



**PRÁCTICAS DE LABORATORIO
PARA PROGRAMAS DE
INGENIERÍA INFORMÁTICA**

An abstract, light blue and white graphic of a circuit board pattern, consisting of various lines, circles, and dots, resembling a network or data flow diagram.

PRÁCTICAS DE LABORATORIO PARA PROGRAMAS DE
INGENIERÍA INFORMÁTICA





PRÁCTICAS DE LABORATORIO EN INGENIERÍA INFORMÁTICA

ISBN Obra Independiente

No. 978-958-57589-8-8

Sello editorial

Escuela Militar de Aviación Marco Fidel Suárez

Responsable ISBN

Jefe Sección Investigación

Teniente Andrea Carolina Gómez Ruge

Jefe del Programa de Ingeniería Informática

Capitán Paola Andrea Zarate Luna

Autores

Paola Andrea Zarate Luna

Sandra Viviana Wilches Arce

Jorge Alberto Correa

Humberto Amaya Alvear

John Alexander Vargas

Sofi Lorena Riocampo

Diseño, Diagramación e impresión

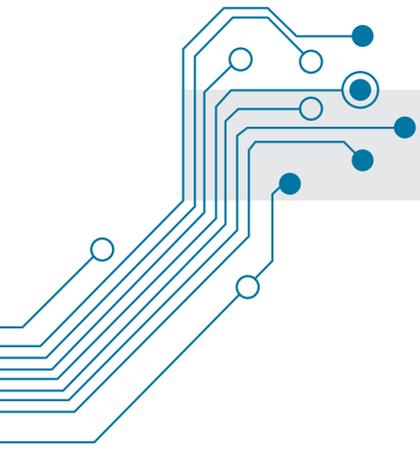
Milenio Editores e Impresores E.U.

ESCUELA MILITAR DE AVIACIÓN

“MARCO FIDEL SUÁREZ”

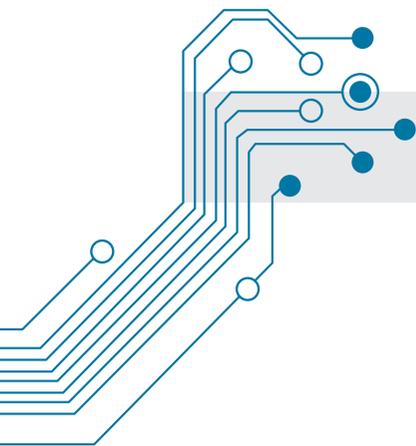
PROGRAMA DE INGENIERÍA INFORMÁTICA

SANTIAGO DE CALI, 2018



Contenido

INTRODUCCIÓN.....	5
SISTEMA DE INFORMACIÓN GEOGRÁFICA – SIG.....	7
ESTRUCTURA DE DATOS.....	29
LENGUAJES DE PROGRAMACIÓN.....	41
BASES DE DATOS	75
INFORMÁTICA	83
REFERENCIAS	89



Introducción

El cambio cultural y generacional del siglo XXI requiere también de un cambio y una adaptación de la docencia y las didácticas del aula, el reto está orientado ahora a romper paradigmas de la educación convencional que requieren un cambio en los sistemas de enseñanza vigentes.

En tal sentido el desarrollo de guías de laboratorio para programas de ingeniería informática, permite mantener informados a los estudiantes de las propuestas y trabajos prácticos de las asignaturas y también normaliza la presentación de las didácticas en escenarios de laboratorios, asegurando una mejor calidad de la docencia en coordinación con la comunidad académica y alineados con los propósitos de formación profesional para ingenieros informáticos.

En este texto se presenta una propuesta didáctica de trabajo orientada al desarrollo de las prácticas de laboratorio para programas de ingeniería informática. Este material les permitirá a los estudiantes una participación más activa y el desarrollo de sus competencias a través de prácticas de laboratorio, consecuente con una mejor apropiación del conocimiento teórico de asignaturas como estructura de datos, lenguajes de programación, ingeniería de software, seguridad informática, sistemas operativos, bases de datos y sistema de información geográfica.

Por otra parte, las guías de práctica de los laboratorios servirán de mapa de ruta para los docentes que trabajan las materias mencionadas anteriormente y para los estudiantes que las cursan. El esquema propuesto en cada una de ellas permite que la temática de laboratorio se mantenga en el tiempo conforme a lo estipulado; algunas preguntas están orientadas a la búsqueda de información que complemente su conocimiento del tema.

La ejecución de las prácticas de laboratorios es una oportunidad para comprobar si los conceptos han sido aprehendidos y afianzar conocimientos; estas actividades buscan la aplicación de la teoría con casos de uso aplicados a la realidad.

En cada capítulo se exponen las guías de laboratorio por asignatura donde se detallarán los materiales y equipos requeridos para el desarrollo de estas, así como las actividades, resultados y conclusiones esperados al final de cada práctica.

1.

SISTEMAS DE INFORMACIÓN GEOGRÁFICA (SIG)

A través de las prácticas de laboratorio de Sistemas de Información Geográfica (SIG I) los estudiantes podrán comprender la funcionalidad de los sistemas de información geográfica, vistos como un conjunto de herramientas capaces de integrar, almacenar, editar, analizar, compartir y mostrar la información geográficamente referenciada, que permiten la organización, almacenamiento, manipulación y análisis de grandes cantidades de datos procedentes del mundo real que están vinculados a una referencia espacial, facilitando la incorporación de aspectos sociales, económicos y ambientales que conducen a la toma de decisiones de una manera más eficaz.

Los SIG permiten a los usuarios crear consultas interactivas, analizar la información espacial, editar datos, mapas y presentar los resultados de todas estas operaciones.

PRÁCTICA: CONCEPTOS BÁSICOS SIG SISTEMA DE INFORMACIÓN GEOGRÁFICA – (SIG)

1. Objetivos

- Entender los conceptos y términos básicos manejados en los Sistemas de Información Geográfica.
- Investigar en Internet las siguientes definiciones y posteriormente definirás con sus propias ideas.

2. Marco Teórico

Un sistema de Información Geográfica - SIG es un conjunto de herramientas que integra y relaciona diversos componentes (usuarios, hardware, software, procesos) que permiten la organización, almacenamiento, manipulación, análisis y modelización de grandes cantidades de datos procedentes del mundo real que están vinculados a una referencia espacial, facilitando la incorporación de aspectos sociales-culturales, económicos y ambientales que conducen a la toma de decisiones de una manera más eficaz.

En un sentido más genérico, los SIG son herramientas que permiten a los usuarios crear consultas interactivas, analizar la información espacial, editar datos, mapas y presentar los resultados de todas estas operaciones. La tecnología de los SIG puede ser utilizada para investigaciones científicas, la gestión de los recursos, la gestión de activos, la arqueología, la evaluación del impacto ambiental, la planificación urbana, la cartografía, la sociología, la geografía histórica, el marketing, la logística por nombrar unos pocos.

3. Material y equipo

- Guía de laboratorio Conceptos básicos SIG I
- Computador
- Internet

4. Actividades, resultados y conclusiones

Realizar la investigación sobre los conceptos y definiciones de las palabras que se relacionan a continuación.

- a. Geomática
- b. Cartografía.
 - Análoga
 - Digital
 - Temática
 - Militar
- c. Topografía.
- d. Geodesia
 - Clásica
 - Satelital.
- e. Geoide y Elipsoide
- f. Coordenadas polares y Coordenadas geográficas.
- g. Meridianos y paralelos.
- h. Husos horarios o zonas horarias
- i. Coordenadas WGS 84 y Magna Sirgas.
- j. DATUM.
- k. Escala y factor de escala
- l. Herramientas CAD para cartografía.
- m. Digitalización de cartografía.
- n. Proyecciones cartográficas.

5. Bibliografía

- International GPS Service: <http://igscb.jpl.nasa.gov>
- International Earth Rotation Service: <http://www.iers.org>
- International Terrestrial Reference Frame: <http://itrf.ensg.ign.fr>

Elaborado por: Docente Ing. Jorge Correa

PRÁCTICA: SISTEMA DE POSICIONAMIENTO GLOBAL - GPS SISTEMA DE INFORMACION GEOGRÁFICA – (SIG)

1. Objetivos

- Afianzar los conocimientos adquiridos en clase referentes al funcionamiento de los Sistemas Globales de Navegación Satelital GNSS.

2. Marco Teórico

El sistema de posicionamiento global desarrollado por el Departamento de Defensa de los EE.UU. proporcionará nuevas oportunidades para la Topografía y Geodesia. Facilita nuevos medios numéricos y técnicas para la medición que las que venían realizando los geodestas en épocas anteriores. Las técnicas GPS requieren no sólo el conocimiento operativo de los receptores sino un amplio conocimiento de los fundamentos matemáticos para un mejor control de calidad de los resultados. Por el momento y sin que todavía se hayan desarrollado nuevas experiencias además de las geodésicas y topográficas de aplicación militar se ha utilizado el sistema GPS para:

- | | |
|--------------------------------------|--------------------------------------|
| -Navegación en tierra. | - Alta precisión cinemática. |
| - Navegación en mar. | - Guía de robots y otras máquinas. |
| - Navegación en aire. | - Topografía catastral. |
| - Navegación espacial. | - Densificación de redes geodésicas. |
| - Navegación en ríos. | - Fotogrametría. |
| - Navegación en puertos. | - Deformación terrestre. |
| - Navegación en vehículos de recreo. | - Control activo de estaciones. |
| - Hidrografía. | |

3. Material y equipo

- Guía de laboratorio
- Un computador
- Conexión a Internet

4. Actividades, resultados y conclusiones

Realizar la investigación sobre los conceptos y temas que se relacionan a continuación:

- a. ¿En qué consiste el sistema GPS? ¿Cuál es su principio básico de funcionamiento?

- b. ¿Qué otros sistemas de posicionamiento se conocen?

- c. ¿Cuáles son los segmentos en los que se divide el sistema GPS?. Explique la función de cada uno de ellos.

- d. ¿Cuáles son los dos tipos de GPS que existen?, ¿Cuál es su diferencia?

- e. ¿Cuáles son las frecuencias portadoras y los códigos con los que opera el sistema GPS? Explíquelos brevemente.

f. ¿Cuáles son las degradaciones voluntarias? Explique.

g. ¿Qué tipos de servicio provee el sistema GPS?

h. ¿Qué es un receptor monofrecuencia?, ¿Qué es un receptor bifrecuencia?, ¿Cuál es más preciso? ¿Por qué?

i. ¿En qué consiste el modo natural o de posicionamiento absoluto? ¿Y el modo diferencial o posicionamiento relativo?

j. ¿Qué es la pseudo-distancia?

k. ¿Qué errores puede presentar el sistema GPS? Explique brevemente.

l. Explique brevemente los métodos de medición con GPS.

m. ¿En qué consiste el formato RINEX?

n. ¿Qué es DOP? PRN?

o. ¿Qué es un geoide? ¿Qué es un elipsoide?

p. ¿Qué es un sistema geodésico de referencia? (ejemplos) ¿Qué es WGS84? ¿Qué es el DATUM?

q. ¿En qué consiste la Disponibilidad Selectiva?, ¿El anti-spoofing?

r. ¿Por qué es tan importante la precisión en el tiempo para el sistema GPS?

s. Enumere cinco aplicaciones del GPS.

5. Bibliografía

- PROVENZA Guiu José Luis. El sistema de posicionamiento global-4769019.pdf
Recuperado de: <https://dialnet.unirioja.es/descarga/articulo/4769019.pdf>
- International GPS Service. Recuperado de: <http://igscb.jpl.nasa.gov>
- International Earth Rotation Service. Recuperado de: <http://www.iers.org>
- International Terrestrial Reference Frame. Recuperado de: <http://itrf.ensg.ign.fr>

Elaborado por: Docente Ing. Jorge Correa

PRÁCTICA: SISTEMA DE INFORMACIÓN GEOGRÁFICA I SISTEMA DE INFORMACIÓN GEOGRÁFICA – (SIG)

1. Objetivos

- Familiarizar al estudiante con los datos mostrados por un SIG.

2. Marco teórico

La georreferenciación es el uso de coordenadas en un mapa para asignar una ubicación espacial a entidades cartográficas. Todos los elementos de una capa de mapa tienen una ubicación geográfica y una extensión específicas que permiten situarlos en la superficie de la Tierra o cerca de ella. La capacidad de localizar de manera precisa las entidades geográficas es fundamental tanto en la representación cartográfica como en SIG.

La correcta descripción de la ubicación y la forma de entidades requiere un marco para definir ubicaciones del mundo real. Un sistema de coordenadas geográficas se utiliza para asignar ubicaciones geográficas a los objetos. Un sistema de coordenadas de latitud-longitud global es uno de esos marcos. Otro marco es un sistema de coordenadas cartesianas o planas que surge a partir del marco global.

3. Material y equipo

- Guía de laboratorio
- Un computador por estudiante
- Conexión a Internet

4. Actividades, resultados y conclusiones

Desde el navegador ingresar a la página web:

<http://maps.google.com/maps/ms?ie=UTF8&hl=en&t=p&msa=0&msid=106484775090296685271.0004681a37b713f6b5950&-ll=32.639375,-110.390625&spn=15.738151,25.488281&z=5>

a. Analizar qué datos importantes se están manejando. ¿Qué utilidad tiene contar con estos datos georreferenciados?, ¿Qué indican estos datos?

b. ¿Qué aplicación GIS se está mostrando en esta página?

c. ¿Qué aplicación similar se le ocurre que podría ser trabajada como posible proyecto para la institución donde estudia?

5. Bibliografía

- PROVENZA Guiu José Luis. El sistema de posicionamiento global-4769019.pdf
Recuperado de: <https://dialnet.unirioja.es/descarga/articulo/4769019.pdf>
- International GPS Service. Recuperado de: <http://igs.cb.jpl.nasa.gov>
- International Earth Rotation Service. Recuperado de: <http://www.iers.org>
- International Terrestrial Reference Frame. Recuperado de: <http://itrf.ensg.ign.fr>

Elaborado por: Docente Ing. Jorge Correa

PRÁCTICA: INFORMACIÓN ESPACIAL Y DATOS GEOGRÁFICOS

SISTEMA DE INFORMACIÓN GEOGRÁFICA – (SIG)

1. Objetivos

- Proporcionar una introducción a los conceptos básicos sobre información espacial y datos geográficos.

2. Marco teórico

Los SIG multiplican al tiempo que facilitan las posibilidades de componer un mapa. Pero esta facilidad puede ser también una trampa si no tenemos unos conocimientos adecuados de composición cartográfica. Un mapa es un documento que nos tiene que hablar, ha de ser capaz de transmitir al usuario la información que la persona que lo realiza ha determinado. Para ello, ha de estar sujeto a las normas y convenciones del lenguaje cartográfico, es lo que denominamos sintaxis cartográfica. A continuación, vamos a realizar un breve repaso de aquellos elementos más relevantes en un mapa. Un mapa es una representación de la realidad y no la realidad misma. Para representar esa realidad deberemos de utilizar unas convenciones. En primer lugar, la realidad a representar es generalmente volumétrica y por lo tanto implica un cambio de tres dimensiones a dos. Este cambio de tres a dos dimensiones se suele suplir describiendo la tercera dimensión como un atributo.

3. Material y equipo

- Guía de laboratorio Información espacial y datos geográficos.
- Un computador por estudiante.
- Internet

4. Actividades, resultados y conclusiones

Para la realización de este taller deben ver, leer el vídeo documento: “Qué tiene de especial lo espacial?” disponible en el siguiente link:

<http://www.comunidadism.es/video/%C2%BFque-tiene-de-especial-lo-espacial> la plataforma Blackboard y realizar las siguientes actividades:

Defina con sus propias palabras ¿Qué considera usted que son datos espaciales?

- Muestre un ejemplo de un “mapa mental” relacionado con alguna actividad diaria que realicen.
- ¿Por qué es importante conocer sobre GIS?
- En el último párrafo de la pág. 10 del documento relacionado anteriormente hay dos referencias estos videos:
 - La revolución Geo-espacial
 - Layers of the world
- Se debe realizar un corto resumen de lo que entendieron de cada video

5. Bibliografía

- Domínguez Bravo Javier (2000). “Breve Introducción a la Cartografía y a los Sistemas de Información Geográfica (SIG)”.
- Sistema de GPS. Recuperado de: <http://www.trimble.com>
- International GPS Service. Recuperado de: <http://igscb.jpl.nasa.gov>
- International Earth Rotation Service. Recuperado de: <http://www.iers.org>
- International Terrestrial Reference Frame. Recuperado de: <http://itrf.ensg.ign.fr>

Elaborado por: Docente Ing. Jorge Correa

PRÁCTICA: SIG ON-LINE SISTEMA DE INFORMACIÓN GEOGRÁFICA – (SIG)

1. Objetivos

- Interactuar con un sistema de información geográfica OnLine.
- Familiarizarse con los datos de salida que genera un SIG.

2. Marco teórico

Si llevamos las capacidades SIG a un navegador, además de estas dispondremos en ese navegador de muchas otras posibilidades, tales como la representación de elementos multimedia (vídeo, sonido, etc.) o el uso de hipervínculos.

El navegador es hoy en día la aplicación versátil por excelencia, y ello hace que podamos añadir a las capacidades SIG una larga serie de otras funcionalidades no relacionadas directamente con la información geográfica, y no presentes en su mayoría en los SIG de escritorio.

3. Material y equipo

- Guía de laboratorio Conceptos básicos SIG I
- Un computador por cadete
- Internet

4. Actividades, resultados y conclusiones

Desde su navegador web ingresar a la siguiente página y responda las preguntas que se relacionan a continuación:

http://map.gpgis.com/gisapplications_basic/common_project/web/main_page.aspx

- Analizar qué datos importantes se están manejando.
- Identifiquen las capas (layers) que están disponibles.
- ¿Qué tipo de coordenadas se muestran en este sistema?
- Habiliten dichos layers y realicen una descripción de cada uno de ellos.
- Con la herramienta de selección de información, doble clic a diferentes polígonos. Describa la información mostrada.
- Ingrese a google street view. ¿Qué se muestra allí?
- Haga uso de la herramienta de consulta y realicen algunas consultas de ejemplo. ¿Qué resultados son mostrados?, ¿Cómo se muestran? Por ejemplo, consulte la parcela o lote No. 11745
- ¿Qué aplicación GIS se está mostrando en esta página?
- ¿Qué aplicación similar se le ocurre que podría ser trabajada como posible proyecto para la escuela?

5. Bibliografía

- MORALES, AURELIO. (2016). “Siete ventajas del web Mapping sobre SIG de escritorio”. Recuperado de: <http://mappinggis.com/2016/03/7-ventajas-las-tecnologias-web-mapping-los-sig-escritorio/>
- Sistema de Posicionamiento global - GPS.
Recuperado de: <http://www.trimble.com>

Elaborado por: Docente Ing. Jorge Correa

PRÁCTICA: GIS PARA INFORMACIÓN ORDEN PÚBLICO SISTEMA DE INFORMACIÓN GEOGRÁFICA – (SIG)

1. Objetivos

- Diseñar un bosquejo de un SIG para información de Orden Público.

2. Marco Teórico

Las organizaciones encargadas de la Defensa Nacional trabajan con Sistemas de Información Geográfica integrados en el resto de la organización, ya que gracias a los enfoques corporativos comparten recursos, datos y aplicaciones. En resumen, los sistemas están siendo reemplazados por las aplicaciones integradas de usos múltiples, en una arquitectura orientada a servicios.

Con un SIG corporativo las organizaciones de Defensa e inteligencia pueden:

- Proporcionar una infraestructura común para construir y desplegar soluciones SIG.
- Ampliar las capacidades geoespaciales a usuarios que normalmente no trabajaban con información geográfica.
- Mejorar las capacidades de los sistemas corporativos, aprovechando la ventaja del análisis geoespacial.
- Usar del SIG en todos los niveles de la organización y así incrementar la eficiencia operativa global.

3. Material y equipo

- Guía de laboratorio GIS para información de orden público.
- Un computador por estudiante.
- Internet

4. Actividades, resultados y conclusiones

- a) Defina en sus propias palabras qué es un GIS.

- b) ¿Cuáles son los componentes de un GIS?

- c) Enumere 5 preguntas que puede resolver un SIG.

- d) Qué escala muestra más detalle 1:5000, 1:10000 o 1:25000? ¿Por qué?

5. Bibliografía

- SIG Corporativo. Esri España. Recuperado de:
<http://www.esri.es/es/soluciones/defensa-e-inteligencia/sig-corporativo/>
- Sistema de Posicionamiento global - GPS.
Recuperado de: <http://www.trimble.com>
- International GPS Service. Recuperado de: <http://igscb.jpl.nasa.gov>
- International Earth Rotation Service. Recuperado de: <http://www.iers.org>
- International Terrestrial Reference Frame. Recuperado de. <http://itrf.ensg.ign.fr>

PRÁCTICA: SENSORES REMOTOS SISTEMA DE INFORMACIÓN GEOGRÁFICA – (SIG)

1. Objetivos

- Familiarizar al estudiante con los datos referentes a los satélites dedicados a percepción remota y sus principales características.
- Recrear los conceptos explicados en clase, así como anotar los puntos claves de la teoría en cuestión.
- Entender la naturaleza de una imagen digital y conocer las diferencias que existen entre las imágenes de diversos programas satelitales.

2. Marco Teórico

La percepción remota se define como la ciencia y arte de obtener información útil sobre los objetos, áreas o fenómenos bajo investigación mediante análisis de datos adquiridos por dispositivos que no están en contacto físico con estos. La percepción remota involucra varios elementos:

- Fuente de energía o iluminación que provea energía electromagnética al objeto de interés.
- Atmósfera, ya que la energía interactúa con este elemento al viajar del sensor al objeto y viceversa.
- Objetos, dependiendo de sus propiedades es la interacción o respuesta a la energía recibida y reflejada.
- Sensor remoto, que recoge y graba la radiación electromagnética reflejada o emitida por el objeto y la atmósfera. Este instrumento se monta en una plataforma llamada satélite.
- Transmisión, recepción y procesamiento. La energía grabada por el sensor se transmite a una estación receptora, en donde los datos se procesan y son convertidos a imágenes digitales.
- Interpretación y análisis, consiste en interpretar la imagen para extraer la información de los objetos captados.
- Usuario final que le da una aplicación a la información extraída de las imágenes para un mejor conocimiento de los objetos de interés.

3. Material y equipo

- Guía de laboratorio Taller de sensores remotos SIG II
- Un computador por estudiante
- Internet
- Sistema de Información geográfica

4. Actividades, resultados y conclusiones

En la base de datos de la Agencia Espacial Europea <http://database.eohandbook.com/database/instrumenttable.aspx>

- Busque las características del sensor Ali y el sensor Hyperion.
- Resolución espacial, espectral, radiométrica.
- Bandas espectrales que maneja (longitudes de onda), Ancho de escena (swath), periodo de revisita.

a. Ingrese al sitio web Geospatial Data Service Centre de Holanda y responda las siguientes preguntas:

<http://gdsc.nlr.nl/FlexCatalog/catalog.html#>

a) ¿Cuántos satélites de observación operativos posee Argentina?

b) ¿Cuántos satélites operativos tiene España?

c) ¿Cuántos satélites ópticos de 1000 metros de resolución tiene Canadá?

a) ¿Cuántos satélites operativos de 2 metros de resolución tiene Taiwan?

b) ¿Qué satélites se deberían poner en funcionamiento en 2012?

c) ¿Qué países hacen parte de la constelación DMC?

d) ¿Qué satélites europeos que captan longitudes de onda Azul del visible son más recomendables para hacer cartografía a 1:25?000?

e) ¿Qué satélites de radar en banda L sirven para hacer cartografía a 1:25?000?

f) ¿Qué satélites de resolución menor o igual a 1 m toman en infrarrojo cercano? (800 a 900 nanómetros)

b. A continuación, diligencie la siguiente tabla:

Misión	Nombre de la Plataforma	Nombre del sensor	Bandas	Swath	País	Revisita
SPOT						
WorldView						
EROS-C						
RapidEye						
Pleiades						
Landsat 7 ETM+						
MODIS						

5. Bibliografía

Posada, Elena, IGAC, & Ramírez Héctor. (2011). Manual de prácticas de percepción remota con el programa ERDAS Parte 1 IMAGINE.

Recuperado de: http://www.un-spider.org/sites/default/files/ManualERDAS_web.pdf

Elaborado por: Docente Ing. Jorge Correa

PRÁCTICA: SENSORES REMOTOS II SISTEMA DE INFORMACIÓN GEOGRÁFICA – (SIG)

1. Objetivos

- Familiarizar al estudiante con el análisis e interpretación de objetos a partir de una imagen satelital.
- Identificar estructuras, objetos, accidentes geográficos y cobertura a partir de una imagen satelital en falso color.
- Familiarizar al estudiante con la composición de imágenes satelitales a partir de sus bandas.
- Reconocer los diferentes tipos de sensores.

