Elderly at home

TUTORÍA 2

NÚRIA MORENO CHAMORRO Y JUDITH ZORÍO VENTURA

2020

ÍNDICE

[**1.** **Organización** 2](#_Toc38036667)

[**2.** **Diseño conceptual** 2](#_Toc38036668)

[**3.** **Diseño lógico** 2](#_Toc38036669)

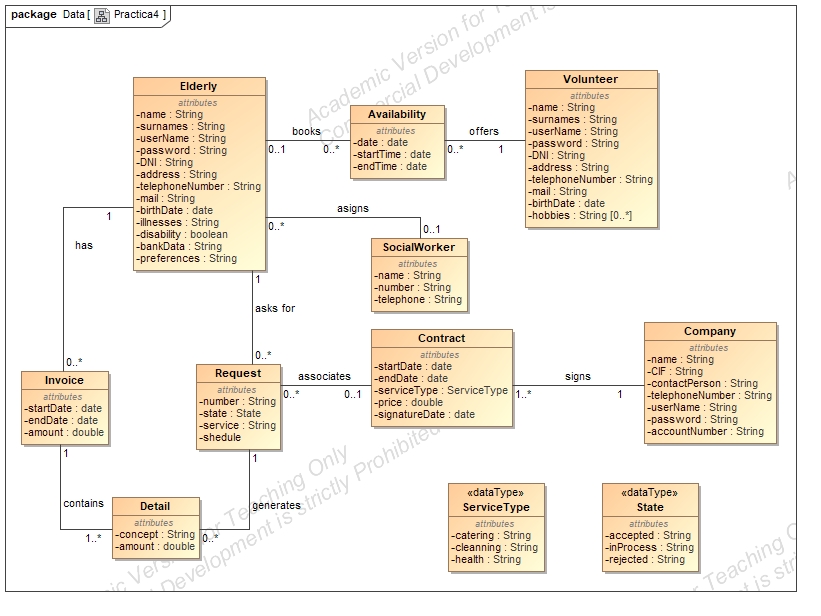
[**4.** **Diseño físico** 3](#_Toc38036670)

[**5.** **Sitemap** 7](#_Toc38036671)

# **Organización**

Desde la primera tutoría el trabajo se ha estado organizando a través de llamadas por medio de Skype entre las dos componentes del grupo, ya que debido a la situación de confinamiento no se han podido realizar las reuniones habituales. En estas reuniones online se decidió que Núria Moreno sería la encargada de corregir el site map y el diseño conceptual, mientras que Judith Zorío se encargaría de corregir la base de datos y los errores en el proyecto Maven. Con respecto a la tutoría, decidimos que Núria se encargaría de la parte de front-office y Judith de la parte de back-office. Las siguientes estadísticas de GitHub reflejan el trabajo realizado por ambas componentes del grupo:

# **Diseño conceptual**



# **Diseño lógico**

**Elderly**(name, surNames, userName, password, DNI, telephoneNumber, mail, birthDate, hobbies, llnesses, disability, socialWorker)

hobbies, llnesses, disability

socialWorker clave ajena a la tabla SocialWorker. Acepta nulos. Regla de borrado restrict, actualización cascada.

**BankData**(IBAN, bank, branch office, DNI)

DNI clave ajena a la tabla Elderly. No acepta nulos. Borrado restrict y actualización en cascada

**Address**(street, postal code, city, country, DNI\_elderly, DNI\_volunteer)

DNI\_elderly clave ajena a la tabla Elderly. Acepta nulos. Borrado restrict y actualización en cascada

DNI\_volunteer clave ajena a la tabla Volunteer. Acepta nulos. Borrado restrict y actualización en cascada

**Volunteer**(name, surnames, userName, password, DNI, telephoneNumber, mail, birthDate, hobbies)

hobbies acepta nulos.

**Invoice**(code, startDate, endDate, amount, DNI)

DNI clave ajena a la tabla Elderly. No acepta nulos. Borrado restrict actualización en cascada

**Detail**(concept, amount, code, number)

code clave ajena a la tabla Invoice. No acepta nulos. Borrado restrict y actualización en cascada

number clave ajena a la tabla Request. No acepta nulos. Borrado restrict y actualización en cascada.

