



Tabla Contenido

TALLER MODELO C4.....	1
OBJETIVO.....	1
INTRODUCCIÓN.....	1
Orígenes del modelo C4.....	1
Cómo está constituido el modelo C4.....	3
Herramientas para crear el modelo C4.....	8
DESCRIPCIÓN DEL TALLER.....	9

TALLER MODELO C4

Última modificación: Febrero 18 de 2024

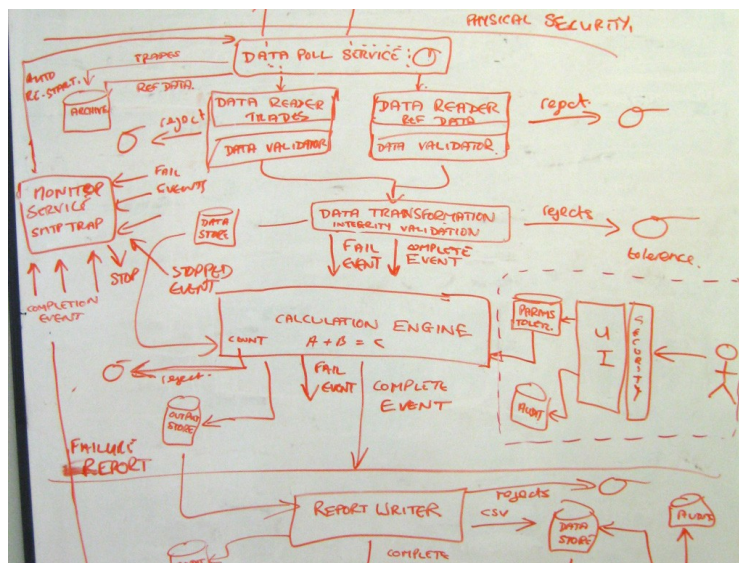
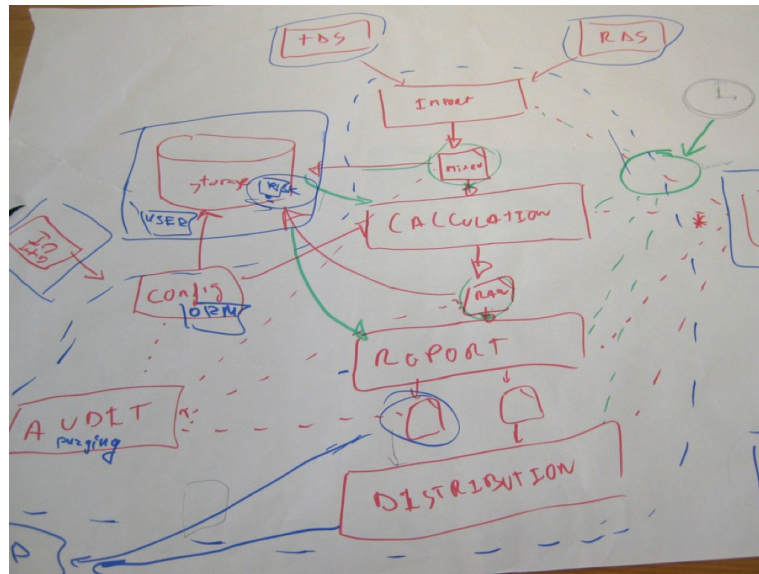
OBJETIVO

- Conocer y aplicar el modelo C4 como técnica que se centra en crear diagramas claros y concisos para comunicar la arquitectura de un sistema de software a diferentes partes interesadas, desde desarrolladores hasta gerentes

INTRODUCCIÓN

Orígenes del modelo C4

Si le pedimos a alguien del **sector de la construcción** que nos comunique visualmente la arquitectura de un edificio, obtendremos planos claros y bien detallados de las plantas, pisos, secciones transversales, etc. En cambio, si le pedimos a un **desarrollador de software** que comunique la arquitectura de un sistema de software mediante diagramas, lo más probable es que nos encontremos con un **lío confuso de cajas y líneas...** notación incoherente (código de colores, formas, estilos de línea, etc.), nombres ambiguos, relaciones sin etiquetar, terminología genérica, opciones tecnológicas ausentes, abstracciones mezcladas, etc., tal como lo muestran los siguientes modelos:



El modelo C4 es una técnica que se centra en crear *diagramas claros y concisos* para *comunicar la arquitectura* de un sistema de software a diferentes partes *interesadas*, desde desarrolladores hasta gerentes.



