



CU  
CoSta  
Sur

UNIVERSIDAD DE GUADALAJARA  
CENTRO UNIVERSITARIO DE LA COSTA SUR  
DEPARTAMENTO DE INGENIERÍAS

## Manual de Prácticas de Laboratorio

### Computación

Laboratorio de  
Programación Avanzada

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

Elaborado por:

Ing. Alvarado Ornelas José Guadalupe  
 Ing. Avalos Delgadillo Carlos Alan  
 Mtro. Dávila Guerrero Jorge Luis  
 Mtra. Hernández Del Rio Andrea Alejandra  
 Ing. Medina Reyes Gerardo Joel  
 Ing. Ortega Estrada Gilberto  
 Dr. Karlos Emmanuel Espinoza Ramos  
 Dr. Luis Isidro Aguirre Salas

Presidente de la Academia.

**ING. JOSÉ GUADALUPE ALVARADO ORNELAS**

Responsable del Laboratorio de Programación.

**DR. KARLOS EMMANUEL ESPINOZA RAMOS**

Jefe del Departamento de Ingenierías.

**DR. DANIEL EDÉN RAMÍREZ ARREOLA**

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## **OBJETIVO GENERAL.**

El estudiante analizará, diseñará y desarrollará aplicaciones para resolver problemas de propósito general. Así mismo, realizará pruebas que verifiquen el correcto funcionamiento de los programas y que se ajustan a los requisitos de análisis y diseño.

## **CONSIDERACIONES GENERALES**

El estudiante debe cumplir con el Reglamento General de Uso de Laboratorios publicado en el “Compendio de reglamentos del Departamento de Ingeniería”.

## **SEGURIDAD E HIGIENE EN EL USO DEL LABORATORIO**

En caso de alguna contingencia (sismo, incendio o cualquier evento que ponga en riesgo su integridad) evacúe el laboratorio inmediatamente, siguiendo las normas de seguridad implementadas en los simulacros.

Así mismo es de suma importancia que las personas que hagan uso de las instalaciones de los laboratorios, conozcan las ubicaciones de los extintores, botiquines de primeros auxilios y salidas de emergencia.

Es importante resaltar los siguientes puntos referentes a la seguridad e higiene que se deben seguir para el uso de laboratorio y que se encuentran plasmados en el Reglamento Interno del laboratorio “Programación Avanzada”:

1. Mantener y dejar limpia su área de trabajo.
2. No arrojar papeles ni basura al piso.
3. No introducir alimentos y bebidas.
4. No fumar.
5. El alumno deberá dejar su mochila y/o bolsa en los estantes designados para los mismos, respetando todo objeto ajeno que allí se encuentre.
6. No utilizar dispositivos de reproducción de música y audio.
7. Está prohibido sentarse sobre las mesas de trabajo o pararse en las sillas.
8. El alumno debe comportarse siguiendo el Código de Ética de la Universidad de Guadalajara dentro de las instalaciones del laboratorio, hacer uso

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

apropiado del lenguaje oral y escrito; respetar a sus profesores, compañeras y compañeros de clase.

9. Antes de iniciar las prácticas, el maestro inspeccionará las condiciones físicas del laboratorio y de encontrar situaciones que representen riesgo grave, deberá reportar dicha situación al responsable del laboratorio y/o al asistente o auxiliar del mismo, para que sea corregida, en caso de que no exista la posibilidad de atención inmediata, la práctica será suspendida.
10. Si durante la práctica surgiera una condición que ponga en riesgo grave la seguridad y salud de las personas, equipos, materiales o instalaciones, se procederá a suspender la práctica debiendo informar de la situación al responsable de laboratorio, asistente o auxiliar del mismo, elaborando por escrito el reporte correspondiente.
11. El profesor deberá cumplir con el uso del equipo de protección personal básico de laboratorio. El maestro que no cumpla con estos requisitos no podrá realizar la práctica. El auxiliar notificará la situación al responsable de laboratorio y/o al jefe de departamento quien elaborará un reporte de faltas al reglamento.
12. Es responsabilidad del profesor verificar que antes de iniciar la práctica, todos los alumnos cuenten con el equipo de protección personal y el código de vestimenta necesario para realizar la práctica. El alumno que no cumpla con los requisitos anteriores, no podrá realizar la práctica.
13. El profesor deberá asegurarse que los alumnos utilicen adecuadamente el equipo de protección personal durante el desarrollo de la práctica.
14. El profesor llevará un registro de los alumnos que sean observados sin usar su equipo de protección personal o usándolo de manera inadecuada, cada registro contará como una falta al reglamento del laboratorio.
15. La acumulación de 4 faltas al reglamento del laboratorio, implica la suspensión para el alumno de la práctica en el semestre y la no acreditación de la misma.
16. El profesor deberá permanecer en el laboratorio durante todo el desarrollo de la práctica.
17. Por razones de seguridad y orden está prohibido en el laboratorio:

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

- Correr.
  - Usar sombrero, gorra y/o pañoleta en la cabeza.
  - Ingresar personas ajenas a la institución o al grupo que desarrolla la práctica.
  - Usar calzado inadecuado.
  - Usar el cabello largo (las personas con esta característica deberán recoger su cabello y sujetarlo adecuadamente, como medida de prevención para evitar riesgos).
  - Usar pantalón corto o bermuda.
  - Y en general todo acto y/o conducta que incite al desorden.
18. Todo alumno que sufra una lesión deberá reportarla al maestro encargado de la práctica y de no encontrarse éste, deberá dirigirse con el responsable de laboratorio y/o asistente del mismo.
19. Todo trabajador universitario que sufra una lesión deberá reportarla a su jefe inmediato.
20. Todo accidente ocurrido en los laboratorios deberá ser atendido para su control, por la primera persona capacitada y enterada de la situación.
21. Al término de la práctica, el maestro será responsable de supervisar que los alumnos ordenan y limpian su lugar de trabajo. Asegurando que el laboratorio sea entregado al encargado en condiciones óptimas.
22. La persona que se presente bajo el influjo de alcohol o drogas, que incurra en actos de violencia, daño a la propiedad intencional o negligencia o tome objetos o valores sin autorización, será reportado de manera inmediata ante la H. Comisión de Responsabilidades y Sanciones del CU Costa Sur.

## SEGURIDAD EN LA EJECUCIÓN DE LAS PRÁCTICAS.

Para el desarrollo de las prácticas se pueden presentar los siguientes peligros y su riesgo asociado y es importante que el estudiante los considere y tome las medidas de prevención pertinentes:

No	Peligro o fuente de energía	Riesgo asociado
1	Ingresar con bebidas al laboratorio	Electrochoque, daño a los equipos.

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## INDICE

PRÁCTICA 1. VARIABLES .....	9
PRÁCTICA 2. CONDICIONALES SIMPLES, DOBLES. ....	11
PRÁCTICA 3. CONDICIONALES MÚLTIPLES .....	13
PRÁCTICA 4. CONTADORES Y ACUMULADORES .....	15
PRÁCTICA 5. CICLOS (FOR, WHILE, DO-WHILE).....	17
PRÁCTICA 6. FUNCIONES Y METODOS .....	19
PRÁCTICA 7. ARREGLOS UNIDIMENSIONALES.....	21
PRÁCTICA 8. ARREGLO UNIDIMENSIONAL, ASIGNACIÓN TAMAÑO .....	24
PRÁCTICA 9. ARREGLOS COMO CONTADORES .....	28
PRÁCTICA 10. ARREGLOS BIDIMENSIONALES .....	30
PRÁCTICA 11. ARREGLOS BIDIMENSIONALES “RESERVACIONES” .....	33
PRÁCTICA 12. ARREGLOS BIDIMENSIONALES “CUADRO MAGICO” .....	38
PRÁCTICA 13. PROGRAMA CALSE PERSONA.....	43
PRÁCTICA 14. PROGRAMA: CLASE PACIENTE .....	45
PRÁCTICA 15. OPERACIONES ARITMETICAS .....	47
PRÁCTICA 16. PRODUCTOS .....	49
PRÁCTICA 17. FIGURAS GEOMETRICAS .....	51
PRÁCTICA 18. INTERBLOQUEO .....	53
PRÁCTICA 19. SIMULACIÓN DE MEMORIA .....	55
PRÁCTICA 20. GRUPOS, USUARIOS Y PERMISOS .....	57
PRÁCTICA 21. FICHEROS Y DIRECTORIOS .....	61

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

<b>PRÁCTICA 22. CONCEPTOS BASICOS SOBRE LINUX .....</b>	<b>66</b>
<b>PRÁCTICA 23. INSTALACIÓN DE SISTEMA OPERATIVO BASADO EN UNIX</b>	<b>69</b>
<b>PRÁCTICA 24. PROCESOS: FUNCIÓN FORK .....</b>	<b>71</b>
<b>PRÁCTICA 25. GESTIÓN BÁSICA DE HILOS POSIX EN C.....</b>	<b>73</b>
<b>PRÁCTICA 26. HILOS INDEPENDIENTES DE LA PLATAFORMA CON JAVA</b>	<b>75</b>
<b>PRÁCTICA 27. SIMULACIÓN DE MEMORIA .....</b>	<b>77</b>
<b>PRÁCTICA 28. CSS: CLASES, ID, SELECTORES .....</b>	<b>79</b>
<b>PRÁCTICA 29. BOOTSTRAP: LAYOUT Y COMPONENTES .....</b>	<b>81</b>
<b>PRÁCTICA 30. BOOTSTRAP: PLANTILLAS .....</b>	<b>83</b>
<b>PRÁCTICA 31. JavaScript: DOM DE LA PAGINA.....</b>	<b>86</b>
<b>PRÁCTICA 32. JavaScript: FUNCTIONS, OBJECTS, EVENTS.....</b>	<b>2</b>
<b>PRÁCTICA 33. JavaScript: FORMULARIOS .....</b>	<b>4</b>
<b>PRÁCTICA 34. DESARROLLO DE APLICACIÓN CLIENTE SERVIDOR CON ENVÍO DE OBJETOS .....</b>	<b>6</b>
<b>PRÁCTICA 35. DESARROLLO DE APLICACIÓN SERVIDOR TCP CONCURRENTE .....</b>	<b>8</b>
<b>PRÁCTICA 36. DESARROLLO DE APLICACIÓN JAVA-RMI .....</b>	<b>10</b>
<b>PRÁCTICA 37. PUBLICAR SERVICIO DESDE UNA APLICACIÓN WEB.....</b>	<b>12</b>
<b>PRÁCTICA 38. PUBLICAR SERVICIO DESDE UNA APLICACIÓN WEB.....</b>	<b>14</b>
<b>PRÁCTICA 39. APP CON DISEÑO BASADO EN RESTRICCIONES E ÍCONO</b> <b>PUBLICAR SERVICIO DESDE UNA APLICACIÓN WEB .....</b>	<b>16</b>
<b>PRÁCTICA 40. APP UTILIZANDO LINEAR LAYOUT .....</b>	<b>18</b>
<b>PRÁCTICA 41. USO DE INTENTS .....</b>	<b>20</b>
<b>PRÁCTICA 42. USO DE INTENTS ABRIR OTRA ACTIVIDAD Y ENVIAR DATOS .....</b>	<b>22</b>

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

<b>PRÁCTICA 43. DISEÑO PORTRAIT Y LANDSCAPE .....</b>	<b>24</b>
<b>PRÁCTICA 44. LISTA DE COMPONENTES PERSONALIZADOS .....</b>	<b>26</b>
<b>PRÁCTICA 45. ACELERÓMETRO .....</b>	<b>28</b>
<b>PRÁCTICA 46. BASE DE DATOS SQLite.....</b>	<b>30</b>
<b>PRÁCTICA 47. DESARROLLO DE APLICACIÓN PARA ESTABLECER LA COMUNICACIÓN BLUETOOTH .....</b>	<b>32</b>

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## PRÁCTICA 1. VARIABLES

<b>Carrera:</b>	
<b>Nombre de la materia:</b>	

Código	Nombre completo de los alumnos

<b>Fecha:</b>	
---------------	--

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## Objetivos.

**Objetivo General:** El alumno conocerá la forma de declaración de variables, asignación de valores a variables, operadores aritméticos, mostrar valores en etiquetas, concatenación de datos, generar eventos.

**Objetivo Específico:** EL estudiante será capaz al término de esta, de identificar las variables y los diferentes procesos de entrada y de salida que puede hacer con las mismas.

## Materiales y equipos.

Un equipo de cómputo con un IDE de programación que reconozca la sintaxis del lenguaje de programación Java de preferencia NetBeans versión 8.2.

**Trabajo previo.** Tener instalado el IDE NetBeans 8.2.

## Metodología o Desarrollo.

1. Elegir un problema específico del banco de prácticas perteneciente a este objetivo.
2. Realizar la creación del proyecto y la interfaz gráfica con los componentes necesarios para la resolución del problema.
3. Escribir y explicar el proceso de declaración, lectura y asignación de variables desde un componente específico.
4. Conocer los diferentes operadores aritméticos e identificar los que se deben de aplicar, respecto del problema que se esté resolviendo.
5. Mostrar los resultados de las diferentes operaciones en componentes de salida.

## Resultados y conclusiones.

El alumno debe analizar y comparar los resultados teóricos, simulados y experimentales para la declaración de variables, asignación de valores a variables, operadores aritméticos, mostrar valores en etiquetas, concatenación de datos, generar eventos. Además, el alumno realizará la verificación de la correcta ejecución del programa realizando las pruebas de escritorio y verificarlas con el programa en ejecución.

## Referencias.

## Anexos (Rúbrica de práctica en laboratorio y rúbrica de reporte de práctica).

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## **PRÁCTICA 2. CONDICIONALES SIMPLES, DOBLES.**

<b>Carrera:</b>	
<b>Nombre de la materia:</b>	

Código	Nombre completo de los alumnos

<b>Fecha:</b>	
---------------	--

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## Objetivos.

**Objetivo General:** El alumno conocerá la sintaxis de una condición simple y doble, operadores de relación.

**Objetivo Específico:** EL estudiante será capaz al término de esta, de conocer e identificar los problemas donde estipular las estructuras condicionales simple o dobles respecto del problema en cuestión aplicando la correcta redacción del enunciado condicional apoyándose en los operadores de relación, así como de su correcta redacción.

## Materiales y equipos.

Un equipo de cómputo con un IDE de programación que reconozca la sintaxis del lenguaje de programación Java de preferencia NetBeans versión 8.2.

