



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS  
CENTRO DE DESARROLLO, FACULTAD 4

## TITULO DE LA TESIS

**Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas**

Autor: Nombre y apellidos del primer autor

Tutora: MSc. Maybel Díaz Capote

**La Habana, 2024**

*aquí va la frase*

*aquí va la dedicatoria de la tesis*

---

## Agradecimientos

---

*aquí van los agradecimientos*

---

## Declaración de autoría

---

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales sobre esta, con carácter exclusivo.

Para que así conste firmamos la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

---

Nombre y apellidos del primer autor  
Autor

---

MSc. Maybel Díaz Capote  
Tutora

---

## Resumen

---

aquí poner el resumen, no pasar las 250 palabras

**Palabras clave:** aquí van las palabras claves, ordenadas alfabeticamente.

<b>Introducción</b>	<b>1</b>
<b>1 Nombre del cap</b>	<b>2</b>
1.1 Sección de prueba . . . . .	2
<b>2 Nombre del cap</b>	<b>4</b>
2.1 Sección de prueba . . . . .	4
2.2 Ejemplos de código fuente . . . . .	4
<b>3 Ejemplo de tabla grande</b>	<b>10</b>
3.1 Tabla aleatoria . . . . .	10
3.2 Tablas de ingeniería . . . . .	11
<b>4 Nombre del cap</b>	<b>16</b>
<b>Conclusiones</b>	<b>17</b>
<b>Recomendaciones</b>	<b>18</b>
<b>Apéndices</b>	<b>19</b>
<b>A Proyectos y Avaes</b>	<b>20</b>
<b>B Otro reporte</b>	<b>21</b>
B.1 Otra sección de muestra . . . . .	21
B.2 Otro ensamble Industrial . . . . .	21

---

## Índice de figuras

---

1.1	Vista del Arco de Moncloa . . . . .	2
-----	-------------------------------------	---



1.1	Caption de Prueba . . . . .	3
3.1	Feasible triples for highly variable Grid, MLMMH. . . . .	10
3.1	continued from previous page . . . . .	11
3.2	Historia de usuario # 1 . . . . .	11
3.3	Tarjeta CRC # 1 . . . . .	12
3.4	Tarjeta CRC # 2 . . . . .	12
3.5	Estimación de esfuerzo por historia de usuario . . . . .	12
3.6	Estimación de esfuerzo por historia de usuario . . . . .	13
3.7	Plan de duración de las iteraciones . . . . .	13
3.8	Tarea de ingeniería # 1 . . . . .	13
3.9	Tarea de desarrollo # 1 . . . . .	13
3.10	Tarea de desarrollo # 2 . . . . .	14
3.11	Tarea de ingeniería # 2 . . . . .	14
3.12	Prueba de aceptación # 1 . . . . .	14
3.13	Prueba de aceptación # 2 . . . . .	14



Escribir aquí el capítulo 1. Ejemplo de acrónimo es la Universidad de las Ciencias Informáticas (UCI). Esto es una cita de ejemplo (**Ou2013**). En la próxima oración se muestra un ejemplo de un acrónimo en inglés. Los sistemas de Supervisión, Control y Adquisición de Datos (SCADA, por sus siglas en inglés) son de gran importancia para la industria.

## 1.1. Sección de prueba

Esta es una prueba de número: 5.3 donde se utiliza la coma automáticamente como separador.  
Esto es una cita de ejemplo (**Alfadhlan2011**; **Mathew2010**).



Figura 1.1. Vista del Arco de Moncloa

En la Tabla 1.1 se observa ...

Tabla 1.1. Caption de Prueba

No. 1	No. 2	No. 3	No. 4	No. 5
12	12	56	56	56
12	12	56	56	56

- Uno
- Dos
- Tres
- Cuatro
- Cinco

2.1. Sección de prueba

Escribir aquí el capítulo 2. A continuación un ejemplo de enlace al anexo Bloque Corona. Este es un ejemplo de matriz con bordes numerados (Mathew2010).

