
	Universidad Nacional de San Agustín de Arequipa Facultad de Ingeniería de Producción y Servicios Escuela Profesional de Ingeniería de Sistemas	
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		
Aprobación: 2022/03/01	Código: GUIA-PRLD-001	Página: 1

INFORME DE LABORATORIO

INFORMACION BASICA					
ASIGNATURA:	Tecnología de Objetos				
TITULO DE LA PRACTICA:	POO: Herencia, Composición, Agregación y Dependencia				
NUMERO DE LA PRACTICA:	02	AÑO LECTIVO:	2025 - B	N° SEMESTRE:	VI
FECHA DE PRESENTACION:	21 / 09 / 2025	HORA DE PRESENTACION:	--:-- PM		
INTEGRANTE (s): ▪ Huayhua Hillpa, Yourdyy Yossimar				NOTA:	
DOCENTE (s): ▪ Mg. Escobedo Quispe, Richart Smith					

1. Tarea

1.1. Objetivo

- Implementar un sistema sencillo en Java aplicando los tipos de relaciones entre clases y luego dibujar el diagrama UML correspondiente.
- Comparar con otros lenguajes de programación: Go, Python, JavaScript. (Sólo uno de ellos) Sobre su implementación.

1.2. Problema propuesto:

Herencia: Clase Persona (atributos: nombre, edad). Clases hijas: Profesor y Estudiante.

Composición: Clase Curso está compuesta por un Horario (si el curso deja de existir, el horario también).

Agregación: Clase Universidad tiene una lista de Cursos (los cursos pueden existir fuera de la universidad, pero la universidad los “agrupa”).

Dependencia: Clase Reporte depende del Estudiante porque genera un reporte temporal de sus datos.

Respetar los atributos privados y crear constructores, getters y setters (JavaBeans). Implementar toString() en cada clase.

Crear un programa principal (Main).



Se crean 2 profesores y 3 estudiantes.

Se crean 2 cursos (cada uno con un horario).

Se agreguen los cursos a la universidad.

Se genera un reporte de un estudiante.

Dibujar el diagrama UML mostrando las relaciones:

	<p>Universidad Nacional de San Agustín de Arequipa Facultad de Ingeniería de Producción y Servicios Escuela Profesional de Ingeniería de Sistemas</p>	
<p>Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLD-001</p>	<p>Página: 2</p>

2. Equipos, materiales y temas utilizados

- Sistema operativo: Microsoft Windows [Versión 10.0.26100.6584]
- MiKTeX-pdfTeX 4.15 (MiKTeX 23.4) L^AT_EX.
- Strawberry Perl (requerido por MiKTeX para la ejecución de scripts auxiliares en la compilación de ciertos paquetes).
- Visual Studio Code 1.104.0 x64
- Git version 2.41.0.windows.1
- Cuenta activa en GitHub para la gestión de repositorios remotos.
- plantUML
- POO.
- Lenguaje de programación Java.
- Lenguaje de programación JavaScript.

3. URL de Repositorio Github

- URL del Repositorio GitHub para clonar o recuperar.
- <https://github.com/yhuayhuahi/Teo.git>
- URL para el laboratorio (02) en el Repositorio GitHub.
- <https://github.com/yhuayhuahi/Teo/tree/main/laboratorios/lab02>

4. Desarrollo de las actividades

4.1. Actividades

4.1.1. Diagrama de clases UML

Para resolver el problema propuesto se plantea el siguiente diagrama de clases UML:

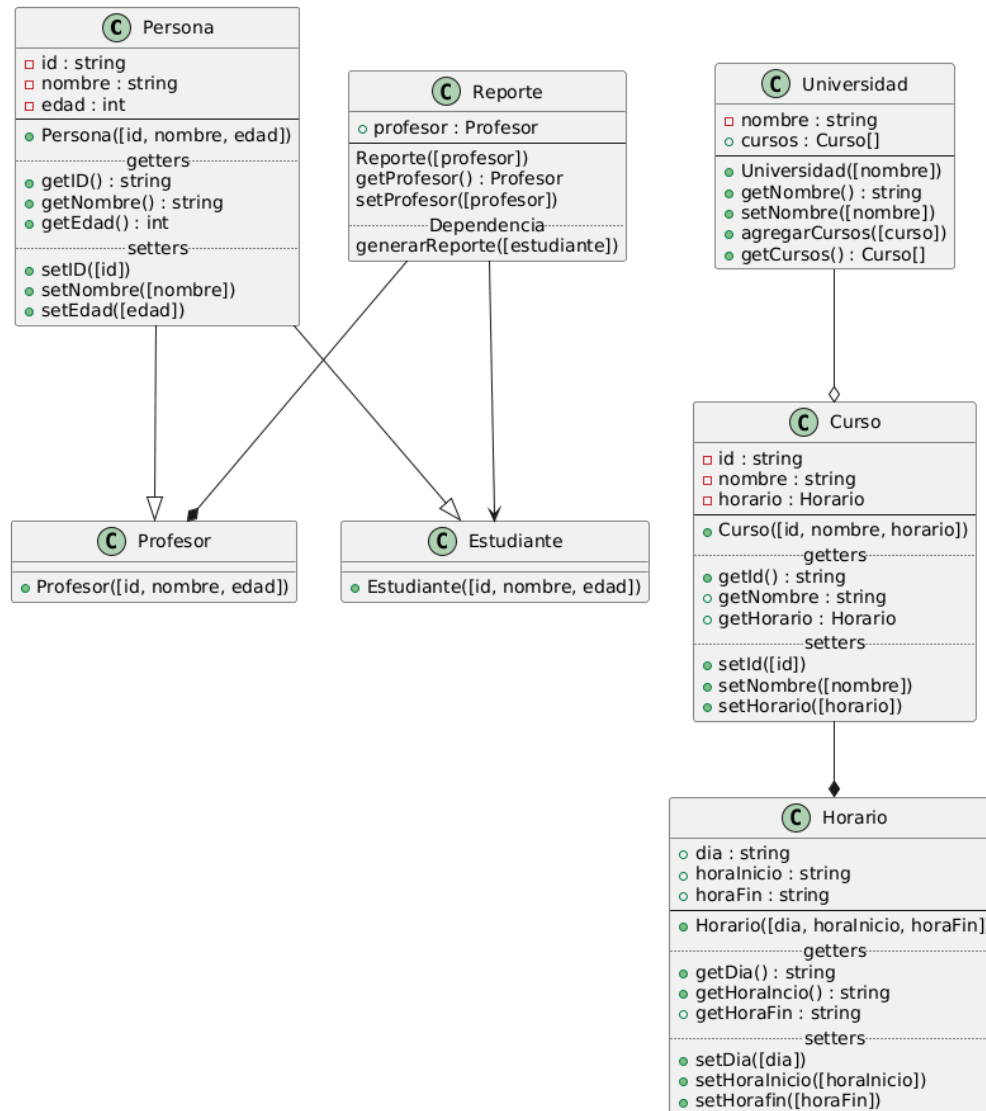


Figura 1: Diagrama UML de las clases y sus relaciones



4.1.2. Función Main en Java y JavaScript

Primero se tiene la función Main en Java, en esta se usa todas las clases creadas dentro de la carpeta **clases**, además se crean 2 profesores, 3 estudiantes, 2 cursos y se genera un reporte de un estudiante.

Listing 1: Función Main en Java

```



1 package com.tuorg.poo;
2 import com.tuorg.poo.clases.*;
3
4 public class App {
5     public static void main(String[] args) {
6         System.out.println("=== SISTEMA UNIVERSITARIO ===\n");
7
8         // 1. Crear 2 profesores
9         Profesor profesor1 = new Profesor("29386481", "Dr. Garcia", 45);
  
