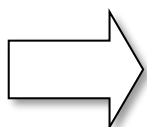
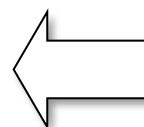


FORMACIÓN PROFESIONAL DUAL



INFORME DE PRÁCTICA



CÓDIGO N° 89001677



DIRECCIÓN ZONAL

FORMACIÓN PROFESIONAL DUAL

CFP/UCP/ESCUELA: Senati

ESTUDIANTE: **GANDY WILLIAM HUMIRI QUISPE** _____

ID: **1546329@senati.pe** _____

CARRERA: **Ingeniería de Software con Inteligencia Artificial**

INSTRUCTOR: _____

SEMESTRE: **GERMAN WILIAN LEON MARIN** DEL: 02/12/2024__ AL: 07/12/2024



INSTRUCCIONES PARA EL USO DEL INFORME DE PRÁCTICA DE TRABAJO SEMANAL

1. PRESENTACIÓN.

El Informe de Práctica de trabajo semanal es un documento de control, en el cual el estudiante, registra diariamente, durante la semana, las tareas, operaciones que ejecuta en su formación práctica en SENATI y en la Empresa.

2. INSTRUCCIONES PARA EL USO DEL INFORME DE PRÁCTICA.

- 2.1 En el cuadro de rotaciones, el estudiante, registrará el nombre de las áreas o secciones por las cuales rota durante su formación práctica, precisando la fecha de inicio y término.
- 2.2 Con base al PEA proporcionado por el instructor, el estudiante transcribe el PEA en el informe de práctica. El estudiante irá registrando y controlando su avance, marcando en la columna que corresponda.
- 2.3 En la hoja de informe semanal, el estudiante registrará diariamente los trabajos que ejecuta, indicando el tiempo correspondiente. El día de asistencia al centro para las sesiones de tecnología, registrará los contenidos que desarrolla. Al término de la semana totalizará las horas.
De las tareas ejecutadas durante la semana, el estudiante seleccionará la más significativa y hará una descripción del proceso de ejecución con esquemas y dibujos correspondientes que aclaren dicho proceso.
- 2.4 Semanalmente, el estudiante registrará su asistencia, en los casilleros correspondientes.
- 2.5 Semanalmente, el Monitor revisará, anotará las observaciones y recomendaciones que considere; el Instructor revisará y calificará el Informe de Práctica haciendo las observaciones y recomendaciones que considere convenientes, en los aspectos relacionados a la elaboración de un Informe Técnico (términos técnicos, dibujo técnico, descripción de la tarea y su procedimiento, normas técnicas, seguridad, etc.)
- 2.6 Si el PEA tiene menos operaciones (151) de las indicadas en el presente formato, puede eliminar alguna página. Asimismo, para el informe de las semanas siguientes, debe agregar las semanas que corresponda.
- 2.7 Escala de calificación:

| CUANTITATIVA | CUALITATIVA | CONDICIÓN |
|---------------------|--------------------|------------------|
| 16,8 – 20,0 | Excelente | Aprobado |
| 13,7 – 16,7 | Bueno | |
| 10,5 – 13,6 | Aceptable | |
| 00 – 10,4 | Deficiente | Desaprobado |

INFORME SEMANAL

4 SEMESTRE SEMANA N°4 DEL 02/12/2024 AL07/12DEL 2024

| DÍA | TRABAJOS EFECTUADOS | HORAS |
|-----------|--|-------|
| LUNES | Recordando java, hicimos algunos ejercicios para recodar 1 No Supervisado (Clustering con distancias simples) 2 Usa Algoritmos Post-order Trasversal 3 Define y crea la estructura Árbol 4 Ejercicios de pre-orden, in-orden, pos-orden | 5:15 |
| MARTES | | |
| MIÉRCOLES | | |
| JUEVES | | |
| VIERNES | | |

