

Título del informe

Tema a tratar

Integrantes: Integrante 1
Integrante 2
Profesor: Profesor 1
Auxiliares: Auxiliar 1
Auxiliar 2
Ayudantes: Ayudante 1
Ayudante 2
Ayudante del laboratorio: Ayudante 1

Fecha de realización: 21 de mayo de 2018
Fecha de entrega: 21 de mayo de 2018
Santiago, Chile

Resumen

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Índice de Contenidos

Lista de Códigos

1.	Codigo en c.	1
2.	Ejemplo en C#.	2
3.	Suma en C++.	3
4.	Docker.	3
5.	Resultado del análisis con TEFAME.	3
6.	Ejemplo en HTML5.	4
7.	Ejemplo en Matlab.	4
8.	Ejemplo xml.	5
9.	Ejemplo en Python.	5
10.	Ejemplo en Java.	5
11.	Ejemplo en Javascript.	6
12.	Un arreglo en JSON.	6
13.	Imágenes múltiples.	6
14.	Algo de perl.	7
15.	Ejemplo php.	8
16.	Ejemplo con ruby.	9
17.	Merge two tables.	10
18.	Algoritmo.	11

```
1 #include <stdio.h>
2 int main(){
3     int i, j, rows;
4
5     printf("Enter number of rows: ");
6     scanf("%d",&rows);
7
8     for(i=1; i<=rows; ++i){
9         for(j=1; j<=i; ++j){
10             printf("* ");
11         }
12         printf("\n");
13     }
14     return 0;
15 }
```

Código 1: Codigo en c.

```
1 #include <stdio.h>
2 int main(){
3     int i, j, rows;
4
5     printf("Enter number of rows: ");
6     scanf("%d",&rows);
7
8     for(i=1; i<=rows; ++i){
9         for(j=1; j<=i; ++j){
10             printf("* ");
11         }
12         printf("\n");
13     }
14     return 0;
15 }
```

Código 2: Ejemplo en C#.

```
1  /*
2  * C# Program to Get a Number and Display the Sum of the Digits
3  */
4  using System;
5  using System.Collections.Generic;
6  using System.Linq;
7  using System.Text;
8
9  namespace Program
10 {
11     class Program
12     {
13         static void Main(string[] args)
14         {
15             int num, sum = 0, r;
16             Console.WriteLine("Enter a Number : ");
17             num = int.Parse(Console.ReadLine());
18             while (num != 0)
19             {
20                 r = num % 10;
21                 num = num / 10;
22                 sum = sum + r;
23             }
24             Console.WriteLine("Sum of Digits of the Number : "+sum);
25             Console.ReadLine();
26
27         }
28     }
29 }
```

Código 3: Suma en C++.

```

1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     int n, sum = 0;
7
8     cout << "Enter a positive integer: ";
9     cin >> n;
10
11     for (int i = 1; i <= n; ++i) {
12         sum += i;
13     }
14
15     cout << "Sum = " << sum;
16     return 0;
17 }

```

Código 4: Docker.

```

1 version: '2'
2 services:
3 web:
4 build: .
5 ports:
6 - "5000:5000"
7 volumes:
8 - ./code
9 - logvolume01:/var/log
10 links:
11 - redis
12 redis:
13 image: redis
14 volumes:
15 logvolume01: {}

```

Código 5: Resultado del análisis con TEFAME.

```

1 TEFAME - Toolbox para Elemento Finitos y Analisis
2 Matricial de Estructuras en MATLAB
3
4 Propiedades de entrada modelo:
5
6 Nodos:
7 Numero de nodos: 4
8 Coordenadas nodo N1: 0 0
9 Coordenadas nodo N2: 800 0
10 Coordenadas nodo N3: 400 400
11 Coordenadas nodo N4: 400 800
12
13 Elementos:

```

```

14 Numero de elementos: 6
15 Elemento E1:  Largo: 800          Area: 20          Eo: 200000
16 Elemento E2:  Largo: 565.6854    Area: 20          Eo: 200000
17 Elemento E3:  Largo: 565.6854    Area: 20          Eo: 200000
18 Elemento E4:  Largo: 894.4272    Area: 20          Eo: 200000
19 Elemento E5:  Largo: 400          Area: 20          Eo: 200000
20 Elemento E6:  Largo: 894.4272    Area: 20          Eo: 200000
21
22 Resultados del analisis:
23
24 Desplazamientos nodos:
25 Desplazamientos nodo N1: 0 0
26 Desplazamientos nodo N2: 0.016 0
27 Desplazamientos nodo N3: 0.008 -0.013
28 Desplazamientos nodo N4: 0.053 -0.016
29
30 Reacciones:
31 Reacciones nodo N1: -80 -20
32 Reacciones nodo N2: 0 140
33 Reacciones nodo N3: 0 0
34 Reacciones nodo N4: 0 0
35
36 Esfuerzos Elementos:
37 Elemento E1: -78.4273          TRACCION
38 Elemento E2: 23.836           COMPRESION
39 Elemento E3: 23.836           COMPRESION
40 Elemento E4: -41.2047         TRACCION
41 Elemento E5: 33.7093          COMPRESION
42 Elemento E6: 137.6807         COMPRESION

```

Código 6: Ejemplo en HTML5.

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Página</title>
5 </head>
6 <body>
7   <style>
8     .titulo {
9       color: #ff0000;
10    }
11  </style>
12  <div class="titulo">Hola</div>
13 </body>
14 </html>

```

Código 7: Ejemplo en Matlab.

```

1 % Se crea gráfico
2 f = figure(1);
3 hold on;

```

```

4 movegui(f, 'center');
5 xlabel('td/Tn'); ylabel('FAD=Umax/Uf0');
6 title('Espectro de pulso de desplazamiento');
7
8 for j = 1:length(BETA)
9     fad = ones(1, NDATOS); % Arreglo para el FAD, uno para cada r
10    for i = 1:NDATOS
11        [t, u_t, ~, ~] = main(BETA(j), r(i), M, K, FO, 0);
12        fad(i) = max(abs(u_t)) / uf0;
13    end
14 mx = find(fad == max(fad(:)));
15 fprintf('BETA=%.2f, MAX: FAD=%.3f, TD/TN=%.3f\n', BETA(j), fad(mx), tdn(mx));
16 plot(tdn, fad, 'DisplayName', strcat('\beta=', sprintf('%.2f', BETA(j))));
17 end

```

Código 8: Ejemplo xml.

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
3 xmlns:xs="http://www.w3.org/2001/XMLSchema">
4   <xs:element name="points">
5     <xs:complexType>
6       <xs:sequence>
7         <xs:element maxOccurs="unbounded" name="point">
8           <xs:complexType>
9             <xs:attribute name="x" type="xs:unsignedShort" use="required" />
10            <xs:attribute name="y" type="xs:unsignedShort" use="required" />
11          </xs:complexType>
12        </xs:element>
13      </xs:sequence>
14    </xs:complexType>
15  </xs:element>
16 </xs:schema>

```

Código 9: Ejemplo en Python.

```

1 import numpy as np
2
3 def incmatrix(genl1, genl2):
4     m = len(genl1)
5     n = len(genl2)
6     M = None # Comentario 1
7     VT = np.zeros((n*m, 1), int) # Comentario 2

```

Código 10: Ejemplo en Java.

```

1 import java.io.IOException;
2 import javax.servlet.*;
3
4 // Hola mundo
5 public class Hola extends GenericServlet {
6     public void service(ServletRequest request, ServletResponse response)

```



```

7  throws ServletException, IOException{
8      response.setContentType("text/html");
9      PrintWriter pw = response.getWriter();
10     pw.println("Hola, mundo!");
11     pw.close();
12 }
13 }

```

Código 11: Ejemplo en Javascript.

```

1 $.urlParam = function (name) {
2     let results = new RegExp('[\?&]' + name + '=(^&#)*').exec(window.location.href);
3     if (results == null) {
4         return null;
5     } else {
6         return decodeURI(results[1]) || 0;
7     }
8 };

