

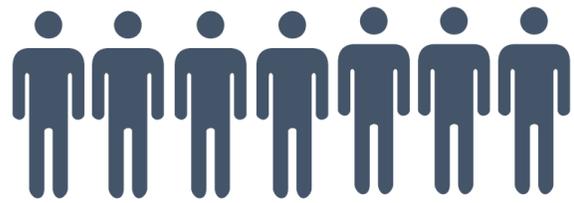
Desarrollo de una aplicación de gestión y monitorización de nodos de red

Sergio Gómez Bachiller
Operador del Servicio de Informática
Área de Apoyo a la Docencia y a la Investigación
Universidad de Córdoba

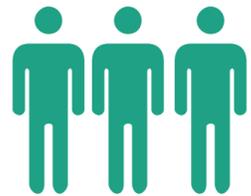
¿Quién soy y de dónde vengo?



Área de Apoyo a la Docencia y a la Investigación



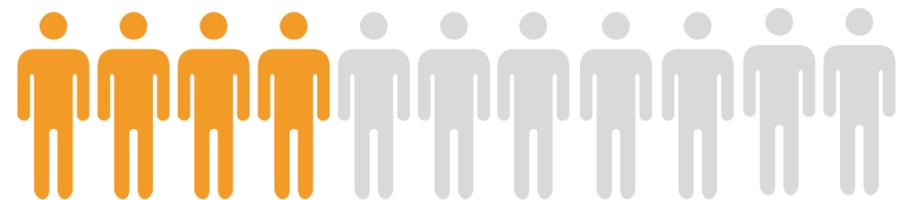
Área de Sistemas



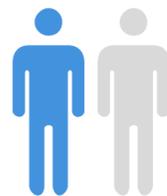
Área de Comunicaciones



Área de Aplicaciones



Área de Atención al Usuario



Jefe de Servicio + Administrativo

Estado del arte

Herramientas previas

Aplicación en Microsoft Access

Desarrollo propio

20 años de antigüedad

Problemas inherentes al tiempo y a la arquitectura

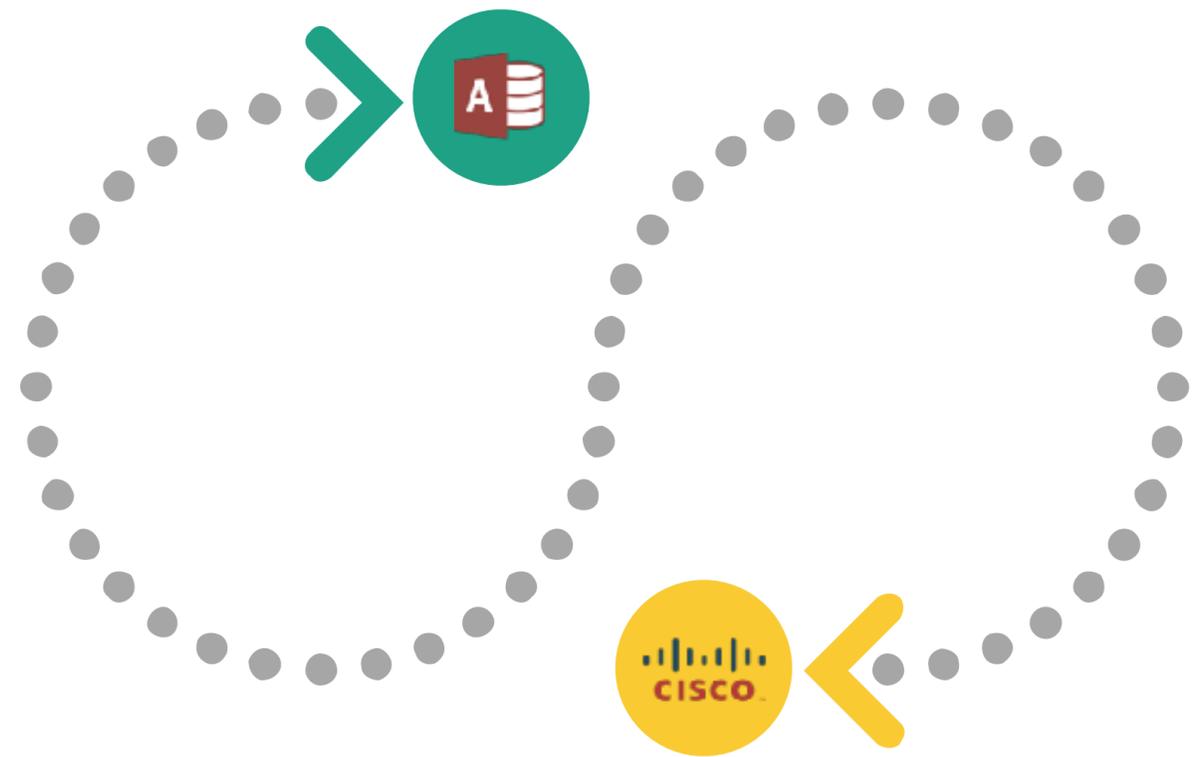
Cisco Works

Software privativo

Sustituido por Cisco Prime en el año 2011

