MODELO C4

INGENIERÍA DE SOFTWARE I

2º DE GRADO EN INGENIERÍA INFORMÁTICA
CURSO 2024/2025

Dr. Francisco José García-Peñalvo / fgarcia@usal.es
Dra. Alicia García-Holgado / aliciagh@usal.es
Dra. Andrea Vázquez-Ingelmo / andreavazquez@usal.es

Departamento de Informática y Automática
Universidad de Salamanca

ÍNDICE

- La brecha entre los modelos y el código
- El modelo C4
  - Nivel de contexto
  - Nivel de contenedores
  - Nivel de componentes
  - Nivel de código
  - Notación
- C4 y los casos de uso

LA BRECHA ENTRE LOS MODELOS Y EL CÓDIGO

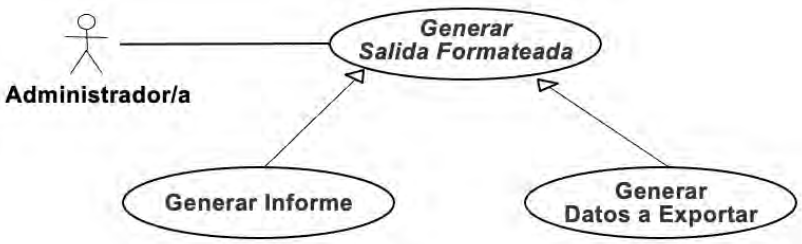
La descripción de una arquitectura software está compuesta por muchas vistas

Separación entre vistas lógicas y de desarrollo

Problemas a la hora de reflejar y trazar de forma precisa las entidades modeladas en el código

La estructura del código debería reflejar la arquitectura de nuestro sistema

LA BRECHA ENTRE LOS MODELOS Y EL CÓDIGO



¿?

¿?

```
package rentalStore;
import java.util.Enumeration;
import java.util.Vector;

class Customer {
    private String _name;
    private Vector<Rental> _rentals = new Vector<Rental>();

    public Customer(String name) {
        _name = name;
    }
    public String getMovie(Movie movie) {
        Rental rental = new Rental(new Movie("", Movie.NEW_RELEASE), 10);
        Movie m = rental._movie;
        return movie.getTitle();
    }
    public void addRental(Rental arg) {
        _rentals.addElement(arg);
    }
    public String getName() {
        return _name;
    }
```

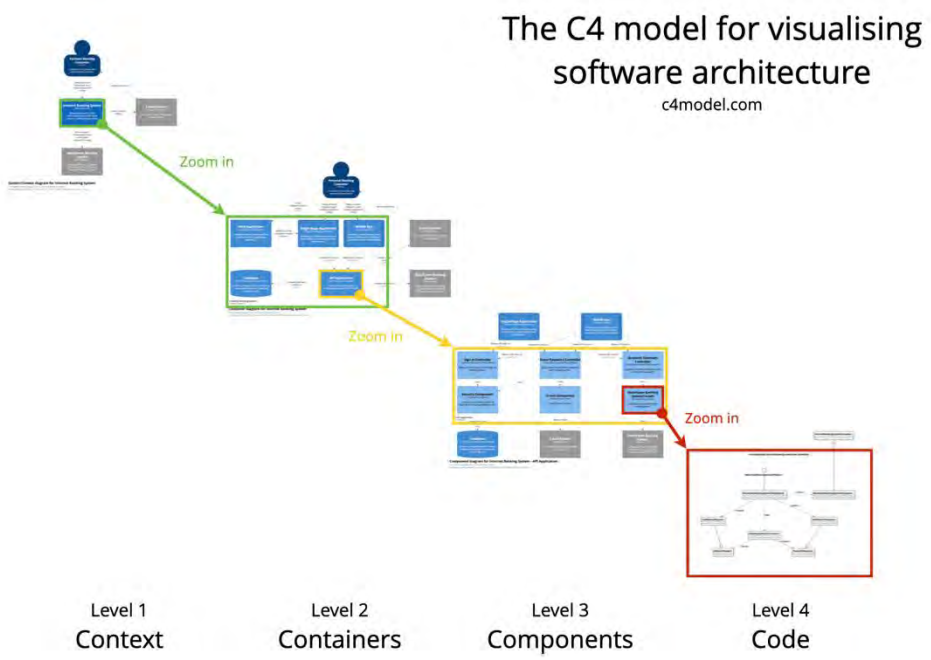
EL MODELO C4 [1-5]