**Trabajo previo.** Tener instalado el IDE NetBeans 8.2. y haber realizado y comprendido los conceptos referentes a las variables

## Metodología o Desarrollo.

1. Elegir un problema específico del banco de prácticas perteneciente a este objetivo.
2. Realizar la creación del proyecto y la interfaz gráfica con los componentes necesarios para la resolución del problema. Explicar la sintaxis y el uso de una estructura condicional simple y una estructura condicional doble.
3. Explicar la sintaxis y el uso de los operadores de relación en un enunciado condicional. Identificar las diferencias de la aplicación de estas estructuras y aplicarla respecto del problema en cuestión.
4. Escribir el enunciado condicional en la estructura que resuelva el problema en cuestión.
5. Mostrar los resultados de las diferentes operaciones en componentes de salida.

## Resultados y conclusiones.

El alumno debe analizar y comparar los resultados teóricos, simulados y experimentales para identificar los problemas donde estipular las estructuras condicionales simples o dobles respecto del problema en cuestión aplicando la correcta redacción del enunciado condicional apoyándose en los operadores de relación, así como de su correcta redacción. Además, el alumno realizará la verificación de la correcta ejecución del programa realizando las pruebas de escritorio y verificarlas con el programa en ejecución.

## Referencias.

**Anexos (Rúbrica de práctica en laboratorio y rúbrica de reporte de práctica).**

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## PRÁCTICA 3. CONDICIONALES MÚLTIPLES

<b>Carrera:</b>	
<b>Nombre de la materia:</b>	

Código	Nombre completo de los alumnos

<b>Fecha:</b>	
---------------	--

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## Objetivos.

**Objetivo General:** El alumno conocerá la sintaxis para una estructura condicional simple, estructura condicional múltiple, estructuras anidadas, lectura de datos char e identificar la tecla enter, Error cuando sean letras en lugar de números.

**Objetivo Específico:** EL estudiante será capaz al término de esta, de conocer e identificar los problemas donde estipular las estructuras condicionales simple o múltiples, leer y condicionar valores tipo cadena y tipo carácter, así como de utilizar la tecla Enter como desencadenador del evento.

## Materiales y equipos.

Un equipo de cómputo con un IDE de programación que reconozca la sintaxis del lenguaje de programación Java de preferencia NetBeans versión 8.2.

**Trabajo previo.** Tener instalado el IDE NetBeans 8.2. y haber realizado y comprendido los conceptos referentes a las variables. Escribir y razonar los enunciados condicionales.

## Metodología o Desarrollo.

1. Elegir un problema específico del banco de prácticas perteneciente a este objetivo
2. Realizar la creación del proyecto y la interfaz gráfica con los componentes necesarios para la resolución del problema.
3. Explicar la sintaxis y el uso de la estructura condicional múltiple.
4. Identificar la aplicación de esta estructura y aplicarla respecto del problema en cuestión.
5. Escribir el enunciado condicional en la estructura que resuelva el problema en cuestión identificando si el tipo de dato es un carácter o una cadena de caracteres.
6. Mostrar los resultados de las diferentes operaciones en componentes de salida.

## Resultados y conclusiones.

El alumno debe analizar y comparar los resultados teóricos, simulados y experimentales para identificar los problemas donde estipular las estructuras condicionales múltiples respecto del problema en cuestión, aplicando la correcta redacción del enunciado condicional, identificando si el tipo de dato es un carácter o una cadena. Además, el alumno realizará la verificación de la correcta ejecución del programa realizando las pruebas de escritorio y verificarlas con el programa en ejecución.

## Referencias.

## Anexos (Rúbrica de práctica en laboratorio y rúbrica de reporte de práctica).

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## **PRÁCTICA 4. CONTADORES Y ACUMULADORES**

<b>Carrera:</b>	
<b>Nombre de la materia:</b>	

Código	Nombre completo de los alumnos

<b>Fecha:</b>	
---------------	--

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## Objetivos.

**Objetivo General:** El alumno conocerá la sintaxis para escribir los contadores y acumuladores.

**Objetivo Específico:** EL estudiante será capaz al término de esta, de conocer e identificar los problemas donde estipular contadores y acumuladores, así como de conocer e implementar los operadores aritméticos incrementales y decrementales.

## Materiales y equipos.

Un equipo de cómputo con un IDE de programación que reconozca la sintaxis del lenguaje de programación Java de preferencia NetBeans versión 8.2.

**Trabajo previo.** Tener instalado el IDE NetBeans 8.2. y haber realizado y comprendido los conceptos referentes a las variables. Escribir y razonar las estructuras condicionales simples, dobles y múltiples.

## Metodología o Desarrollo.

1. Elegir un problema específico del banco de prácticas perteneciente a este objetivo.
2. Realizar la creación del proyecto y la interfaz gráfica con los componentes necesarios para la resolución del problema.
3. Conocer y aplicar la sintaxis y el uso de las variables tipo contador o acumulador.
4. Identificar la aplicación de los operadores aritméticos incrementales y decrementales.
5. Escribir la variable contador o acumulador en el bloque necesario para llevar el conteo incremental o decremental donde sea necesario respecto del programa que se esté solucionando.
6. Mostrar los resultados de las diferentes operaciones en componentes de salida.

## Resultados y conclusiones.

El alumno debe analizar y comparar los resultados teóricos, simulados y experimentales para identificar el problema, el espacio específico y el enunciado que realice el conteo de los valores en su forma incremental o decremental, además del valor que necesite su aumento o disminución en el problema. Además, el alumno realizará la verificación de la correcta ejecución del programa realizando las pruebas de escritorio y verificarlas con el programa en ejecución.

## Referencias.

## Anexos (Rúbrica de práctica en laboratorio y rúbrica de reporte de práctica).

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## **PRÁCTICA 5. CICLOS (FOR, WHILE, DO-WHILE)**

<b>Carrera:</b>	
<b>Nombre de la materia:</b>	

Código	Nombre completo de los alumnos

<b>Fecha:</b>	
---------------	--

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## Objetivos.

**Objetivo General:** El alumno conocerá la sintaxis para escribir los denominados ciclos o iteraciones y los implementará en la resolución de problemas

**Objetivo Específico:** EL estudiante será capaz al término de esta práctica, de conocer e identificar las estructuras iterativas for, while y do-while, así como identificar la diferencia entre las estructuras, la aplicación y resolución de problemas comunes con las mismas para con ello simplificar el código

## Materiales y equipos.

Un equipo de cómputo con un IDE de programación que reconozca la sintaxis del lenguaje de programación Java de preferencia NetBeans versión 8.2.

**Trabajo previo.** Tener instalado el IDE NetBeans 8.2. y haber realizado y comprendido los conceptos referentes a las variables. Escribir y razonar las estructuras condicionales simples, dobles y múltiples, contadores, acumuladores, operadores incrementales y decrementales.

## Metodología o Desarrollo.

1. Elegir un problema específico del banco de prácticas perteneciente a este objetivo
2. Realizar la creación del proyecto y la interfaz gráfica con los componentes necesarios para la resolución del problema.
3. Conocer la sintaxis de las estructuras iterativas for, while y do-while.
4. Identificar la diferencia en la aplicación de las estructuras iterativas for, while y do-while.
5. Reconocer la o las instrucciones que necesiten iteración respecto del problema en solución, seleccionar la estructura adecuada y aplicarla.
6. Mostrar los resultados de las diferentes operaciones en componentes de salida.

## Resultados y conclusiones.

El alumno debe analizar y comparar los resultados teóricos, simulados y experimentales para identificar la estructura iterativa que mejor se adapte a la resolución del problema e implementarla con las debidas instrucciones. Además, el alumno realizará la verificación de la correcta ejecución del programa realizando las pruebas de escritorio y verificarlas con el programa en ejecución.

## Referencias.

## Anexos (Rúbrica de práctica en laboratorio y rúbrica de reporte de práctica).

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## PRÁCTICA 6. FUNCIONES Y METODOS

<b>Carrera:</b>	
<b>Nombre de la materia:</b>	

Código	Nombre completo de los alumnos

<b>Fecha:</b>	
---------------	--

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## Objetivos.

**Objetivo General:** El alumno conocerá la sintaxis para escribir los las funciones y métodos e identificará la diferencia en la aplicación de ambos.

**Objetivo Específico:** EL estudiante será capaz al término de esta práctica, de conocer e identificar los métodos o funciones en sus diferentes formatos, vacías o de retorno, la forma en cómo se implementan e identificará la diferencia entre ellos. Además, determinará cuál es el indicado para la resolución del problema en cuestión.

## Materiales y equipos.

Un equipo de cómputo con un IDE de programación que reconozca la sintaxis del lenguaje de programación Java de preferencia NetBeans versión 8.2.

**Trabajo previo.** Tener instalado el IDE NetBeans 8.2. y haber realizado y comprendido los conceptos referentes a las variables. Escribir y razonar las estructuras condicionales simples, dobles y múltiples, contadores, acumuladores, operadores incrementales y decrementales, estructuras iterativas.

## Metodología o Desarrollo.

1. Elegir un problema específico del banco de prácticas perteneciente a este objetivo
2. Realizar la creación del proyecto y la interfaz gráfica con los componentes necesarios para la resolución del problema.
3. Conocer la sintaxis de las funciones o métodos y la diferencia entre ambos.
4. Conocer la diferencia entre las funciones de retorno y vacías y los métodos.
5. Crear la función o el método que soluciones el problema y encierre las instrucciones necesarias.
6. Llamar la función en los diferentes apartados que sean necesarios.
7. Mostrar los resultados de las diferentes operaciones en componentes de salida.

## Resultados y conclusiones.

El alumno debe analizar y comparar los resultados teóricos, simulados y experimentales para implementar métodos o funciones que realicen la actividad necesaria para la resolución del problema, así como el correcto llamado de la misma. Además, el alumno realizará la verificación de la correcta ejecución del programa realizando las pruebas de escritorio y verificarlas con el programa en ejecución.

## Referencias.

**Anexos (Rúbrica de práctica en laboratorio y rúbrica de reporte de práctica).**

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## **PRÁCTICA 7. ARREGLOS UNIDIMENSIONALES**

<b>Carrera:</b>	
<b>Nombre de la materia:</b>	

Código	Nombre completo de los alumnos

<b>Fecha:</b>	
---------------	--

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## Objetivos.

### Objetivo General.

El alumno comprenderá la importancia de las estructuras de datos utilizando los objetos arreglo. En los cuales podrá agregar información relacionada de un mismo tipo.

### Objetivo Específico.

Identificara las distintas maneras de declarar un arreglo

Identificará la forma de acceder a los índices a través de una estructura de control

### Materiales y Equipo.

- Equipo de Cómputo
- IDE para desarrollo

### Metodología o Desarrollo.

Las estructuras de datos son colecciones de elementos de datos relacionados. Los objetos arreglo son estructuras de datos que consisten en elementos de datos relacionados, del mismo tipo. Los arreglos facilitan el procesamiento de grupos de valores relacionados. Los arreglos conservan la misma longitud una vez creados.

Para la siguiente práctica, se creará el siguiente programa declarando arreglos de distintos tipos de datos, así como 3 formas distintas de declarar un arreglo y acceder a la información almacenada dentro de este.

```

10  * @author Gilbert
11  */
12  public class ArreglosIMEC {
13
14  /**
15   * @param args the command line arguments
16  */
17  public static void main(String[] args) {
18    // TODO code application logic here
19    String[] datos = new String[5]; //Se declaró un arreglo de 5 posiciones de memoria
20    String[] estados = {"Jalisco", "México", "Oaxaca", "Estado de México", "Colima"}; //Se inicializa un arreglo con datos almacenados
21    int[] dias = new int[6];// Se declara un arreglo entero de 6 posiciones
22    //
23    dias[0] = 10;
24    dias[1] = 4; // Llenamos el arreglo accediendo a una posición de memoria a la vez
25    dias[2] = 12;
26    dias[3] = 1;
27    dias[4] = 65;
28    dias[5] = 10;
29
30    for (int indice = 0; indice < 5; indice++) { // A través de un ciclo for podemos imprimir cada posición de memoria
31      System.out.println(estados[indice]); //Impresión del arreglo estados
32    }
33
34    System.out.printf("%s%8s%n", "Indice", "Valor"); //Utilizamos el Format para darle formato a la impresión
35    for (int indice = 1; indice < dias.length; indice++) {
36      System.out.printf("%5d%8d%n", indice, dias[indice]);
37    }
38  }
39
40}
41

```

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## Resultado y Conclusiones

Al finalizar la práctica, el alumno podrá relacionar lo visto teóricamente con respecto a las estructuras de datos y la aplicación de arreglos.

Se realizará la modificación de los datos ingresados. Así como la declaración de un nuevo arreglo de 20 posiciones al cual a través de un ciclo for se debe llenar del 1 al 20, en imprimirllo.

## Referencias

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## **PRÁCTICA 8. ARREGLO UNIDIMENSIONAL, ASIGNACIÓN TAMAÑO**

<b>Carrera:</b>	
<b>Nombre de la materia:</b>	

Código	Nombre completo de los alumnos

<b>Fecha:</b>	
---------------	--

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## Objetivos.

### Objetivo General

Relacionar los conocimientos teóricos de la clase con la práctica.

El alumno conocerá la declaración de arreglos a través de la interfaz del usuario.

### Objetivo Específico

Identificara las formas de llenar información dentro de un arreglo.

Podrá indicarle al usuario que controle el tamaño del arreglo para los datos que asignará dentro del arreglo.

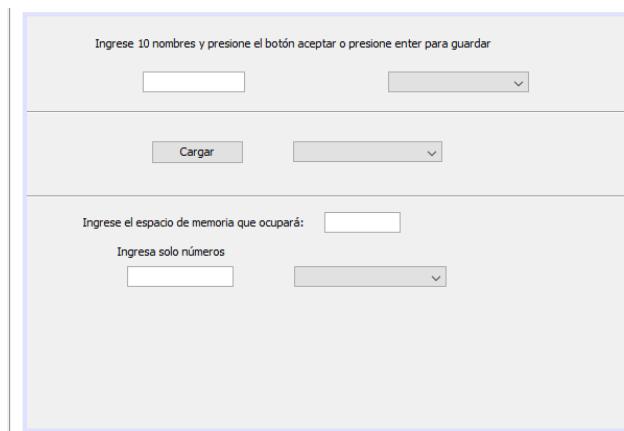
Identificará el uso de algunas estructuras de control y eventos.