Sin soporte desde el año 2015

Problemas por falta de actualizaciones



Metodologías clásicas

¿Qué salió mal?



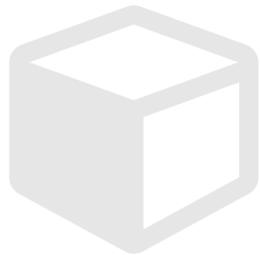
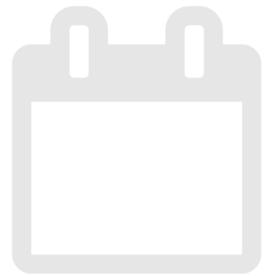
Demasiado amplio

No hay suficiente personal

No hay producto hasta el final

Metodologías ágiles

Programación



Reducir objetivos

Definir los MVP

Pruebas

Objetivos



Usuarios

Gestión e identificación de usuarios



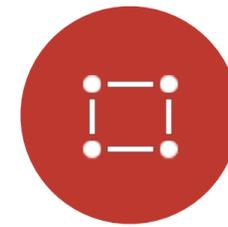
Permisos

Gestión de roles y seguridad



Organización

Gestión de unid. organizativas, edificios y racks



Nodos

Gestión de elementos de la red



Esquema

Gestión de los datos de la red



Auditoría

Registro y consulta de conexiones

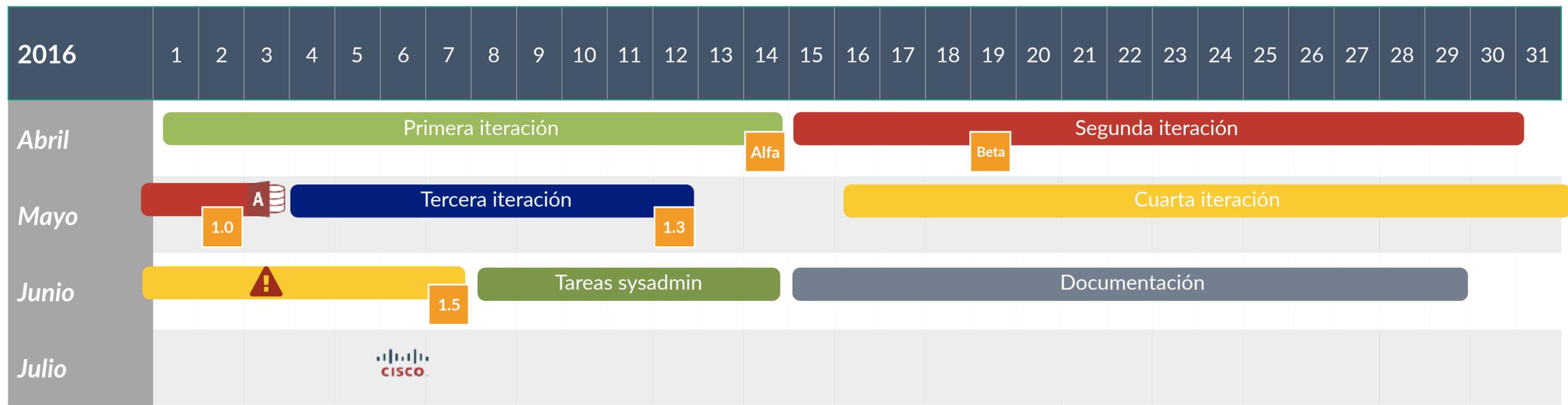


Monitorización

Control de estados y envío de alertas

Desarrollo

Planificación



Metodologías ágiles

Pruebas



Behaviour-driven Development (BDD)