Surge como solución para aliviar la brecha entre modelo y código

Permite comunicar la arquitectura de un sistema en función del detalle que se quiera proporcionar

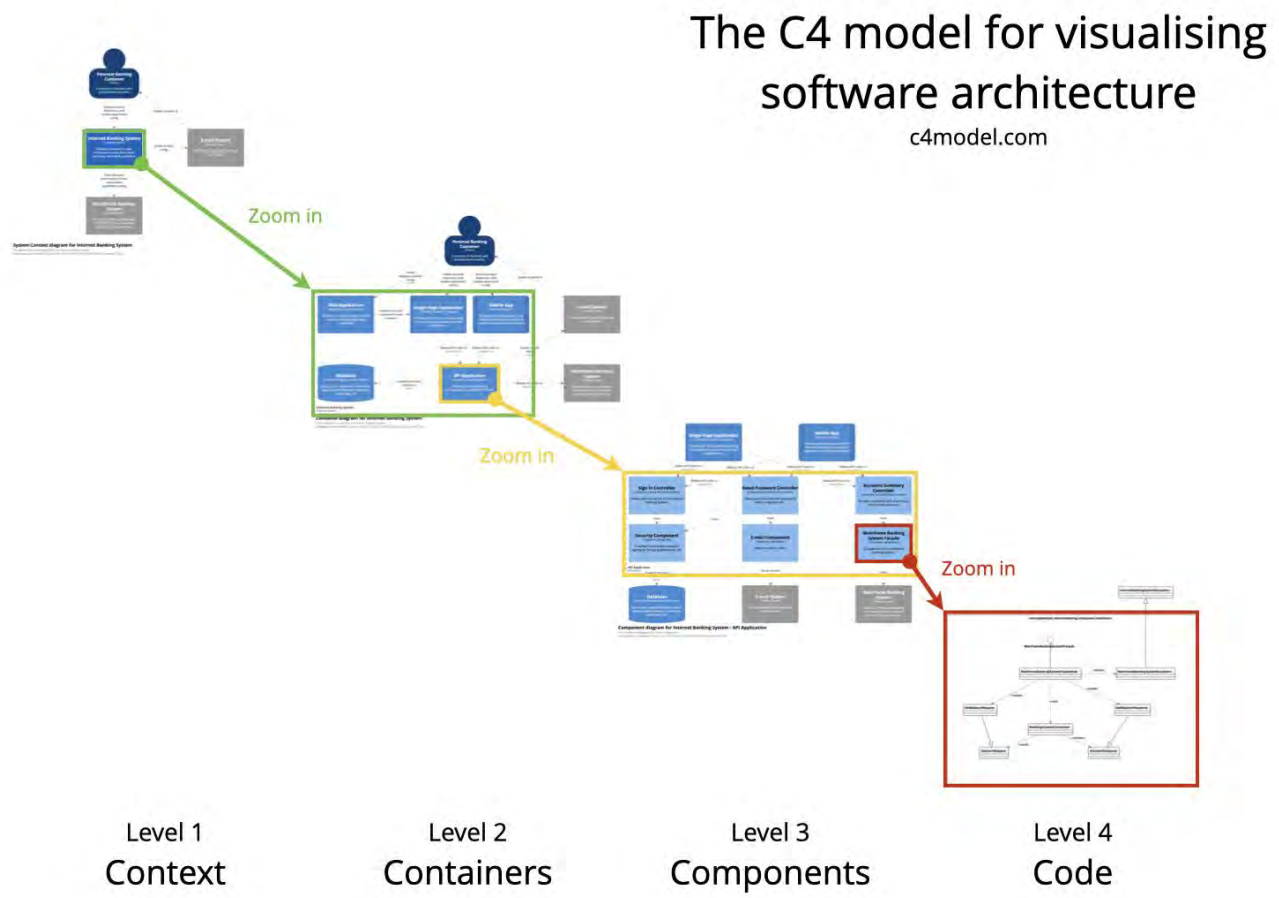
Está basado en cuatro niveles que describen el sistema con distintos grados de granularidad

- El nivel de contexto
- El nivel de contenedores
- El nivel de componentes
- El nivel de código



Rumbaugh et al. [6]