```

	<p>Universidad Nacional de San Agustín de Arequipa Facultad de Ingeniería de Producción y Servicios Escuela Profesional de Ingeniería de Sistemas</p>	
<p>Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLD-001</p>	<p>Página: 4</p>

```

10  Profesor profesor2 = new Profesor("29386489","Dra. Martinez", 38);
11
12  System.out.println("Profesores creados:");
13  System.out.println("- " + profesor1.getNombre() + ", " + profesor1.getEdad() + " años");
14  System.out.println("- " + profesor2.getNombre() + ", " + profesor2.getEdad() + " años\n");
15  ;
16
17  // 2. Crear 3 estudiantes
18  Estudiante estudiante1 = new Estudiante("12345678","Ana López", 20);
19  Estudiante estudiante2 = new Estudiante("87654321","Carlos Ruiz", 22);
20  Estudiante estudiante3 = new Estudiante("11223344","María González", 19);
21
22  System.out.println("Estudiantes creados:");
23  System.out.println("- " + estudiante1.getNombre() + ", " + estudiante1.getEdad() + " años");
24  System.out.println("- " + estudiante2.getNombre() + ", " + estudiante2.getEdad() + " años");
25  System.out.println("- " + estudiante3.getNombre() + ", " + estudiante3.getEdad() + " años\n");
26
27  // 3. Crear 2 cursos (cada uno con un horario)
28  Horario horario1 = new Horario("Lunes", "08:00", "10:00");
29  Curso curso1 = new Curso("CS101", "Programación I", horario1);
30
31  Horario horario2 = new Horario("Miércoles", "14:00", "16:00");
32  Curso curso2 = new Curso("MAT201", "Matemáticas Discretas", horario2);
33
34  System.out.println("Cursos creados:");
35  System.out.println("- " + curso1.getNombre() + " (" + curso1.getId() + ")");
36  System.out.println(" Horario: " + curso1.getHorario().getDia() + " de " +
37  curso1.getHorario().getHoraInicio() + " a " + curso1.getHorario().
38  getHoraFin());
39  System.out.println("- " + curso2.getNombre() + " (" + curso2.getId() + ")");
40  System.out.println(" Horario: " + curso2.getHorario().getDia() + " de " +
41  curso2.getHorario().getHoraInicio() + " a " + curso2.getHorario().
42  getHoraFin() + "\n");
43
44  // 4. Crear universidad y agregar los cursos
45  Universidad universidad = new Universidad("Universidad Tecnológica Nacional");
46  universidad.agregarCursos(curso1);
47  universidad.agregarCursos(curso2);
48
49  System.out.println("Universidad: " + universidad.getNombre());
50  System.out.println("Cursos agregados a la universidad:");
51  for (Curso curso : universidad.getCursos()) {
52      System.out.println("- " + curso.getNombre() + " (" + curso.getId() + ")");
53  }
54  System.out.println();
55
56  // 5. Generar un reporte de un estudiante
57  Reporte reporte = new Reporte(profesor1);
58  System.out.println("=== GENERACIÓN DE REPORTE ===");
59  reporte.generarReporte(estudiante1);
60
61  // Generar otro reporte para mostrar más funcionalidad
62  Reporte reporte2 = new Reporte(profesor2);
63  reporte2.generarReporte(estudiante2);

```

	<p>Universidad Nacional de San Agustín de Arequipa Facultad de Ingeniería de Producción y Servicios Escuela Profesional de Ingeniería de Sistemas</p>	
<p>Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLD-001</p>	<p>Página: 5</p>

```

61         System.out.println("\n=== FIN DEL PROGRAMA ===");
62     }
63 }
64

```



Ahora se tiene la función Main en JavaScript, en esta se usa todas las clases creadas dentro de la carpeta **clases**, además se crean 2 profesores, 3 estudiantes, 2 cursos y se genera un reporte de un estudiante.

