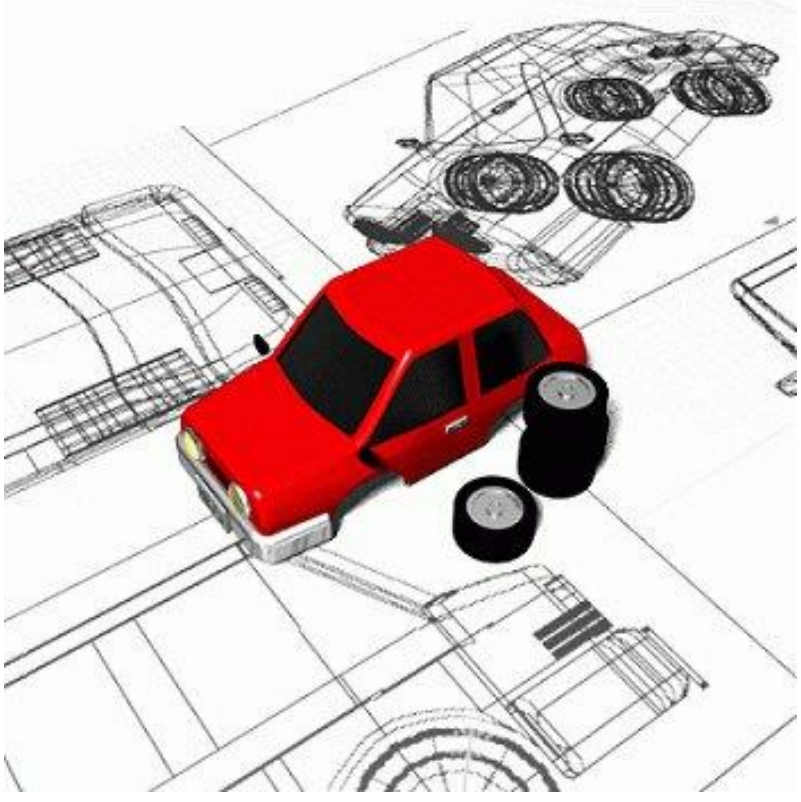
CONCEPTUAL SCHEMA

Carlos Mario Zapata J.

SPECIFICATION

SPECIFICATION



A FORMAL SPECIFICATION

- UML is “semi-formal”
- Source code is “formal”
- Why? According Van Lamsweerde, a specification is *formal* when it has:
  - Rules for determining grammar well-formedness (syntax)
  - Rules for interpreting sentences in a precise way inside the considered domain (semantics)
  - Rules for inferring useful information from the specification (proof theory)

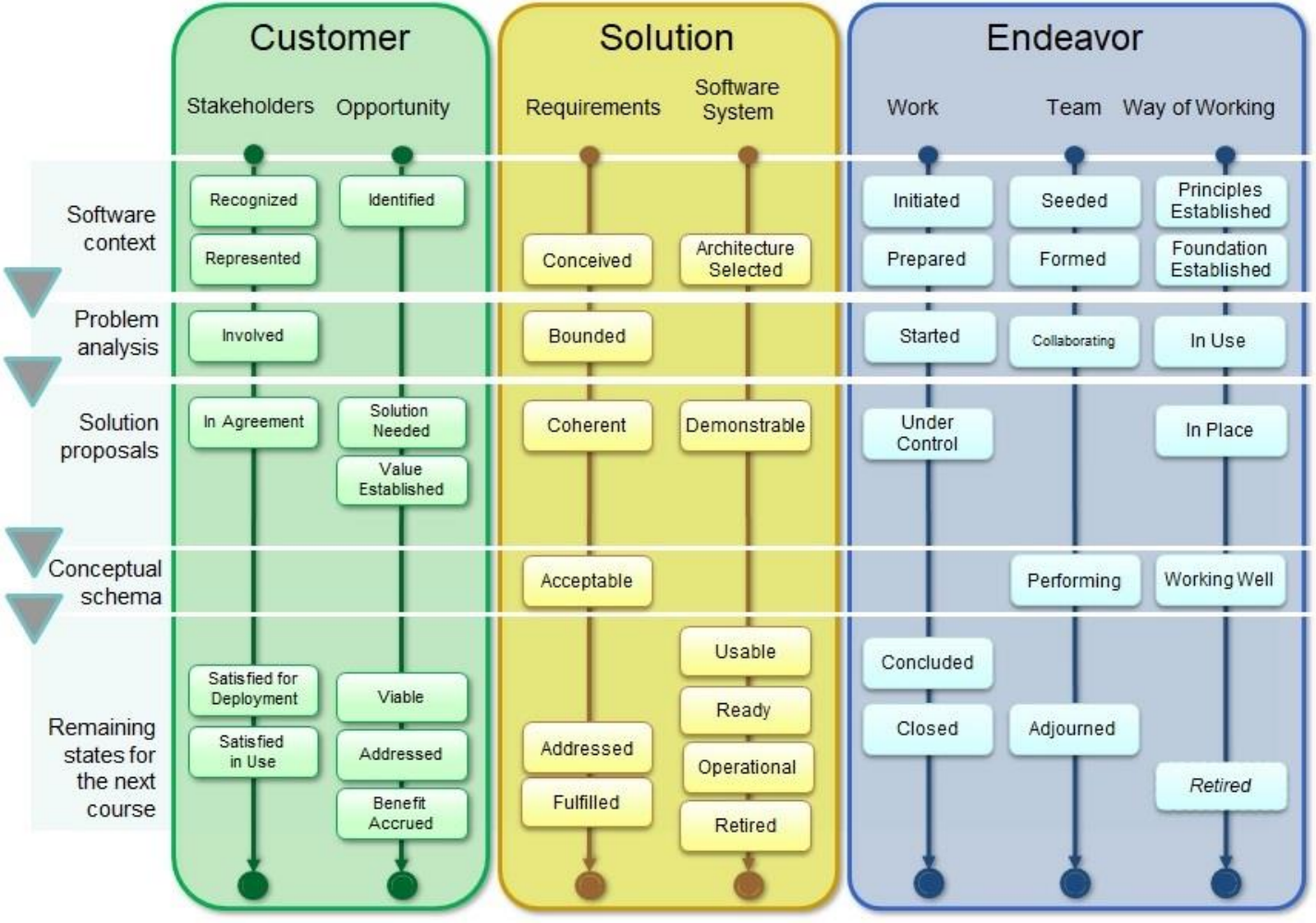
A FORMAL SPECIFICATION

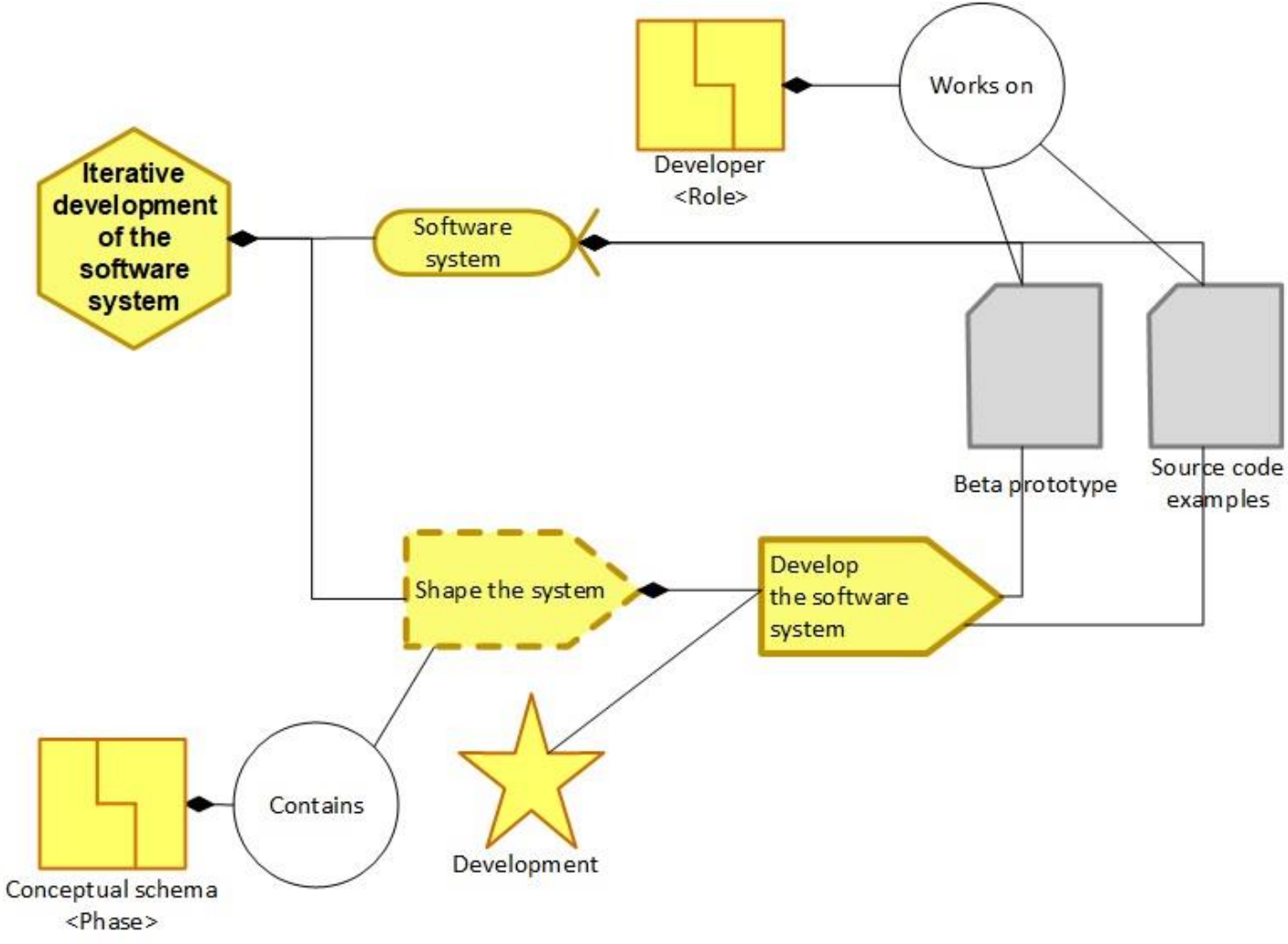
- What is a good specification?
  - Requirements adequacy
  - Internal consistency
  - Lack of ambiguity
  - Completeness
  - Minimality

