que aparecen con valores reales.

```

POST /Service1.svc HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: http://tempuri.org/IServicio/ShowClientes

<?xml version='1.0' encoding='utf-8'?>
<soap:Envelope xmlns:soap="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ShowClientes xmlns="http://tempuri.org/" />
  </soap:Body>
</soap:Envelope>

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version='1.0' encoding='utf-8'?>
<soap:Envelope xmlns:soap="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ShowClientesResponse xmlns="http://tempuri.org/">
      <ShowClientesResponseResult>
        <ShowClientesResponseResult>
          <ShowClientesResponseResult>
            <ShowClientesResponseResult>
              <ShowClientesResponseResult>
                <ShowClientesResponseResult>
                  <ShowClientesResponseResult>
                    <ShowClientesResponseResult>
                      <ShowClientesResponseResult>
                        <ShowClientesResponseResult>
                          <ShowClientesResponseResult>
                            <ShowClientesResponseResult>
                              <ShowClientesResponseResult>
                                <ShowClientesResponseResult>
                                  <ShowClientesResponseResult>
                                </ShowClientesResponseResult>
                              </ShowClientesResponseResult>
                            </ShowClientesResponseResult>
                          </ShowClientesResponseResult>
                        </ShowClientesResponseResult>
                      </ShowClientesResponseResult>
                    </ShowClientesResponseResult>
                  </ShowClientesResponseResult>
                </ShowClientesResponseResult>
              </ShowClientesResponseResult>
            </ShowClientesResponseResult>
          </ShowClientesResponseResult>
        </ShowClientesResponseResult>
      </ShowClientesResponseResult>
    </ShowClientesResponse>
  </soap:Body>
</soap:Envelope>