### Materiales y Equipo

- Equipo de cómputo
- IDE de desarrollo

### Metodología o Desarrollo

Crea la siguiente interfaz gráfica con un JFRAME llamado interfaz



A los objetos que tenemos en el JFRAME indicará los siguientes nombres de variables:

1. Al primer JTextField le indicará el nombre de variable txtNombres
2. Al primer combo box le indicará el nombre de cboNombres

En la segunda sección

3. Al JButton le asignará el nombre de btnCargar
4. Al JComboBox le asignará el nombre de cboEstados

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

En la tercera sección

5. Al JTextField le asignará el nombre de txtTamArreglo
6. Al JTextField le asignará el nombre de txtNumeros
7. Al JComboBox le asignará el nombre de cboNumeros

Al inicio de la clase agregará el siguiente código, así como en el constructor de la clase

```

9  /*
10  * @author Gilbert
11  */
12 public class interfaz extends javax.swing.JFrame {
13
14     String nombres[] = new String[10];//Declaramos un arreglo de 10 posiciones
15     String estados[] = {"Baja California", "Aguascalientes", "Jalisco", "Oaxaca", "Durango");//Inicializamos
16     int contador = 0;//Variable de tipo entero
17     int contador2 = 0;
18     int numerosDecimales[];//Declaramos arreglo de tipo entero, pero sin indicar su tamaño
19
20     public interfaz() {
21         initComponents();
22         setLocationRelativeTo(this); //Indicamos que nuestra ventana aparezca centrada
23     }

```

Creará los siguientes eventos de tipo actionPerformed en los siguientes objetos:

- txtNombre

```

144
145     private void txtNombreActionPerformed(java.awt.event.ActionEvent evt) {
146         if (contador <= 9) {
147             nombres[contador] = txtNombre.getText();
148             cboNombres.addItem(nombres[contador]);
149             txtNombre.setText("");
150             contador++;
151         }
152

```

- btnCargar

```

154
155     private void btnCargarActionPerformed(java.awt.event.ActionEvent evt) {
156         for (int indice = 0; indice < estados.length; indice++) {
157             cboEstados.addItem(estados[indice]);
158         }
159

```

- txtTamArreglo

```

162
163     private void txtTamArregloActionPerformed(java.awt.event.ActionEvent evt) {
164         numerosDecimales = new int[Integer.parseInt(txtTamArreglo.getText())];
165         txtTamArreglo.setEnabled(false);
166         System.out.println(numerosDecimales.length);

```

- TxtNumeros

```

168
169     private void txtNumerosActionPerformed(java.awt.event.ActionEvent evt) {
170         if (contador2 < numerosDecimales.length) {
171             numerosDecimales[contador2] = Integer.parseInt(txtNumeros.getText());
172             cboNumeros.addItem(String.valueOf(numerosDecimales[contador2]));
173             contador2++;

```

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## Resultado y Conclusiones

El alumno relacionará lo visto en clase con los resultados obtenido de la práctica que acaba de realizar.

Así mismo, realizará las siguientes modificaciones,

1. Asignar nuevos arreglos de distintos tipos de datos
2. Prevenir que el usuario solo ingrese valores enteros al asignar el tamaño

## Referencias

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## PRÁCTICA 9. ARREGLOS COMO CONTADORES

<b>Carrera:</b>	
<b>Nombre de la materia:</b>	

<b>Código</b>	<b>Nombre completo de los alumnos</b>

<b>Fecha:</b>	
---------------	--

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## Objetivos.

### Objetivo General

El alumno podrá hacer uso de los arreglos unidimensionales para el conteo de registro de datos, y no solo como una forma de almacenar valores.

### Objetivo Específico

Identificara el uso de la librería SecureRandom

Identificará el uso de un arreglo unidimensional como contador.

Materiales y Equipo

### Materiales y Equipo

-IDE de desarrollo

### Metodología o Desarrollo

Para la práctica creará el siguiente programa para ejecutarse en consola

```

3
4   import java.security.SecureRandom;
5
6   /**
7    * @author Gilbert
8    */
9   public class ArreglosIMEC3 {
10
11
12   public static void main(String[] args) {
13       SecureRandom aRandom = new SecureRandom(); // Creamos una variable para generar números aleatorios
14       int[] repeticiones = new int[10]; //Arreglo de tipo entero de 10 posiciones
15       int numAleatorio; //Variable de tipo entero sin inicializar
16       for (int i = 0; i < 100; i++) { //Ciclo for que dará 100 vueltas
17           numAleatorio = aRandom.nextInt(10); //Indicamos que se generen números aleatorios del 0 al 9
18           ++repeticiones[numAleatorio]; //El número aleatorio generado lo asignamos de posición de memoria
19           //los signo ++ nos permiten sumar en 1 dependiendo la posición de memoria
20       }
21       System.out.printf("%s%10s%n", "Indice", "Repeticiones"); //Damos formato para imprimir el resultado
22       for (int indice = 0; indice < repeticiones.length; indice++) { //Ciclo for para imprimir los valores
23           System.out.printf("%d%10d%n", indice, repeticiones[indice]);
24       }
25   }
26
27 }
```

## Resultado y Conclusiones

Una vez terminado, el alumno pondrá en uso lo relacionado con la práctica. Para aplicar el código realizado en otros programas, que pueden servir para encuestas, etc.

## Referencias

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## **PRÁCTICA 10. ARREGLOS BIDIMENSIONALES**

Carrera:	
Nombre de la materia:	

Código	Nombre completo de los alumnos

Fecha:	
--------	--

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## Objetivos.

### Objetivo General

El alumno conocerá las formas en que puede declarar un arreglo bidimensional, y como acceder a cada una de las posiciones de memoria a través de un iterador anidado en este caso en particular, la estructura de control for.

### Objetivo Específico

Declarar arreglos bidimensionales

Iterar a través de los arreglos mediante el ciclo for anidado

Inicializar, y hacer referencia a los elementos.

### Materiales y Equipo

-Equipo de cómputo

-IDE de desarrollo

### Metodología o Desarrollo

Los arreglos multidimensionales de dos dimensiones se utilizan con frecuencia para representar tablas de valores, con datos ordenados en filas y columnas. Para identificar un elemento específico de una tabla, debemos especificar dos índices.

Por convención, el primero identifica la fila y el segundo la columna. Los arreglos que requieren dos índices para identificar un elemento específico se llaman arreglos bidimensionales.

Para poner en práctica lo anterior, crearemos la siguiente interfaz gráfica

```

10  * @author Gilbert
11  */
12  public class ArreglosIMEC6 {
13
14  public static void main(String[] args) {
15      int arreglo[][] = new int[10][10];//Declaramos un arreglo de 10 x 10 de tipo entero
16
17      for (int fila = 0; fila < arreglo.length; fila++) {//Creamos un ciclo for anidado para acceder a cada fila y columna del arreglo
18          //E indicamos el método length para indicar el límite de este ciclo por la fila
19          for (int columna = 0; columna < arreglo[fila].length; columna++) {//ciclo for para acceder a cada columna del arreglo bidimensional
20              //utilizamos el método length después de los corchetes de la fila para indicar el tamaño de la columnas
21              arreglo[fila][columna] = (fila + 1) * (columna + 1);//guardamos los valores resultado de la operación por fila y columna
22              System.out.printf("%10d", arreglo[fila][columna]);//imprimimos los valores con formato
23
24          }
25          System.out.println();
26      }
27  }
28
29
30 }

```

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## Resultado y Conclusiones

Una vez terminada la práctica, el alumno relacionará lo visto en clase presencial teórica, con lo implementado dentro de la práctica. Podrá sacar conclusiones entre el uso de un arreglo de una sola dimensión y de dos dimensiones. Modificando los valores de entrada.

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## **PRÁCTICA 11. ARREGLOS BIDIMENSIONALES “RESERVACIONES”**

<b>Carrera:</b>	
<b>Nombre de la materia:</b>	

Código	Nombre completo de los alumnos

<b>Fecha:</b>	
---------------	--

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## Objetivos.

### Objetivo General

El alumno pondrá en práctica sus conocimientos, implementando el arreglo bidimensional para almacenar datos en un registro de reservaciones de camiones. En la forma de como guardar y accede a esta misma información, así como aplicar conocimiento aprendidos anteriormente, como métodos, arreglos unidimensionales, paso de referencia.

### Objetivo Específico

Utilizar arreglos bidimensionales para almacenar datos

Obtendrá listas y tabla de valores

Aplicar iterador para el desplazamiento dentro del arreglo

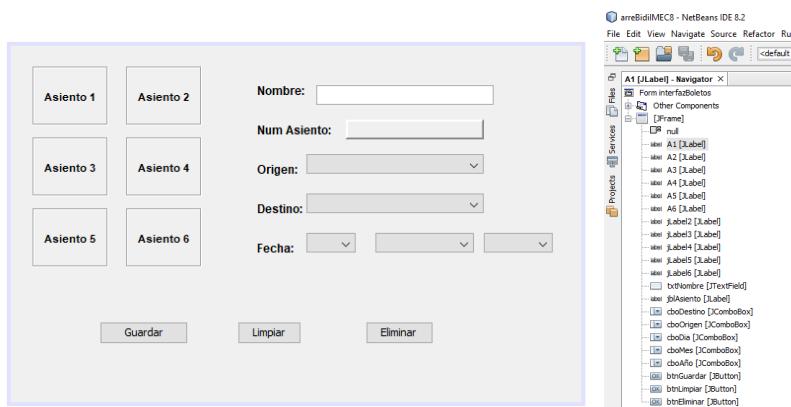
### Materiales y Equipo

-Equipo de cómputo.

-IDE de desarrollo.

### Metodología o Desarrollo

Para poner en práctica el almacenamiento de información a través de los arreglos bidimensionales, creamos la siguiente interfaz, declarando las siguientes variables a los objetos.



	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

En el código empezaremos con lo siguiente:

```

2 import java.awt.Color;
3 import javax.swing.JComboBox;
4 import javax.swing.JLabel;
5
6 /**
7  * 
8  * @author Gilbert
9 */
10 public class interfazBoletos extends javax.swing.JFrame {
11
12     String boletos[][] = new String[6][7];//Declaramos un arreglo bidimensional de 6 x 7
13     JLabel etiquetas[] = new JLabel[6];//Creamos un arreglo unidimensional de tipo JLabel de 6 posiciones
14     String origenDestino[] = {"Selección", "Guadalajara", "Autlán", "Casimiro Castillo", "La Huerta", "El Grullo", "Melaque", "Cd. Guzman"};
15     String mes[] = {"Enero", "Febrero", "Marzo", "Abril", "Mayo", "Junio", "Julio", "Agosto", "Septiembre", "Octubre", "Noviembre", "Diciembre"};
16     JComboBox combo[] = new JComboBox[5];//Creamos un arreglo de tipo JComboBox de 5 posiciones
17     Color colorEtiquetas[] = {Color.LIGHT_GRAY, Color.DARK_GRAY, Color.RED};//Creamos un arreglo de tipo COLOR e inicializamos los siguientes valores
18     int asientoSeleccionado = 0;

```

Creamos el método cargar, para guardar las variables creados de tipo JComboBox y JLabel a un arreglo.

```

29 public void cargar() { // Creamos el siguiente método para inicializar los arreglos de tipo JComboBox y JLabel
30     combo[0] = cboOrigen;
31     combo[1] = cboDestino;
32     combo[2] = cboDia;
33     combo[3] = cboMes;
34     combo[4] = cboAño;
35
36     etiquetas[0] = A1;
37     etiquetas[1] = A2;
38     etiquetas[2] = A3;
39     etiquetas[3] = A4;
40     etiquetas[4] = A5;
41     etiquetas[5] = A6;
42
43     for (int i = 0; i < origenDestino.length; i++) {//Cargar orígenes y destinos a los combo box
44         cboOrigen.addItem(origenDestino[i]);
45         cboDestino.addItem(origenDestino[i]);
46     }
47
48     for (int i = 0; i < mes.length; i++) {
49         cboMes.addItem(mes[i]);
50     }
51
52     for (int i = 0; i < 31; i++) {
53         cboDia.addItem(" " + (i + 1));
54     }
55     cboAño.addItem("2020");
56     cboAño.addItem("2021");
57
58     for (int i = 0; i < etiquetas.length; i++) {
59         etiquetas[i].setBackground(colorEtiquetas[0]);
60         etiquetas[i].setOpaque(true);
61     }
62
63 }

```

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

Creamos los siguientes métodos:

```

64
65  public void onOff(boolean actDes) { //Creamos un método con un paso de parametros de tipo booleano para activar y desactivar el formulario
66    txtNombre.setEnabled(actDes);
67    for (int i = 0; i < combo.length; i++) {
68      combo[i].setEnabled(actDes);
69    }
70    btnGuardar.setEnabled(actDes);
71    btnEliminar.setEnabled(actDes);
72    btnLimpiar.setEnabled(actDes);
73  }
74
75
76  public void limpiar() { //Creamos un método que nos permitirá limpiar los JTextField y posicionar en 0 los combo box
77    txtNombre.setText("");
78    jblAsiento.setText("");
79    for (int i = 0; i < combo.length; i++) {
80      combo[i].setSelectedIndex(0);
81    }
82  }
83
84  public void mostrar(int seleccion) { //Creamos un método que nos mostrará los datos guardados según la mesa seleccionada
85    txtNombre.setText(boletos[seleccion][0]);
86    jblAsiento.setText(boletos[seleccion][1]);
87
88    cboOrigen.setSelectedItem(boletos[seleccion][2]);
89    cboDestino.setSelectedItem(boletos[seleccion][3]);
90    cboDia.setSelectedItem(boletos[seleccion][4]);
91    cboMes.setSelectedItem(boletos[seleccion][5]);
92    cboAño.setSelectedItem(boletos[seleccion][6]);
93  }
94

```

A cada etiqueta que representa la mesa, le crearemos un evento de tipo MouseClicked y agregaremos el siguiente código, tomando en cuenta que, por cada mesa, en la variable asientoSeleccionado se indica el valor de 0, al siguiente el valor de 1 y así consecutivamente.

```

274  private void A1MouseClicked(java.awt.event.MouseEvent evt) {
275    asientoSeleccionado = 0;
276    onOff(true);
277
278    if (boletos[asientoSeleccionado][0] == null) {
279      etiquetas[asientoSeleccionado].setBackground(colorEtiquetas[1]);
280      jblAsiento.setText(etiquetas[asientoSeleccionado].getText());
281    } else {
282      mostrar(asientoSeleccionado);
283    }
284  }
285
286
287  private void A2MouseClicked(java.awt.event.MouseEvent evt) {
288    asientoSeleccionado = 1;
289    onOff(true);
290
291    if (boletos[asientoSeleccionado][0] == null) {
292      etiquetas[asientoSeleccionado].setBackground(colorEtiquetas[1]);
293      jblAsiento.setText(etiquetas[asientoSeleccionado].getText());
294    } else {
295      mostrar(asientoSeleccionado);
296    }
297  }
298
299  private void A3MouseClicked(java.awt.event.MouseEvent evt) {
300    asientoSeleccionado = 2;
301    onOff(true);
302
303    if (boletos[asientoSeleccionado][0] == null) {
304      etiquetas[asientoSeleccionado].setBackground(colorEtiquetas[1]);
305      jblAsiento.setText(etiquetas[asientoSeleccionado].getText());
306    } else {
307      mostrar(asientoSeleccionado);
308    }
309  }

```

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

Al botón btnGuardar le creamos un evento de tipo ActionPerformed y agregamos el siguiente código.

```

346
347  □ private void btnGuardarActionPerformed(java.awt.event.ActionEvent evt) {
348      boletos[asientoSeleccionado][0] = txtNombre.getText();
349      boletos[asientoSeleccionado][1] = jblAsiento.getText();
350      boletos[asientoSeleccionado][2] = (String) cboOrigen.getSelectedItem();
351      boletos[asientoSeleccionado][3] = (String) cboDestino.getSelectedItem();
352      boletos[asientoSeleccionado][4] = (String) cboDia.getSelectedItem();
353      boletos[asientoSeleccionado][5] = (String) cboMes.getSelectedItem();
354      boletos[asientoSeleccionado][6] = (String) cboAño.getSelectedItem();
355
356      etiquetas[asientoSeleccionado].setBackground(colorEtiquetas[2]);
357      onOff(false);
358      limpiar();
359  }

```

Inicializamos los métodos siguientes en el constructor de la clase.

```

public interfazBoletos() {
    initComponents();
    setSize(700, 400);
    setLocationRelativeTo(this);
    cargar();
    onOff(false);

}

```

## Resultado y Conclusiones

El alumno en esta práctica hará las siguientes modificaciones.