Listing 2: Función Main en JavaScript

```

1  import { Profesor } from './clases/Profesor.js'
2  import { Estudiante } from './clases/Estudiante.js'
3  import { Curso } from './clases/Curso.js'
4  import { Horario } from './clases/Horario.js'
5  import { Universidad } from './clases/Universidad.js'
6  import { Reporte } from './clases/Reporte.js'
7
8  console.log('=== SISTEMA UNIVERSITARIO ===\n')
9
10 // 1. Crear 2 profesores
11 const profesor1 = new Profesor('29386481','Dr. García', 45)
12 const profesor2 = new Profesor('29386489','Dra. Martínez', 38)
13
14 console.log('Profesores creados:')
15 console.log('- ' + profesor1.nombre + ', ' + profesor1.edad + ' años')
16 console.log('- ' + profesor2.nombre + ', ' + profesor2.edad + ' años\n')
17
18 // 2. Crear 3 estudiantes
19 const estudiante1 = new Estudiante('12345678','Ana López', 20)
20 const estudiante2 = new Estudiante('87654321','Carlos Ruiz', 22)
21 const estudiante3 = new Estudiante('11223344','María González', 19)
22
23 console.log('Estudiantes creados:')
24 console.log('- ' + estudiante1.nombre + ', ' + estudiante1.edad + ' años')
25 console.log('- ' + estudiante2.nombre + ', ' + estudiante2.edad + ' años')
26 console.log('- ' + estudiante3.nombre + ', ' + estudiante3.edad + ' años\n')
27
28 // 3. Crear 2 cursos (cada uno con un horario)
29 const horario1 = new Horario('Lunes', '08:00', '10:00')
30 const curso1 = new Curso("CS101", "Introducción a la Programación", horario1)
31
32 const horario2 = new Horario('Miércoles', '14:00', '16:00')
33 const curso2 = new Curso("MAT201", "Matemáticas Discretas", horario2)
34
35 console.log('Cursos creados:')
36 console.log('- ' + curso1.nombre + ' (' + curso1.id + '), ' + curso1.horario.dia +
37             ' ' + curso1.horario.horaInicio + '-' + curso1.horario.horaFin)
38 console.log('- ' + curso2.nombre + ' (' + curso2.id + '), ' + curso2.horario.dia +
39             ' ' + curso2.horario.horaInicio + ' a ' + curso2.horario.horaFin + '\n')
40
41 // 4. Crear universidad y agregar los cursos
42 const universidad = new Universidad("Universidad Tecnológica Nacional")
43 universidad.agregarCursos(curso1)
44 universidad.agregarCursos(curso2)
45
46 console.log("Universidad: " + universidad.nombre)
47 console.log('Cursos agregados a la universidad:')
48 universidad.cursos.forEach(curso => {

```

	<p>Universidad Nacional de San Agustín de Arequipa Facultad de Ingeniería de Producción y Servicios Escuela Profesional de Ingeniería de Sistemas</p>	
<p>Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLD-001</p>	<p>Página: 6</p>

```



47     console.log('- ' + curso.nombre + ' (' + curso.id + ')')
48 })
49
50 console.log("\n");
51
52 // 5. Generar un reporte de un estudiante
53 const reporte = new Reporte(profesor1)
54 console.log("=== GENERACIÓN DE REPORTE ===")
55 console.log(reporte.generarReporte(estudiante1))
56
57 const reporte2 = new Reporte(profesor2)
58 console.log("\n=== GENERACIÓN DE REPORTE ===")
59 console.log(reporte2.generarReporte(estudiante2))
60
61 console.log("\n === FIN DEL PROGRAMA ===");

```

4.1.3. Pruebas de ejecución:

- Se verificó la creación de los objetos de las clases Profesor, Estudiante, Curso y Universidad.
- Se comprobó que los reportes generados mostraran la información correcta de los estudiantes.
- Se validó que los horarios de los cursos se establecieran correctamente.

Ejecución en Java:

	Universidad Nacional de San Agustín de Arequipa Facultad de Ingeniería de Producción y Servicios Escuela Profesional de Ingeniería de Sistemas	
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		
Aprobación: 2022/03/01	Código: GUIA-PRLD-001	Página: 7

```

~/.../laboratorios/lab02/l-java  0  ?main  ?2 ~2 -1  34ms  ♥
10:45:12 mvn exec:java
[INFO] --- exec:3.3.0:java (default-cli) @ poo ---
=== SISTEMA UNIVERSITARIO ===

Profesores creados:
- Dr. García, 45 años
- Dra. Martínez, 38 años

Estudiantes creados:
- Ana López, 20 años
- Carlos Ruiz, 22 años
- María González, 19 años

Cursos creados:
- Programación I (CS101)
  Horario: Lunes de 08:00 a 10:00
- Matemáticas Discretas (MAT201)
  Horario: Miércoles de 14:00 a 16:00

Universidad: Universidad Tecnológica Nacional
Cursos agregados a la universidad:
- Programación I (CS101)
- Matemáticas Discretas (MAT201)



=== GENERACIÓN DE REPORTE ===
Reporte del Profesor: Dr. García, acerca del Estudiante: Ana López
Reporte del Profesor: Dra. Martínez, acerca del Estudiante: Carlos Ruiz

=== FIN DEL PROGRAMA ===
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time:  1.136 s

```

Figura 2: Ejecución del programa en Java

Ejecución en JavaScript:

	Universidad Nacional de San Agustín de Arequipa Facultad de Ingeniería de Producción y Servicios Escuela Profesional de Ingeniería de Sistemas	
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		
Aprobación: 2022/03/01	Código: GUIA-PRLD-001	Página: 8

```

~/.../laboratorios/lab02/l-javascript  O ?main ≡ ?2
10:41:44 npm start

> l-javascript@1.0.0 start
> node index.js

=== SISTEMA UNIVERSITARIO ===

Profesores creados:
- Dr. García, 45 años
- Dra. Martínez, 38 años

Estudiantes creados:
- Ana López, 20 años
- Carlos Ruiz, 22 años
- María González, 19 años

Cursos creados:
- Introducción a la Programación (CS101), Lunes 08:00-10:00
- Matemáticas Discretas (MAT201), Miércoles 14:00 a 16:00

Universidad: Universidad Tecnológica Nacional
Cursos agregados a la universidad:
- Introducción a la Programación (CS101)
- Matemáticas Discretas (MAT201)

=== GENERACIÓN DE REPORTE ===
Reporte del Profesor: Dr. García
, acerca del Estudiante: Ana López

=== GENERACIÓN DE REPORTE ===
Reporte del Profesor: Dra. Martínez
, acerca del Estudiante: Carlos Ruiz



```

Figura 3: Ejecución del programa en JavaScript

4.2. Commits realizados

4.2.1. Primer Commit

- Este commit se realizó después de iniciar un mini-proyecto en Java usando Maven como herramienta.
- Se crearon las clases requeridas representadas en el diagrama de clases UML, elaborado en PlantUML.
- Una clase main para probar el funcionamiento de las clases implementadas.

	<p>Universidad Nacional de San Agustín de Arequipa Facultad de Ingeniería de Producción y Servicios Escuela Profesional de Ingeniería de Sistemas</p>	
<p>Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLD-001</p>	<p>Página: 9</p>

```

~/.git/Teo/laboratorios/lab02  O /main ≡ 0 ?4 ~1 -7 11ms ♥
16:48:31 git add .
~/.git/Teo/laboratorios/lab02  O /main ≡ 0 ~1 | 0 +5 ~6 -1 226ms ♥
16:49:15 git commit -m "Código completo en java, mejor organización de carpetas"
[main f0b2fcc] Código completo en java, mejor organización de carpetas
12 files changed, 152 insertions(+), 16 deletions(-)
create mode 100644 laboratorios/lab02/l-java/.gitignore
delete mode 100644 laboratorios/lab02/l-java/Main.java
create mode 100644 laboratorios/lab02/l-java/README.md
create mode 100644 laboratorios/lab02/l-java/pom.xml
create mode 100644 laboratorios/lab02/l-java/src/main/java/com/tuorg/poo/App.java
rename laboratorios/lab02/l-java/{Composicion_y_Agregacion => src/main/java/com/tuorg/poo/clases}/
rename laboratorios/lab02/l-java/{Herencia => src/main/java/com/tuorg/poo/clases}/
rename laboratorios/lab02/l-java/{Composicion_y_Agregacion => src/main/java/com/tuorg/poo/clases}/
rename laboratorios/lab02/l-java/{Herencia => src/main/java/com/tuorg/poo/clases}/
create mode 100644 laboratorios/lab02/l-java/src/main/java/com/tuorg/poo/clases/Re
rename laboratorios/lab02/l-java/{Composicion_y_Agregacion => src/main/java/com/tu
~/.git/Teo/laboratorios/lab02  O /main ↑1 0 ~1 313ms ♥
16:49:57 git push

```

Figura 4: Primer commit

4.2.2. Segundo Commit

- Este commit se hizo despues de modificar el acceso de alguna propiedades de las clases a private.
- Antes de este commit, todas las propiedades eran publicas.

```



~/.git/Teo/laboratorios/lab02  O /main ≡ 0 ~1 5.665s ♥
16:50:15 git add .
~/.git/Teo/laboratorios/lab02  O /main ≡ 0 ~1 | 0 +1 ~6 175ms ♥
16:55:46 git commit -m "Algunos ajustes en seguridad de datos, uso de private"
[main 3038cad] Algunos ajustes en seguridad de datos, uso de private
7 files changed, 31 insertions(+), 17 deletions(-)
create mode 100644 laboratorios/lab02/commit01.png
~/.git/Teo/laboratorios/lab02  O /main ↑1 0 ~1 338ms ♥
16:56:30 git push
Enumerating objects: 36, done.
Counting objects: 100% (36/36), done.
Delta compression using up to 12 threads
Compressing objects: 100% (14/14), done.
Writing objects: 100% (19/19), 76.08 KiB | 3.46 MiB/s, done.
Total 19 (delta 8), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (8/8), completed with 8 local objects.
To https://github.com/yhuayhuahi/Teo.git
f0b2fcc..3038cad main -> main

```

Figura 5: Segundo commit

4.2.3. Tercer Commit

- Este commit se hizo despues de replicar el código de Java a JavaScript.
- Se creo una carpeta llamada l-javascript y se implemento todo el código en este lenguaje.
- Se verificó que el código en JavaScript funcione correctamente.