```

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```

<string xmlns="http://tempuri.org/">Listado de Clientes</string>

```

Figura 14: Servicio Web
Fuente: (Quishpe, 2021)



3.4. Creación de proyecto

- Se crea el proyecto de sitio web (Figura 15)

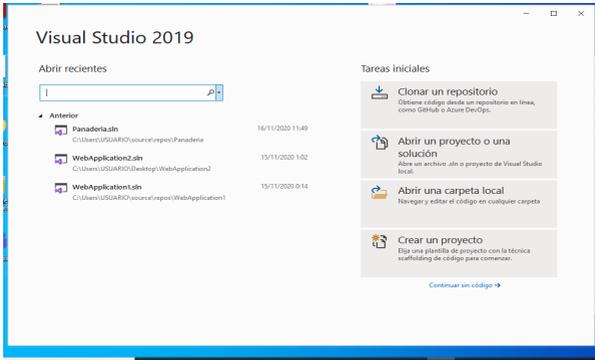


Figura 15: Creación de proyecto
Fuente: (Quishpe, 2021)

- Se escoge el lenguaje de programación y se crea el proyecto
- Se agrega un nuevo formulario
- Se abre el proyecto en la sección de código

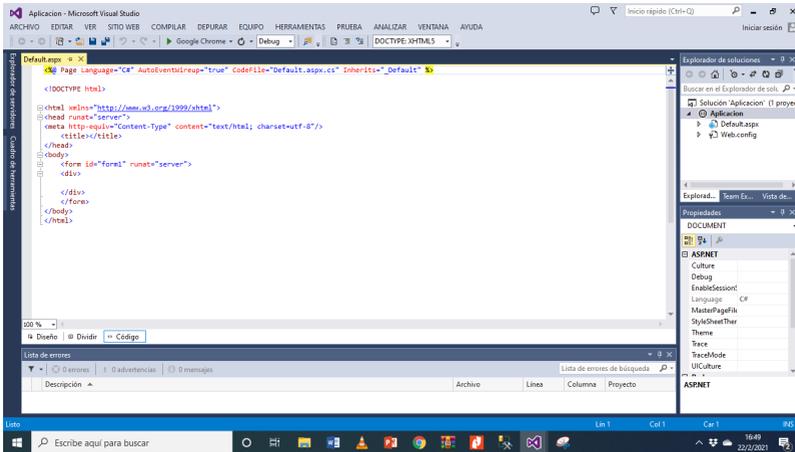


Figura 16: Código HTML
Fuente: (Quishpe, 2021)

- Se agrega la referencia
- Se enlaza la referencia del Servicio Web

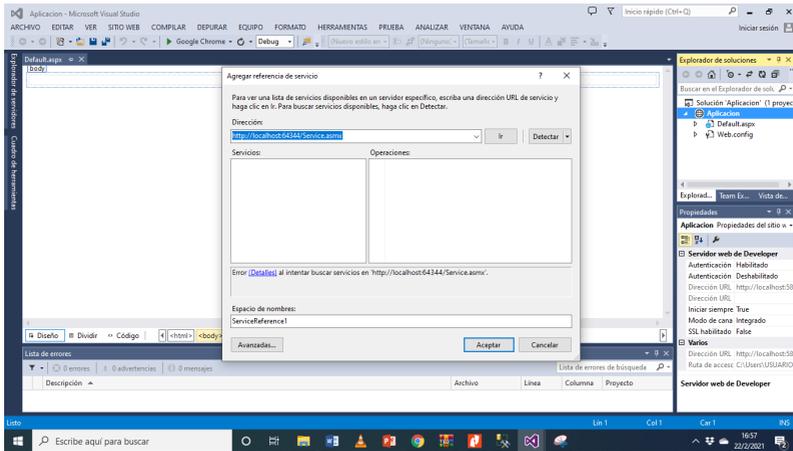


Figura 17: Enlace de referencia Web

Fuente: (Quishpe, 2021)

- Se conecta la aplicación con el Servicio Web

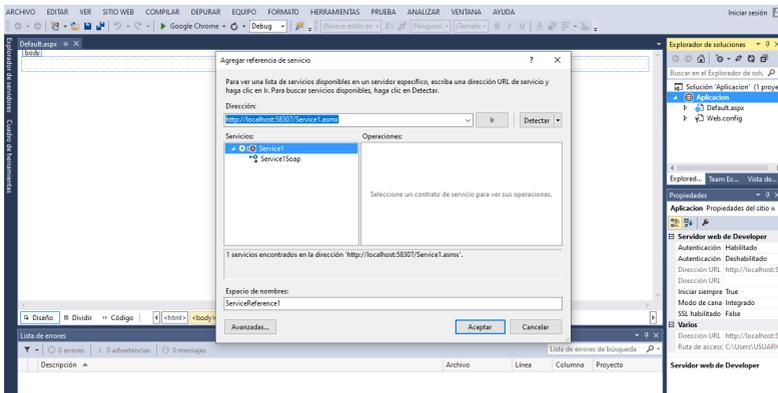


Figura 18: Conexión de Servicio Web

Fuente: (Quishpe, 2021)

- Se conecta el Servicio Web con la aplicación, en el que se describe la secuencia que se produce al llamar Servicio Web XML (Figura 19).

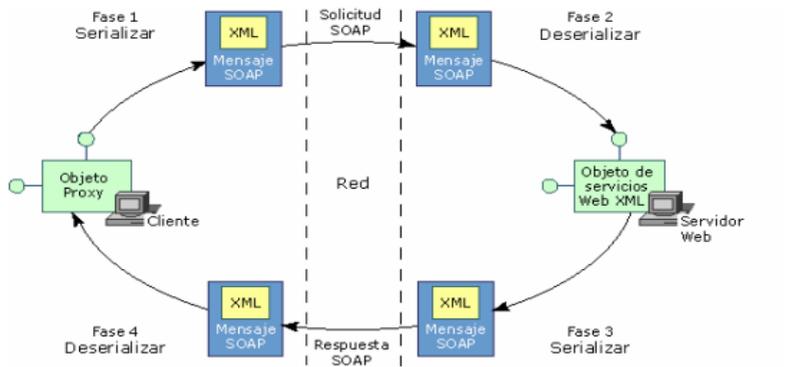


Figura 5.21 Proceso de comunicación entre un cliente y un servicio Web XML.

Figura 19: Proceso de comunicación entre cliente y servicio web

Fuente: (Gallegos, 2013)

- Se crea una instancia de una clase de proxy de Servicio Web XML, en la que el cliente invoca un método en la clase, con el que se serializan los argumentos del método del Servicio Web XML en un mensaje SOAP (Figura 19).

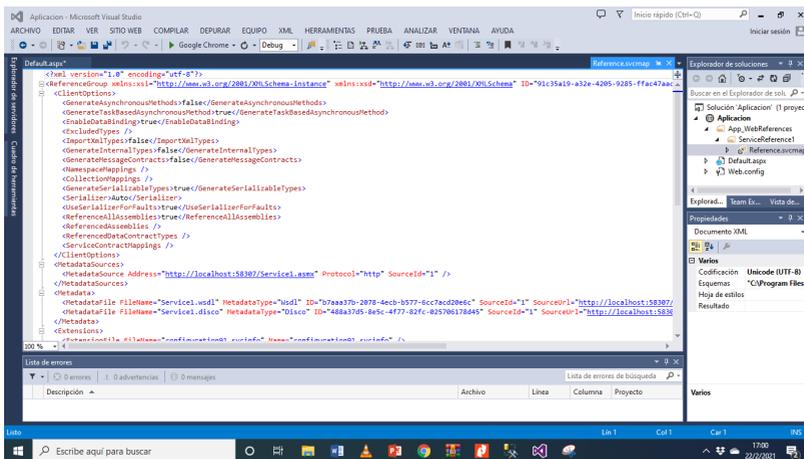


Figura 20: Resultado de Conexión

Fuente: (Quishpe, 2021)

4. Conclusiones

- Con el diseño de soluciones tecnológicas utilizando .NET mediante el desarrollo de páginas web con Servicios Web que ayudan a la integración de aplicaciones, utilizando una arquitectura centralizada para la administración de datos, con el manejo de SOAP para el transporte de HTTP y XML para resolver la escalabilidad de conexiones entre aplicaciones.
- UDDI (Universal Description, Discovery and Integration) gestiona el manejo de servicios Web, que realiza un registro distribuido de empresas y sus descripciones de servicio implementado en un formato XML.

Bibliografía

- Calvo, J.; Gracia, J; Bayo, E. (2014). Aplicación web para el análisis y diseño de estructuras. Informes de la Construcción.
- Chappell, David (2004). Enterprise Service Bus: Theory in Practice. O'Reilly Media Inc. Unite State of America.
- Enríquez, E. (2007). SOA (Arquitectura Orientada a Servicios). SOA (Arquitectura Orientada a Servicios). Ibarra, Ecuador: Obtenido de <http://repositorio.utn.edu.ec/bitstream/123456789/587/1/CAPITULOS.pdf>.
- Gallegos, M. C. (2013). Estudio de la tecnología .Net para el desarrollo de aplicaciones e implementación de servicios Web XML. Ibarra: UTN.
- Huanzhuo Y; Shuai C.; Fang, G.; Yuning, H. (2008). SOA-based conceptual model for continuous auditing: A discussion. Conference. on APPLIED COMPUTER & APPLIED COMPUTATIONAL SCIENCE (ACACOS '08), 400-405.
- Ildapena. (2009). <http://www.xperimentos.com>. Obtenido de <http://www.xperimentos.com>: <http://www.xperimentos.com/2009/02/03/bus-de-integracion-de-servicios-corporativos-enterprise-service-bus/>
- Saffirio, M. (2006). Tecnologías de Información y Gestión de

Procesos de Negocios (BPM). Obtenido de Tecnologías de Información y Gestión de

W3C Consortium. Servicios Web Architecture. Obtenido de <http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/#whatis>.

Procedimiento para la instalación y configuración de canales satelitales libres sobre Linux

César Napoleón Grijalva Maigua
cgrijalva@pucesi.edu.ec

Pontificia Universidad Católica del Ecuador Sede Ibarra -
Escuela de Ingeniería
Ecuador

INTRODUCCIÓN

Actualmente el servicio de la televisión por cable y la tradicional señal a través de antenas terrestres está perdiendo poco a poco popularidad debido al apogeo del Internet y las nuevas tecnologías. Sitios webs como www.youtube.com, www.netflix.com, www.vimeo.com entre otros servicios de streaming de video y el uso de dispositivos tecnológicos como Google Chromecast, Anycast, Tv Box etc., cada vez ganan mayor uso por parte de los usuarios que simplemente deben pagar por el servicio de Internet.

No obstante, estamos en un punto de transición con las tecnologías es decir se sigue utilizando el servicio de televisión por cable que por lo general es de pago, también antenas terrestres en la cual la señal no es muy buena, el uso de dispositivos como Tv box o Chromecast que funcionan con el servicio de internet; y ciertos usuarios que utilizan la televisión satelital objetivo de estudio. En primer lugar, para la implementación de la televisión satelital se debe disponer básicamente de antenas parabólicas o bien para banda Ku o para banda C y decodificadores, de hecho, los proveedores de televisión por cable utilizan esta tecnología para brindar el servicio a los usuarios, lógicamente sin revelar su infraestructura y cómo funciona. El objetivo por tanto y propósito de este capítulo es dar a conocer esta alternativa de la televisión satelital que permita al lector instalar, configurar equipos, canales de televisión libres y digitalizarlos para que puedan ser consumidos desde los diferentes dispositivos como son: ordenador, teléfono móvil, televisión led, etc. utilizando

tecnologías a nuestro alcance y permitiendo que el usuario pueda tener la alternativa de no restringir el uso de servicios de entretenimiento como es la televisión prepagada.

En realidad, detrás de un servicio tecnológico existe una infraestructura que puede ser muy compleja o simple, dependiendo al final del servicio que se ofrezca. Como requisito previo para poder emprender esta tarea es que el lector tenga conocimientos básicos de la tecnología por ejemplo instalación de software, sistemas Linux, manejo de dispositivos (televisión, móvil, ordenador, etc.), configuración básica de redes de ordenadores y por lo tanto consiga implementar por su cuenta un servicio de televisión satelital es decir adquiriera la capacidad de instalar, configurar equipos y sintonizar canales de televisión libres para su uso particular.

Desde el contexto anterior la señal satelital será digitalizada para convertirla en un servicio equivalente a IPTV que son las siglas de Televisión por Protocolo de Internet. Es decir, la televisión llega o se difunde a través de una dirección IP. En los últimos años, IPTV ha estado ofreciendo TV en vivo y bajo demanda servicios a través de redes basadas en IP. “Al ser una red con un ancho de banda suficiente es adecuado y más cómodo apreciar los contenidos digitales, IPTV ofrece una amplia variedad de canales con alta calidad e interactividad en comparación con otros sistemas de televisión tradicional” (Suliman Mohamed, Azad, & Al-Sakib, 2018).

Por otro lado, en cuanto a la normativa legal la Agencia de Regulación y Control de las Telecomunicaciones (ARCOTEL) menciona en el artículo 16, literal 3 la siguiente resolución de febrero de 2021:

Art. 16.- Todas las personas, en forma individual o colectiva, tienen derecho a:

La creación de medios de comunicación social y al acceso en igualdad de condiciones al uso de las frecuencias del espectro

radioeléctrico para la gestión de estaciones de radio y televisión públicas, privadas y comunitarias, y a bandas libres para la explotación de redes inalámbricas. (ARCOTEL, 2021).

También en Artículo 96 de la Ley Orgánica de Telecomunicaciones (LOT) se determina que el uso del espectro radioeléctrico técnicamente distinguirá las siguientes aplicaciones:

- Espectro de uso libre: Son aquellas bandas de frecuencias que pueden ser utilizadas por el público en general, con sujeción a lo que establezca el ordenamiento jurídico vigente y sin necesidad de título habilitante, ni registro.
- Espectro para uso determinado en bandas libres: Son aquellas bandas de frecuencias denominadas libres que pueden ser utilizadas para los servicios atribuidos por la Agencia de Regulación y Control y tan sólo requieren de un registro.
- Espectro para usos determinados: Son aquellos establecidos por la Agencia de Regulación y Control; dentro de este grupo pueden existir asignaciones de uso privativo o compartido.
- Espectro para usos experimentales: Son aquellas bandas de frecuencias destinadas a la investigación científica o para pruebas temporales de equipo.
- Espectro reservado: Son aquellas bandas de frecuencias destinadas a la seguridad pública y del Estado (Ministerio de Telecomunicaciones, 2015).

1. FUNDAMENTOS TEÓRICOS

1.1. Antenas Parabólicas