```

Código 12: Un arreglo en JSON.

```

1 {"menu": {
2     "id": "file",
3     "value": "File",
4     "popup": {
5         "menuitem": [
6             {"value": "New", "onclick": "CreateNewDoc()"},
7             {"value": "Open", "onclick": "OpenDoc()"},
8             {"value": "Close", "onclick": "CloseDoc()"}
9         ]
10    }
11 }}

```

Código 13: Imágenes múltiples.

```

1 \begin{images}[\label{imagenmultiple}]{Ejemplo de imagen múltiple.}
2     \addimage{ejemplos/test-image}{width=6.5cm}{Ciudad}
3     \addimage{ejemplos/test-image-wrap}{width=5cm}{Apolo}
4     \addimage{ejemplos/test-image}{width=12cm}{Ciudad más grande}
5 \end{images}

```

Código 14: Algo de perl.

```
1 #!/usr/bin/perl
2 use strict;
3 use warnings;
4
5 # first, create your message
6 use Email::MIME;
7 my $message = Email::MIME->create(
8     header_str => [
9         From      => 'you@example.com',
10        To        => 'friend@example.com',
11        Subject   => 'Happy birthday!',
12    ],
13    attributes => {
14        encoding => 'quoted-printable',
15        charset  => 'ISO-8859-1',
16    },
17    body_str => "Happy birthday to you!\n",
18 );
19
20 # send the message
21 use Email::Sender::Simple qw(sendmail);
22 sendmail($message);
```

Código 15: Ejemplo php.

```
1 <?php
2 $target_dir = "uploads/";
3 $target_file = $target_dir . basename($_FILES["fileToUpload"]["name"]);
4 $uploadOk = 1;
5 $imageFileType = strtolower(pathinfo($target_file,PATHINFO_EXTENSION));
6 // Check if image file is a actual image or fake image
7 if(isset($_POST["submit"])) {
8     $check = getimagesize($_FILES["fileToUpload"]["tmp_name"]);
9     if($check !== false) {
10         echo "File is an image - " . $check["mime"] . ".";
11         $uploadOk = 1;
12     } else {
13         echo "File is not an image.";
14         $uploadOk = 0;
15     }
16 }
17 ?>
```

Código 16: Ejemplo con ruby.

```
1 class DataFile < ActiveRecord::Base
2   attr_accessor :upload
3
4   def self.save_file(upload)
5
6     file_name = upload['datafile'].original_filename if (upload['datafile'] != '')
7     file = upload['datafile'].read
8
9     file_type = file_name.split('.').last
10    new_name_file = Time.now.to_i
11    name_folder = new_name_file
12    new_file_name_with_type = "#{new_name_file}." + file_type
13
14    image_root = "#{RAILS_CAR_IMAGES}"
15
16
17    Dir.mkdir(image_root + "#{name_folder}");
18    File.open(image_root + "#{name_folder}/" + new_file_name_with_type, "wb") do |f|
19      f.write(file)
20    end
21
22  end
23 end
```

Código 17: Merge two tables.

```
1 SELECT ChargeNum, CategoryID, SUM(Hours)
2 FROM KnownHours
3 GROUP BY ChargeNum, CategoryID
4 UNION ALL
5 SELECT ChargeNum, 'Unknown' AS CategoryID, SUM(Hours)
6 FROM UnknownHours
7 GROUP BY ChargeNum
```

```
100 <?xml version="1.0" encoding="utf-8"?>
101 <xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
102   xmlns:xs="http://www.w3.org/2001/XMLSchema">
103   <xs:element name="points">
104     <xs:complexType>
105       <xs:sequence>
106         <xs:element maxOccurs="unbounded" name="point">
107           <xs:complexType>
108             <xs:attribute name="x" type="xs:unsignedShort" use="required" />
109             <xs:attribute name="y" type="xs:unsignedShort" use="required" />
110           </xs:complexType>
111         </xs:element>
112       </xs:sequence>
113     </xs:complexType>
114   </xs:element>
115 </xs:schema>
```

Código 18: Algoritmo.

```
1 input: int N, int D
2 output: int
3 begin
4   res  $\leftarrow$  0
5   while  $N \geq D$ 
6     N  $\leftarrow$  N - D
7     res  $\leftarrow$  res + 1
8   end
9   return res
10 end
```