EL MODELO C4



Rumbaugh et al. [6]

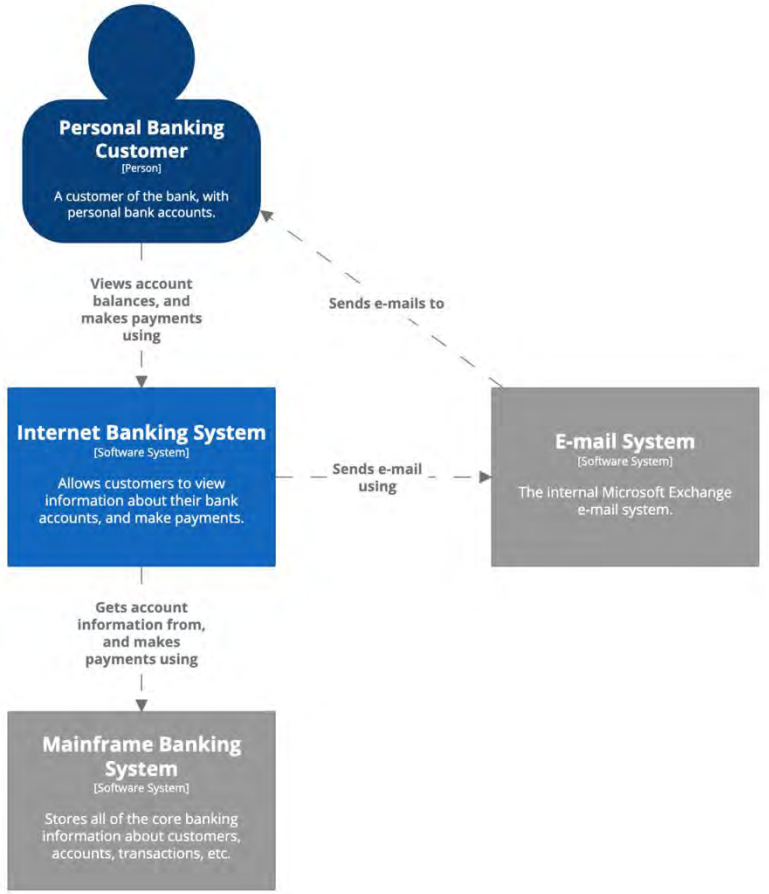
NIVEL DE CONTEXTO

Nos permite tener una imagen genérica de nuestro sistema y sus interacciones con el exterior

En este nivel podemos especificar los sistemas externos con los que interactúa nuestro propio sistema

El sistema que modelamos se considera una "**caja negra**", solo nos interesan sus relaciones externas

También permite identificar los usuarios finales que harán uso de la funcionalidad de nuestro software