Básicamente una antena parabólica consiste en un plato de unos 60cm, comúnmente son de 60, 80 y 100 cm de diámetro, para banda tipo KU. Y antenas parabólicas de entre 120, 150 y más cm, para banda tipo C, por supuesto que existen antenas parabólicas de mayores tamaños y diferentes tipos.

Estas antenas suelen ir ocupadas con un amplificador denominado LNB (Low Noise Block “bloque de bajo ruido”) para recibir correctamente la señal, que llega debilitada desde el satélite (normalmente geoestacionario a 35800 km). “En la televisión digital por satélite se emplean frecuencias del rango SHF (entre 3 y 30 GHz) típicamente en banda KU (“Kurz-unten band”) entre 11,70 y 12,75 GHz, en banda C y Ka. El tipo de banda Ku se usa principalmente en las comunicaciones satelitales, siendo la televisión la más utilizada” (Huidobro & Ordóñez, 2014).



Ilustración 1. Antenas parabólicas banda ku y C respectivamente

En este capítulo nos enfocaremos en antenas para banda ku, sin embargo, vale mencionar que la *banda C* proporciona transmisiones de más baja potencia que la *banda ku*, pero de mayor cobertura geográfica, es más confiable bajo condiciones adversas del clima, con un plato de la antena receptora más grande, del orden de 3 metros, aunque con un mayor margen de error de apuntamiento, las frecuencias de *banda C* están más congestionadas y son más vulnerables hacia interferencias terrestres. “El plato de la antena receptora para *banda ku* puede ser más pequeño, del orden de 1,2 metros de diámetro, pero la cobertura es menor” (Huidobro, 2014).

1.2. Que es el LNB y Tipos

LNB es un dispositivo que permite reducir el ruido es decir es un conversor de reducción de ruido (Low Noise Block) por sus siglas en inglés, es el elemento activo encargado de recibir la señal. “La ganancia (G) de una antena depende fundamentalmente de su superficie diámetro y de la frecuencia de la señal recibida, así, a

mayor superficie y frecuencia de la señal, mayor será la ganancia. De su calidad dependerá la calidad de la imagen en la pantalla. El LNB irá situado en el foco de la antena parabólica” (Hidalgo & Cerdá Filiu, 2019).

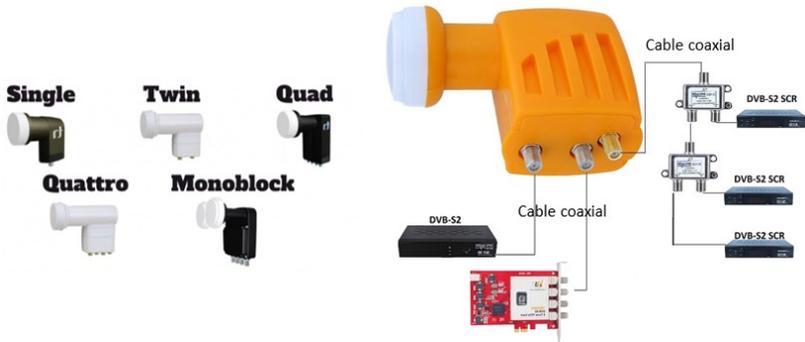


Ilustración 2. Esquema de la conexión de un LNB al decodificador

El LNB se sitúa en el foco del reflector y tiene como misión tratar la señal de frecuencia de la *banda ku* proveniente de la antena, de modo que la amplifica sin incorporar “ruido” y la transporta a la banda de frecuencia FI, frecuencia intermedia (950 MHz-2150 MHz), para su canalización sobre el cable coaxial de bajada a las unidades interiores. A continuación, se indica los tipos de LNB.

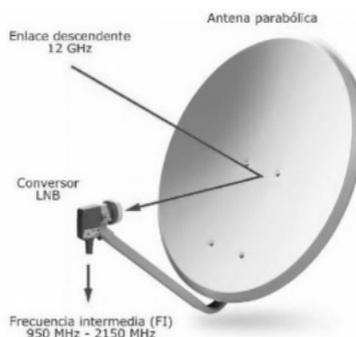


Ilustración 3. Antena parabólica tipo offset con el convertidor LNB

- LNB Single: El tipo LNB tiene una única salida.
- LNB Twin: Este tipo de convertidor LNB fue diseñado para tener dos salidas con cable coaxial (en una sola antena

parabólica), dirigidos a nuestro receptor.

- LNB Quad: Este tipo de LNB fue diseñado para hacer frente a la necesidad de tener 4 LNB instalados (en una sola antena parabólica).
- LNB Quattro: Este tipo de conversor LNB posee 4 polaridades, es decir, 4 salidas con polaridades diferentes. Vertical alto, vertical baja, horizontal alta y horizontal baja. Utilizados para multi-switchs.
- LNB Monoblock: Finalmente este es un conversor LNB de dos focos con una separación comprendida entre ellos (Monoblock de 4.3º o Monoblock de 6º). Normalmente sirven para captar la señal de dos satélites cercanos en una sola antena parabólica de 80cm. Los satélites más habituales de obtener esta señal en España son Astra y HotBird o Astra 23º y Astra 19º.
- Según (Profesional, 2021), los tipos de LNB son: Single, Twin, Quad, Quattro, y Monoblock.

1.3. Captar una Señal Libre

En primer lugar, cuando se instale la antena parabólica es importante que tenga una buena ubicación es decir que esté libre de obstáculos que puedan causar interferencia a la señal, puede ser por ejemplo en el techo o terraza de una casa, en la cima de un edificio, etc.

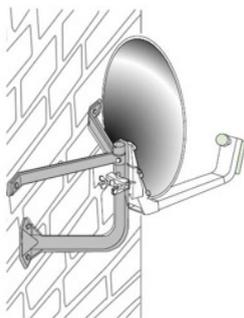


Ilustración 4. Ubicación de la antena parabólica sobre una pared

Para lograr captar una señal debemos orientar la antena

parabólica en el ángulo apropiado para que apunte en la dirección del satélite, podemos ayudarnos de un buscador de satélites como el que se indica en la siguiente ilustración, también desde algunos receptores de TV como el de AZBOX o a través de aplicaciones móviles disponibles en Google Play o Apple App Store.

“Existen sitios webs con aplicaciones que permiten calcular la longitud, latitud, acimut y elevación como por ejemplo el de Hispasat, S.A una empresa española operadora de satélites de comunicaciones que ofrece cobertura en América, Europa y el Norte de África” (HISPASAT, 2021).

“Los buscadores de TV son medidores portátiles que incorporan una base de datos actualizable de las emisiones que se reciben en el lugar de instalación, de manera que la orientación se simplifica al poder comprobar la calidad de los canales recibidos según la orientación de la antena” (Millán Esteller, 2018).



Ilustración 5. Receptor de Tv Azbox, aplicación móvil para establecer la posición de un satélite y dispositivo digital de búsqueda

Para Ecuador la mayoría de sus canales vía satélite se concentran en el Eutelsat 117 West A (*banda C*), siendo la oferta en *streaming* en general de carácter complementario. Sin embargo, se puede apuntar a otros satélites próximos a nuestra zona. “Debemos conocer el tipo de banda que soporta el satélite a veces pueden soportar tanto *banda ku* como *banda C* o una de las dos y también si son canales de tv o radio satelital libre o fta (free to air por sus siglas en inglés)

esto significa que cualquier persona puede acceder a ellos ya que no están codificados y sin la necesidad de pagar siendo totalmente legal” (EDS Blog - Exploradores de Satélites, 2021).

Básicamente necesitamos 3 datos para orientar la antena hacia el satélite esto es: azimut, elevación y polarización del LNB. Esta información se ingresa en la configuración del software de *Tvheadend* que veremos más adelante.

Azimut. “Tomando como referencia al norte se indica en grados el ángulo de giro horizontal de la antena, en el sentido de las agujas del reloj” (DIESL, 2021).

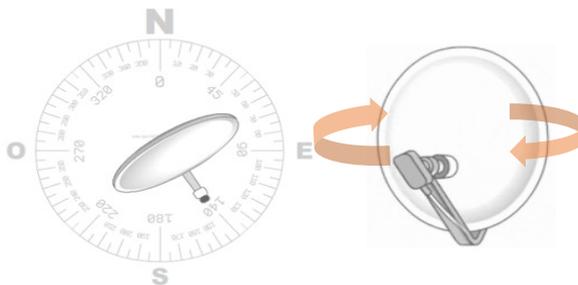


Ilustración 6. Posición del plato, azimut

Elevación. “Es el ángulo de inclinación/elevación de la antena sobre la vertical en el horizonte, que puede ajustarse con facilidad en la montura de la antena” (DIESL, 2021).

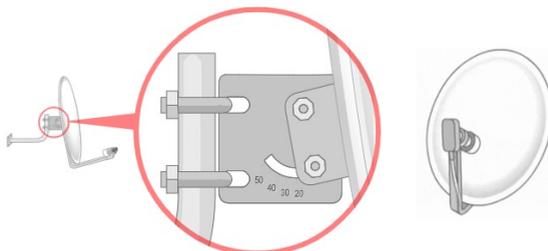


Ilustración 7. Grado de inclinación marcada en el soporte del plato

Polarización. “Es la rotación que debe tener el LNB respecto a la vertical del suelo. Se mide en grados” (DIESL, 2021).

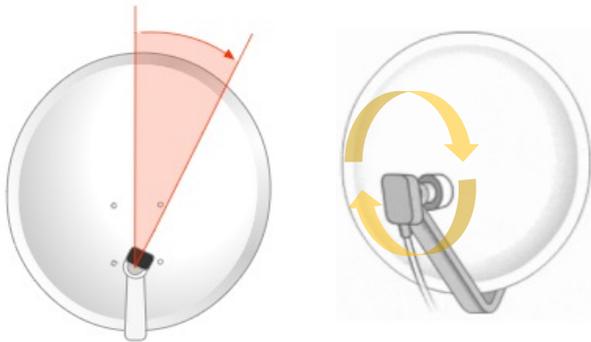


Ilustración 8. Rotación del LNB, se mide en grados

Todos estos datos dependen de dos factores: Nuestra posición geográfica y la posición del satélite cuya señal queremos captar. Se puede utilizar herramientas en sitios web como el de www.dishpointer.com mismo que utiliza *Google Maps* como base.

Este sistema nos dará nuestra ubicación y luego trazará una recta en color rojo misma que indica la dirección del satélite como puede verse en la ilustración, es decir en ese sentido es a donde debemos direccionar la antena parabólica.

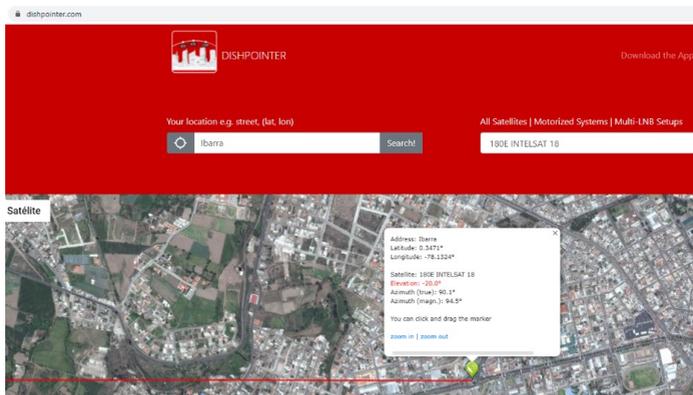


Ilustración 9. El sitio web de Dishpointer permite calcular el acimut y ajustes de elevación

Búsqueda de canales mediante el receptor de televisión, esto puede variar dependiendo del modelo o tipo de receptor, la

búsqueda de canales se puede parametrizar por satélites, frecuencia (transpondedor) específica, también para que busque canales libres y/o encriptados, etc.

1.4. Tarjeta Sintonizadora TBS 6909

La tarjeta sintonizadora TBS 6909 es un dispositivo que se instala o se conecta en un ordenador permitiendo sintonizar televisión en vivo. Básicamente esta tarjeta tiene un multiconmutador integrado que permite sólo 4 conexiones por satélite, ya sea desde un LNB Quad o Quattro. El TBS6909 permite capturar y transmitir señales en vivo de DVB-S y DVB-S2 de 8 transpondedores diferentes desde una ranura de expansión PCIe, por lo que es ideal para su uso en entornos de servidor IPTV. Cuando la ranura PCIe en el servidor es limitada, TBS6909 le permite recibir más canales que el sintonizador de TV dual o cuádruple habitual. Por lo tanto, es muy recomendable para sistemas de transmisión y transmisión de IPTV (TBSDTB, 2021).