El modelo C4 fue creado en el 2011 por **Simon Brown**, un arquitecto de software con una amplia experiencia en el diseño y desarrollo de sistemas de software empresariales. Brown es conocido por su trabajo en arquitectura de software y por promover prácticas de diseño arquitectónico efectivas.

Cómo está constituido el modelo C4

El modelo C4 consta de cuatro niveles de abstracción: **Contexto**, **Contenedor**, **Componente** y **Clase**, cada uno proporcionando *diferentes perspectivas* de la arquitectura del sistema. Es útil para comprender la estructura y las relaciones dentro de un sistema de software y facilita la colaboración y comunicación entre equipos.

1. Diagrama de contexto. Un diagrama de contexto del sistema es un buen punto de partida para diagramar y documentar un sistema informático, ya que te permite *ver el panorama general*. Muestra el sistema como una *caja en el centro*, rodeada por sus usuarios y los otros sistemas con los que interactúa.

El detalle no es importante aquí, ya que se trata de una vista ampliada que muestra una imagen general del sistema. Hay que centrarse en las personas (actores, roles, personajes, etc.) y los sistemas de software, más que en las tecnologías, los protocolos y otros detalles de bajo nivel. Es el tipo de diagrama que se puede mostrar a personas sin conocimientos técnicos.

Ámbito de aplicación: Un único sistema de software.

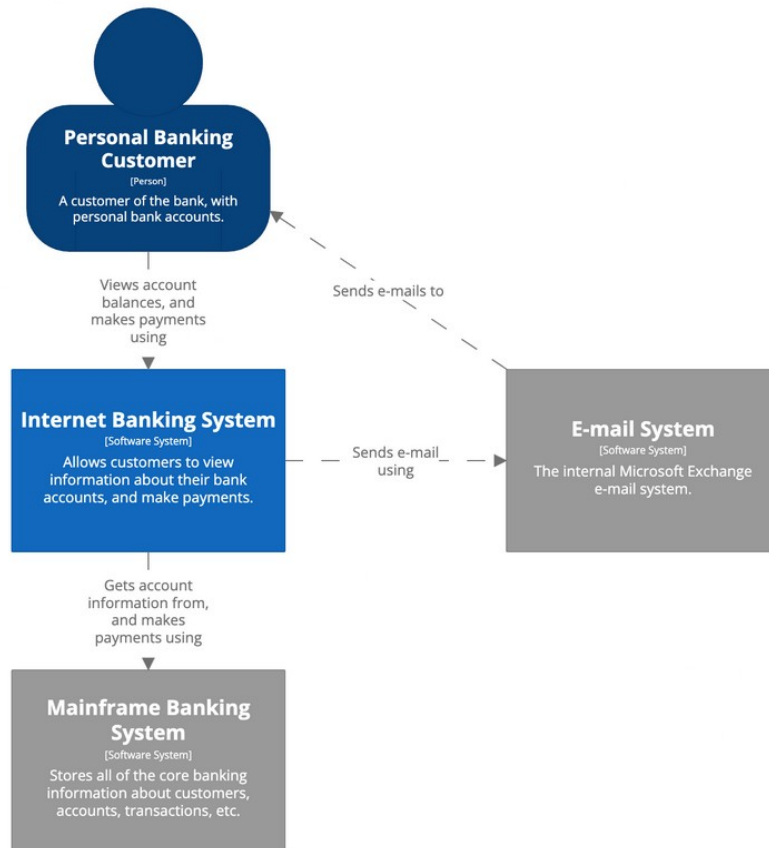
Elementos primarios: El sistema de software en el ámbito de aplicación.

Elementos de **apoyo:** Personas (por ejemplo, usuarios, actores, roles o personas) y sistemas de software (dependencias externas) que están directamente conectados con el sistema de software en el ámbito. Por lo general, estos otros sistemas de software se encuentran fuera del alcance o los límites de su propio sistema de software, y usted no tiene la responsabilidad o la propiedad de ellos.