Metodologías ágiles

Pruebas



Iteración

Iteración

Iteración

Restricciones

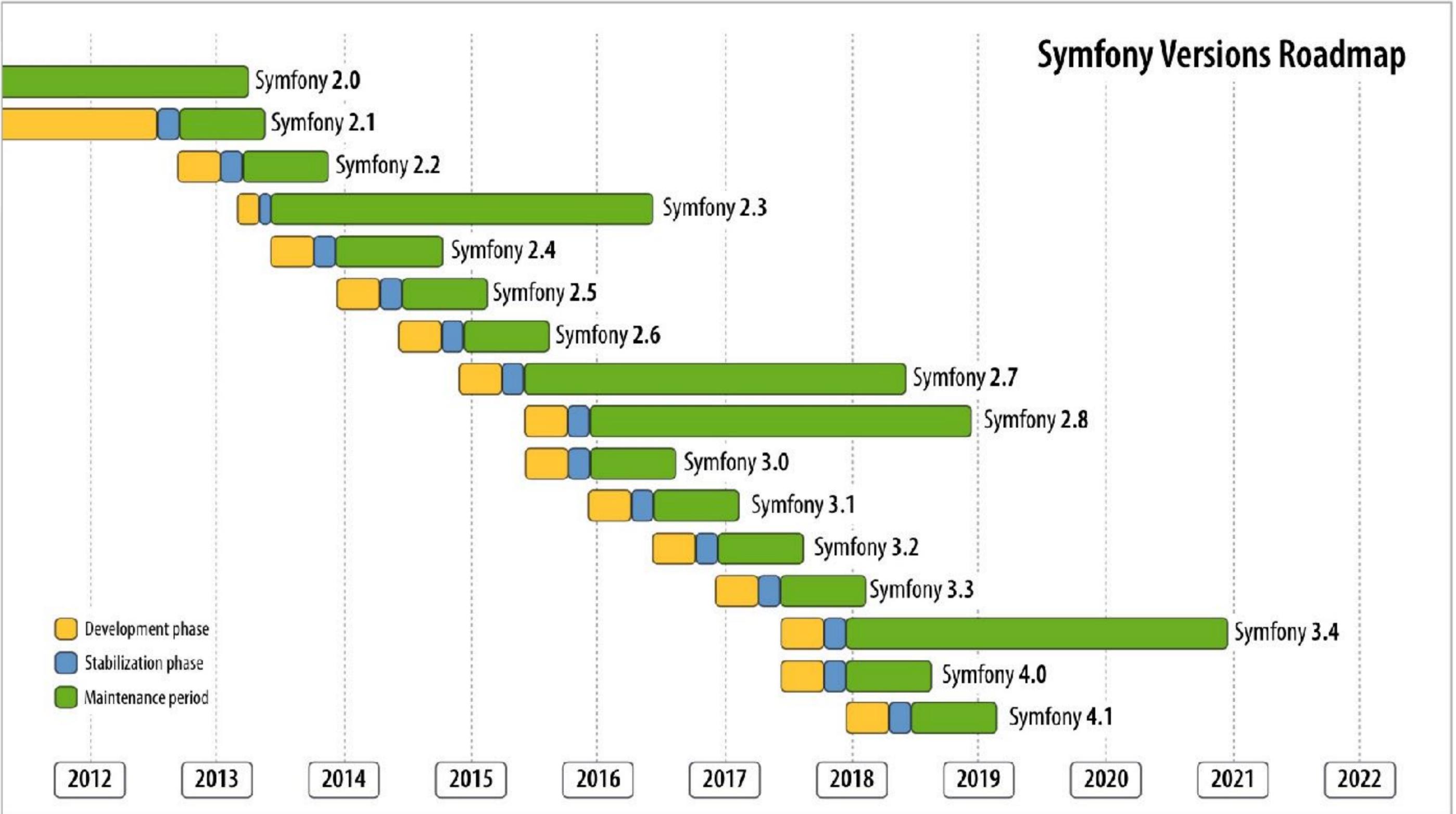
Factores dato



Restricciones

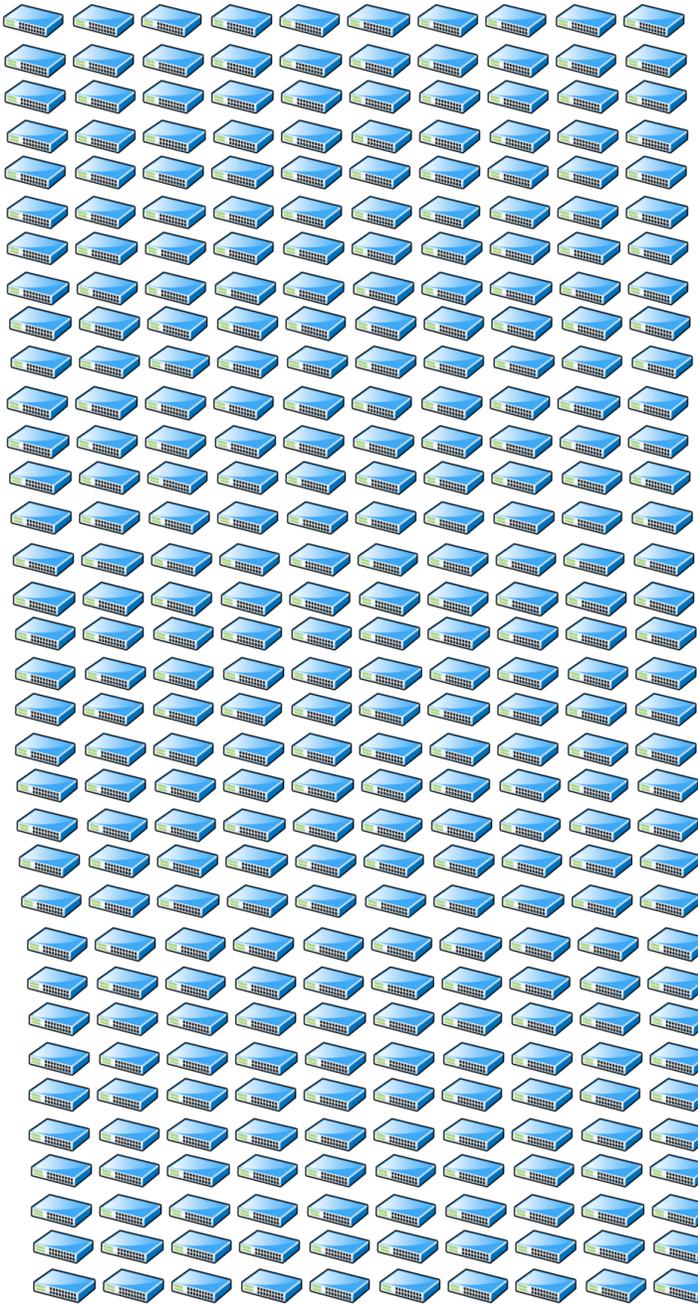
Factores estratégicos





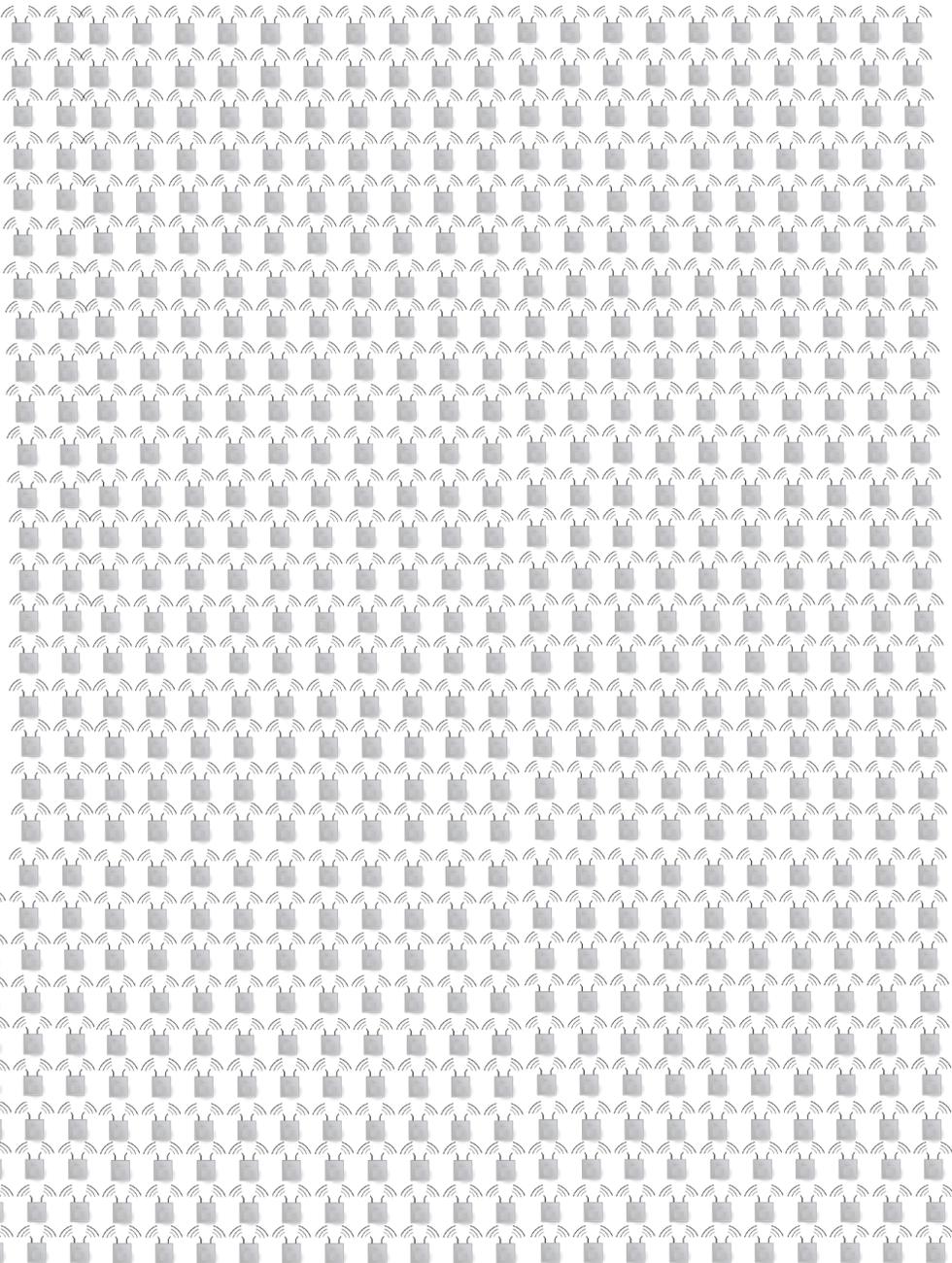
El problema

Catalogar y gestionar



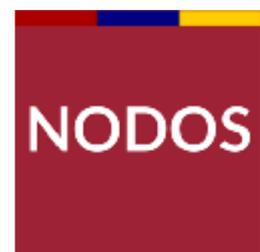
450

750



La solución

Funcionamiento



RabbitMQ™



Desarrollo

Gestión

Control de versiones + 

 + Documentación

Gestión de tareas + 

 + Estadísticas



<https://gitlab.uco.es/si/nodos>

Desarrollo orientado a comportamiento

Pruebas



Pruebas de unidad

PHPSpec

```
class NodeEventLogMailerSpec extends ObjectBehavior
{
  function let(...) {}