1. Debe verificar que el formulario este completamente lleno para poder guardar. (Evitar guardar valores vacíos)
2. Al ir seleccionando cualquier asiento este debe cambiar de color para indicar que el asiento ha sido seleccionado, si selecciono otro, el anteriormente seleccionado debe volver a su color normal.
3. Configurar los botones de limpiar y eliminar.
4. Al momento de eliminar un registro debe mostrar un JOptionPane en el cual pregunte si estás seguro que quieres eliminar, si es así el asiento debe volver a su color cuando no ha estado seleccionado y el formulario debe bloquearse de nueva cuenta.
5. Comentar el código.

## Referencias

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## **PRÁCTICA 12. ARREGLOS BIDIMENSIONALES “CUADRO MAGICO”**

<b>Carrera:</b>	
<b>Nombre de la materia:</b>	

Código	Nombre completo de los alumnos

<b>Fecha:</b>	
---------------	--

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión Junio 2020

## Objetivos.

## **Objetivo General**

El alumno comprenderá de una mejor manera la manipulación de los arreglos bidimensionales, para realizar juegos, crear métodos, y combinar el uso de arreglos unidimensionales con arreglos bidimensionales

## **Objetivo Especifico**

Identificará como hacer manipulaciones comunes de arreglos bidimensionales y unidimensionales

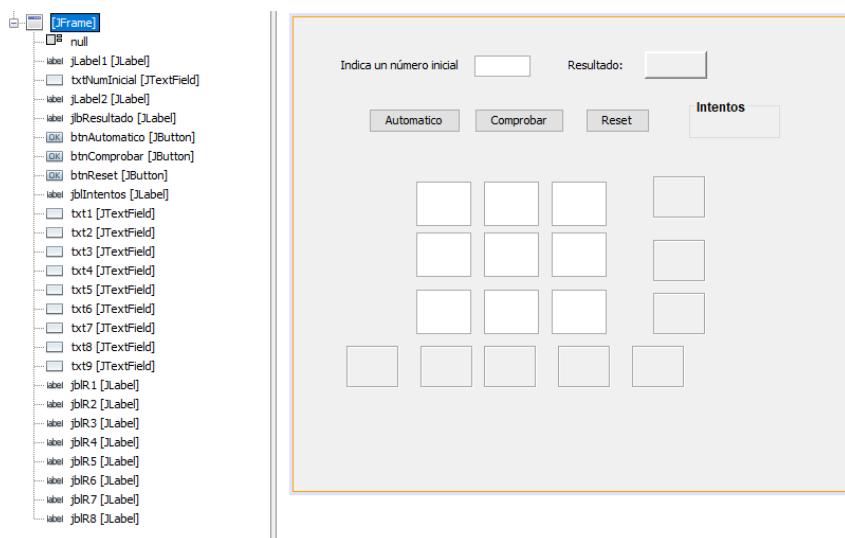
Identificará la forma de realizar sumas de elementos almacenados en los arreglos bidimensionales

## Materiales y Equipo

- Equipo de cómputo
  - IDE de desarrollo

## Metodología o Desarrollo

Continuando con los ejercicios en donde aplicamos el uso de arreglos bidimensionales, y arreglos unidimensionales de tipo objetos, creamos la siguiente interfaz y declaramos los siguientes nombres de variables a los objetos.



	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

Importamos y declaramos lo siguiente:

```

2   import javax.swing.JLabel;
3   import javax.swing.JTextField;
4
5   /**
6   * @author Gilbert
7   */
8   public class cuadroMagico extends javax.swing.JFrame {
9
10
11   int serieNum[] = new int[9];// Declaramos un arreglo de tipo entero
12   int cuadroMagico[][] = new int[3][3]; //Declaramos un arreglo bidimensional
13   JLabel etiquetas[] = new JLabel[8];//Arreglo de tipo JLabel
14   JTextField cajasTexto[] = new JTextField[9];//Arreglo de tipo JTextField

```

Creamos el método cargar, para inicializar los arreglos

```

23
24   public void cargar() { //Cargamos los arreglos con los elementos de tipo JLabel y JTextField
25     etiquetas[0] = jblR1;
26     etiquetas[1] = jblR2;
27     etiquetas[2] = jblR3;
28     etiquetas[3] = jblR4;
29     etiquetas[4] = jblR5;
30     etiquetas[5] = jblR6;
31     etiquetas[6] = jblR7;
32     etiquetas[7] = jblR8;
33
34     cajasTexto[0] = txt1;
35     cajasTexto[1] = txt2;
36     cajasTexto[2] = txt3;
37     cajasTexto[3] = txt4;
38     cajasTexto[4] = txt5;
39     cajasTexto[5] = txt6;
40     cajasTexto[6] = txt7;
41     cajasTexto[7] = txt8;
42     cajasTexto[8] = txt9;
43
44 }

```

Creamos los siguientes métodos que estaremos utilizando:

```

46
47   public void onOff(boolean actDes) { // Método para habilitar o deshabilitar
48     for (int indice = 0; indice < cajasTexto.length; indice++) {
49       cajasTexto[indice].setEnabled(actDes);
50     }
51
52   public int calcular(int numeroInicial) { //Método de tipo entero calcular los números dependiendo el valor inicial
53     int resultado = 0;
54     serieNum[0] = numeroInicial;
55     for (int i = 1; i < serieNum.length; i++) { //agregar los 9 números al arreglo serieNum
56       serieNum[i] = serieNum[i - 1] + 1;
57       System.out.println(" " + serieNum[i]);
58     }
59
60     for (int i = 0; i < serieNum.length; i++) { //suma los números del arreglo serieNum
61       resultado = resultado + serieNum[i];
62     }
63     resultado = resultado / 3;
64     cargarArrBidimensional();
65
66   }

```

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

Creamos el método cargarArrBidimensional para asignar la posición de los números según la matriz

```

68
69  public void cargarArrBidimensional() {//Método para llenar los textField con el resultado correcto
70      cuadroMagico[0][0] = serieNum[1];
71      cuadroMagico[0][1] = serieNum[8];
72      cuadroMagico[0][2] = serieNum[3];
73      cuadroMagico[1][0] = serieNum[6];
74      cuadroMagico[1][1] = serieNum[4];
75      cuadroMagico[1][2] = serieNum[2];
76      cuadroMagico[2][0] = serieNum[5];
77      cuadroMagico[2][1] = serieNum[0];
78      cuadroMagico[2][2] = serieNum[7];
79
80  }

```

Creamos el método mostrarResultado, el cual nos servirá para llenar de forma automática nuestro JTextField y mostrar el resultado del cuadro mágico

```

82
83  public void mostrarResultado() {//Método para cargar el resultado a los textField
84      int posicion = 0;
85      int temp;
86      for (int fila = 0; fila < cuadroMagico.length; fila++) {
87          temp = 0;
88          for (int columna = 0; columna < cuadroMagico[0].length; columna++) {
89              cajasTexto[posicion].setText(" " + cuadroMagico[fila][columna]);
90              temp = temp + cuadroMagico[fila][columna];
91              etiquetas[fila].setText(" " + temp);
92              posicion++;
93              System.out.println("posición " + posicion);
94          }
95          for (int columna = 0; columna < cuadroMagico[0].length; columna++) {
96              temp = 0;
97              for (int fila = 0; fila < cuadroMagico.length; fila++) {
98                  temp = temp + cuadroMagico[fila][columna];
99                  etiquetas[columna + 3].setText(" " + temp);
100             }
101            temp = 0;
102            for (int i = 0; i < cuadroMagico.length; i++) {
103                temp = temp + cuadroMagico[i][i];
104            }
105            etiquetas[6].setText(" " + temp);
106            for (int i = 2; i > cuadroMagico.length; i--) {
107                temp = temp + cuadroMagico[i][i];
108            }
109            etiquetas[7].setText(" " + temp);
110        }
111    }

```

Creamos un evento de tipo ActionPerfomert en el JTextField txtNumInicial, el cual nos permitirá indicar el número con el cual queremos iniciar nuestro cuadro mágico.

```

289
290  private void txtNumInicialActionPerformed(java.awt.event.ActionEvent evt) {
291      int res = calcular(Integer.parseInt(txtNumInicial.getText()));
292      jlbResultado.setText(" " + res);
293      onOff(true);
294      txtNumInicial.setEnabled(false);
295
296  private void btnAutomaticoActionPerformed(java.awt.event.ActionEvent evt) {
297      mostrarResultado();
298  }

```

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

Inicializamos los métodos en el constructor de la clase.

```

15
16  public cuadroMagico() {
17    initComponents();
18    setSize(500, 500);
19    setLocationRelativeTo(this);
20    cargar();
21    onOff(false);
22
  }

```

### Resultado y Conclusiones

El alumno deberá realizar las siguientes modificaciones

1. Corregir la suma de la diagonal de izquierda a derecha, mencionada en el video.
2. Cuando el programa inicie los botones deben estar bloqueados y se deben activar cuando el usuario indique el número con el cual quiere iniciar.
3. El usuario debe llenar los JTextField del cuadro mágico y dar clic en el botón comprobar los números se deben guardar en un arreglo bidimensional en donde se realizará la suman (similar a lo que se hizo en el botón automático) para verificar si su resultado es correcto, en los JLabel se debe mostrar el total sumado-
4. Si el resultado es incorrecto, en el JLabel de intentos se debe incrementar en 1, hasta un límite de 3 intentos.
5. Si el resultado es correcto, debe aparecer un mensaje que diga que lo ha logrado.
6. Al terminar los 3 intentos ya no podrá seguir jugando, y se activa el botón automatico para que el usuario pueda ver la respuesta correcta.
7. El botón reset limpia todos los TextField, bloquea todo y desbloquea el JTextField para indicar un nuevo número y comenzar a jugar de nuevo.

### Referencias

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## PRÁCTICA 13. PROGRAMA CALSE PERSONA

<b>Carrera:</b>	
<b>Nombre de la materia:</b>	

<b>Código</b>	<b>Nombre completo de los alumnos</b>

<b>Fecha:</b>	
---------------	--

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## Objetivos

Relacionar los conocimientos teóricos de clase con la práctica  
 Desarrollar las habilidades necesarias para comprender como funcionan los objetos, e instancias múltiples.

## Conceptos

- Clases
- Objetos
- Instancias

## Materiales y Equipo.

- 1 PC
- Entorno de desarrollo integrado compatible, que reconozca la sintaxis del lenguaje de programación Java.
- Máquina Virtual de Java versión 6 o superior (JRE) y Java Development Kit (JDK)

## Metodología o Desarrollo.

1. Desarrollar una aplicación en el lenguaje de programación Java y crear una Clase nueva llamada Persona, con los atributos: Nombres, Paterno, Materno, Fecha de Nacimiento.
2. Crear métodos en la clase persona para acceder a los atributos.
3. Crear múltiples instancias de la clase persona.

## Resultados y conclusiones.

Relacionar los conocimientos obtenidos en clase con la práctica.  
 Conocer el funcionamiento de las clases y los objetos, así como de las instancias múltiples.

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## **PRÁCTICA 14. PROGRAMA: CLASE PACIENTE**

Carrera:	
Nombre de la materia:	

Código	Nombre completo de los alumnos

Fecha:	
--------	--

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## Objetivos

Relacionar los conocimientos teóricos de clase con la práctica  
 Desarrollar las habilidades necesarias para comprender como funcionan los métodos.

## Conceptos

- Métodos
- Palabra clave this

## Materiales y Equipo

- 1 PC
- Entorno de desarrollo integrado compatible, que reconozca la sintaxis del lenguaje de programación Java.
- Máquina Virtual de Java versión 6 o superior (JRE) y Java Development Kit (JDK)

## Metodología o Desarrollo

Desarrollar una aplicación en el lenguaje de programación Java creando una clase personalizada llamada Paciente declarar los atributos y métodos necesarios para calcular el índice de masa corporal de múltiples pacientes.

## Resultados y conclusiones

Relacionar los conocimientos obtenidos en clase con la práctica.  
 Reafirmar el conocimiento de métodos.

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## PRÁCTICA 15. OPERACIONES ARITMÉTICAS

<b>Carrera:</b>	
<b>Nombre de la materia:</b>	

Código	Nombre completo de los alumnos

<b>Fecha:</b>	
---------------	--

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## Objetivos

Relacionar los conocimientos teóricos de clase con la práctica.

Desarrollar las habilidades necesarias para comprender como funcionan la sobrecarga de métodos.

## Conceptos

- Métodos
- Ámbito de variables y métodos
- Sobrecarga de métodos

## Materiales y Equipo

- 1 PC
- Entorno de desarrollo integrado compatible, que reconozca la sintaxis del lenguaje de programación Java.
- Máquina Virtual de Java versión 6 o superior (JRE) y Java Development Kit (JDK)

## Metodología o Desarrollo

Desarrollar una aplicación en el lenguaje de programación Java creando una clase personalizada llamada Matemáticas con los atributos y métodos necesarios para realizar las operaciones aritméticas básicas.

## Resultados y conclusiones

Relacionar los conocimientos obtenidos en clase con la práctica.

Aprender a utilizar la sobrecarga de métodos en el lenguaje Java.