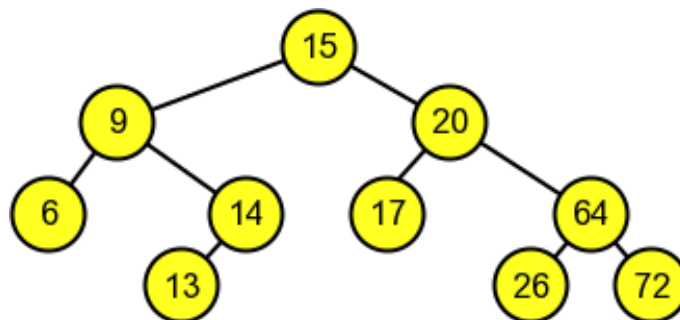
| | | |
|--------|-------|--|
| SÁBADO | | |
| | TOTAL | |

Tarea más significativa: descripción de árbol binarios y nodos

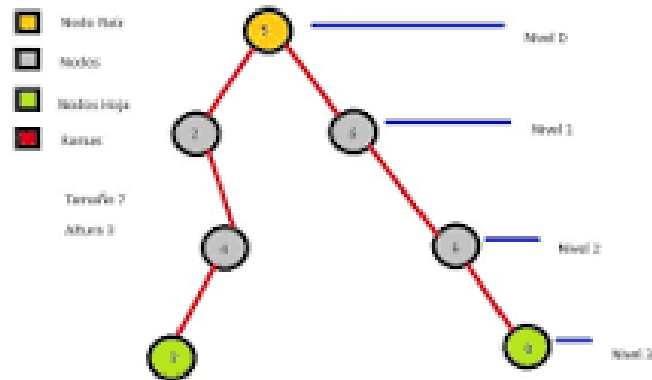
Descripción del proceso:

Tipos de Árboles

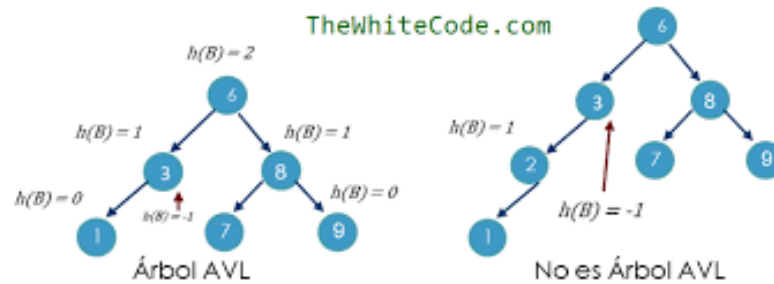
1. **Arbol Binario:** en este caso cada nodo tiene como máximo dos hijos y es llamado hijo izquierdo y hijo derecho



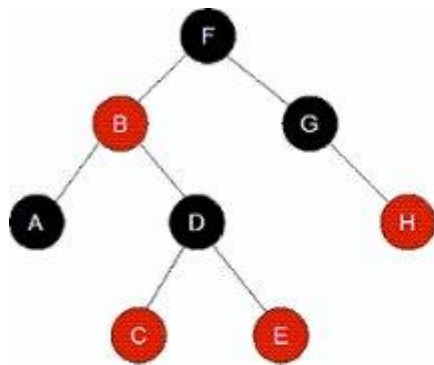
2. **Arbol Binario de Búsqueda** el árbol binario en el que para cada nodo y todos los valores en el subárbol izquierdo son menores y todos los valores en el subárbol derecho son mayores