  function it_is_initializable(...) {}
  function it_sends_email(...) {}
  function it_not_sends_email_if_logs_are_empty(...) {}
}
```

Pruebas de unidad

PHPSpec

```
class NodeEventLogMailerSpec extends ObjectBehavior
{
  function it_sends_email(
    \Swift_Mailer $mailer,
    EngineInterface $engine,
    NodeEventLog $nodeEventLog
  )
  {
    $nodeEventLog->getAction()->willReturn(NodeEventLog::UP);

    $engine->render('NodeEventLog/email.html.twig', [
      'nodos_down' => [],
      'nodos_up' => [$nodeEventLog],
    ])->shouldBeCalled();

    $mailer->send(Argument::type('\Swift_Message'))->shouldBeCalled();

    $this->notifyTo(['email@mail.com'], [$nodeEventLog]);
  }
}
```

Pruebas de unidad

PHPSpec

```
class UserNodeVoterSpec extends ObjectBehavior
{
    function let(...) {}
    function it_is_initializable(...) {}
    function it_is_a_voter(...) {}
    function it_votes_access_granted_to_edit_to_faculty_staff(...) {}
    function it_votes_access_denied_to_edit_if_operator_not_belong_to_building(...)
    function it_votes_access_granted_to_view_to_faculty_staff(...) {}
    function it_votes_access_denied_to_show_if_operator_not_belong_to_building(...)
    function it_votes_access_denied_if_operator_have_no_buildings(...) {}
    function it_votes_access_abstain_to_other_actions(...) {}
    function it_votes_access_abstain_to_other_objects(...) {}
    function it_votes_access_granted_to_other_roles(...) {}
}
```

