

Universidad Carlos III

Curso Ingeniería del Software 2020-21

Práctica  
Curso 2020-21

INSTAJOBS

Fecha: **27/10/2020** - ENTREGA: **07/12/2020**

GRUPO: **81**

EQUIPO: **09**

Alumnos: **Jorge Rodríguez Fraile / Carlos Rubio Olivares**

**Victor Manuel Qiu / María Sánchez Blázquez**

PÁGINA DE ESTADO DE DOCUMENTO

Registro de Cambios

|  |  |  |  |
| --- | --- | --- | --- |
| Versión | Fecha | Autor | Cambios en la versión |
| 0 | 17-10-2020 | Equipo | Primer borrador, secciones 1 completa, 2.1. |
| 0.1.1 | 18-10-2020 | Jorge | 2.2 |
| 0.1.2 | 19-10-2020 | Equipo | Secciones 2.3, 2.4, 2.5. |
| 0.2.1 | 20-10-2020 | Victor | Sección 2.6 |
| 0.2.2 | 22-10-2020 | Equipo | Sección 3 completa |
| 0.3.1 | 23-10-2020 | Carlos | Corrección de requisitos. |
| 0.3.2 | 24-10-2020 | Equipo | Pequeñas correcciones y sección 5.1 |
| 0.4 | 26-10-2020 | María | Revisión de errores |
| 1 | 27-10-2020 | Equipo | Revisión preentrega 1. |

Horas invertidas en el proyecto

|  |  |  |  |
| --- | --- | --- | --- |
| NOMBRE | INDIVIDUAL | EQUIPO | TOTAL |
| Jorge Rodríguez Fraile | 5 | 13 | **18** |
| Carlos Rubio Olivares | 2 | 13 | **15** |
| Victor Manuel Qiu Pan | 3 | 15 | **18** |
| María Sánchez Blázquez | 3 | 13 | **16** |
| TOTAL | **13** | **54** | **67** |

Distribución de responsabilidades

|  |  |
| --- | --- |
| NOMBRE | Descripción de sus principales contribuciones al proyecto |
| Jorge Rodríguez Fraile | Requisitos no funcionales, sección 2, 3 y 5, revisión del proyecto, coordinación de requisitos. |
| Carlos Rubio Olivares | Requisitos funcionales, sección 1, 3 y 5 revisión del proyecto, control de errores, coordinador del proyecto. |
| Victor Manuel Qiu Pan | Requisitos no funcionales, sección 1, 2 y 3, revisión del proyecto. |
| María Sánchez Blázquez | Requisitos funcionales, sección 1, 2, 5 y revisión del proyecto, coordinación de corrección. |

TABLA DE CONTENIDO

[**1**](#_heading=h.1t3h5sf) **Introducción 5**

[1.1](#_heading=h.26in1rg) Propósito del documento 5

[1.2](#_heading=h.z337ya) Visión general del documento 5

[1.3](#_heading=h.2xcytpi) Referencias 6

[**2**](#_heading=h.3whwml4) **Descripción general 7**

[2.1](#_heading=h.d8xiwwle9mpb) Perspectiva del producto 7

[2.2](#_heading=h.2bn6wsx) Alcance del software 9

[2.3](#_heading=h.qsh70q) Capacidades generales 10

[2.4](#_heading=h.3as4poj) Restricciones generales 11

[2.5](#_heading=h.1pxezwc) Características de los usuarios: roles y capacidades 11

[2.6](#_heading=h.lmgizo5ev5ft) Entorno operacional 12

[**3**](#_heading=h.49x2ik5) **Requisitos 14**

[3.1](#_heading=h.2p2csry) Justificación de la clasificación de requisitos 14

[3.2](#_heading=h.147n2zr) Justificación de la plantilla de requisitos 15

[3.3](#_heading=h.3o7alnk) Requisitos funcionales 15

[3.4](#_heading=h.23ckvvd) Requisitos no funcionales 19

[3.5](#_heading=h.ihv636) Vocabulario del dominio 22

[3.5.1](#_heading=h.i9jqmgk2yiuh) Vocabulario del modelo conceptual 22

[3.5.2](#_heading=h.6roe2eumncwc) Vocabulario técnico 22

[**4**](#_heading=h.32hioqz) **Arquitectura 23**

[4.1](#_heading=h.1hmsyys) Modelo conceptual 23

[4.2](#_heading=h.41mghml) Modelo de implementación 24

[4.3](#_heading=h.ti55rvd4y97v) Especificación del diseño de componentes 25

[4.3.1](#_heading=h.l6xxmkg95eec) Componente A (repetir para cada componente) 25

[**5**](#_heading=h.2grqrue) **Listas y tablas 27**

[5.1](#_heading=h.u9psszbd8ru6) Consistencia entre requisitos: conflictos, redundancias, acoplamientos 27