Padre

1

2

3

4

1

2

3

4

0

1

0

0

0

0

0

1

1

0

0

0

0

0

1

0

→ Seleccionada aleatoriamente

→ Seleccionada aleatoriamente

Madre

1

2

3

4

1

2

3

4

1

0

0

0

0

1

0

0

0

0

1

0

0

0

0

1

Hijo

1

2

3

4

1

2

3

4

0

1

0

0

0

0

1

0

1

0

0

0

0

0

0

1

(2.1.1)

Este es un ejemplo de cita (Mathew2010) Esta es una prueba de cómo se escribe un pseudocódigo en  $\text{\LaTeX}$ .

2.2. Ejemplos de código fuente

Este es un ejemplo de cita (Mathew2010)

**Algoritmo 1** Estrategia de BellmanKalaba

---

```
1: procedure BELLMANKALABA( $G, u, l, p$ )
2:   for all  $v \in V(G)$  do
3:      $l(v) \leftarrow \infty$ 
4:   end for
5:    $l(u) \leftarrow 0$ 
6:   repeat
7:     for  $i \leftarrow 1, n$  do
8:        $min \leftarrow l(v_i)$ 
9:       for  $j \leftarrow 1, n$  do
10:        if  $min > e(v_i, v_j) + l(v_j)$  then
11:           $min \leftarrow e(v_i, v_j) + l(v_j)$ 
12:           $p(i) \leftarrow v_j$ 
13:        end if
14:      end for
15:       $l'(i) \leftarrow min$ 
16:    end for
17:     $changed \leftarrow l \neq l'$ 
18:     $l \leftarrow l'$ 
19:  until  $\neg changed$ 
20: end procedure
```

---

Código fuente 2.1. Búsqueda de máximo

---

```
int maxSearch( const int* v )
{
    /**
     * This is a text that takes
     * two lines
     */
    int currentMax = 0;
    for( int i = 0; i < MAX_NUMBER; i++)
    {
        currentMax = max(v[i], currentMax); // here we are using std::max
    }

    return currentMax;
}
```

---

Código fuente 2.2. Ejemplo de código en Java

---

```
1 /**
2  * HelloWorldApp class prints
3  * "Hello World!" to standard output
4  */
5 public class HelloWorldApp{
```

---

```
6 public static void main(String argv[])
7 {
8     // Display string
9     System.out.println("Hello World!");
10 }
11 }
```

---

Código fuente 2.3. Ejemplo de código en PHP

---

```
1 <?php
2 /**
3  * Retrieves a list of models based on the current search/filter conditions.
4  * @return CActiveDataProvider the data provider that can return the models
5  * based on the search/filter conditions.
6  */
7 public function search()
8 {
9     // Warning: Please modify the following code to remove attributes that
10    // should not be searched.
11
12    $criteria=new CDbCriteria;
13    $criteria->compare('idtarea',$this->idtarea);
14    $criteria->compare('nombre',$this->nombre,true);
15    $criteria->compare('descripcion',$this->descripcion,true);
16    $criteria->compare('fecha',$this->fecha,true);
17    $criteria->compare('idinstructora',$this->idinstructora);
18
19    // $user=Yii::app()->user->name;
20    $rol=Usser::model()->getRolThisUser($user);
21    $rol=Usser::model()->getNameRol($rol);
22    if('Instructora'==$rol)
23    {
24        $criteria->join="JOIN instructora ON(t.idinstructora=instructora.
25        idinstructora ) ";
26        $criteria->condition="instructora.idusuario='".$user."'";
27    }
28
29    return new CActiveDataProvider($this, array(
30        'criteria'=>$criteria,
31        'pagination'=>array('pageSize'=>5)
32    ));
33 }
```

---

Código fuente 2.4. Ejemplo de código en PHP

---

```
1 public function evidenciaXUsuarioAction($id)
2 {
3     $em = $this->getDoctrine()->getManager();
4
5     $usuario = $em->getRepository('GIVIPBundle:Usuario')->find($id);
6     if(!$usuario){
7         $this->get('session')->getFlashBag()->add('error', 'El usuario
8         seleccionado no existe en el sistema.');
```