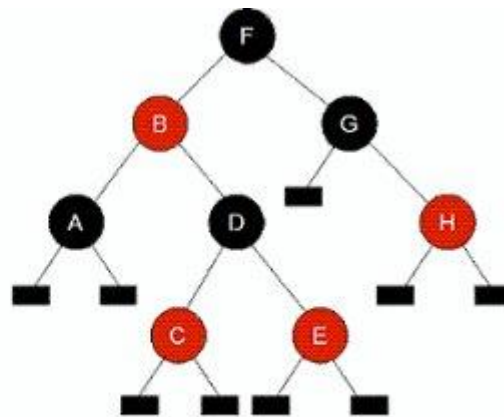
3. **Arbol AVL** Un árbol binario de búsqueda auto balanceado donde la diferencia de altura entre los subárboles izquierdo y derecho de cualquier nodo es como maximo uno



4. **Árbol Rojo-Negro:** Un tipo de árbol binario de búsqueda auto balanceado donde cada nodo tiene un color (rojo o negro) y se mantienen ciertas propiedades para asegurar el balanceo



Árbol Rojo-Negro



Árbol Rojo-Negro
(con sus nodos externos)

Nodos: Un nodo es la unidad básica de un árbol. Cada nodo contiene un valor y referencias a sus nodos hijos.

Hojas (Leaves)

- 1 Las hojas son nodos que no tienen hijos
- 2 Representan los nodos terminales del árbol. No tienen descendientes

Recorridos de Árboles

1. **Pre-order (Preorden)**
 - 1 Visita el nodo raíz, luego el subárbol izquierdo y finalmente el subárbol derecho.
 - 2 Útil para copiar el árbol o evaluar expresiones.
2. **In-order (Inorden)**

- 1 Visita el subárbol izquierdo, luego el nodo raíz y finalmente el subárbol derecho.
- 2 Produce una secuencia ordenada de valores en un árbol binario de búsqueda.
3. **Post-order (Postorden)**
 - 1 Visita el subárbol izquierdo, luego el subárbol derecho y finalmente el nodo raíz.
 - 2 Útil para eliminar el árbol o evaluar expresiones en notación postfija.

Operaciones en Árboles

1. Inserción

Función: Añadir un nuevo nodo al árbol. En un BST, se coloca en la posición correcta para mantener el orden.