**Availability**(date, startTime, endTime, DNI\_elderly, DNI\_volunteer)

DNI\_elderly clave ajena a la tabla Elderly. Acepta nulos. Borrado restrict actualización en cascada

DNI\_volunteer clave ajena a la tabla Volunteer. No acepta nulos. Borrado restrict actualización en cascada

**Request(**number, state, service, schedule, DNI, idContract)

DNI clave ajena a la tabla Elderly. No acepta nulos. Borrado restrict actualización en cascada

idContract clave ajena a la tabla Contract. Acepta nulos. Borrado restrict actualización en cascada.

**Company**(name, CIF, contactPerson, telephoneNumber, userName, password)

**Contract**(id, startDate, endDate, serviceType, price, signatureDate, CIF)

CIF clave ajena a la tabla Company. No acepta nulos. Borrado restrict actualización en cascada

# **Diseño físico**

CREATE TABLE Elderly (

name VARCHAR(50),

surNames VARCHAR(50),

userName VARCHAR(50),

password VARCHAR(50),

telephoneNumber INTEGER,

mail VARCHAR(50),

birthDate DATE,

hobbies VARCHAR(100) NULL,

DNI VARCHAR(50),

illnesses VARCHAR(100) NULL,

disability VARCHAR(50) NULL,

socialWorker VARCHAR(50) NULL,

CONSTRAINT cp\_elderly PRIMARY KEY (DNI)

);

CREATE TABLE Volunteer (

name VARCHAR(50),

surNames VARCHAR(50),

userName VARCHAR(50),

password VARCHAR(50),

telephoneNumber INTEGER,

mail VARCHAR(50),

birthDate DATE,

hobbies VARCHAR(100) NULL,

DNI VARCHAR(50),

CONSTRAINT cp\_volunteer PRIMARY KEY (DNI)

);

CREATE TABLE Company (

name VARCHAR(50),

CIF VARCHAR(50),

contactPerson VARCHAR(50),

telephoneNumber VARCHAR(50),

userName VARCHAR(50),

password VARCHAR(50),

CONSTRAINT cp\_company PRIMARY KEY (CIF)

);

CREATE TABLE Address (

street VARCHAR(50),

postalCode INTEGER,

city VARCHAR(50),

country VARCHAR(50),

DNI VARCHAR(50),

CONSTRAINT cp\_address PRIMARY KEY (DNI),

CONSTRAINT ca\_address\_elderly FOREIGN KEY(DNI) REFERENCES Elderly(DNI) ON DELETE RESTRICT ON UPDATE CASCADE

);

CREATE TABLE Availability (

data DATE,

startTime TIME,

endTime TIME,

DNI\_elderly VARCHAR(50) NULL,

DNI\_volunteer VARCHAR(50),

CONSTRAINT cp\_availability PRIMARY KEY (data, startTime, endTime, DNI\_volunteer),

CONSTRAINT ca\_availability\_elderly FOREIGN KEY(DNI\_elderly) REFERENCES Elderly(DNI) ON DELETE RESTRICT ON UPDATE CASCADE,

CONSTRAINT ca\_availability\_volunteer FOREIGN KEY(DNI\_volunteer) REFERENCES Volunteer(DNI) ON DELETE RESTRICT ON UPDATE CASCADE

);

CREATE TABLE BankData (

IBAN VARCHAR(50),

bank VARCHAR(50),

branch\_office VARCHAR(50),

DNI VARCHAR(50),

CONSTRAINT cp\_bankdata PRIMARY KEY (IBAN),

CONSTRAINT ca\_bankdata\_person FOREIGN KEY(DNI) REFERENCES Elderly(DNI) ON DELETE RESTRICT ON UPDATE CASCADE

);

CREATE TABLE Contract (

id VARCHAR(50),

startDate DATE,

endDate DATE,

serviceType enum('catering', 'cleanning', 'health') NOT NULL default 'cleanning',

price FLOAT,

signatureDate DATE,

CIF VARCHAR(50),

CONSTRAINT cp\_contract PRIMARY KEY (id),

CONSTRAINT ca\_contract\_company FOREIGN KEY(CIF) REFERENCES Company(CIF) ON DELETE RESTRICT ON UPDATE CASCADE