---

```
9         return $this->redirect($this->generateUrl('evidencia'));
10     }
11     $nombre = $usuario->getNombre() ." ". $usuario->getApellidos();
12     if($this->get('security.context')->isGranted('ROLE_VIP1')){
13         $consulta = $em->createQuery("select e, i, u from GIVIPBundle:Evidencia e
14                                     JOIN e.indicador i JOIN e.usuario u WHERE e.
15                                     usuario=:id");
16     }
17     else
18     {
19         $consulta = $em->createQuery("select e, i, u from GIVIPBundle:Evidencia e
20                                     JOIN e.indicador i JOIN e.usuario u WHERE e.usuario=:id and e.estado=
21                                     true");
22     }
23
24     $consulta->setParameter('id', $id);
25
26     $res = $consulta->getResult();
27     $paginator = $this->get('ideup.simple_paginator');
28     $paginator->setItemsPerPage(10, 'resultado');
29     $entities = $paginator->paginate($res, 'resultado')->getResult();
30
31     $deleteForm = $this->createDeleteForm(0);
32     $buscarForm = $this->createSearchForm();
33
34     return $this->render('GIVIPBundle:Evidencia:evidenciausuario.html.twig',
35         array(
36             'entities' => $entities, 'delete_form' => $deleteForm->createView(),
37             'buscar_form' => $buscarForm->createView(), 'usuario'=>$nombre
38         ));
39 }
```

---

Código fuente 2.5. Ejemplo de código en Python

```
1 class TelegramRequestHandler(object):
2     def handle(self):
3         addr = self.client_address[0]          # Client IP-adress
4         telegram = self.request.recv(1024)     # Recieve telegram
5         print "From: %s, Received: %s" % (addr, telegram)
6         return
```

---

Código fuente 2.6. Ejemplo de código en htmlcssjs

```
1 @media only screen and (min-width: 768px) and (max-width: 991px) {
2
3     #main {
4         width: 712px;
5         padding: 100px 28px 120px;
6     }
7
8     /* .mono {
9         font-size: 90%;
10    } */
11
12    .cssbtn a {
```

---



```
13     margin-top: 10px;
14     margin-bottom: 10px;
15     width: 60px;
16     height: 60px;
17     font-size: 28px;
18     line-height: 62px;
19 }
```

---

Código fuente 2.7. Ejemplo de código en htmlcssjs

---

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Listings Style Test</title>
5     <meta charset="UTF-8">
6     <style>
7       /* CSS Test */
8       * {
9         padding: 0;
10        border: 0;
11        margin: 0;
12      }
13    </style>
14    <link rel="stylesheet" href="css/style.css" />
15  </head>
16  <header> hey </header>
17  <article> this is a article </article>
18  <body>
19    <!-- Paragraphs are fine -->
20    <div id="box">
21      <p>
22        Hello World
23      </p>
24      <p>Hello World</p>
25      <p id="test">Hello World</p>
26      <p></p>
27    </div>
28    <div>Test</div>
29    <!-- HTML script is not consistent -->
30    <script src="js/benchmark.js"></script>
31    <script>
32      function createSquare(x, y) {
33        // This is a comment.
34        var square = document.createElement('div');
35        square.style.width = square.style.height = '50px';
36        square.style.backgroundColor = 'blue';
37
38        /*
39         * This is another comment.
40         */
41        square.style.position = 'absolute';
42        square.style.left = x + 'px';
43        square.style.top = y + 'px';
44
45        var body = document.getElementsByTagName('body')[0];
46        body.appendChild(square);
```

