

PSP0 Ejercicio 1A

1. Resumen del plan del proyecto nivel PSP0

PSP0 Project Plan Summary - Program 1A

Student	<u>1-Rubén Ignacio Couoh Ku</u>	Date	<u>Tuesday, January 10, 2017</u>
Program	<u>Correlation</u>	Program#	<u>1A</u>
Instructor	<u>Carlos Mojica</u>	Language	<u>Node v6.9.4 LTS</u>

	Plan	Actual	To Date	To Date%
Time in Phase (min.)				
Planning		10	10	7.3
Design		1	1	0.7
Code		97	97	70.8
Compile		16	16	11.7
Test		9	9	6.6
Postmortem		4	4	2.9
Total	180.0	137	137	100.0

Defects Injected

Planning	0	0	0.0
Design	0	0	0.0
Code	7	7	100.0
Compile	0	0	0.0
Test	0	0	0.0
Total Development	7	7	100.0

Defects Removed

Planning	0	0	0.0
Design	0	0	0.0
Code	0	0	0.0
Compile	6	6	85.7
Test	1	1	14.3
Total Development	7	7	100.0
After Development	0	0	

2. Forma de registro de tiempos.

Project	Phase	Date	Start	Int.	Stop	Delta	Comments
1	PLAN	01/10/17	16:14:00		16:24:00	10.0	Revisión de requerimientos y solución de dudas.
1	DLD	01/11/17	14:27:00		14:28:00	1.0	División del problema en clases.
1	CODE	01/11/17	14:33:00	1.0	15:15:00	41.0	Codificación de la lista.
1	CODE	01/11/17	16:21:00		16:50:00	29.0	Conificación de la entrada de datos.
1	CODE	01/12/17	12:08:00	1.0	12:36:00	27.0	Codificación de las formulas matemáticas.
1	COMPILE	01/12/17	12:54:00		13:10:00	16.0	Compilación de todo el programa
1	TEST	01/12/17	14:18:00		14:27:00	9.0	Ejecución de las pruebas con los tres conjuntos de datos suministrados en las diapositivas.
1	PM	01/12/17	15:01:00		15:05:00	4.0	Validación de los registros de tiempos y defectos contra el resumen de plan de proyecto.

3. Forma de registro de defectos.

Project	Date	Num	Type	Injected	Removed	FixTime	Fix Ref.	Description
1	01/12/17	1	20	CODE	COMPILE	2.0		Eliminación de un identificador no cualificado. [delete firstNode]
1	01/12/17	2	20	CODE	COMPILE	1.0		Acceso a un método estático con ":" [Rmath::avg(list);]
1	01/12/17	3	40	CODE	COMPILE	1.0		Acceso a una variable no definida [rl.close();]
1	01/12/17	4	40	CODE	COMPILE	4.0		Flujo de datos cerrado anticipadamente. [io.close();]
1	01/12/17	5	40	CODE	COMPILE	1.0		Acceso a la variable List en lugar de list
1	01/12/17	6	40	CODE	COMPILE	1.0		No se regreso el valor de la media en la función avg() de la clase RMath
1	01/12/17	7	80	CODE	TEST	2.0		No se tomo en cuenta el ultimo nodo de la lista en ciclo while que calcula la media y la desviación estándar [while (node != list.end())]

4. Código fuente del programa.

```

1  /**
2   *  Rubén Ignacio Couoh Ku.
3   */
4
5  class Node {
6      constructor(value, next)
7      {
8          this.value = value;
9          this.next = next;
10     }
11 }
12
13 module.exports = Node;

```

```
1  /**
2   *   Rubén Ignacio Couch Ku.
3   **/
4
5   var Node = require('./Node.js');
6
7   class List {
8     constructor()
9     {
10      this.head = null;
11      this.tail = null;
12    }
13
14    begin () {
15      let first = this.head;
16      return first;
17    }
18
19    end () {
20      let last = this.tail;
21      return last;
22    }
23
24    get isEmpty()
25    {
26      return this.head == null;
27    }
28
29    push(value)
30    {
31      let newNode = new Node(value, null);
32
33      if (this.isEmpty) {
34        this.head = newNode;
35        this.tail = this.head;
36      } else {
37        this.tail.next = newNode;
38        this.tail = this.tail.next;
39      }
40    }
41  }
```

```
42 pop()
43 {
44     let value = null;
45     if (!this.isEmpty) {
46         let newFirst = this.head.next;
47         value = this.head.value;
48
49         if (this.head === this.tail) {
50             delete this.head;
51             this.head = this.tail = null;
52         } else {
53             delete this.head;
54             this.head = this.newFirst;
55         }
56     }
57     return value;
58 }
59
60 peek()
61 {
62     let value = null;
63     if (!this.isEmpty) {
64         value = this.head.value;
65     }
66
67     return value;
68 }
69
70
71 get size()
72 {
73     let count = 0;
74     if (!this.isEmpty) {
75         let tmp = this.head;
76
77         while (tmp !== null) {
78             count++;
79             tmp = tmp.next;
80         }
81     }
82
83     return count;
84 }
85 }
86
87 module.exports = List;
```

```

1  /**
2   *   Rubén Ignacio Couch Ku.
3   **/
4
5  class RMath {
6      constructor() { }
7
8      static avg (list)
9      {
10         let node = list.begin(),
11             mean = 0,
12             sum = 0;
13
14         if (!list.isEmpty) {
15             while ( node !== null) {
16                 sum += node.value;
17                 node = node.next;
18             }
19         }
20         mean = sum / list.size;
21         return mean;
22     }
23
24     static stdDev(list)
25     {
26         let mean = RMath.avg(list);
27         let node = list.begin(),
28             sum = 0,
29             n = list.size,
30             dev = 0;
31
32         if (!list.isEmpty) {
33             while ( node !== null) {
34                 let xi = node.value;
35                 sum += Math.pow(xi - mean, 2);
36                 node = node.next;
37             }
38
39             dev = Math.sqrt(sum/(n-1));
40         }
41
42         return dev;
43     }
44 }
45
46 module.exports = RMath;