“La capacidad de un transpondedor puede aumentar cuando se utiliza el estándar DSB-S2 de segunda generación en un 30% en relación con el anterior estándar DVB-S” (Pratt & Allnutt, 2020).



Ilustración 10. Tarjeta sintonizadora TBS-6909 DVB-s2

El cable coaxial desde el LNB irá conectado a uno de los puertos de la tarjeta TBS-6909 DVB-s2. Pueden instalarse múltiples tarjetas DVB-S2 dependiendo de la cantidad de ranuras PCI-

express que soporte el *Mainboard* y en este sentido colocar más antenas parabólicas que apunten a otros satélites con señales libres. Entre los beneficios que se puede tener al utilizar esta tarjeta son un sintonizador DVB-S2, podemos ver y grabar programas de radio y televisión por satélite en nuestro servidor o PC principal previamente realizando una instalación y configuración apropiadas, tiene soporte EPG (Guía electrónica de programas), descarga de datos de alta velocidad vía satélite, alta fiabilidad.

2. INFRAESTRUCTURA TECNOLÓGICA

2.1. Procedimiento Preliminar

En primera instancia el proceso en forma ordenada fundamentalmente es el siguiente:

- Instalar una *antena parabólica* como vimos en la sección anterior
- Captar con nuestra antena una señal libre como percibimos anteriormente.
- Instalar un Sistema Operativo, preferentemente LINUX UBUNTU
- Disponer e instalar una tarjeta sintonizadora como la TBS6909 DVB-S2 8 para PCI Express en nuestro ordenador.
- Instalar un software libre como es el tvheadend en Linux para recibir streams de video de nuestra fuente en este caso una señal de TV por satélite recibida mediante la tarjeta de video TBS6909
- Configurar el software tvheadend, y retransmitir a otros puntos de nuestra red local.
- Transcodificar la señal a otro formato de video.
- Instalar clientes de tvheadend, como kodi, vlc, etc.

En resumen, para un funcionamiento mínimo y apropiado el usuario debe disponer de al menos una antena parabólica satelital para *banda ku*, un ordenador, una tarjeta sintonizadora TBS6909 DVB-S2,

dispositivos de red como un *router* y/o un switch básicamente.

En consecuencia para una buena transmisión y calidad de video la solución es crear una infraestructura de red local cliente servidor privada (*protocolo de comunicación IP, redes inalámbricas y a través de cable UTP cat6*) de esta manera mejoraremos las transmisiones de video *streaming*, considerando básicamente el uso de un servidor y disponible al menos una tarjeta TBS6909 misma que nos permitirá centralizar y digitalizar la señal proveniente de las antenas parabólicas y a través de un software agrupar los diferentes canales libres o de pago disponibles de los satélites. La instalación, configuración y puesta en marcha de esta infraestructura lo veremos paulatinamente.

En primer lugar, en el servidor Linux se instalará la tarjeta TBS DBV-S2 (*mínimo 1 tarjeta*) aquí depende del número de ranuras de expansión de PCI express que disponga el mainboard, luego se instalará y configurará la plataforma para la televisión satelital y se capturará la señal de satélites que provean canales de TV de libre acceso.

2.2. Infraestructura Básica

A continuación, se puede apreciar una infraestructura básica de lo descrito. El medio de comunicación al usuario final puede ser mediante enlaces de cable UTP y enlaces inalámbricos wifi, y también mediante por ejemplo antenas inalámbricas, considerando que en la actualidad los dispositivos inalámbricos de largo alcance han mejorado ampliamente.

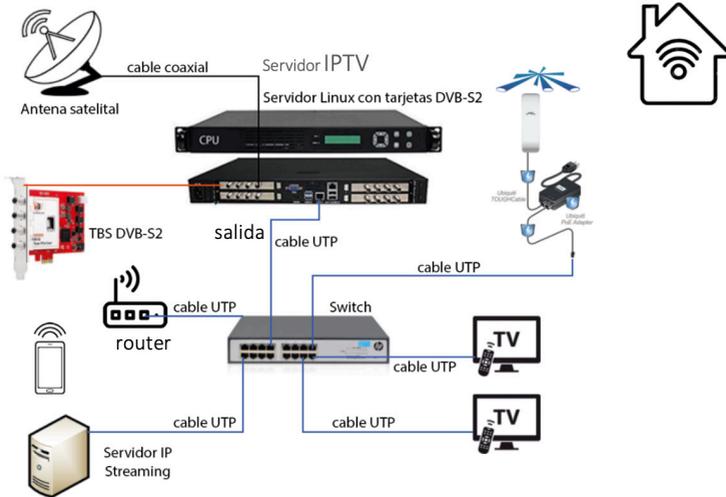


Ilustración 11. Infraestructura básica IPTV o transmisión de video streaming por IP. Elaboración propia.

Como se observó en la sección anterior de la antena satelital y desde el dispositivo LNB se conecta un cable coaxial que se dirige a un dispositivo decodificador en este caso el TBS DVB-S2 mismo que digitalizará la señal, luego en la salida se utilizará un cable de red UTP desde la tarjeta de red hacia un switch o también puede ser hacia un router para luego conectar a los diferentes dispositivos como son televisores, teléfonos móviles, ordenadores, etc. Esto puede apreciarse en el diagrama anterior de la infraestructura básica IPTV.

“En España se emplea el estándar DVB-S (Digital Video Broadcasting-Satellite) y su evolución DVB-S2, mismo que permite mayor eficiencia espectral. La imagen se codifica en MPEG2 para definición estándar y MPEG4 para alta definición” (Huidobro & Ordóñez, 2014).

En nuestro caso el receptor será nuestra tarjeta de sintonización TBS 6909 profesional de 8, sintonizador DVB-S y DVB-S2 con interfaz PCI Express.

2.3. Instalación de Linux Ubuntu Server

Ubuntu 20.04 Server Edition proporciona una base común y minimalista para una variedad de aplicaciones de servidor, como servicios de archivo, impresión, alojamiento web, alojamiento de correo electrónico, etc. Esta versión admite cuatro arquitecturas de 64 bits, sin embargo, utilizaremos la primera:

Los requisitos mínimos por experiencia del sistema recomendado son:

- PROCESADOR: Intel Core i7 de 4.8 GHz o superior
- RAM: 8 GB o más
- Disco sólido: Un mínimo de 500 gigabytes

Para realizar una instalación adecuada es importante utilizar *Ubuntu Server* más no *Ubuntu Desktop*. En *Linux Ubuntu Desktop* se presentan inconvenientes al momento de instalar los drivers de la tarjeta de TBS, sin embargo, en la versión de *Linux Ubuntu Server* en la práctica funciona adecuadamente, pero se debe instalar el entorno gráfico ya que por defecto Linux Ubuntu Server no lo incluye, caso contrario se pueden presentar errores de sonido.

Se supone que el lector tiene clara la instalación de Linux Ubuntu Server, solamente se profundizará en las partes importantes de la instalación. En primer lugar, se descarga Ubuntu Server del sitio web oficial: <https://ubuntu.com/download/server>. Se realiza la configuración de la red muy importante para la comunicación entre los diferentes dispositivos, al ser DHCPv4 se obtiene automáticamente la dirección IP, se recomienda ingresar manualmente las direcciones IP apropiadas para una red local, por ejemplo, la IP 192.168.204.129

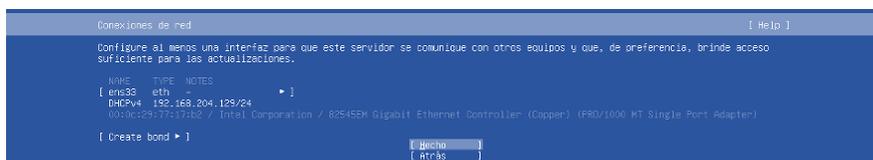


Ilustración 12. Configuración de la red en el momento de la instalación de Linux Ubuntu Server

Es conveniente optar por un disco grande considerando que el tamaño de videos para streaming u otros formatos de archivos multimedia ocupan un considerable espacio en disco, en este caso por ejemplo se instaló un disco de 20 GB, pero sería apropiado un disco de al menos 1TB o superior. Finalmente se confirma esta configuración, se selecciona continuar y se da enter en [hecho].

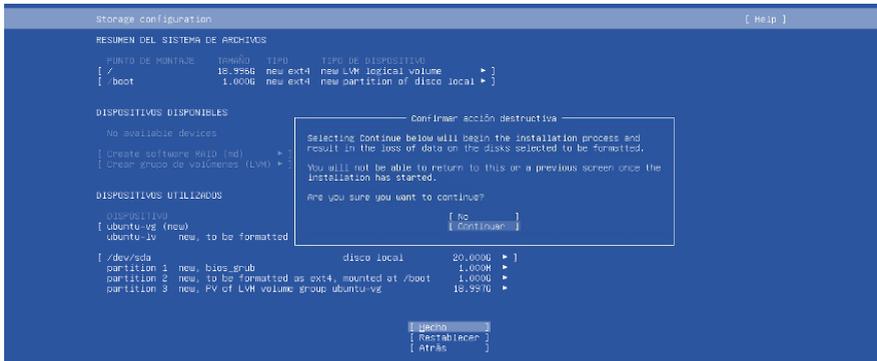


Ilustración 13. Configuración del almacenamiento, es decir para un disco de 20 GB

A continuación, se indica un formulario donde se ingresan los datos del perfil, esto es nombre y apellido, nombre del servidor, se elige un nombre de usuario, y se registra la contraseña. Finalmente se confirma la contraseña. Se da *enter* en [hecho]. Si se desea se puede activar la opción de Instalar servidor OpenSSH para un acceso remoto seguro. Es recomendable activar esta opción ya que va a permitir subir archivos como por ejemplo archivos de video o multimedia hacia el servidor. Damos *enter* en [hecho].

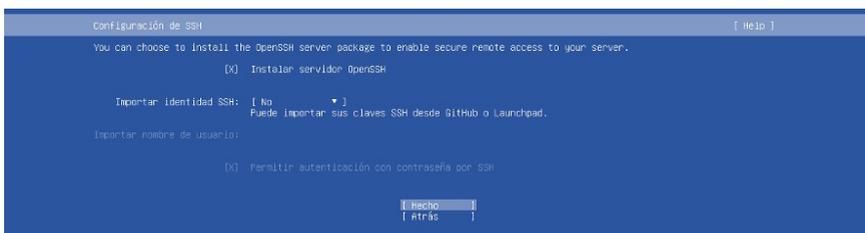


Ilustración 14. Instalación del Servidor OpenSSH para transferencia segura de archivos

Finalmente se inicia la instalación de Linux Ubuntu Server 20.04.2. cuando termine la instalación, damos *enter* en reiniciar la instalación y se ingresa al sistema. Hasta este punto se ha realizado una instalación básica de Linux Ubuntu Server 20.04.2, como puede apreciarse no se instala el entorno gráfico. En la práctica es muy importante instalar el entorno gráfico, caso contrario se tendrá inconvenientes al momento de instalar los drivers de la tarjeta TBS 6909. Por lo tanto, luego de esta instalación se procederá con la instalación de los drivers de esta tarjeta sintonizadora profesional con interfaz PCI Express capaz de recibir canales de TV digital de 8 transpondedores de satélite diferentes simultáneamente.

2.4. Instalación del Modo Gráfico

Previo a la instalación del modo gráfico, primeramente, se actualiza el sistema Linux.

Se ejecuta en el terminal Linux el comando `root@svriptycanal:/home/cesarg# apt update`

Y a continuación se ejecuta el comando `root@svriptycanal:/home/cesarg# apt upgrade`

En la práctica tomará un tiempo a que se actualicen las librerías, esto puede tardar unos minutos dependiendo de nuestro ancho de banda o velocidad de Internet. A continuación, se ejecutan los siguientes comandos:

```
root@svriptycanal:/home/cesarg# add-apt-repository universe
root@svriptycanal:/home/cesarg# add-apt-repository
multiverse
root@svriptycanal:/home/cesarg# apt install -no-install-recommends ubuntu-desktop
```

En este momento iniciará la instalación del escritorio de Ubuntu, por experiencia tomará un tiempo hasta que se instale todas las librerías dependiendo nuevamente de la velocidad del Internet.

```
root@svriptvcanal:/home/cesarg# apt upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
  linux-headers-5.4.0-65 linux-headers-5.4.0-65-generic
  linux-image-5.4.0-65-generic
  linux-modules-5.4.0-65-generic linux-modules-extra-5.4.0-65-generic
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  linux-headers-5.4.0-66 linux-headers-5.4.0-66-generic linux-image-5.4.0-66-generic
  linux-modules-5.4.0-66-generic linux-modules-extra-5.4.0-66-generic
The following packages will be upgraded:
  dmccmgr friendly-recovery gnupg gnupg-110n gnupg-utils gps gps-agent gps-wks-client
  gps-wks-server speedtest speedtest2 grub-common grub-pc grub-pc-bin grub2-common libldap-2.4-2
  libldap-common libpython3.8 libpython3.8-minimal libpython3.8-stdlib linux-generic
  linux-headers-generic linux-image-generic open-vm-tools pollinate python3-software-properties
  python3 python3.8 python3.8-minimal screen software-properties-common
34 upgraded, 5 newly installed, 0 to remove and 0 not upgraded.
Need to get 80.0 MB of archives.
After this operation, 361 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://ec.archive.ubuntu.com/ubuntu focal-updates/main amd64 libpython3.8 amd64 3.8.5-1~20.04.
2 [1624 kB]
1% [1: libpython3.8 1047 KB/1624 KB 64%] 40.0 kB/s 36min 12s_
```

Ilustración 15. Actualización del sistema Linux Ubuntu mediante los comandos indicados

2.5. Configuración e Instalación de Drivers de la Tarjeta



Ilustración 16. Tarjeta TBS6909 DVB-S2 como codificador de señal de antenas satelitales

Antes de instalar cualquier paquete en Linux y previo a la instalación a los drivers de TBS es muy recomendable ejecutar los siguientes comandos de actualización:

```
root@svriptvcanal:/home/cesarg# apt update
root@svriptvcanal:/home/cesarg# apt upgrade
```

Se instala el paquete *module-assistant* que permite instalar módulos para nuestro kernel de Linux, es decir, se encarga de compilar e instalar las librerías necesarias para el kernel por ejemplo para usar un hardware no soportado por el mismo y se desplegarán las opciones que se quiera descargar, compilar e instalar.

```
root@svriptvcanal:/home/cesarg# module-assistant
root@svriptvcanal:/home/cesarg# m-a
```

Al ejecutar estos comandos se presenta una pantalla de configuración. Modo interactivo y se selecciona UPDATE, PREPARE, EXIT al final se da clic en el botón OK

Seguidamente se instala la librería de perl para acceder a la información de la tabla de procesos y el paquete patchutils para trabajar con parches.

```
root@svriptvcanal:/home/cesarg# apt-get install libproc-
processtable-perl
```