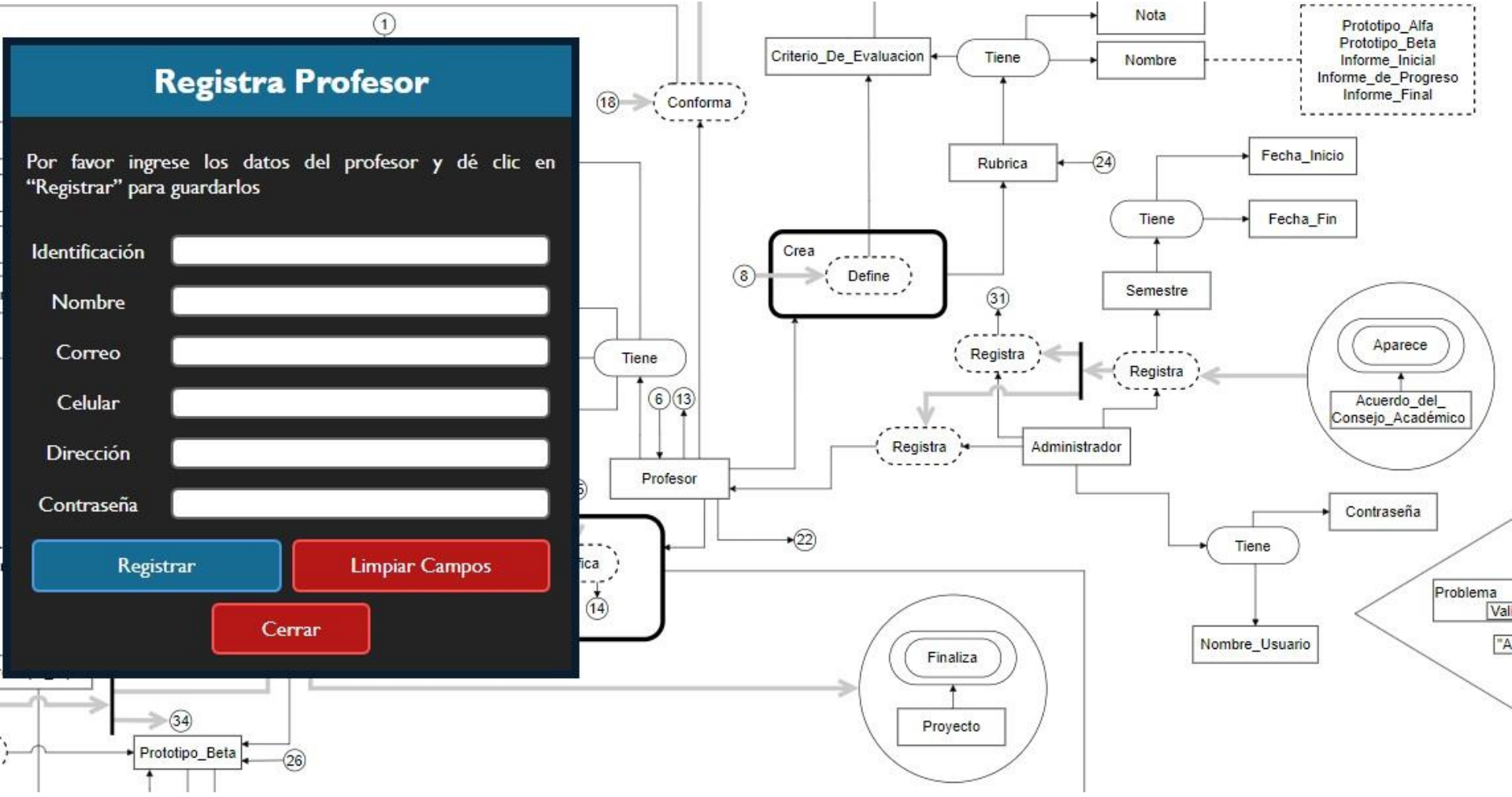
A FORMAL SPECIFICATION

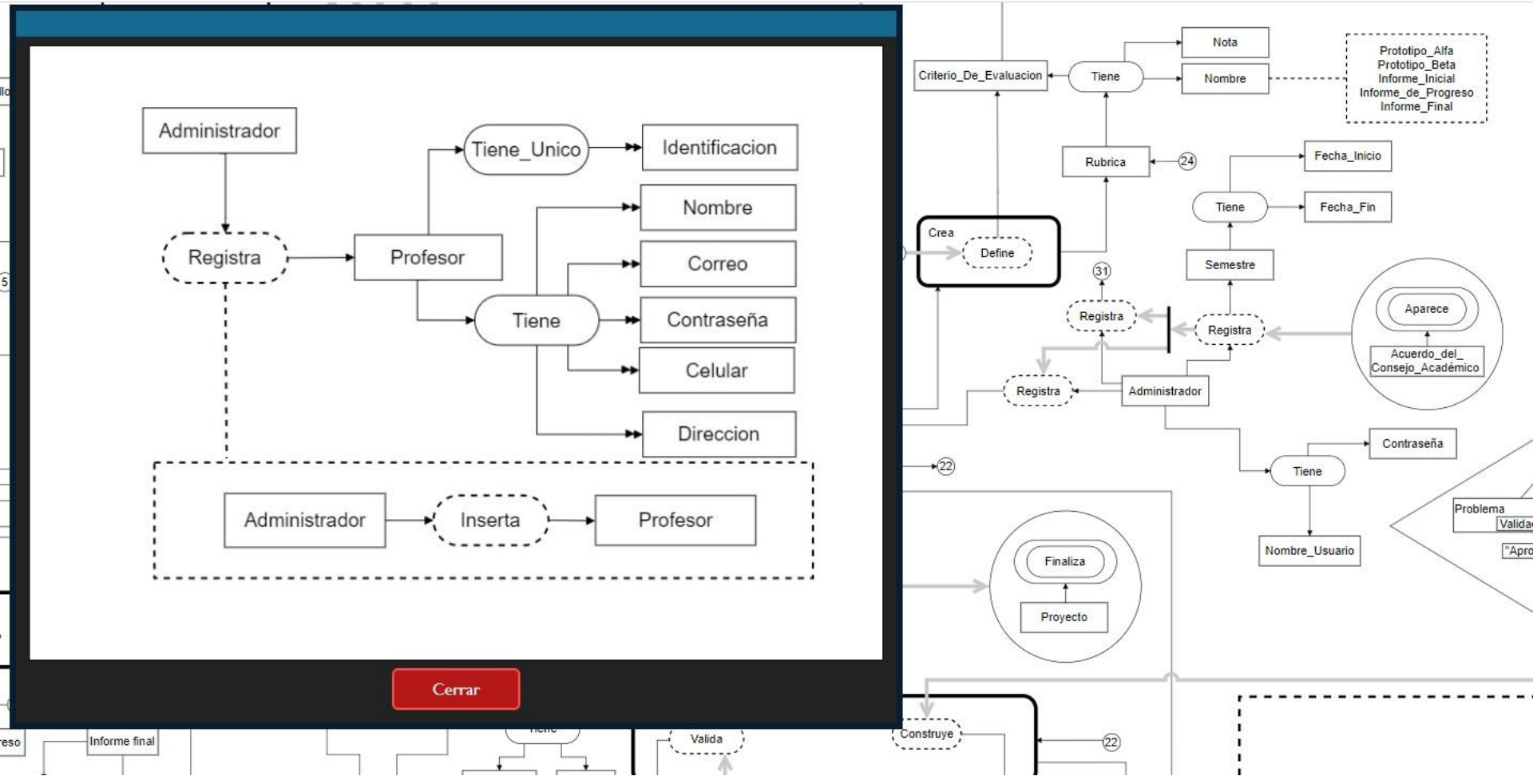
- Why specifications?
  - Design, validation, documentation, technical communication, reengineering, reuse
  - Identification of formulation problems
  - Automated proof devices (theorem provers)
  - Abstraction level decrease
  - Counter-example generation
  - Full scenario generation
  - Specification animation
  - Consistency and refinement

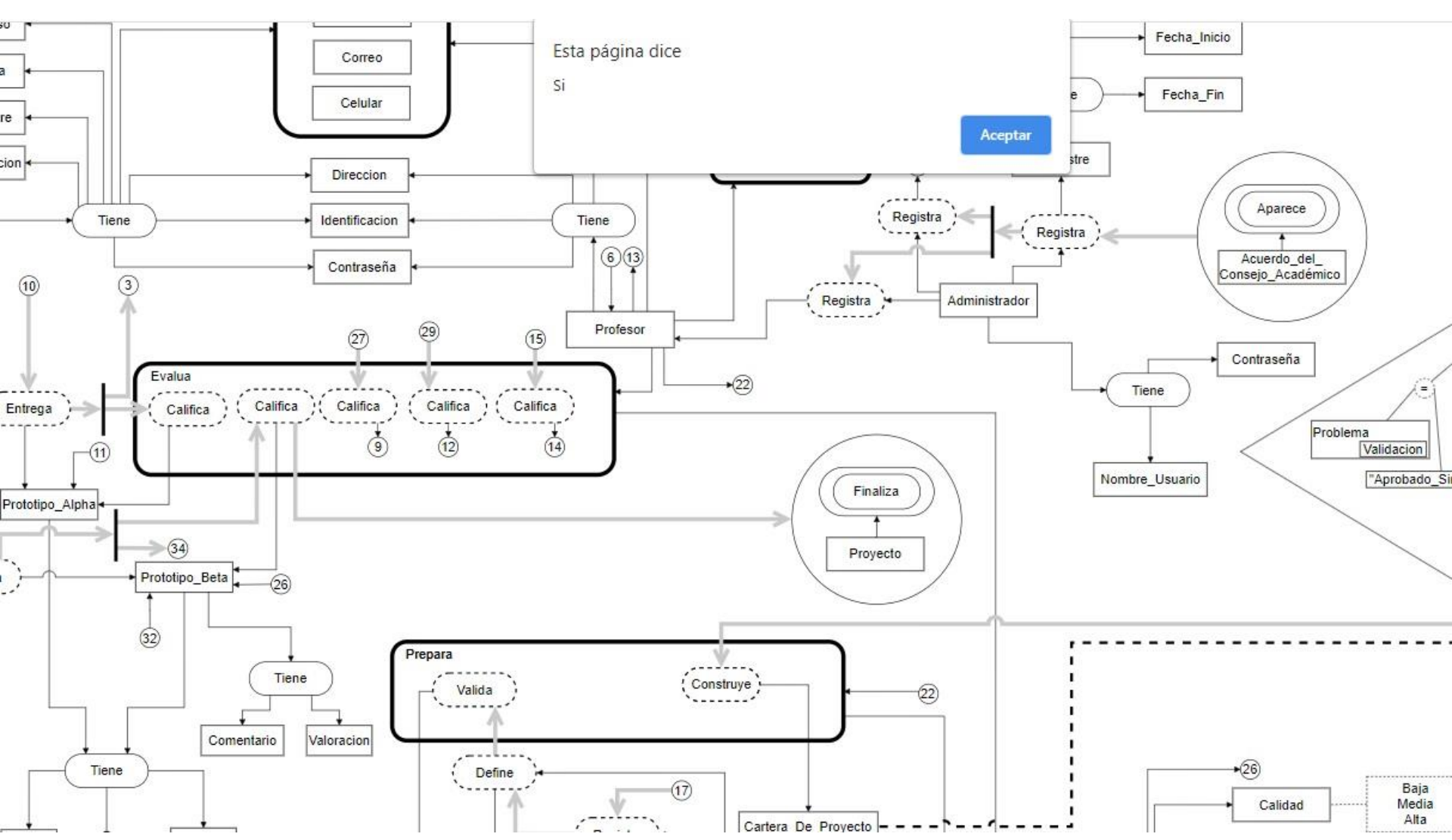
UNC-METHOD IN THE ESSENCE

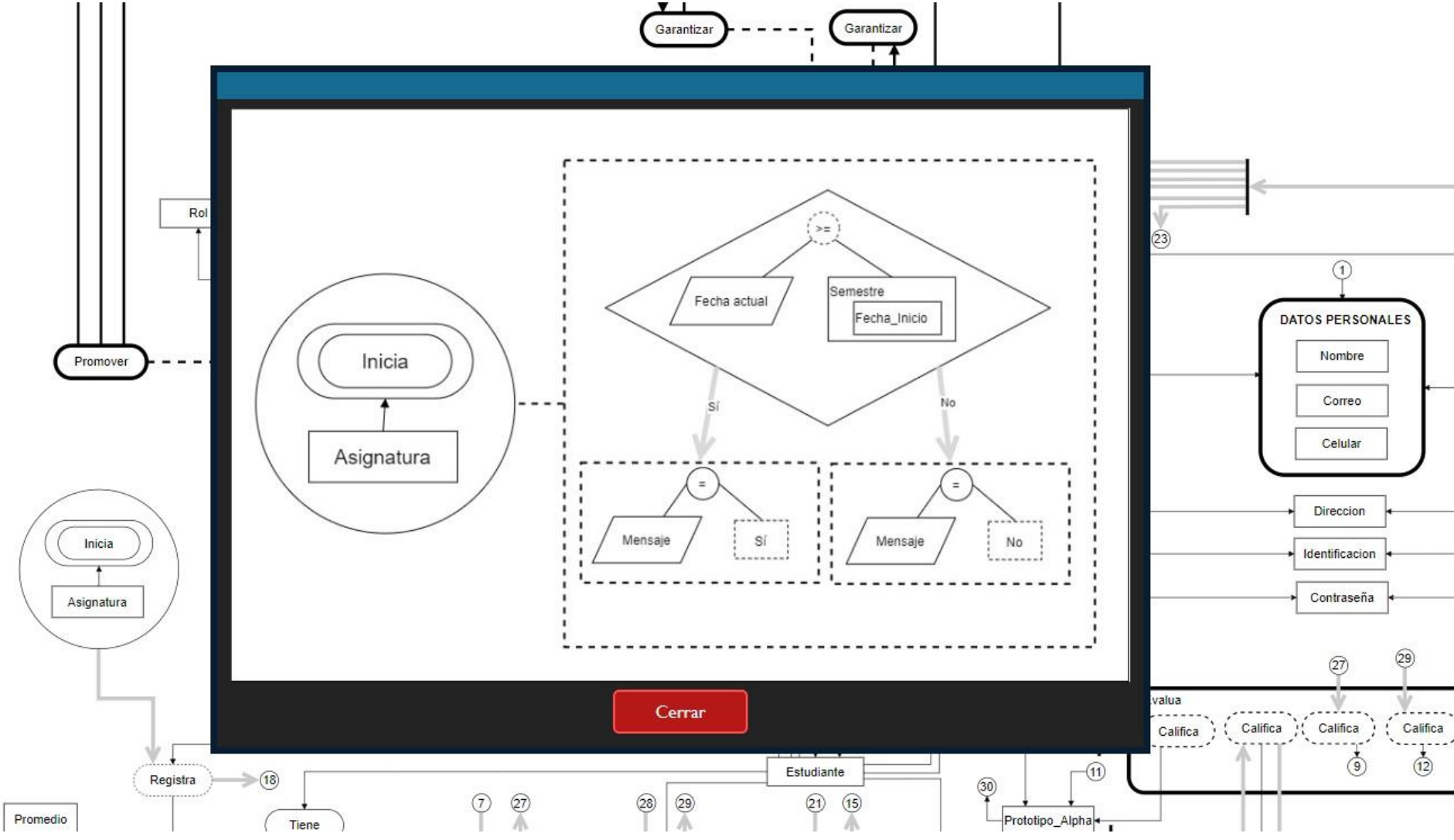


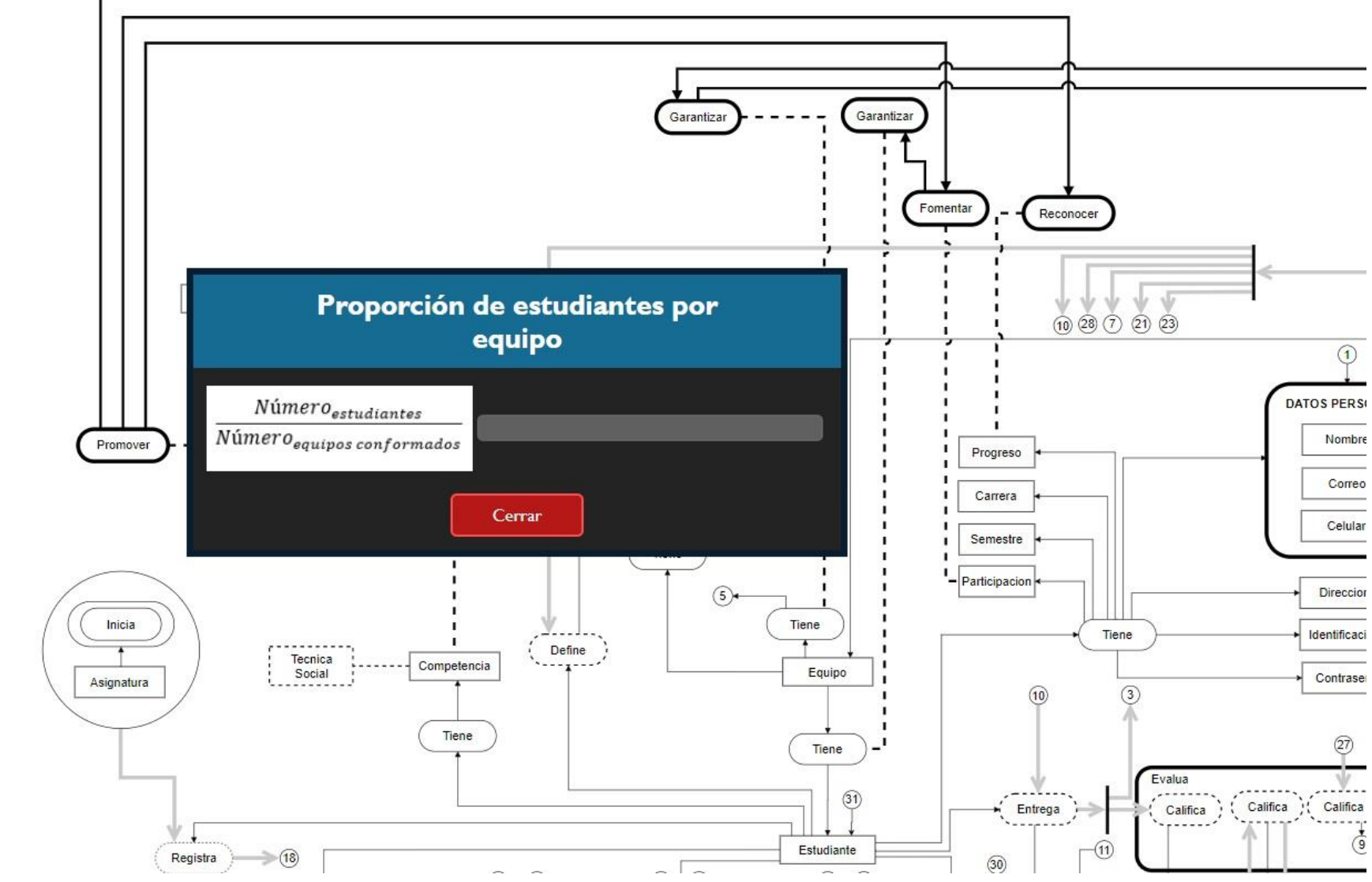

Iterative development of the software system
Software system
Developer
<Role>
Works on
Beta prototype
Source code examples
Shape the system
Develop the software system
Contains
Development
Conceptual schema
<Phase>