Destinatarios: Todo el mundo, tanto técnicos como no técnicos, dentro y fuera del equipo de desarrollo de software.

Recomendado para la mayoría de los equipos: Sí.

A continuación un ejemplo de este diagrama de contexto:



2. Diagrama de contenedores. Una vez que modelado el contexto del sistema se amplía los límites del sistema con un diagrama de contenedor. Un "contenedor" es algo así como una aplicación web del lado del servidor, una aplicación de una sola página, una aplicación de escritorio, una aplicación móvil, un esquema de base de datos, un sistema de archivos, etc. Básicamente, un contenedor *es una unidad ejecutable/desplegable por separado* que ejecuta código o almacena datos.

El diagrama del contenedor muestra la forma de alto nivel de la arquitectura de software y cómo se distribuyen las responsabilidades en ella. También muestra las principales opciones tecnológicas y cómo se comunican los contenedores entre sí. Es un diagrama sencillo, centrado en la tecnología de alto nivel, que resulta útil tanto para los desarrolladores de software como para el personal de soporte y operaciones.

Ámbito de aplicación: Un único sistema de software.

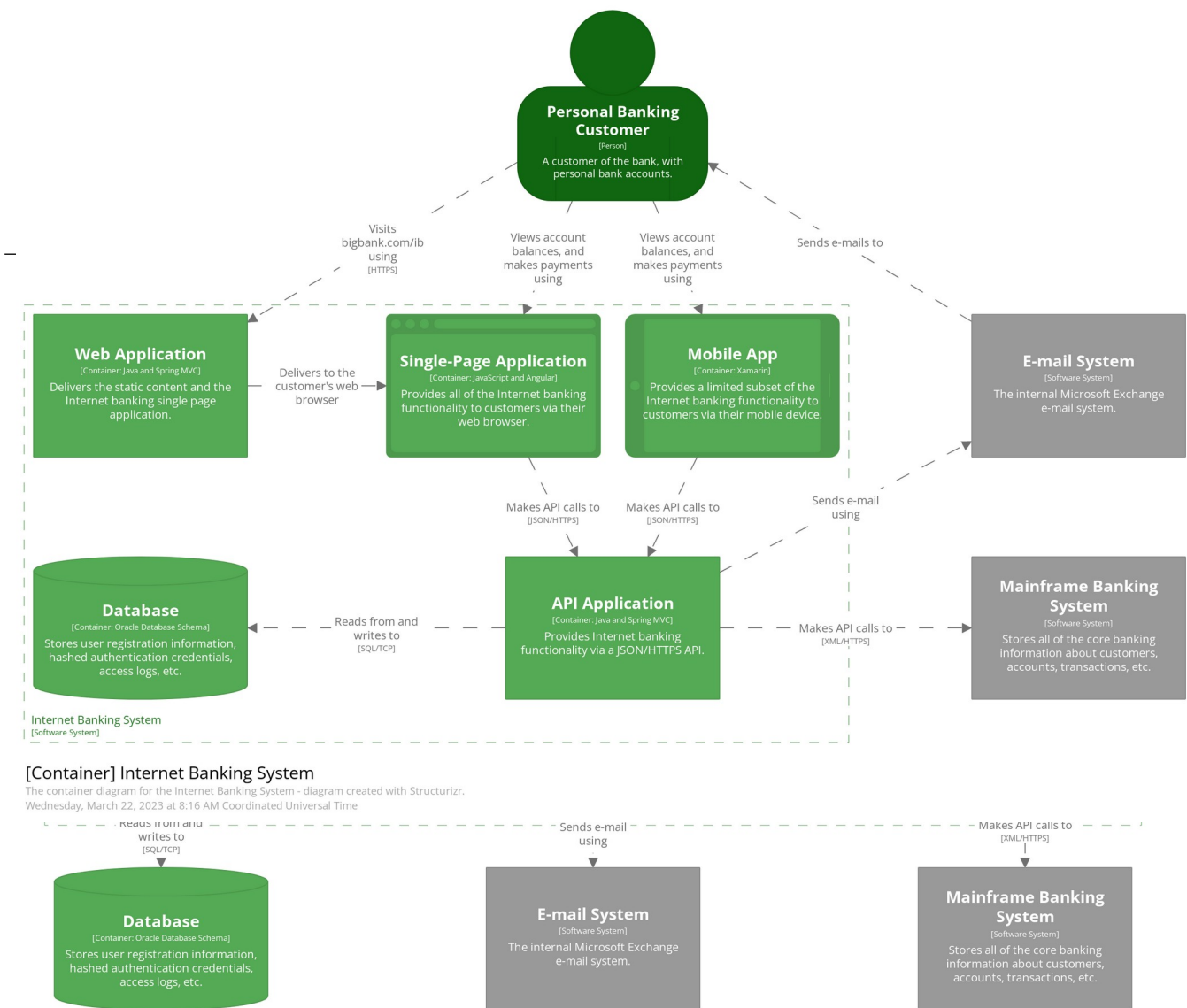
Elementos primarios: Contenedores dentro del sistema software en alcance.