Pruebas de unidad

PHPSpec

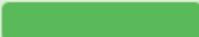
```
class UserNodeVoterSpec extends ObjectBehavior
{
  function let(...)
  {
    $token->getUser()->willReturn($user);
    $user->hasRole('ROLE_FACULTY')->willReturn(true);
    $user->getUsername()->willReturn('johndoe');
    $networkNode->getBuilding()->willReturn($building);
    $building->getOperators()->willReturn([$operator]);
  }

  function it_votes_access_denied_to_edit_if_operator_not_belong_to_building(...)
  {
    $operator->getUsername()->willReturn('johnnydoe');
    $this->vote($token, $networkNode, ['ROLE_APP_ADMIN_USER_NODE_EDIT'])
      ->shouldReturn(Voter::ACCESS_DENIED)
    ;
  }
}
```

Desarrollo

Cobertura de las pruebas

/home/sergio/Developer/nodos/nodos/src / AppBundle / (Dashboard)

	Code Coverage								
	Lines			Functions and Methods			Classes and Traits		
Total		99.15%	930 / 938		99.18%	365 / 368		96.10%	74 / 77
📁 Command		100.00%	66 / 66		100.00%	9 / 9		100.00%	4 / 4
📁 Consumer		100.00%	37 / 37		100.00%	4 / 4		100.00%	2 / 2
📁 Controller		100.00%	19 / 19		100.00%	6 / 6		100.00%	4 / 4
📁 Entity		100.00%	310 / 310		100.00%	219 / 219		100.00%	15 / 15
📁 Event		100.00%	8 / 8		100.00%	6 / 6		100.00%	2 / 2
📁 EventListener		94.34%	50 / 53		88.89%	16 / 18		88.89%	2 / 4
📁 Factory		100.00%	56 / 56		100.00%	22 / 22		100.00%	9 / 9
📁 Form		100.00%	17 / 17		100.00%	7 / 7		100.00%	2 / 2
📁 Repository		96.32%	131 / 136		96.15%	25 / 26		93.75%	15 / 16
📁 Security		100.00%	33 / 33		100.00%	9 / 9		100.00%	4 / 4
📁 Services		100.00%	103 / 103		100.00%	17 / 17		100.00%	7 / 7
📁 Twig		100.00%	13 / 13		100.00%	5 / 5		100.00%	1 / 1
📁 Utils		100.00%	87 / 87		100.00%	20 / 20		100.00%	5 / 5
📄 AppBundle.php		100.00%	0 / 0		100.00%	0 / 0		100.00%	1 / 1
📄 NodosEvents.php		100.00%	0 / 0		100.00%	0 / 0		100.00%	1 / 1

Legend

Low: 0% to 50% Medium: 50% to 90% High: 90% to 100%

Generated by PHP_CodeCoverage 3.3.3 using PHPDBG 7.0.6-12+donate.sury.org~xenial+4 at Tue Jun 7 11:30:43 CEST 2016.

Pruebas de integración

Behat

login.feature

Feature: Managing node connections

In order to know the devices connected to our network

As a network administrator

I want to register all device connections on every network port

Pruebas de integración

Behat

Background:

Given the next network nodes:

name
switch01

And the next ports on network node "switch01":

index	name	alias
1	Fa0/1	Port 1
2	Fa0/2	Port 2

And the next entries on the arp table:

mac	ip
00:11:22:33:44:55	192.168.0.1

And the next network connections:

node	port	mac	ip
switch01	Fa0/1	00:11:22:33:44:55	192.168.0.1

Pruebas de integración

Behat

Scenario: Register a new IP in a new port

Given the next network connections:

node	port	mac	ip
switch01	Fa0/2	00:11:22:33:44:66	192.168.0.2

Then there are 1 open connection on "switch01" port "Fa0/1"

And there are 1 open connection on "switch01" port "Fa0/2"

Demo

Gestión de Nodos de la Universidad de Córdoba



Conclusiones

Estadísticas



Líneas de código



Líneas de código de pruebas



Commits

Conclusiones

Futuras mejoras



Desarrollo de una aplicación de gestión y monitorización de nodos de red

Sergio Gómez Bachiller
Operador del Servicio de Informática
Área de Apoyo a la Docencia y a la Investigación
Universidad de Córdoba