https://c4model.com/

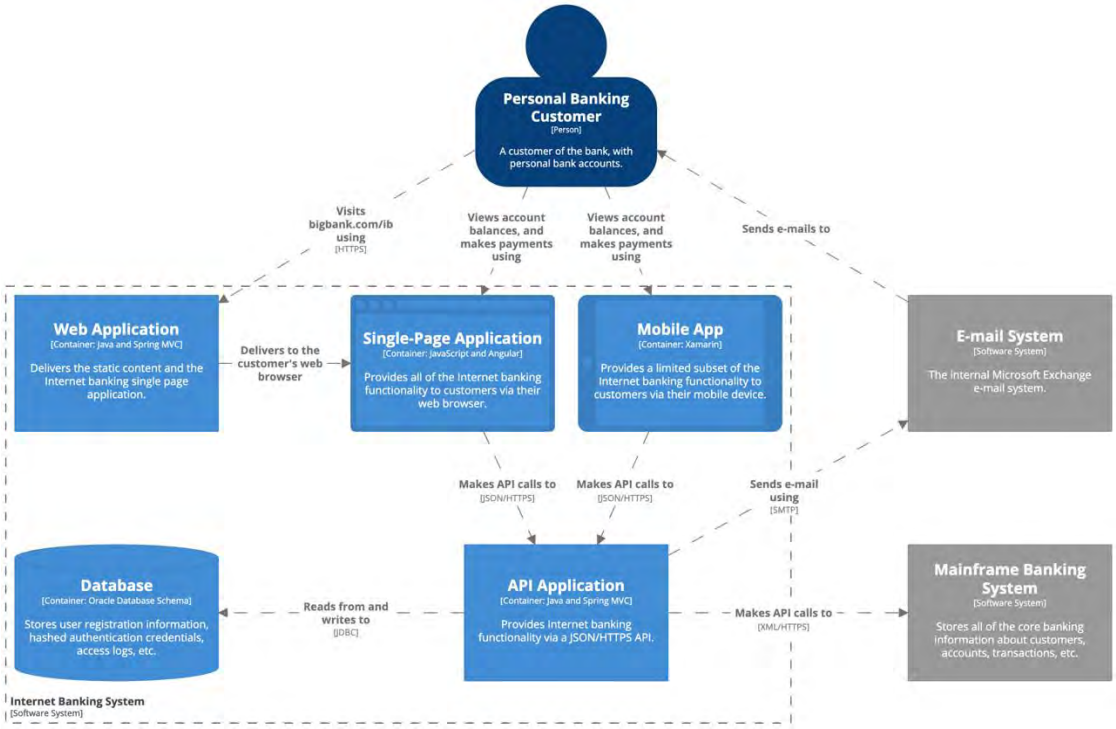
NIVEL DE CONTENEDORES

Los contenedores referencian cualquier entidad que ejecuta código o almacena datos

Pueden verse como unidades desplegables o ejecutables

Ejemplos de **contenedores**

- Aplicaciones web
- Servicios web
- Aplicaciones de escritorio
- Bases de datos
- Sistema de ficheros



https://c4model.com/

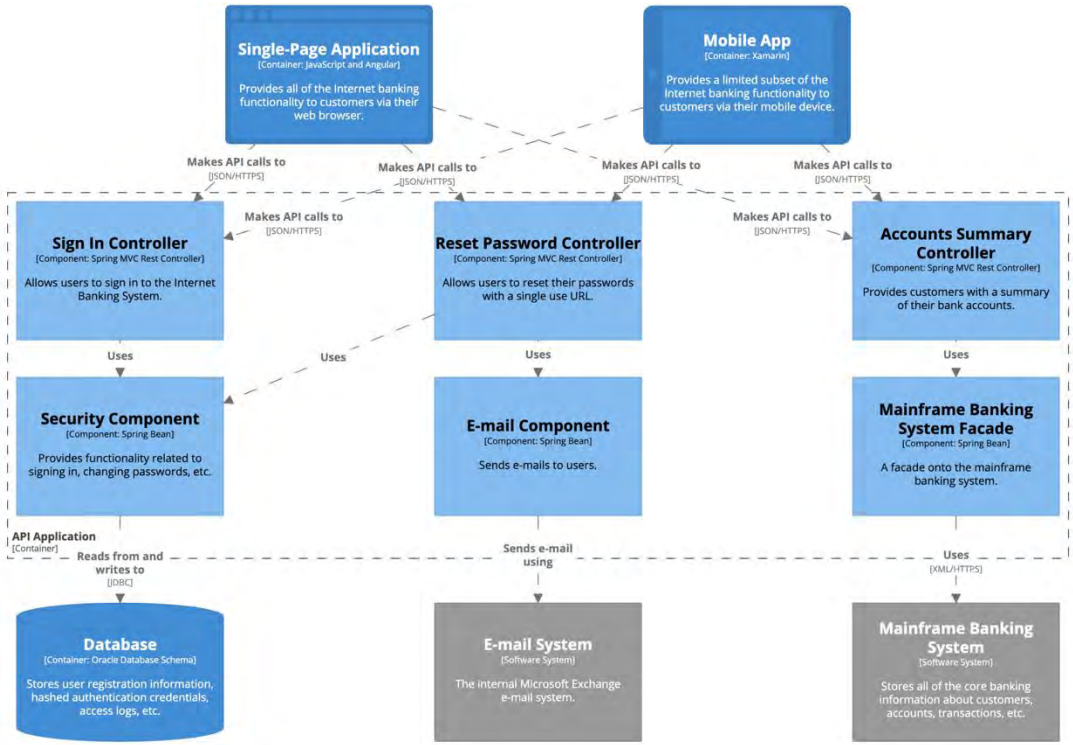
NIVEL DE COMPONENTES

Dentro de cada contenedor podemos encontrar diversos **componentes**

Los componentes representan un grupo de funcionalidades

Los componentes pueden tener relaciones entre sí y entre los usuarios finales

Muestra la **responsabilidad** de cada componente a alto nivel, así como los detalles de implementación (tecnología utilizada, etc.)