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## PRÁCTICA 16. PRODUCTOS

<b>Carrera:</b>	
<b>Nombre de la materia:</b>	

Código	Nombre completo de los alumnos

<b>Fecha:</b>	
---------------	--

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## Objetivos

Relacionar los conocimientos teóricos de clase con la práctica  
 Desarrollar las habilidades necesarias para comprender como funciona los constructores.

## Conceptos

-Constructores

## Materiales y Equipo

- 1 PC
- Entorno de desarrollo integrado compatible, que reconozca la sintaxis del lenguaje de programación Java.
- Máquina Virtual de Java versión 6 o superior (JRE) y Java Development Kit (JDK)

## Metodología o Desarrollo

- Desarrollar una aplicación en el lenguaje de programación Java, crear una clase llamada Producto con sus atributos y métodos y múltiples constructores.
- Crear múltiples instancias de la clase Producto.

## Resultados y conclusiones.

Relacionar los conocimientos obtenidos en clase con la práctica.  
 Conocer cómo funcionan los constructores y para qué sirven.

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## PRÁCTICA 17. FIGURAS GEOMETRICAS

<b>Carrera:</b>	
<b>Nombre de la materia:</b>	

Código	Nombre completo de los alumnos

<b>Fecha:</b>	
---------------	--

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## Objetivos

Relacionar los conocimientos teóricos de clase con la práctica  
 Desarrollar las habilidades necesarias para comprender como funciona la herencia en programación orientada a objetos.

## Conceptos

- Herencia
- Clase base y clase derivada

## Materiales y Equipo

- 1 PC
- Entorno de desarrollo integrado compatible, que reconozca la sintaxis del lenguaje de programación Java.
- Máquina Virtual de Java versión 6 o superior (JRE) y Java Development Kit (JDK)

## Metodología o Desarrollo

- Desarrollar una aplicación en el lenguaje de programación Java.
- Crear las clases Figura, Triangulo, Rectángulo y Círculo con sus atributos y métodos necesarios para representar el concepto de herencia simple en Java.

## Resultados y Conclusiones

Relacionar los conocimientos obtenidos en clase con la práctica.  
 Comprender como funciona la herencia simple, clases base y clases derivadas.

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## PRÁCTICA 18. INTERBLOQUEO

<b>Carrera:</b>	
<b>Nombre de la materia:</b>	

<b>Código</b>	<b>Nombre completo de los alumnos</b>

<b>Fecha:</b>	
---------------	--

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## Objetivos

Relacionar los conocimientos teóricos de clase con la práctica.  
Comprender como funcionan los interbloqueos.

## Conceptos

- Hilos (Threads)
- Procesos
- Concurrencia
- Preemptive
- Non-preemptive

## Materiales y Equipo

- 1 PC con sistema operativo basado en debían instalado.
- Compilador gcc instalado.
- Un editor de texto.

## Metodología o Desarrollo

1. Codificar un programa en C utilizando la librería pthread y la función pthread\_create para crear hilos.
2. Crear dos contadores, y dos hilos secundarios, cada uno de los cuales atienden a un contador y elimina la condición de carrera sobre una variable, generando una variable cada uno.
3. Para compilar utilizar el parámetro -pthread de gcc.
4. Al finalizar la práctica realizar reporte.

## Resultados y Conclusiones

Relacionar los conocimientos obtenidos en clase con la práctica.  
Conocer el funcionamiento de los Hilos (Threads) en el lenguaje C.

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## PRÁCTICA 19. SIMULACIÓN DE MEMORIA

<b>Carrera:</b>	
<b>Nombre de la materia:</b>	

Código	Nombre completo de los alumnos

<b>Fecha:</b>	
---------------	--

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## Objetivos

Relacionar los conocimientos teóricos de clase con la práctica.

Comprender como crear hilos en el lenguaje Java.

## Conceptos

- Hilos (Threads)
- Procesos
- Concurrencia

## Materiales y Equipo

- 1 PC
- Entorno de desarrollo integrado compatible, que reconozca la sintaxis del lenguaje de programación Java.
- Máquina Virtual de Java versión 6 o superior (JRE) y Java Development Kit (JDK)

## Metodología o Desarrollo

1. Codificar un programa en el lenguaje Java utilizando la clase Graphics para dibujar gráficos en un JPanel.
2. Realizar una representación de la ejecución de programas y como se cargan en la memoria.
3. Al finalizar la práctica realizar reporte.

## Resultados y Conclusiones

Relacionar los conocimientos obtenidos en clase con la práctica.

Conocer el funcionamiento la memoria y representarlo de forma gráfica

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## **PRÁCTICA 20. GRUPOS, USUARIOS Y PERMISOS**

<b>Carrera:</b>	
<b>Nombre de la materia:</b>	

Código	Nombre completo de los alumnos

<b>Fecha:</b>	
---------------	--

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

### Objetivos.

Gestionar como los permisos de grupos, usuarios del sistema de archivos del sistema operativo Linux.

### Materiales y Equipo.

- Un equipo de cómputo con conexión a internet y con el software putty.
- Servidor del Centro Universitario con Linux

### Trabajo previo.

Conectarse por SSH al Servidor del Centro Universitario con Linux

### Metodología o Desarrollo.

1. Completa la siguiente tabla:

654	
	rwxr--r--
	rwxrwxrwx
520	
764	
	r-----

2. Crea los grupos oficina1 y oficina2.
3. Crea los usuarios paco y pablo. Estos usuarios deben pertenecer únicamente al grupo oficina1.
4. Crea los usuarios alba y nerea. Estos usuarios deben pertenecer únicamente al grupo oficina2.
5. Como usuario paco Crea un fichero con nombre topsecret.txt en su directorio de trabajo al que únicamente él tenga acceso, tanto de lectura como de escritura.
6. Crea otro fichero, también como usuario paco, con nombre ventas\_trimestre.txt al que tengan acceso, tanto para leer como para escribir todos los usuarios que pertenezcan al mismo grupo. Se deben dejar los permisos que haya por defecto para el dueño y para el resto de usuarios. Comprueba como usuario pablo que puedes modificar el fichero.
7. Como usuario albo, crea un fichero con nombre empleados.txt al que pueda acceder cualquier usuario para leer su contenido, y cualquier usuario del mismo grupo para leer o escribir.

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

8. Copia el fichero empleados.txt al directorio de trabajo de alumno (crea también el usuario alumno si no está creado). Cambia el propietario y el grupo al que pertenece el fichero, ahora debe ser alumno.

9. Como usuario pablo, copia un programa del directorio /usr/bin al directorio de trabajo con un nombre diferente. Por ejemplo, xclock se puede copiar como reloj. Mira los permisos de este programa. Comprueba que se puede ejecutar. Puede que sea necesario dar permiso para que otros usuarios distintos al actual puedan ejecutar aplicaciones en el entorno gráfico, basta con ejecutar como administrador: xhost +.

10. Cambia los permisos de reloj de tal forma que sólo lo pueda ejecutar el propietario del archivo.

11. Crea el usuario modesto, perteneciente a oficina2. Dentro de su directorio de trabajo, crea un directorio de nombre compartido\_con.todos.

12. Cambia de usuario en el entorno gráfico (botón salir y botón cambiar de usuario) y entra como modesto. Crea con OpenOffice.org Calc los ficheros telefono\_contactos.ods, gastos\_marzo.ods y sueldos.ods. Inserta varias entradas en cada uno de los ficheros y grábalo todo en el directorio compartido\_con.todos.

13. Da permiso de lectura a la carpeta compartido\_con.todos y a todos los ficheros que contenga para todos los usuarios.

14. Restringe el acceso de escritura sobre el fichero telefono\_contactos para que sólo lo puedan modificar los usuarios del grupo al que pertenece su propietario.

15. Cambia los permisos de gastos\_marzo para que sólo pueda modificarlo su propietario y leerlo cualquiera del mismo grupo.

16. Cambia los permisos de sueldos para que sólo su dueño tenga acceso a él, tanto para lectura como para escritura.

17. Si un usuario tiene permiso de lectura sobre un fichero, pero ese fichero se encuentra dentro de un directorio sobre el que no tiene permiso de lectura, ¿podrá leer el fichero?, haz la prueba.

### **Resultados y conclusiones.**

El alumno debe aprender a gestionar los permisos de ususarios y grupos del sistema de archivos del sistema operativo Linux.

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

**Referencias.**

Luis José Sánchez González, Manual práctico de Linux con ejercicios. IES Campanillas. Málaga. España, Cork Institute of Technology. Cork. Irlanda.

**Anexos (Rúbrica de práctica en laboratorio y rúbrica de reporte de práctica).**

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## PRÁCTICA 21. FICHEROS Y DIRECTORIOS

<b>Carrera:</b>	
<b>Nombre de la materia:</b>	

Código	Nombre completo de los alumnos

<b>Fecha:</b>	
---------------	--

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

### **Objetivos.**

Gestionar ficheros y directorios en un sistema operativo Linux.

### **Materiales y Equipo.**

- Un equipo de cómputo con conexión a internet y con el software putty.
- Servidor del Centro Universitario con Linux

### **Trabajo previo.**

Conectarse por SSH al Servidor del Centro Universitario con Linux

### **Metodología o Desarrollo.**

1. ¿En qué directorio se encuentran los ficheros de configuración del sistema?
2. Para entrar en un sistema Linux hace falta a) nombre de usuario, contraseña y dirección IP, b) nombre de usuario y contraseña o c) únicamente una contraseña.
3. Muestra el contenido del directorio actual.
4. Muestra el contenido del directorio que está justo a un nivel superior.
5. ¿En qué día de la semana naciste?, utiliza la instrucción cal para averiguarlo.
6. Muestra los archivos del directorio /bin
7. Suponiendo que te encuentras en tu directorio personal (/home/nombre), muestra un listado del contenido de /usr/bin a) con una sola línea de comando, b) moviéndote paso a paso por los directorios y c) con dos líneas de comandos.
8. Muestra todos los archivos que hay en /etc y todos los que hay dentro de cada subdirectorio, de forma recursiva (con un solo comando).
9. Muestra todos los archivos del directorio /usr/X11R6/bin ordenados por tamaño (de mayor a menor). Sólo debe aparecer el nombre de cada fichero, sin ninguna otra información adicional.
10. Muestra todos los archivos del directorio /etc ordenados por tamaño (de mayor a menor) junto con el resto de características, es decir, permisos, tamaño, fechas de la última modificación, etc. El tamaño de cada fichero debe aparecer en un formato “legible”, o sea, expresado en Kb, Mb, etc.
11. Muestra todos los archivos del directorio /bin ordenados por tamaño (de menor a mayor). Sólo debe aparecer el tamaño y el nombre de cada fichero, sin ninguna otra

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

información adicional. El tamaño de cada fichero debe aparecer en un formato “legible”, o sea, expresado en Kb, Mb, etc.

12. Muestra el contenido del directorio raíz utilizando como argumento de ls una ruta absoluta.

13. Muestra el contenido del directorio raíz utilizando como argumento de ls una ruta relativa. Suponemos que el directorio actual es /home/elena/documentos.

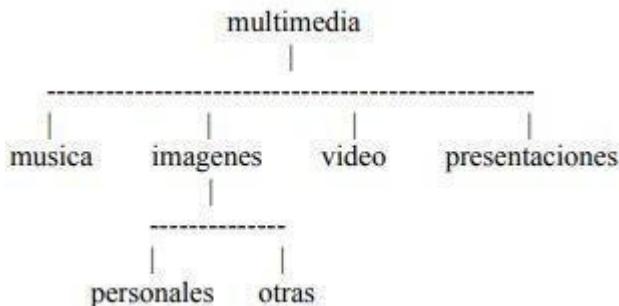
14. Crea el directorio gastos dentro del directorio personal.

15. ¿Qué sucede si se intenta crear un directorio dentro de /etc?

16. Muestra el contenido del fichero /etc/fstab.

17. Muestra las 10 primeras líneas del fichero /etc/bash.bashrc

18. Crea la siguiente estructura de directorios dentro del directorio de trabajo personal:



19. Crea un fichero vacío dentro del directorio música, con nombre estilos\_favoritos.txt

20. Utiliza tu editor preferido para abrir el fichero estilos\_favoritos.txt e introduce los estilos de música que más te gusten. Guarda los cambios y sal.

21. Muestra todo el contenido de estilos\_favoritos.txt

22. Muestra las 3 primeras líneas de estilos\_favoritos.txt

23. Muestra la última línea de estilos\_favoritos.txt

24. Muestra todo el contenido del fichero estilos\_favoritos.txt excepto la primera línea.

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

Se supone que no sabemos de antemano el número de líneas del fichero.

25. Muestra todos los archivos del directorio actual que son imágenes jpg.
26. Muestra todos los archivos del directorio /usr/bin que empiecen por la letra j.
27. Muestra los archivos que empiecen por k y tengan una a en la tercera posición, dentro del directorio /usr/bin.
28. Muestra los archivos del directorio /bin que terminen en n.
29. Muestra todos los archivos que hay en /etc y todos los que hay dentro de cada subdirectorio, de forma recursiva.
30. Crea un directorio en tu directorio de trabajo con nombre prueba. Copia el archivo gzip del directorio /bin al directorio prueba. Crea un duplicado de gzip con nombre gzip2 dentro de prueba.
31. Cambia el nombre de prueba a prueba2. Crea prueba3 en el mismo nivel que prueba2 y mueve todos los ficheros de prueba2 a prueba3. Borra prueba2.
32. Crea un fichero vacío con nombre “\*? ¿Hola caracola? \*”. ¿Se puede? En caso de que se pudiera, ¿sería recomendable poner nombres así? Razona la respuesta.
33. Crea un directorio con nombre multimedia\_pruebas y copia en él todo el contenido del directorio multimedia. A continuación, crea en multimedia/video/ dos ficheros, uno con nombre peliculas.txt y otro con nombre actores.txt. Edita el fichero peliculas.txt e introduce el nombre de tu película favorita. A continuación, crea en multimedia\_pruebas/video/ otro fichero que también tenga por nombre peliculas.txt, editalo y esta vez escribe el nombre de tus cinco películas favoritas. Ahora haz una copia de todo el contenido de multimedia en multimedia\_prueba de tal forma que sólo se copien los contenidos nuevos, es decir, si hay coincidencia en el nombre de un archivo se respetará el que se haya modificado más recientemente. Para comprobar que se ha hecho todo correctamente, basta mirar si en multimedia\_prueba/video está el archivo vacío actores.txt y además el archivo peliculas.txt debe contener 5 películas y no 1.

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

34. Borra el directorio multimedia/imagenes/otras. El sistema debe pedir al usuario que confirme el borrado.

35. Mueve el archivo peliculas.txt, que está dentro de multimedia/video, al directorio que está justo a un nivel superior. Ahora el archivo debe llamarse mis\_peliculas.txt en lugar de películas.