```
47     };
48
49     // Please take a look at +=
50     window.addEventListener('mousedown', function(event) {
51         // German umlaut test: Berührungspunkt ermitteln
52         var x = event.touches[0].pageX;
53         var y = event.touches[0].pageY;
54         var lookAtThis += 1;
55     });
56 </script>
57 </body>
58 </html>
```

---

## Ejemplo de tabla grande

## 3.1. Tabla aleatoria

Tabla 3.1. Feasible triples for highly variable Grid, MLMMH.

Time (s)	Triple chosen	Other feasible triples
0	(1, 11, 13725)	(1, 12, 10980), (1, 13, 8235), (2, 2, 0), (3, 1, 0)
2745	(1, 12, 10980)	(1, 13, 8235), (2, 2, 0), (2, 3, 0), (3, 1, 0)
5490	(1, 12, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
8235	(1, 12, 16470)	(1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0)
10980	(1, 12, 16470)	(1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0)
13725	(1, 12, 16470)	(1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0)
16470	(1, 13, 16470)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
19215	(1, 12, 16470)	(1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0)
21960	(1, 12, 16470)	(1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0)
24705	(1, 12, 16470)	(1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0)
27450	(1, 12, 16470)	(1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0)
30195	(2, 2, 2745)	(2, 3, 0), (3, 1, 0)
32940	(1, 13, 16470)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
35685	(1, 13, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
38430	(1, 13, 10980)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
41175	(1, 12, 13725)	(1, 13, 10980), (2, 2, 2745), (2, 3, 0), (3, 1, 0)
43920	(1, 13, 10980)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
46665	(2, 2, 2745)	(2, 3, 0), (3, 1, 0)
49410	(2, 2, 2745)	(2, 3, 0), (3, 1, 0)
52155	(1, 12, 16470)	(1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0)
54900	(1, 13, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
57645	(1, 13, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
60390	(1, 12, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
63135	(1, 13, 16470)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
65880	(1, 13, 16470)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
68625	(2, 2, 2745)	(2, 3, 0), (3, 1, 0)
71370	(1, 13, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
74115	(1, 12, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
76860	(1, 13, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
79605	(1, 13, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)

Continued on next page

Tabla 3.1. continued from previous page

Time (s)	Triple chosen	Other feasible triples
82350	(1, 12, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
85095	(1, 12, 13725)	(1, 13, 10980), (2, 2, 2745), (2, 3, 0), (3, 1, 0)
87840	(1, 13, 16470)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
90585	(1, 13, 16470)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
93330	(1, 13, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
96075	(1, 13, 16470)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
98820	(1, 13, 16470)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
101565	(1, 13, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
104310	(1, 13, 16470)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
107055	(1, 13, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
109800	(1, 13, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