2. Búsqueda

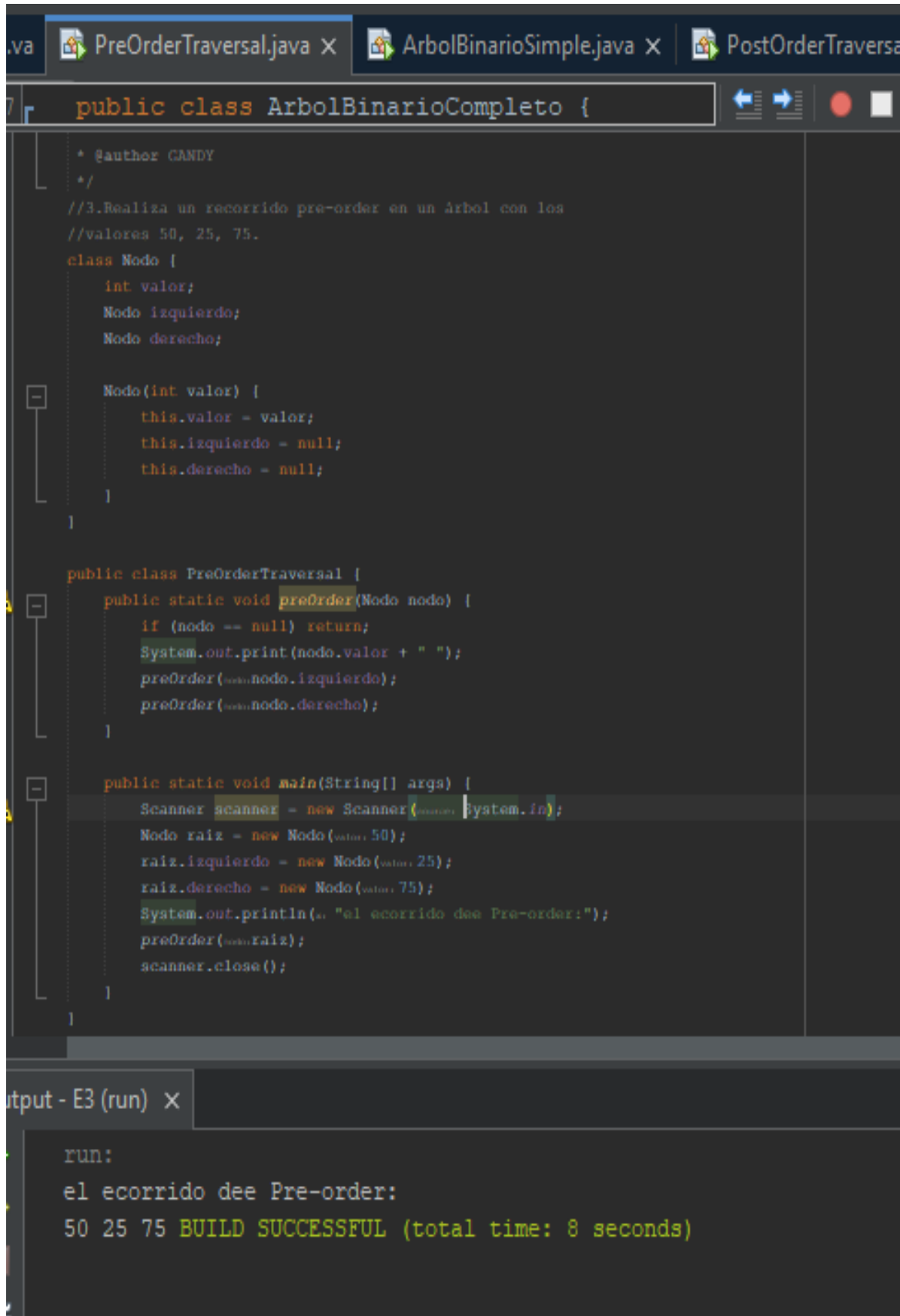
Función: Encontrar un nodo en el árbol. En un BST, se puede hacer de manera eficiente comparando valores.

3. Eliminación

Función: Eliminar un nodo del árbol. Puede ser más complejo, especialmente si el nodo tiene dos hijos.

Ejemplode:

Aquí tenemos un ejemplo de Pre-order, este ejercicio pertenece al tema de recorridos de arboles. El recorrido preorder visita primero el nodo raíz, luego el subárbol izquierdo y finalmente el subárbol derecho



```
PreOrderTraversal.java x ArbolBinarioSimple.java x PostOrderTraversa
7 public class ArbolBinarioCompleto {
    * @author CANDY
    */
    //3.Realiza un recorrido pre-order en un Árbol con los
    //valores 50, 25, 75.
    class Nodo {
        int valor;
        Nodo izquierdo;
        Nodo derecho;

        Nodo(int valor) {
            this.valor = valor;
            this.izquierdo = null;
            this.derecho = null;
        }
    }

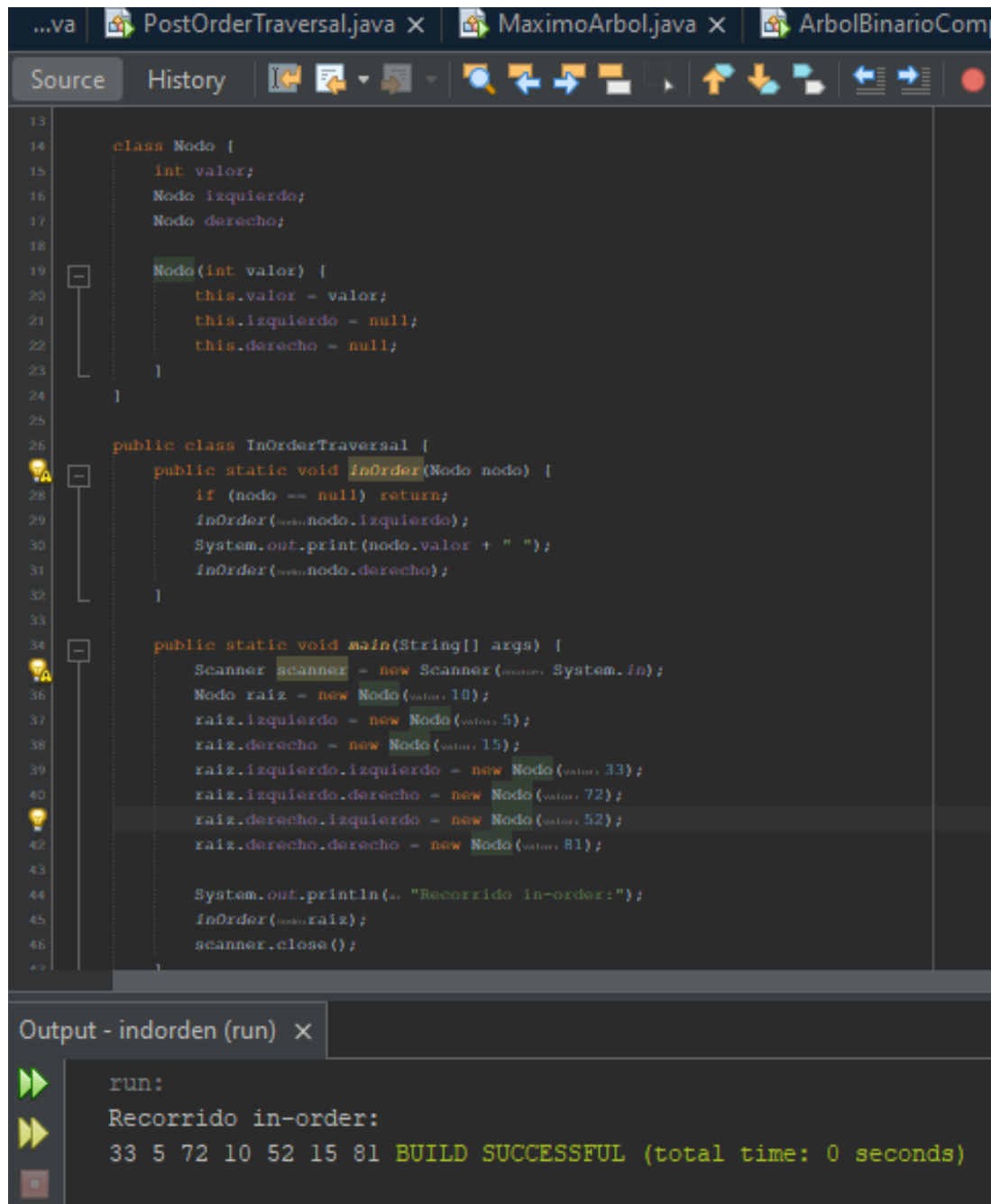
    public class PreOrderTraversal {
        public static void preOrder(Nodo nodo) {
            if (nodo == null) return;
            System.out.print(nodo.valor + " ");
            preOrder(nodo.izquierdo);
            preOrder(nodo.derecho);
        }

        public static void main(String[] args) {
            Scanner scanner = new Scanner(System.in);
            Nodo raiz = new Nodo(50);
            raiz.izquierdo = new Nodo(25);
            raiz.derecho = new Nodo(75);
            System.out.println("el ecorrido dee Pre-order:");
            preOrder(raiz);
            scanner.close();
        }
    }
}
```

```
Output - E3 (run) x
run:
el ecorrido dee Pre-order:
50 25 75 BUILD SUCCESSFUL (total time: 8 seconds)
```