2. Marco Teórico

Los sensores remotos captan información de los objetos sin que para ello deban tener un contacto directo con estos, utilizado generalmente en aplicaciones de recursos naturales, porque brindan información confiable sobre superficies extensas con alta precisión. Los sensores remotos son una importante herramienta para la construcción de mapas, planos, estudios urbanos, interpretación de recursos y planificación territorial.

3. Material y equipo

- Guía de laboratorio
- Un computador por estudiante
- Internet
- Base de datos online de plataformas satelitales.

4. Actividades, resultados y conclusiones

4.1 A partir de la siguiente imagen deben identificar las estructuras y objetos que sean familiares (ríos, carreteras, cultivos, puentes, edificaciones, lagos, lagunas, suelo seco, suelo húmedo, aeropuertos, nubes, etc.)



4.2 ¿Por qué existen algunas zonas en color rojo?

4.3 ¿Qué puede indicar el color rojo a la orilla?

4.4 ¿En la parte superior izquierda hay un río, sin embargo, se muestra en color rojo, por qué cree que sucede esto?

4.5 Por qué en los otros ríos éstos se ven negros?

4.6 Por qué los sensores hiperespectrales tienen más alta resolución espectral que los demás sensores?

Señale según el uso:

- | | | |
|------------------------|-----------------------|--|
| a). Infrarrojo Medio | <input type="radio"/> | Detecta el calor emitido por la superficie terrestre |
| b). Infrarrojo Cercano | <input type="radio"/> | Estimar humedad en vegetación |
| c). Infrarrojo Lejano | <input type="radio"/> | Discriminar vegetación |

5. Bibliografía

Posada, Elena, IGAC, & Ramírez Héctor. (2011). Manual de prácticas de percepción remota con el programa ERDAS Parte 1 IMAGINE.

Recuperado de:

http://www.un-spider.org/sites/default/files/ManualERDAS_web.pdf

- ARONOFF, S. (2005). Remote Sensing for Gis Managers. En S. Aronoff, Remote Sensing for gis managers. California: ESRI.
- Centre for Remote Imaging Sensing and Processing. What is Remote Sensing? Disponible en: <http://www.crisp.nus.edu.sg/~research/tutorial/intro.htm>
- INSTITUTO GEOGRÁFICO AGUSTÍN CODAZZI. Interpretación visual de imágenes de sensores remotos y su aplicación en levantamientos de cobertura y uso de la tierra. Centro de Investigación y Desarrollo de Información Geográfica, CIAF, 2005

Elaborado por: Docente Ing. Jorge Correa

PRÁCTICA: VIEW AND DOWNLOAD SATELLITE IMAGERY SISTEMA DE INFORMACION GEOGRÁFICA – (SIG)

1. Objetivos

- Show the use of visors for imaging an area of interest.
- Use viewers to download satellite images.
- Acquaint the student with the images viewers.

2. Background / Scenario

A satellite photo or satellite image can be defined as the visual representation of the information captured by a sensor mounted on an artificial satellite . These sensors collect information reflected by the surface of the Earth which is then sent back to it and conveniently processed, delivers valuable information on the characteristics of the area represented.

3. Required Resources

- Laboratory Guide
- One computer per student
- Internet
- Online database of satellite platforms.

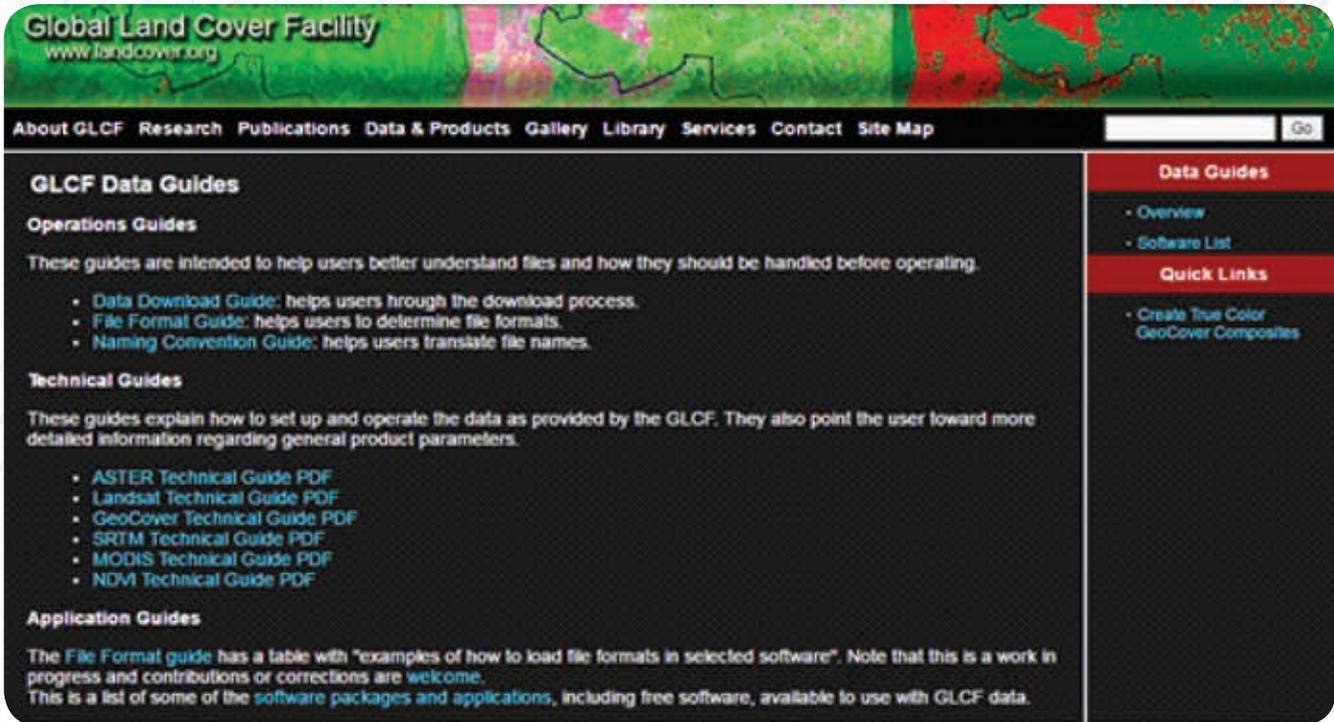
4. Activities, results and conclusions

Below are guidelines for navigating three satellite image web archives.

4.1 Global Land Cover Facilities (GLCF) – <http://landcover.org/>

To see the list of available satellite image data and derived raster products, click the *Data & Products* tab at the top of the page.

A *Data Guides* link on the left of the page leads to technical details about the file formats, as well as Landsat, GeoCover and SRTM technical guides.



To search for available Landsat Data for your area of interest click the Landsat link that can be found on the Data & Products page. The link leads to an overview of the data, as well as links to both a web interface and FTP interface for locating and downloading Landsat imagery.

Use the Landsat in web interface link to look for imagery based on a map search. Landsat images are referenced with a Path/Row number via the World Reference System 2 WRS.

Once you know the reference number for your area of interest, you can search for imagery based on the Path/Row.

On the map search page, you will see a world map, you can choose the type of imagery you are interested in finding on the left, and click the Update Map button to see the areas that are available in red.

Use the magnifying glass tool on the map to zoom in to your area of interest. Note that there are tabs above the map for searching based on Path/Row, Lat/Long, Place Name, Draw (trace a search area polygon), and Map Layers (select other map layers to be viewed).

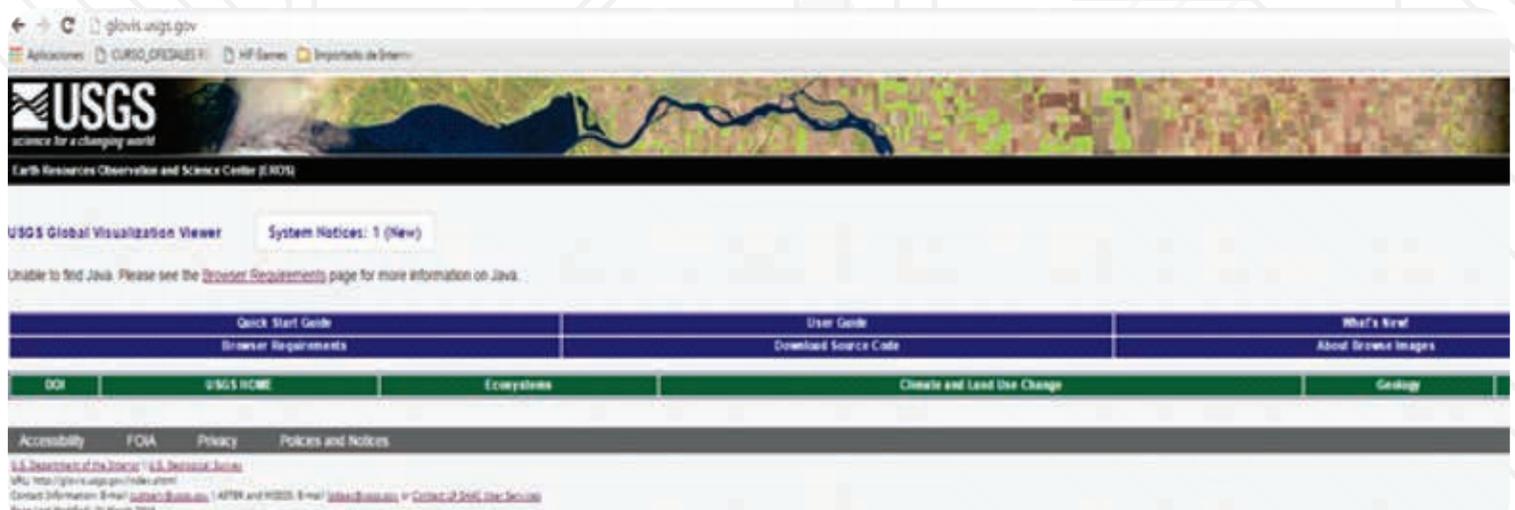
Once you have identified your area of interest, use the select tool (arrow with plus sign) to select the tile. You will see the number of images available below the map. Click Preview and Download to see a preview image and additional information. You will need a compression utility to unzip the files once downloaded.

Once you have the Path/Row number of your area of interest, you can use the FTP server (<ftp://ftp.glcg.umiacs.umd.edu/glcg/Landsat/>). There may be images available on the FTP server that are not displayed on the web interface.

4.2 Global Visualization Viewer (GLOVIS) – <http://glovis.usgs.gov/>

You will see a similar map interface on the GLOVIS web site. There is a Select Sensor drop down menu where you will find the available data sets. Click the *About Browse Images* button to read more details about the available imagery.

Choose a latitude/longitude location, or simply click on the map to view the available imagery. Once you are in the viewer, you can change the *Sensor*, *Resolution* and *Map Layers* using the menu bar above the map.



For Landsat images, you should see nine images in an interactive mosaic in the viewer. You can click on any image to bring it to the top. To scan the different dates for this image, click the *Previous Scene* and *Next Scene* buttons on the left. You can also choose to exclude images based on the amount of cloud cover with the *Max Cloud* option.

Right clicking on an image brings up additional options such as viewing metadata, a preview image. Choose *add scene to list* for ordering. Clicking Order will bring you to the appropriate USGS web site for purchasing the image. You can either have disks mailed to you, or download via FTP.

4.3 Earth Explorer – <http://earthexplorer.usgs.gov>

Choose Guest to enter the Data Set Selection menu, or register as a user for additional options such as saving search results.

In the Data Set Selection area, you will find a list of the available data sources, as well as Related Links to additional information. Choose the data set that you would like to search. In the Spatial Coverage box you can define your area of interest on a map, using coordinates, or a place name. Click the Continue button to advance.

You can choose the (Additional Search Criteria...) link to limit the search by Path/Row, cloud cover, entity ID, day or night, or data classification for Landsat. You can also limit the search by choosing a date range in the Acquisition Date box. Choose a large number for records returned in order to view the complete range of available data. Click Search and wait for the results.

5. Bibliography

- Elena Posada, IGAC Héctor Mauricio Ramírez Daza, IGAC Norma Carolina Espejo Delgado, IGAC. Manual de prácticas de percepción remota con el programa ERDAS Parte 1 IMAGINE 201. Recuperado de: http://www.un-spider.org/sites/default/files/ManualERDAS_web.pdf
- ARONOFF, S. (2005). Remote Sensing for Gis Managers. En S. Aronoff, Remote Sensing for gis managers. California: ESRI Press.
- Centre for Remote Imaging Sensing and Processing. What is Remote Sensing? Disponible en: <http://www.crisp.nus.edu.sg/~research/tutorial/intro.htm>
- INSTITUTO GEOGRÁFICO AGUSTÍN CODAZZI. Interpretación visual de imágenes de sensores remotos y su aplicación en levantamientos de cobertura y uso de la tierra. Centro de Investigación y Desarrollo de Información Geográfica, CIAF, 2005.

Elaborado por: Docente Ing. Jorge Correa

PRÁCTICA: TELEDETECCIÓN ESPACIAL SISTEMA DE INFORMACIÓN GEOGRÁFICA – (SIG)

1. Objetivos

- Familiarizar al estudiante con la identificación de objetos a partir de una imagen satelital de manera visual.
- Familiarizar a los estudiantes con los diferentes tipos de imágenes satelitales proporcionadas por diferentes plataformas.
- Orientar al estudiante con la identificación de objetos y cobertura usando imágenes en falso color.

2. Marco Teórico

La teledetección o detección remota es la adquisición de información a pequeña o gran escala de un objeto o fenómeno, consiste en recoger información a través de diferentes dispositivos de un objeto concreto o un área. Actualmente, el término se refiere al uso de tecnologías de sensores para adquisición de imágenes, incluyendo: instrumentos a bordo de satélites o aerotransportados, usos en electrofisiología, y difiere en otros campos relacionados con imágenes como por ejemplo en imagen médica.

3. Material y equipo

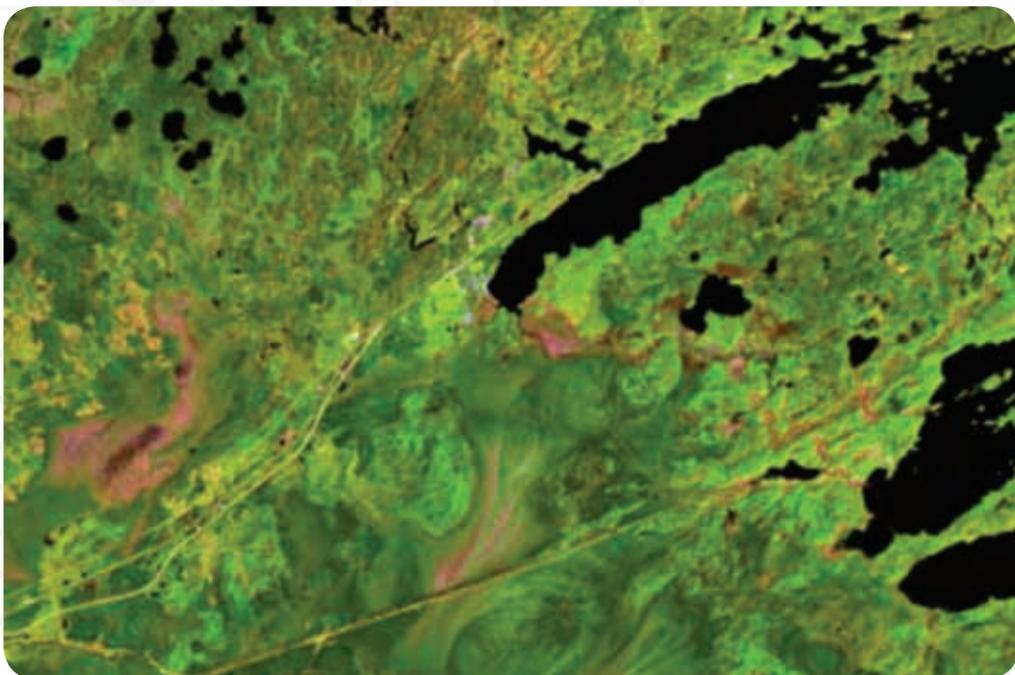
- Guía de laboratorio
- Un computador por estudiante
- Internet

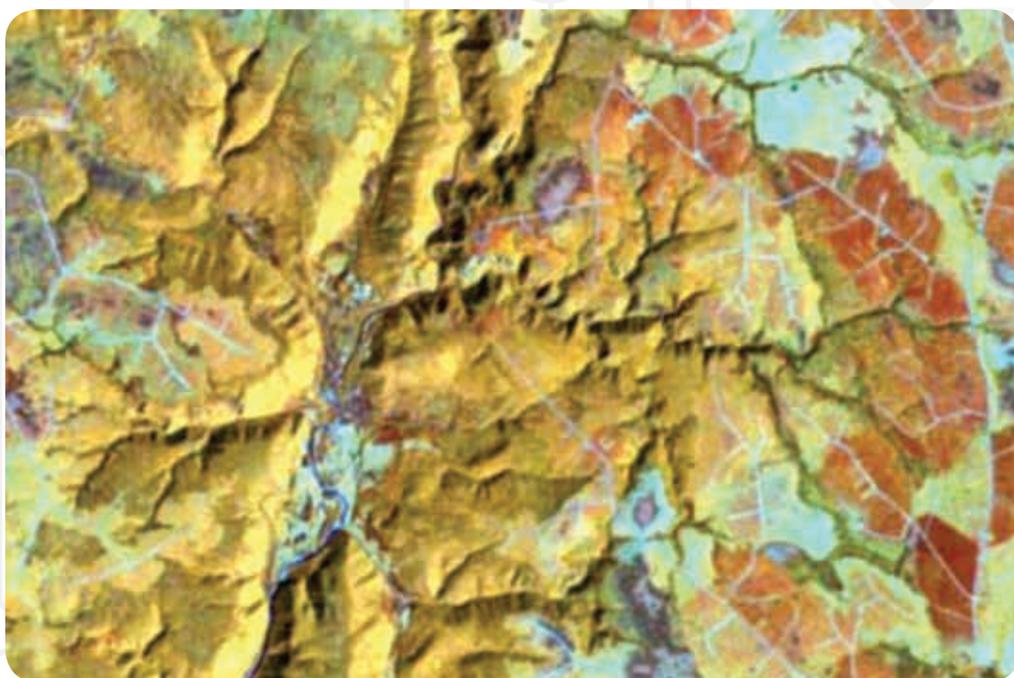
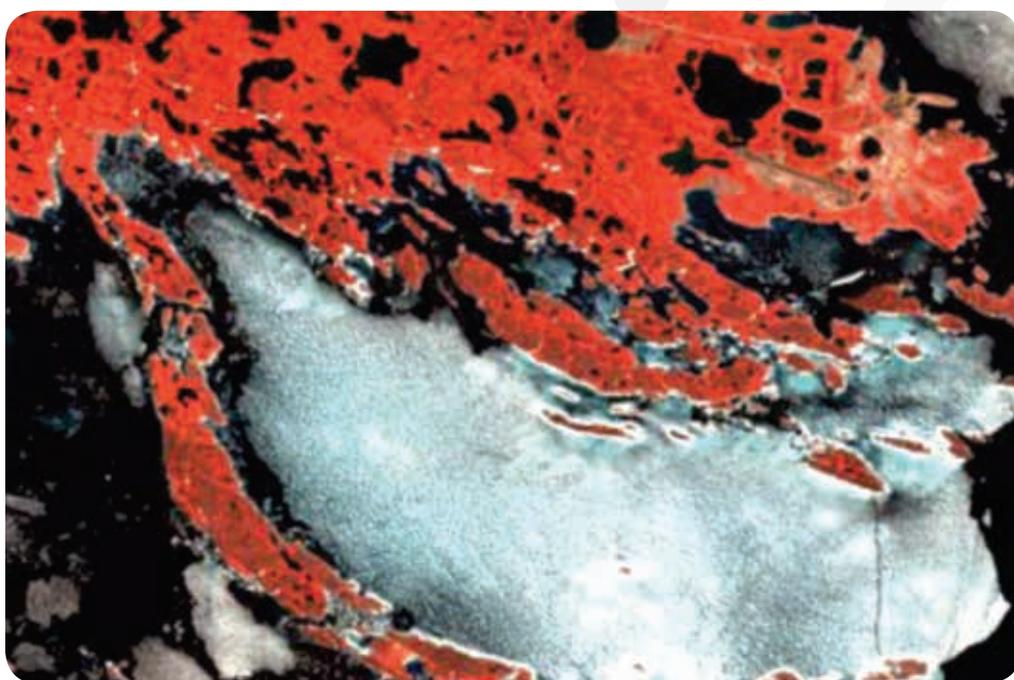
4. Actividades, resultados y conclusiones

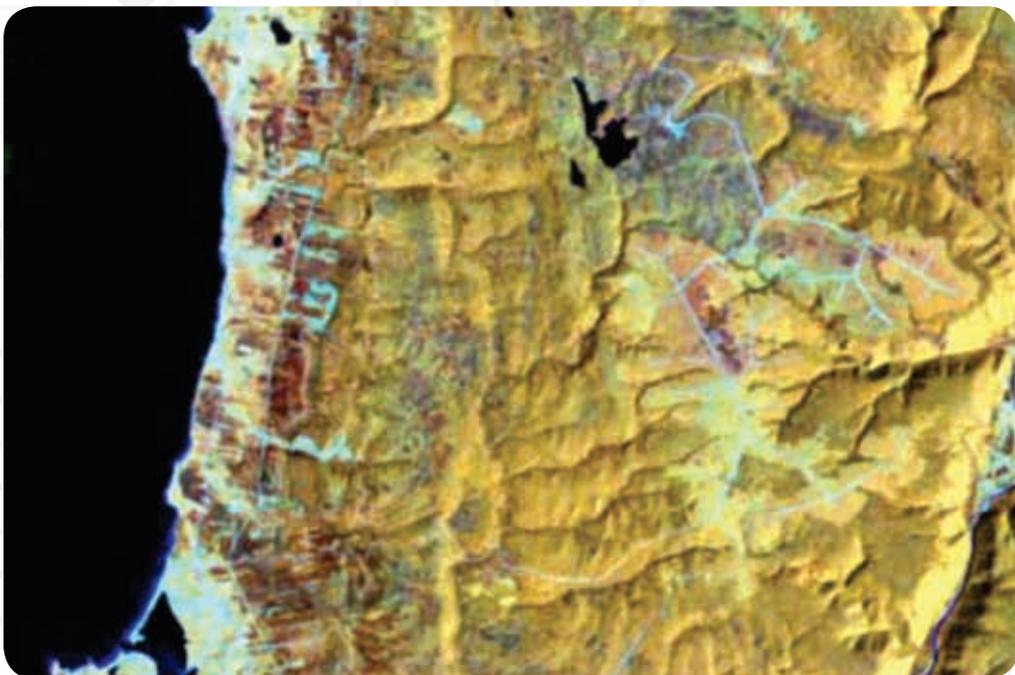
A continuación, usted encontrará 12 porciones de imágenes satelitales. La idea es encontrar la imagen que contiene las características descritas seguidamente. Entre el número de la imagen frente a cada descripción. Tener en cuenta que algunas imágenes se usan más de una vez.