112545	(1, 12, 16470)	(1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0)
115290	(1, 13, 16470)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
118035	(1, 13, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
120780	(1, 13, 16470)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
123525	(1, 13, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
126270	(1, 12, 16470)	(1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0)
129015	(2, 2, 2745)	(2, 3, 0), (3, 1, 0)
131760	(2, 2, 2745)	(2, 3, 0), (3, 1, 0)

### 3.2. Tablas de ingeniería

Tabla 3.2. Historia de usuario # 1

Historia de usuario	
Número: 1	Nombre: Cargar archivo
Usuario: Especialista	
Prioridad en negocio: Alta	Riesgo en desarrollo: Bajo
Puntos estimados: 0.8	Iteración asignada: 1
Programador responsable: Ernesto Gil Capote	
<p><b>Descripción:</b> Permite cargar los datos de los casos de estudio a utilizar en el módulo con el objetivo de determinar secuencias factibles. Estos están almacenados en archivos con extensión .asp. La estructura está compuesta por los parámetros de entrada de la aplicación para cada caso de estudio tales como: dimensión, nombre de la estructura mecánica, cantidad de piezas, la MD correspondiente y el vector de herramientas en caso de que lo posea.</p> <ul style="list-style-type: none"> <li>• rojo</li> <li>• blanco</li> </ul>	
<p><b>Observaciones:</b> En caso de que el usuario no cargue un archivo, el sistema debe mostrar un mensaje de alerta con dicha explicación. Si se introduce un archivo que contiene caracteres no válidos o falta alguno de los elementos necesarios para la ejecución del módulo, este debe lanzar la excepción correspondiente. En caso de que el usuario no cargue un archivo, el sistema debe mostrar un mensaje de alerta con dicha explicación. Si se introduce un archivo que contiene caracteres no válidos o falta alguno de los elementos necesarios para la ejecución del módulo, este debe lanzar la excepción correspondiente.</p>	

Esta es referencia a una tabla de historia de usuario 3.2

Tabla 3.3. Tarjeta CRC # 1

Tarjeta CRC	
<b>Clase:</b> AlgoritmoGenetico	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> <li>• Crear el conjunto de genes necesarios para la creación de un individuo</li> <li>• Realizar el proceso de mutación y recombinación</li> <li>• Crear el conjunto de genes necesarios para la creación de un individuo</li> <li>• Realizar el proceso de mutación y recombinación</li> </ul>	Población OperadorProbabilidad OperadorParejas OperadorReproduccion Población

Tabla 3.4. Tarjeta CRC # 2

Tarjeta CRC	
<b>Clase:</b> AlgoritmoGenetico	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> <li>• Crear el conjunto de genes necesarios para la creación de un individuo</li> <li>• Realizar el proceso de mutación y recombinación</li> <li>• Crear el conjunto de genes necesarios para la creación de un individuo</li> <li>• Realizar el proceso de mutación y recombinación</li> </ul>	Población OperadorProbabilidad OperadorParejas OperadorReproduccion Población

Tabla 3.5. Estimación de esfuerzo por historia de usuario

Iteración	Historias de usuario		Puntos estimados (semanas)
1	1	Cargar Archivo	0.8
	2	Determinar Secuencia de Ensamble	0.2
	3	Gestionar restricción cambios de herramienta	0.5
	4	Graficar secuencias de ensamble	0.6
2	5	Graficar secuencias de ensamble	0.3
	6	Graficar secuencias de ensamble	0.3
	7	Graficar secuencias de ensamble	0.3
	8	Graficar secuencias de ensamble	0.2
