

LABORATORIO DIRIGIDO

```
import java.text.SimpleDateFormat;
import java.time.LocalDateTime;
import java.util.ArrayList;
/*
    Autor: <colocar nombre completo>
    Fecha: <colocar la fecha actual>
*/
class Principal{
    public static void main(String[] args) throws Exception{
        //Creamos un objeto para manejo de fechas
        SimpleDateFormat sdf = new SimpleDateFormat("dd-MM-yyyy");
        //Creamos la institucion educativa
        InstitucionEducativa ie1 = new
InstitucionEducativa(1,"1092828764","Intelectuales SAC");
        //Creamos una sede
        Sede sede1 = new Sede(1,"Intelectuales SAC San Miguel","Av.
Universitaria 1801");
        //Inicializamos los programas académicos de la sede
        sede1.setProgramasAcademicos(new ArrayList<>());
        //Inicializamos las sedes de la institucion educativa
        ie1.setSedes(new ArrayList<>());
        //Asociamos la sede a la institucion educativa
        ie1.getSedes().add(sede1);
        //Creamos un curso de la sede
        Curso curso1 = new Curso("LENGUAJE DE PROGRAMACION 2", "INF282", 'P',
1500.00, 4, 5, sdf.parse("20-03-2023"), sdf.parse("08-07-2023"), DiaSemana.Martes,
LocalTime.of(8,00,00), LocalTime.of(11,00,00));
        //Creamos dos talleres de la sede
        Taller taller1 = new Taller("PYTHON PARA INVESTIGACION
CUANTITATIVA", "TAL725", 'V', 500.00, sdf.parse("28-03-2023"), LocalTime.of(13,00,00),
LocalTime.of(16,00,00));
        Taller taller2 = new Taller("INTRODUCCION A LATEX", "TAL331", 'V', 600.00,
sdf.parse("30-03-2023"), LocalTime.of(18,00,00), LocalTime.of(22,00,00));
        //Asociamos el curso y los talleres a la sede
        sede1.getProgramasAcademicos().add(curso1);
        sede1.getProgramasAcademicos().add(taller1);
        sede1.getProgramasAcademicos().add(taller2);
        //Consultamos los programas disponibles en la primera sede de la
institucion educativa
        String reporte = ie1.consultarProgramas(0);
        //Imprimimos el reporte
        System.out.print(reporte);
    }
}
```

```
InstitucionEducativa.java x
1  import java.util.ArrayList;
2
3  class InstitucionEducativa{
4      private int idInstitucionEducativa;
5      private String RUC;
6      private String nombre;
7      ArrayList<Sede> sedes;
8
9      public InstitucionEducativa(int idInstitucionEducativa, String RUC, String nombre){
10         this.idInstitucionEducativa = idInstitucionEducativa;
11         this.RUC = RUC;
12         this.nombre = nombre;
13     }
14
15     public String consultarProgramasDeSede(int indiceSede){
16         String cadena = "Programas disponibles para "+sedes.get(indiceSede).getNombre() + "\n";
17         cadena = cadena + sedes.get(indiceSede).consultarProgramas();
18         return cadena;
19     }
20
21     public ArrayList<Sede> getSedes(){
22         return sedes;
23     }
24
25     public void setSedes(ArrayList<Sede>sedes){
26         this.sedes = sedes;
27     }
28 }
```

```
InstitucionEducativa.java x Sede.java x
1  import java.util.ArrayList;
2
3  class Sede{
4      private int idSede;
5      private String nombre;
6      private String direccion;
7      private ArrayList<ProgramaAcademico> programasAcademicos;
8      public Sede(int idSede, String nombre, String direccion){
9         this.idSede = idSede;
10         this.nombre = nombre;
11         this.direccion = direccion;
12     }
13
14     public String getNombre(){
15         return nombre;
16     }
17
18     public String consultarProgramas(){
19         String cadena = " ";
20         for(ProgramaAcademico pa : programasAcademicos){
21             cadena = cadena + pa.consultarDatos() + "\n";
22         }
23         return cadena;
24     }
25
26     public ArrayList<ProgramaAcademico> getProgramasAcademicos(){
27         return programasAcademicos;