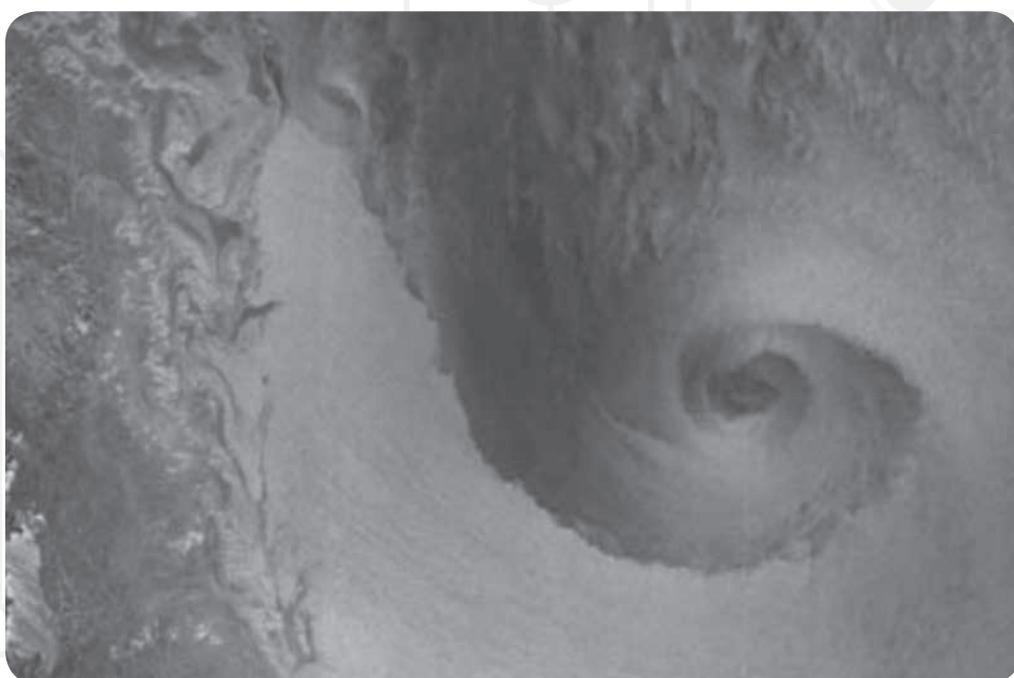
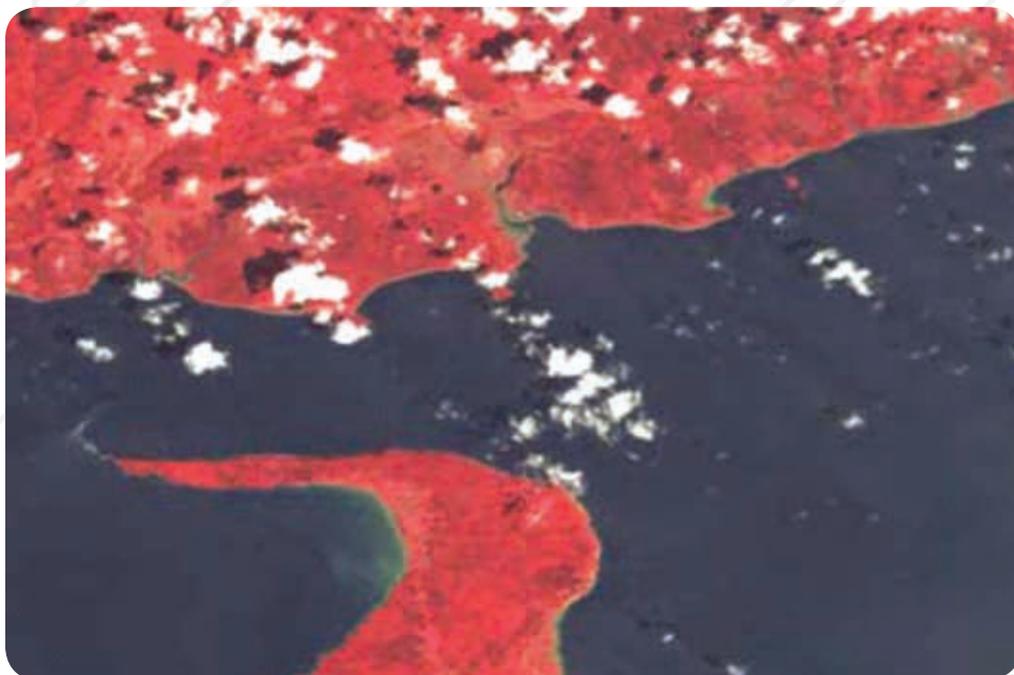
Características:

- Dos pistas de carrera una más pequeña dentro de otra. Algunos caminos y dos canchas de golf se ven también en la imagen.
- Un aeropuerto en los límites de la ciudad. Se puede además ver un pequeño río junto a otro más grande en la imagen.
- El efecto de un huracán en la superficie del océano.
- Una playa que muestra una ciudad costera y un rompeolas donde pueden atracar los botes.
- Cinco puentes sobre un río. Tres de ellos pasan sobre islas en dicho río.
- Muchas nubes y sus sombras sobre la tierra y el agua.
- Un área costera mostrando flujos de hielo en la más grande de sus bahías. La parte terrestre muestra varios lagos. Además, se ven algunos islotes.
- Una península en forma de anzuelo.
- Una línea costera escarpada mostrando cuevas y bahías.
- Una granja cerca de dos ríos. También se pueden apreciar varios caminos que conducen hacia un pueblo.
- Muchos lagos en un área rocosa y escarpada. Una sección que corresponde a pantano.
- Una región boscosa que muestra áreas taladas y caminos de acceso a dicha región.
- La desembocadura de un río y la capa de sedimento arrastrada hacia el mar.
- Un área con una variedad extensa de cultivos.









5. Bibliografía

- Elena Posada, IGAC Héctor Mauricio Ramírez Daza, IGAC Norma Carolina Espejo Delgado, IGAC. (2011). Manual de prácticas de percepción remota con el programa ERDAS Parte 1 IMAGINE 2011.

Recuperado de:

http://www.un-spider.org/sites/default/files/ManualERDAS_web.pdf

- ARONOFF, S. (2005). Remote Sensing for Gis Managers. En S. Aronoff, Remote Sensing for gis managers. California: ESRI Press.
- Centre for Remote Imaging Sensing and Processing. What is Remote Sensing?
- Disponible en: <http://www.crisp.nus.edu.sg/~research/tutorial/intro.htm>
- INSTITUTO GEOGRÁFICO AGUSTÍN CODAZZI. Interpretación visual de imágenes de sensores remotos y su aplicación en levantamientos de cobertura y uso de la tierra. Centro de Investigación y Desarrollo de Información Geográfica, CIAF, 2005

Elaborado por: Docente Ing. Jorge Correa



2.

ESTRUCTURA DE DATOS

A través de las prácticas de laboratorio de *Estructura de Datos* los estudiantes podrán comprender la funcionalidad y las ventajas que representan dichas estructuras, las cuales permiten manejar grandes volúmenes de información, de manera eficiente.

Las estructuras de datos son clave para diseñar algoritmos eficientes. Por esta razón es importante que la agrupación de datos tenga un cierto tipo de estructura y organización interna, porque esto favorece al momento de la programación e influye en el algoritmo que se va a usar para resolver un determinado problema.

PRÁCTICA: ESTRUCTURA CONTENEDORA DE TAMAÑO FIJO ARREGLOS UNIDIMENSIONALES ASIGNATURA: ESTRUCTURA DE DATOS

1. Objetivos:

- Reconocer y utilizar las estructuras contenedoras de tamaño fijo – Arreglo Unidimensionales como objetos, los cuales se pueden modelar en el mundo real, en la representación de una secuencia de valores finitos y homogéneos. Agregar PDU simples en modo de tiempo real.
- Utilizar las estructuras cíclicas para manipular las estructuras contenedoras, reconociendo que estas se pueden aplicar en otro tipo de soluciones del mundo real.
- Crear un proyecto completo en Java, utilizando el Ambiente de Desarrollo Eclipse.
- Crear paquetes que simulen la estructura de todo objeto en el mundo real.
- Crear clases en cada paquete que modelen la propiedad de modularidad de la Programación Orientada a objetos.

2. Marco Teórico

Una Estructura Contenedora de Tamaño Fijo – Arreglos Unidimensionales, es un espacio de memoria, donde se puede almacenar un conjunto de datos finitos y homogéneos.

Este espacio de memoria es estático, lo cual significa que en tiempo de ejecución no se puede disminuir o ampliar la capacidad de los datos. El espacio de memoria es referenciado por un dato definido en tiempo de compilación.

Para gestionar una Estructura Contenedora de Tamaño Fijo – Arreglos Unidimensionales, se requiere del uso y aplicación de las estructuras cíclicas (do-while, while, o for).

3. Material y Equipo

- Guía de laboratorio
- Computador
- Ambiente de Desarrollo Eclipse
- JDK Java
- Documentación de la plataforma Java

4. Actividades, Resultados y Conclusiones

En el presente laboratorio se abordará la actividad de crear paquetes, clases, y Estructura Contenedora de Tamaño Fijo – Arreglos Unidimensionales, para su manipulación de datos.

Paso 1. Crear un proyecto en Eclipse con su nombre completo y código. Dentro del proyecto se deben crear dos paquetes. El primer paquete con el nombre de mundo del problema. El segundo paquete con el nombre interfaz del usuario.

El path de cada paquete debe ser:

piinf.g90d.estructuradatos.mundodelproblema

piinf.g90d.estructuradatos.interfazDelUsuario

Paso 2. En el paquete mundoDelProblema crear una clase llamada ArregloUnidimensional.

En dicha clase, crear un arreglo unidimensional de N (donde N = al valor que usted desee). Además, crear los métodos necesarios de construcción, consulta y modificación.

Paso 3. En el paquete interfazDelUsuario crear la clase principal la cual debe llevar como nombre el nombre del proyecto. Dentro de la clase principal, crear el método principal main(). Dentro del método principal main(), desarrollar todo el código necesario que permita:

Crear un primer objeto instanciador con el código_1.

Crear un primer objeto instanciador con el código_2.

Cargar el arreglo de cada objeto con números aleatorios, utilizando la clase Random del paquete java.util. (Las características de la Clase Random y sus métodos se pueden consultar en la siguiente URL:

<http://docs.oracle.com/javase/7/docs/api/>)

Crear un tercer objeto instanciador con el código_3.

Paso 4. En el método main(), cargar el tercer arreglo con todos los elementos que se encuentren en el primer arreglo del objeto1 y que no se encuentren en el arreglo del objeto2

Paso 5. Cree un menú de opciones en el método principal que permita:

1. Crear arreglo 1.
2. Crear arreglo 2.
3. Crear arreglo 3.
4. Mostrar los tres arreglos
5. Salir

Conclusiones: _____

5. Bibliografía

- Oracle. (04 de 02 de 2015). <http://docs.oracle.com/javase/7/docs/api/>. Obtenido de: <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
- Villalobos, J. A., & Casallas G., R. (2006). *Fundamentos de Programación. Aprendizaje Activo Basado en Casos*. Bogotá - Colombia: Pearson Prentice Hall.

Elaborado por: Docente Humberto Amaya

PRÁCTICA: ESTRUCTURA CONTENEDORA DE TAMAÑO FIJO CADENA DE CARACTERES – STRING

1. Objetivos:

- Reconocer y utilizar las estructuras contenedoras de tamaño fijo – Cadena de Caracteres como objetos, los cuales se pueden modelar en el mundo real, en la representación de una secuencia de caracteres finitos y homogéneos.
- Utilizar las estructuras cíclicas para manipular las estructuras contenedoras, reconociendo que estas se pueden aplicar en otro tipo de soluciones del mundo real.
- Crear un proyecto completo en Java, utilizando el Ambiente de Desarrollo Eclipse.
- Crear paquetes que simulen la estructura de todo objeto en el mundo real.
- Crear clases en cada paquete que modelen la propiedad de modularidad de la programación Orientada a Objetos.

2. Marco Teórico

Una Estructura Contenedora de Tamaño Fijo – Cadena de Caracteres, presenta la misma estructura de composición de un Arreglo Unidimensional, la cual se puede conceptuar como un espacio de memoria, donde se almacena un conjunto de datos finitos y homogéneos, donde el compilador la finaliza con un carácter especial null o '\0'.

Este espacio de memoria es estático, lo cual significa que en tiempo de ejecución se genera la cadena, no permitiendo posteriormente realizar algún cambio a la estructura. Solo se puede simular que la variable referente apunte a otro espacio cadena de mayor o menor tamaño. El espacio de memoria es referenciado por un dato definido en tiempo de compilación o en tiempo de ejecución.

Para gestionar una Estructura Contenedora de Tamaño Fijo – Cadena de Caracteres, se requiere del uso y aplicación de las estructuras cíclicas (do-while, while, o for).

3. Materia y Equipo

- Guía de laboratorio
- Computador
- Ambiente de Desarrollo Eclipse
- JDK Java
- Documentación de la plataforma Java

4. Actividades, Resultados y Conclusiones

En el presente laboratorio, de acuerdo con los anteriores laboratorios, se abordará la manera de crear paquetes, clases, y Estructura Contenedora de Tamaño Fijo – Cadena de Caracteres, para su manipulación de datos.

Paso 1. Crear un proyecto en Eclipse con su nombre completo y código. Dentro del proyecto se debe crear dos paquetes. El primer paquete con el nombre de mundoDelProblema. El segundo paquete con el nombre interfazDelUsuario.

El path de cada paquete debe ser:

emavi.piinf.g90d.estructuradatos.mundodelproblema emavi.piinf.g90d.estructuradatos.interfazDelUsuario

Paso 2

En el paquete mundoDelProblema crear una clase llamada CadenaDeCaracteres, en la cual se creará una cadena de caracteres, por medio del parámetro recibido en el método Constructor.

Paso 3

En la clase **CadenaDeCaracteres** crear un método que retorne la longitud de la cadena de caracteres.

Paso 4

En la clase **CadenaDeCaracteres** crear un método que retorne el número de vocales que contiene la cadena de caracteres.

Paso 5

En la clase **CadenaDeCaracteres** crear un método que retorne la cadena convertida en MAYÚSCULAS.

Paso 6

En la clase **CadenaDeCaracteres** crear un método que retorne la cadena, donde las vocales sean MAYÚSCULAS y los demás caracteres minúsculas.

Paso 7

En la clase **CadenaDeCaracteres** crear un método que retorne la cadena al revés.

Paso 8

En la clase **CadenaDeCaracteres** crear un método que retorne la cadena + el nombre suyo.

Paso 9

En el paquete **interfazDelUsuario** crear la clase principal la cual debe llevar como nombre el nombre del proyecto. Dentro de la clase principal, crear el método principal **main()**. Dentro del método principal **main()**, desarrollar todo el código necesario que pueda mostrar por medio de un menú, las respuestas de las preguntas: 2, 3, 4, 5, 6, 7, y 8.

Paso 10

Una vez que compile y se ejecute normalmente, se debe comprimir (punto zip) el proyecto y subirlo a la plataforma, como evidencia de que usted realiza el presente laboratorio.

Conclusiones: _____

5. Bibliografía

- Oracle. (04 de 02 de 2015).<http://docs.oracle.com/javase/7/docs/api/>. Obtenido de <http://www.oracle.com/technetwork/java/javase/downloads/inde.html>
- Villalobos, J. A., & Casallas G., R. (2006). Fundamentos de Programación. Aprendizaje Activo Basado en Casos. Bogotá - Colombia: Pearson Prentice Hall. Tabla de Código AscII. Recuperado de <http://ascii.cl/es/>

Elaborado por: Docente Humberto Amaya

PRÁCTICA: ESTRUCTURA CONTENEDORA DE TAMAÑO FIJO ARREGLOS BIDIMENSIONALES ESTRUCTURA DE DATOS

1. Objetivos:

- Reconocer y utilizar las estructuras contenedoras de tamaño fijo – Arreglo Bidimensionales como objetos, los cuales se pueden modelar en el mundo real, en la representación de una secuencia de valores finitos y homogéneos, entre coordenadas de filas y columnas.
- Utilizar las estructuras cíclicas para manipular las estructuras contenedoras, reconociendo que estas se pueden aplicar en otro tipo de soluciones del mundo real.
- Crear un proyecto completo en Java, utilizando el Ambiente de Desarrollo Eclipse.
- Crear paquetes que simulen la estructura de todo objeto en el mundo real.
- Crear clases en cada paquete que modelen la propiedad de modularidad de la Programación Orientada a Objetos

2. Marco Teórico

Una Estructura Contenedora de Tamaño Fijo – Arreglos Bidimensionales, es un espacio de memoria, donde se puede almacenar un conjunto de datos finitos y homogéneos.

Este espacio de memoria es estático, lo cual significa que en tiempo de ejecución no se puede disminuir o ampliar la capacidad de los datos. El espacio de memoria es referenciado por un dato definido en tiempo de compilación, donde los elementos se encuentran ubicados en el juego de coordenadas de filas x columna.

Para gestionar una Estructura Contenedora de Tamaño Fijo – Arreglos Bidimensionales, se requiere del uso y aplicación de las estructuras cíclicas (do-while, while, o for), con un patrón de doble recorrido.

3. Material y Equipos

- Guía de laboratorio
- Computador
- Ambiente de Desarrollo Eclipse
- JDK Java
- Documentación de la plataforma Java

4. Actividades, Resultados y Conclusiones

En esta práctica de laboratorio se abordará la manera de crear paquetes, clases, y Estructura Contenedora de Tamaño Fijo – Arreglos Bidimensionales, para su manipulación de datos.

Es importante que tenga en cuenta este cuadro comparativo, para la creación de una estructura estática – Arreglo Bidimensional:

Conceptos	Arreglo Unidimensional	Arreglo Bidimensional
Definición de características	Private static int DIM = XX; Private [] int arrU;	Private static int DIMF = XX; Private static int DIMC = XX; Private [] [] int arrB;
Método Constructor	arrU = new int [DIM];	arrB = new int [DIMF] [DIMC];
Métodos de Consulta	getLongitud() getArrU() getElemento(int pos)	getLongitud() getArrB() getElemento(int posF, int posC)
Métodos de Modificación	setCargarCeros() setElemento(int elemento, int pos)	setCargarCeros() setElemento(int elemento, int posF, int posC)

ESTUDIO DE CASO:

Un grupo de personal, decidieron a finales del año 2016 crear una Empresa Comercial de Dispositivos Móviles. Durante los últimos cinco meses del año 2017, han estado comercializando con cinco marcas básicas de celulares en tres ciudades de Colombia: Cali, Medellín y Bogotá.

Las marcas y precios son los siguientes:

	Huawei	\$278.000,00
	Samsung	\$935.000,00
	Sony	\$821.090,00
	LG	\$495.000,00
	Nokia	\$686.000,00

Sus ventas por unidades se ven reflejadas en los siguientes cuadros:

SUCURSAL					
	MES	MES	MES	MES	MES
Huawei	15	7	8	12	14
Samsung	11	5	3	7	5
Sony	10	3	16	9	17
LG	6	5	7	4	6
Nokia	7	19	10	6	12

SUCURSAL					
	MES	MES	MES	MES	MES
Huawei	7	13	22	1	4
Samsung	3	1	0	20	2
Sony	14	12	11	5	14
LG	3	4	2	1	3
Nokia	15	4	3	4	16

4.1. Paso 1

Crear un proyecto en Eclipse con su nombre completo y código. Dentro del proyecto se deben crear dos paquetes. El primer paquete con el nombre de mundoDelProblema. El segundo paquete con el nombre interfazDelUsuario.

SUCURSAL					
	MES	MES	MES	MES	MES
Huawei	4	22	3	5	2
Samsung	11	13	2	4	0
Sony	13	3	15	20	1
LG	5	17	2	11	2
Nokia	14	0	20	5	3

El path de cada paquete debe ser:

piinf.g90d.estructuradatos.mundoNombreEstudiante

piinf.g90d.estructuradatos. interfazNombreEstudiante

4.2. Paso 2

En el paquete *mundoNombreEstudiante* crear una clase llamada *ArregloBidimensional*. En dicha clase, desarrollar todos los métodos (constructor, consulta y modificación) de un arreglo bidimensional de una dimensión de $N \times N$, donde $N = 5$.

4.3. Paso 3

En el paquete *mundoNombreEstudiante* crear una clase llamada *EmpresaComercialDispositivosMoviles*. En dicha clase, crear como atributo tres objetos instanciadores (*sucursalCali*, *sucursalMedellin*, *sucursalBogota*) de tipo clase *ArregloBidimensional*

4.4. Paso 4

Con el fin de facilitar la cargada de los datos en el arreglo Bidimensional, se recomienda crear como atributo un arreglo unidimensional por cada sucursal, en la Clase *EmpresaComercialDispositivosMoviles*, de la siguiente forma:

```
public int [] arrCali = {15, 7,8, 12, 14, 11, 5, 3, 7, 5, 10, 3, 6, 9, 12, 6, 5, 7, 4, 6, 7, 9, 10, 6, 12} ;
```

```
public int [] arrMedellin = {7, 3, 2, 1, 4, 3, 1, 0, 2, 2, 4, 2, 1, 5, 4, 3, 4, 2, 1, 3, 5, 4, 3, 4, 6} ;
```

```
public int [] arrBogota = {4, 2, 3, 5, 2, 1, 3, 2, 4, 0, 3, 3, 5, 2, 1, 5, 7, 2, 1, 2, 4, 0, 2, 5, 3} ;
```

4.5. Paso 5

En la clase llamada *EmpresaComercialDispositivosMoviles*, crear tres métodos que permitan a su interior cargar cada arreglo bidimensional de acuerdo al objeto instanciador (*sucursalCali*, *sucursalMedellin*, *sucursalBogota*).

4.6. Paso 6

En la clase *EmpresaComercialDispositivosMoviles*, crear los métodos necesarios que den respuesta a cada uno de los siguientes requerimientos:

Se desea conocer:

- El valor total de las ventas de la empresa.
- Cuál ha sido el mes de mayor valor de venta en la empresa.
- Cuál ha sido el mes de menor valor de venta en la empresa.
- Cuál ha sido el celular que más se ha vendido en la empresa.
- Cuál ha sido el celular que menos se ha vendido en la empresa.
- Por cada sucursal, el celular y mes que más se vendió.
- Por cada sucursal, el celular y mes que menos se vendió.
- Por cada sucursal, el mes de mayor valor de venta.
- Por cada sucursal, el mes de menor valor de venta

4.7. Paso 7

En el paquete *interfazNombreEstudiante* crear las clases y métodos necesarios, que permitan diseñar un menú que muestre cada una de las respuestas de los requerimientos de pregunta 6.

Conclusiones: _____

5. Bibliografía

- Oracle. (04 de 02 de 2015). <http://docs.oracle.com/javase/7/docs/api/>. Obtenido de: <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
- Villalobos, J. A., & Casallas G., R. (2006). *Fundamentos de Programacion. Aprendizaje Activo Basado en Casos*. Bogota - Colombia: Pearson Prentice Hall

Elaborado por: Docente Humberto Amaya

PRÁCTICA: LISTA SENCILLAMENTE ENLAZADA ESTRUCTURA DE DATOS

1. Objetivos:

- Reconocer y utilizar las estructuras contenedoras de tamaño variable – Lista Sencillamente Enlazada, las cuales se pueden modelar en el mundo real, en la representación de una secuencia de datos finitos.
- Reconocer y utilizar el concepto Nodo, como la estructura básica para la composición de una estructura de datos.
- Crear un proyecto completo en Java, utilizando el Ambiente de Desarrollo Eclipse.
- Crear paquetes que simulen la estructura de todo objeto en el mundo real.
- Crear clases en cada paquete que modelen la propiedad de encapsulamiento, modularidad y jerarquía de clases de la Programación Orientada a Objetos.

2. Marco Teórico

Una Estructura Contenedora de Tamaño Variable – Estructura de Datos Lista Sencillamente Enlazada, es un espacio de memoria, donde se puede almacenar un conjunto de datos lineales, finitos y homogéneos.

Este espacio de memoria es dinámico, lo cual significa que en tiempo de ejecución se an creando o eliminando los espacios necesarios para almacenar los datos simples o tipo Clase. El espacio de memoria es referenciado por un dato de tipo Nodo llamado Cabecera, definido en tiempo de compilación.

Básicamente la gestión de una Estructura de Datos: Lista Sencillamente Enlazada, está dada por las siguientes operaciones:

- Crear Lista.
- Determinar si la Lista está Vacía.
- Insertar Datos en la Lista al Comienzo y al Final
- Eliminar Datos de la Lista al Comienzo o Final.
- Listar o Mostrar los datos de la Lista.

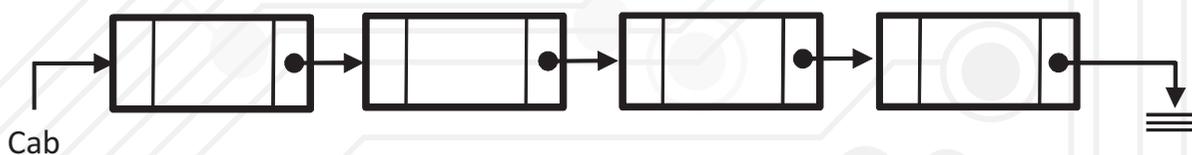
Para gestionar una Estructura de Datos Lista Sencillamente Enlazada, se requiere previamente del conocimiento de:

- Paradigma Orientado a Objetos: Modularidad, Encapsulamiento, y Jerarquía de Clases.
- Programación Orientada a Objetos: Clases, Método Constructor, Métodos de consulta y modificación, atributos, instanciamiento, y paquetes.
- Lenguaje Java: Estructuras de Control Condicional (simple, doble, anidada y múltiple), Estructuras Cíclicas (do-while, while, o for), patrones de Recorrido.

3. Material y Equipo

- Guía de laboratorio
- Computador
- Ambiente de Desarrollo Eclipse
- JDK Java
- Documentación de la plataforma Java

4. Actividades, resultados y conclusiones



En el presente laboratorio se abordará la manera de crear una Estructura de Datos: Lista Sencillamente Enlazada, para su manipulación de datos.

Paso 1.

Crear un proyecto en Eclipse con su nombre completo y código. Dentro del proyecto se deben crear dos paquetes. El primer paquete con el nombre de mundoDelProblema. El segundo paquete con el nombre interfazDelUsuario.

El path de cada paquete debe ser:

piinf.g90d.estructuradatos.mundodelproblema

piinf.g90d.estructuradatos.interfazDelUsuario

Paso 2

En el paquete **mundoDelProblema** crear una clase llamada **Nodo**.