	9	Graficar secuencias de ensamble	0.2
3	10	Determinar Secuencia de Ensamble	0.5
4	11	Graficar secuencias de ensamble	0.1
	12	Graficar ensamble	0.1
<b>Total</b>			<b>4.1</b>

Tabla 3.6. Estimación de esfuerzo por historia de usuario

Iteración	Historias de usuario		Puntos estimados (semanas)
1	1	Cargar Fichero	0.8
	2	Determinar Secuencia de Ensamble	0.2
	3	Gestionar restricción cambios de herramienta	0.5
	4	Esta es la prueba para el tinguiri	0.5
2	5	Graficar secuencias de ensamble	0.3
	6	Graficar secuencias de ensamble	0.3
	7	Graficar secuencias de ensamble	0.2
	8	Graficar secuencias de ensamble	0.2
3	9	Determinar Secuencia de Ensamble	2.5
4	10	Graficar secuencias de ensamble	1.1
<b>Total</b>			<b>6.6</b>

Esto es una prueba de Tabla 3.5

Tabla 3.7. Plan de duración de las iteraciones

Iteración	Historias de usuario		Duración (semanas)
1	1	Cargar Fichero	3
	2	Determinar Secuencia de Ensamble	
	3	Gestionar restricción cambios de herramienta	
	4	Esta es la prueba para el tinguiri	
2	5	Graficar secuencias de ensamble	1.0
	6	Graficar secuencias de ensamble	
	7	Graficar secuencias de ensamble	
	8	Graficar secuencias de ensamble	
3	9	Determinar Secuencia de Ensamble	2.5
4	10	Graficar secuencias de ensamble	1.1
<b>Total</b>			<b>7.6</b>

Esto es una prueba de Tabla 3.7

Tabla 3.8. Tarea de ingeniería # 1

Tarea	
Número de tarea: 1	Número de Historia de usuario: 4
Nombre de la tarea: Registrar usuario	
Tipo de tarea: Registrarse	Puntos estimados: 1
Fecha de inicio: 6 de mayo de 2014	Fecha de fin: 24 de octubre de 2014
Descripción: El usuario introduce sus datos para poder registrarse.	

Tabla 3.9. Tarea de desarrollo # 1

Tarea	
Número de tarea: 1	Número de Historia de usuario: 4
Nombre de la tarea: Registrar usuario	
Tipo de tarea: Registrarse	Puntos estimados: 1
Fecha de inicio: 6 de mayo de 2014	Fecha de fin: 24 de octubre de 2014

Continúa en la próxima página

Tabla 3.9. Continuación de la página anterior

<b>Programador responsable:</b> Alberto Medina Ramírez
<b>Descripción:</b> El usuario introduce sus datos para poder registrarse.

Tabla 3.10. Tarea de desarrollo # 2

Tarea	
<b>Número de tarea:</b> 2	<b>Número de Historia de usuario:</b> 4
<b>Nombre de la tarea:</b> Registrar usuario	
<b>Tipo de tarea:</b> Registrarse	<b>Puntos estimados:</b> 1
<b>Fecha de inicio:</b> 6 de mayo de 2014	<b>Fecha de fin:</b> 24 de octubre de 2014
<b>Descripción:</b> El usuario introduce sus datos para poder registrarse.	

Tabla 3.11. Tarea de ingeniería # 2

Tarea	
<b>Número de tarea:</b> 2	<b>Número de Historia de usuario:</b> 4
<b>Nombre de la tarea:</b> Registrar usuario	
<b>Tipo de tarea:</b> Registrarse	<b>Puntos estimados:</b> 1
<b>Fecha de inicio:</b> 6 de enero de 2014	<b>Fecha de fin:</b> 24 de noviembre de 2014
<b>Programador responsable:</b> Alberto Medina Ramírez	
<b>Descripción:</b> El usuario introduce sus datos para poder registrarse.	

Referencia 3.9

Tabla 3.12. Prueba de aceptación # 1

Caso de prueba de aceptación	
<b>Código:</b> HU2_P1	<b>Historia de usuario:</b> 2
<b>Nombre:</b> Autenticar usuario en el sistema.	
<b>Descripción:</b> Prueba para la funcionalidad autenticar usuario.	
<b>Condiciones de ejecución:</b> El usuario debe estar previamente registrado. El usuario y contraseña deben ser válidos.	
<b>Pasos de ejecución:</b> Se intenta autenticar un usuario en el sistema con los datos válidos.	
<b>Resultados esperados:</b> El usuario se autentifica correctamente en el sistema.	

Tabla 3.13. Prueba de aceptación # 2