	<p>Universidad Nacional de San Agustín de Arequipa Facultad de Ingeniería de Producción y Servicios Escuela Profesional de Ingeniería de Sistemas</p>	
<p>Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLD-001</p>	<p>Página: 10</p>

```

~/.git/laboratorios/lab02/l-javascript  o  |main  ≡  7  ?6  ~1  -2  689ms  ♥
18:05:18  git add .
warning: in the working copy of 'laboratorios/lab02/l-javascript/package.json',
it touches it
~/.git/laboratorios/lab02/l-javascript  o  |main  ≡  7  ?1  ~1  |  7  +11  -2  216m
18:06:12  git commit -m "Implementación de las clases completa en javascript"
[main 6fc8a0d] Implementación de las clases completa en javascript
13 files changed, 250 insertions(+), 11 deletions(-)
create mode 100644 laboratorios/lab02/l-javascript/.gitignore
delete mode 100644 laboratorios/lab02/l-javascript/Persona.mjs
create mode 100644 laboratorios/lab02/l-javascript/README.md
create mode 100644 laboratorios/lab02/l-javascript/clases/Curso.js
create mode 100644 laboratorios/lab02/l-javascript/clases/Estudiante.js
create mode 100644 laboratorios/lab02/l-javascript/clases/Horario.js
create mode 100644 laboratorios/lab02/l-javascript/clases/Persona.js
create mode 100644 laboratorios/lab02/l-javascript/clases/Profesor.js
create mode 100644 laboratorios/lab02/l-javascript/clases/Reporte.js
create mode 100644 laboratorios/lab02/l-javascript/clases/Universidad.js
create mode 100644 laboratorios/lab02/l-javascript/index.js
delete mode 100644 laboratorios/lab02/l-javascript/index.mjs
create mode 100644 laboratorios/lab02/l-javascript/package.json
~/.git/laboratorios/lab02/l-javascript  o  |main  ↑1  7  ?1  ~1  231ms  ♥
18:06:57  █

```

Figura 6: Tercer commit

4.2.4. Cuarto Commit

- Este commit se hizo despues de agregar el método toString() en cada clase.
- Este método permite representar el objeto como una cadena de texto

```



~/.git/laboratorios/lab02/l-java  o  |main  ≡  7  ?2  ~2  -1  5.249s  ♥
10:45:24  cd ..
~/.git/Teo/laboratorios/lab02  o  |main  ≡  7  ?2  ~16  -1  25ms  ♥
11:22:09  git add l-java/ l-javascript/
~/.git/Teo/laboratorios/lab02  o  |main  ≡  7  ?2  ~1  -1  |  7  ~15  262ms  ♥
11:22:31  git commit -m "Añadiendo los metodos to_string de cada clase"
[main 81d1a24] Añadiendo los metodos to_string de cada clase
15 files changed, 93 insertions(+), 5 deletions(-)
~/.git/Teo/laboratorios/lab02  o  |main  ↑1  7  ?2  ~1  -1  349ms  ♥
11:22:51  git push
Enumerating objects: 57, done.
Counting objects: 100% (57/57), done.
Delta compression using up to 12 threads
Compressing objects: 100% (24/24), done.
Writing objects: 100% (29/29), 3.51 KiB | 256.00 KiB/s, done.
Total 29 (delta 14), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (14/14), completed with 13 local objects.
To https://github.com/yhuayhuahi/Teo.git
 6fc8a0d..81d1a24  main -> main
~/.git/Teo/laboratorios/lab02  o  |main  ≡  7  ?2  ~1  -1  3.159s  ♥
11:23:00  █

```

Figura 7: Cuarto commit

4.3. Estructura del laboratorio



A continuación se muestra la estructura de archivos y carpetas del laboratorio realizado: Claramente los archivos de compilación de Java y otros que se pudieron generar no se subieron al repositorio.