En dicha clase, crear los siguientes atributos:

```
//Atributos del campo de la Información del Nodo
private String nombre;
private int codigo;
//Atributos del campo de Referencia
protected Nodo sig;
```

Paso 3

De acuerdo a los atributos de la Clase **Nodo**, crear el método de construcción.

```
//METODO CONSTRUCTOR
```

```
public Nodo(String nombre,int codigo)
```

```
{
    //Desarrollar código...
}
```

Paso 4

De acuerdo a los atributos de la Clase **Nodo**, crear los métodos de consulta.

```
//Método de consulta que retorna el nombre del cadete
```

```
public String getConsultarNombre()
```

```
{
    //Desarrollar código...
}
```

```
//Metodo de consulta que retorna el código del cadete
```

```
public int getConsultarCodigo()
```

```
{
    //Desarrollar código...
}
```

Paso 5

De acuerdo a los atributos de la Clase **Nodo**, crear los métodos de modificación.

```
//Método que modifica el nombre del cadete
```

```
public void setModificarNombre(String nuevoNombre) {
```

```
//Desarrollar código...
}
```

```
//Método que modifica el código del cadete
```

```
public void setModificarCodigo(int nuevoCodigo)
```

```
{
```

```
//Desarrollar código...
```

```
}
```

Paso 6

En el paquete *munDoDelProblema* crear una clase llamada *Lista*.

En dicha clase, crear los siguientes atributos:

```
//Atributos de la Lista
```

```
private Nodo cab; //Definición de la cabeza de la lista
```

Paso 7

Para inicializar la lista vacía, crear el método de construcción donde `cab = null`.

Tenga en cuenta que una Lista Sencillamente Enlazada está vacía cuando su cabecera apunta a null.

Paso 8

Crear el método `getListaVacía()` de tipo boolean que determine si la lista está vacía.

```
public boolean getListaVacía()
```

```
{
```

```
//Desarrollar código...
```

```
}
```

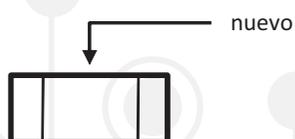
Paso 9

En la clase *Lista*, crear el método llamado `setInnsertarComienzo()` que permita insertar un dato al comienzo en la lista. El método debe tener como parámetros los datos nombre y código.

Los pasos a realizar para insertar un elemento al comienzo de la lista son:

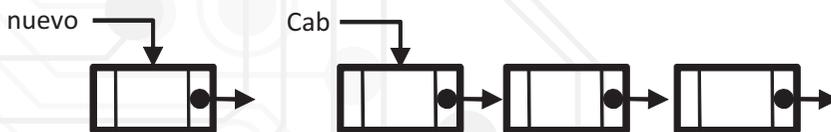
a) Creación de un nuevo nodo

```
Nodo nuevo = new Nodo(nom,cod);
```



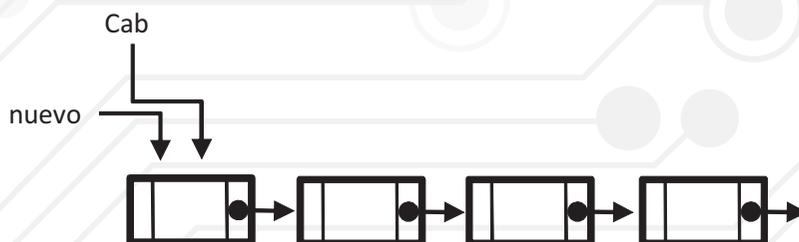
b) Poner a apuntar a nuevo en su campo siguiente, a donde apunta cabecera.

```
nuevo.sig = cab;
```



c) Cabecera apunta donde está apuntando nuevo.

```
cab = nuevo;
```



Paso 10

En la clase **Lista**, crear el método llamado **setInnserirFinal()** que permita insertar un dato al final de la lista. El método debe tener como parámetros los datos necesarios definidos en la clase **Nodo**.

Tenga en cuenta que si la lista esta vacía, se debe llamar el método **setInnserirComienzo()**.

El código a desarrollar al interior del método es el siguiente:

```
Nodo nuevo = new Nodo(nom,cod);
    Nodo aux = cab;
    //Se crea un ciclo para llevar un punter auxiliar hasta el último nodo de la lista
Mientras (aux.sig != nulo)
    {
        aux = aux.sig;
    }
    nuevo.sig = aux.sig;
    aux.sig = nuevo;
```

Paso 11

En la clase **Lista**, crear el método llamado **setEliminarComienzo()** que permita eliminar un dato o nodo al comienzo de la lista.

Tener en cuenta, si la lista está vacía.

El código a desarrollar al interior del método es el siguiente:

```
cab = cab.sig;
```

Paso 12

En la clase **Lista**, crear el método llamado **setEliminarFinal()** que permita eliminar un dato o nodo al final de la lista.

Tener en cuenta, si la lista está vacía.

El código a desarrollar al interior del método es el siguiente:

```
Nodo cabAux = cab;
    Mientras(cabAux.sig.sig != nulo)
    {
        cabAux = cabAux.sig;
    }
    cabAux = cabAux.sig.sig;
```

Paso 13

En el paquete **interfazDelUsuario** crear la clase principal la cual debe llevar como nombre el nombre del proyecto. Dentro de la clase principal.

Método principal **main()**. Dentro del método principal **main()**, desarrollar todo el código necesario que permita generar un menú con las siguientes opciones:

- a) Insertar al Comienzo
- b) Insertar al Final
- c) Eliminar al Comienzo
- d) Eliminar al Final
- e) Mostrar lista

Conclusiones:

5. Bibliografía

- Becerra, S. Cesar A. (2000). *Estructura de Datos en Java*. Bogotá – Colombia: Editorial Kimpres.
- Oracle. (04 de 02 de 2015). <http://docs.oracle.com/javase/7/docs/api/>. Obtenido de: <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
- Villalobos, J. A., & Casallas G., R. (2006). *Fundamentos de Programacion. Aprendizaje Activo Basado en Casos*. Bogota - Colombia: Pearson Prentice Hall.

Elaborado por: Docente Humberto Amaya

3.

LENGUAJES DE PROGRAMACIÓN

A través de las prácticas de laboratorio de “Lenguajes de Programación” los estudiantes podrán comprender las ventajas de estos lenguajes, que son un conjunto de instrucciones con una determinada estructura y contenido que permiten ejecutar una tarea específica; estas instrucciones corresponden al código fuente que está diseñado para cumplir un propósito específico.

En nuestro diario vivir interactuamos con distintos tipos de dispositivos electrónicos como son los computadores, tablets, celulares, portátiles, entre otros; todos ellos tienen un lenguaje de programación que facilita la traducción de órdenes y controla el comportamiento de estos dispositivos para que puedan ser utilizados por el usuario, ahí radica la importancia de estos lenguajes.

PRÁCTICA: PARADIGMA DE PROGRAMACIÓN – IMPERATIVO LENGUAJES DE PROGRAMACIÓN

1. Objetivos:

- Construir una perspectiva histórica de la Programación Imperativa.
- Identificar los aspectos comunes en los lenguajes imperativos.
- Conocer las características de los primeros lenguajes de programación.
- Reconocer los componentes principales de los Lenguajes Imperativos.
- Experimentar la programación con el lenguaje Algol.

2. Marco Teórico

El lenguaje Algorítmico se desarrolló en la década de los años 60 y se convirtió en el estándar para el diseño; sin embargo, a pesar de su uso nunca llegó a sobrepasar como lenguaje de uso comercial, su uso quedó aislado solamente a investigaciones científicas.

Un comité de representantes de Europa y América diseñaron ALGOL en 1958, en una respuesta a la creciente necesidad de tener un solo lenguaje de programación que fuese universal e independiente de la arquitectura de las computadoras de la época.

Los tres objetivos que rigieron el desarrollo de Algol fueron:

1. Debía ser tan cercano como fuese posible a la notación matemática estándar y debía ser legible dando pocas explicaciones adicionales.
2. Debía ser posible usarlo para la descripción de procesos de cómputo en las publicaciones científicas.
3. Debía ser traducible, de manera mecánica, a programas en lenguaje máquina.

Para sorpresa de todos, el diseño original de ALGOL tomó sólo 8 días (el lenguaje consistía en unos cuantos constructores de propósito general y no en una cantidad innumerable de funciones barrocas). Así, Algol 60 (la primera revisión de Algol 58) estaba lista para mayo de 1960.

Algol influyó a muchos lenguajes de la época, por lo que nos daremos cuenta que mucha de la sintaxis de otros lenguajes old school es parecida a la de Algol. (Juan Carlos Espinosa Cenicerros)

Una Estructura Contenedora de Tamaño Fijo – Arreglos Bidimensionales, es un espacio de memoria, donde se puede almacenar un conjunto de datos finitos y homogéneos.

Este espacio de memoria es estático, lo cual significa que en tiempo de ejecución no se puede disminuir o ampliar la capacidad de los datos. El espacio de memoria es referenciado por un dato definido en tiempo de compilación, donde los elementos se encuentran ubicados en el juego de coordenadas de filas x columna.

Para gestionar una Estructura Contenedora de Tamaño Fijo – Arreglos Bidimensionales, se requiere uso y aplicación de las estructuras cíclicas (do-while, while, o for), con un patrón de doble recorrido.

3. Material y Equipos

- Laboratorio de informática
- Computador
- Guía de laboratorio
- VirtualBox
- Interpretador Algol.

Agradecimiento especial a Juan Carlos Espinoza Cenicerros por permitir que parte de su material de clase sea utilizado en esta guía.

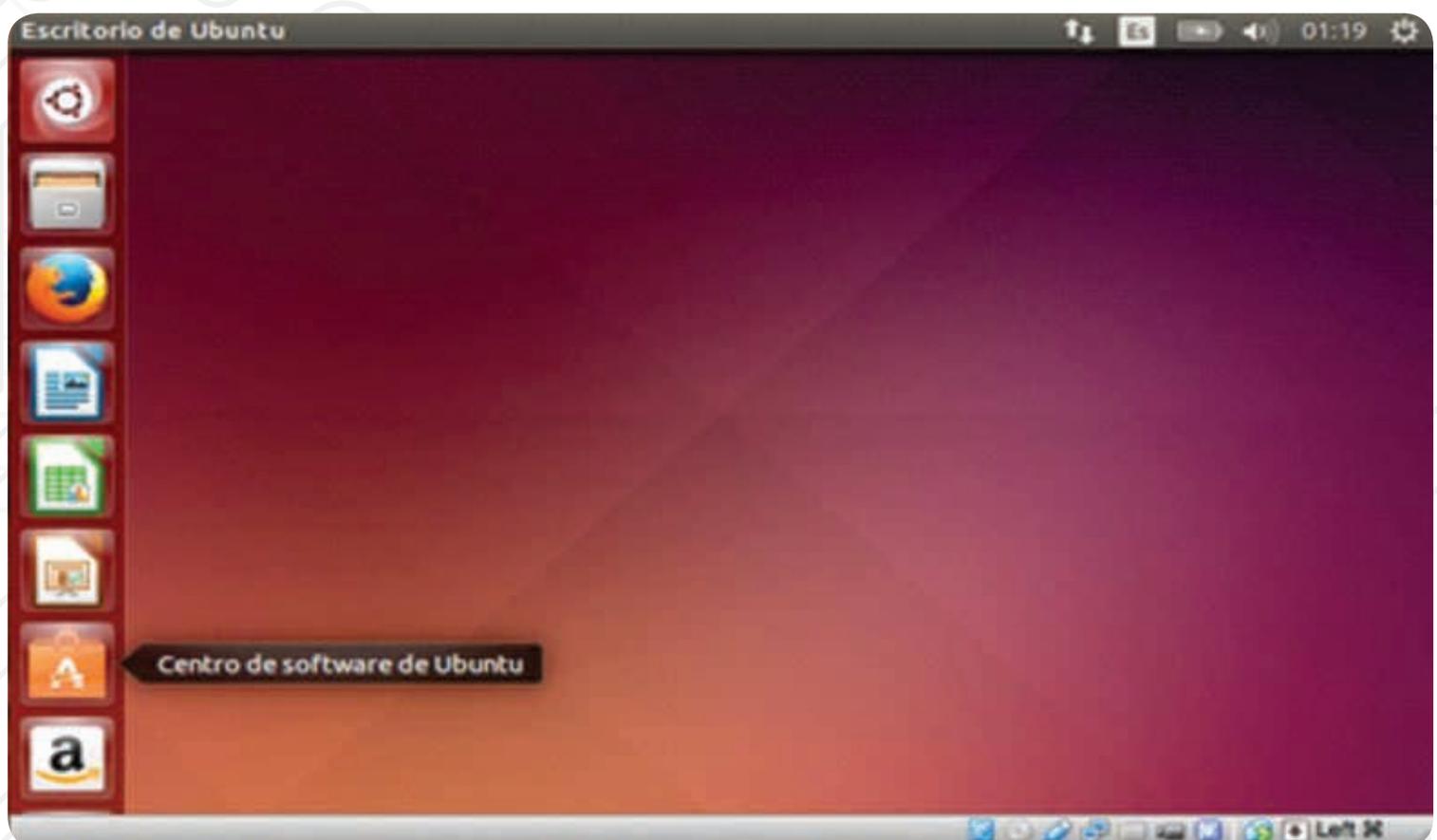
4. Actividades, Resultados y Conclusiones

Paso 1

Luego de encender el computador inicie el programa VirtualBox y active la máquina virtual Linux(Ubuntu)

Paso 2: Instalación del lenguaje de programación Algol

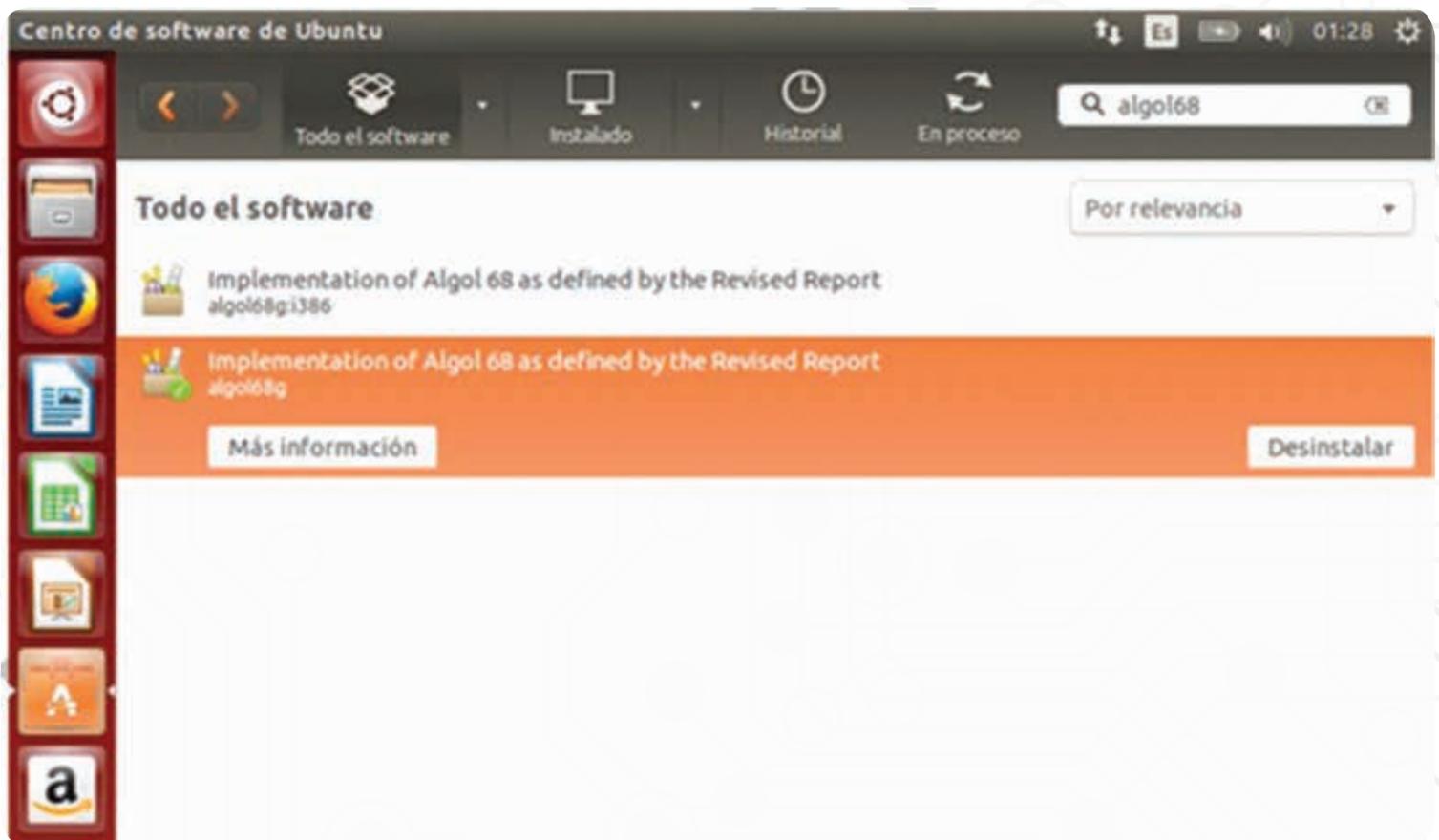
- Seleccionamos Centro de Software de Ubuntu



b) En la parte superior derecha en el buscador digitamos el nombre del lenguaje a instalar; en este caso **algol68**



c) Elegimos el algo correspondiente a la arquitectura del computador; en este caso la de 64



d) Se presiona el botón instalar

Paso 3: Variables y Datos

Al igual que en muchos lenguajes de programación, Algol maneja tipos simples de datos y otros más complejos; están, por ejemplo:

INT: Enteros

REAL: Números punto flotante

DOUBLE: Punto flotante de precisión doble

BOOL: Lógicos y Boleanos

COMPLEX: Complejos

ASTRING: Cadenas de caracteres

NODE: Nodos (para grafos, arboles, etc).

FILE: Manejo de archivos

Identifique cuáles de los anteriores tipos de datos están presentes en un lenguaje de programación moderno como el C++ o el Java

Paso 4

En Algol a estos tipos de datos se les conoce como mode-indicators (Indicadores de modo), y en otros lenguajes se le llama type identifier (indicadores de tipo [de dato]). Los indicadores de datos se pueden complementar con el identificador de modo LONG

Las variables se declaran con la siguiente sintaxis:

max int	$2^{31} - 1$	32-bit Pentium
long max int	$10^{35} - 1$	
long long max int	$10^{70} - 1$	
6048000 • 6 048 000		default precision
LONG 266716800000 • LONG 266 716 800 000		
LONG LONG		
3930061525912861057173624287137506221892737197425280369698987		

[mode-identifier] [var name]

Por ejemplo:

INT número

CHAR letra STRING nombre

Para declarar una variable e inicializarla se sigue esta sintaxis:

[mode-identifier] [var name] = [value]

Por ejemplo:

REAL pi: =3.14159; CHAR letra: = " A"; STRING nombre: = "Juan Carlos";

Compare la forma de declaración de variable en Algol versus en C++ y Java.

Paso 5: Entrada y Salida de Datos

Para mostrar los datos en pantalla (salida), se utiliza la función **print()**; lo que se va a imprimir se escribe dentro del doble paréntesis y encerrado en comillas por ejemplo:

Print (("hola" mundo"))

Si se quiere pasar un espacio se escribe **new line** dentro de los paréntesis, por ejemplo;

print (("Hola, mundo", new line));

Para tomar datos desde el teclado se utiliza el proceso read; se sigue la siguiente sintaxis:

[mode---identifier] [var name]=read [mode---identifier [low---case]];

Por ejemplo:

INTx = read int;

¿Qué funciones o métodos se emplean en C++ o Java para desplegar la información en el monitor y para obtener información desde el teclado?

Paso 6: Funciones Matemáticas

Algol incluye algunas funciones que nos pueden ser muy útiles al momento de escribir nuestro código;

aquí se muestran algunas:

sqrt(x): Raíz Cuadrada

curt(x): Raíz Cúbica

exp(x): Potencias

ln(x): Logaritmo Natural

log(x): Logaritmo base 10

sin(x): Seno

arcsin(x): Seno Inverso

cos(x): Coseno

arccos(x): Coseno Inverso

tan(x): Tangente

¿Existen las mismas (o similares) funciones y/o métodos en C++ o Java? Lístelas.

Paso 7: Estructura del código

Todo código en Algol tiene que estar delimitado por cláusulas que marquen su inicio y su final de esta forma:

```
BEGIN
```

```
...
```

```
PROCEDURES
```

```
...
```

```
END
```

Con ello podemos definir el código más corto que podemos escribir :)

```
BEGIN
```

```
print (("HOLA MUNDO", new line));
```

```
END
```

Los comentarios se limitan mediante la etiqueta COMMENT:

```
BEGIN
```

```
    COMMENT
```

```
    Este es mi primer programa en ALGOL
```

```
    COMMENT
```

```
    print (("HOLA MUNDO", new line))
```

```
END
```

SECUENCIAS DE CONTROL

Condicional

```
BEGIN
```

```
IF INT x = read int;
```

```
x < 0
```

```
THEN print ((x, " es negativo"))
```

```
ELSE print ((x, " no es negativo"))
```

```
FI
```

```

END
Ciclos FOR
BEGIN
    NT j := 10; FOR i TO j DO
        print (("HOLA", new line))
    OD

```

```

END
Desde un valor a otro final, incremento
BEGIN
    FOR i FROM -10 TO 10 DO
        print ((i, new line))
    OD

```

```

END
Desde un valor a otro final, dos en dos
BEGIN
    FOR i FROM 0 BY 2 TO 10 DO
        print ((i, new line))
    OD

```

```

END
Desde un valor a otro final, decremento
BEGIN
    FOR i FROM 10 DOWNTO 1 DO
        print ((i, new line))
    OD

```

```

END
Ciclo WHILE
BEGIN
    INT x;
    WHILE
        Read(X);
        Int > 0
    DO
        Print (("Mas abajo..."))
    OD

```

END
 Compare las Sentencia de Control presentes en Algol y las disponibles en C++ y Java.

Paso 8: Ejemplo

Escriba un pequeño código en Algol que nos calcule potencias. Abrimos el Gedit en Ubuntu. Después escriba este código:

```

BEGIN
  INT base; INT exp;
  LONG INT pot := 1;
  print(("BASE: "));
  read(base);
  print(("EXPONENTE: "));
  read(exp);
  FOR i TO exp DO
    pot := pot * base
  OD;
  print((base," elevado a la potencia",exp," es",pot))
END

```

Paso 9: Almacenamiento

Luego de terminar la edición del archivo, cree un directorio con un nombre significativo como por ejemplo /LaboratorioAlgol68.

Guarde el archivo con un nombre representativo y con extensión a68, dentro del directorio que acaba de crear. Por ejemplo: Potencia.a68

Paso 10: Interpretación del código

Abra una Terminal y mediante el intérprete de comandos vaya al directorio donde guardó el archivo que acaba de crear.

Dado que la versión del Algol68 que ha instalado es un interpretador no un compilador, cada vez que quiera ejecutar el programa este se debe traducir.

En la terminal digite: a68g potencia. a68 y luego presione Enter

Haga un ‘ pantallazo’ al terminar la ejecución y muestre los resultados.

Paso 11: Resultados

Conclusiones:

5. Bibliografía.

- Abelson y Sussman: Capítulo 3.3 (Modeling with Mutable Data, apartados 3.3.1-3.3.3)
- Programming in Scala: Capítulo 7 (Built-in Control Structures) y Capítulo 5.7 (Object equality)

Elaborado por: Docente Carlos Tavera

PRÁCTICA PROGRAMACIÓN LÓGICA LENGUAJES DE PROGRAMACIÓN

1. Objetivos

- Identificar los fundamentos de la programación lógica.
- Familiarizar a los estudiantes con la aplicación de los conceptos de programación lógica y las ventajas que ofrece.

2. Marco Teórico

La programación lógica es la descomposición de programas en sus componentes lógicos y sus componentes de control. Con lenguajes de programación lógica de bajo nivel, estos componentes determinan la solución del problema, por eso los componentes de control pueden variar para proporcionar alternancia de ejecución de un programa lógico. El lenguaje de programación lógica por excelencia es Prolog, que cuenta con diversas variantes. La más importante es la programación lógica con restricciones, que posibilita la resolución de ecuaciones lineales además de la demostración de hipótesis.