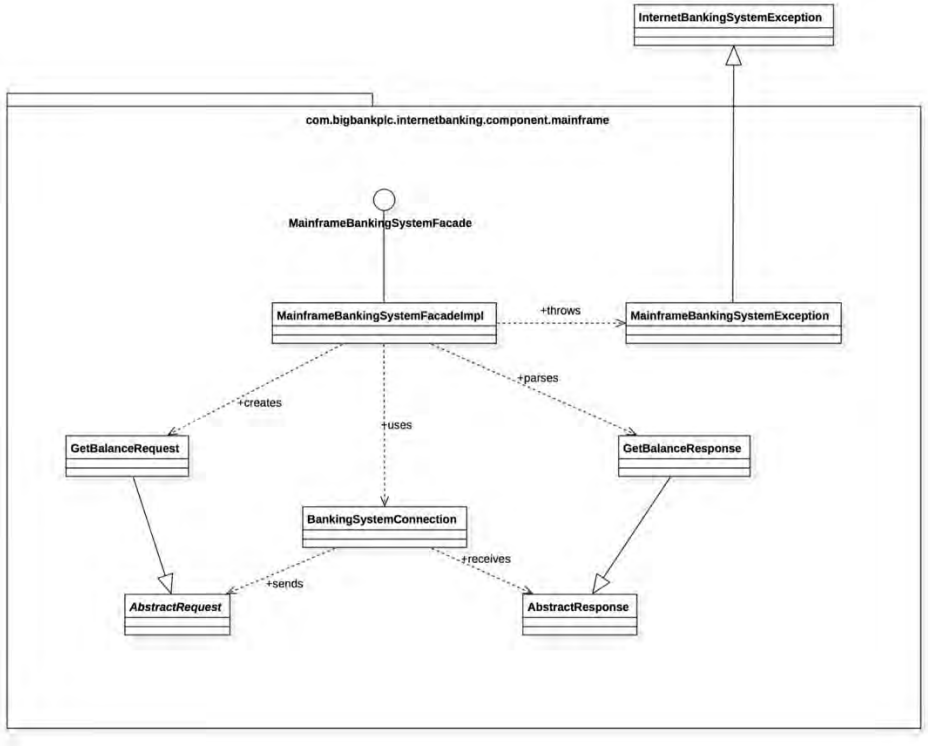


https://c4model.com/

NIVEL DE CÓDIGO

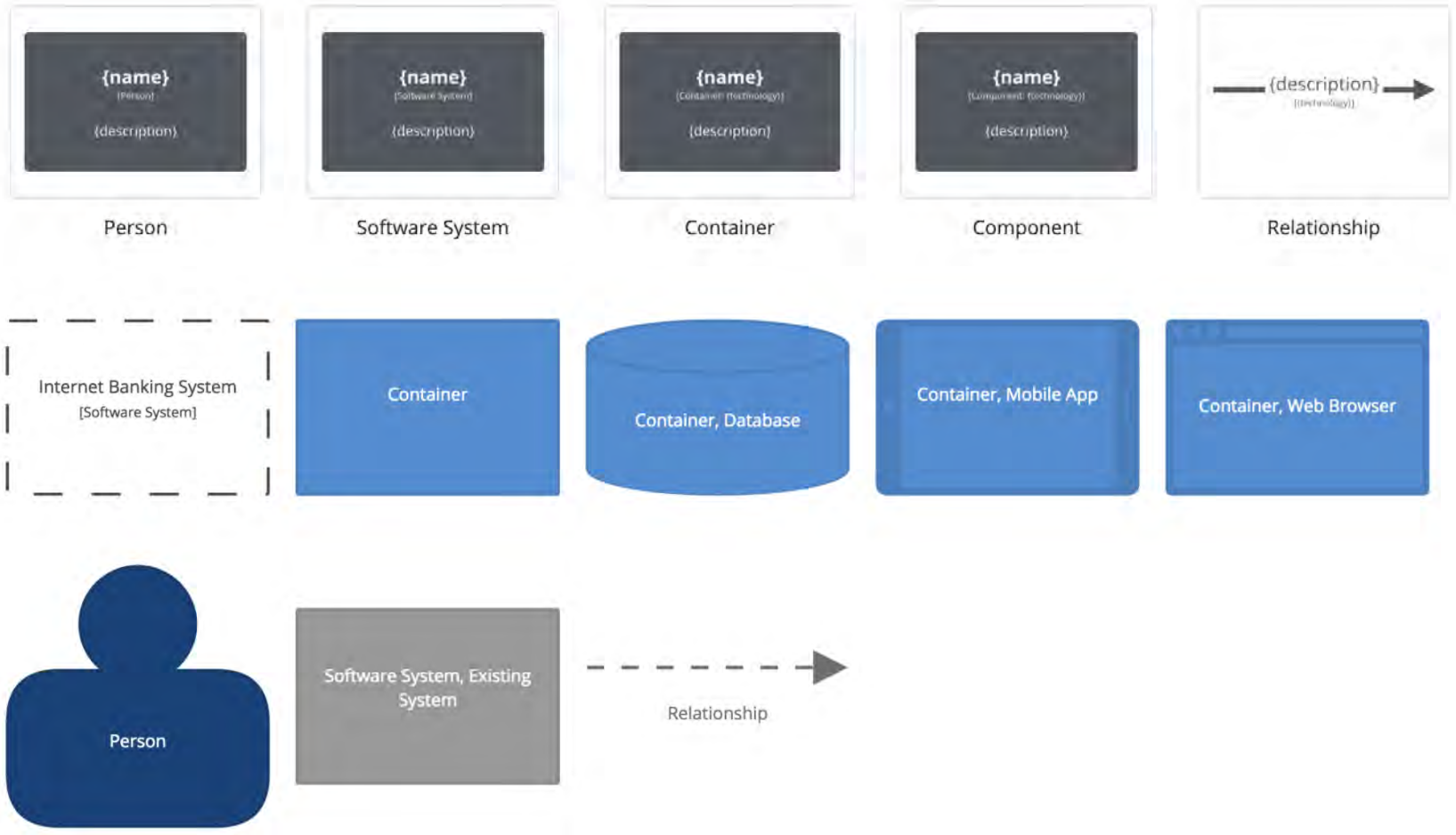
En este nivel se muestran detalles de la implementación de cada componente

Se pueden utilizar diagramas de clase, de entidad relación, o similares



https://c4model.com/

NOTACIÓN



https://c4model.com/

C4 Y LOS CASOS DE USO

Los casos de uso de UML también sirven para especificar el contexto del sistema y los actores que interactúan con los diversos bloques funcionales

Es posible transformar los diagramas de casos de uso al modelo C4

Se propone especificar mediante los niveles de contexto y contenedores la solución al taller de casos de uso

(evaluación continua)

REFERENCIAS

1. S. Brown, *Software Architecture for Developers - Volume 1. Technical leadership and the balance with agility*. Leanpub book, 2016.
2. S. Brown, *Software Architecture for Developers - Volume 2. Visualise, document and explore your software architecture*. Leanpub book, 2019.