Ejercicio: in-order

aquí tenemos un ejemplo y es Implementar un árbol binario con los valores 10, 5, 15, 33, 72, 52, 81 y realizar un recorrido inorder para imprimir los valores en orden ascendente



```
13
14 class Nodo {
15     int valor;
16     Nodo izquierdo;
17     Nodo derecho;
18
19     Nodo(int valor) {
20         this.valor = valor;
21         this.izquierdo = null;
22         this.derecho = null;
23     }
24 }
25
26 public class InOrderTraversal {
27     public static void inOrder(Nodo nodo) {
28         if (nodo == null) return;
29         inOrder(nodo.izquierdo);
30         System.out.print(nodo.valor + " ");
31         inOrder(nodo.derecho);
32     }
33
34     public static void main(String[] args) {
35         Scanner scanner = new Scanner(System.in);
36         Nodo raiz = new Nodo(10);
37         raiz.izquierdo = new Nodo(5);
38         raiz.derecho = new Nodo(15);
39         raiz.izquierdo.izquierdo = new Nodo(33);
40         raiz.izquierdo.derecho = new Nodo(72);
41         raiz.derecho.izquierdo = new Nodo(52);
42         raiz.derecho.derecho = new Nodo(81);
43
44         System.out.println("Recorrido in-order:");
45         inOrder(raiz);
46         scanner.close();
47     }
48 }
```

Output - indorden (run) X

```
run:
Recorrido in-order:
33 5 72 10 52 15 81 BUILD SUCCESSFUL (total time: 0 seconds)
```

Ejercicio de post-orden

Aui tenemos un ejemplo y este ejercicio pertenece al tema de recorridos de arboles. El recorrido post order visita primero el subárbol izquierdo, luego el subárbol derecho y finalmente el nodo raíz

```
27 public class ArbolBinarioCompleto {
9     * @author GANDY
10    */
11    //5.Has un recorrido post-order en un árbol
12    //con los valores 10, 5, 15.
13    class Nodo {
14        int valor;
15        Nodo izquierdo;
16        Nodo derecho;
17        Nodo(int valor) {
18            this.valor = valor;
19            this.izquierdo = null;
20            this.derecho = null;
21        }
22    }
23    public class PostOrderTraversal {
24        public static void postOrder(Nodo nodo) {
25            if (nodo == null) return;
26            postOrder(nodo.izquierdo);
27            postOrder(nodo.derecho);
28            System.out.print(nodo.valor + " ");
29        }
30        public static void main(String[] args) {
31            Scanner scanner = new Scanner(System.in);
32            Nodo raiz = new Nodo(valor: 10);
33            raiz.izquierdo = new Nodo(valor: 5);
34            raiz.derecho = new Nodo(valor: 15);
35            System.out.println("Recorrido Post-order:");
36            postOrder(nodo: raiz);
37            scanner.close();
38        }
39    }
40 }
```

Output - E5 (run) ×

```
run:
Recorrido Post-order:
5 15 10 BUILD SUCCESSFUL (total time: 0 seconds)
```

| |
|--|
| El Instructor que revisa los informes de Prácticas realizará la retroalimentación directamente en la plataforma LMS Blackboard |
| |
| |
| |

| AUTOCONTROL DE ASISTENCIA POR EL ESTUDIANTE | | | | | | | | | | | |
|---|---|----------------------|---|-----------|---|--------------|---|-------------------|---|----------------------|---|
| LUNES | | MARTES | | MIÉRCOLES | | JUEVES | | VIERNES | | SÁBADO | |
| M | T | M | T | M | T | M | T | M | T | M | T |
| ASISTENCIA A SENATI | | <input type="text"/> | | | | INASISTENCIA | | INJUSTIFICADAS: I | | <input type="text"/> | |
| | | | | | | | | JUSTIFICADAS : FJ | | <input type="text"/> | |



**PROPIEDAD INTELECTUAL DEL SENATI. PROHIBIDA SU
REPRODUCCIÓN Y VENTA SIN LA AUTORIZACIÓN
CORRESPONDIENTE**