 INGENIERIA SISTEMAS	Universidad Nacional de San Agustín de Arequipa Facultad de Ingeniería de Producción y Servicios Escuela Profesional de Ingeniería de Sistemas	 ABET
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		
Aprobación: 2022/03/01	Código: GUIA-PRLD-001	Página: 11

```

~/..../Teo/laboratorios/lab02  ↵ ↵main ≡ ↵ ?2 ~1 -1
11:57:31 tree /f
Listado de rutas de carpetas para el volumen Windows
El número de serie del volumen es A025-E3A0
C:.\
├── diagrama.puml
├── README.md
├── informe-latex
│   ├── desarrollo.tex
│   ├── main.tex
│   └── rubricas.tex
├── build
│   ├── main.aux
│   ├── main.fdb_latexmk
│   ├── main.fls
│   ├── main.log
│   ├── main.out
│   ├── main.pdf
│   └── main.synctex.gz
└── img
    ├── commit01.png
    ├── commit02.png
    ├── commit03.png
    ├── commit04.png
    ├── diagrama.png
    ├── logo_abet.png
    ├── logo_episunsa.png
    ├── prueba_ejecucion_java.png
    └── prueba_ejecucion_javascript.png



```

 INGENIERIA SISTEMAS	Universidad Nacional de San Agustín de Arequipa Facultad de Ingeniería de Producción y Servicios Escuela Profesional de Ingeniería de Sistemas	 ABET
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		
Aprobación: 2022/03/01	Código: GUIA-PRLD-001	Página: 12

```

11:57:31 ~/.../Teo/laboratorios/lab02 1/2 main 1/2 ?2 ~1 -1
tree /f
├── java
│   ├── .gitignore
│   ├── pom.xml
│   └── README.md
├── .vscode
│   └── settings.json
├── src
│   ├── main
│   │   ├── java
│   │   │   ├── com
│   │   │   │   ├── tuorg
│   │   │   │   │   ├── poo
│   │   │   │   │   │   App.java
│   │   │   │   │   └── clases
│   │   │   │   │       Curso.java
│   │   │   │   │       Estudiante.java
│   │   │   │   │       Horario.java
│   │   │   │   │       Persona.java
│   │   │   │   │       Profesor.java
│   │   │   │   │       Reporte.java
│   │   │   │   │       Universidad.java
│   └── target
│       ├── classes
│       │   ├── com
│       │   │   ├── tuorg
│       │   │   │   ├── poo
│       │   │   │   │   App.class

```

	<p>Universidad Nacional de San Agustín de Arequipa Facultad de Ingeniería de Producción y Servicios Escuela Profesional de Ingeniería de Sistemas</p>	
<p>Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLD-001</p>	<p>Página: 13</p>

```



11:57:31 ~/ ... /Teo/laboratorios/lab02  main  ?2 ~1 -1
tree /f
├── clases
│   ├── Curso.class
│   ├── Estudiante.class
│   ├── Horario.class
│   ├── Persona.class
│   ├── Profesor.class
│   ├── Reporte.class
│   └── Universidad.class
├── generated-sources
│   ├── annotations
│   ├── maven-status
│   │   ├── maven-compiler-plugin
│   │   │   ├── compile
│   │   │   │   ├── default-compile
│   │   │   │   │   ├── createdFiles.lst
│   │   │   │   │   └── inputFiles.lst
│   └── test-classes
├── l-javascript
│   ├── .gitignore
│   ├── index.js
│   ├── package.json
│   └── README.md
└── clases
    ├── Curso.js
    ├── Estudiante.js
    ├── Horario.js
    ├── Persona.js
    ├── Profesor.js
    └── Reporte.js

```

5. Cuestionario

5.1. ¿Qué diferencias puede resaltar en las implementaciones de los dos lenguajes de programación en POO?