Caso de prueba de aceptación	
<b>Código:</b> HU2_P1	<b>Historia de usuario:</b> 2
<b>Nombre:</b> Autenticar usuario en el sistema.	
<b>Descripción:</b> Prueba para la funcionalidad autenticar usuario.	
<b>Condiciones de ejecución:</b> El usuario debe estar previamente registrado. El usuario y contraseña deben ser válidos.	

Continúa en la próxima página

Tabla 3.13. Continuación de la página anterior

<b>Pasos de ejecución:</b> Se intenta autenticar un usuario en el sistema con los datos válidos.
<b>Resultados esperados:</b> El usuario se autentifica correctamente en el sistema.

Prueba de link 3.12



---

Nombre del cap

---

Escribir aquí el capítulo 3. Esto es un ejemplo de cómo se puede *linkear* el anexo Controlador Industrial. Esto es otra cita de la inicial como ejemplo también (**Ou2013**). Esto es una prueba de link al Anexo B.2. Esto es una prueba de link a la palabra sentencia

---

## Conclusiones

---

Escribir aquí las conclusiones.

---

## Recomendaciones

---

Escribe aquí las recomendaciones de tu trabajo.

# **Apéndices**



REPORTE DEL PLANIFICADOR DE SECUENCIA DE ENSAMBLE

Ing. Maybel Díaz Capote  
Universidad de las Ciencias Informáticas, La Habana, Cuba  
Fecha: sábado 5 de octubre de 2013

DETALLES DEL REPORTE:

Caso de estudio: Bloque Corona  
Reorientaciones: sí  
Cambios de herramientas: no  
Vértices: 17  
Aristas: 136  
Número total de hormigas: 11

Secuencias de ensamble:

Hormiga: 1  
Reorientaciones: 2

Salida:  
(11, +z)-(10, +z)-(16, +x)-(15, +x)-(14, +x)-(8, +x)-(12, +z)-(9, +z)-(5, +z)-(7, +z)-(17, +z)-(4, +z)-(6, +z)-(3, +z)-(2, +z)-(13, +z)-(1, +z)

Hormiga: 2  
Reorientaciones: 3

Salida:  
(11, +z)-(10, +z)-(16, +x)-(15, +x)-(14, +x)-(8, +x)-(12, +z)-(9, +z)-(5, +z)-(4, +z)-(6, +z)-(3, +z)-(2, +z)-(13, +z)-(7, +y)-(17, +y)-(1, +y)

Hormiga: 3  
Reorientaciones: 3

Salida:  
(11, +z)-(10, +z)-(16, +x)-(15, +x)-(14, +x)-(8, +x)-(12, +z)-(9, +z)-(5, +z)-(3, +z)-(6, +z)-(17, +z)-(1, +z)-(13, +z)-(7, +y)-(4, +y)-(2, +y)

Hormiga: 4  
Reorientaciones: 2

Salida:  
(11, +z)-(10, +z)-(16, +x)-(15, +x)-(14, +x)-(8, +x)-(12, +z)-(9, +z)-(5, +z)-(7, +z)-(17, +z)-(4, +z)-(13, +z)-(1, +z)-(6, +z)-(3, +z)-(2, +z)

Hormiga: 5  
Reorientaciones: 3

Salida:  
(11, +z)-(10, +z)-(16, +x)-(14, +x)-(15, +x)-(12, +z)-(9, +z)-(5, +z)-(4, +z)-(6, +z)-(7, +z)-(17, +z)-(1, +z)-(2, +z)-(13, +z)-(8, +x)-(3, +x)

Hormiga: 6  
Reorientaciones: 2

Salida:  
(11, +z)-(10, +z)-(16, +x)-(15, +x)-(14, +x)-(8, +x)-(12, +z)-(9, +z)-(5, +z)-(7, +z)-(17, +z)-(4, +z)-(1, +z)-(6, +z)-(2, +z)-(13, +z)-(3, +z)

### B.1. Otra sección de muestra

De una sola sentencia

### B.2. Otro ensamble Industrial



#### REPORTE DEL PLANIFICADOR DE SECUENCIA DE ENSAMBLE

Ing. Maybel Díaz Capote  
Universidad de las Ciencias Informáticas, La Habana, Cuba  
Fecha: sábado 5 de octubre de 2013

##### DETALLES DEL REPORTE:

**Caso de estudio:** Bloque Corona  
**Reorientaciones:** sí  
**Cambios de herramientas:** no  
**Vértices:** 17  
**Aristas:** 136  
**Número total de hormigas:** 11

##### Secuencias de ensamble:

**Hormiga: 1**  
Reorientaciones: 2

Salida:  
(11, +z)-(10, +z)-(16, +x)-(15, +x)-(14, +x)-(8, +x)-(12, +z)-(9, +z)-(5, +z)-(7, +z)-(17, +z)-(4, +z)-(6, +z)-(3, +z)-(2, +z)-(13, +z)-(1, +z)

**Hormiga: 2**  
Reorientaciones: 3

Salida:  
(11, +z)-(10, +z)-(16, +x)-(15, +x)-(14, +x)-(8, +x)-(12, +z)-(9, +z)-(5, +z)-(4, +z)-(6, +z)-(3, +z)-(2, +z)-(13, +z)-(7, +z)-(17, +z)-(1, +z)