[5.2](#_heading=h.z6xrdwy7p8x5) Trazabilidad requisitos – modelo conceptual (clases) 27

[5.3](#_heading=h.a7qxylc4gw0c) Trazabilidad requisitos – modelo de implementación (componentes) 27

# Introducción

## Propósito del documento

Es un documento de especificaciones de la aplicación desarrollada con el propósito de cumplir con los requisitos de los usuarios y con los objetivos de negocio de dicho sistema.. El documento servirá de guía y de diario de bitácora para poder establecer los márgenes de la aplicación y en qué contexto se está creando. Esperamos que la siguiente información sea lo suficientemente útil para poder entender la aplicación correctamente y no genere ningún tipo de desinformación o malentendido.

Este documento va dirigido a los clientes que han encargado la creación de esta aplicación de trabajo, de modo que, en este, se definan de una manera formal los requisitos, capacidades y restricciones que deberá de poseer la aplicación. Este documento servirá también como contrato de conformidad con el cliente, de esta manera al firmarlo, quede por escrito la conformidad por parte del cliente con los requisitos y el tiempo de desarrollo especificados.

## Visión general del documento

En las siguientes páginas, se podrá encontrar una descripción general del proyecto (objetivos, desarrollo), una lista de requisitos necesarios para crear la aplicación (tanto funcionales, como no funcionales) e incluso una revisión de la arquitectura del proyecto. También se podrán encontrar datos estadísticos de las horas de trabajo invertidas en este documento, a parte una revisión estructural de la propia aplicación, comparándola con otras aplicaciones existentes en cuanto a eficiencia, tecnología, etc.

Este documento está dividido en varias partes: la introducción, la descripción general, los requisitos, la arquitectura y, por último, las listas de tablas.

La introducción servirá para introducir al lector qué es este documento, su propósito y cómo está organizado. En la siguiente parte que será la descripción general, se procederá a explicar sin entrar en detalles, qué es el producto y cuáles son las capacidades y restricciones que debe de tener.

A continuación, se procederá con la parte de requisitos, en la que se establecerá los requisitos tanto funcionales como no funcionales que debe de tener la aplicación para cumplir con las características y restricciones definidas en el apartado anterior.

En la parte de arquitectura, se procederá a entrar en una parte más técnica del documento, donde se establecerá el modo de implementación que se usará para la creación de la aplicación, definiendo los componentes que deberá de poseer esta.

Por último, en la sección de listas y tablas, se especificará de una forma más visual, los componentes necesarios que deberá de tener nuestra aplicación, así como su modo de implementación.

## Referencias

[1] LinkedIn (2020). [online] Disponible en: <https://www.linkedin.com/> [Acceso: 27/10/2020]

[2] InfoJobs (2020). [online] Disponible en: <https://www.infojobs.net/> [Acceso: 27/10/2020]

[3] Job Today (2020). [online] Disponible en: [https://jobtoday.com/](https://jobtoday.com/es/trabajos/madrid) [Acceso: 27/10/2020]

[4] [Web Accessibility Initiative (2020)](https://www.w3.org/WAI/). [online] Disponible en: <https://www.w3.org/WAI/> [Acceso: 27/10/2020]

[5] Hobo. [online] Disponible en: <https://www.hobo-web.co.uk/best-screen-size/> [Acceso: 27/10/2020]

[6] CABRERA, Roman; DEL ROCIO, Rolendy. Efectos del uso del celular en términos de seguridad informática en millennials al tratar sus datos personales. 2020.

[7] Draw.io [online] Disponible en: <https://app.diagrams.net/>[Acceso: 23/11/2020]

[8] Lucid [online] Disponible en: <https://www.lucidchart.com/> [Acceso: 23/11/2020]

[9] Protección de Datos Personales y garantía de los derechos digitales [[BOE-A-2018-16673](https://www.boe.es/buscar/doc.php?id=BOE-A-2018-16673)] Disponible en: <https://www.boe.es/eli/es/lo/2018/12/05/3/con> [Acceso: 23/11/2020]

[10] Google Drive [online] Disponible en: <https://drive.google.com/> [Acceso: 23/11/2020]

[10] Requisitos SMART [online] Disponible en: <https://www.win.tue.nl/~wstomv/edu/2ip30/references/smart-requirements.pdf> [Acceso: 23/11/2020]

[11] Ley del Estatuto de los Trabajadores [[BOE-A-2015-11430](https://www.boe.es/buscar/doc.php?id=BOE-A-2015-11430)] Disponible en:

https://www.boe.es/eli/es/rdlg/2015/10/23/2/con [Acceso: 23/11/2020]

# Descripción general

## Perspectiva del producto

Estas webs han sido elegidas por ser grandes referentes en el mercado de búsqueda virtual de trabajo. Por otro lado, todas enfocan esta idea de diferentes maneras, lo que puede resultar en una gran mezcla de diseños para nuestra app, además de poder tener diversos enfoques a la vez.