Elementos de apoyo: Personas y sistemas de software directamente conectados a los contenedores.

Destinatarios: Personal técnico dentro y fuera del equipo de desarrollo de software; incluidos arquitectos de software, desarrolladores y personal de operaciones/soporte.

Recomendado para la mayoría de los equipos: Sí.

A continuación un ejemplo de diagrama de contenedores:



[Component] Internet Banking System - API Application
The component diagram for the API Application - diagram created with Structurizr.
Wednesday, March 22, 2023 at 8:16 AM Coordinated Universal Time

3. Diagrama de componentes. A continuación, se puede ampliar y descomponer aún más cada contenedor para identificar los principales *componentes estructurales* y sus interacciones. El diagrama de componentes muestra cómo un contenedor está formado por una serie de "*componentes*", cuáles son cada uno de esos componentes, sus responsabilidades y los detalles de tecnología/implementación.

Alcance: Un único contenedor.

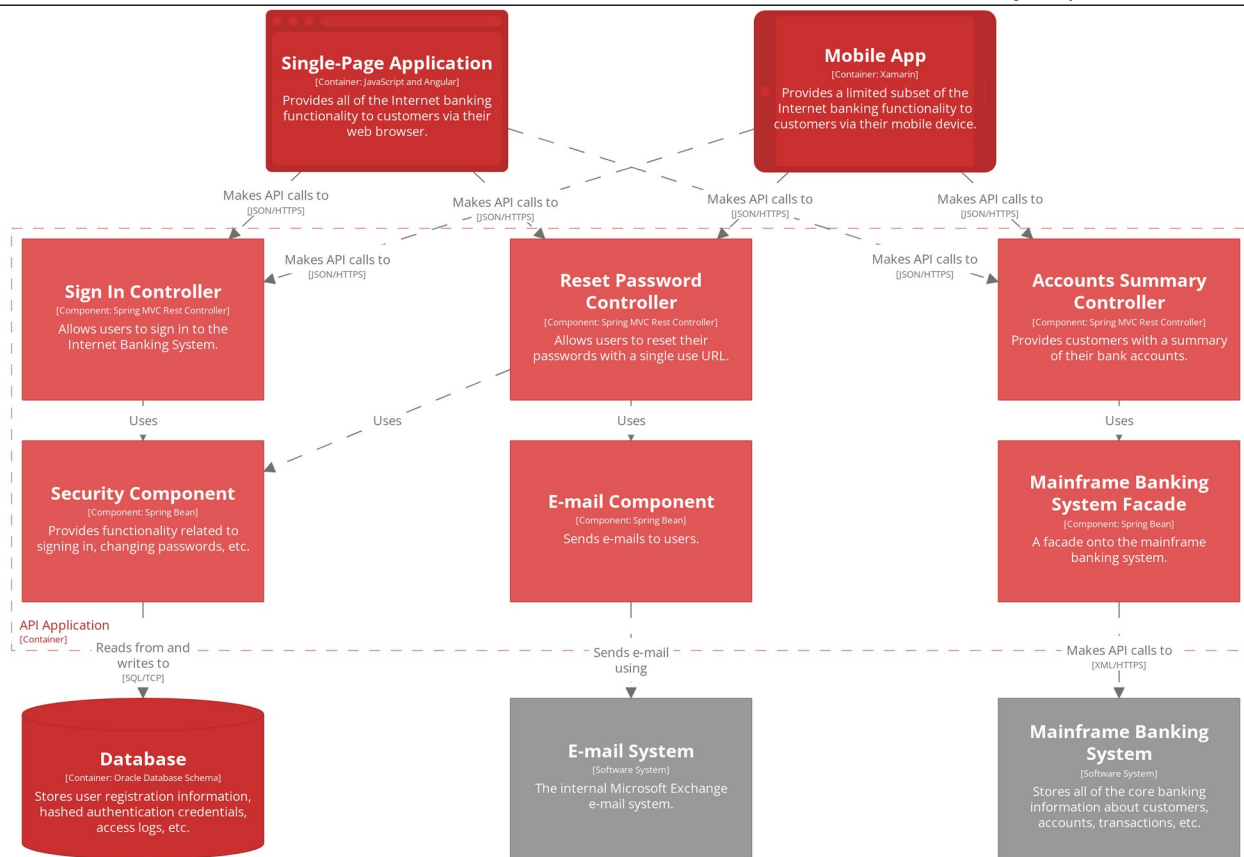
Elementos primarios: Componentes dentro del contenedor en alcance.