```
root@svriptvcanal:/home/cesarg# apt-get install patchutils
```

A continuación, se descargan los drivers para sistemas Linux del sitio oficial <https://www.tbsiptv.com/downloads> y se busca el que corresponde a la tarjeta TBS 6909 DBV-S2 como puede apreciarse en la siguiente captura de pantalla.

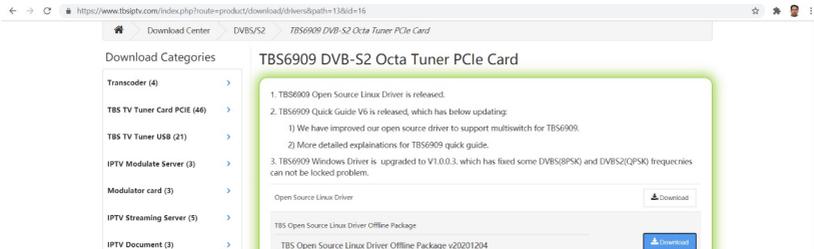
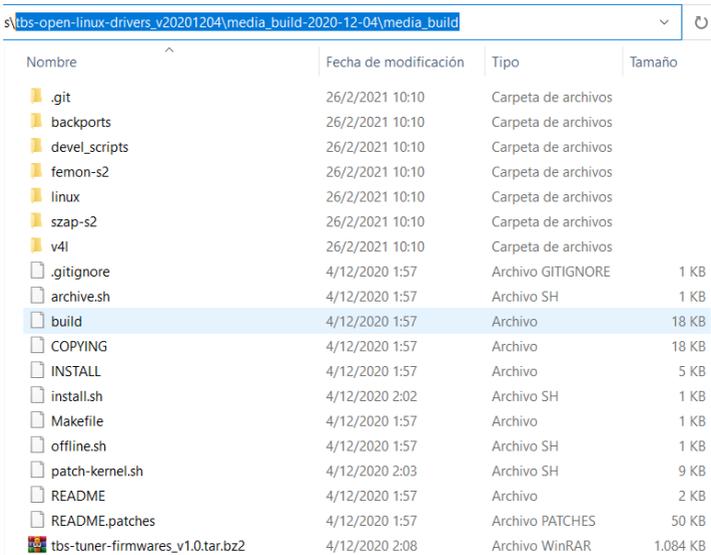


Ilustración 17. Descarga del driver de TBS para sistemas Linux

El archivo descargado se llama *tbs-open-linux-drivers_v20201204.zip* particularmente en la práctica se recomienda descargar este archivo ya que se encuentra más actualizado. Debido a que también se ofrece descargar e instalar desde https://github.com/tbsdtv/linux_media/wiki; sin embargo, en la experiencia suele dar inconvenientes ya que en el mismo sitio se menciona que no contiene todos los controladores de TBS. También es oportuno recordar que el sistema operativo debe ser Linux Ubuntu Server y no Ubuntu Desktop ya que este último suele dar problemas, la versión del *kernel* también es otro punto

a considerar, caso contrario no podrá instalar los drivers y por tanto no se reconocerá la tarjeta de TBS correctamente.

Aclarado lo anterior previo a la instalación se crea una carpeta llamada *tbsdrivers* puede ser en el directorio home de nuestro Linux, ubicamos el archivo descargado, *tbs-open-linux-drivers_v20201204.zip*.descomprimos el archivo e internamente existe otro archivo comprimido llamado *media_build-2020-12-04.tar.bz2* y se vuelve a descomprimir.



Nombre	Fecha de modificación	Tipo	Tamaño
.git	26/2/2021 10:10	Carpeta de archivos	
backports	26/2/2021 10:10	Carpeta de archivos	
devel_scripts	26/2/2021 10:10	Carpeta de archivos	
femon-s2	26/2/2021 10:10	Carpeta de archivos	
linux	26/2/2021 10:10	Carpeta de archivos	
szap-s2	26/2/2021 10:10	Carpeta de archivos	
v4l	26/2/2021 10:10	Carpeta de archivos	
.gitignore	4/12/2020 1:57	Archivo GITIGNORE	1 KB
archive.sh	4/12/2020 1:57	Archivo SH	1 KB
build	4/12/2020 1:57	Archivo	18 KB
COPYING	4/12/2020 1:57	Archivo	18 KB
INSTALL	4/12/2020 1:57	Archivo	5 KB
install.sh	4/12/2020 2:02	Archivo SH	1 KB
Makefile	4/12/2020 1:57	Archivo	1 KB
offline.sh	4/12/2020 1:57	Archivo SH	1 KB
patch-kernel.sh	4/12/2020 2:03	Archivo SH	9 KB
README	4/12/2020 1:57	Archivo	2 KB
README.patches	4/12/2020 1:57	Archivo PATCHES	50 KB
tbs-tuner-firmwares_v1.0.tar.bz2	4/12/2020 2:08	Archivo WinRAR	1.084 KB

Ilustración 18. Listado de librerías del driver para TBS DBV-S2

2.6. Pasos para la Instalación del Driver

Paso 1.

Se ingresa a la carpeta *media_build* y se ejecuta los siguientes comandos, al final se reinicia el sistema operativo linux.

```
root@svriptycanal:/media_build# make dir DIR= .//  
media  
root@svriptycanal:/media_build# make distclean  
root@svriptycanal:/media_build# init 6
```

Paso 2.

Se ejecuta el comando *make* mismo que permitirá compilar el código.

```
root@svriptvcanal:/media_build # make allyesconfig
root@svriptvcanal:/media_build # make -j4
root@svriptvcanal:/home/cesarg# make install
root@svriptvcanal:/home/cesarg# reboot
```

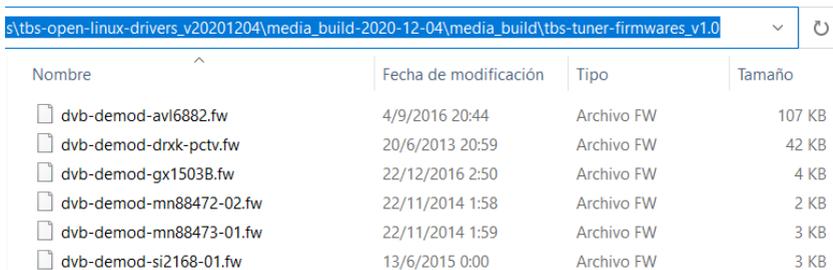
Paso 3.

Si todo lo anterior se ha ejecutado sin errores se puede verificar si el dispositivo está funcionando y activo correctamente, así que se ingresa el siguiente comando.

```
root@svriptvcanal:/home/cesarg# lspci -vvv | grep 6909
```

Paso 4.

A continuación, se instala el firmware, es decir un programa básico que controla los circuitos electrónicos integrados del dispositivo de TBS. En la misma carpeta de *media_build* se tiene el archivo *tbs-turner-firmwares_v1.0.tar.bz2*, y se descomprime.



Nombre	Fecha de modificación	Tipo	Tamaño
<input type="checkbox"/> dvb-demod-avl6882.fw	4/9/2016 20:44	Archivo FW	107 KB
<input type="checkbox"/> dvb-demod-drxk-pctv.fw	20/6/2013 20:59	Archivo FW	42 KB
<input type="checkbox"/> dvb-demod-gx1503B.fw	22/12/2016 2:50	Archivo FW	4 KB
<input type="checkbox"/> dvb-demod-mn88472-02.fw	22/11/2014 1:58	Archivo FW	2 KB
<input type="checkbox"/> dvb-demod-mn88473-01.fw	22/11/2014 1:59	Archivo FW	3 KB
<input type="checkbox"/> dvb-demod-si2168-01.fw	13/6/2015 0:00	Archivo FW	3 KB

Ilustración 19. Archivos del tbs_turner-firmwares

Paso 5.

Seguidamente, todos estos archivos se copian al directorio */lib/firmware*, es necesario reiniciar el sistema operativo Linux. A continuación, se verifica que los controladores han sido cargados correctamente, y que los sintonizadores hayan sido reconocidos por el sistema. Para ello se abre un terminal y se ingresa el comando *dmesg* (diagnostic message / mensajes de diagnóstico), es una herramienta exclusiva de sistemas Linux con la capacidad de listar los avisos temporales generados por el kernel, como se indica a continuación:

```
root@svriptvcanal:/home/cesarg# dmesg | grep frontend
root@svriptvcanal:/home/cesarg# dmesg | grep TBS
```

Este comando debe devolver los cuatro sintonizadores detectados independientemente, a los cuales les asignará un identificador de adaptador, y un *frontend*.

2.7. Instalación del Servidor Tvheadend

En primera instancia *tvheadend* es un software de servidor que lee transmisiones de TV para Linux, FreeBSD, Android y las publica como transmisiones de Internet. Admite múltiples entradas, un sintonizador SAT> IP, DVB-S, DVB-S2, DVB-C, DVB-T, DVB-T2, ATSC, ISDB-T, IPTV, y HDHomeRun. Se utiliza generalmente para enviar video a dispositivos receptores como televisiones inteligentes y decodificadores a través de una red LAN. Se puede grabar programas como una grabadora de video digital, y también tiene la capacidad de transcodificar de MPEG2 a h264 y h265. (TVHEADEND, 2021).

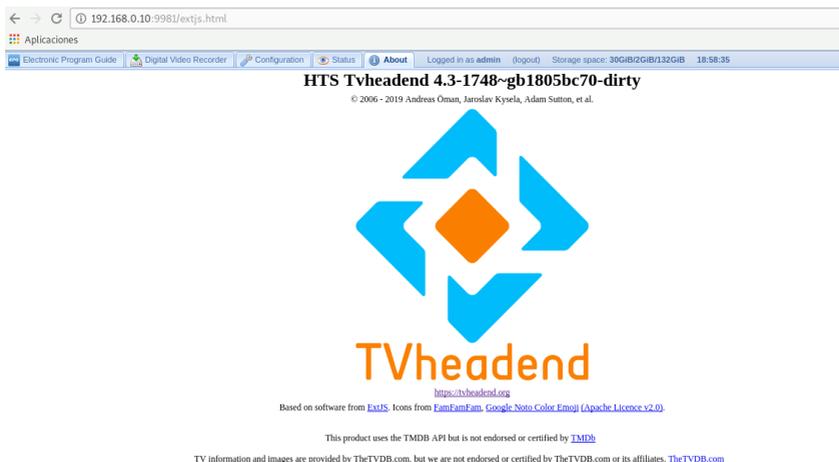


Ilustración 20. Aplicación de servidor de software libre Tvheadend

El sitio oficial de Tvheadend es <https://tvheadend.org/projects/tvheadend>, tvheadend ofrece HTTP (VLC, MPlayer), HTSP (Kodi, Movian) y SAT> IP streaming. El código está alojado en github <https://github.com/tvheadend/tvheadend>. A continuación, se indican los pasos para la instalación:

Paso 1.

Se ejecuta el siguiente comando para instalar el paquete requerido