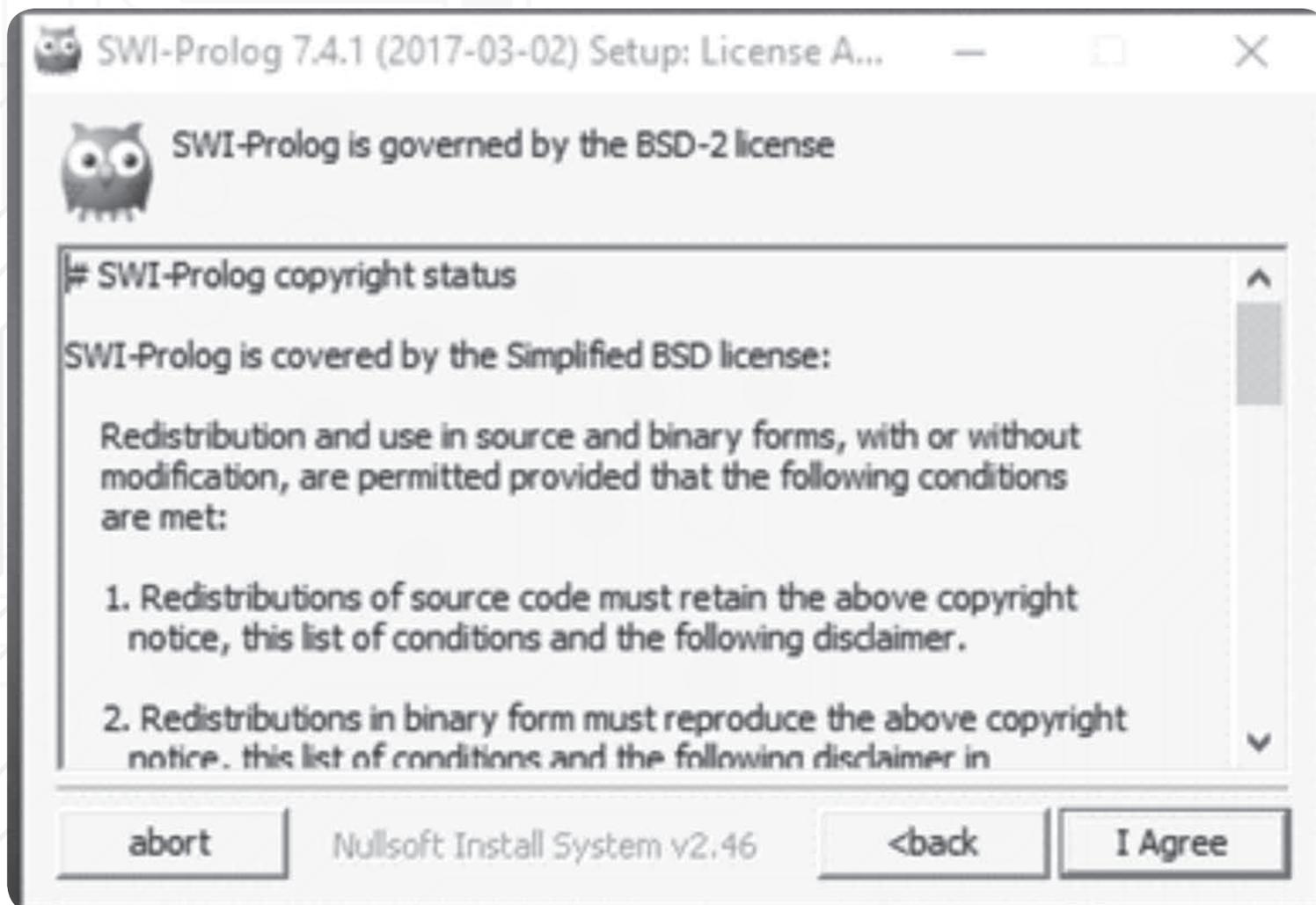
3. Material y equipo

- Laboratorio de informática
- Computador
- Guía de laboratorio.

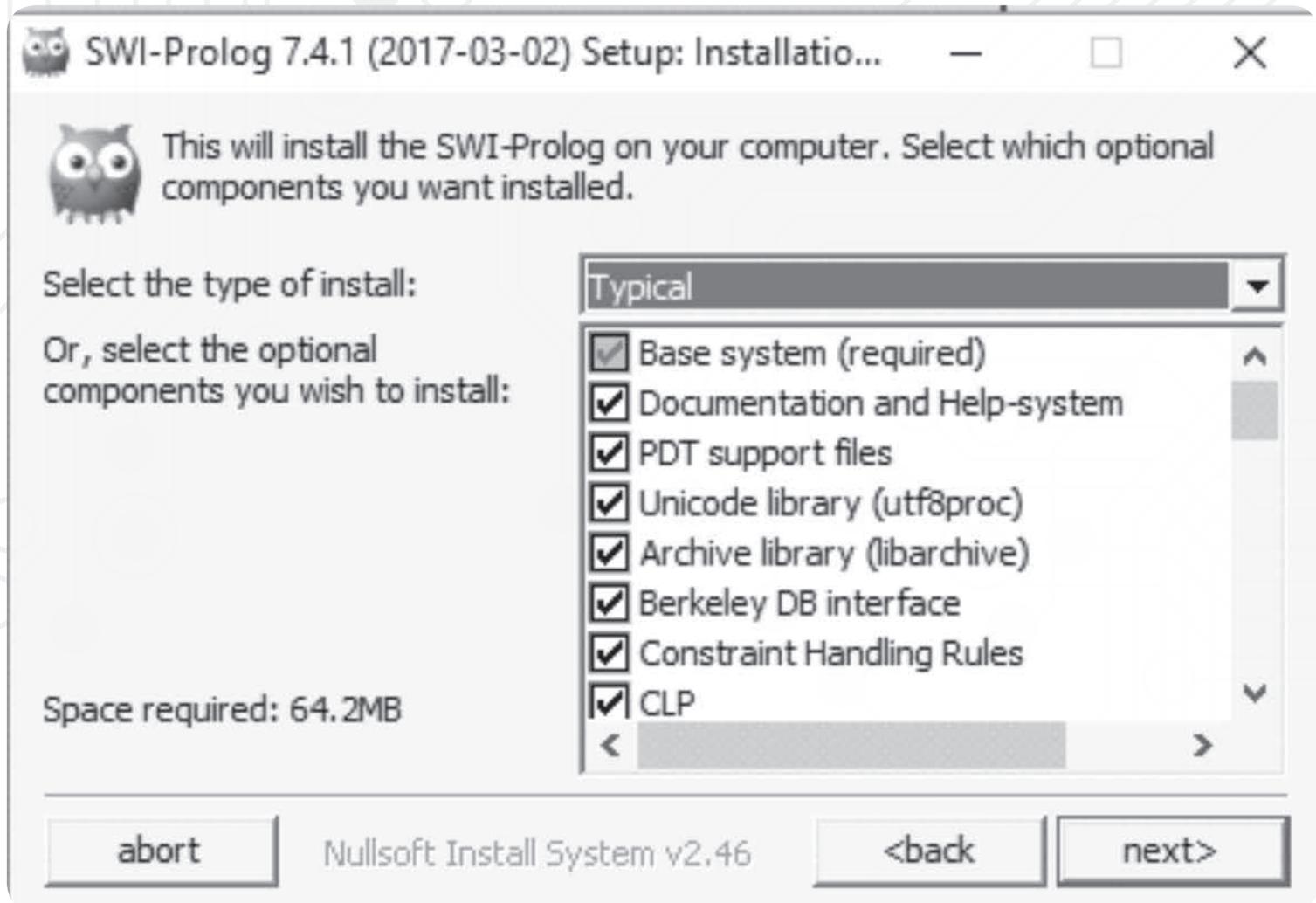
4. Actividades, resultados y conclusiones

Descargar de la página <http://www.swi-prolog.org/download/stable> al archivo de instalación dependiendo del sistema operativo de su máquina.

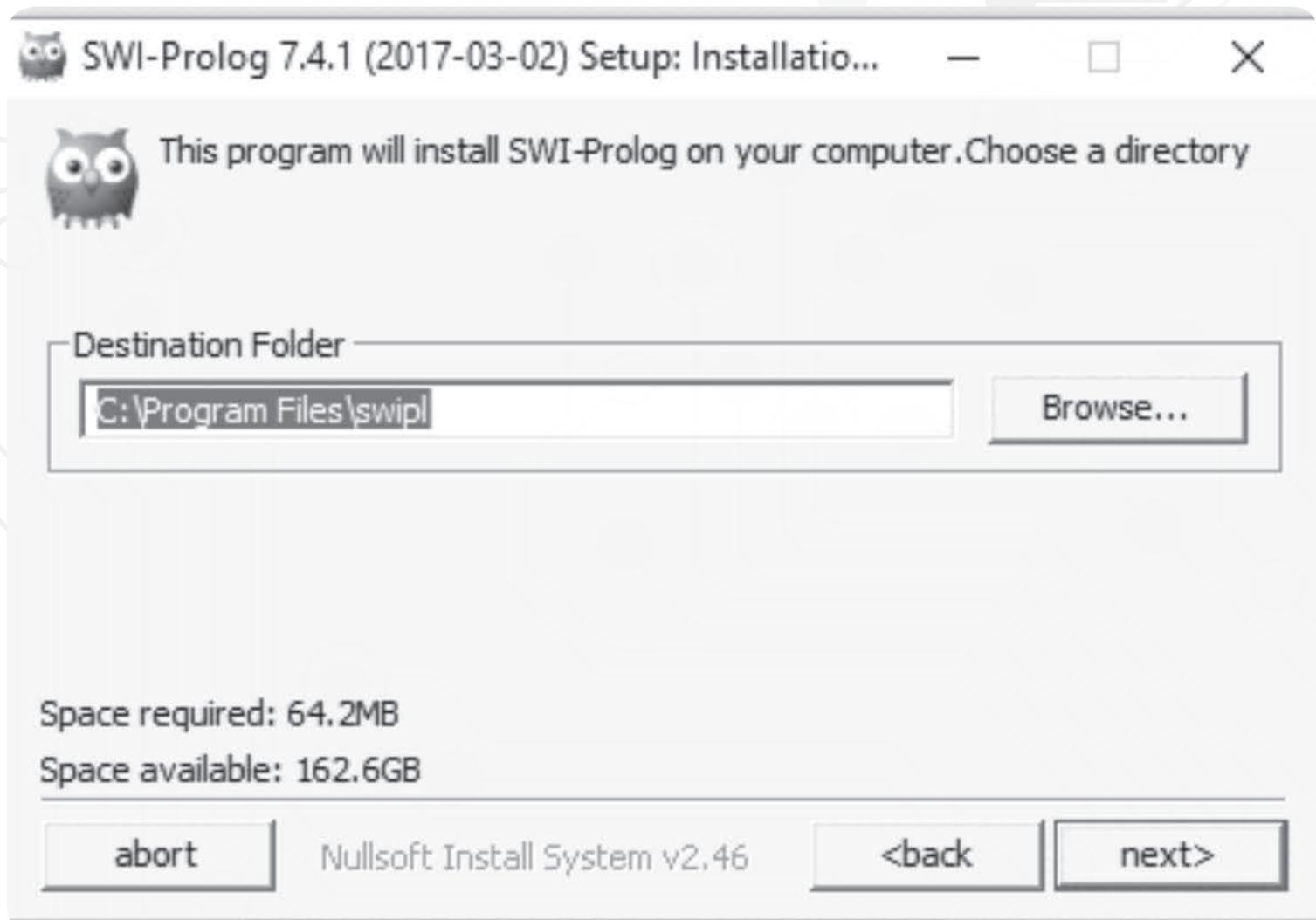
Hacer clic en el instalador y en el botón “I agree” ...estás de acuerdo con la licencia.



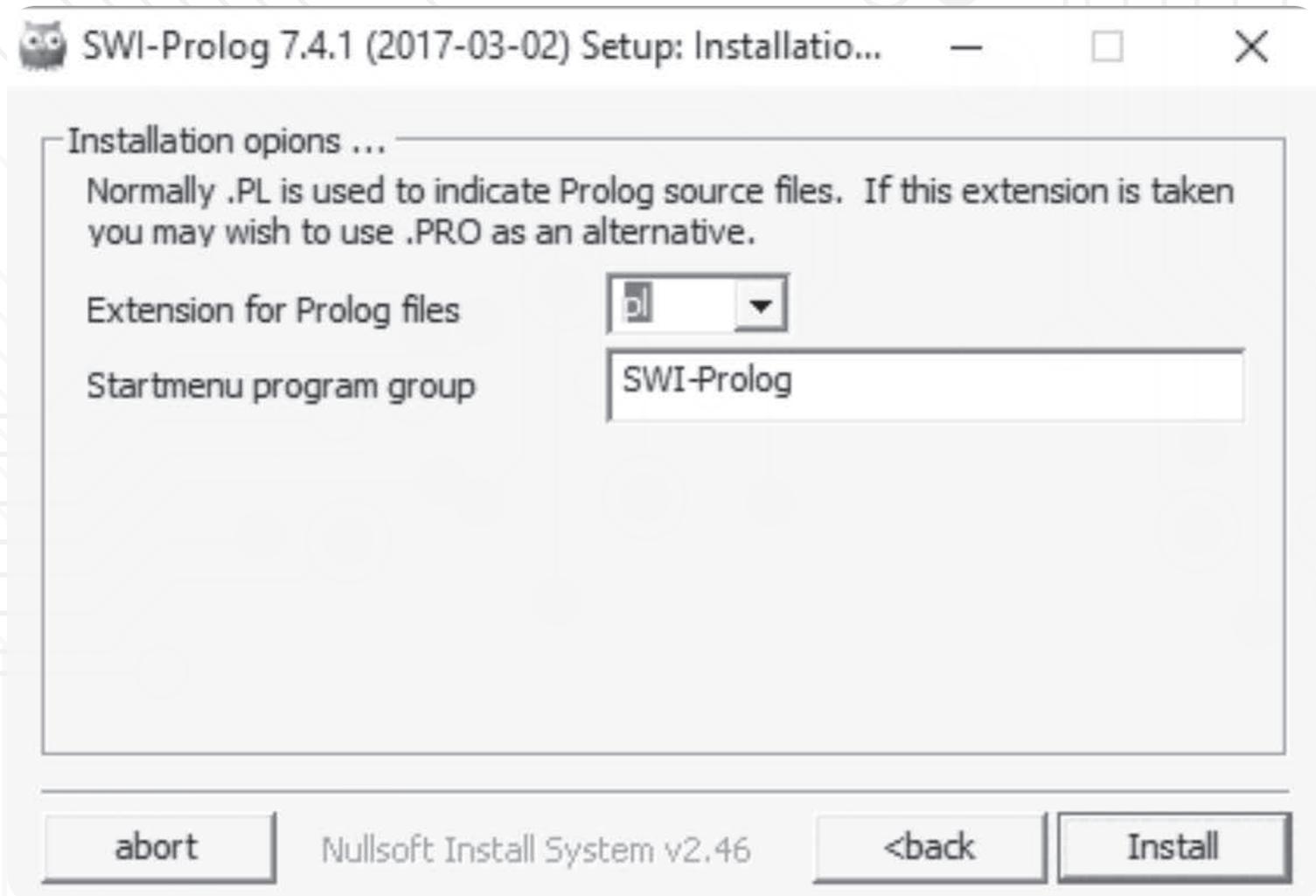
Se pueden dejar los ítems seleccionados...



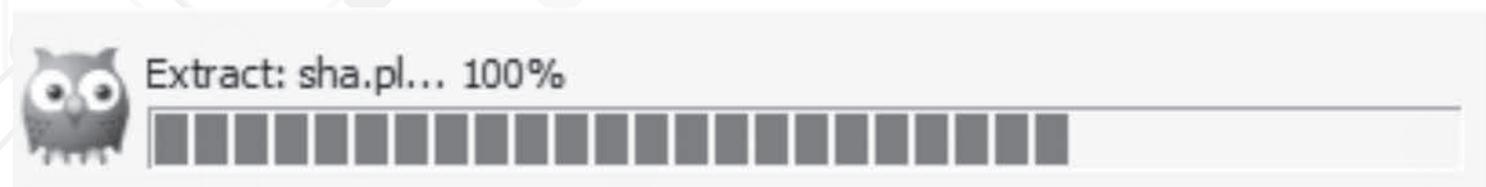
Clic en el botón Next ...



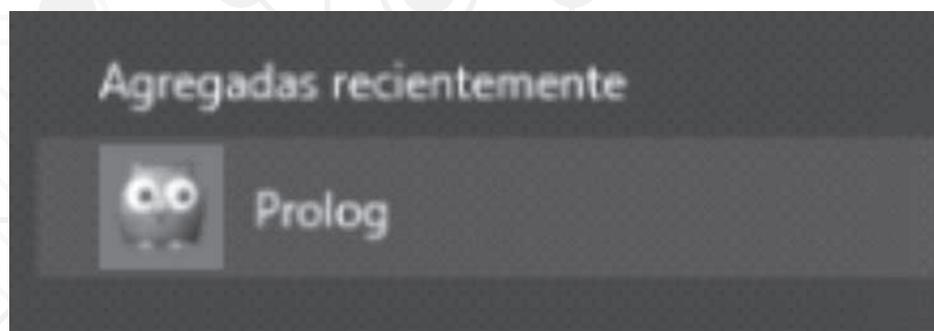
Clic en el botón Install ...



Esperar mientras se instala...



Buscamos el prolog instalado y lo iniciamos...



Al iniciar quedamos en el interpretador...

Lenguaje declarativo (opuesto a procedimentales) basado en reglas de la lógica

PROLOG = "Programming in Logic"

Programas en
Prolog

Hechos

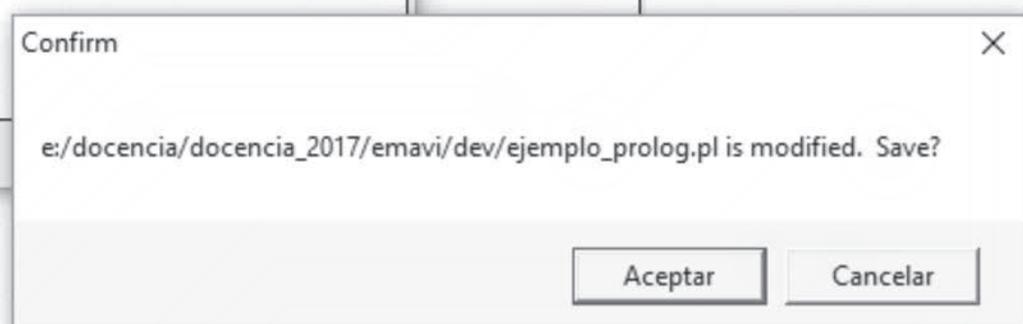
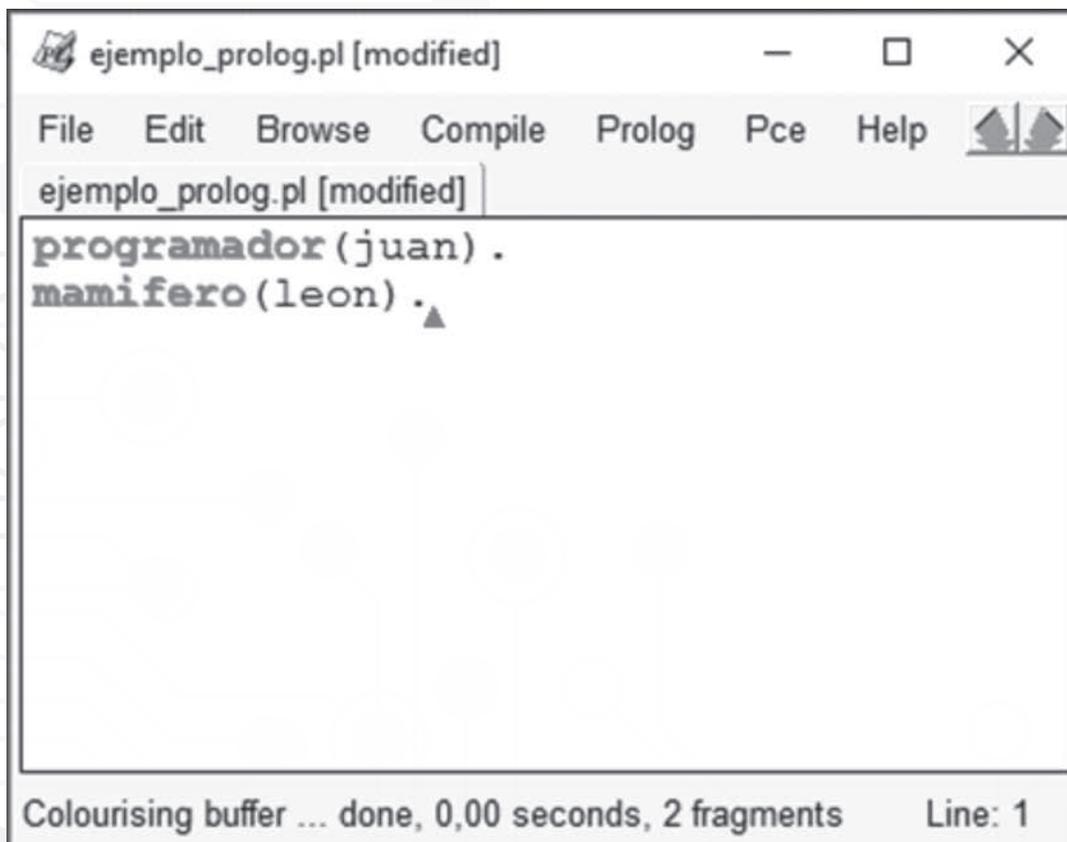
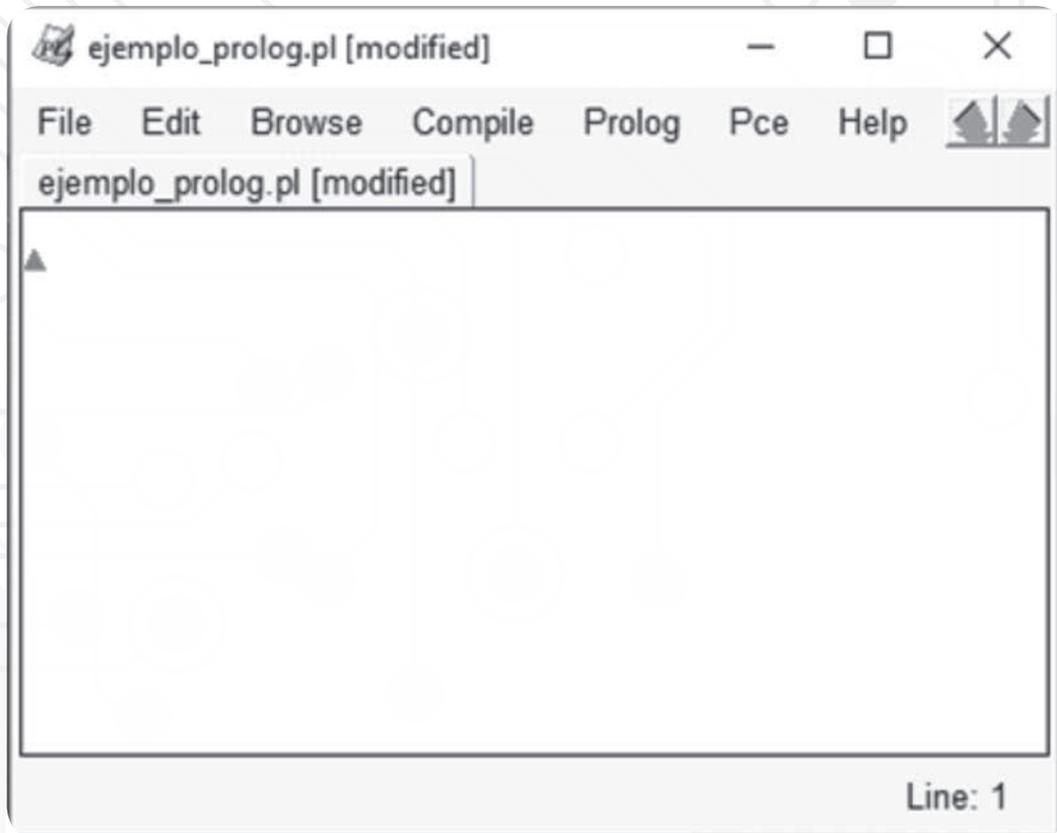
Reglas

- La información se extrae por medio de consultas
- Proposiciones: o "Juan es un programador" o "El león es un mamífero"
- Programador(Juan).
- Mamífero(león).

Nótese que se anota primero el predicado y entre paréntesis el sujeto de la proposición.

En el intérprete de prolog, en el menú seleccionamos New para editar una base de conocimiento en prolog.

Y creamos un archivo llamado ejemplo_prolog. (El archivo es de extensión .pl)



?-
 e:/docencia/docencia_2017/emavi/dev/ejemplo_prolog compiled 0.00 sec, 0 clauses

Al preguntar en el sistema si Juan es programador, obtenemos verdadero como resultado.

```
?- programador(juan).
true.
-
?- mamifero(juan).
false.
```

Si pregunto quién puede ser un programador, obtenemos como respuesta Juan. Las variables van en Mayúsculas, y las constantes en minúsculas.

```
?- programador(X).
X = juan.

?- mamifero(Y).
Y = leon.
```

Sintaxis:

- Variables: Mayúsculas.
- Constantes: Minúsculas.
- Cada afirmación debe terminar con un punto.
- No se aceptan constantes con espacios en blanco, en lugar de ello se usa el guión bajo (_). Por ejemplo: primer_ministro.
- Los comentarios empiezan con el símbolo %

Relaciones:

“Juan es el padre de Miguel”
padre(juan, miguel).