28     }
29
30     public void setProgramasAcademicos(ArrayList<ProgramaAcademico> programasAcademicos){
31         this.programasAcademicos = programasAcademicos;
32     }
33
34 }
```

```
InstitucionEducativa.java x Sede.java x ProgramaAcademico.java x
1 //se coloca abstract porque hablamos de una clase padre
2 //se coloca implements porque va a implementar una interface
3 abstract class ProgramaAcademico implements IConsultable{
4     //Colocamos static porque queremos crear un indice correlativo
5     private static int indice = 1;
6     private int idProgramaAcademico;
7     private String nombre;
8     private String clave;
9     private char modalidad;
10    private double precio;
11
12    public ProgramaAcademico(String nombre, String clave,
13                             char modalidad, double precio){
14        this.idProgramaAcademico = indice;
15        this.nombre = nombre;
16        this.clave = clave;
17        this.modalidad = modalidad;
18        this.precio = precio;
19        indice++;
20    }
21
22    public String getClave(){
23        return clave;
24    }
25
26    public String getNombre(){
27        return nombre;
28    }
29
30    public double getPrecio(){
31        return precio;
32    }
33 }
```

```
InstitucionEducativa.java x Sede.java x ProgramaAcademico.java x Curso.java x
1 import java.time.LocalDateTime;
2 import java.util.Date;
3 //Colocamos extends porque es una clase hija de ProgramaAcademico
4 class Curso extends ProgramaAcademico{
5     private int cantHorasClasePorSemana;
6     private float cantCreditos;
7     private Date fechaInicio;
8     private Date fechaFin;
9     private DiaSemana diaDictado;
10    private LocalDateTime horaInicioClases;
11    private LocalDateTime horaFinClases;
12
13    //Para el constructor de la clase hija se deben agregar como parametros los atributos de la clase padre
14    public Curso(String nombre, String clave, char modalidad, double precio,
15                int cantHorasClasePorSemana, float cantCreditos, Date fechaInicio,
16                Date fechaFin, DiaSemana diaDictado, LocalDateTime horaInicioClases, LocalDateTime horaFinClases){
17        //el super realiza la construccion de los atributos de la clase padre
18        super(nombre,clave,modalidad,precio);
19        this.cantHorasClasePorSemana = cantHorasClasePorSemana;
20        this.cantCreditos = cantCreditos;
21        this.fechaInicio = fechaInicio;
22        this.fechaFin = fechaFin;
23        this.diaDictado = diaDictado;
24        this.horaInicioClases = horaInicioClases;
25        this.horaFinClases = horaFinClases;
26    }
27
28    public String consultarDatos(){
29        return "CURSO: "+ getClave() + " - "+getNombre()+"- S/. "+ getPrecio()+" - CRED: "+ cantCreditos;
30    }
31 }
```

```

1  import java.util.Date;
2  import java.time.LocalDateTime;
3  class Taller extends ProgramaAcademico{
4      private Date fechaRealizacion;
5      private LocalDateTime horaInicioTaller;
6      private LocalDateTime horaFinTaller;
7
8      public Taller(String nombre, String clave, char modalidad, double precio,
9                  Date fechaRealizacion, LocalDateTime horaInicioTaller, LocalDateTime horaFinTaller){
10         super(nombre,clave,modalidad,precio);
11         this.fechaRealizacion = fechaRealizacion;
12         this.horaInicioTaller = horaInicioTaller;
13         this.horaFinTaller = horaFinTaller;
14     }
15
16     public String consultarDatos(){
17         return "CURSO: " + getClave() + " - "+getNombre()+"- S/. "+ getPrecio()+" - FECHA: "
18             + fechaRealizacion;
19     }
20 }

```

```

1  interface IConsultable{
2      String consultarDatos();
3  }

```

```

1  enum DiaSemana{
2      Lunes, Martes, Miercoles, Jueves, Viernes, Sabado
3  }

```