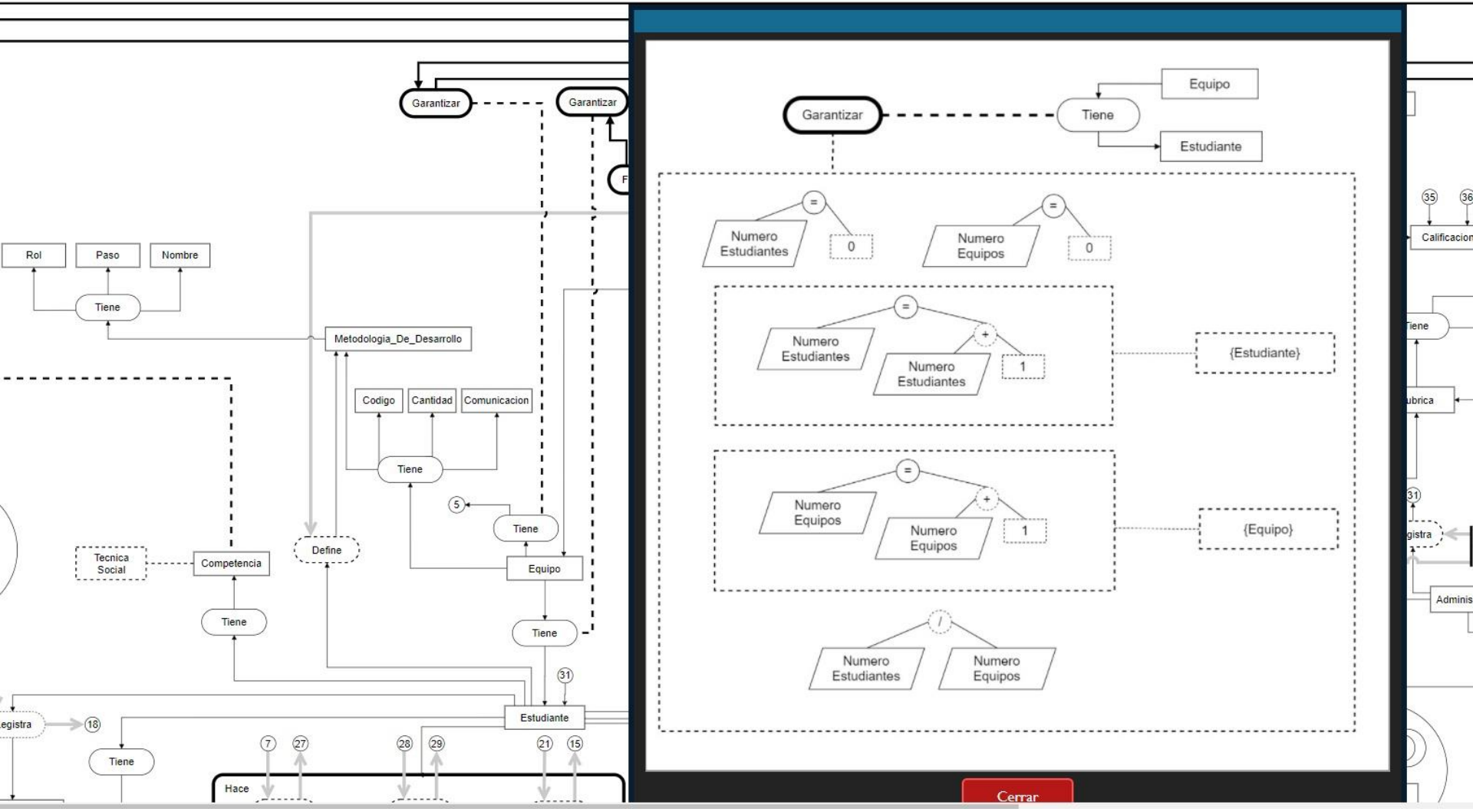

Registra Profesor
Por favor ingrese los datos del profesor y dé clic en "Registrar" para guardarlos
Identificación
Nombre
Correo
Celular
Dirección
Contraseña
Registrar
Limpiar Campos
Cerrar
Conforma
Criterio_De_Evaluacion
Tiene
Nota
Nombre
Prototipo_Alfa
Prototipo_Beta
Informe_Inicial
Informe_de_Progreso
Informe_Final
Rubrica
Fecha_Inicio
Fecha_Fin
Crea
Define
Semestre
Registra
Aparece
Acuerdo_del_ Consejo_Académico
Profesor
Administrador
Contraseña
Nombre_Usuario
Finaliza
Proyecto
Prototipo_Beta
Problema

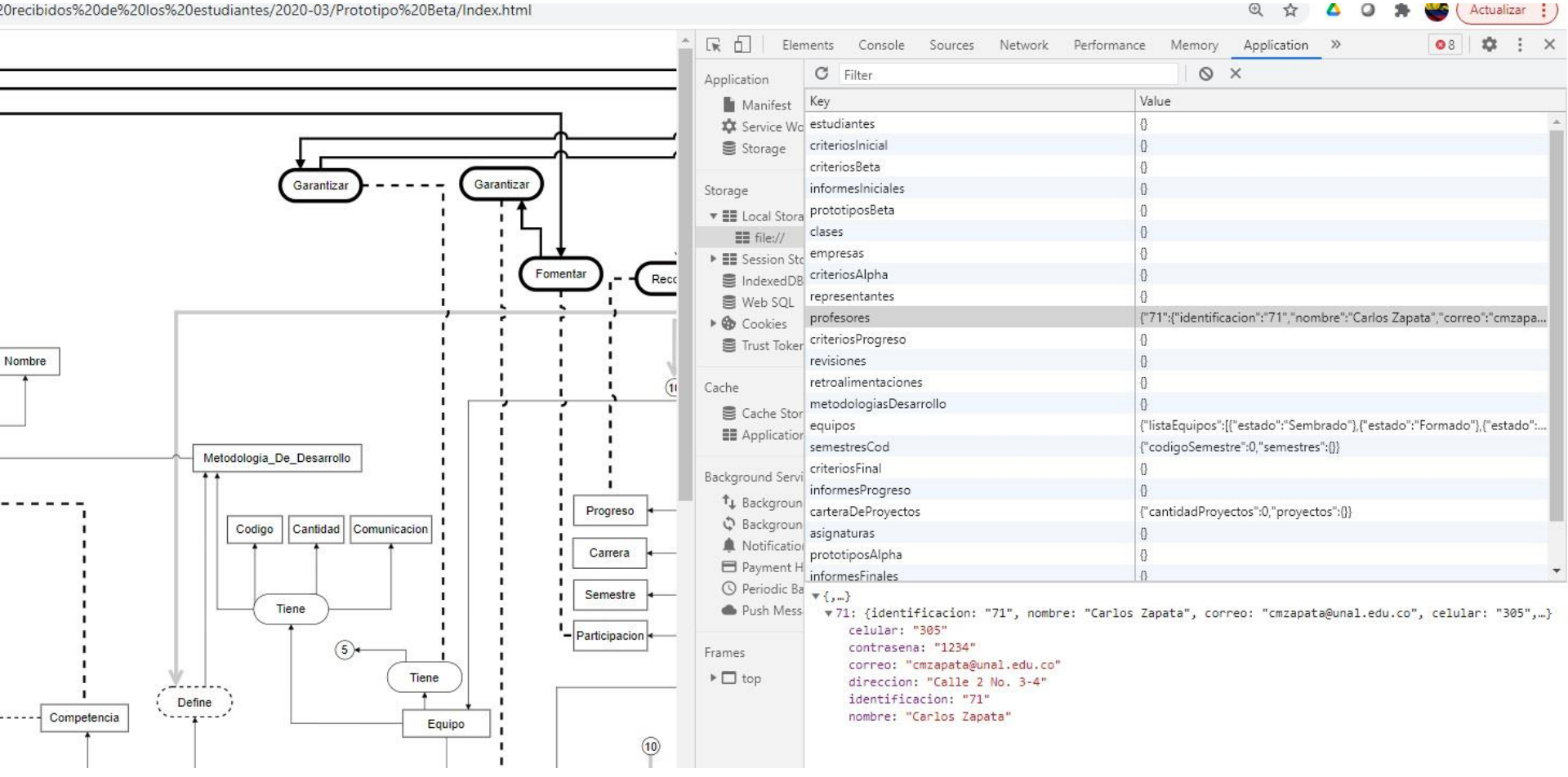

Administrador
Registra
Profesor
Tiene_Unico
Identificacion
Nombre
Correo
Tiene
Contraseña
Celular
Direccion
Administrador
Inserta
Profesor
Cerrar
Criterio_De_Evaluacion
Tiene
Nota
Nombre
Prototipo_Alfa
Prototipo_Beta
Informe_Inicial
Informe_de_Progreso
Informe_Final
Rubrica
24
Fecha_Inicio
Tiene
Fecha_Fin
Crea
Define
Semestre
31
Registra
Registra
Aparece
Acuerdo_del_
Consejo_Académico
Registra
Administrador
22
Contraseña
Tiene
Problema
Nombre_Usuario
Finaliza
Proyecto
Construye
22
Informe final
Valida


Esta página dice
Si
Aceptar
Correo
Celular
Direccion
Identificacion
Contraseña
Tiene
Tiene
Fecha_Inicio
Fecha_Fin
Registra
Registra
Registra
Aparece
Acuerdo_del_ Consejo_Académico
Administrador
Profesor
Contraseña
Tiene
Nombre_Usuario
Problema
Validacion
Evalua
Califica
Califica
Califica
Califica
Califica
Entrega
Prototipo_Alpha
Prototipo_Beta
Finaliza
Proyecto
Tiene
Comentario
Valoracion
Tiene
Prepara
Valida
Construye
Define
Cartera De Proyecto
Calidad
Baja
Media
Alta


Garantizar
Garantizar
Rol
Promover
Inicia
Asignatura
>=
Fecha actual
Semestre
Fecha_Inicio
Sí
No
=
Mensaje
Sí
=
Mensaje
No
Cerrar
Inicia
Asignatura
Registra
18
Promedio
Tiene
Estudiante
7
27
28
29
21
15
30
11
Prototipo_Alpha
23
1
DATOS PERSONALES
Nombre
Correo
Celular
Direccion
Identificacion
Contraseña
27
29
valua
Califica
Califica
Califica
Califica
9
12


Garantizar
Garantizar
Fomentar
Reconocer
Promover
Proporción de estudiantes por equipo
$\frac{Número_{estudiantes}}{Número_{equipos\ conformados}}$
Cerrar
10 28 7 21 23
1
DATOS PERS
Nombre
Correo
Celular
Progreso
Carrera
Semestre
Participacion
Direccion
Identificaci
Contrase
Tiene
Inicia
Asignatura
Tecnica Social
Competencia
Define
5
Tiene
Equipo
Tiene
Tiene
10
3
27
Evalua
Entrega
Califica
Califica
Califica
31
Registra
18
Estudiante
11
30
9


Garantizar
Garantizar
Rol
Paso
Nombre
Tiene
Metodologia_De_Desarrollo
Codigo
Cantidad
Comunicacion
Tiene
5
Tiene
Tecnica Social
Competencia
Define
Equipo
Tiene
Tiene
31
Estudiante
18
Tiene
7
27
28
29
21
15
Hace
Garantizar
Equipo
Tiene
Estudiante
=
Numero Estudiantes
0
=
Numero Equipos
0
=
Numero Estudiantes
+
Numero Estudiantes
1
{Estudiante}
=
Numero Equipos
+
Numero Equipos
1
{Equipo}
/
Numero Estudiantes
Numero Equipos
Cerrar
35
36
Calificacion
Tiene
ubrica
31
gistra
Adminis


20recibidos%20de%20los%20estudiantes/2020-03/Prototipo%20Beta/Index.html
Actualizar
Garantizar
Garantizar
Fomentar
Nombre
Metodologia_De_Desarrollo
Codigo
Cantidad
Comunicacion
Tiene
Progreso
Carrera
Semestre
Participacion
5
Tiene
Define
Competencia
Equipo
10
Elements
Console
Sources
Network
Performance
Memory
Application
Application
Manifest
Service Wo
Storage
Storage
Local Stora
file://
Session Sto
IndexedDB
Web SQL
Cookies
Trust Token
Cache
Cache Stor
Applicatior
Background Servi
Backgroun
Backgroun
Notificatio
Payment H
Periodic Ba
Push Mess
Frames
top
Filter
Key
Value
estudiantes {}
criteriosInicial {}
criteriosBeta {}
informesIniciales {}
prototiposBeta {}
clases {}
empresas {}
criteriosAlpha {}
representantes {}
profesores {"71":{"identificacion":"71","nombre":"Carlos Zapata","correo":"cmzapa...
criteriosProgreso {}
revisiones {}
retroalimentaciones {}
metodologiasDesarrollo {}
equipos {"listaEquipos":[{"estado":"Sembrado"},{"estado":"Formado"},{"estado":...
semestresCod {"codigoSemestre":0,"semestres":{}}
criteriosFinal {}
informesProgreso {}
carteraDeProyectos {"cantidadProyectos":0,"proyectos":{}}
asignaturas {}
prototiposAlpha {}
informesFinales {}
{,…}
71: {identificacion: "71", nombre: "Carlos Zapata", correo: "cmzapata@unal.edu.co", celular: "305",…}
celular: "305"
contrasena: "1234"
correo: "cmzapata@unal.edu.co"
direccion: "Calle 2 No. 3-4"
identificacion: "71"
nombre: "Carlos Zapata"

SOURCE CODE EXAMPLES

Filename: AlbumModel.php

Source code:

```
function edit($title, $desc, $date, $thumb, $place, $sorttype, $subalbum_sort_type, $image_sortdirection, $album_sortdirection, $password, $password_hint, $custom_data, $album_theme, $watermark){
$sql = "UPDATE Album SET
desc= '$desc', date= '$date', thumb='$thumb', place='$place', sorttype='$sorttype', image_sortdirection='$image_sortdirection', album_sortdirection='$ album_sortdirection ', password='$password', password_hint='$password_hint', custom_data='$custom_data', album_theme='$album_theme', watermark='$watermark'
WHERE title = '$title';
$consulta = $this->query($sql);
}
```

SOURCE CODE EXAMPLES

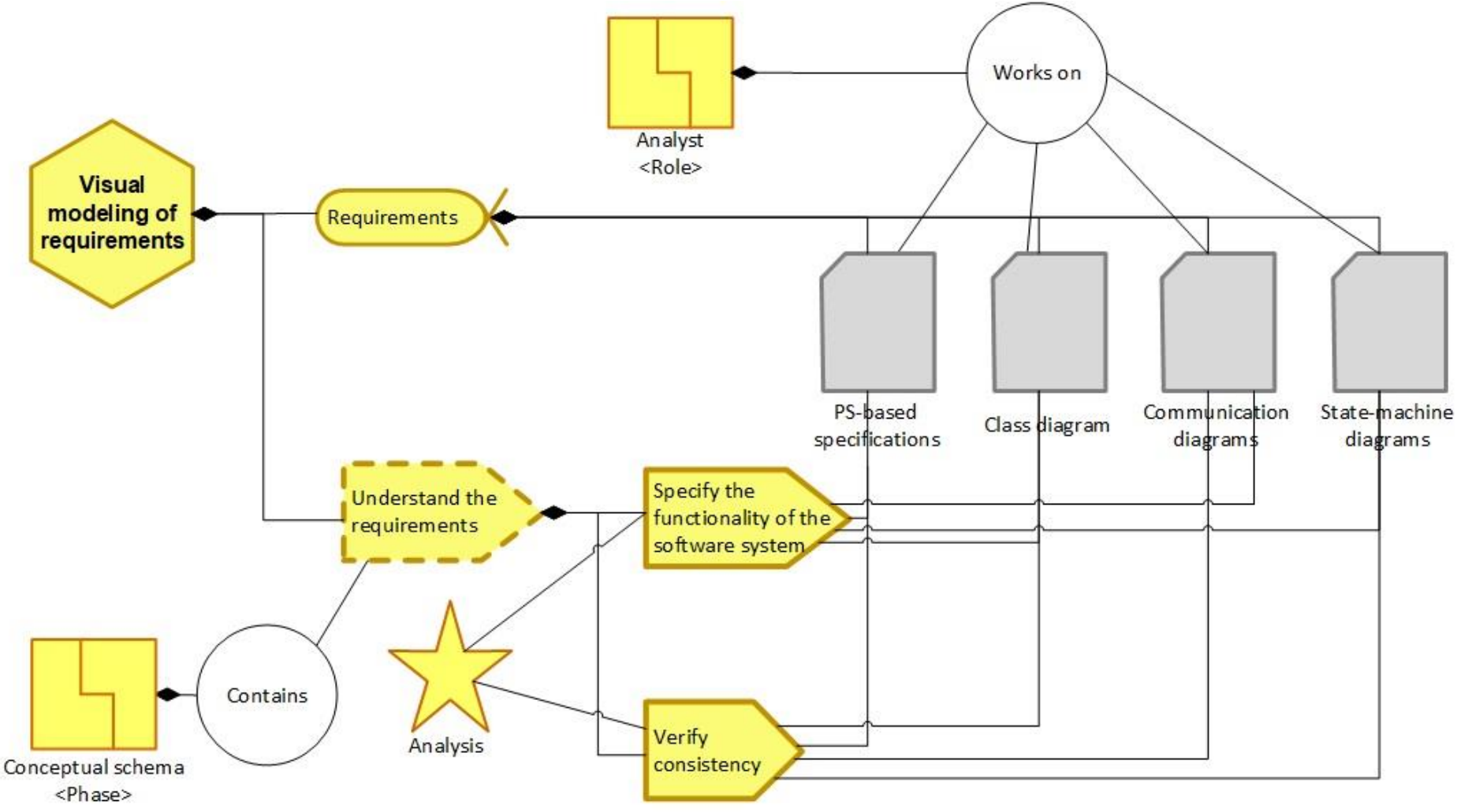
Filename: Album/edit.html
Source code:

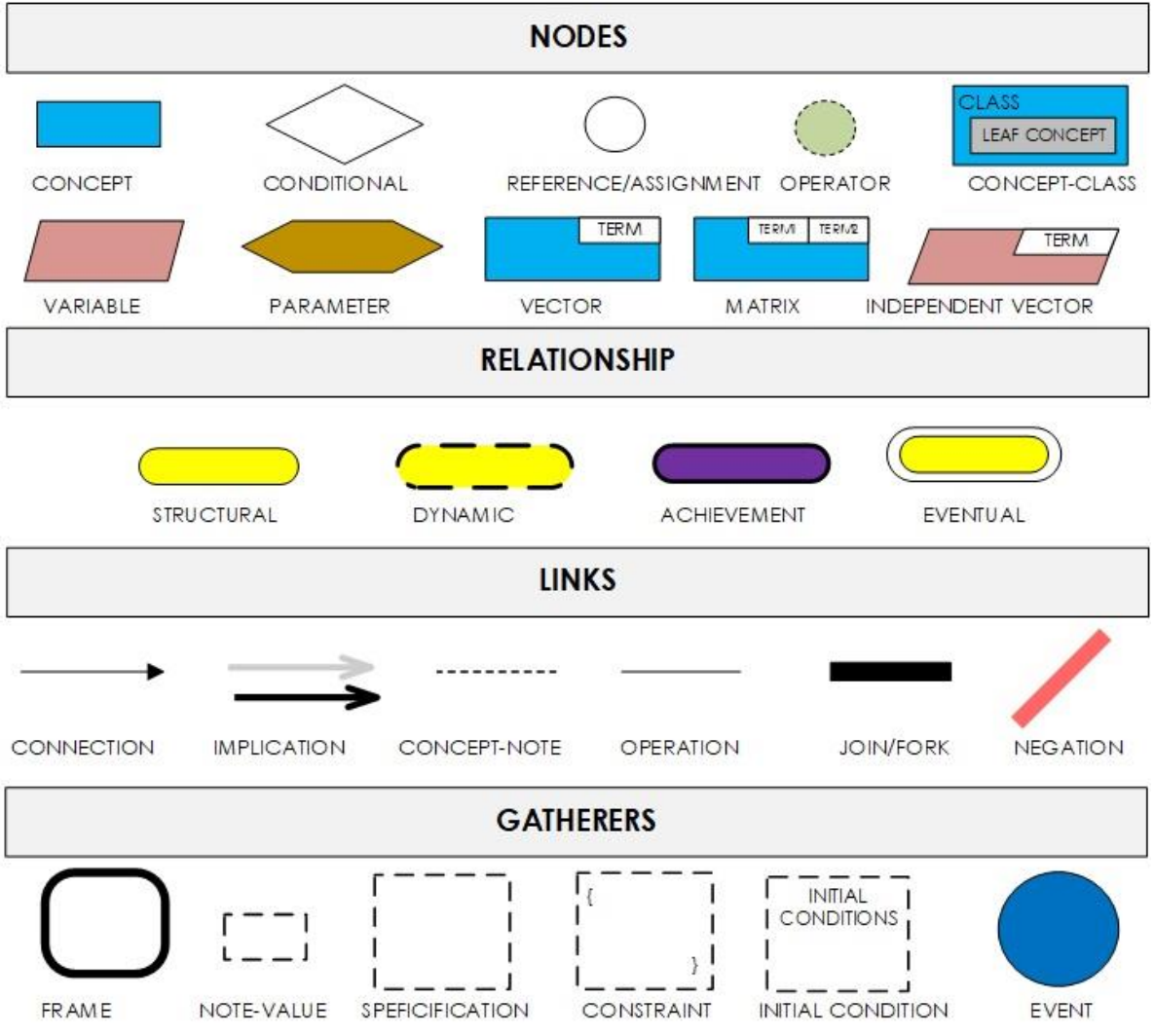
```
<FORM action="edit" method="post">
<label for="title">Title:</label>
<input name="title" type="text" id="title"/>
<label for="desc">Desc:</label>
<input name="desc" type="text" id="desc"/>
<label for="date">Date:</label>
<input name="date" type="text" id="date"/>
<label for="thumb">Thumb:</label>
<input name="thumb" type="text" id="thumb"/>
<label for="place">Place:</label>
<input name="place" type="text" id="place"/>
<label for="sort_type">Sort_type:</label>
<input name="sort_type" type="text" id="sort_type"/>
<label for="
subalbum_sort_type">Subalbum_sort_type:</label>
<input name="subalbum_sort_type" type="text"
id="subalbum_sort_type"/>
<label for=" image_sortdirection">Image_sortdirection:</label>
<input name="image_sortdirection" type="text" id="
image_sortdirection"/>
<label for=" album_sortdirection">Album_sortdirection:</label>
<input name="album_sortdirection" type="text" id="
album_sortdirection"/>
<label for=" password">Password:</label>
<input name="password" type="text" id=" password"/>
<label for=" password_hint">Password_hint:</label>
<input name="password_hint" type="text" id="
password_hint"/>
<label for="custom_data">Custom_data:</label>
<input name=" custom_data" type="text" id=" custom_data"/>
<label for="album_theme">Album_theme:</label>
<input name=" album_theme" type="text" id=" album_theme"/>
<label for="watermark">Watermark:</label>
<input name=" watermark" type="text" id="watermark"/>
<input name=create type='submit' value='Edit'/>
</FORM>
```

SOURCE CODE EXAMPLES

Filename: AlbumController.php
Source code:

```
function edit() {
$data = $_POST;
if(!empty($data)){
$title = $_POST['title'];
$desc = $_POST['desc'];
$date = $_POST['date'];
$place = $_POST['place'];
$sort_type = $_POST['sort_type'];
$subalbum_sort_type =
$_POST['subalbum_sort_type'];
$image_sortdirection =
$_POST['image_sortdirection'];
$album_sortdirection =
$_POST['album_sortdirection'];
$password = $_POST['password'];
$password_hint =
$_POST['password_hint'];
$custom_data = $_POST['custom_data'];
$album_theme = $_POST['album_theme'];
$watermark = $_POST['watermark'];
$album = new AlbumModel();
$album->edit($title, $desc, $date, $thumb,
$place, $sort_type, $subalbum_sort_type,
$image_sortdirection, $album_sortdirection,
$password, $password_hint, $custom_data,
$album_theme, $watermark);
$this->list();
} else {
$this->view->show("Album/edit.html", $data);
}
}
```

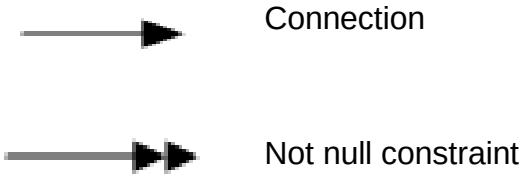

Works on
Analyst
<Role>
Visual modeling of requirements
Requirements
PS-based specifications
Class diagram
Communication diagrams
State-machine diagrams
Understand the requirements
Specify the functionality of the software system
Contains
Analysis
Verify consistency
Conceptual schema
<Phase>

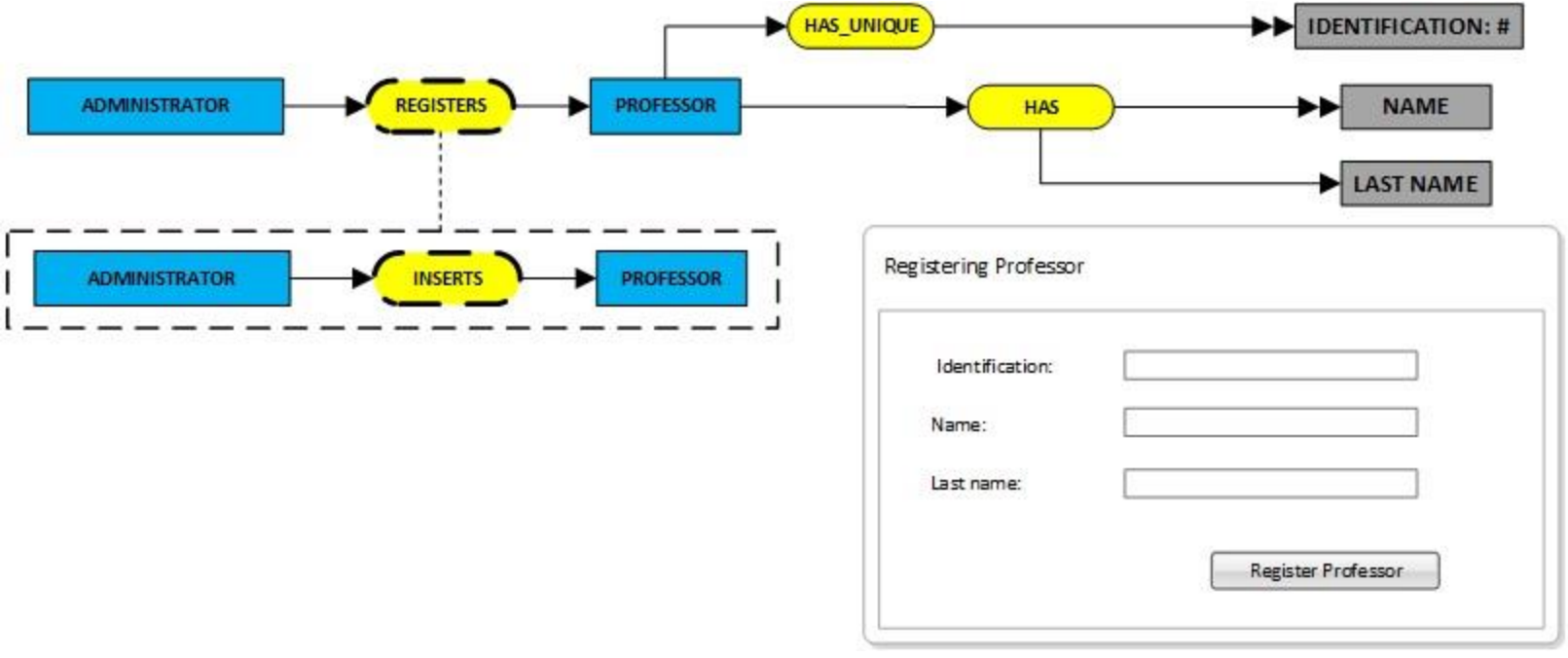

NODES
CONCEPT
CONDITIONAL
REFERENCE/ASSIGNMENT
OPERATOR
CLASS
LEAF CONCEPT
CONCEPT-CLASS
VARIABLE
PARAMETER
TERM
VECTOR
TERM1
TERM2
MATRIX
TERM
INDEPENDENT VECTOR
RELATIONSHIP
STRUCTURAL
DYNAMIC
ACHIEVEMENT
EVENTUAL
LINKS
CONNECTION
IMPLICATION
CONCEPT-NOTE
OPERATION
JOIN/FORK
NEGATION
GATHERERS
FRAME
NOTE-VALUE
SPEFICIFICATION
CONSTRAINT
INITIAL CONDITIONS
INITIAL CONDITION
EVENT