Reglas:

- Sentencias condicionales
- “Si el león come carne, entonces es carnívoro”
carnívoro(león):- comecarne(león)

Conjunciones:

- Emplea el operador lógico AND
- Se utiliza la coma (,)

```
tia (X,Y):-hermana (X,Z) , padre (Z,Y) .
```

Regla
(Condición)

AND

Disyunciones:

- Emplea el operador lógico OR
- Se utiliza el punto y coma (;)

```
hijo (X,Y):-padre (Y,X) ; madre (Y,X) .
```

Regla
(Condición)

OR

Ejemplo:

```
padre(juan, camilo).
hermana(maria, juan).
tia(X, Y):-hermana(X, Z), padre(Z, Y).
```

Compilando:

```
?- tia(X,juan).
false.

?- tia(X,camilo).
X = maria.

?- tia(Maria, X).
Maria = maria,
X = camilo.

?- ■
```

5. Bibliografía

- P. Julián, M. Alpuente (2007). Programación Lógica. Teoría y Práctica. Pearson Prentice Hall.

Elaborado por: Docente Ing. John Vargas

HERENCIA Y POLIMORFISMO LENGUAJES DE PROGRAMACIÓN

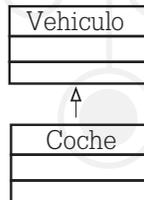
1. Objetivos

- Identificar las funcionalidades y ventajas que ofrece la herencia el polimorfismo en la programación orientada a objetos.
- Familiarizar a los estudiantes con la aplicación de los conceptos de herencia y polimorfismo.

2. Marco Teórico

La herencia es un mecanismo por el que se pueden crear nuevas clases a partir de otras existentes, heredando, y posiblemente modificando, y/o añadiendo otras operaciones, también es posible añadir atributos, es importante verificar que una operación o atributo no puede ser suprimido en el mecanismo de herencia.

Relación de herencia: → Todos los carros son vehículos



Nomenclatura

clase original	superclase	padre	Vehículo
clase extendida	subclase	hijo	Coche

El polimorfismo es la característica de la programación orientada a objetos que permite modificar la instancia de un objeto en tiempo de ejecución basado en una jerarquía de herencia. De esta forma, es posible generar una relación de vinculación denominada “binding”. El polimorfismo se puede realizar con clases superiores normales, abstractas e interfaces. El objetivo del polimorfismo consiste en poder acceder a diferentes servicios en tiempo de ejecución sin necesidad de implementar diferentes referencias a objetos. Esta característica provee una gran flexibilidad en el proceso de desarrollo y ejecución de la aplicación.

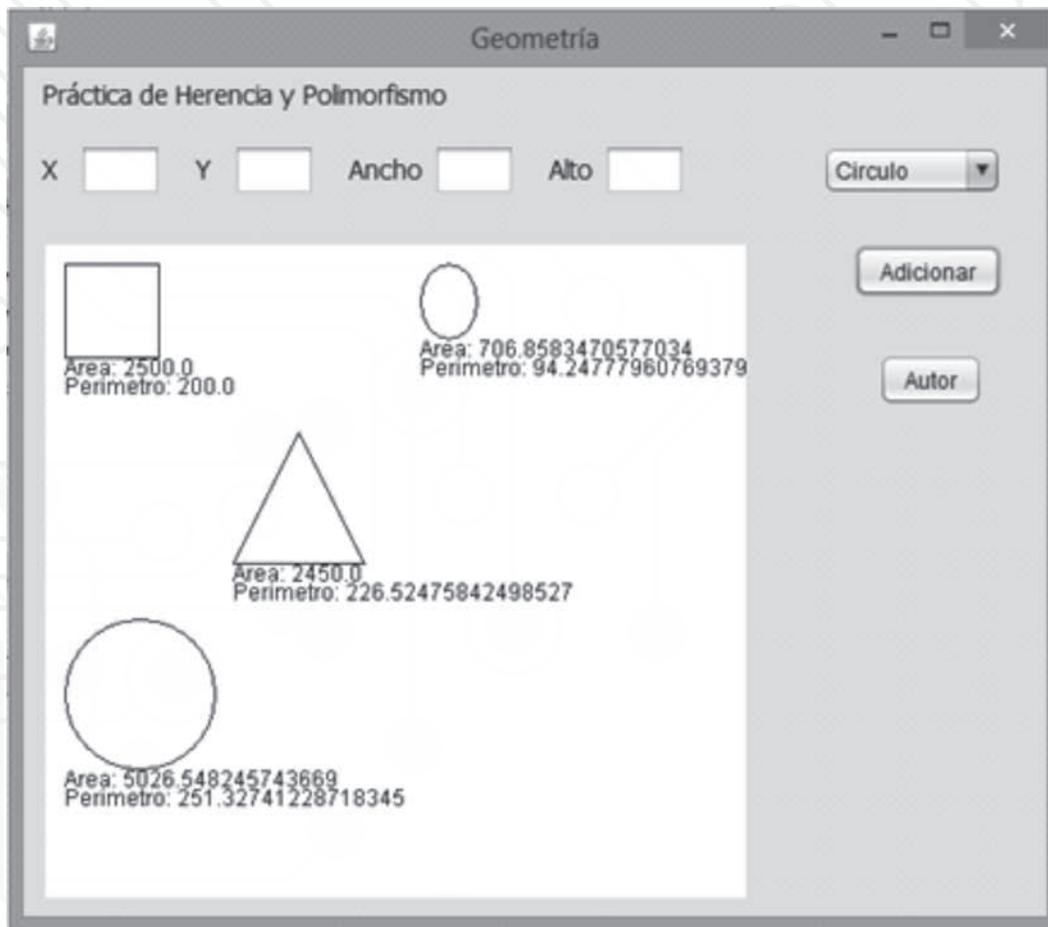
3. Material y equipo

- Guía de laboratorio
- Computador

4. Actividades, resultados y conclusiones

Para realizar el taller se deben seguir los siguientes pasos.

- ▶ Creamos un nuevo proyecto llamado Geometría, sin clase principal.
- ▶ Adicionamos los paquetes gui y modelo.
- ▶ Creamos una nueva clase en el paquete modelo llamada Figura.
- ▶ Crear la clase Rectángulo que hereda de la clase Figura
- ▶ Crear la clase Ventana que es un JFrame en el paquete gui
- ▶ Crear una clase llamada Tablero que es un Canvas en el paquete gui.
- ▶ Compilar para generar las clases construidas hasta el momento.



Para completar el taller es necesario implementar la clase Circulo.java. La instrucción para dibujar un círculo es:
Circulo.java

```

package modelo;

import java.awt.Graphics;

public abstract class Figura {
    private double area, perimetro;
    private int x, y, ancho, alto;

    public Figura(int x, int y, int ancho, int alto) {
        this.x = x;
        this.y = y;
        this.ancho = ancho;
        this.alto = alto;
    }

    public double getArea() {
        return area;
    }

    public double getPerimetro() {
        return perimetro;
    }

    public void setArea(double area) {
        this.area = area;
    }
}
    
```

```

public void setPerimetro(double perimetro) {
    this.perimetro = perimetro;
}

```

```

public int getX() {
    return x;
}

```

```

public int getAncho() {
    return ancho;
}

```

```

public int getAlto() {
    return alto;
}

```

```

public abstract void dibujar(Graphics dibujo);
public abstract void calcularArea();
public abstract void calcularPerimetro();
}

```

Rectangulo.java

```

package modelo;

import java.awt.Graphics;

public class Rectangulo extends Figura {

    public Rectangulo(int x, int y, int ancho, int alto) {
        super(x, y, ancho, alto);
    }

    @Override
    public void calcularArea()
    {
        double area = getAncho() * getAlto();
        System.out.println(area);
        setArea(area);
    }
}

```

```

@Override
public void calcularPerimetro()
{
    double perimetro = 2 * getAncho() + 2 * getAlto();
    setPerimetro(perimetro);
}

@Override
public void dibujar(Graphics dibujo)
{
    dibujo.drawRect(getX(), getY(), getAncho(), getAlto());
    dibujo.drawString("Area: " + getArea(), getX(), getY() + getAlto() + 10);
}
}

```

Triangulo.java

```

package modelo;
import java.awt.Graphics;
import javax.swing.JOptionPane;

public class Triangulo extends Figura {
    private int x[];
    private int y[];

    public Triangulo(int x, int y, int ancho, int alto) {
        super(x, y, ancho, alto);

        this.x = new int[]{x, x + ancho / 2, x + ancho};
        this.y = new int[]{y + alto, y, y + alto};
    }

    @Override
    public void calcularArea()
    {
        double area = getAncho() * getAlto() / 2.0;
        setArea(area);
    }

    @Override
    public void calcularPerimetro()
    {
        int b = getAncho(); // base
        int a = getAlto(); // altura

        double perimetro = 2.0 * Math.sqrt(a*a + (b/2.0)*(b/2.0)) + b;
        System.out.println(perimetro);
        setPerimetro(perimetro);
    }
}

```

```
@Override
public void dibujar(Graphics dibujo)
{
    dibujo.drawPolygon(this.x, this.y, this.x.length);
    dibujo.drawString("Perimetro: " + getPerimetro(), getX(), getY() + getAlto() + 10);
}
}
```

Tablero.java

```
package gui;
import java.awt.Canvas;
import java.awt.Graphics;
import modelo.Figura;

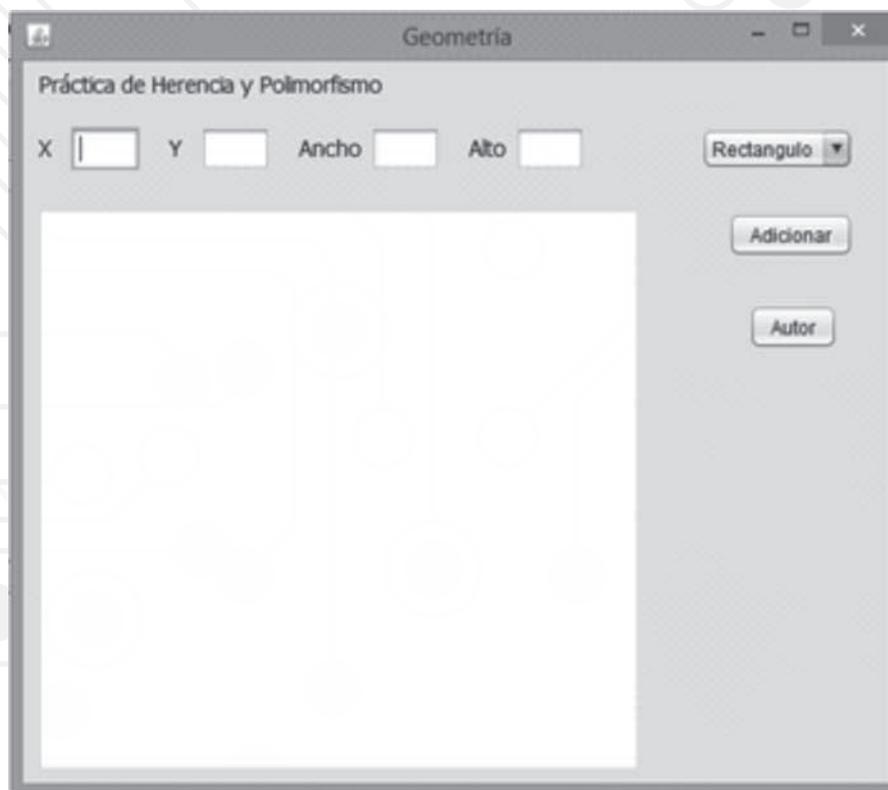
public class Tablero extends Canvas {
    private Figura lista[];
    private int cantidadFiguras;

    public Tablero()
    {
        lista = new Figura[100];
        cantidadFiguras = 0;
    }

    public void adicionarFigura(Figura f)
    {
        lista[cantidadFiguras++] = f;
    }

    @Override
    public void paint(Graphics dibujo)
    {
        for (int i = 0; i < cantidadFiguras; i++) {
            lista[i].dibujar(dibujo); // Polimorfismo
        }
    }
}
```

Ventana.java



```
private void botonAdicionarActionPerformed(java.awt.event.ActionEvent evt) {
    String value=lista.getSelectedItem().toString();
    int x = Integer.parseInt(entradaX.getText());
    int y = Integer.parseInt(entradaY.getText());
    int ancho = Integer.parseInt(entradaAncho.getText());
    int alto = Integer.parseInt(entradaAlto.getText());

    Figura f = null;
    switch (value) {
        case "Rectángulo":
            f = new Rectángulo(x, y, ancho, alto);
            break;
        case "Triángulo":
            f = new Triángulo(x, y, ancho, alto);
            break;
    }

    f.calcularArea();
    f.calcularPerimetro();
    tablero.adicionarFigura(f);
    tablero.repaint();

    entradaX.setText("");
    entradaY.setText("");
    entradaAncho.setText("");
    entradaAlto.setText("");
}
}
```

5. Bibliografía

- P. Julián, M. Alpuente. (2007). Programación Lógica. Teoría y Práctica, Pearson Prentice Hall.

Elaborado por: Docente Ing. John Vargas

PRÁCTICA REPASO HERENCIA Y POLIMORFISMO LENGUAJES DE PROGRAMACIÓN

1. Objetivos:

- Identificar los fundamentos de la programación orientada a objetos.
- Los estudiantes se familiarizarán y aplicarán los conceptos de herencia, polimorfismo y desarrollo de GUI en un programa orientado a objetos con Java.

2. Marco Teórico

En esta actividad se utilizarán los conceptos básicos de programación orientada a objetos como clases, objetos, herencia en un programa hecho con netbeans, usando el paquete de java.swing para el diseño de la interfaz gráfica de usuario.

3. Material y equipo

- Guía de laboratorio
- Netbeans
- JDK de Java

4. Actividades, resultados y conclusiones

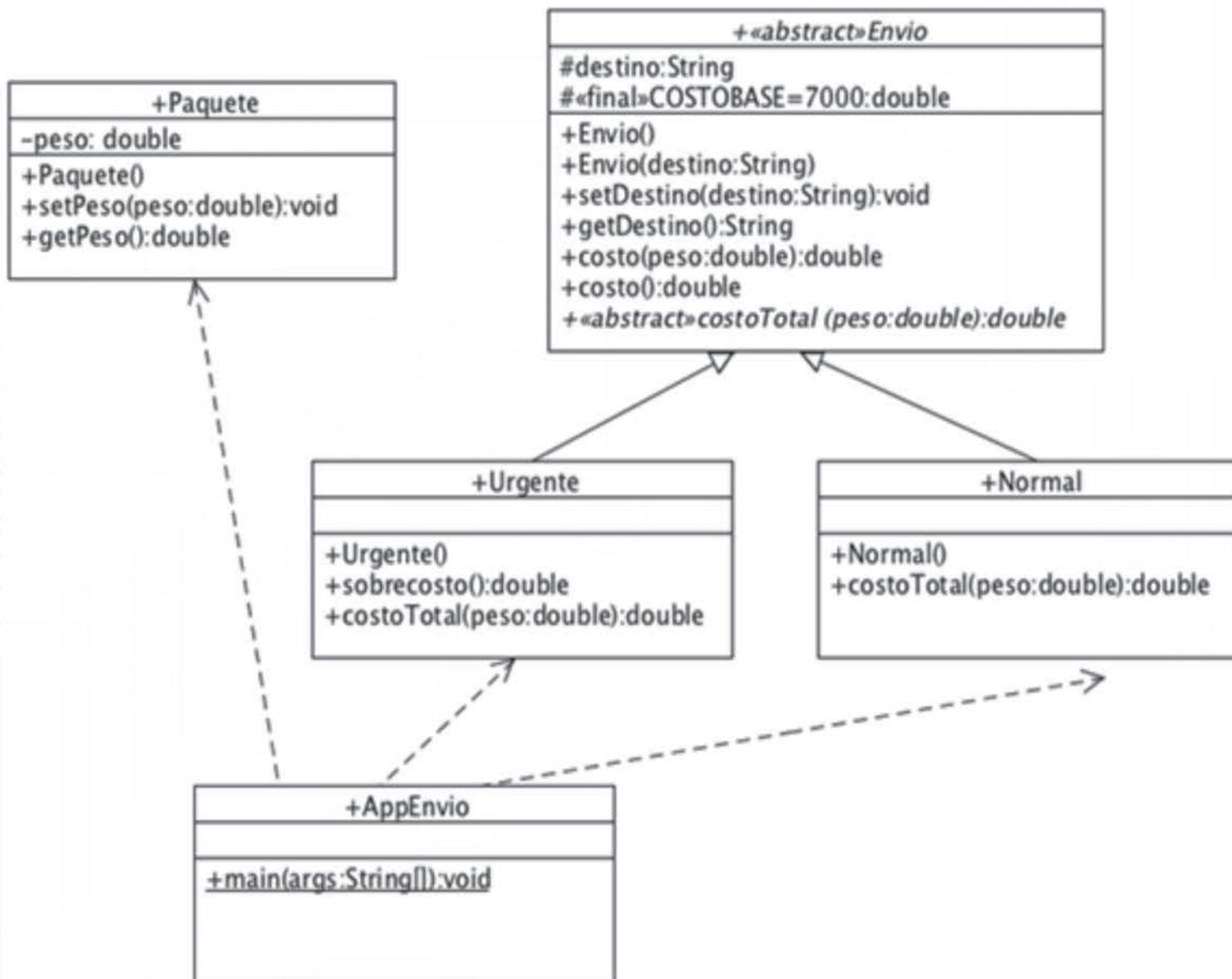
El sistema de mensajería administrado por la empresa “El Envión”, considera dos tipos de servicio de entrega: urgente y normal, diferenciando al primero con un sobrecosto del 50% sobre el valor base del servicio que es de \$7000 y que se cobra para ambos. Los precios parciales con los que se calcula el valor final a pagar, dependen del peso del paquete (en kilogramos) y del destino a donde llegará el paquete.

Las siguientes tablas ilustran las condiciones:

PESO (KILOGRAMOS)	
RANGO	COSTO PARCIAL (PESOS)
Menos de 5	2000
Entre 5 y 20	10000
Más de 20	15000

DESTINOS	
DESTINO	COSTO PARCIAL (PESOS)
Bogota	3000
Medellin	4000
Otra Ciudad	6000

Diseñar una solución, en java, que determine el precio final a pagar por el envío de un paquete en la empresa mencionada.



Distribución de las clases



Clases: Envio.java

Atributos, métodos constructores: por defecto y parametrizado; métodos getter y setter.

```
package envios;

public abstract class Envio {
    private String destino;
    protected final double COSTOBASE = 7000;

    public Envio() {
    }

    public Envio(String destino) {
        this.destino = destino;
    }

    public String getDestino() {
        return destino;
    }

    public void setDestino(String destino) {
        this.destino = destino;
    }
}
```

Métodos para el cálculo del costo

```
public double costo()
{
    switch (destino) {
        case "Bogotá":
            return 3000;
        case "Medellin":
            return 4000;
    }

    return 6000;
}

public double costo(double peso)
{
    if (peso < 5) {
        return 2000;
    }
}
```

```

if (peso < 20) {
    return 10000;
}

return 15000;
}

```

```

public double costoTotal(double peso)
{
    return COSTOBASE + costo() + costo(peso);
}

```

Urgente.java

```

package envios;

```

```

public class Urgente extends Envio {

```

```

    public Urgente() {
    }

```

```

    public Urgente(String destino) {
        super(destino);
    }

```

```

@Override

```

```

public double costoTotal(double peso)
{
    return COSTOBASE * 1.5 + costo() + costo(peso);
}

```

```

}

```

Normal.java

```

package envios;

public class Normal extends Envio {

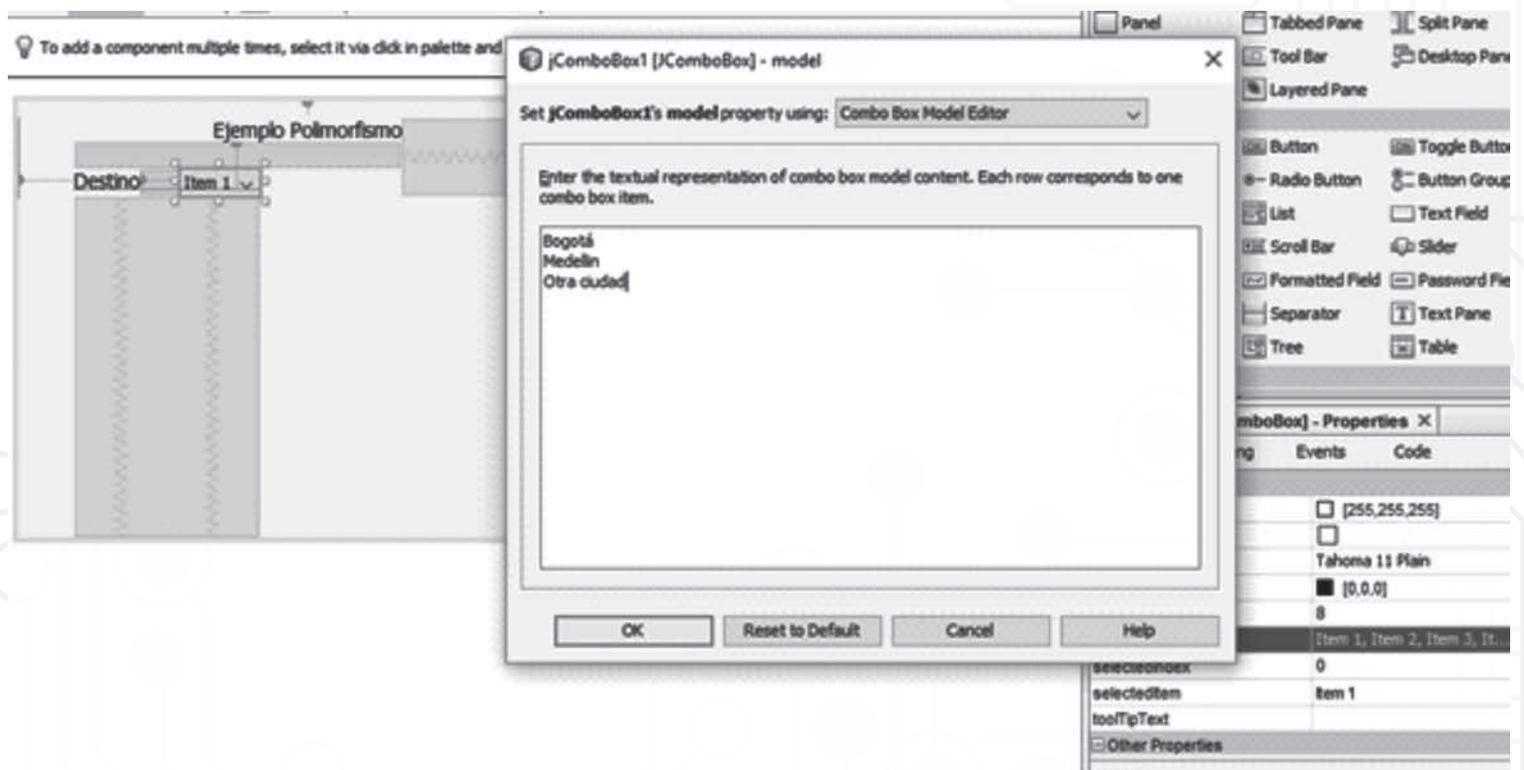
    public Normal() {
    }

    public Normal(String destino) {
        super(destino);
    }
}

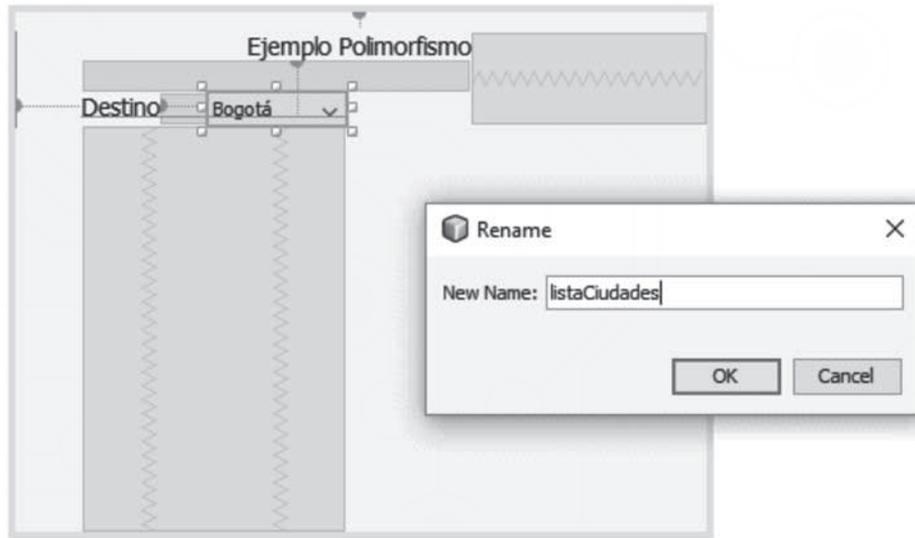
```

Principal.java

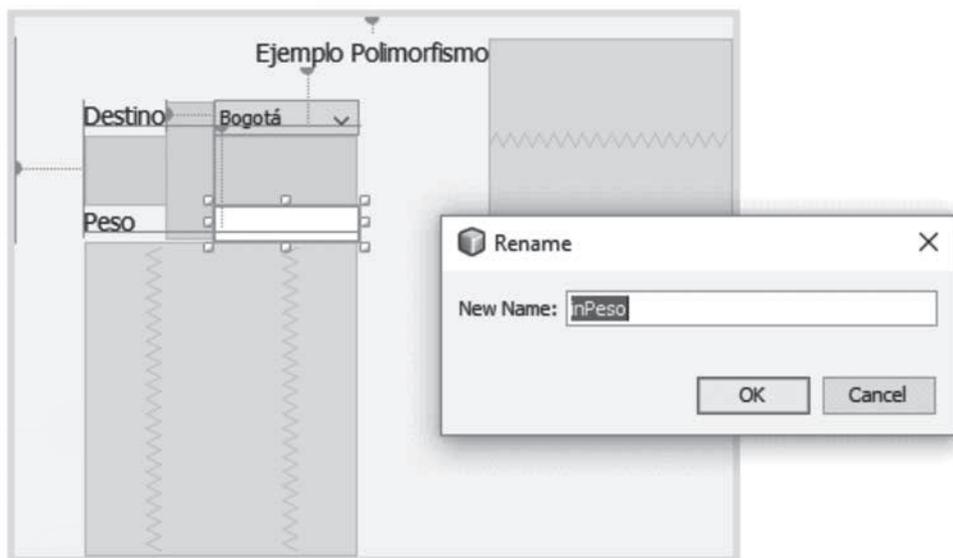
Se diseña la interfaz, y se usa el combobox para la lista de ciudades. (Bogotá, Medellín, Otras ...)