```

```

1  /**
2   *  Rubén Ignacio Couch Ku.
3   */
4
5   const readline = require('readline');
6   const fs = require('fs');
7
8
9   let List = require('./List.js');
10  let RMath = require('./RMath.js');
11
12  let io = readline.createInterface({
13    input: process.stdin,
14    output: process.stdout
15  });
16
17  console.log('*****');
18  console.log('*** Entradas disponibles ***');
19  console.log('***      1) Teclado.      ***');
20  console.log('***      2) Archivo.      ***');
21  console.log('*****');
22
23  io.question('Seleccione una opción > ', onInput);
24
25  function onInput(input)
26  {
27    input = Number.parseInt(input);
28    switch (input) {
29      case 1:
30        io.question('Ingresa los números separados por un [espacio] >', onResponse);
31        break;
32      case 2:
33        io.question('¿Cuál es el nombre del archivo? > ', onResponse);
34        break;
35      default:
36        console.log('Opción incorrecta!');
37        process.exit(0);
38        break;
39    }
40
41    function onResponse(response) {
42      io.close();
43      readValues(input, response);
44    }
45  }
46

```

```

47
48 function readValues(input, param)
49 {
50     let values = [];
51     if (input === 1) {
52         values = param.split(' ').map((value) => {return Number.parseFloat(value)});
53         onValues(null, values);
54     } else if (input === 2) {
55         values = readFile(param, onValues);
56     }
57 }
58
59 function readFile(file, cb)
60 {
61     fs.readFile(file, 'utf-8', (err, data) => {
62         let values = [];
63         if (!err) {
64             values = data.split('\n').map((value)=>{ return Number.parseFloat(value)});
65         }
66         cb(err, values);
67     });
68 }
69
70 function onValues(err, values)
71 {
72     if (err) {
73         console.log(err);
74         process.exit(0);
75     }
76
77     var list = new List();
78     for (value of values) {
79         list.push(value);
80     }
81
82     let mean = RMath.avg(list);
83     let stdDev = RMath.stdDev(list);
84
85     console.log(`Mean: ${mean.toFixed(2)} `);
86     console.log(`Standard Deviation: ${stdDev.toFixed(2)} `);
87 }

```

5. Interfaz de usuario.

```
*****
*** Entradas disponibles ***
***   1) Teclado.   ***
***   2) Archivo.   ***
*****
Seleccione una opción > |
```

```
*****
*** Entradas disponibles ***
***   1) Teclado.   ***
***   2) Archivo.   ***
*****
Seleccione una opción > 2
¿Cuál es el nombre del archivo? > |
```

6. Resultados.

Entradas:

< >	test1.txt	x	≡	< >	test2.txt	x	≡	< >	test3.txt	x	≡
1	160			1	186			1	15.0		
2	591			2	699			2	69.9		
3	114			3	132			3	6.5		
4	229			4	272			4	22.4		
5	230			5	291			5	28.4		
6	270			6	331			6	65.9		
7	128			7	199			7	19.4		
8	1657			8	1890			8	198.7		
9	624			9	788			9	38.8		
10	1503			10	1601			10	138.2		

Salidas:

```
*****
*** Entradas disponibles ***
***      1) Teclado.      ***
***      2) Archivo.      ***
*****

Seleccione una opción > 2
¿Cuál es el nombre del archivo? > test1.txt

Mean: 550.60
Standard Deviation: 572.03
```

```
*****
*** Entradas disponibles ***
***      1) Teclado.      ***
***      2) Archivo.      ***
*****

Seleccione una opción > 2
¿Cuál es el nombre del archivo? > test2.txt

Mean: 638.90
Standard Deviation: 625.63
```

```
*****
*** Entradas disponibles ***
***      1) Teclado.      ***
***      2) Archivo.      ***
*****

Seleccione una opción > 2
¿Cuál es el nombre del archivo? > test3.txt

Mean: 60.32
Standard Deviation: 62.26
```

Resultados esperados:

1	160	186	15.0
2	591	699	69.9
3	114	132	6.5
4	229	272	22.4
5	230	291	28.4
6	270	331	65.9
7	128	199	19.4
8	1657	1890	198.7
9	624	788	38.8
10	1503	1601	138.2
Desv. estándar	572.03	625.63	62.26

Tabla 6.1

Conclusiones:

Todas las pruebas ejecutadas con los datos de entrada proporcionados en la tabla 6.1 fueron satisfactorias.