```
root@svriptvcanal:/home/cesarg# apt-get -y install coreutils wget apt-transport-https lsb-release ca-certificates
```

Paso 2.

Se agrega el repositorio de clave PGP

```
root@svriptvcanal:/home/cesarg# wget -qO- https://dooser.io/keys/tvheadend/tvheadend/pgp | sudo apt-key add -
```

Paso 3. Se agrega el repositorio

```
root@svriptvcanal:/home/cesarg# add-apt-repository 'deb https://apt.tvheadend.org/stable bionic main'
```

Paso 4.

Se instala tvheadend y se reinicia el ordenador con el comando #
init 6

```
root@svriptvcanal:/home/cesarg# apt-get update  
root@svriptvcanal:/home/cesarg# apt-get install tvheadend -y
```

Paso 5.

Se presenta una pantalla como la siguiente y se ingresa el nombre de usuario de tipo Administrador, puede ser Admin como se indica en la pantalla siguiente:

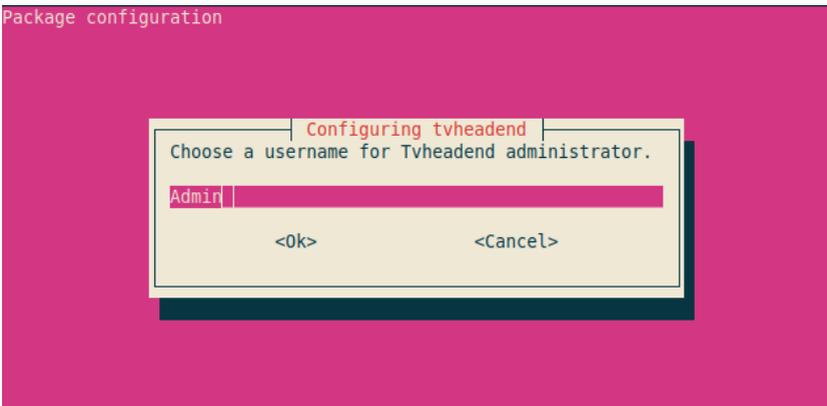


Ilustración 21. Pantalla de configuración de tvheadend, usuario administrador

Paso 6.

Seguidamente es necesario ingresar una contraseña adecuada digitando una combinación de letras, números o caracteres alfanuméricos.

Paso 7.

Se presenta el link con un número de puerto de acceso, se puede ingresar en un navegador web como por ejemplo *http://localhost:9981*

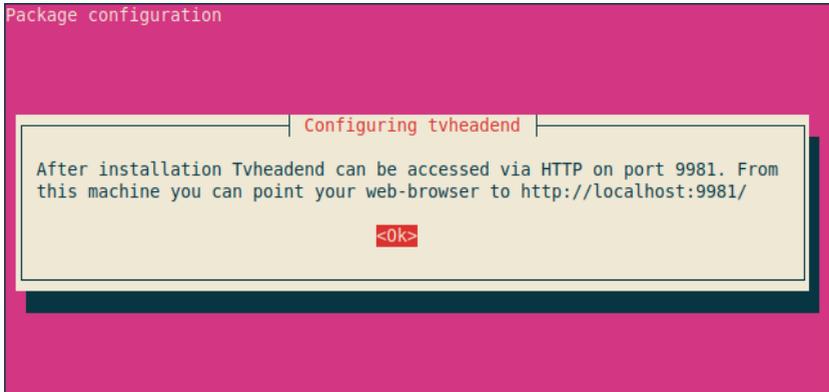


Ilustración 22. Configuración de tvheadend, link y número de puerto 9981

Hasta este punto finalmente se tiene instalada la aplicación de servidor del *tvheadend*, es de aquí adelante que se procede con la configuración de este software, para tener acceso a los diferentes canales de televisión.

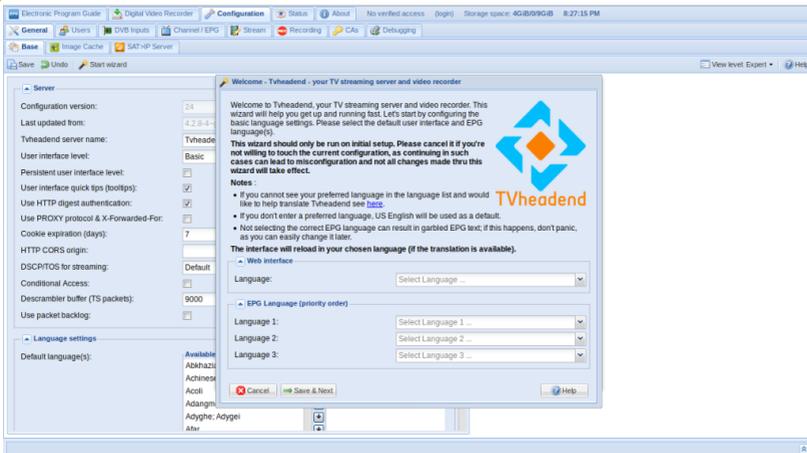


Ilustración 23. Pantalla principal del servidor de tvheadend

En caso de que el servidor se detenga se ingresa el siguiente comando.

```
root@svriptvcanal:/home/cesarg# tvheadend -C
root@svriptvcanal:/home/cesarg# service tvheadend
restart
```

2.8. Configuración Final de la Aplicación Tvheadend

El software tvheadend es un servidor de transmisión de TV y una grabadora de video digital. Esta aplicación según (TVHEADEND, 2021) admite las siguientes entradas: DVB-C(2), DVB-T(2), DVB-S(2), ATSC, SAT>IP, HDHomeRun, IPTV (UDP, HTTP). También admite las siguientes salidas: HTTP, HTSP (own protocol), SAT>IP

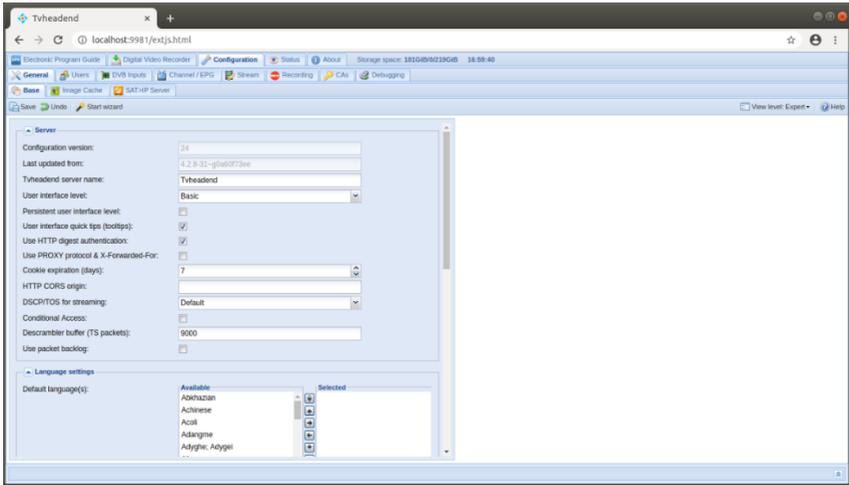


Ilustración 24. Datos iniciales de configuración

A continuación, en la siguiente pantalla, en la opción de TV adapters puede observarse el *Universal LNB only* activo con un punto color verde, si está en rojo quiere decir que está inactivo. Al lado en el formulario se selecciona la red que hace referencia al satélite HISPASAT 74.0 W

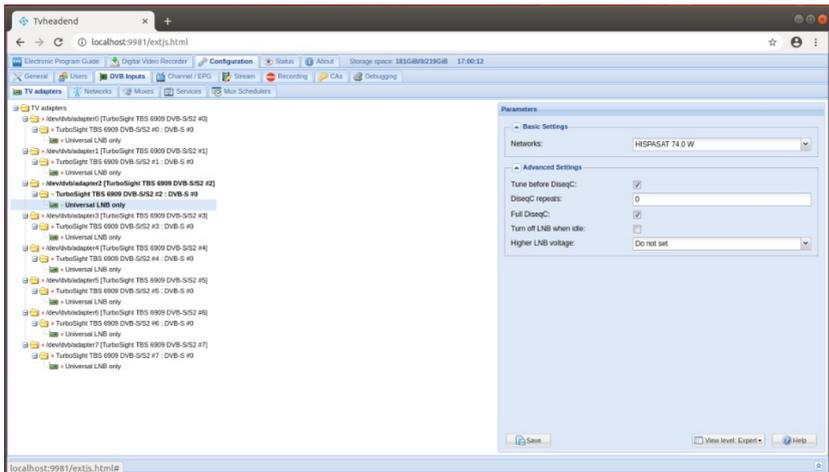


Ilustración 25. Configuración de tvheadend, TV adapters

Seguidamente, Turbosight TBS 6909 DVB-S/S2 #2: DVS-S #0, También puede observarse que está activado, esto es automático, si es que la instalación de los drivers fue realizada correctamente. En el formulario de la parte derecha se ingresa el nombre, se selecciona Universal LNB only, y las demás opciones se deja por defecto, tal como se indica en la pantalla a continuación:

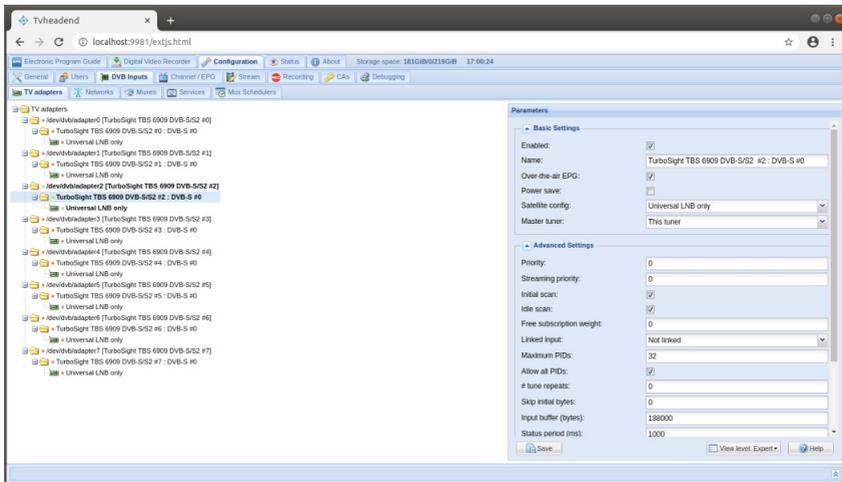


Ilustración 26. Configuración. Ingreso y selección de parámetros básicos

En la pantalla siguiente en la pestaña servicios se presenta un listado de todos los canales disponibles del satélite de HISPASAT 74.0 W, en el icono  que se encuentra en la parte izquierda damos clic para descargar el archivo que será ejecutado por ejemplo con VLC, como se indicaba anteriormente.