**Hormiga: 3**  
Reorientaciones: 3

Salida:  
(11, +z)-(10, +z)-(16, +x)-(15, +x)-(14, +x)-(8, +x)-(12, +z)-(9, +z)-(5, +z)-(3, +z)-(6, +z)-(17, +z)-(1, +z)-(13, +z)-(7, +z)-(4, +z)-(2, +z)

**Hormiga: 4**  
Reorientaciones: 2

Salida:  
(11, +z)-(10, +z)-(16, +x)-(15, +x)-(14, +x)-(8, +x)-(12, +z)-(9, +z)-(5, +z)-(7, +z)-(17, +z)-(4, +z)-(13, +z)-(1, +z)-(6, +z)-(3, +z)-(2, +z)

**Hormiga: 5**  
Reorientaciones: 3

Salida:  
(11, +z)-(10, +z)-(16, +x)-(14, +x)-(15, +x)-(12, +z)-(9, +z)-(5, +z)-(4, +z)-(6, +z)-(7, +z)-(17, +z)-(1, +z)-(2, +z)-(13, +z)-(8, +x)-(3, +x)

**Hormiga: 6**  
Reorientaciones: 2

Salida:  
(11, +z)-(10, +z)-(16, +x)-(15, +x)-(14, +x)-(8, +x)-(12, +z)-(9, +z)-(5, +z)-(7, +z)-(17, +z)-(4, +z)-(1, +z)-(6, +z)-(2, +z)-(13, +z)-(3, +z)



# REPORTE DEL PLANIFICADOR DE SECUENCIA DE ENSAMBLE

Ing. Maybel Díaz Capote  
Universidad de las Ciencias Informáticas, La Habana, Cuba  
Fecha: sábado 5 de octubre de 2013

## DETALLES DEL REPORTE:

**Caso de estudio:** Bloque Corona  
**Reorientaciones:** sí  
**Cambios de herramientas:** no  
**Vértices:** 17  
**Aristas:** 136  
**Número total de hormigas:** 11

### Secuencias de ensamble:

#### Hormiga: 1

Reorientaciones: 2

Salida:

(11, +z)-(10, +z)-(16, +x)-(15, +x)-(14, +x)-(8, +x)-(12, +z)-(9, +z)-(5, +z)-(7, +z)-(17, +z)-(4, +z)-(6, +z)-(3, +z)-(2, +z)-(13, +z)-(1, +z)

#### Hormiga: 2

Reorientaciones: 3

Salida:

(11, +z)-(10, +z)-(16, +x)-(15, +x)-(14, +x)-(8, +x)-(12, +z)-(9, +z)-(5, +z)-(4, +z)-(6, +z)-(3, +z)-(2, +z)-(13, +z)-(7, -y)-(17, -y)-(1, -y)

#### Hormiga: 3

Reorientaciones: 3

Salida:

(11, +z)-(10, +z)-(16, +x)-(15, +x)-(14, +x)-(8, +x)-(12, +z)-(9, +z)-(5, +z)-(3, +z)-(6, +z)-(17, +z)-(1, +z)-(13, +z)-(7, +y)-(4, +y)-(2, +y)

#### Hormiga: 4

Reorientaciones: 2

Salida:

(11, +z)-(10, +z)-(16, +x)-(15, +x)-(14, +x)-(8, +x)-(12, +z)-(9, +z)-(5, +z)-(7, +z)-(17, +z)-(4, +z)-(13, +z)-(1, +z)-(6, +z)-(3, +z)-(2, +z)

#### Hormiga: 5

Reorientaciones: 3

Salida:

(11, +z)-(10, +z)-(16, +x)-(14, +x)-(15, +x)-(12, +z)-(9, +z)-(5, +z)-(4, +z)-(6, +z)-(7, +z)-(17, +z)-(1, +z)-(2, +z)-(13, +z)-(8, +x)-(3, +x)

#### Hormiga: 6

Reorientaciones: 2

Salida:

(11, +z)-(10, +z)-(16, +x)-(15, +x)-(14, +x)-(8, +x)-(12, +z)-(9, +z)-(5, +z)-(7, +z)-(17, +z)-(4, +z)-(1, +z)-(6, +z)-(2, +z)-(13, +z)-(3, +z)

**Hormiga: 7**

Reorientaciones: 2

Salida:

(8, -x)-(14, -x)-(15, -x)-(16, -x)-(5, +z)-(4, +z)-(6, +z)-(7, +z)-(17, +z)-(2, +z)-(13, +z)-(1, +z)-(9, -z)-(10, -z)-(11, -z)-(12, -z)-(3, -z)

**Hormiga: 8**

Reorientaciones: 2

Salida:

(8, -x)-(14, -x)-(15, -x)-(16, -x)-(5, +z)-(4, +z)-(6, +z)-(7, +z)-(17, +z)-(2, +z)-(13, +z)-(1, +z)-(3, -z)-(9, -z)-(10, -z)-(12, -z)-(11, -z)

**Hormiga: 9**

Reorientaciones: 3

Salida:

(8, -x)-(14, -x)-(15, -x)-(16, -x)-(10, -z)-(11, -z)-(9, +z)-(5, +z)-(4, +z)-(7, +z)-(17, +z)-(1, +z)-(6, +z)-(3, +z)-(2, +z)-(13, +z)-(12, -x)

**Hormiga: 10**

Reorientaciones: 2

Salida:

(8, -x)-(14, -x)-(15, -x)-(16, -x)-(5, +z)-(4, +z)-(6, +z)-(7, +z)-(17, +z)-(2, +z)-(13, +z)-(1, +z)-(3, -z)-(9, -z)-(10, -z)-(11, -z)-(12, -z)

**Hormiga: 11**

Reorientaciones: 2

Salida:

(11, +z)-(10, +z)-(16, +x)-(15, +x)-(14, +x)-(8, +x)-(12, +z)-(9, +z)-(5, +z)-(7, +z)-(17, +z)-(4, +z)-(1, +z)-(6, +z)-(3, +z)-(2, +z)-(13, +z)

**Secuencia de salida premiada****Reorientaciones: 2****Salida:**

**(8, -x)-(15, -x)-(14, -x)-(16, -x)-(10, -z)-(11, -z)-(12, +z)-(9, +z)-(5, +z)-(4, +z)-(7, +z)-(6, +z)-(13, +z)-(3, +z)-(17, +z)-(2, +z)-(1, +z)**