

# 1 User requirements

Each user requirement is uniquely identified by an ID, which follows the format *UR-YY-XX*, where *YY* identifies the type of the requirement, either a capacity (*CA*) or a restriction (*RE*); and *XX* identifies the sequential number of the requirement within that type, starting at *01*.

Table 1 provides the template used for the specification of the requirements, including the description of each attribute.

Table 1: User requirement template

UR-YY-XX	
<b>Description</b>	Detailed description of the requirement.
<b>Necessity</b>	Priority of the requirement for the user ( <i>Essential</i> , <i>Convenient</i> or <i>Optional</i> ).
<b>Priority</b>	Priority of the requirement for the developer ( <i>High</i> , <i>Medium</i> or <i>Low</i> ).
<b>Stability</b>	Indicates the requirement's variability trough the development process ( <i>Constant</i> , <i>Constant</i> or <i>Very unstable</i> ).
<b>Verifiability</b>	Ability to test the validity of the requirement ( <i>High</i> , <i>Medium</i> or <i>Low</i> ).

Table 2: Requirement UR-CA-01

UR-CA-01	
<b>Description</b>	Mi carro me lo robaron estando de romería.
<b>Necessity</b>	Essential
<b>Priority</b>	Low
<b>Stability</b>	Constant
<b>Verifiability</b>	Medium

Table 3: Requirement UR-CA-02

<b>UR-CA-02</b>	
<b>Description</b>	Mi carro me lo robaron anoche cuando dormía.
<b>Necessity</b>	Convenient
<b>Priority</b>	High
<b>Stability</b>	Constant
<b>Verifiability</b>	High

Table 4: Requirement UR-RE-01

<b>UR-RE-01</b>	
<b>Description</b>	Mi carro me lo robaron estando de romería.
<b>Necessity</b>	Optional
<b>Priority</b>	Medium
<b>Stability</b>	Inconstant
<b>Verifiability</b>	Medium

Table 5: Requirement UR-RE-02

<b>UR-RE-02</b>	
<b>Description</b>	Mi carro me lo robaron anoche cuando dormía.
<b>Necessity</b>	Convenient
<b>Priority</b>	Low
<b>Stability</b>	Very unstable
<b>Verifiability</b>	Low

Requirement UR-CA-01 is very cool.

## 2 System requirements

Each software requirement is uniquely identified by an ID, which follows the format *UR-YY-XX*, where *YY* identifies the type of the requirement, either functional (*FN*) or non-functional (*NF*); and *XX* identifies the sequential number of the requirement within that type, starting at *01*.

Table 6 provides the template used for the specification of the requirements, including the description of each attribute.

Table 6: Software requirement template

<b>UR-YY-XX</b>	
<b>Description</b>	Detailed description of the requirement.
<b>Necessity</b>	Priority of the requirement for the user ( <i>Essential</i> , <i>Convenient</i> or <i>Optional</i> ).
<b>Priority</b>	Priority of the requirement for the developer ( <i>High</i> , <i>Medium</i> or <i>Low</i> ).
<b>Stability</b>	Indicates the requirement's variability trough the development process ( <i>Constant</i> , <i>Constant</i> or <i>Very unstable</i> ).
<b>Verifiability</b>	Ability to test the validity of the requirement ( <i>High</i> , <i>Medium</i> or <i>Low</i> ).
<b>Origin</b>	User requirements that derived this requirement.

Table 7: Requirement SR-FN-01

<b>SR-FN-01</b>	
<b>Description</b>	¿Dónde estará mi carro?
<b>Necessity</b>	Convenient
<b>Priority</b>	Low
<b>Stability</b>	Constant
<b>Verifiability</b>	High
<b>Origin</b>	UR-CA-01, UR-CA-02

Table 8: Requirement SR-FN-02

<b>SR-FN-02</b>	
<b>Description</b>	¿Dónde estará mi carro?
<b>Necessity</b>	Convenient
<b>Priority</b>	Low
<b>Stability</b>	Constant
<b>Verifiability</b>	High
<b>Origin</b>	UR-CA-01, UR-CA-02

Table 9: Requirement SR-NF-01

<b>SR-NF-01</b>	
<b>Description</b>	¿Dónde estará mi carro?
<b>Necessity</b>	Optional
<b>Priority</b>	High
<b>Stability</b>	Inconstant
<b>Verifiability</b>	Medium
<b>Origin</b>	UR-RE-01

Table 10: Requirement SR-NF-02

<b>SR-NF-02</b>	
<b>Description</b>	¿Dónde estará mi carro?
<b>Necessity</b>	Essential
<b>Priority</b>	High
<b>Stability</b>	Very unstable
<b>Verifiability</b>	Low
<b>Origin</b>	UR-RE-02

Requirement SR-NF-02 is trully atomic.