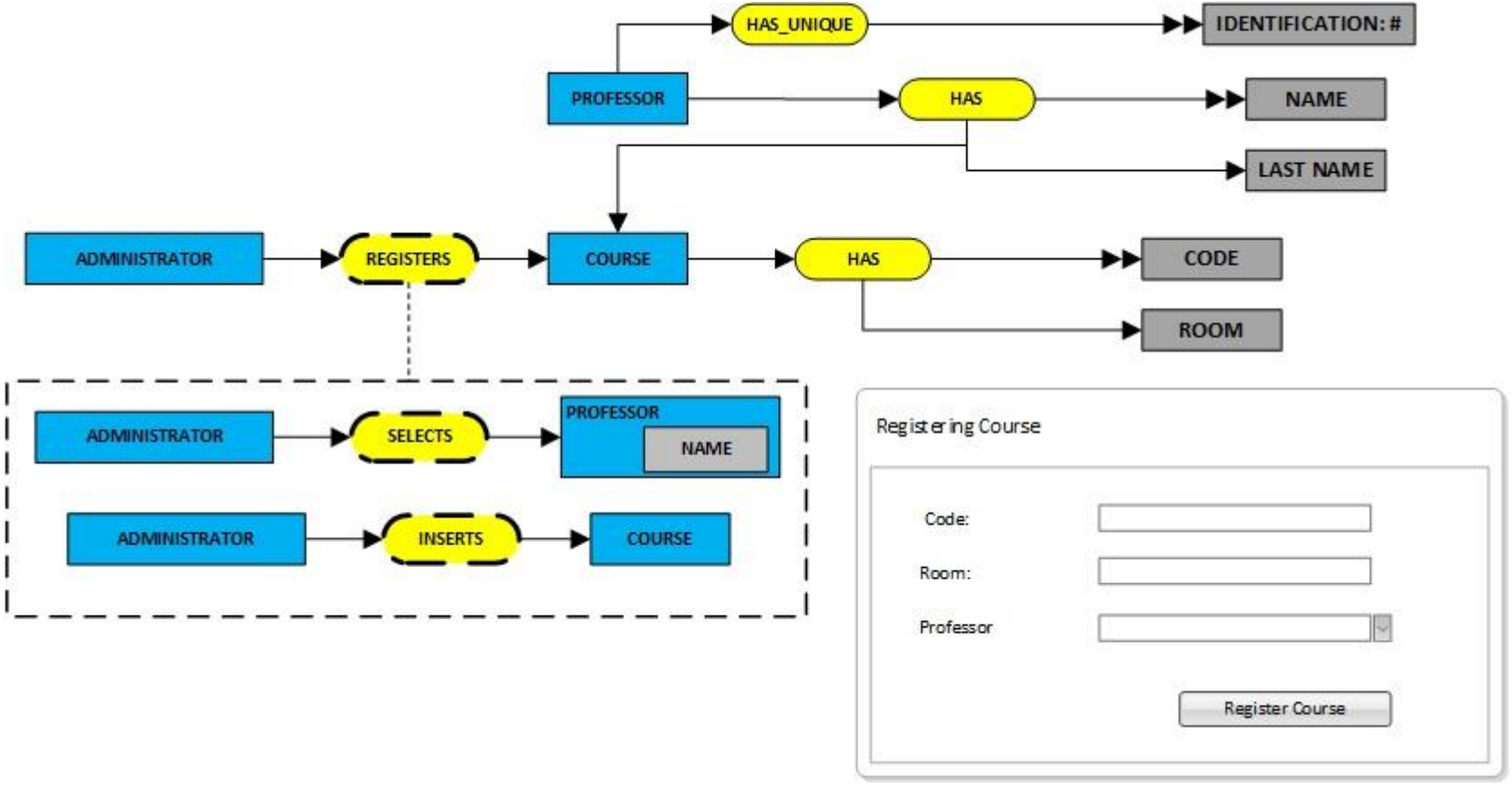
ATOMIC DYNAMIC RELATIONSHIPS

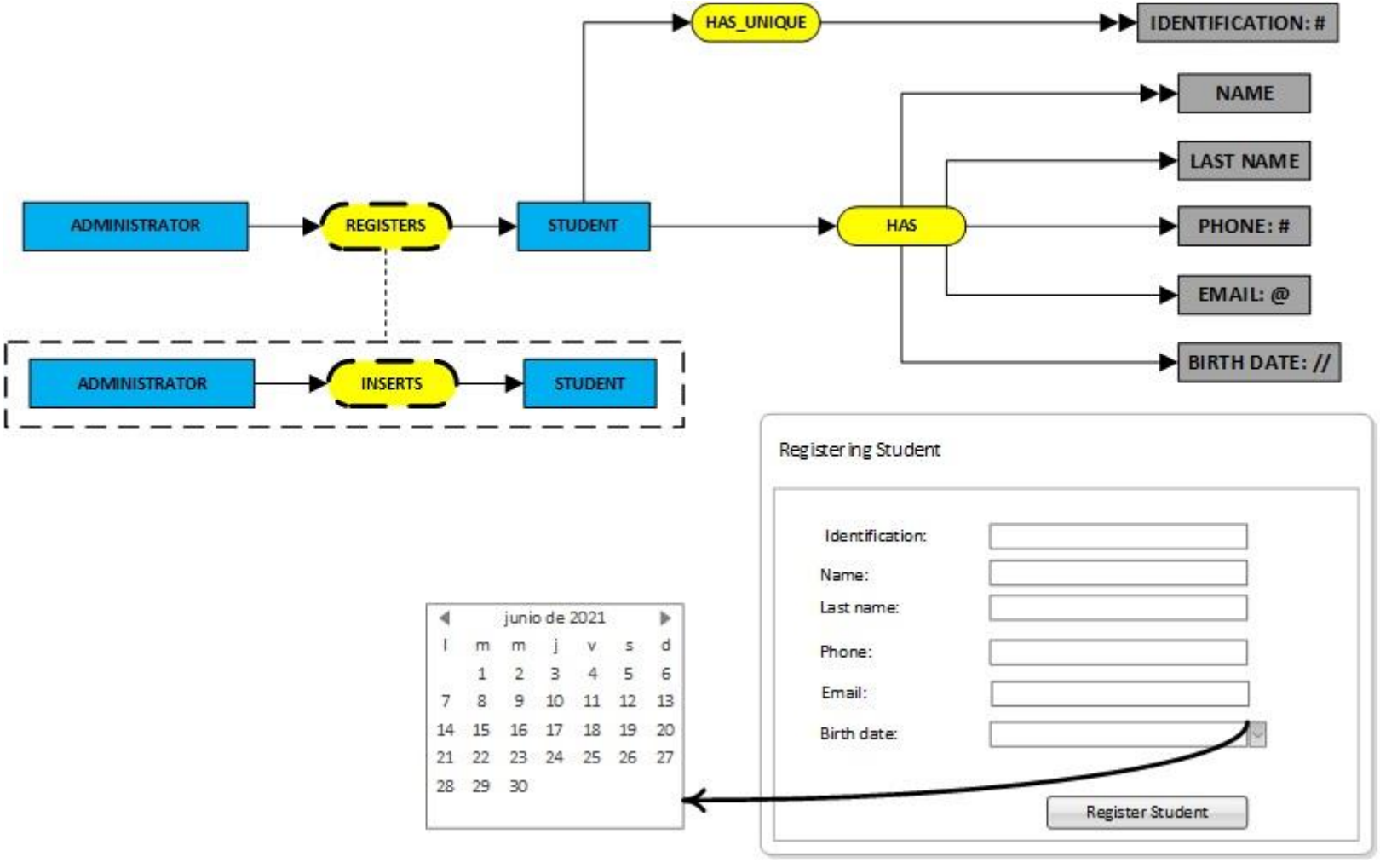
DATA TYPES AND CONSTRAINTS

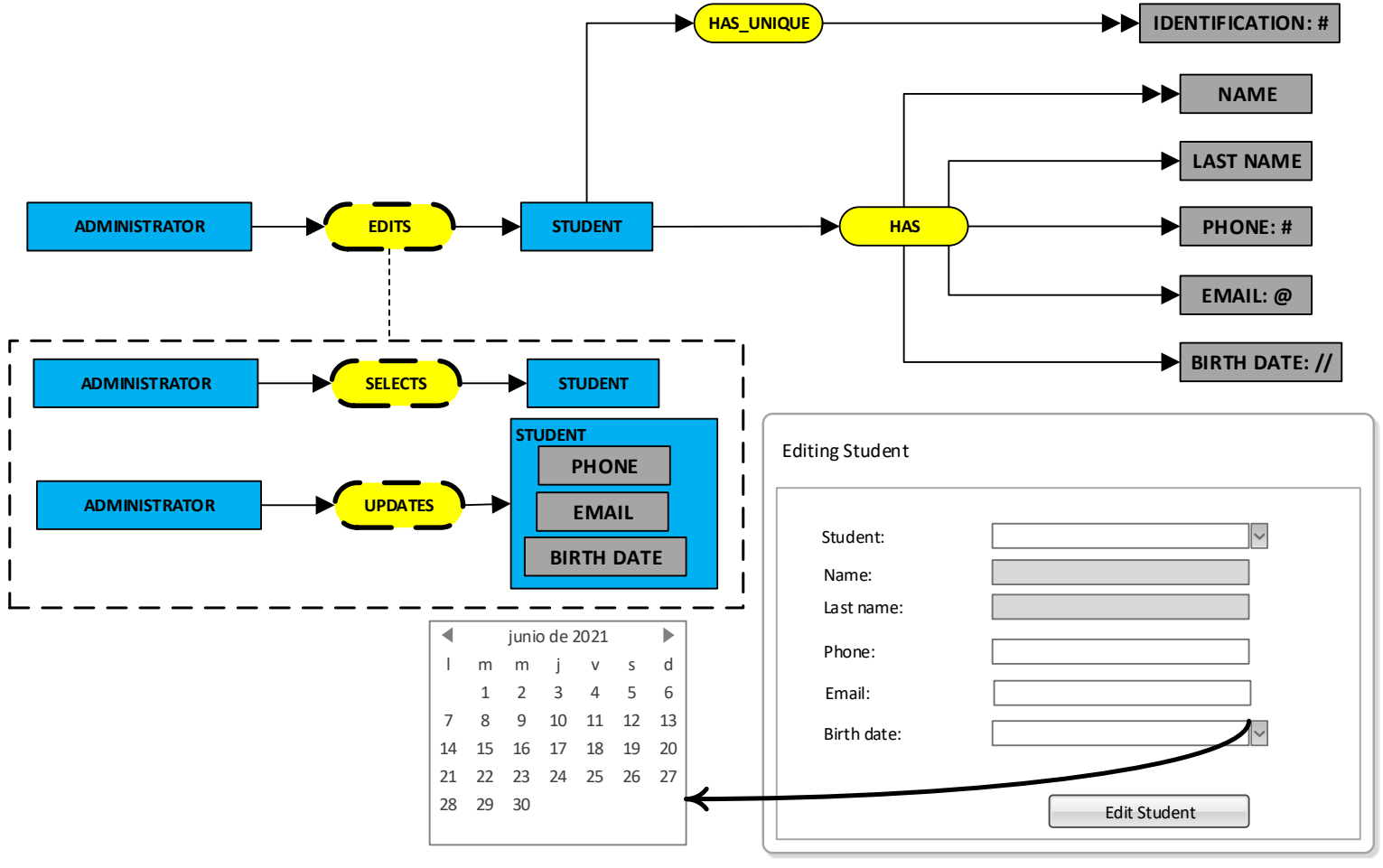
- Date Concept: //
- Number Concept: #
- Boolean Concept: ?
- Email Concept: @
- Text (default) Concept

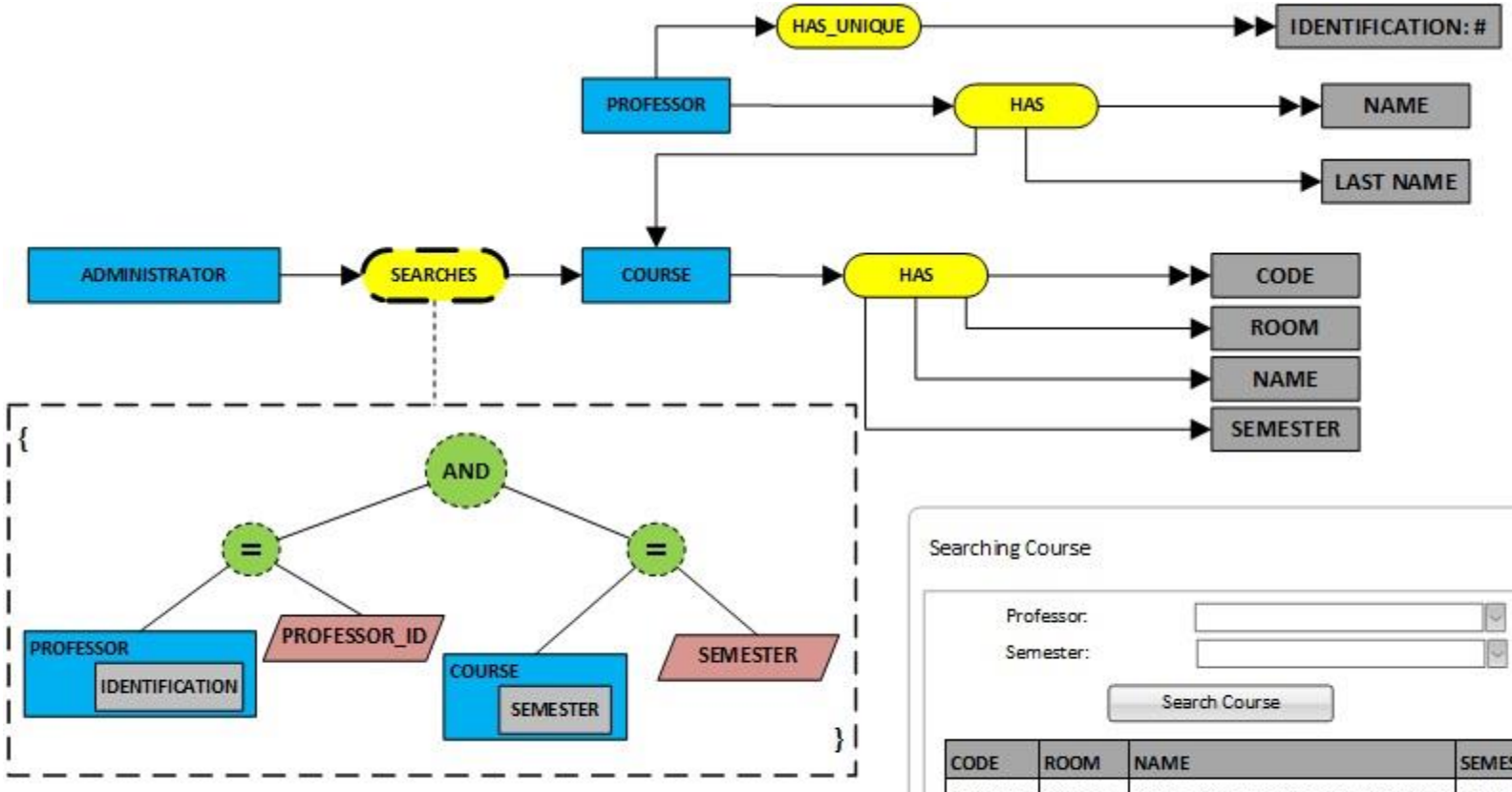


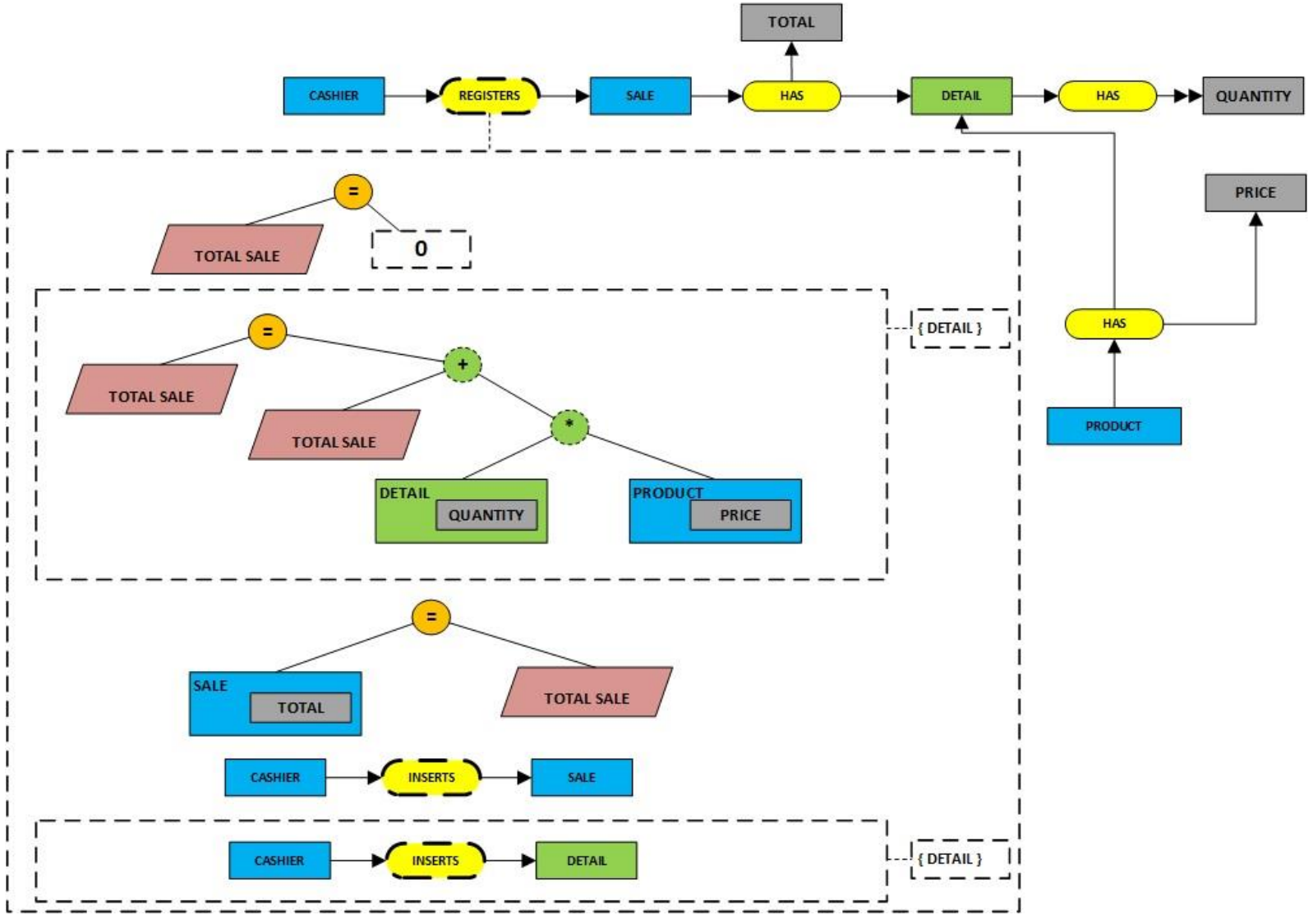

ADMINISTRATOR
REGISTERS
PROFESSOR
HAS_UNIQUE
IDENTIFICATION: #
HAS
NAME
LAST NAME
ADMINISTRATOR
INSERTS
PROFESSOR
Registering Professor
Identification:
Name:
Last name:
Register Professor