- **Sintaxis:** Java tiene una sintaxis más estricta y verbosa, mientras que JavaScript es más flexible y concisa.
- **Tipado:** Java es un lenguaje de tipado estático. JavaScript es un lenguaje de tipado dinámico.
- **Clases y Prototipos:** Java utiliza clases para definir objetos y herencia, mientras que JavaScript utiliza prototipos. Aunque ES6 introdujo la sintaxis de clases en JavaScript, internamente sigue utilizando prototipos.
- **Encapsulamiento:** En Java, el encapsulamiento se logra mediante modificadores de acceso (private,

	Universidad Nacional de San Agustín de Arequipa Facultad de Ingeniería de Producción y Servicios Escuela Profesional de Ingeniería de Sistemas	
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		
Aprobación: 2022/03/01	Código: GUIA-PRLD-001	Página: 14

protected, public). En JavaScript, el encapsulamiento se puede lograr utilizando closures o la nueva sintaxis de campos privados (con el prefijo `private`).

- **Herencia:** Java utiliza herencia basada en clases, mientras que JavaScript utiliza herencia basada en prototipos. Esto afecta cómo se crean y extienden los objetos.

5.2. ¿En su implementación dónde se puede evidenciar la protección de datos o la seguridad utilizando la técnica de la programación orientada a objetos?



Sí, en la implementación de las clases se puede evidenciar la protección de datos a través del uso de modificadores de acceso como `private` y `protected`. Esto permite que ciertas propiedades y métodos sean inaccesibles desde fuera de la clase, lo que ayuda a mantener la integridad del objeto y a prevenir modificaciones no autorizadas. Además, el uso de métodos `getter` y `setter` proporciona un control adicional sobre cómo se accede y modifica el estado interno de un objeto.

6. Rúbricas

6.1. Entregable Informe

Cuadro 1: Tipo de Informe

Informe	
Latex	El informe está en formato PDF desde Latex, con un formato limpio (buena presentación) y fácil de leer.

	Universidad Nacional de San Agustín de Arequipa Facultad de Ingeniería de Producción y Servicios Escuela Profesional de Ingeniería de Sistemas	
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		
Aprobación: 2022/03/01	Código: GUIA-PRLD-001	Página: 15

6.2. Rúbrica para el contenido del Informe y demostración



- El alumno debe marcar o dejar en blanco en celdas de la columna **Checklist** si cumple con el ítem correspondiente.
- Si un alumno supera la fecha de entrega, su calificación será sobre la nota mínima aprobada, siempre y cuando cumpla con todos los ítems.
- El alumno debe autocalificarse en la columna **Estudiante** de acuerdo a la siguiente tabla:

Cuadro 2: Niveles de desempeño

	Nivel			
Puntos	Insatisfactorio 25 %	En Proceso 50 %	Satisfactorio 75 %	Sobresaliente 100 %
2.0	0.5	1.0	1.5	2.0
4.0	1.0	2.0	3.0	4.0

Cuadro 3: Rúbrica para contenido del Informe y demostración

Contenido y demostración		Puntos	Checklist	Estudiante	Profesor
1. GitHub	Hay enlace URL activo del directorio para el laboratorio hacia su repositorio GitHub con código fuente terminado y fácil de revisar.	2	X	2	
2. Commits	Hay capturas de pantalla de los commits más importantes con sus explicaciones detalladas. (El profesor puede preguntar para refrendar calificación).	4	X	3	
3. Código fuente	Hay porciones de código fuente importantes con numeración y explicaciones detalladas de sus funciones.	2	X	1.5	
4. Ejecución	Se incluyen ejecuciones/pruebas del código fuente explicadas gradualmente.	2	X	2	
5. Pregunta	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación).	2	X	2	
6. Fechas	Las fechas de modificación del código fuente están dentro de los plazos de fecha de entrega establecidos.	2	X	2	
7. Ortografía	El documento no muestra errores ortográficos.	2	X	2	
8. Madurez	El Informe muestra de manera general una evolución de la madurez del código fuente, explicaciones puntuales pero precisas y un acabado impecable. (El profesor puede preguntar para refrendar calificación).	4	X	3	
Total		20		17.5	

	<p>Universidad Nacional de San Agustín de Arequipa Facultad de Ingeniería de Producción y Servicios Escuela Profesional de Ingeniería de Sistemas</p>	
<p>Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLD-001</p>	<p>Página: 16</p>

7. Referencias

- [1] GeeksforGeeks, “What is Memoization? A Complete Tutorial”, GeeksforGeeks. URL: <https://www.geeksforgeeks.org/dsa/what-is-memoization-a-complete-tutorial/>
- [2] GeeksforGeeks, “Coin Change Problem in C++”, GeeksforGeeks. URL: <https://www.geeksforgeeks.org/cpp/coin-change-problem-in-cpp/>