);

CREATE TABLE Invoice (

code VARCHAR(50),

startDate DATE,

endDate DATE,

amount FLOAT,

DNI VARCHAR(50),

CONSTRAINT cp\_invoice PRIMARY KEY (code),

CONSTRAINT ca\_invoice\_elderly FOREIGN KEY(DNI) REFERENCES Elderly(DNI) ON DELETE RESTRICT ON UPDATE CASCADE,

CONSTRAINT invoice\_amount CHECK(amount>0)

);

CREATE TABLE Request (

number VARCHAR(50),

state VARCHAR(50) DEFAULT 'in process',

service VARCHAR(50),

schedule VARCHAR(50),

DNI VARCHAR(50),

idContract VARCHAR(50) NULL,

CONSTRAINT cp\_request PRIMARY KEY (number),

CONSTRAINT ca\_request\_elderly FOREIGN KEY(DNI) REFERENCES Elderly(DNI) ON DELETE RESTRICT ON UPDATE CASCADE,

CONSTRAINT ca\_request\_contract FOREIGN KEY(idContract) REFERENCES Contract(id) ON DELETE RESTRICT ON UPDATE CASCADE,

CONSTRAINT request\_service CHECK(service='catering' OR service='cleanning' OR service='health'),

CONSTRAINT request\_state CHECK(state='accepted' OR state='rejected' OR state='in process')

);

CREATE TABLE Detail (

concept VARCHAR(50),

amount FLOAT,

invoiceCode VARCHAR(50),

requestCode VARCHAR(50),

CONSTRAINT cp\_detail PRIMARY KEY (concept, amount, code),

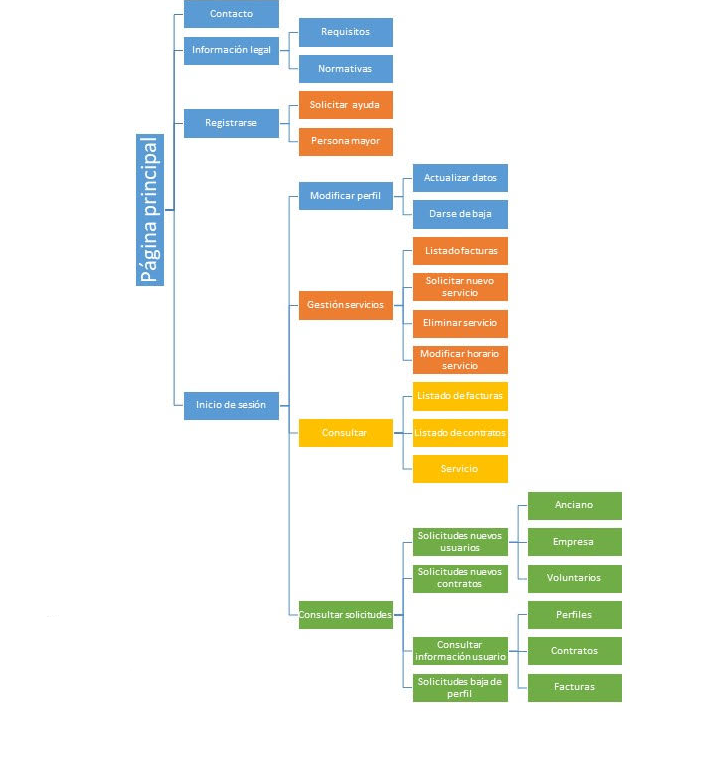
CONSTRAINT ca\_detail\_invoice FOREIGN KEY(invoiceCode) REFERENCES Invoice(code) ON DELETE RESTRICT ON UPDATE CASCADE,

CONSTRAINT ca\_detail\_request FOREIGN KEY(requestCode) REFERENCES Request(code) ON DELETE RESTRICT ON UPDATE CASCADE,

CONSTRAINT detail\_amount CHECK(amount>0)

);

# **Sitemap**



*El color naranja representa las vistas que utilizarán las personas mayores, el amarillo las vistas que utilizarán las empresas, el verde las vistas del comité y el azul las vistas comunes a todos los usuarios.*