### **Resultados y conclusiones.**

El alumno debe aprender a gestionar el sistema de archivos del sistema operativo Linux.

### **Referencias.**

Luis José Sánchez González, Manual práctico de Linux con ejercicios. IES Campanillas. Málaga. España, Cork Institute of Technology. Cork. Irlanda.

### **Anexos (Rúbrica de práctica en laboratorio y rúbrica de reporte de práctica).**

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## **PRÁCTICA 22. CONCEPTOS BASICOS SOBRE LINUX**

<b>Carrera:</b>	
<b>Nombre de la materia:</b>	

Código	Nombre completo de los alumnos

<b>Fecha:</b>	
---------------	--

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

### **Objetivos.**

Aplicar los conceptos básicos sobre Linux.

### **Materiales y Equipo.**

- Un equipo de cómputo con conexión a internet y con el software putty.
- Servidor del Centro Universitario con Linux

### **Trabajo previo.**

Conectarse por SSH al Servidor del Centro Universitario con Linux

### **Metodología o Desarrollo.**

1. Un sistema operativo es a) un programa que permite al usuario realizar tareas específicas, b) un procesador de textos, c) un programa que permite al usuario interactuar con el ordenador y sus componentes o d) ninguna de las respuestas anteriores es correcta.
2. Una distribución de Linux es a) el núcleo del SO, junto con un programa de instalación y una selección de aplicaciones, b) el núcleo del SO, junto con un entorno gráfico y una selección de aplicaciones o c) las dos respuestas anteriores son correctas.
3. Cita a) alguna distribución de Linux buena en el apartado gráfico y en juegos 3D y b) alguna distribución de Linux con servidores preinstalados.
4. a) ¿Cuáles son las diez distribuciones de Linux más populares del último mes? b) y el último año?
5. a) ¿Quién creó la primera versión de Linux?, b) ¿con qué objetivo?
6. Para ejecutar comandos de Linux en una máquina remota mediante telnet hace falta instalar antes Linux en la máquina local. a) Verdadero b) Falso.
7. a) ¿La primera versión de Linux se creó para un tipo de máquina concreta o podía funcionar con cualquier hardware? b) Sigue lo mismo hoy día.
8. La mascota de Linux se llama a) Tuxedo, b) Tux, c) Pingu o d) todas las anteriores son correctas.
9. ¿Quién fue el artista encargado de dibujar la primera versión de la mascota de Linux?
10. Citar tres distribuciones de Linux ligeras (con pocas necesidades en cuanto a recursos hardware) que se puedan instalar en un lápiz de memoria.

### **Resultados y conclusiones.**

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

El alumno debe comprender las bases que fundan al sistema operativo Linux.

### **Referencias.**

Luis José Sánchez González, Manual práctico de Linux con ejercicios. IES Campanillas. Málaga. España, Cork Institute of Technology. Cork. Irlanda.

**Anexos (Rúbrica de práctica en laboratorio y rúbrica de reporte de práctica).**

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## **PRÁCTICA 23. INSTALACIÓN DE SISTEMA OPERATIVO BASADO EN UNIX**

<b>Carrera:</b>	
<b>Nombre de la materia:</b>	

Código	Nombre completo de los alumnos

<b>Fecha:</b>	
---------------	--

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## Objetivos

Relacionar los conocimientos teóricos de clase con la práctica.

Desarrollar las habilidades necesarias para el uso y comprensión de sistemas operativos.

## Conceptos

- Sistema operativo
- Software

## Materiales y Equipo

- 1 PC
- Software para montar Máquina virtual VirtualBox
- Imagen iso de sistema operativo Linux basado en debian

## Metodología o Desarrollo

1. Instalar el software virtualbox.
2. Activar en el BIOS la virtualización.
3. Crear una máquina virtual y asignarle recursos virtuales suficientes, e instalar el sistema operativo como máquina virtual a partir de la imagen ISO.

## Resultados y Conclusiones

Relacionar los conocimientos obtenidos en clase con la práctica.

Aprender a utilizar sistemas operativos basados en Unix.

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## PRÁCTICA 24. PROCESOS: FUNCIÓN FORK

<b>Carrera:</b>	
<b>Nombre de la materia:</b>	

Código	Nombre completo de los alumnos

<b>Fecha:</b>	
---------------	--

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## Objetivos

Relacionar los conocimientos teóricos de clase con la práctica.  
 Comprender como crear procesos hijos en el lenguaje C.  
 Aprender a pausar y detener procesos hijos.

## Conceptos

- Procesos
- Estados de un proceso

## Materiales y Equipo

- 1 PC con sistema operativo basado en debían instalado.
- Compilador gcc instalado.
- Un editor de texto.

## Metodología o Desarrollo

1. Codificar programas en C utilizando la función Fork para crear procesos hijos.
2. Utilizar la función Wait y Exec.
3. Compilar los programas utilizando el compilador de GNU GCC (GNU Compiler Collection)

## Resultados y Conclusiones

Relacionar los conocimientos obtenidos en clase con la práctica.  
 Conocer cómo funcionan los procesos y como pueden programarse, así como identificarlos en el monitor de procesos.

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## PRÁCTICA 25. GESTIÓN BÁSICA DE HILOS POSIX EN C

<b>Carrera:</b>	
<b>Nombre de la materia:</b>	

Código	Nombre completo de los alumnos

<b>Fecha:</b>	
---------------	--

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## Objetivos

Relacionar los conocimientos teóricos de clase con la práctica.  
 Comprender como crear hilos en el lenguaje C.

## Conceptos

- Hilos (Threads)
- Procesos
- Concurrencia

## Materiales y Equipo

- 1 PC con sistema operativo basado en debían instalado.
- Compilador gcc instalado.
- Un editor de texto.

## Metodología o Desarrollo

1. Codificar un programa en C utilizando la librería pthread y la función pthread\_create para crear hilos.
2. Crear un hilo secundario que incremente continuamente un valor de un contador, entre tanto el hilo principal se bloquee esperando una entrada por parte del usuario, y una vez que la recibe imprime el valor que alcanzó el contador y termina el proceso.
3. Para compilar utilizar el parámetro -pthread de gcc.
4. Al finalizar la práctica realizar reporte.

## Resultados y Conclusiones

Relacionar los conocimientos obtenidos en clase con la práctica.  
 Conocer el funcionamiento de los Hilos (Threads) en el lenguaje C

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## PRÁCTICA 26. HILOS INDEPENDIENTES DE LA PLATAFORMA CON JAVA

<b>Carrera:</b>	
<b>Nombre de la materia:</b>	

Código	Nombre completo de los alumnos

<b>Fecha:</b>	
---------------	--

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## Objetivos

Relacionar los conocimientos teóricos de clase con la práctica.

Comprender como crear hilos en el lenguaje Java

## Conceptos

- Hilos (Threads)
- Procesos
- Concurrencia

## Materiales y Equipo

- 1 PC
- Entorno de desarrollo integrado compatible, que reconozca la sintaxis del lenguaje de programación Java.
- Máquina Virtual de Java versión 6 o superior (JRE) y Java Development Kit (JDK)

## Metodología o Desarrollo

1. Codificar un programa Java utilizando la interface Runnable para crear hilos.
2. Crear una ventana utilizando la Clase JFrame y agregar un evento MousePressed.
3. En el evento MousePressed crear un hilo bajo demanda que incremente continuamente un valor de un contador en una etiqueta JLabel.

## Resultados y Conclusiones

Relacionar los conocimientos obtenidos en clase con la práctica.

Conocer el funcionamiento de los Hilos (Threads) en el lenguaje Java

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## PRÁCTICA 27. SIMULACIÓN DE MEMORIA

<b>Carrera:</b>	
<b>Nombre de la materia:</b>	

Código	Nombre completo de los alumnos

<b>Fecha:</b>	
---------------	--

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## Objetivos

Relacionar los conocimientos teóricos de clase con la práctica.

Comprender como crear hilos en el lenguaje Java.

## Conceptos

- Hilos (Threads)
- Procesos
- Concurrencia

## Materiales y Equipo

- 1 PC
- Entorno de desarrollo integrado compatible, que reconozca la sintaxis del lenguaje de programación Java.
- Máquina Virtual de Java versión 6 o superior (JRE) y Java Development Kit (JDK)

## Metodología o Desarrollo

1. Codificar un programa en el lenguaje Java utilizando la clase Graphics para dibujar gráficos en un JPanel.
2. Realizar una representación de la ejecución de programas y como se cargan en la memoria.
3. Al finalizar la práctica realizar reporte.

## Resultados o Conclusiones

Relacionar los conocimientos obtenidos en clase con la práctica.

Conocer el funcionamiento la memoria y representarlo de forma gráfica

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## **PRÁCTICA 28. CSS: CLASES, ID, SELECTORES**

Carrera:	
Nombre de la materia:	

Código	Nombre completo de los alumnos

Fecha:	
--------	--

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## Objetivos.

**Objetivo General:** El alumno conocerá la forma de implementar instrucciones CSS para resaltar el aspecto de una página web.

**Objetivo Específico:** Al término de esta práctica, el estudiante será capaz de realizar la correcta forma de insertar instrucciones CSS, utilizar los selectores adecuados de clase, id o elemento, así como de las propiedades básicas de diseño

## Materiales y Equipo.

Un equipo de cómputo con editor de código multiplataforma con los plugins necesarios que faciliten la rápida escritura de lenguaje, HTML, CSS, Bootstrap y JavaScript. Pueden ser Sublime 3, Visual Studio, Atom, Brackets, Notepad++.

## Trabajo previo.

Tener instalado el editor de código multiplataforma que haya elegido el alumno

## Metodología o Desarrollo.

1. Elegir un problema específico del banco de prácticas perteneciente a este objetivo.
2. Maquetar el documento HTML identificando los apartados necesarios de la página.
3. Realizar el documento con HTML que será la estructura de la interfaz gráfica de la página con las etiquetas necesarias para la resolución del problema.
4. Conocer y elegir la forma en que insertarán las declaraciones CSS en el documento ya sea de forma interna, externa o en línea.
5. Elegir la forma en que se le dictarán las instrucciones a los elementos HTML de la página, los cuales pueden ser por clase, identificadores o elemento.
6. Crear la clase o identificador y agregarle las declaraciones necesarias con las propiedades y valores, que den el formato al elemento HTML.
7. Verificar los resultados comparándola con la del problema especificado.

## Resultados y conclusiones.

El alumno debe poder crear por cuenta propia, páginas que tengan un aspecto más claro y estructurado, definiendo bien los apartados, clases e identificadores que serán utilizados por Bootstrap o JavaScript y poder cambiarle sus propiedades en tiempo de ejecución.

## Referencias.

<https://www.w3schools.com/css/default.asp>

## Anexos (Rúbrica de práctica en laboratorio y rúbrica de reporte de práctica).

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## **PRÁCTICA 29. BOOTSTRAP: LAYOUT Y COMPONENTES**

<b>Carrera:</b>	
<b>Nombre de la materia:</b>	

Código	Nombre completo de los alumnos

<b>Fecha:</b>	
---------------	--

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## Objetivos.

**Objetivo General:** El alumno conocerá Bootstrap como el marco de trabajo más popular para HTML, CSS y JavaScript para desarrollar sitios web responsivos y móviles.

**Objetivo Específico:** Al término de esta práctica, el estudiante será capaz de crear páginas web responsivas, utilizando contenedores para especificar sus apartados, así como, conocer e implementar los componentes prediseñados que Bootstrap ofrece para un rápido diseño y estructura de una página, además de poder realizar las modificaciones necesarias con instrucciones CSS en caso de requerirse.

## Materiales y Equipo.

Un equipo de cómputo con editor de código multiplataforma colo los plugins necesarios que faciliten la rápida escritura de lenguaje, HTML, CSS, Bootstrap y JavaScript. Pueden ser Sublime 3, Visual Studio, Atom, Brackets, Notepad++.

## Trabajo previo.

Tener instalado el editor de código multiplataforma que haya elegido el alumno

## Metodología o Desarrollo.

1. Elegir un problema específico del banco de prácticas perteneciente a este objetivo.
2. Maquetar el documento con las etiquetas HTML que permitan identificar los apartados como containers principales (header, footer, nav, aside, section, article, etc)
3. Seleccionar los componentes prediseñados de Bootstrap que realicen la tarea que se requiere en cada apartado e incrustarlos en la página.
4. Modificar o adaptar los componentes cambiando imágenes, vínculos, colores, rutas, etc.
5. Adaptar la interfaz para los formatos móvil o pc agregando las clases Bootstrap necesarias.
6. Rectificar en caso de ser necesario con instrucciones CSS la interfaz

## Resultados y conclusiones.

El alumno debe poder crear por cuenta propia, páginas responsivas que se adapten en interfaz para móviles y pc, que tengan un aspecto más claro y estructurado, definiendo bien los apartados, clases e identificadores de Bootstrap que serán utilizados por JavaScript y poder cambiarle sus propiedades en tiempo de ejecución.

## Referencias.

- <https://www.w3schools.com/bootstrap4/default.asp>
- <https://getbootstrap.com/>

## Anexos (Rúbrica de práctica en laboratorio y rúbrica de reporte de práctica).

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## PRÁCTICA 30. BOOTSTRAP: PLANTILLAS

Carrera:	
Nombre de la materia:	

Código	Nombre completo de los alumnos

Fecha:	
--------	--

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## Objetivos.

**Objetivo General:** El alumno conocerá Bootstrap como el marco de trabajo más popular para HTML, CSS y JavaScript para desarrollar sitios web responsivos y móviles.

**Objetivo Específico:** Al término de esta práctica, el estudiante será capaz de adaptar plantillas prediseñadas en Bootstrap a páginas web responsivas, utilizando contenedores para especificar sus apartados, así como, conocer e implementar los componentes prediseñados que Bootstrap ofrece para un rápido diseño y estructura de una página, además de poder realizar las modificaciones necesarias con instrucciones CSS en caso de requerirse.

## Materiales y Equipo.

Un equipo de cómputo con editor de código multiplataforma colo los plugins necesarios que faciliten la rápida escritura de lenguaje, HTML, CSS, Bootstrap y JavaScript. Pueden ser Sublime 3, Visual Studio, Atom, Brackets, Notepad++.

## Trabajo previo.

Tener instalado el editor de código multiplataforma que haya elegido el alumno.