```

mc [root@srv_lptv]:usr/share/dvb/dvb-s
File Edit View Search Terminal Tabs Help

root@srv_lptv:/home/vscriptv
mc [root@srv_lptv]:usr/share/dvb/dvb-s

/usr/share/dvb/dvb-s/Hispasat-30.0W [----] 0 L:[ 1+14 15/298] *(244 /5230b) 0010 0x00A
# Hispasat 30.0W
# freq pol sr fec

[CHANNEL]
<----->DELIVERY_SYSTEM = DVBS2
<----->FREQUENCY = 10730000
<----->POLARIZATION = HORIZONTAL
<----->SYMBOL_RATE = 27500000
<----->INNER_FEC = 3/4
<----->ROLLOFF = AUTO
<----->MODULATION = PSK/8
<----->STREAM_ID = 1
<----->PLS_CODE = B
<----->INVERSION = AUTO
[]
[CHANNEL]
<----->DELIVERY_SYSTEM = DVBS2
<----->FREQUENCY = 10730000
<----->POLARIZATION = VERTICAL
<----->SYMBOL_RATE = 30000000
<----->INNER_FEC = 3/4
<----->ROLLOFF = AUTO
<----->MODULATION = PSK/8
<----->STREAM_ID = 1
<----->PLS_CODE = B
<----->INVERSION = AUTO

```

Ilustración 29. Parámetros de configuración el satélite Hispasat-30.0W

2.9. Clientes para Tvheadend

Tenemos muchos clientes para el tvheadend los más conocidos son kodi, tvhguide, serviio, y vlc, los hay para Linux, Windows o dispositivos móviles algunos son libres y otros de pago, se puede inclusive desarrollar aplicaciones web o aplicaciones móviles. Desde estos clientes nos conectaremos al servidor y podremos visualizar los canales de televisión disponibles.



Ilustración 30. Mediante el reproductor de VLC podemos acceder a los canales detectados

Por experiencia se debe tener instalado uno de los siguientes clientes, como por ejemplo VLC player. Desde el servidor tvheadend en la pestaña de Channels (Canales) se da clic en el icono play, a continuación, se descargará un archivo que contiene la dirección del stream del canal, como se indica a continuación, seguidamente estos archivos pueden ser ejecutados por un reproductor como el VLC. VLC es un reproductor multimedia libre y de código abierto multiplataforma y un «framework» que reproduce la mayoría de los archivos multimedia, así como DVD, Audio CD, VCD y diversos protocolos de transmisión (VIDEOLAN, 2021).

```
#EXTM3U
```

```
#EXTINF:-1,LA RED 2
```

```
http://localhost:9981/stream/channel/dec23e03d81e-  
da6c20f920641ec39820?ticket=B1D5239F1E57D53A8C7D-  
56B23B70C866322EBA68
```

```
#EXTM3U
```

```
#EXTINF:-1,TVN 4
```

```
http://localhost:9981/stream/channel/72c4c4164787070e2c-  
65f89d773537fe?ticket=0EF2AC7097EA-  
7299D36AA9220174ED9C230F747E
```

Estos archivos pueden cargarse desde el reproductor VLC Player, opción Medio, abrir ubicación de red, pestaña red, y podemos introducir la URL, o también hacer clic derecho sobre el archivo y abrir con VLC. Se presentará el canal como en la ilustración anterior.

3. Conclusiones

La tecnología de IPTV puede estar al alcance de todos si se dispone del hardware y software libre mencionados en este capítulo, de esta manera podemos disfrutar de canales libres de televisión y aprovechar la tecnología dentro de este ámbito.

A continuación, se presentan algunos ejemplos de líneas de estudio como puede ser: Desarrollo de Aplicaciones en JavaScript

con *nodejs* bajo un servidor IPTV *streaming* que permitan el manejo de *streaming* de video, mediante un reproductor de video desde la web, el acceso a un repositorio de videos con formato .mp4. y el acceso al flujo de bytes (video *streaming*) recibidos del satélite al receptor, es decir lo que hace el VLC *player*, pero a un reproductor web. También se puede considerar desarrollar aplicaciones móviles para un mejor control de acceso a los diferentes canales *streaming*. Mejoramiento de la infraestructura de IPTV a través de nuevos recursos de hardware y software.

4. Bibliografía

- ARCOTEL. (21 de 01 de 2021). *Agencia de Regulación y Control de las Telecomunicaciones*. Obtenido de <https://www.arcotel.gob.ec/>:<https://www.arcotel.gob.ec/wp-content/uploads/downloads/2021/02/Resolucion-ARCOTEL-2021-0121.pdf>
- DIESL. (08 de 07 de 2021). *DIESL*. Obtenido de www.diesl.es: <https://www.diesl.es/Monograficos/Manual-de-Instalacion-de-una-Antena-Parabolica>
- DISHPOINTER. (30 de 11 de 2021). *DISHPOINTER*. Obtenido de <https://www.dishpointer.com>: <https://www.dishpointer.com>
- EDS Blog - Exploradores de Satélites. (08 de 07 de 2021). *PORTALEDS*. Obtenido de www.portaleds.com: <https://www.portaleds.com/espanol/listar.php?sat=2432>
- Hidalgo, I. T., & Cerdá Filiu, L. M. (2019). *Infraestructuras comunes de telecomunicación en viviendas y edificios*. Madrid: Parainfo, SA.
- HISPASAT. (30 de 11 de 2021). *HISPASAT*. Obtenido de <https://www.hispasat.com>:<https://www.hispasat.com/es/informacion-util/apunta-tu-antena-a-hispasat>
- Huidobro, J. M. (2014). *Tecnologías, Redes y Servicios*. Madrid: RA-MA, S.A.
- Huidobro, J. M., & Ordóñez, J. L. (2014). *Comunicaciones por Radio*. Madrid: RA-MA.

- Millán Esteller, J. M. (2018). *Técnicas y procesos en infraestructuras*. Madrid: Parainfo, SA.
- Ministerio de Telecomunicaciones. (18 de 02 de 2015). *Ministerio de Telecomunicaciones*. Obtenido de <https://www.telecomunicaciones.gob.ec>: <https://www.telecomunicaciones.gob.ec/wp-content/uploads/downloads/2016/05/Ley-Org%C3%A1nica-de-Telecomunicaciones.pdf>
- Pratt, T., & Allnutt, J. (2020). *Satellite Communications*. Hoboken NJ: John Wiley & Sons Ltd.
- Profesional, T. (02 de 02 de 2021). *www.tdtprofesional.com*. Obtenido de <https://www.tdtprofesional.com/>: <https://www.tdtprofesional.com/blog/tipos-de-lnb-satelites/#:~:text=LNB%20Quattro%3A%20Este%20tipo%20de,Utilizados%20para%20multiswitch>.
- Suliman Mohamed, F., Azad, S., & Al-Sakib, K. (2018). *IPTV Delivery Networks*. Hoboken, NJ: John Wiley & Sons Ltd.
- TBSDTB. (30 de 11 de 2021). TBSDTB. Obtenido de <https://www.tbsdtv.com/>: <https://www.tbsdtv.com/products/tbs6909-dvb-s2-8-tuner-pcie-card.html>
- TVHEADEND. (15 de 01 de 2021). *Tvheadend*. Obtenido de <https://tvheadend.org>: <https://tvheadend.org/projects/tvheadend>
- VIDEOLAN. (30 de 11 de 2021). *VLC media player*. Obtenido de <https://www.videolan.org>: <https://www.videolan.org/vlc/index.es.html>



PUCE
IBARRA

ISBN: 978-9978-375-64-8



9 789978 375648