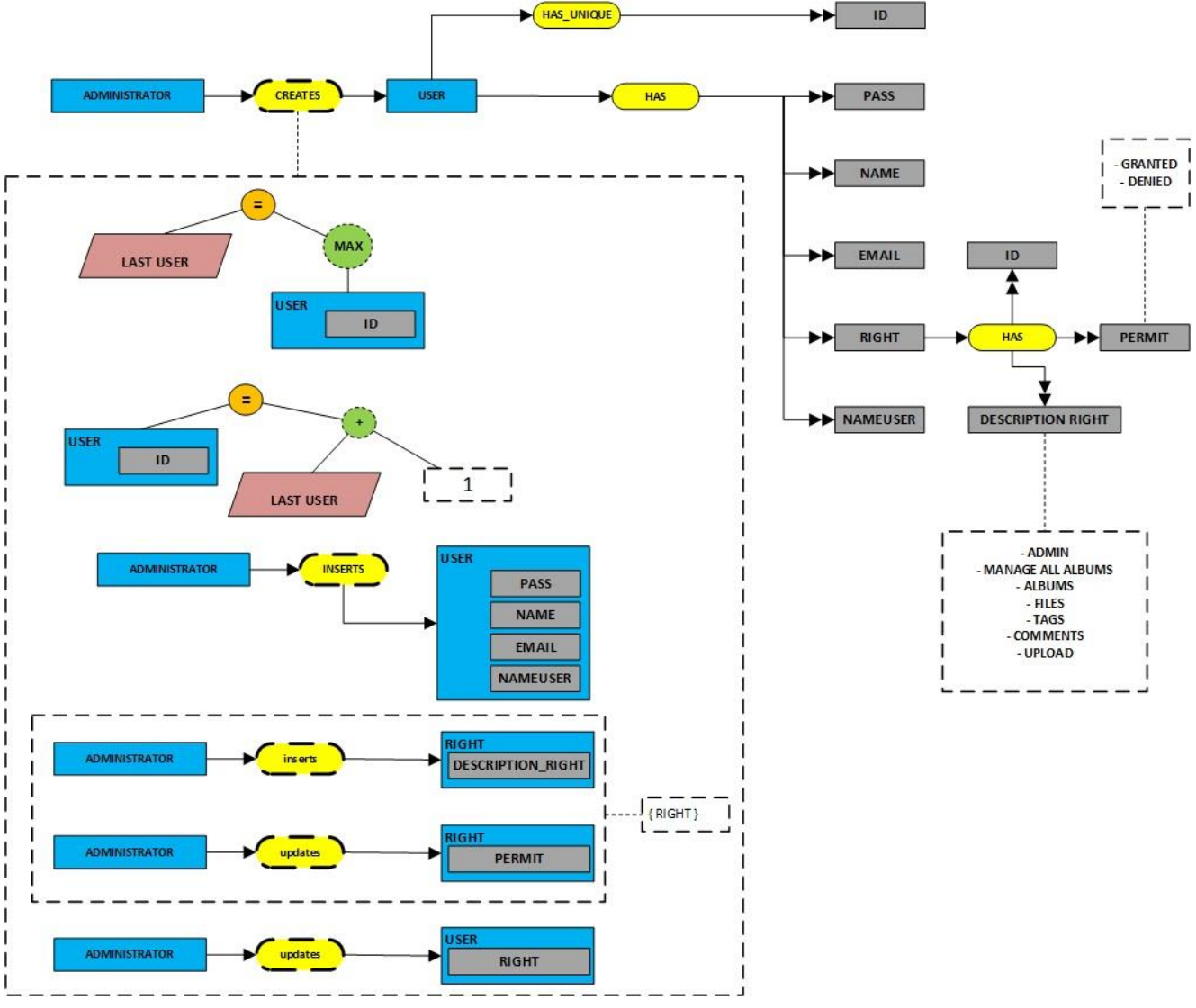

HAS_UNIQUE
IDENTIFICATION: #
PROFESSOR
HAS
NAME
LAST NAME
ADMINISTRATOR
REGISTERS
COURSE
HAS
CODE
ROOM
ADMINISTRATOR
SELECTS
PROFESSOR
NAME
ADMINISTRATOR
INSERTS
COURSE
Registering Course
Code:
Room:
Professor
Register Course

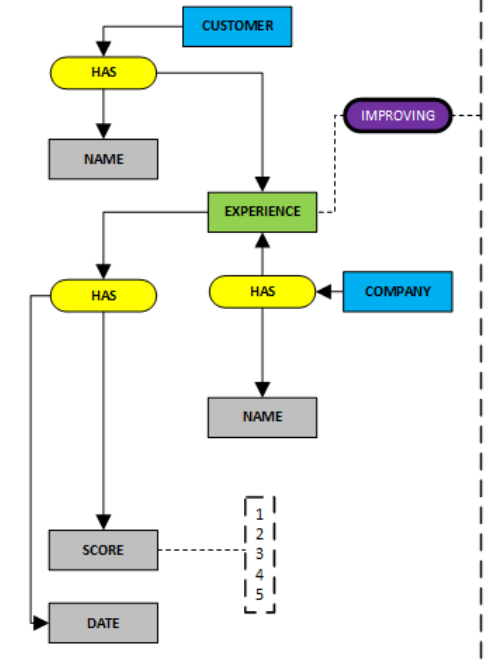

HAS_UNIQUE
IDENTIFICATION: #
NAME
LAST NAME
PHONE: #
EMAIL: @
BIRTH DATE: //
ADMINISTRATOR
REGISTERS
STUDENT
HAS
ADMINISTRATOR
INSERTS
STUDENT
Registering Student
Identification:
Name:
Last name:
Phone:
Email:
Birth date:
Register Student
junio de 2021
l m m j v s d
1 2 3 4 5 6
7 8 9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30


HAS_UNIQUE
IDENTIFICATION: #
NAME
LAST NAME
PHONE: #
EMAIL: @
BIRTH DATE: //
ADMINISTRATOR
EDITS
STUDENT
HAS
ADMINISTRATOR
SELECTS
STUDENT
ADMINISTRATOR
UPDATES
STUDENT
PHONE
EMAIL
BIRTH DATE
junio de 2021
l m m j v s d
1 2 3 4 5 6
7 8 9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30
Editing Student
Student:
Name:
Last name:
Phone:
Email:
Birth date:
Edit Student

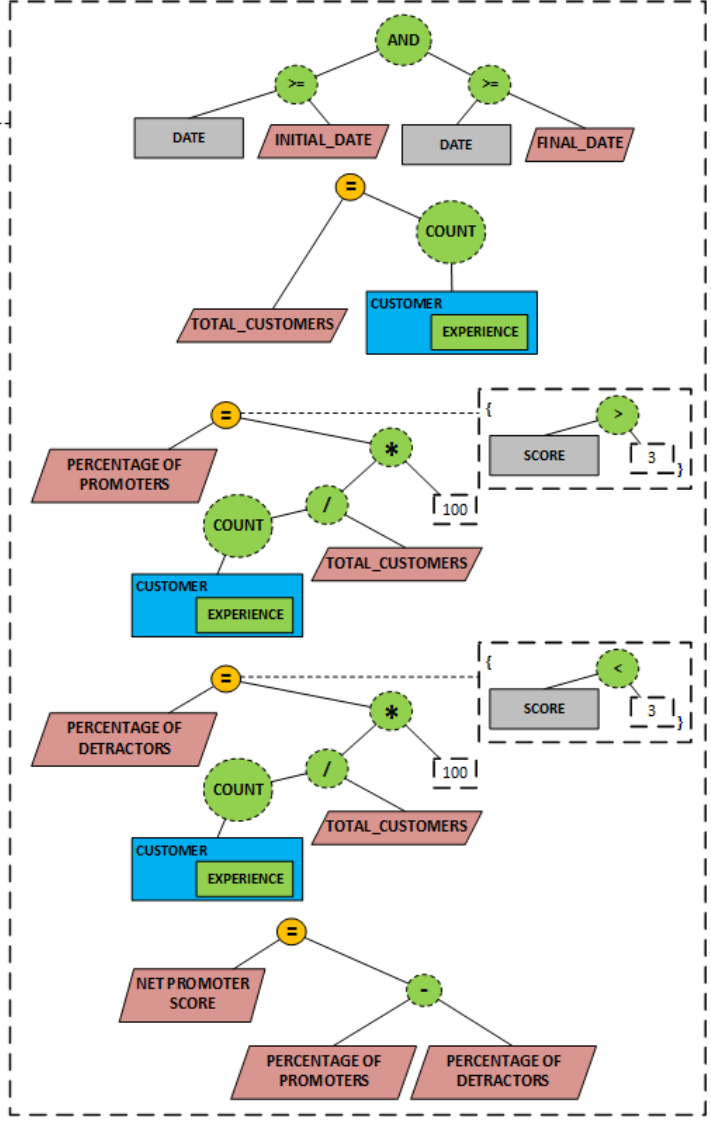

HAS_UNIQUE
IDENTIFICATION: #
PROFESSOR
HAS
NAME
LAST NAME
ADMINISTRATOR
SEARCHES
COURSE
HAS
CODE
ROOM
NAME
SEMESTER
{
AND
=
=
PROFESSOR
IDENTIFICATION
PROFESSOR_ID
COURSE
SEMESTER
SEMESTER
}
Searching Course
Professor:
Semester:
Search Course
CODE
ROOM
NAME
SEMESTER
3007851
VIRTUAL
SOFTWARE PROJECT MANAGEMENT
2021-1S
3007852
VIRTUAL
REQUIREMENTS ENGINEERING
2021-1S
3008407
VIRTUAL
COMPUTATIONAL LINGUISTICS
2021-1S

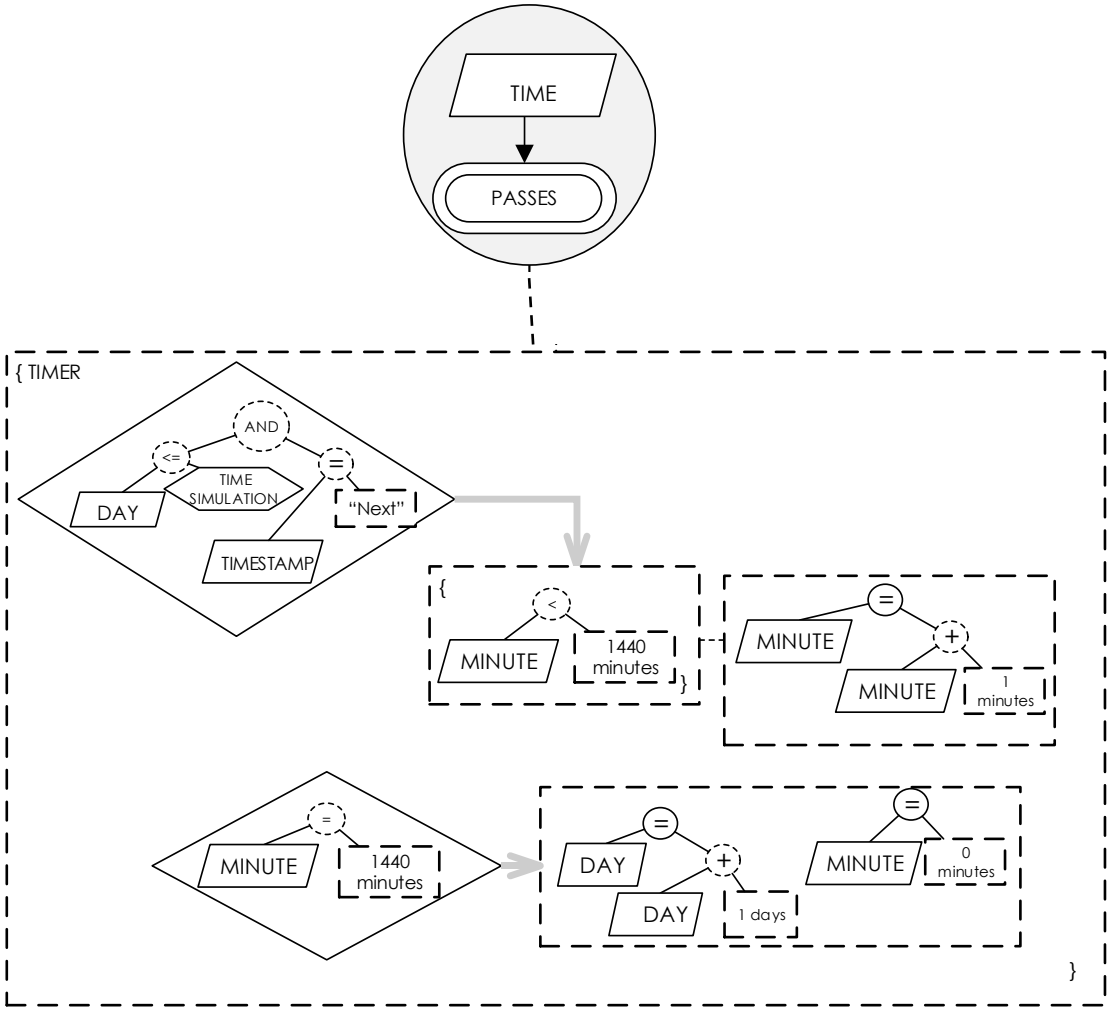

TOTAL
CASHIER
REGISTERS
SALE
HAS
DETAIL
HAS
QUANTITY
PRICE
=
TOTAL SALE
0
{ DETAIL }
HAS
=
TOTAL SALE
+
TOTAL SALE
*
DETAIL
QUANTITY
PRODUCT
PRICE
PRODUCT
=
SALE
TOTAL
TOTAL SALE
CASHIER
INSERTS
SALE
CASHIER
INSERTS
DETAIL
{ DETAIL }


HAS_UNIQUE
ID
ADMINISTRATOR
CREATES
USER
HAS
PASS
NAME
EMAIL
RIGHT
NAMEUSER
- GRANTED
- DENIED
ID
HAS
PERMIT
DESCRIPTION RIGHT
- ADMIN
- MANAGE ALL ALBUMS
- ALBUMS
- FILES
- TAGS
- COMMENTS
- UPLOAD
=
LAST USER
MAX
USER
ID
=
+
USER
ID
LAST USER
1
ADMINISTRATOR
INSERTS
USER
PASS
NAME
EMAIL
NAMEUSER
ADMINISTRATOR
inserts
RIGHT
DESCRIPTION_RIGHT
ADMINISTRATOR
updates
RIGHT
PERMIT
{RIGHT}
ADMINISTRATOR
updates
USER
RIGHT