## Metodología o Desarrollo.

1. Elegir un problema específico del banco de prácticas perteneciente a este objetivo.
2. Elegir la plantilla que más se identifique para la resolución del problema desde páginas como:  
<https://w3layouts.com/>  
<https://bootstrapmade.com/>  
<https://startbootstrap.com/>
3. Seleccionar y agregar los componentes prediseñados de Bootstrap faltantes que realicen la tarea que se requiere en cada apartado e incrustarlos en la página.
4. Modificar o adaptar los componentes cambiando imágenes, vínculos, colores, rutas, etc.
5. Adaptar la interfaz para los formatos móvil o pc agregando las clases Bootstrap necesarias.
6. Rectificar en caso de ser necesario con instrucciones CSS la interfaz.

## Resultados y conclusiones.

El alumno debe poder adaptar plantillas a páginas responsivas que se adapten en interfaz para móviles y pc, que tengan un aspecto más claro y estructurado, definiendo bien los apartados, clases e identificadores de Bootstrap que serán utilizados por JavaScript y poder cambiarle sus propiedades en tiempo de ejecución.

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

**Referencias.**

<https://www.w3schools.com/bootstrap4/default.asp>  
<https://getbootstrap.com/>  
<https://w3layouts.com/>  
<https://bootstrapmade.com/>  
<https://startbootstrap.com/>

**Anexos (Rúbrica de práctica en laboratorio y rúbrica de reporte de práctica).**

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## PRÁCTICA 31. JavaScript: DOM DE LA PAGINA

<b>Carrera:</b>	
<b>Nombre de la materia:</b>	

Código	Nombre completo de los alumnos

<b>Fecha:</b>	
---------------	--

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Computación</b>	Fecha de Revisión
		Junio 2020

## Objetivos.

**Objetivo General:** El alumno conocerá el lenguaje de programación más popular de la web, JavaScript para crear páginas web dinámicas.

**Objetivo Específico:** Al término de esta práctica, el estudiante conocerá la estructura de objetos (DOM Document Object Model) que genera el navegador cuando se carga un documento y se puede alterar mediante Javascript para cambiar dinámicamente los contenidos y aspecto de la página

## Materiales y Equipos.

Un equipo de cómputo con editor de código multiplataforma colo los plugins necesarios que faciliten la rápida escritura de lenguaje, HTML, CSS, Bootstrap y JavaScript. Pueden ser Sublime 3, Visual Studio, Atom, Brackets, Notepad++.

## Trabajo previo.

Tener instalado el editor de código multiplataforma que haya elegido el alumno

## Metodología o Desarrollo.

1. Elegir un problema específico del banco de prácticas perteneciente a este objetivo.
2. Crear el archivo JavaScript externo al documento HTML y vincularlo
3. Crear identificadores a los tags (etiquetas) que sean necesarias para modificar datos o estructura en tiempo de ejecución.
4. Conocer y aplicar los métodos de JavaScript para obtener y modificar el DOM
  - getElementById
  - getElementsByName
  - getElementsByClassName
  - getAttribute
  - getAttributeNode
  - hasAttribute
  - innerHTML
  - innerText
  - remove ()
  - removeAttribute()

5. Aplicar los requeridos respecto de la práctica en cuestión

**Resultados y conclusiones.**

El alumno debe poder obtener objetos del DOM de la página y hacerles cambios de interfaz o valores en tiempo de ejecución por medio de JavaScript y así cambiar sus propiedades.

**Referencias.**

<https://www.w3schools.com/js/default.asp>

**Anexos (Rúbrica de práctica en laboratorio y rúbrica de reporte de práctica).**

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Programación Imperativa</b>	Fecha de Revisión

## **PRÁCTICA 32. JavaScript: FUNCTIONS, OBJECTS, EVENTS**

<b>Carrera:</b>	
<b>Nombre de la materia:</b>	

Código	Nombre completo de los alumnos

<b>Fecha:</b>	
---------------	--

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Programación Imperativa</b>	Fecha de Revisión

## Objetivos.

**Objetivo General:** El alumno conocerá el lenguaje de programación más popular de la web, JavaScript para crear páginas web dinámicas.

**Objetivo Específico:** Al término de esta práctica, el estudiante aprenderá a estructurar funciones, objetos y eventos para estructurar el programa y la página, y con ello, alterar mediante Javascript, dinámicamente los contenidos y aspecto de la página.

## Materiales y Equipos.

Un equipo de cómputo con editor de código multiplataforma colo los plugins necesarios que faciliten la rápida escritura de lenguaje, HTML, CSS, Bootstrap y JavaScript. Pueden ser Sublime 3, Visual Studio, Atom, Brackets, Notepad++.

## Trabajo previo.

Tener instalado el editor de código multiplataforma que haya elegido el alumno.

## Metodología o Desarrollo.

1. Elegir un problema específico del banco de prácticas perteneciente a este objetivo.
2. Crear el archivo JavaScript externo al documento HTML y vincularlo
3. Crear identificadores a los tags (etiquetas) que sean necesarias para modificar datos o estructura en tiempo de ejecución.
4. Conocer la forma de estructurar funciones vacías y de retorno, con y sin parámetros.
5. Comprender la forma de estructurar las variables tipo objeto, definiéndolos, agregar y acceder a sus propiedades.
6. Conocer los diferentes tipos de eventos que se pueden utilizar en una página y su aplicación.
7. Utilizar los eventos, crear las funciones y estructurar objetos para resolver el problema en cuestión

## Resultados y conclusiones.

El alumno debe poder obtener objetos del DOM de la página y hacerles cambios de interfaz o valores en tiempo de ejecución por medio de JavaScript y así cambiar sus propiedades.

## Referencias.

<https://www.w3schools.com/js/default.asp>

## Anexos (Rúbrica de práctica en laboratorio y rúbrica de reporte de práctica).

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Programación Imperativa</b>	Fecha de Revisión

## PRÁCTICA 33. JavaScript: FORMULARIOS

Carrera:	
Nombre de la materia:	

Código	Nombre completo de los alumnos

Fecha:	
--------	--

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Programación Imperativa</b>	Fecha de Revisión

## Objetivos.

**Objetivo General:** El alumno conocerá el lenguaje de programación más popular de la web, JavaScript para crear páginas web dinámicas.

**Objetivo Específico:** Al término de esta práctica, el estudiante aprenderá a validar los datos de los formularios directamente en el navegador del usuario y con ello evitar recargar la página en caso de que el usuario cometa errores al llenar los formularios.

## Materiales y Equipos.

Un equipo de cómputo con editor de código multiplataforma colo los plugins necesarios que faciliten la rápida escritura de lenguaje, HTML, CSS, Bootstrap y JavaScript. Pueden ser Sublime 3, Visual Studio, Atom, Brackets, Notepad++.

## Trabajo previo.

Tener instalado el editor de código multiplataforma que haya elegido el alumno.

## Metodología o Desarrollo.

1. Elegir un formulario específico del banco de prácticas perteneciente a este objetivo.
2. Crear el archivo JavaScript externo al documento HTML y vincularlo.
3. Crear el formulario o adaptar una plantilla del mismo mediante Bootstrap
4. Crear identificadores a los tags (etiquetas) de los campos del formulario.
5. Conocer y acceder a las propiedades type, name, form, value de los elementos.
6. Obtener el valor de los campos de tipo input, RadioButton, checkbox, Select.
7. Crear restricciones para:
8. Evitar el envío duplicado de un formulario
9. Limitar el tamaño de caracteres de un textarea
10. Restringir los caracteres permitidos en un cuadro de texto
11. Validar un campo de texto obligatorio
12. Validar un campo de texto con valores numéricos
13. Validar que se ha seleccionado una opción de una lista
14. Validar una dirección de email
15. Validar una fecha
16. Validar un número de teléfono
17. Validar que un CheckBox o un RadioButton ha sido seleccionado

## Resultados y conclusiones.

El alumno debe poder validar formularios mediante el lenguaje JavaScript para poder enviar datos libres de errores al sistema de almacenamiento o base de datos

## Referencias.

<https://www.w3schools.com/js/default.asp>

<https://www.arkaitzgarro.com/javascript/capitulo-16.html>

Anexos (Rúbrica de práctica en laboratorio y rúbrica de reporte de práctica).

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Programación Imperativa</b>	Fecha de Revisión

## PRÁCTICA 34. DESARROLLO DE APLICACIÓN CLIENTE SERVIDOR CON ENVÍO DE OBJETOS

<b>Carrera:</b>	
<b>Nombre de la materia:</b>	

Código	Nombre completo de los alumnos

<b>Fecha:</b>	
---------------	--

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Programación Imperativa</b>	Fecha de Revisión

## Objetivos

Relacionar los conocimientos teóricos de clase con la práctica  
 Desarrollar las habilidades necesarias para comprender como funciona la programación C-S a bajo nivel con Sockets.  
 Determinar cómo establecer una conexión única para cada cliente.

## Conceptos

- Creación/ implementación y supresión de sockets.
- Desarrollo del lado del servidor con sockets.
- Desarrollo del lado del cliente con sockets.

## Materiales y Equipos

- 1 PC
- Entorno de desarrollo integrado compatible, que reconozca la sintaxis del lenguaje de programación Java.
- Máquina Virtual de Java versión 6 o superior (JRE) y Java Development Kit (JDK)

## Metodología o Desarrollo

1. Desarrollar una aplicación en el lenguaje de programación Java utilizando sockets.
2. Crear una clase para el servidor, declarar y crear una instancia de la ServerSocket, aceptar la conexión.
3. Declarar y crear la instancia para el Socket ambas del paquete Java.net y las clases ObjectInputStream/ ObjectOutputStream para manejar los flujos de entrada/salida de los sockets, la aplicación deberá administrar peticiones de manera concurrente.
4. Al finalizar la práctica realizar reporte.

## Resultados y Conclusiones

Relacionar los conocimientos obtenidos en clase con la práctica.  
 Conocer el funcionamiento a bajo nivel de sockets.  
 Desarrollar habilidades necesarias para desarrollar aplicaciones productivas cliente-servidor.

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Programación Imperativa</b>	Fecha de Revisión

## PRÁCTICA 35. DESARROLLO DE APLICACIÓN SERVIDOR TCP CONCURRENTE

<b>Carrera:</b>	
<b>Nombre de la materia:</b>	

Código	Nombre completo de los alumnos

<b>Fecha:</b>	
---------------	--

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Programación Imperativa</b>	Fecha de Revisión

## Objetivos

Relacionar los conocimientos teóricos de clase con la práctica  
 Desarrollar las habilidades necesarias para comprender como funciona la programación C-S a bajo nivel con Sockets.  
 Determinar cómo establecer una conexión única para cada cliente.

## Conceptos

- Concepto de socket.
- Dominios y Tipos de sockets.
- Creación/ implementación y supresión de sockets.
- Desarrollo del lado del servidor con sockets.
- Desarrollo del lado del cliente con sockets.

## Materiales y Equipo

- 1 PC
- Entorno de desarrollo integrado compatible, que reconozca la sintaxis del lenguaje de programación Java.
- Máquina Virtual de Java versión 6 o superior (JRE) y Java Development Kit (JDK)

## Metodología o Desarrollo

Desarrollar una aplicación en el lenguaje de programación Java utilizando sockets, crear una clase para el servidor, declarar y crear una instancia de la clase ServerSocket y aceptar la conexión. Declarar y crear la instancia para el Socket ambas del paquete Java.net y las clases BufferedReader/PrintWriter y/o ObjectInputStream/ObjectOutputStream para manejar los flujos de entrada/salida de los sockets además de una clase definida por el programador para la conexión, la aplicación deberá administrar desde el servidor la conexión de 2 clientes.

## Resultados y Conclusiones

Relacionar los conocimientos obtenidos en clase con la práctica.  
 Conocer el funcionamiento a bajo nivel de sockets.  
 Desarrollar habilidades necesarias para que el alumno pueda desarrollar aplicaciones cliente-servidor multi cliente.

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Programación Imperativa</b>	Fecha de Revisión

## **PRÁCTICA 36. DESARROLLO DE APLICACIÓN JAVA-RMI**

<b>Carrera:</b>	
<b>Nombre de la materia:</b>	

Código	Nombre completo de los alumnos

<b>Fecha:</b>	
---------------	--

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Programación Imperativa</b>	Fecha de Revisión

## Objetivos

Relacionar los conocimientos teóricos de clase con la práctica  
 Desarrollar las habilidades necesarias para comprender como funciona la programación con Java RMI.

## Conceptos

- El API Java RMI.
- Jerarquía de objetos RMI.
- Llamadas a procedimientos remotos.

## Materiales y Equipo

- 1 PC.
- Entorno de desarrollo integrado compatible, que reconozca la sintaxis del lenguaje de programación Java.
- Máquina Virtual de Java versión 6 o superior (JRE) y Java Development Kit (JDK).

## Metodología o Desarrollo

1. Desarrolle una aplicación en el lenguaje de programación Java utilizando RMI.
2. Defina la interfaz Remote.
3. Implemente la interfaz Remote.
4. Desarrollar el Servidor.
5. Desarrollar el Cliente.
6. Desarrollar el código para envío-recepción de información.
7. Al finalizar la práctica realizar reporte.

## Resultados y Conclusiones

Relacionar los conocimientos obtenidos en clase con la práctica.

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Programación Imperativa</b>	Fecha de Revisión

## **PRÁCTICA 37. PUBLICAR SERVICIO DESDE UNA APLICACIÓN WEB**

<b>Carrera:</b>	
<b>Nombre de la materia:</b>	

Código	Nombre completo de los alumnos

<b>Fecha:</b>	
---------------	--

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Programación Imperativa</b>	Fecha de Revisión

## Objetivos

Relacionar los conocimientos teóricos de clase con la práctica  
 Desarrollar las habilidades necesarias para comprender como funciona la programación de servicios web.

## Conceptos

Visión general de servicios web XML.

## Materiales y Equipo

- 1 PC.
- Entorno de desarrollo integrado compatible, que reconozca la sintaxis del lenguaje de programación Java.
- Máquina Virtual de Java versión 6 o superior (JRE) y Java Development Kit (JDK).

## Metodología o Desarrollo

Desarrollar una aplicación web y publicar un servicio para ser consumido de forma remota, enviando parámetros, al finalizar la práctica realizar reporte.

## Resultados y Conclusiones

Relacionar los conocimientos obtenidos en clase con la práctica.  
 Conocer el funcionamiento básico de un servicio web.  
 Desarrollar habilidades necesarias para desarrollar aplicaciones distribuidas.

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Programación Imperativa</b>	Fecha de Revisión

## **PRÁCTICA 38. PUBLICAR SERVICIO DESDE UNA APLICACIÓN WEB**

<b>Carrera:</b>	
<b>Nombre de la materia:</b>	

Código	Nombre completo de los alumnos

<b>Fecha:</b>	
---------------	--

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Programación Imperativa</b>	Fecha de Revisión

## Objetivos

Relacionar los conocimientos teóricos de clase con la práctica  
 Desarrollar las habilidades necesarias para comprender como funciona la programación de servicios web.

