

Práctica 14

| DOCENTE | CARRERA | CURSO |
|-------------------------------------|---|--|
| MSc. Vicente Enrique Machaca Arceda | Escuela Profesional de Ingeniería de Software | Fundamentos de Lenguajes de Programación |

| PRÁCTICA | TEMA | DURACIÓN |
|----------|--------------------------------------|----------|
| 14 | Programación funcional - Memoization | 2 horas |

1. Competencias del curso

- Conocer el desarrollo histórico de los lenguajes de programación y los paradigmas de programación.
- Comprender el papel de los diferentes mecanismos de abstracción en la creación de facilidades definidas por el usuario así como los beneficios de los lenguajes intermedios en el proceso de compilación.

2. Competencias de la práctica

- Programar tareas básicas en lenguaje ensamblador.

3. Equipos y materiales

- Latex
- Conexión a internet
- IDE de desarrollo

4. Entregables

- Se debe elaborar un informe en **Latex** donde se responda a cada ejercicio de la Sección 6.
- En el informe se debe agregar un enlace al repositorio Github donde esta el código.
- En el informe se debe agregar el código fuente así como capturas de pantalla de la ejecución y resultados del mismo.
- Por cada 5 minutos de retraso, el alumno tendrá un punto menos.

5. Datos del estudiante

- Grupo: 03
- Integrante:
 - Jonathan Aguirre Soto

6. Ejercicios

1. El número de Euler, $e \simeq 2,71828$, puede ser representado como la siguiente suma infinita :

$$e = \frac{1}{0!} + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \dots \quad (1)$$

Desarrolle un programa que entregue un valor aproximado de e , calculando esta suma hasta que la diferencia entre dos sumandos consecutivos sea menor que 0,0001. Utilice recursividad y memoization.

```
<html>
<body>
<script>

function factorial (num){

    var suma = 1;

    if(num === 0)
        return 1;

    for (i = 1; i <= num; i++) {
        suma = suma * i;
    }
    return suma;
}

var Euler = (function(){
    var contador = 0;
    var memo = {};

    function funcion1(num){

        var value;

        if (num in memo){
            value = memo[num];
        }

        else{

            contador = contador + 1;
            let x = (1/factorial(num)) - (1/factorial(num+1));

            if (x <= 0.0001){
                value = num;
                console.log("La diferencia es menor o igual a 0.0001 en la paso
                    de recursivo numero: ", contador);
            }

            else
                value = (1/factorial(num) + 1/factorial(funcion1(num+1)));
        }
    }
})

</script>
</body>
</html>
```

```
        memo[num] = value;
    }

    return value;
}

return funcion1;
})();

console.log(Euler(3));
console.log(Euler(5));

</script>
</body>
</html>
```

- Ejecución del programa.

| | |
|---|---|
| La diferencia es menor o igual a 0.0001 en la paso de recursivo numero: 6 | Ejercicio 1 Practica 14.html:38 |
| 1.1666666666666667 | Ejercicio 1 Practica 14.html:54 |
| 1.0083333333333333 | Ejercicio 1 Practica 14.html:55 |

2. Una manera de calcular el seno y coseno es con la siguiente fórmula, la cantidad de operandos define la precisión del valor de seno/coseno. Desarrolle un programa que calcule el seno/coseno aproximado, utilice memoization y recursividad.

$$\text{sen}(x) = \frac{x^1}{1!} + \frac{x^3}{3!} + \frac{x^5}{5!} + \frac{x^7}{7!} + \dots \quad (2)$$

$$\text{cos}(x) = \frac{x^0}{0!} + \frac{x^2}{2!} + \frac{x^4}{4!} + \frac{x^6}{6!} + \dots \quad (3)$$

3. Desarrolle una función que como parámetro reciba una función recursiva (sin memoization) y retorne dicha función recursiva pero ahora con memoization.

```
<html>
<body>
<script>

function factorial_recursive (num){

    if (num === 0){
        return 1;
    }

    return num * factorial_recursive(num - 1);
}

function factorial_memo(num1){

    var memo = {};

    function factorial(x){

        if (!memo[x]){

            memo[x] = x * num1(x - 1);

        }

        else {

            return memo [x];
        }

        return memo [x]
    }

    return factorial;
}

console.log("Usando la funcin sin Memo: ", factorial_recursive(10));
console.log("Usando la funcin con Memo: ",
    factorial_memo(factorial_recursive)(10));

</script>
```

```
</body>  
</html>
```

- Ejecución del programa.

```
Usando la función sin Memo: 3628800      Ejercicio 3 Practica 14.html:36  
Usando la función con Memo: 3628800     Ejercicio 3 Practica 14.html:37  
>
```

4. Enlace de Github:

<https://github.com/Jona2010/Fundamentos-de-Lenguaje-de-la-Programaci-n>

7. Rúbricas

| Rúbrica | Cumple | Cumple con obs. | No cumple |
|--|--------|-----------------|-----------|
| Informe: El informe debe estar en Latex, con un formato limpio y facil de leer. | 4 | 2 | 0 |
| Implementación: Implementa la funcionalidad correcta de cada ejercicio. | 16 | 8 | 0 |
| Errores ortográficos: Por cada error ortográfico, se le descontara 1 punto. | - | - | - |