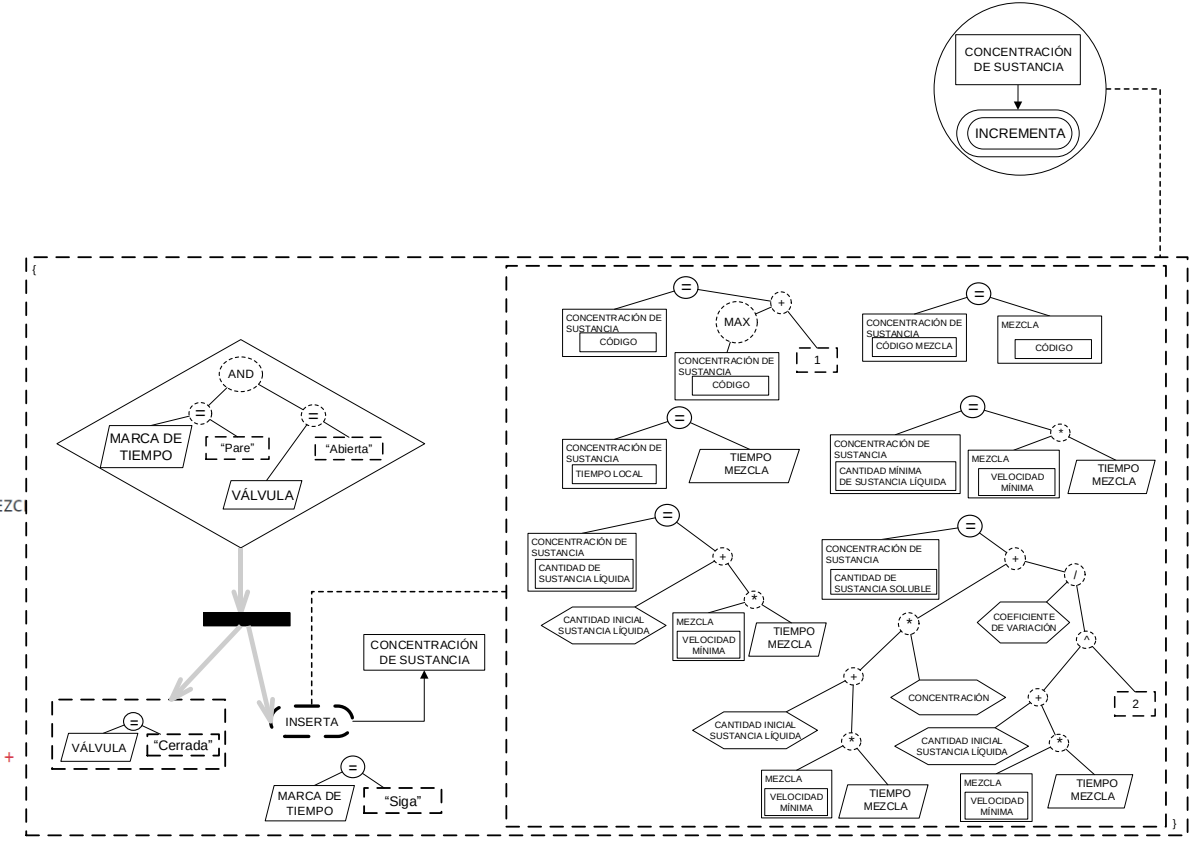
EXPERIENCE			
CUSTOMER.NAME	COMPANY.NAME	DATE	SCORE
JOHN	XYZ INC.	01/07/2021	5
SARAH	XYZ INC.	01/09/2021	3
CHARLES	XYZ INC.	01/10/2021	4
MICK	XYZ INC.	01/10/2021	2
KEN	XYZ INC.	01/11/2021	3
BEVERLY	XYZ INC.	01/12/2021	1
DEBRA	XYZ INC.	01/14/2021	4
VERA	XYZ INC.	01/14/2021	2
RON	XYZ INC.	01/15/2021	5
CRAIG	XYZ INC.	01/16/2021	3
ALBERT	XYZ INC.	01/17/2021	2
MOLLY	XYZ INC.	01/18/2021	1
ASHLEY	XYZ INC.	01/19/2021	3

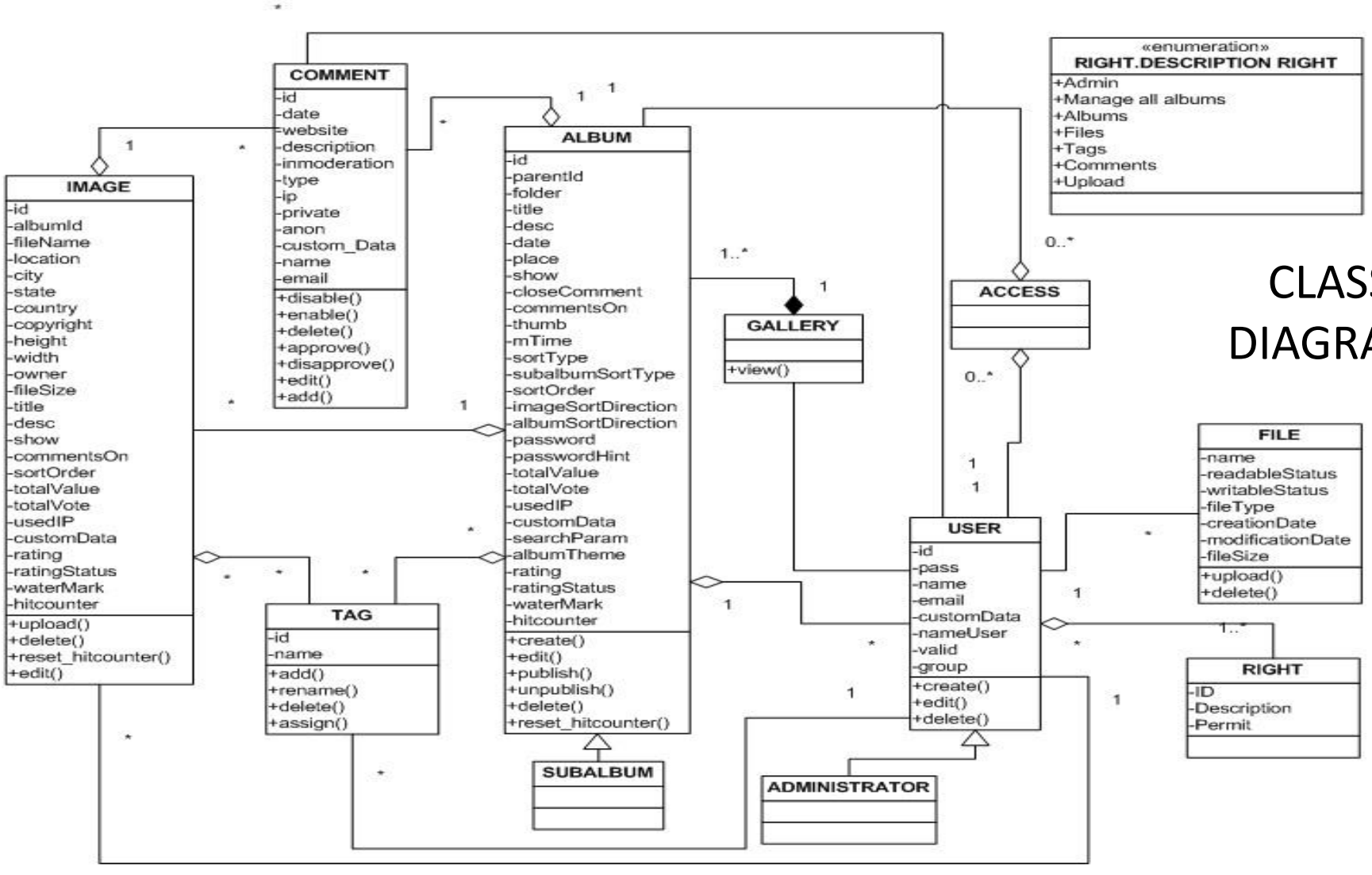





```
def TIME_PASSES():
    global MINUTE
    global DAY
    global TIMESTAMP
    #We verify the bounds of
simulation
    if(DAY <= SIMULATION_TIME
       and
       TIMESTAMP == "Next"):
        #One minute passed
        if(MINUTE<1440):
          MINUTE = MINUTE + 1
    if(MINUTE == 1440):
        #Next day begins
        DAY = DAY + 1
        MINUTE = 0
```

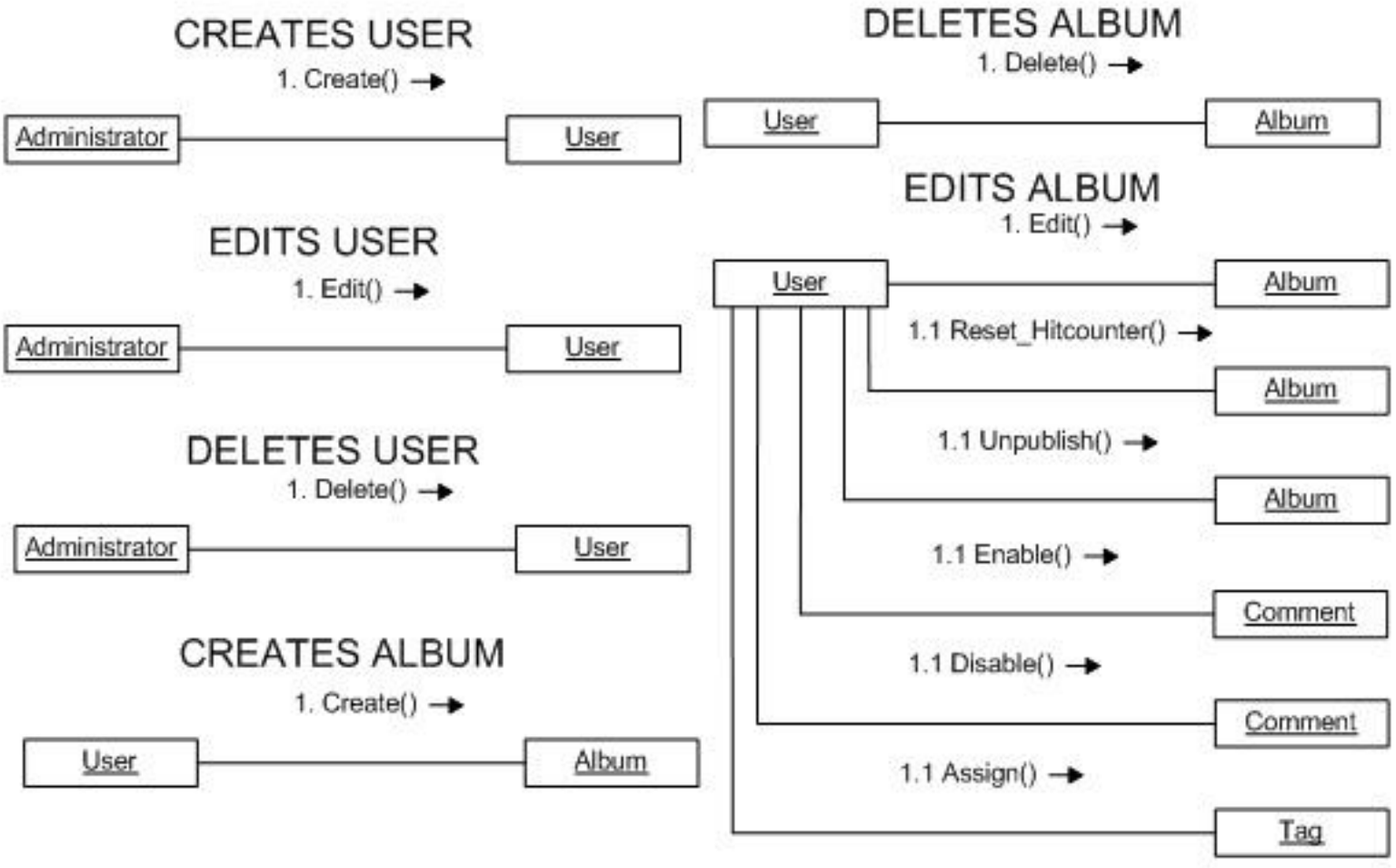
```
CREATE TRIGGER "CONCENTRACION_DE_SUSTANCIA_I"
    AFTER UPDATE OF "MARCA_DE_TIEMPO" OR UPDATE OF "VALVULA" ON "VARIABLE"
DECLARE
    v_MARCA_DE_TIEMPO VARIABLE.MARCA_DE_TIEMPO%TYPE;
    v_VALVULA VARIABLE.VALVULA%TYPE;

    v_CODIGO_CONCENTRACION_DE CONCENTRACION_DE_SUSTANCIA.CODIGO%TYPE;
    v_CODIGO_MEZCLA_CONCENTRA
CONCENTRACION_DE_SUSTANCIA.CODIGO_MEZCLA%TYPE;
    v_CODIGO_MEZCLA MEZCLA.CODIGO%TYPE;

    v_CANTIDAD_MINIMA_SUSTANCIA_LI := v_VELOCIDAD_MINIMA_MEZCLA *
v_TIEMPO_MEZCLA;

    v_CANTIDAD_DE_SUSTANCIA_S := ((V_CANTIDAD_INICIAL_SUSTANCIA_L +
(v_VELOCIDAD_MINIMA_MEZCLA * v_TIEMPO_MEZCLA)) * v_CONCENTRACION) +
(power((V_CANTIDAD_INICIAL_SUSTANCIA_L + (v_VELOCIDAD_MINIMA_MEZCLA *
v_TIEMPO_MEZCLA)), 2) / v_COEFICIENTE_DE_VARIACIO);

    INSERT INTO CONCENTRACION_DE_SUSTANCIA (CODIGO,
CANTIDAD_DE_SUSTANCIA_LIQUIDA, CANTIDAD_MINIMA_DE_SUSTANCIA_L,
CANTIDAD_DE_SUSTANCIA_SOLUBLE, CODIGO_MEZCLA, TIEMPO_LOCAL)
    VALUES (v_CODIGO_CONCENTRACION_DE,v_CANTIDAD_DE_SUSTANCIA_LIQUID,
v_CANTIDAD_MINIMA_SUSTANCIA_LI, v_CANTIDAD_DE_SUSTANCIA_S, v_CODIGO_MEZC
v_TIEMPO_LOCAL_CONCENTRAC);
  END IF;
END;

IF v_MARCA_DE_TIEMPO = 'PARE' AND v_VALVULA = 'ABIERTA' THEN
  v_CODIGO_CONCENTRACION_DE := v_CODIGO_CONCENTRACION_DE + 1;
  v_CODIGO_MEZCLA_CONCENTRA := v_CODIGO_MEZCLA;
  v_TIEMPO_LOCAL_CONCENTRAC := v_TIEMPO_MEZCLA;
  v_CANTIDAD_DE_SUSTANCIA_LIQUID :=  V_CANTIDAD_INICIAL_SUSTANCIA_L +
v_CANTIDAD_MINIMA_SUSTANCIA_LI;
```