3. S. Brown. (2022). *The C4 model for visualising software architecture. Context, Containers, Components, and Code*. Disponible en: https://c4model.com/.
4. J. Vivanco. (2019). El modelo C4 de documentación para la Arquitectura de Software. En: *Medium*. Disponible en: https://d66z.short.gy/ig0WAX.
5. A. Vázquez-Ingelmo, A. García- Holgado y F. J. García-Peñalvo, "C4 model in a Software Engineering subject to ease the comprehension of UML and the software development process," en *2020 IEEE Global Engineering Education Conference (EDUCON), (27-30 April 2020, Porto, Portugal)* pp. 919-924, USA: IEEE, 2020. doi: 10.1109/EDUCON45650.2020.9125335.
6. J. Rumbaugh, I. Jacobson y G. Booch, *The Unified Modeling Language reference manual*, 2ª ed. (Object Technology Series). Boston, MA, USA: Addison Wesley, 2005.

MODELO C4

INGENIERÍA DE SOFTWARE I

2º DE GRADO EN INGENIERÍA INFORMÁTICA
CURSO 2024/2025

Dr. Francisco José García-Peñalvo / fgarcia@usal.es
Dra. Alicia García-Holgado / aliciagh@usal.es
Dra. Andrea Vázquez-Ingelmo / andreavazquez@usal.es

Departamento de Informática y Automática
Universidad de Salamanca