

## LISTA DE COTEJO PARA REPORTE DE PRÁCTICA

INSTITUTO TECNOLÓGICO SUPERIOR DE SAN ANDRÉS TUXTLA		NOMBRE DEL CURSO: Lenguajes y Autómatas II		
NOMBRE DEL DOCENTE: MTI. Víctor Manuel Chontal Amador		FIRMA DEL DOCENTE		
DATOS GENERALES DEL PROCESO DE EVALUACIÓN				
NOMBRE DEL ALUMNO: RAMIREZ FIGUEROA JARED		No. DE CONTROL: 191U0188		
PRODUCTO: Reporte de ejercicios	FECHA:26-Abr-2023	PERIODO ESCOLAR: Febrero – Julio 2023		
INSTRUCCIONES DE APLICACIÓN				
Revisar las actividades que se solicitan y marque con una X en los apartados "SI" cuando la evidencia se cumple; en caso contrario marque "NO". En la columna "OBSERVACIONES" escriba indicaciones que puedan ayudar al alumno a saber cuáles son las condiciones no cumplidas, si fuese necesario.				
VALOR DEL REACTIVO	CARACTERÍSTICA A CUMPLIR (REACTIVO)	CUMPLE		OBSERVACIONES
		SI	NO	
12%	<b>Desarrollo:</b> Sigue una metodología y sustenta todos los pasos que se realizaron..	SI		
42%	<b>Resultados:</b> Cumplió totalmente con el objetivo esperado, el resultado es correcto.	SI		
6%	<b>Responsabilidad:</b> Entregó la práctica en la fecha y hora señalada.	SI		
60%	<b>CALIFICACIÓN</b>	100		

Hotmail, Noticias, Deportes y Eni x Sistema de Gestión Integral 3 x Evidencias Lenguajes y Automat: x MyWebSQL x +

mail.google.com/mail/u/0/#inbox/FMfcgzGsnLNMSVLQRfxpCxCKRGFbxRpp?projector=1&messagePartId=0.1



**INSTITUTO TECNOLÓGICO  
SUPERIOR DE SAN ANDRÉS  
TUXTLA**

---

**INGENIERIA EN SISTEMAS  
COMPUTACIONALES**

**Lenguajes y Automatas II**

Docente: Víctor Manuel Chontal Amador

Integrantes:  
Jared Ramírez Figueroa  
Johana Jacquelin Osto Mazaba  
Crescencio Rascón Hernández

Hotmail, Noticias, Deportes y Eni x Sistema de Gestión Integral 3 x Evidencias Lenguajes y Automat: x MyWebSQL x +

mail.google.com/mail/u/0/#inbox/FMfcgzGsnLNMSVLQRfxpCxCKRGFbxRpp?projector=1&messagePartId=0.1

**Practica N°1**

Realizar clase en java que lea una expresión y la convierta a una notación prefija

**Desarrollo**

```
import java.util.Stack;
public class Practica {
    public static String toPrefix(String expression) {
        Stack<Character> operatorStack = new Stack<>();
        Stack<String> operandStack = new Stack<>();
        for (int i = expression.length() - 1; i >= 0; i--) {
            char c = expression.charAt(i);
            if (Character.isDigit(c)) {
                operandStack.push(Character.toString(c));
            } else if (isOperator(c)) {
                while (!operatorStack.isEmpty() && precedence(c) <
                    precedence(operatorStack.peek())) {
                    String op1 = operandStack.pop();
                    String op2 = operandStack.pop();
                    char operator = operatorStack.pop();
                    String result = operator + op2 + op1;
                    operandStack.push(result);
                }
            }
        }
    }
}
```