### 3 Use cases

Each use case is uniquely identified by an ID, which follows the format *UC-YY-XX*, where *XX* identifies the sequential number of the use case, starting at *01*.

Table 11 provides the template used for the specification of the use case, including the description of each attribute.

Table 11: Use case template

UC-YY-XX	
<b>Name</b>	Brief description of the use case.
<b>Actors</b>	External agent that executes the use case.
<b>Objetive</b>	The use case's purpose.
<b>Description</b>	Steps that the external agent must take to execute the use case.
<b>Pre-condition</b>	Conditions that must be fulfilled <i>before</i> executing the use case.
<b>Post-condition</b>	Conditions that must be fulfilled <i>after</i> executing the use case.

Table 12: Use case UC-01

UC-01	
<b>Name</b>	Recuperar el carro.
<b>Actors</b>	Manolo
<b>Objetive</b>	Quiere recuperar su carro.
<b>Description</b>	<ol style="list-style-type: none"><li>1. Lo busca</li><li>2. Lo encuentra</li><li>3. Profit</li></ol>
<b>Pre-condition</b>	Quiere recuperar su carro.
<b>Post-condition</b>	Se lo han robao.

I think Use case UC-01 really shows what's going on in the user's head.

## 4 Components

Table 13 provides the template used for the specification of the components, including the description of each attribute.

Table 13: Component template

Name	
<b>Role</b>	Component's function in the system.
<b>Dependencies</b>	Components that depend on this component.
<b>Description</b>	Explanation of the functioning of the component.
<b>Data</b>	Input ( <i>in</i> ) and output ( <i>out</i> ) data of the component.
<b>Origin</b>	Software requirements that derived the component.

Table 14: Component 'Carro'

Carro	
<b>Role</b>	Ser robado
<b>Dependencies</b>	N/A
<b>Description</b>	Es mu bonito, pero propenso a ser robado.
<b>Data</b>	<ul style="list-style-type: none"><li>• <b>in:</b> gente</li><li>• <b>out:</b> más gente</li></ul>
<b>Origin</b>	SR-FN-01

I really did a good job on Component 'Carro'.

## 5 Tests

Each test is uniquely identified by an ID. The ID follows the format *YYY-XX*, where *XX* identifies the sequential number of the test case, starting at *01*, and *YYY* represents the type, either *VET* (verification) or *VAT* (validation).

Table15 provides the template used for the specification of the test case, including the description of each attribute.

Table 15: Test template

YYY-XX	
<b>Description</b>	Test description.
<b>Preconditions</b>	Conditions that must be fulfilled in order to perform the test.
<b>Procedure</b>	Description of the steps to take in order to perform the test.
<b>Postconditions</b>	Conditions that must be fulfilled after performing the test in order to pass.
<b>Origin</b>	Requirements that originated this test.
<b>Evaluation</b>	Result of the test ( <i>OK</i> or <i>Error</i> ).

Table 16: Test VET-01

VET-01	
<b>Description</b>	Verificar que mi carro verdaderamente ha desapareció.
<b>Preconditions</b>	El carro era mío.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. Me dicen que le quitaron</li> <li>2. los clavos que relucían</li> <li>3. creyendo que eran de oro</li> <li>4. de limpios que los tenía</li> </ol>
<b>Postconditions</b>	Desaparece.
<b>Origin</b>	SR-FN-02
<b>Evaluation</b>	OK

Table 17: Test VAT-01

<b>VAT-01</b>	
<b>Description</b>	Verificar que el carro merece la pena recuperarse.
<b>Preconditions</b>	Mi carro era muy querido.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. En mi carro gasté</li> <li>2. una fortuna</li> <li>3. y en mis noches de amor</li> <li>4. llevé la luna</li> </ol>
<b>Postconditions</b>	Lo he usado.
<b>Origin</b>	UR-RE-02
<b>Evaluation</b>	OK



Qué bonito es el Test VET-01.

## 6 Traceability matrixes

	<div> <div>UR-CA-01</div> <div>UR-CA-02</div> </div>	
SR-FN-01	•	•
SR-FN-02	•	•

Figure 1: Autogenerated traceability matrix, SR-FN to UR-CA

	<div> <div>UR-RE-01</div> <div>UR-RE-02</div> </div>	
SR-NF-01	•	
SR-NF-02		•

Figure 2: Autogenerated traceability matrix, SR-NF to UR-RE

	<div> <div>SR-FN-01</div> <div>SR-FN-02</div> </div>	
Carro	•	

Figure 3: Autogenerated traceability matrix, components to SR-FN

	SR-FN-01	SR-FN-02	SR-NF-01	SR-NF-02
VET-01		•		

Figure 4: Autogenerated traceability matrix, VET tests to SR

	UR-CA-01	UR-CA-02	UR-RE-01	UR-RE-02
VAT-01				•

Figure 5: Autogenerated traceability matrix, VAT tests to UR