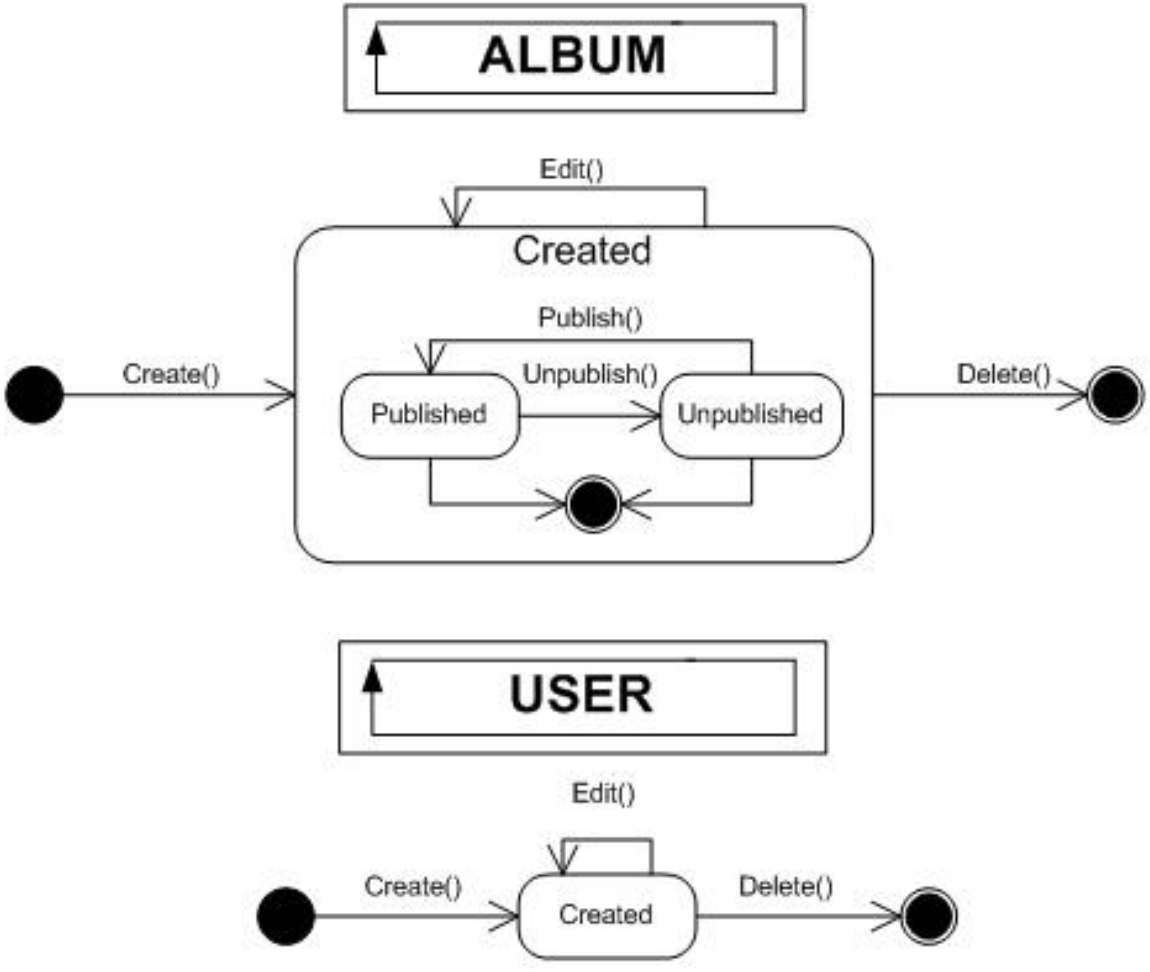


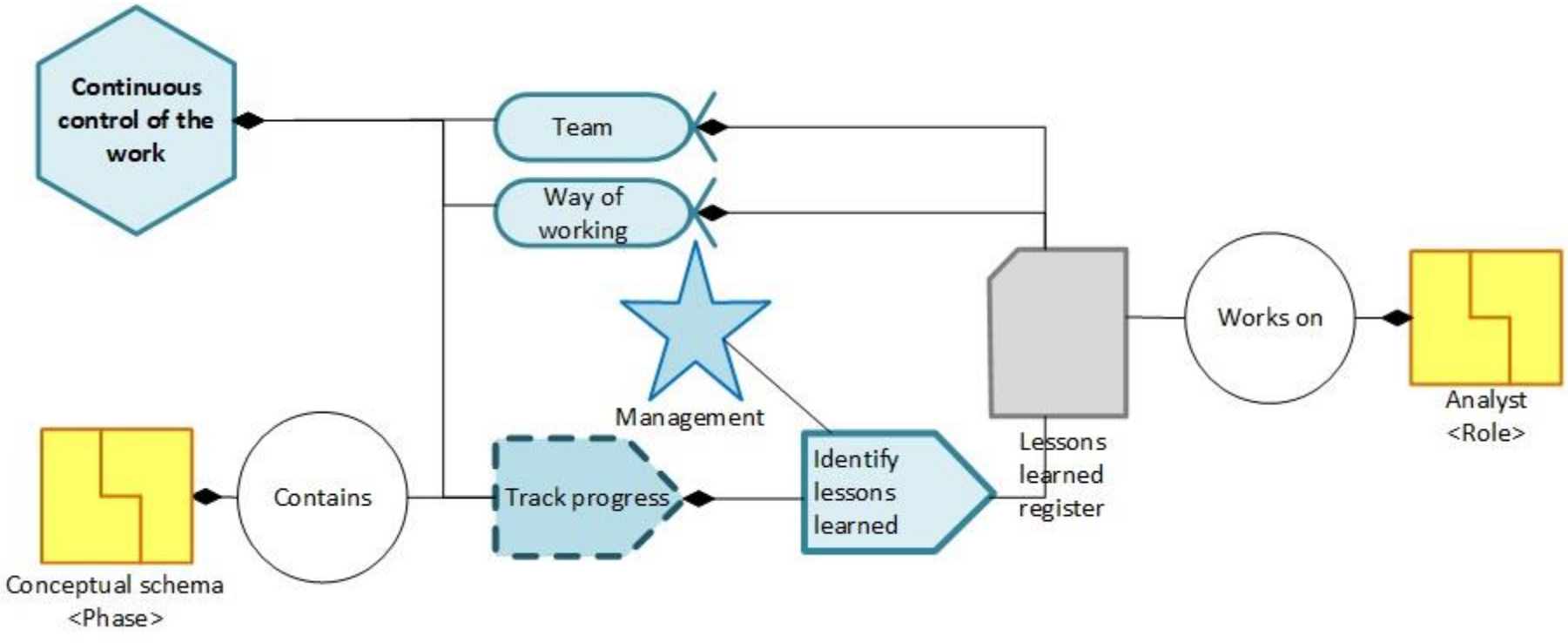
CLASS DIAGRAM

COMMUNICATION DIAGRAMS



STATE MACHINE DIAGRAMS

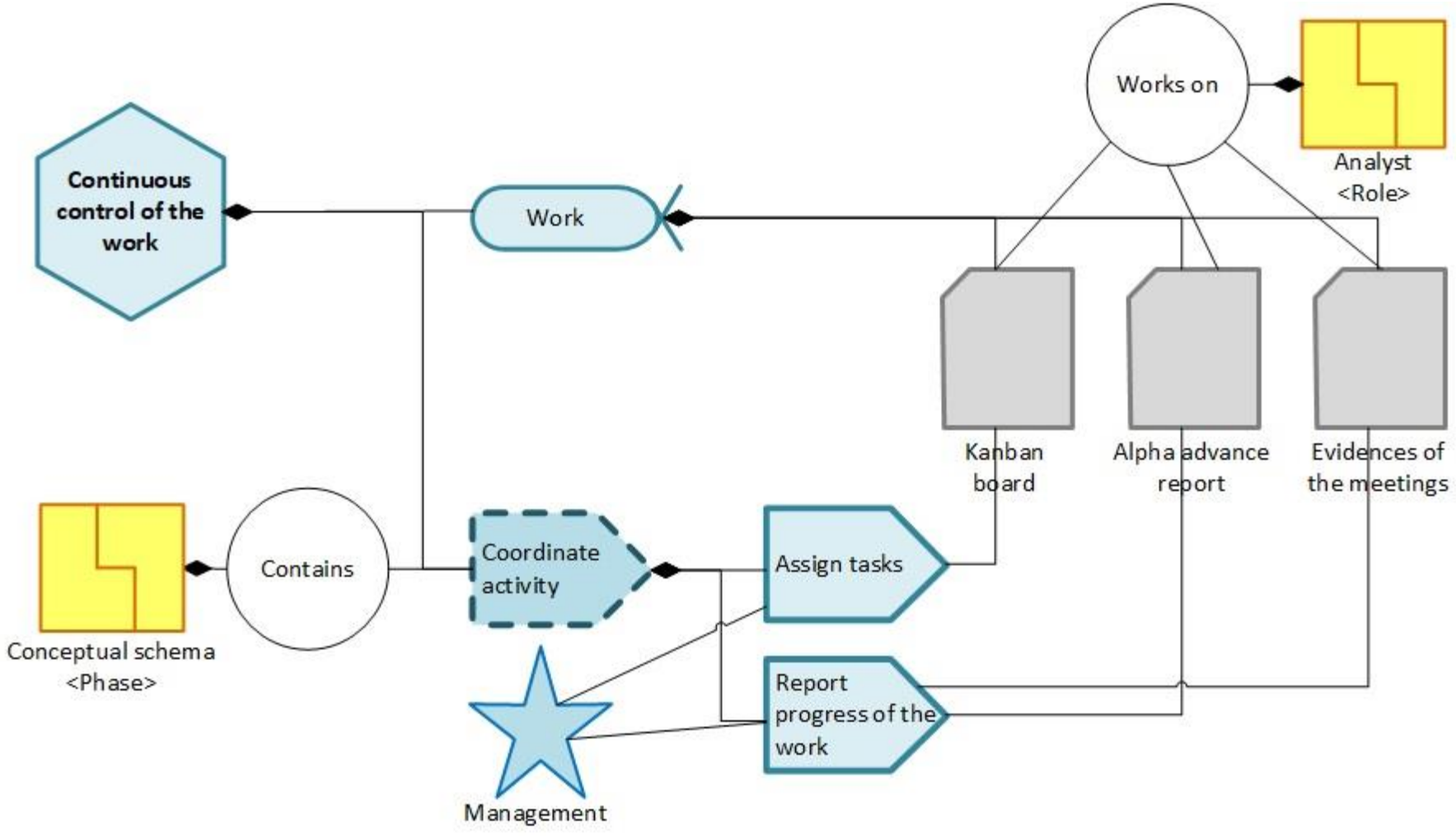



Continuous control of the work
Team
Way of working
Management
Works on
Analyst <Role>
Lessons learned register
Identify lessons learned
Track progress
Contains
Conceptual schema <Phase>

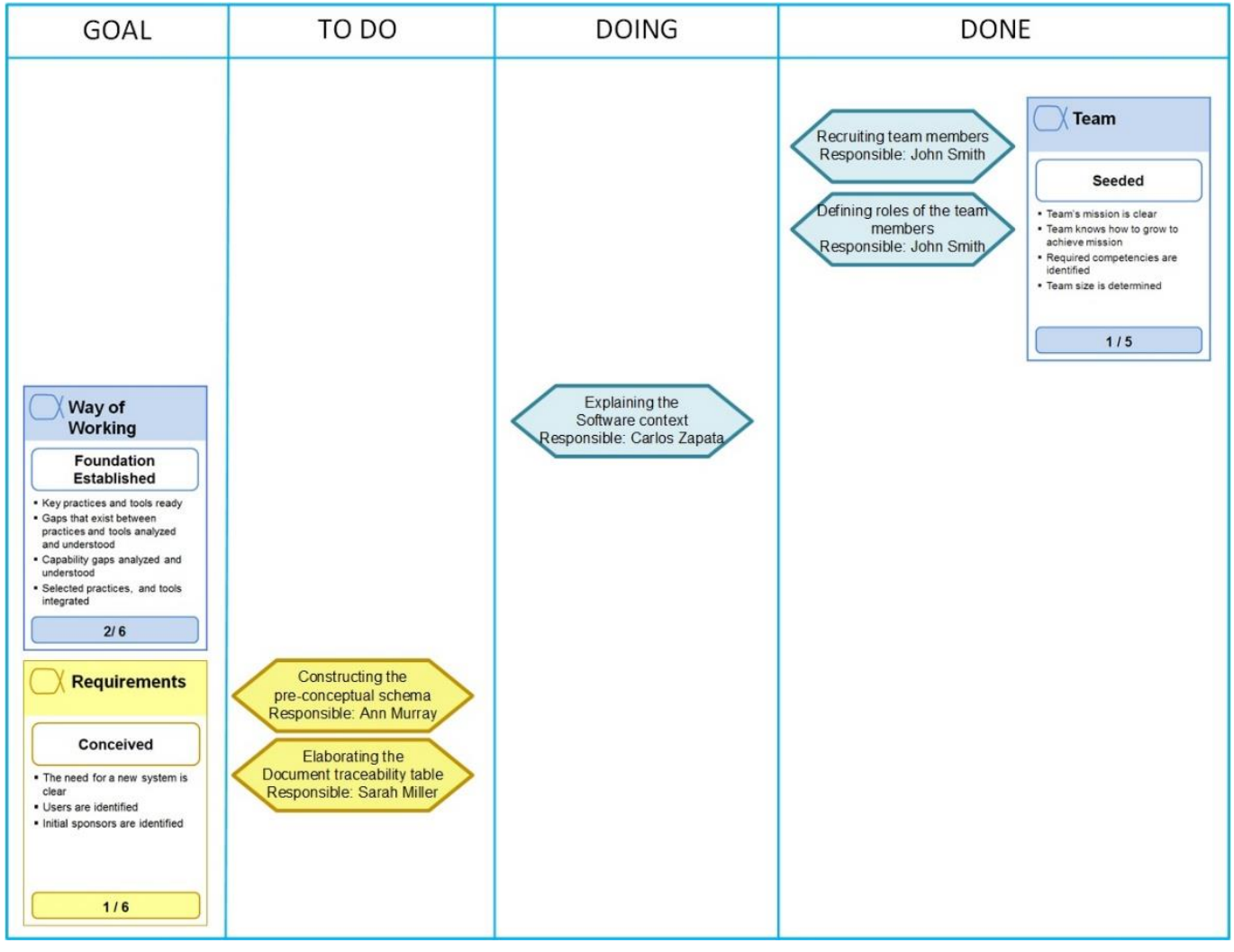
LESSONS LEARNED REGISTER

Team				Date		
Code	Alpha	State	Situation	Lesson learned	Impact	Recommendation

	IMPACT LIKERT	VERY LOW	LOW	MODERATE	HIGH	VERY HIGH
	COLOR	255,0,0	255,150,0	255,255,0	150,255,50	0,255,0


Works on
Analyst
<Role>
Continuous control of the work
Work
Kanban board
Alpha advance report
Evidences of the meetings
Contains
Coordinate activity
Assign tasks
Conceptual schema
<Phase>
Report progress of the work
Management

KANBAN BOARD



ALPHA ADVANCE REPORT

ALPHA ADVANCE REPORT				
STATE	**SUMMARY OF THE ACHIEVEMENT**	**TASK**	**DATE/DURATION**	**CONTENTS/OBSERVATIONS**
	We select our team members from a list of candidates provided by our project manager. Then, we assign responsibilities by studying their curricula vitae.	Recruiting team members Responsible: John Smith	02/23/2021 6 hours	Team: John Smith, Ann Murray, Sarah Miller, George Vincent
		Defining roles of the team members Responsible: John Smith	02/24/2021 8 hours	John Smith: Developer/Tester Ann Murray: Analyst Sarah Miller: Analyst/Architect George Vincnt: Developer/Tester
	We design some work products in order to clarify the requirements. The final work product we completed is the pre-conceptual schema, since it includes all the information we gather from the stakeholder.	Constructing the pre-conceptual schema Responsible: Ann Murray	03/15/2021 8 hours	Elements of the preliminary version of the pre-conceptual schema: 100 concepts 25 structural relationships 23 dynamic relationships 5 events 3 conditionals 18 achievement relationships 9 implications

Deliverable 4: Conceptual Schema

- Corrections to deliverables 1, 2, and 3 and the alpha prototype
- Source code examples
- Beta prototype
- Ps-based specifications for dynamic relationships, events, and achievement relationships
- Class diagram
- Communication diagrams
- State machine diagrams
- Lessons learned register (team and way of working)
- Kanban board
- Alpha advance report
- Evidences of the meetings (minutes of the meetings, photos, chat, etc.)