## Conceptos

Visión general de servicios web XML.

## Materiales y Equipo

- 1 PC.
- Entorno de desarrollo integrado compatible, que reconozca la sintaxis del lenguaje de programación Java.
- Máquina Virtual de Java versión 6 o superior (JRE) y Java Development Kit (JDK).

## Metodología o Desarrollo

Desarrollar una aplicación web para consumir un servicio previamente creado, enviando parámetros diferentes, al finalizar la práctica realizar reporte.

## Resultados y Conclusiones

Relacionar los conocimientos obtenidos en clase con la práctica.

Conocer cómo se consume un servicio web.

Desarrollar habilidades necesarias para desarrollar aplicaciones distribuidas y con acceso a datos.

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Programación Imperativa</b>	Fecha de Revisión

## **PRÁCTICA 39. APP CON DISEÑO BASADO EN RESTRICCIONES E ÍCONOPUBLICAR SERVICIO DESDE UNA APLICACIÓN WEB**

<b>Carrera:</b>	
<b>Nombre de la materia:</b>	

Código	Nombre completo de los alumnos

<b>Fecha:</b>	
---------------	--

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Programación Imperativa</b>	Fecha de Revisión

## Objetivos

Relacionar los conocimientos teóricos de clase con la práctica.

Desarrollar las habilidades necesarias para programar aplicaciones en Java para Android.

## Conceptos

- ConstraintLayout
- Layout
- Activity

## Materiales y Equipo

- 1 PC con Windows 8 o superior y al menos 4 GB de Ram, recomendado 8GB
- Entorno de desarrollo integrado Android Studio
- Simulador Android o teléfono físico con Android 4.3 o superior

## Metodología o Desarrollo

1. Crear una aplicación con el IDE Android Studio.
2. Arrastrar componentes a la actividad principal.
3. Guardar una imagen en la carpeta Drawable.
4. Modificar el archivo Manifest para elegir un ícono.
5. Ejecutar la aplicación en el simulador o celular con Android

## Resultados y Conclusiones

Relacionar los conocimientos obtenidos en clase con la práctica.

Conocer el funcionamiento del diseño ConstraintLayout.

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Programación Imperativa</b>	Fecha de Revisión

## PRÁCTICA 40. APP UTILIZANDO LINEAR LAYOUT

<b>Carrera:</b>	
<b>Nombre de la materia:</b>	

Código	Nombre completo de los alumnos

<b>Fecha:</b>	
---------------	--

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Programación Imperativa</b>	Fecha de Revisión

## Objetivos

Relacionar los conocimientos teóricos de clase con la práctica  
 Desarrollar las habilidades necesarias para comprender como funciona el diseño LinearLayout.

## Conceptos

- 1 PC con Windows 8 o superior y al menos 4 GB de Ram, recomendado 8GB
- Entorno de desarrollo integrado Android Studio
- Simulador Android o teléfono físico con Android 4.3 o superior

## Materiales y Equipo

- 1 PC con Windows 8 o superior y al menos 4 GB de Ram, recomendado 8GB
- Entorno de desarrollo integrado Android Studio
- Simulador Android o teléfono físico con Android 4.3 o superior

## Metodología o Desarrollo

1. Crear una aplicación con el IDE Android Studio seleccionando Empty Activity.
2. Modificar el diseño Constraint por defecto a un LinearLayout y elegir una orientación.
3. Agregar componentes dentro del LinearLayout principal usando pesos en los subcomponentes y también anidar a una celda un LinearLayout con su propia orientación y componentes.

## Resultados y Conclusiones

Relacionar los conocimientos obtenidos en clase con la práctica.  
 Conocer el funcionamiento del diseño LinearLayout.

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Programación Imperativa</b>	Fecha de Revisión

## PRÁCTICA 41. USO DE INTENTS

<b>Carrera:</b>	
<b>Nombre de la materia:</b>	

Código	Nombre completo de los alumnos

<b>Fecha:</b>	
---------------	--

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Programación Imperativa</b>	Fecha de Revisión

## Objetivos

Relacionar los conocimientos teóricos de clase con la práctica

Desarrollar las habilidades necesarias para comprender como usar los intents para disparar servicios o abrir otras actividades.

## Conceptos

- Interface de usuario
- Activity
- Intent

## Materiales y Equipo

-1 PC con Windows 8 o superior y al menos 4 GB de Ram, recomendado 8GB

-Entorno de desarrollo integrado Android Studio

-Simulador Android o teléfono físico con Android 4.3 o superior

## Metodología o Desarrollo

1. Crear una aplicación con el IDE Android Studio seleccionando Empty Activity.
2. Agregar varios componentes gráficos para llamar intents para abrir una url, realizar una llamada o enviar un mensaje.

## Resultados y Conclusiones

Relacionar los conocimientos obtenidos en clase con la práctica.

Conocer el funcionamiento de los intents en Android.

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Programación Imperativa</b>	Fecha de Revisión

## PRÁCTICA 42. USO DE INTENTS ABRIR OTRA ACTIVIDAD Y ENVIAR DATOS

<b>Carrera:</b>	
<b>Nombre de la materia:</b>	

Código	Nombre completo de los alumnos

<b>Fecha:</b>	
---------------	--

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Programación Imperativa</b>	Fecha de Revisión

## Objetivos

Relacionar los conocimientos teóricos de clase con la práctica

Desarrollar las habilidades necesarias para comprender como usar los intents para disparar servicios o abrir otras actividades.

## Conceptos

- Interface de usuario
- Activity
- Intent

## Materiales y Equipo

-1 PC con Windows 8 o superior y al menos 4 GB de Ram, recomendado 8GB

-Entorno de desarrollo integrado Android Studio

-Simulador Android o teléfono físico con Android 4.3 o superior

## Metodología o Desarrollo

1. Crear una aplicación con el IDE Android Studio seleccionando Empty Activity.
2. En el xml de la actividad arrastrar 2 EditText y un Botón en el diseño y anclarlos usando restricciones.
3. Crear una segunda actividad.
4. Escribir en los archivos java el código para abrir la segunda actividad y el envío y recepción de datos a partir de los componentes visuales.

## Resultados y Conclusiones

Relacionar los conocimientos obtenidos en clase con la práctica.

Conocer el funcionamiento de los intents para abrir otras actividades y el envío de datos entre actividades.

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Programación Imperativa</b>	Fecha de Revisión

## PRÁCTICA 43. DISEÑO PORTRAIT Y LANDSCAPE

Carrera:	
Nombre de la materia:	

Código	Nombre completo de los alumnos

Fecha:	
--------	--

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Programación Imperativa</b>	Fecha de Revisión

## Objetivos

Relacionar los conocimientos teóricos de clase con la práctica

Desarrollar las habilidades necesarias para comprender como usar un diseño panorámico personalizado.

## Conceptos

- Interface de usuario
- Activity
- Intent

## Materiales y Equipo

-1 PC con Windows 8 o superior y al menos 4 GB de Ram, recomendado 8GB

-Entorno de desarrollo integrado Android Studio

-Simulador Android o teléfono físico con Android 4.3 o superior

## Metodología o Desarrollo

1. Crear una aplicación con el IDE Android Studio seleccionando Empty Activity.
2. Crear un recurso XML para una actividad Landscape y hacer un diseño personalizado diferente al portrait.

## Resultados y Conclusiones

Relacionar los conocimientos obtenidos en clase con la práctica.

Conocer cómo desarrollar una actividad con diseño panorámico y normal.

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Programación Imperativa</b>	Fecha de Revisión

## **PRÁCTICA 44. LISTA DE COMPONENTES PERSONALIZADOS**

<b>Carrera:</b>	
<b>Nombre de la materia:</b>	

Código	Nombre completo de los alumnos

<b>Fecha:</b>	
---------------	--

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Programación Imperativa</b>	Fecha de Revisión

## Objetivos

Relacionar los conocimientos teóricos de clase con la práctica.

Desarrollar las habilidades necesarias para comprender como usar un diseño panorámico personalizado.

## Conceptos

- Interface de usuario
- Activity
- Lista personalizada

## Materiales y Equipo

-1 PC con Windows 8 o superior y al menos 4 GB de Ram, recomendado 8GB

-Entorno de desarrollo integrado Android Studio

-Simulador Android o teléfono físico con Android 4.3 o superior

## Metodología o Desarrollo

1. Crear una aplicación con el IDE Android Studio seleccionando Empty Activity.
2. Crear un xml para un componente personalizado que será un elemento de la lista.
3. Crear un xml que será la lista.
4. Escribir las etiquetas xml y el código Java para agregar de forma dinámica nuevos elementos a la lista.

## Resultados y Conclusiones

Relacionar los conocimientos obtenidos en clase con la práctica.

Aprender a desarrollar listas con componentes personalizados en la interfaz de Android.

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Programación Imperativa</b>	Fecha de Revisión

## PRÁCTICA 45. ACELERÓMETRO

<b>Carrera:</b>	
<b>Nombre de la materia:</b>	

Código	Nombre completo de los alumnos

<b>Fecha:</b>	
---------------	--

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Programación Imperativa</b>	Fecha de Revisión

## Objetivos

Relacionar los conocimientos teóricos de clase con la práctica

Desarrollar las habilidades necesarias para comprender como usar los sensores del dispositivo Android.

## Conceptos

-Sensores

## Materiales y Equipo

-1 PC con Windows 8 o superior y al menos 4 GB de Ram, recomendado 8GB

-Entorno de desarrollo integrado Android Studio

-Simulador Android o teléfono físico con Android 4.3 o superior con acelerómetro.

## Metodología o Desarrollo

1. Crear una aplicación con el IDE Android Studio seleccionando Empty Activity.
2. Utilizar las Clases SensorManager y Sensor de Android, para trabajar con sensores.
3. Sobreescribir el método `onSensorChanged` para obtener los valores del sensor acelerómetro y mostrarlo en componentes visuales xml.

## Resultados y Conclusiones

Relacionar los conocimientos obtenidos en clase con la práctica.

Aprender a desarrollar aplicaciones con sensores en Android.

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Programación Imperativa</b>	Fecha de Revisión

## PRÁCTICA 46. BASE DE DATOS SQLite

<b>Carrera:</b>	
<b>Nombre de la materia:</b>	

Código	Nombre completo de los alumnos

<b>Fecha:</b>	
---------------	--

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Programación Imperativa</b>	Fecha de Revisión

## Objetivos

Relacionar los conocimientos teóricos de clase con la práctica  
 Desarrollar las habilidades necesarias para programar aplicaciones Android con acceso a datos y persistencia de datos.

## Conceptos

-Base de datos Embebida SQLite

## Materiales y Equipo

- 1 PC con Windows 8 o superior y al menos 4 GB de Ram, recomendado 8GB
- Entorno de desarrollo integrado Android Studio
- Simulador Android o teléfono físico con Android 4.3 o superior
- Software SQLite Studio

## Metodología o Desarrollo

1. Crear una aplicación con el IDE Android Studio seleccionando Empty Activity y utilizando las clases SQLiteOpenHelper y SQLiteDatabase para manipulación de base de datos SQLite.
2. Crear una base de datos utilizando SQLite Studio.  
 Implementar el código para realizar afectaciones en una base de datos SQLite.

## Resultados y Conclusiones

Relacionar los conocimientos obtenidos en clase con la práctica.  
 Aprender a desarrollar aplicaciones con base de datos SQLite y Android.

	<b>UNIVERSIDAD DE GUADALAJARA</b> CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Computación
	<b>Manual de Prácticas</b> <b>Programación Imperativa</b>	Fecha de Revisión

## PRÁCTICA 47. DESARROLLO DE APLICACIÓN PARA ESTABLECER LA COMUNICACIÓN BLUETOOTH

Carrera:	
Nombre de la materia:	

Código	Nombre completo de los alumnos

Fecha:	
--------	--

	<b>UNIVERSIDAD DE GUADALAJARA</b> <b>CENTRO UNIVERSITARIO DE LA COSTA SUR</b> <b>DEPARTAMENTO DE INGENIERÍAS</b>	Academia de Computación
	<b>Manual de Prácticas</b> <b>Programación Imperativa</b>	Fecha de Revisión

## Objetivos

Relacionar los conocimientos teóricos de clase con la práctica  
 Desarrollar las habilidades necesarias para comprender como funciona la comunicación inalámbrica Bluetooth.

## Conceptos

-Bluetooth

## Materiales y Equipo

- 1 PC con Windows 8 o superior y al menos 4 GB de Ram, recomendado 8GB
- Entorno de desarrollo integrado Android Studio
- Simulador Android o teléfono físico con Android 4.3 o superior
- 1 PC con Bluetooth.
- 1 Dispositivo con Bluetooth.
- Entorno de desarrollo integrado compatible, que reconozca la sintaxis del lenguaje de programación Java.
- Máquina Virtual de Java versión 6 o superior (JRE) y Java Development Kit (JDK)

## Metodología o Desarrollo

Desarrollar una aplicación en el lenguaje de programación Java utilizando Bluetooth para detectar dispositivos con Bluetooth habilitado, utilizando la librería Bluecove. Al terminar realizar reporte correspondiente a la práctica.

## Resultados y Conclusiones

Relacionar los conocimientos obtenidos en clase con la práctica.  
 Conocer el funcionamiento de la tecnología inalámbrica Bluetooth.

## DIRECTORIO



### UNIVERSIDAD DE GUADALAJARA

DR. RICARDO VILLANUEVA LOMELÍ  
RECTOR GENERAL

DR. HÉCTOR RAÚL SOLÍS GADEA  
VICERRECTOR EJECUTIVO

MTR. GUILLERMO ARTURO GÓMEZ MATA  
SECRETARIO GENERAL



### CENTRO UNIVERSITARIO DE LA COSTA SUR

DRA. LILIA VICTORIA OLIVER SÁNCHEZ  
RECTORA

DR. HIRINEO MARTÍNEZ BARRAGÁN  
SECRETARIO ACADÉMICO

DR. LUIS CARLOS GÁMEZ ADAME  
SECRETARIO ADMINISTRATIVO

MTR. ENRIQUE JARDEL PELÁEZ  
DIRECTOR DE LA DIVISIÓN DE DESARROLLO REGIONAL

DR. DANIEL EDÉN RAMÍREZ ARREOLA  
JEFE DEL DEPARTAMENTO DE INGENIERÍAS

Av. Independencia Nacional No. 151, Axtlán de Navarro, Jalisco, C.P. 48900  
Tel. (317) 382 5010 [www.cucsur.udg.mx](http://www.cucsur.udg.mx)



Centro Universitario de la Costa Sur



CU Costa Sur UdeG @CUCSur



CU Costa Sur @cucostasur