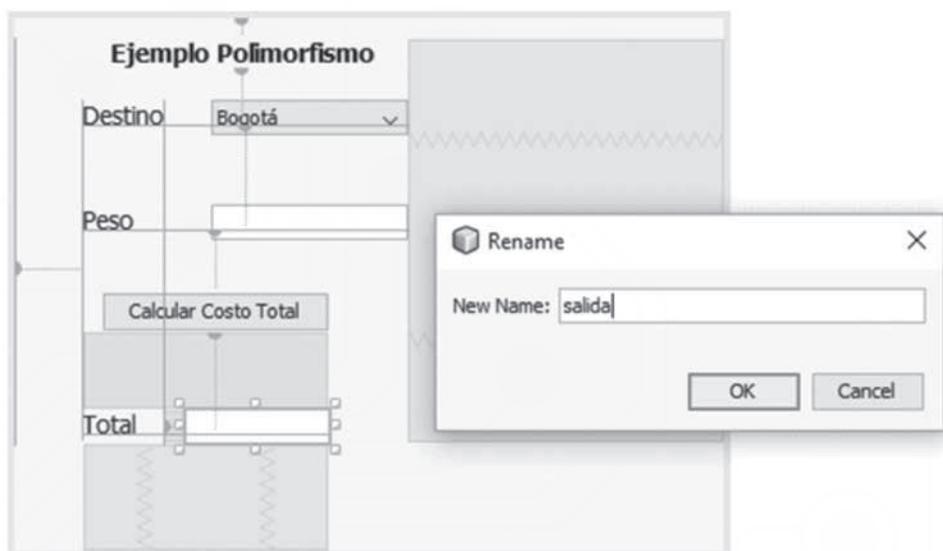
Se nombra el comboBox como listaCiudades



Se adiciona la entrada para el peso, con nombre inPeso



El campo de Texto para la salida



El campo de la salida se marca como no editable

The screenshot shows the 'salida [JTextField] - Properties' dialog box. The 'editable' property is checked as false. Below the main list, a tooltip for the 'editable' property is displayed, explaining its function.

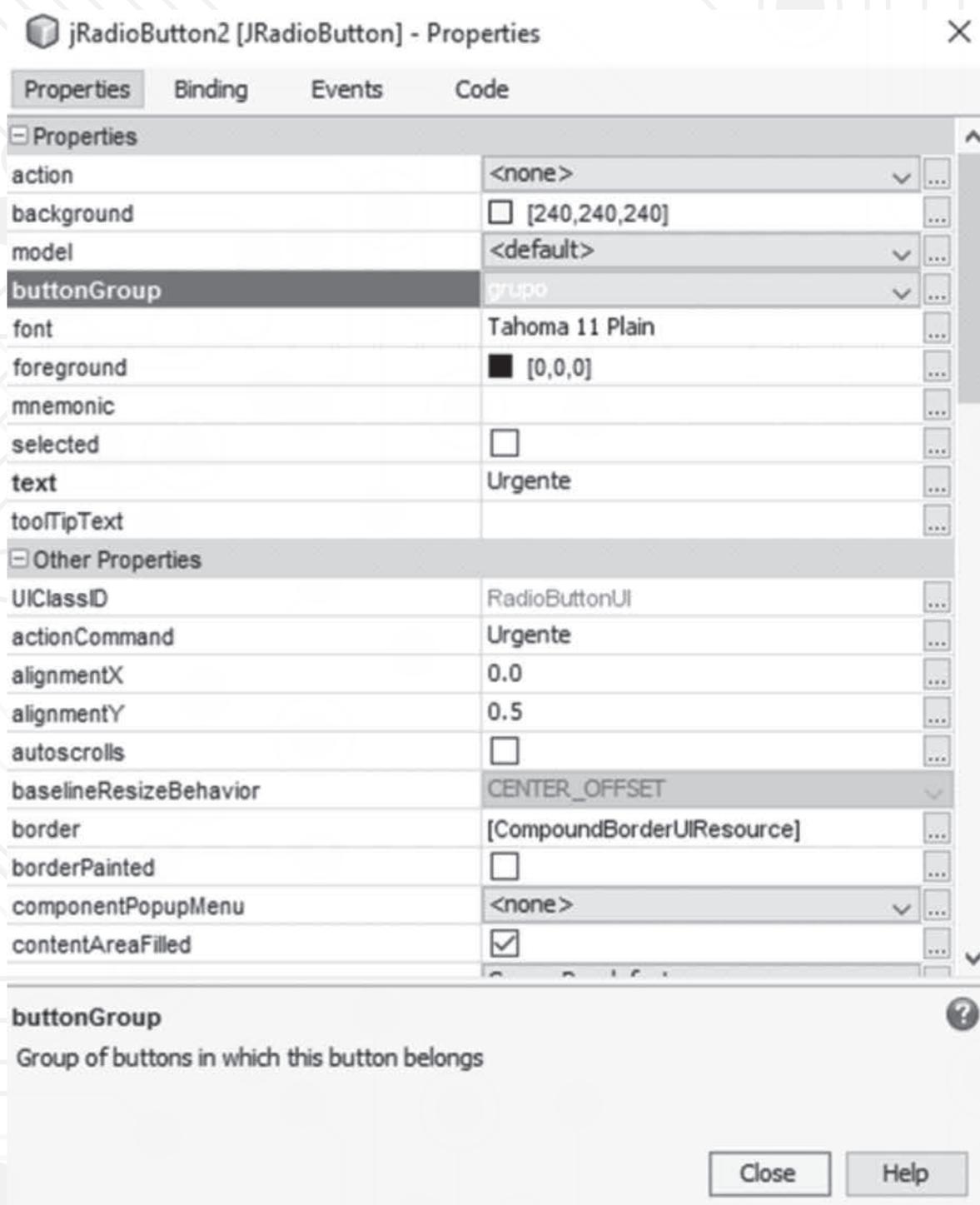
Property	Value
editable	<input type="checkbox"/>
background	[240,240,240]
columns	0
document	<default>
font	Tahoma 11 Plain
foreground	[0,0,0]
horizontalAlignment	LEADING
text	
toolTipText	
Other Properties	
UI	<default>
UIClassID	TextFieldUI
action	<none>
alignmentX	0.5
alignmentY	0.5
autoscrolls	<input checked="" type="checkbox"/>
baselineResizeBehavior	CENTER_OFFSET
border	[XPFillBorder]
caret	<default>
caretColor	[0,0,0]
caretPosition	0

editable
(boolean) specifies if the text can be edited

Close Help

Se adiciona un ButtonGroup que se llamará grupo, después los dos radioButtons para el tipo de envío.

Se debe seleccionar el grupo para los radioButton en la parte de propiedades.



Clase Principal.java



Se debe importar las clases del paquete *envíos*

```
package gui;
```

```
import envios.*;
```

```
public class Principal extends javax.swing.JFrame {
```

```
    /**
     * Creates new form Principal
     */
    public Principal() {
        initComponents();
    }
```

Método el botón calcular total

```
private void botonActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    String destino = listaCiudades.getSelectedItemAt().toString();
    System.out.println(destino);
    double peso = Double.parseDouble(inPeso.getText());
```

```
    Envio encomienda;
    if (tipoUrgente.isSelected()) {
        encomienda = new Urgente(destino);
    } else {
        encomienda = new Normal(destino);
    }
```

```
    salida.setText(Double.toString(encomienda.costoTotal(peso));
```

```
}
```

Ejemplo del programa funcionando:

The screenshot shows a Java Swing window titled "Ejemplo Polimorfismo". It features a dropdown menu for "Destino" with "Bogotá" selected, a text field for "Peso" containing the value "5", and radio buttons for "Tipo" with "Normal" selected and "Urgente" unselected. Below these is a button labeled "Calcular Costo Total". At the bottom, there is a text field for "Total" displaying the value "20000.0".

Importante: Adicionar el botón Autor para la entrega.

5. Bibliografía

Deitel & Deitel. (2016) Décima Edición. ¿Cómo programar en Java?.

Elaborado por: Docente Ing. John Vargas

PRÁCTICA: PROCESOS LENGUAJES DE PROGRAMACIÓN

1. Objetivo

Conocer sobre la creación de procesos concurrentes en Java y la sincronización de recursos.

2. Marco Teórico

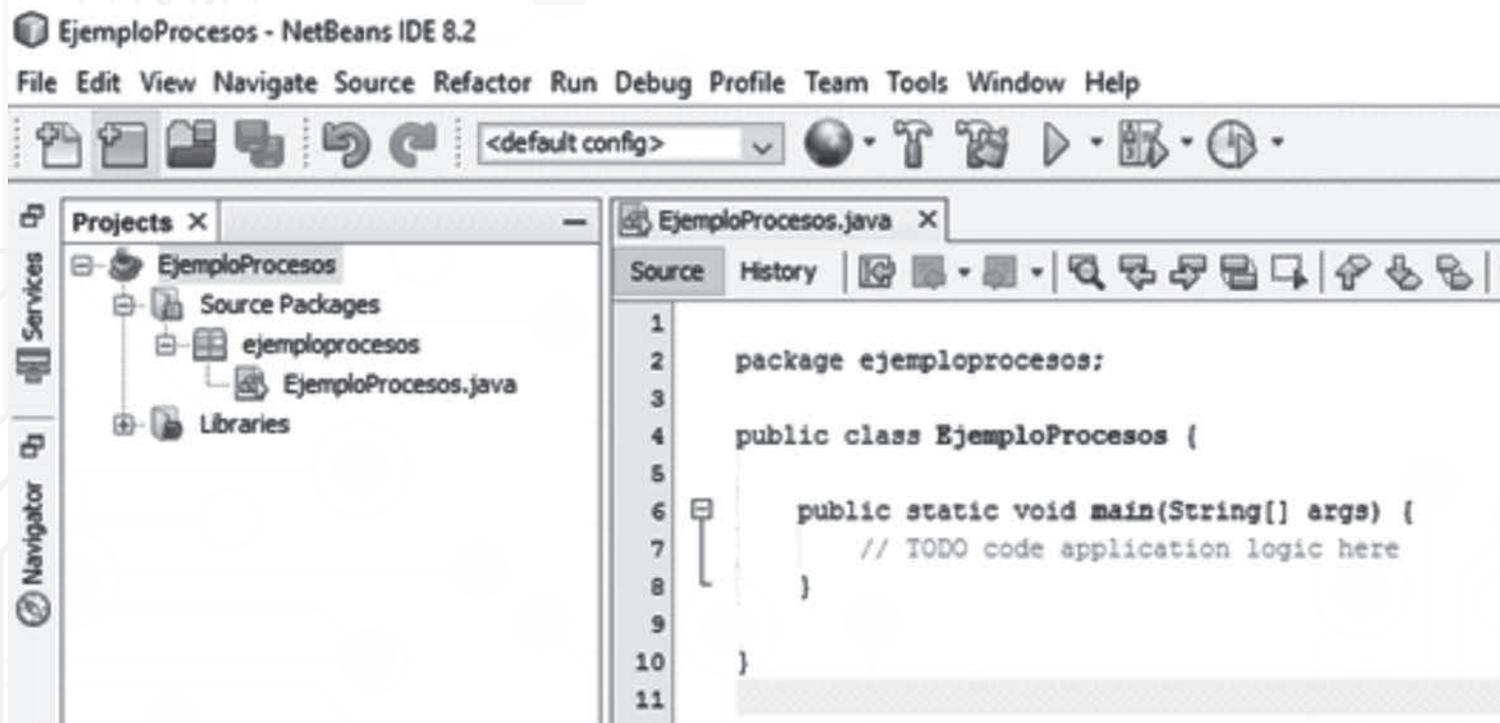
En esta actividad, utilizará las clases del API de Java para la creación de procesos concurrentes y sincronización de información compartida. Entenderá cómo usar subprocesos.

3. Material y equipo

- Guía de laboratorio
- Google Site del curso
<https://sites.google.com/a/emavirtual.edu.co/SistemasOperativos/>
- Netbeans
- JDK de Java

4. Actividades, resultados y conclusiones

1. Crear un nuevo proyecto llamado EjemploProcesos en Netbeans, tipo Java -> Java Application.



Clase Mesa:

```
package ejemploprocesos;

public class Mesa {

    int cantidadCajas;

    public Mesa()
    {
        cantidadCajas = 0;
    }

    public void colocarCaja()
    {
```

```

        cantidadCajas = cantidadCajas + 1;
    }

    public void tomarCajas()
    {
        while (cantidadCajas < 2){
            ;
        }

        cantidadCajas = cantidadCajas - 2;
    }
}

```

Clase Mensajero:

```

package ejemploprocesos;

public class Mensajero extends Thread {

    Mesa mesa;

    public Mensajero(Mesa m) {
        mesa = m;
    }

    @Override
    public void run() {
        while (true) {
            System.out.println("==> MENSAJERO toma dos cajas de la mesa");

            mesa.tomarCajas();
        }
    }
}

```

Clase Cocinero:

```

package ejemploprocesos;

public class Cocinero extends Thread {

    Mesa mesa;

    public Cocinero(Mesa m) {
        mesa = m;
    }

    public void run() {
        while (true) {
            int tiempoCocinando = (int) (Math.random() * 10);
            System.out.println("Cocinero preparando comida por " + tiempoCocinando + " segundos");
            try {
                sleep(tiempoCocinando * 1000);
            } catch (InterruptedException e) {
            }
            System.out.println("Cocinero trae una caja a la mesa");
            mesa.colocarCaja();
        }
    }
}

```

Clase Principal:

```
package ejemploprocesos;

public class EjemploProcesos {

    public static void main(String[] args) {
        Mesa mesa = new Mesa();

        Cocinero procesoCocinero = new Cocinero(mesa);
        Mensajero procesoMensajero = new Mensajero(mesa);

        procesoCocinero.start();
        procesoMensajero.start();
    }
}
```

run:

```
Cocinero preparando comida por 6 segundos
Cocinero trae una caja a la mesa
Cocinero preparando comida por 2 segundos
Cocinero trae una caja a la mesa
Cocinero preparando comida por 0 segundos
Cocinero trae una caja a la mesa
Cocinero preparando comida por 3 segundos
```

Ya hay más de 2 cajas en la mesa, y el mensajero aún no coge una caja.

Se modifica la clase Mesa, para hacer la variable compartida sincronizada:

```
public synchronized void colocarCaja()
{
    cantidadCajas = cantidadCajas + 1;
    notify();
}

public synchronized void tomarCajas()
{
    while (cantidadCajas < 2){
        try {
            wait();
        } catch( InterruptedException e ) {
            ;
        }
    }

    cantidadCajas = cantidadCajas - 2;
}
```

```
run:  
Cocinero preparando comida por 6 segundos  
Cocinero trae una caja a la mesa  
Cocinero preparando comida por 2 segundos  
Cocinero trae una caja a la mesa  
==> MENSAJERO toma dos cajas de la mesa  
Cocinero preparando comida por 0 segundos  
Cocinero trae una caja a la mesa  
Cocinero preparando comida por 4 segundos  
Cocinero trae una caja a la mesa  
==> MENSAJERO toma dos cajas de la mesa  
Cocinero preparando comida por 1 segundos  
Cocinero trae una caja a la mesa  
Cocinero preparando comida por 3 segundos  
Cocinero trae una caja a la mesa  
==> MENSAJERO toma dos cajas de la mesa
```

De esa manera por cada dos cajas, el mensajero las toma.

5. Bibliografía

Deitel & Deitel. (2016) Décima Edición. ¿Cómo programar en Java?

Elaborado por: Docente Ing. John A. Vargas



4.

BASES DE DATOS

A través de las prácticas de laboratorio de “Bases de datos 1” los estudiantes podrán comprender la importancia de tener los datos organizados y relacionados entre sí, información que es recolectada y explotada por los sistemas de información de las organizaciones.

Por medio de estas prácticas de laboratorio los estudiantes conocerán las características principales de los sistemas de bases de datos como son: la independencia lógica y física de los datos, redundancia mínima, integridad, seguridad, consultas, respaldo y recuperación de los datos.

PRÁCTICA: FUNCIONES BASE DE DATOS EN EXCEL BASE DE DATOS I – LABORATORIO

1. Objetivos:

- Reconocer el concepto de base de datos
- Identificar sus elementos
- Usar las funciones de bases de datos en Excel

2. Marco Teórico

Una base de datos es una colección de datos organizados en registros (filas) y en campos (columnas). Un registro contiene toda la información perteneciente a un elemento (ítem), por ejemplo: los datos de un alumno. Un campo contiene un dato perteneciente al elemento, por ejemplo: su nombre, edad, dirección, cursos, notas, etc.

Una base de datos es toda la colección de registros, por ejemplo: cuadro de la malla académica semestral de un alumno en Excel.

3. Material y equipo

- Laboratorio
- Computador por alumno
- Blackboard
- Guía de laboratorio

4. Actividades, resultados y conclusiones

Paso 1. Digitar la siguiente información, teniendo en cuenta las siguientes condiciones:

- La primera fila deberá definir los títulos o nombres de los campos
- El título o nombre de un campo puede contener hasta 255 caracteres.
- No es válido introducir espacios en blanco al empezar un dato en una celda
- Cada columna debe usar el mismo formato
- Tener en cuenta los caracteres en minúsculas o mayúsculas, debido a que Excel los diferencia cuando se efectúan ordenamientos
- Se pueden usar fórmulas para calcular valores de campo.g
- No puede haber columna o filas en blanco entre los datos
- Los valores numéricos no deben llevar punto o coma, esto se establece con el formato de celda.

Apellidos	Nombres	Sexo	Fecha de nacimiento	Estado civil	Zona	Sueldo base	Comisión	Gastos	Total Ingreso
Posada Álvarez	Estefanía	Mujer	20/03/1964	Otros	Centro	3.450.000			
Villarrica Márquez	Walter	Mujer	13/12/1965	Otros	Norte	2.530.000			
Ayala Ferreras	Juan	Hombre	20/07/1969	Otros	Oeste	3.450.000			
Báez Tejado	Jorge	Hombre	23/07/1965	Separado/a	Norte	4.370.000			
Bastidas Soto	Marco	Hombre	16/10/1964	Soltero/a	Norte	2.530.000			
Viáfara Méndez	Josep	Hombre	16/11/1958	Casado/a	Este	1.150.000			
Pérez Jiménez	Esther	Mujer	20/03/1967	Divorciado	Norte	5.750.000			
Vallado Gómez	Laura	Mujer	17/09/1966	Casado/a	Sur	1.230.000			
Raya García	Raquel	Mujer	10/10/1967	Casado/a	Este	1.840.000			
Cruz Molina	Juan	Hombre	29/05/1960	Otros	Este	2.760.000			
Baldes Romero	María Isabel	Mujer	11/03/1968	Divorciado/a	Oeste	1.840.000			
Cuéllar Gómez	Adriana	Hombre	23/03/1964	Soltero/a	Norte	690.000			
López Borrero	Gerardo	Hombre	13/01/1966	Divorciado	Norte	920.000			
Moreno Garcés	Eliot	Hombre	02/09/1957	Soltero/a	Sur	7.130.000			
Galvis Carrero	Jordi	Hombre	18/07/1969	Casado/a	Oeste	2.300.000			
Zamudio Figueroa	Luis	Hombre	30/01/1969	Soltero/a	Oeste	2.530.000			
Morales Parra	Laura	Mujer	02/04/1967	Casado/a	Este	1.150.000			
Zafra Paz	Jordi	Hombre	03/09/1958	Otros	Norte	1.380.000			
Blanco Polonia	Dona	Mujer	31/05/1966	Separado/a	Centro	920.000			
García Cifuentes	Julio	Hombre	21/09/1967	Otros	Sur	2.530.000			

Paso 2. Desarrolle los siguientes puntos:

- Identifique cuántos atributos existen en la lista de datos
- Calcule las celdas de Comisión siendo el 7% de las ventas, Gastos siendo el 1,2% de las ventas y el Total de ingresos es igual al Sueldo base más la comisión menos los gastos.
- Ordenar los datos por Nombre de forma ascendente
- Ordenar los datos por Zona de forma descendente
- Filtrar los datos, y muestre solo las mujeres

Paso 3. Práctica con funciones de Base de Datos

Excel posee un conjunto de funciones que comienzan con los iniciales BD y que sirven para trabajar con bases de datos.

BDfunción(base_de_datos;nombre_de_campo;criterios)

Base de datos= Rango de casillas o nombre de la tabla.

Nombre de campo= Columna donde se encuentran los datos sobre los que se realiza la operación.

Criterio= Rango donde se encuentra el criterio (previamente indicado en forma de tabla).

Paso 4. Digite la siguiente información:

Nombre	Edad	Sexo	Grado	Calificación	Tipo de Evaluación
Emmanuel Rodríguez	16	M	1	9	Extraordinario
Jonatan Vásquez	17	M	2	7	Ordinario
Abraham Gáitan	20	M	3	8	Ordinario
Paola Linderos	16	F	1	8	Extraordinario
Alejandra Bautista	20	F	3	8	Extraordinario
Adolfo Ferruzca	18	M	2	7	Ordinario
Rolando Campos	18	M	3	6	Ordinario
Daniela Peña	16	F	1	7	Extraordinario

Grabar el archivo con el nombre de Calificaciones.XLS

Con las funciones de base de datos genere:

- Cuántas son las personas menores de edad
- La calificación más alta del personal masculino
- La calificación más baja del personal femenino
- El promedio de notas del grado 1, 2 y 3
- Cuánta evaluación extraordinaria hubo
- Ordene la información por el nombre de forma descendente
- Filtre la información por calificación=7
- Al filtro anterior aplíquelo otro filtro por sexo=m
- Ordene la información por el grado y calificación
- Contar el total de alumnos con evaluación ordinaria

5. Bibliografía

- Soporte Microsoft Office 2016

<https://support.office.com/es-es/article/funciones-de-base-de-datos-referencia-ad87e69b-fc20-4d3d-9d52-d7dc023f5c23>

Elaborado por: Docente Ing. Sofi Lorena Riocampo, Ms.C.

PRÁCTICA: BASE DE DATOS EN ACCESS – MER – MODELO RELACIONAL BASE DE DATOS I – LABORATORIO

1. Objetivos:

- Crear una base de datos en Access
- Identificar sus elementos
- Diseñar el diagrama Entidad Relación
- Realizar el Modelo Relacional

2. Marco Teórico

Los Sistemas Gestores de Base de Datos, permiten que unos conjuntos de datos se interrelacionen con el fin de poder acceder a los datos de manera ordenada y eficiente; las aplicaciones de los sistemas de base de datos pueden ser: Banca, Líneas aéreas, Universidades, Ventas, Producción, Recurso humano, entre otras.