Elementos de apoyo: Contenedores (dentro del sistema de software en alcance) más personas y sistemas de software directamente conectados a los componentes.

Destinatarios: Arquitectos y desarrolladores de software.

Recomendado para la mayoría de los equipos: No, sólo crea diagramas de componentes si crees que añaden valor, y considera automatizar su creación para una documentación de larga duración.

A continuación un ejemplo de un diagrama de componentes:



[Component] Internet Banking System - API Application
 The component diagram for the API Application - diagram created with Structurizr.
 Wednesday, March 22, 2023 at 8:16 AM Coordinated Universal Time

4. Diagrama de código (o de clases). Por último, se puede hacer zoom en cada componente para mostrar cómo se implementa como código; utilizando diagramas de clases UML, diagramas de relación de entidades o similares.

Se trata de un nivel de detalle opcional que suele estar disponible en herramientas como los IDE. Lo ideal sería que este diagrama se generara automáticamente mediante herramientas (por ejemplo, un IDE o una herramienta de modelado UML), y debería considerar mostrar sólo aquellos atributos y métodos que le permitan contar la historia que desea contar. No se recomienda este nivel de detalle salvo para los componentes más importantes o complejos.

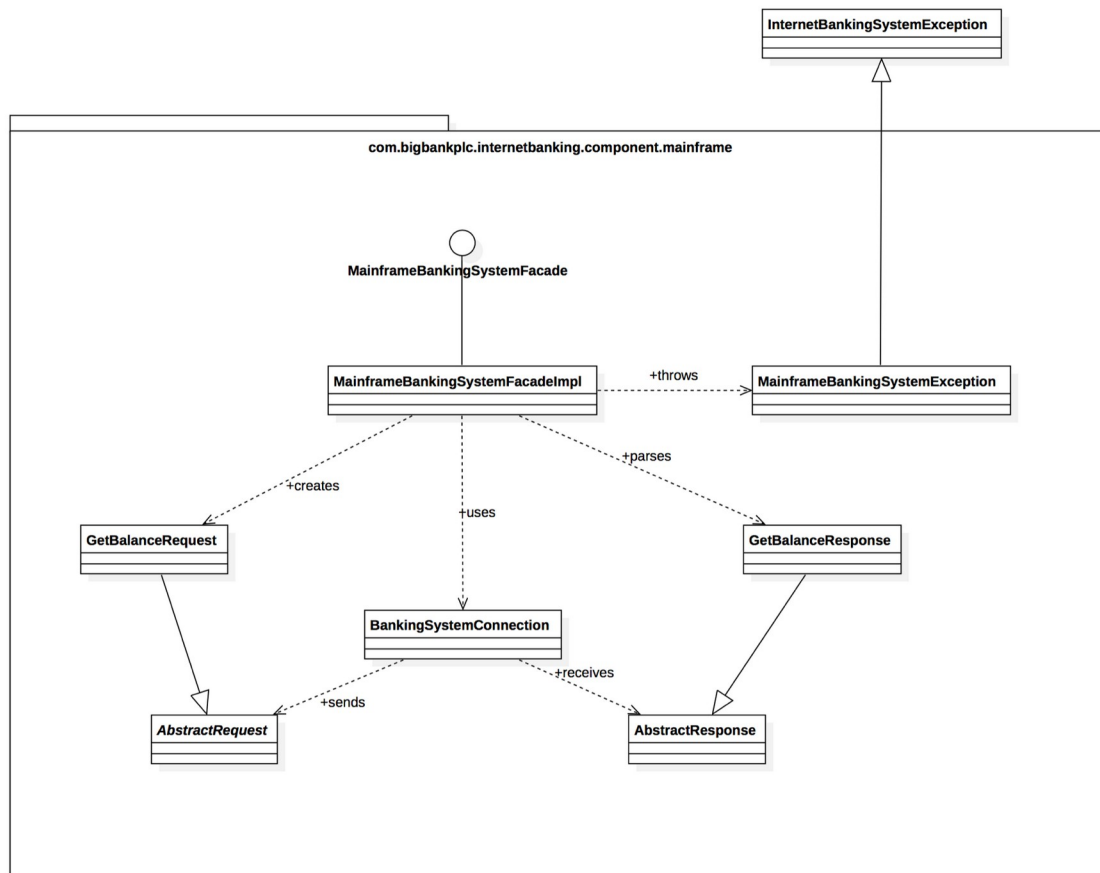
Ámbito de aplicación: Un único componente.

Elementos primarios: Elementos de código (por ejemplo, clases, interfaces, objetos, funciones, tablas de bases de datos, etc.) dentro del componente en alcance.

Destinatarios: Arquitectos y desarrolladores de software.

Recomendado para la mayoría de los equipos: No, sobre todo para la documentación de larga duración, ya que la mayoría de los IDE pueden generar este nivel de detalle bajo demanda.

A continuación un ejemplo de un diagrama de clases:



Herramientas para crear el modelo C4

Existen varias herramientas de software que puedes utilizar para trabajar con el modelo C4 y crear diagramas de arquitectura. Algunas de las más populares son:

- **Draw.io:** Es una herramienta de diagramación gratuita y de código abierto que puedes utilizar para crear diagramas basados en el modelo C4. Es fácil de usar y se puede integrar con varias plataformas de almacenamiento en la nube.
- **Structurizr:** Es una plataforma en línea que te permite crear y compartir diagramas de arquitectura basados en el modelo C4. Proporciona una interfaz intuitiva y funcionalidades para colaborar en tiempo real.
- **Lucidchart:** Esta herramienta de diagramación en línea es muy versátil y ofrece plantillas específicas para el modelo C4. Puedes crear diagramas con facilidad y compartirlos con tu equipo.

- **Visual Paradigm:** Ofrece una amplia gama de herramientas para modelado de arquitectura y soporta el modelo C4. Permite crear diagramas con diferentes niveles de detalle y ofrece opciones de colaboración en equipo.

DESCRIPCIÓN DEL TALLER

En este enlace encuentras un [ejemplo en Google Drive](#) que modela la arquitectura de un parqueadero online. Los diagramas los puedes abrir en la herramienta [Draw.io](#). Basados en este ejemplo del parqueadero y en el ejemplo de la Sección de introducción, diseñar la arquitectura de la aplicación Hyun Seda utilizando el modelo C4. Proponer una solución monolítica utilizando una arquitectura en capas (presentación, dominio y acceso a datos). Se debe entregar:

- El diagrama de contexto
- El diagrama de contenedores
- Un diagrama de componentes de un contenedor importante.
- Un diagrama de clases de un componente importante o complejo.

Trabajar en los equipos de trabajo ya definidos. Entregar los diagramas en un documento PDF.