3. Material y equipo

- Laboratorio
- Computador por alumno
- Blackboard
- Microsoft Office - Access
- Guía de laboratorio

4. Actividades, resultados y conclusiones

Para cada ejercicio debe construirse el MER, la Base de Datos y el Modelo Relacional

1. Ejercicio No. 1

2. INSTITUTO DE COMPUTACIÓN

1. Cree una base de datos nueva con el nombre: Instituto_Computer_(Apellido estudiante)
2. Cree la siguiente tabla:

Tabla: Estudiante

Campo	Tipo	Tamaño/Formato	Observaciones
Matrícula	Autonumérico		Clave Principal
Apellidos	Texto	25	Apellidos completos
Nombres	Texto	25	Primer y Segundo nombre
Sexo	Texto	1	
Edad	Número	2	
Curso	Texto	14	

3. Agregue datos a la tabla de acuerdo a la siguiente información:

Matrícula	Nombre Completo	Sexo	Edad	Curso
1	Amalia María Fernández Torres	F	16	Access
2	Rosario Lucía Céspedes Martínez	F	15	Word
3	Carmen Zavaleta Arauca	F	16	Windows
4	Rocío María Ferrer Lima	F	17	Word
5	Antonio Martín Perales Lizarazu	M	16	Windows
6	Bernardo Martín Carmona Montalván	M	17	Access
7	Luis Alberto Perales Lizarazu	M	15	Word
8	Daniel Cardona Román	M	20	Access
9	Ana Perales Lizárraga	F	16	Windows
10	Jorge Luis Choque Chambi	M	21	Access
11	Pedro Martín Rojas Pérez	M	16	Windows
12	Juan Carlos Perales Lizarazu	M	14	Access
13	Gilberto Farfán Morales	M	18	Word
14	Yolanda María Reyes Copa	F	15	Windows

A continuación, se plantean dos ejercicios en los cuales se debe diseñar el MER y posteriormente crear en Access las tablas necesarias de acuerdo al diseño e introducir por lo menos 10 registros en cada una; así mismo identificar los campos claves y generar el Modelo Relacional.

Ejercicio No. 2

Una compañía de seguros de coches cuyos clientes poseen uno o más coches. Cada coche tiene asociado un número de cero a cualquier valor que almacena el número de accidentes.

Ejercicio No. 3

Un hospital con un conjunto de pacientes médicos. Así mismo con cada paciente un registro de las diferentes pruebas y exámenes realizados.

5. Bibliografía

- Fundamentos de Base de datos, Silberschatz, Korth y Sudarshan

Elaborado por: Docente Ing. Sofi Lorena Riocampo, Ms.C.

PRÁCTICA: COMANDOS SQL BASE DE DATOS I – LABORATORIO

1. Objetivos:

- Conocer los comandos SQL
- Identificar la estructura de cada comando para su uso
- Practicar a través de ejercicios

2. Marco Teórico

SQL (Structured Query Language) es un lenguaje estándar e interactivo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas, con la utilización del álgebra y de cálculos relacionales, el SQL brinda la posibilidad de realizar consultas de información en las bases de datos de manera sencilla. Las consultas se realizan a través de comandos que permiten seleccionar, insertar, actualizar, averiguar la ubicación de los datos, etc. Cuando se usa SQL, debe emplear la sintaxis correcta. La sintaxis es el conjunto de reglas mediante las que se combinan correctamente los elementos de un idioma. La sintaxis SQL se basa en la sintaxis del idioma inglés.

3. Material y equipo

- Laboratorio
- Computador por alumno
- Microsoft Office - Access
- Guía de laboratorio

4. Actividades, resultados y conclusiones

Cláusula **SELECT** y **FROM**

- La cláusula **SELECT** corresponde a la operación de proyección del álgebra. Se utiliza para dar la relación de los atributos deseados en el resultado de una consulta.
- La cláusula **FROM** corresponde a la operación del producto Cartesiano del álgebra relacional. Hace una lista de las relaciones que se van a explorar en la evaluación de la expresión.

Sintaxis

SELECT <<nombre del campo o campos separado por comas>>

FROM <<nombre de la tabla>>

Cláusula **WHERE**

- La cláusula **WHERE** corresponde al predicado de la selección del álgebra relacional. En este comando se utilizan los condicionales lógicos and, or y not; los operadores de comparación Between.

Sintaxis

SELECT <<nombre del campo o campos separado por comas>>

FROM <<nombre de la tabla>>

WHERE <<condición>>

Cláusula **INSERT INTO**

- Añadir una nueva tupla a una tabla

Sintaxis

INSERT INTO <<nombre table>>

VALUES (valores de cada campo separados por coma)

Cláusula **UPDATE**

- Modifica o actualiza datos en una tabla

Sintaxis

UPDATE <<nombre de la tabla>>

SET <<nombre del campo a actualizar>> <<criterio de actualización>>

Cláusula DELETE

- Elimina registros de una tabla

Sintaxis

DELETE FROM <<nombre tabla>>

WHERE <<condición>>

Funciones de agregación

- Estas funciones operan en el multiconjunto de valores de una columna de una relación, y devuelven un valor

avg: valor promedio

min: valor mínimo

max: valor máximo

sum: suma de valores

count: número de valores

Sintaxis

SELECT <<función>> (campo)

FROM <<nombre de la tabla>>

Paso No. 1

- Descargar la base de datos VENTAS, que se encuentra en Actividades – Sentencia SQL Práctica individual.
- Abrir la base de datos de VENTAS
- Realizar las siguientes consultas y grabarlas con el número en letras de cada una, ejemplo: Primera, Segunda, Tercera, etc.

De la tabla CLIENTES:

ID_Cliente	Nombre_empresa	Nombre_contacto	Cargo_contacto	Direccion	Ciudad	Region	Código_pos	País	Telefono	Fax	Limite_cred
ALWAO	Mercado Siempre	Montse Alcaraz	Ejecutivo de Cuentas	Avenida del Rio, 76	Pamplona	NA	84804	E	(801) 555-7834	(801) 555-8853	\$2.500.000,00
ANHE	Colmado Continer	Helena Giménez	Ejecutivo de Cuentas	Código Postal 08004	Pontevedra	PO	98228	E	(206) 555-9574	(206) 555-3541	\$1.000.000,00
ANHE	Cerveceria Mano	María Torres	Asistente Ejecutivo C	Calle Itegrudo, 32	Córdoba	CO	24422	E	(509) 555-8647		\$1.000.000,00
AROUT	Around the Horn	Thomas Hardy	Ejecutivo de Cuentas	Brook Farm	Colchester	Essex	CO7 6JX	UK	(71) 555-7788	(71) 555-6750	\$1.000.000,00
BABUJ	Babu Ji's Exports	G.K.Chatterjee	Propietario	Box 29938	Londres		W11 5LT	UK	(71) 555-8248		\$2.000.000,00
BERGS	Colmado Escandin	Tomás Zamora	Administrador de Pe	Calle Martín, 54	Sevilla	SE	98104	E	(206) 555-3453	(206) 555-8832	\$750.000,00
BLUEL	La Tienda del Lago	Helena Muros	Propietario	Calle Mayor, 243	Palencia	PA	98368	E	(206) 555-3044	(206) 555-4247	\$2.500.000,00
BLUMG	Blum's Goods	Pat Parkes	Director de Márketin	The Blum Building	Londres		NW1 2BP	UK	(71) 555-3013		\$2.000.000,00
BOBCM	Regalos Al-Andalu	Gonzalo López	Director de Márketin	Calle Quevedo, 98	Sevilla	SE	98124	E	(206) 555-4747		\$750.000,00
BOITM	Mercancia Barata	Elisabet López	Director de Contabili	Paseo Thyssen, 13.	Tenafé	TE	T2F 8M4	Canada	(604) 555-4729	(604) 555-3745	\$3.000.000,00
BSBEV	B's Beverages	Victoria Ashworth	Ejecutivo de Cuentas	Fauntleroy Circus	Londres		EC2 5NT	UK	(71) 555-1212		\$2.000.000,00
CACTP	El Súper de Monts	Montse Sabrigues	Comercial	Calle Doctor Lemán	Bilbao	BI	47123	E	(505) 555-2953	(505) 555-9987	\$1.000.000,00
CAESM	Importaciones del	Olivia Lamont	Director de Márketin	Avenida Norte, 56	Salamanca	SA	94985	E	(413) 555-8840	(413) 555-4843	\$750.000,00
CHEAC	Carnicería Luque	Luisa Sevillano	Ejecutivo de Cuentas	Calle Principal, 52	Ávila	AV	98221	E	(206) 555-8647	(206) 555-9928	\$1.000.000,00
COMME	Commoner's Exchs	Terry Hargreaves	Socio de Ventas	Exchange House	Londres		EC1 5WJ	UK	(71) 555-8888	(71) 555-1154	\$2.000.000,00
CONSH	Consolidated Holc	Elizabeth Brown	Ejecutivo de Cuentas	Berkeley Gardens	Londres		W11 6LT	UK	(71) 555-2282	(71) 555-9399	\$2.000.000,00
DOLLE	Grandes Almacene	Sonia Ondarrubia	Asistente Ejecutivo C	Calle del Parque, 92	Salamanca	SA	94103	E	(413) 555-7357		\$750.000,00
DUNNH	Dunn's Holdings	Sylvia Dunn	Propietario	The Dunn Building	Londres		SW1 2XE	UK	(71) 555-9444	(71) 555-5393	\$2.000.000,00
EASTC	Eastern Connectio	Ann Devon	Comercial	35 King George	Londres		W13 6FW	UK	(71) 555-0297	(71) 555-3373	\$2.000.000,00
EMRPT	Empire Trading	Ronald Merrick	Director Comercial	7 Baxter Hill	Londres		SW2 1HQ	UK	(71) 555-3838		\$2.000.000,00
FAMIC	Comida Familiar	Ana Calleja	Asistente de Márketi	Avenida Genil, 76	Granada	GR	43023	E	(509) 555-9938	(509) 555-4733	\$2.000.000,00
FITZD	Videos, etc., S.A.	Susana Martinez	Asistente de Márketi	Paseo de la Costa, 9	Alicante	A	98006	E	(206) 555-5575	(206) 555-7824	\$1.000.000,00

- Mostrar solo los nombres de contacto.
- Mostrar el nombre de las empresas que tienen un límite de crédito mayor a 1'000.000.
- Realizar una consulta que devuelva el nombre de la empresa, nombre del contacto para aquellos que se encuentran en el país UK y la ciudad sea Londres.
- Mostrar los nombres y dirección para las personas con el cargo de ejecutivo de cuenta.
- Mostrar los nombres de las empresas y el contacto para aquellas que tienen un Ejecutivo de Cuentas y ordénelas por el contacto.

6. Contar el número de límites de crédito mayor a 1´500.000
7. Encontrar el límite de crédito más alto y el más bajo.
8. Generar una consulta con el nombre_empresa, nombre_contacto para todos clientes que no tienen asignada una región.
9. Mostrar los clientes con un límite de crédito menor a 1´000.000 y ordenarlos por nombre de la empresa.
10. Generar una consulta de los clientes de la ciudad de Londres y el cargo del contacto sea ejecutivo de cuenta.
11. Añadir cuatro tuplas a la base de datos.
12. Cambiar todos los países identificados con E por EU.
13. Elimine los registros que no tengan número de Fax.

Paso No. 2: Establezca las posibles relaciones entre las tablas existentes y explique de forma enunciativa cada una. Presentar la pantalla con las relaciones.

Paso No. 3: Grabar la nueva base de datos modificada agregándole su APELLIDO al final.

5. Bibliografía

- Soporte Microsoft Office 2016

<https://support.office.com/es-es/article/access-sql-conceptos-b%C3%A1sicos-vocabulario-y-sintaxis-444d0303-cde1-424e-9a74-e8dc3e460671>

- Fundamentos de Base de datos, Silberschatz, Korth y Sudarshan

Elaborado por: Docente Ing. Sofi Lorena Riocampo, Ms.C.

5.

INFORMÁTICA

A través de las prácticas de laboratorio de “Informática I” los estudiantes podrán aplicar los conceptos vistos en clases acerca de esta rama de la Ingeniería, la cual estudia el hardware, las redes de datos y el software, elementos necesarios para tratar información de forma automática.

Actualmente, es difícil concebir un área que no use, de alguna forma, el apoyo de la informática; en ella convergen la programación, las metodologías para el desarrollo de software, la arquitectura de los computadores, las redes de computadores, la inteligencia artificial y los fundamentos de las ciencias de la computación.

Esta disciplina se aplica a numerosas y variadas áreas del conocimiento o la actividad humana, facilitando la toma de decisiones a nivel gerencial, como permitir el control de procesos críticos.

PRÁCTICA: ALGORITMOS SECUENCIALES INFORMÁTICA I – LABORATORIO No. 1

1. Objetivos:

- Desarrollar la habilidad de realizar un procedimiento para la solución de un problema a través de un algoritmo.
- Realizar diagramas de flujo.
- Traducir el diagrama de flujo en un pseudocódigo

2. Marco Teórico

La definición de problema de acuerdo con el Diccionario de RAE, es el “Planteamiento de una situación cuya respuesta desconocida debe obtenerse a través de métodos científicos”, por lo tanto, un algoritmo es una secuencia finita de operaciones realizables, no ambiguas, cuya ejecución da una solución a un problema.

3. Material y equipo

- Laboratorio
- Computador por alumno
- Blackboard
- Microsoft Office - Word

4. Actividades, resultados y conclusiones

Paso No. 1

Escriba un algoritmo para las siguientes situaciones:

- Preparar un emparedado con crema de maní y mermelada
- Transacciones de un cajero automático simple (consignar y retirar)
- Calcular el doble de un número
- Calcular una altura en pulgadas (1 pulgada=2,54 cm), a partir de la altura en centímetros.

Paso No. 2

Implementar el pseudocódigo para los algoritmos anteriores.

Paso No. 3

Realizar los diagramas de flujo de las soluciones propuestas.

5. Bibliografía

- Introducción a la Informática, Ramón A. Mata Toledo y Pauline K. Cushman, 2001

Elaborado por: Docente Ing. Sofi Lorena Riocampo, Ms.C.

PRÁCTICA: ESTRUCTURAS DE DECISIÓN INFORMÁTICA – LABORATORIO No. 2

1. Objetivos:

- Identificar la instrucción sintaxis para aplicar una estructura de decisión
- Aprender la manera de aplicar la toma de decisiones
- Establecer qué instrucciones seguir en cualquiera de los DOS casos, Verdadero ó Falso.

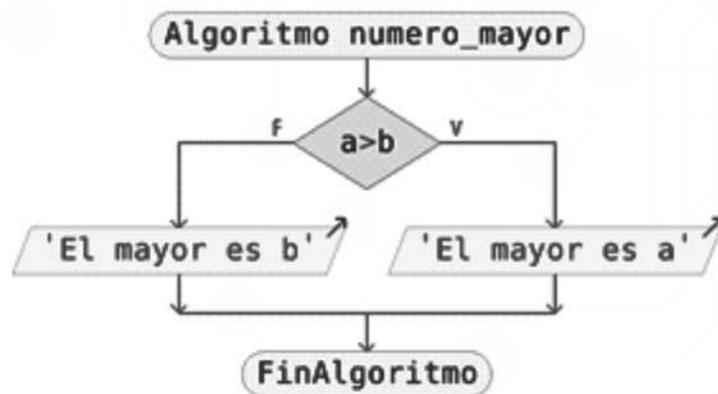
2. Marco Teórico

Dentro de las estructuras de control encontramos los condicionales que se traducen dentro de un lenguaje de programación como una expresión lógica de Falso o Verdadero, siendo en la mayoría de los casos una comparación entre dos datos. En el caso del PSeInt su estructura es:

Sí expresión_lógica Entonces
 acciones_por_verdadero

SiNo
 acciones_por_falso

Fin Sí



3. Material y equipo

- Laboratorio
- Computador por alumno
- Blackboard
- Programa PSeInt

4. Actividades, resultados y conclusiones

Paso No. 1 – Condicionales Simples

Escriba un algoritmo en PSeInt para:

- Hallar el número mayor entre dos números.
- Se pide la edad del usuario; si es mayor de edad debe aparecer un mensaje indicándolo.
- Dado como dato la calificación de un alumno en un examen, escriba “Aprobado” en caso que esa calificación fuese mayor que 8.
- Ingresar el nombre de un empleado, las horas trabajadas, luego Calcular pago bruto (\$5000 la hora), Seguro Social y total a pagar, presentar los resultados del programa. El seguro social es \$12.300 si el sueldo es mayor \$24.000 sino es el 4,5% del sueldo del empleado.

Paso No. 2 – Condicionales Dobles

Escriba un algoritmo en PSeInt para:

- Leer 2 números; si son iguales que los multiplique, si el primero es mayor que el segundo que los reste y si no que los sume.
- Determinar la cantidad de dinero que recibirá un trabajador por concepto de las horas extras trabajadas en una empresa, sabiendo que cuando las horas de trabajo exceden de 40, el resto se consideran horas extras y que estas se pagan al doble de una hora normal cuando no exceden de 8; si las horas extras exceden de 8 se pagan las primeras 8 al doble de lo que se pagan las horas normales y el resto al triple.

Paso No. 3 – Condicionales Múltiples

Escriba un algoritmo en PSeInt para:

- a. Lea de entrada 3 números y que indique cuál es el mayor de ellos.
- b. Se ingresa por teclado un valor entero, mostrar una leyenda que indique si el número es positivo, nulo o negativo.
- c. Calcular la nota definitiva sabiendo que las tres notas tienen los siguientes valores: Primera nota 30%, segunda nota 40% y tercera nota 30%. Además, mostrar si el estudiante aprobó o no la materia teniendo en cuenta que se aprueba con una nota mayor o igual a 3.
- d. Confeccionar un programa que permita cargar un número entero positivo de hasta tres cifras y muestre un mensaje indicando si tiene 1, 2, o 3 cifras. Mostrar un mensaje de error si el número de cifras es mayor.

5. Bibliografía

- Material de apoyo Fernando Palmero
- Material docente Universidad de Sevilla

Elaborado por: Docente Ing. Sofi Lorena Riocampo, Ms.C.

PRÁCTICA: CICLOS, ARREGLOS Y MATRICES INFORMÁTICA – LABORATORIO No. 3

1. Objetivos:

- Aplicar el uso de ciclos repetitivos para la resolución de problemas
- Conocer el concepto de arreglos y plantear la solución con el ciclo Para.
- Practicar el empleo de ciclos para trabajar arreglos y matrices

2. Marco Teórico

Los ciclos son también llamados iteraciones, se usan en programación para ejecutar el mismo conjunto de instrucciones hasta que se cumpla cierta condición. Las instrucciones diseñadas para el manejo de ciclos son: Para (For), Mientras (While), Repetir hasta (Do – While).

Un arreglo es una zona de almacenamientos que contiene una serie de elementos del mismo tipo, y para almacenar los elementos se utiliza comúnmente los ciclos repetitivos, pero en esta ocasión utilizaremos el ciclo PARA.

3. Material y equipo

- Laboratorio
- Computador por alumno
- Blackboard
- Programa PSeInt

4. Actividades, resultados y conclusiones

Realizar los siguientes algoritmos en PSeInt:

- Hacer un algoritmo que lea 40 números e imprima cuántos son positivos, cuántos negativos y cuántos son neutros, además la sumatoria de los positivos y negativos.
- Una persona desea realizar un muestreo con 70 personas para determinar el promedio de peso de los niños, jóvenes, adultos y viejos que existen en una zona habitacional y cuántos son de cada una de las categorías.
 - Niño → 0 a 13 años
 - Jóvenes → 14 a 30 años
 - Adultos → 31 a 60 años
 - Viejos → 61 en adelante
- Hacer un algoritmo que imprima los números del 20 al 10, en orden decreciente.
- Hacer un algoritmo que imprima todos los números naturales que hay desde la unidad hasta un número que introducimos por teclado.
- Cree un arreglo de 5 posiciones y llénelo con los números que el usuario desee, sume los elementos del arreglo y al final imprima el vector y la suma total.
- Desarrolle un programa que permita leer el nombre de 10 productos de una tienda y su correspondiente precio. Al final debe decir cuántos de ellos cuestan más de \$3000, mostrar su nombre y precio.
- Los siguientes ejercicios tienen como propósito que usted escriba ciclos que recorran la matriz completa o partes de ella. Suponga que se ha definido una constante positiva entera N y una matriz mat, de dimensión NxN.
 - a. Escriba un algoritmo que ponga cero en ambas diagonales de la matriz.
 - b. Escriba un algoritmo que ponga cero en la primera y la última fila, y en la primera y la última columna de la matriz.
- Escriba un algoritmo que llene de números la matriz de tal forma que $mat[i][j]$ sea igual a $i+j$.
- Escriba un algoritmo que llene la diagonal principal de la matriz con los números 1,2,3,...N. La diagonal principal de una matriz está formada por las casillas en las cuales el índice de fila y de columna son iguales.
- Escriba un algoritmo que llene todas las filas pares con los números 1,2,3,...N, y las filas impares con los números N,N-1,N-2,...1.

5. Bibliografía

- Universidad Autónoma del Estado de Hidalgo.
www.uaeh.edu.mx/docencia/P_Presentaciones/icbi/ asignatura/Ciclos.pdf
- Repositorio Institucional Continental.
<http://repositorio.continental.edu.pe/handle/continental/1753>
- Material de apoyo Fernando Palmero. <http://aprendizajealgoritmos.blogspot.com/search/label/Actividad%20No.%204%20-%20Estructuras%20Repetitivas%20o%20C%C3%ADclicas>.

Elaborado por: Docente Ing. Sofi Lorena Riocampo, Ms.C.

6.

REFERENCIAS

1. international Earth Rotation Service and Reference Systems Service (s.f.). Recuperado de: <http://www.iers.org>
2. International Terrestrial Reference Frame (s.f.). Recuperado de: <http://itrf.ensg.ign.fr>
3. PROVENZA Guiu José Luis. El sistema de posicionamiento global 4769019.pdf (s.f.). Recuperado de: <https://dialnet.unirioja.es/descarga/articulo/4769019.pdf>
4. International GPS Service (s.f.). Recuperado de: <http://www.igs.org/article>
5. International Earth Rotation Service (s.f.). Recuperado de: <http://www.iers.org>
6. International Terrestrial Reference Frame (s.f.). Recuperado de: <http://itrf.ensg.ign.fr>
7. Domínguez B.J. (octubre, 2000). “Breve Introducción a la Cartografía y a los Sistemas de Información Geográfica (SIG)”.
8. Sistema de GPS (s.f.). Recuperado de: <http://www.trimble.com>
9. MORALES, AURELIO (1 julio 2016). Siete ventajas del web Mapping sobre SIG de escritorio. Recuperado de: <http://mappinggis.com/2016/03/7-ventajas-las-tecnologias-web-mapping-los-sig-escritorio/>
10. Sistema Posicionamiento Global (s.f.) Recuperado de: <http://www.trimble.com>
11. SIG Corporativo. Esri España (s.f.) Recuperado de: <http://www.esri.es/es/soluciones/defensa-e-inteligencia/sig-corporativo/>
12. Posada, Elena, IGAC, & Ramírez Héctor (2011). Manual de prácticas de percepción remota con el programa ERDAS Parte 1 IMAGE.
13. Manual ERDAS_Web. Pdf. (s.f.). Recuperado de: http://www.un-spider.org/sites/default/files/ManualERDAS_web.pdf
14. ARONOFF, S. Remote Sensing for Gis Managers(2005). California: ESRI Press.
15. Centro para detección y procesamiento remoto de imágenes. ¿Qué es la detección remota? Recuperado de: <http://www.crisp.nus.edu.sg/~research/tutorial/intro.htm>
16. INSTITUTO GEOGRÁFICO AGUSTÍN CODAZZI. Interpretación visual de imágenes de sensores remotos y su aplicación en levantamientos de cobertura y uso de la tierra. Centro de Investigación y Desarrollo de Información Geográfico, CIAF (2005)
17. Villalobos, J. A., & Casallas G., R. (2006). Fundamentos de Programación. Aprendizaje Activo Basado en Casos. Bogotá - Colombia: Pearson Prentice Hall. Tabla de Código Ascll. Recuperado de <http://ascii.cl/es/>
18. Becerra, S. César A. (2000). *Estructura de Datos en Java*. Bogotá – Colombia: Editorial Kimpres.
19. P. Julián, M. Alpuente (2007). Programación Lógica. Teoría y Práctica, Pearson Prentice Hall.

20. Deitel & Deitel. (2016) ¿Cómo programar en Java? Décima Edición.
21. PRESSMAN, Roger S. (2010). Ingeniería de software. Un enfoque práctico. 7ª edición. McGraw Hill.
22. PRESSMAN, Roger S. (1997). Ingeniería de software. Un enfoque práctico. 6ª edición. McGraw Hill.
23. SCHACH, Stephen (2004). Análisis y Diseño Orientado a Objetos con UML y el Proceso Unificado. McGrawHill.
24. Proyecto C++ . Universidad Evangélica de El Salvador. Publicado en octubre de 2007.
25. Abelson y Sussman (s.f.). Capítulo 3.3 (Modeling with Mutable Data, apartados 3.3.1-3.3.3)
26. Programming in Scala: Capítulo 7 (Built-in Control Structures) y Capítulo 5.7 (Object equality)
27. <https://lenguajemasmas.wordpress.com/2007/10/01/arreglos-bidimensionales-matrices/>
28. <http://jovanana.blogspot.com/2012/12/arreglos-unidimensionalesbidimensionale.html>
29. <http://programacionunefa.blogspot.com/2008/06/arreglos-unidimensionales.html>