**[1] LinkedIn:**

* Reconocimiento de estudiantes: Si eres estudiante, debes rellenar tus datos sobre dónde y qué estudias.
* Verificación: Para poder crear un perfil, debes verificar tu identidad mediante un email.
* Relación con el mundo laboral: LinkedIn trata las relaciones de trabajo como algo más social, aunque manteniendo un estándar profesional. Posee un sistema de contactos e incluso una timeline para poder acceder a publicaciones de otros usuarios o empresas e interaccionar con ellas.
* Búsqueda de trabajos: Se utilizan los datos de tus contactos, ubicación y publicaciones para recomendar ofertas de trabajo personalizadas, supone una utilización de datos personales, aunque aporta una personalización mayor. Aun así, no posee filtros para poder crear búsquedas más exactas.
* Personalización de perfil: La personalización en esta web es bastante amplia, ya que deja que establezcas las metas alcanzadas tanto en la vida académica como laboral, además, te permite establecer tus puntos fuertes trabajando e incluso da la opción de mostrar enlaces a proyectos realizados anteriormente.
* Accesibilidad: Esta aplicación permite acceder al mundo laboral de manera bastante simple, incluso permitiendo descargar tu perfil en PDF. También ofrece una serie de cursos para familiarizarse con el mundo laboral.
* Claridad de la oferta: En cuanto a las ofertas, la web es bastante específica en cuanto a la definición de la oferta, explica de manera extensa todos los requerimientos del trabajo, su salario, jornada de trabajo, posibles ascensos, etc.
* Comunicación: La comunicación con el empleador es bastante más clara de lo esperado, se tiene una línea directa de intercambio de mensajes ininterrumpida, con el que se puede responder a dudas inmediatas sin necesidad de esperar a envíos de emails o derivados.

**[2] InfoJobs**

* Reconocimiento de estudiantes: Permite introducir la formación que tiene el usuario sea del tipo que sea, además comprueba que los estudios y centros estén validados.
* Verificación: Para acceder a esta aplicación se debe proporcionar obligatoriamente un teléfono y mail de contacto, además de la fecha de nacimiento para comprobar que se esté en edad legal para trabajar.
* Relación con el mundo laboral: Te permite introducir tu empleo actual para buscar trabajos relacionados para cuando se te acabe el contrato, también aparece una opción para indicar que nunca has trabajado, en este caso te pregunta por tu trabajo ideal.
* Búsqueda de trabajos: Te permite recibir notificaciones cuando se oferte un trabajo que se ajuste a tus intereses y cuando se encuentra una oferta interesante, te permite inscribirte mandando tu currículum y una carta de presentación. Estas ofertas se pueden filtrar por salario, estudios, jornada, localización y experiencia.
* Personalización de perfil: El usuario puede crear como perfil un currículum virtual, y para configurarlo se hacen preguntas clave para ir completando la información. Algunas preguntas son sobre experiencias, estudios, habilidades, fortalezas, etc.
* Accesibilidad: Al comenzar en la aplicación, te pregunta en qué idioma quieres utilizarla, también tiene un diseño sencillo con la información más relevante a simple vista y menús con pocas opciones bien indicadas.
* Claridad de la oferta: El empleador debe proporcionar información sobre qué trata el trabajo, dónde se lleva a cabo, cuál es el salario mensual o anual, qué jornada se oferta, contra cuántas personas compites y cuáles son los requisitos para el puesto tanto de estudios como capacidades.
* Comunicación: El método para comunicarte con otros usuarios o el empleador es mediante opiniones en la propia oferta, pero podemos recibir notificaciones de empleos que nos puedan interesar o empleadores que buscan un determinado perfil de persona. Te actualizan todas las mañanas con el estado del mercado que te interesa.

**[3] JobToday**

* Reconocimiento de estudiantes: No hay reconocimiento de estudiantes, porque la oferta de empleo se centra en usuarios que se encuentran ya formados.
* Verificación: No existe ningún mecanismo para verificar la identidad del usuario, permitiendo la posibilidad de crear múltiples cuentas por usuario.
* Relación con el mundo laboral: La aplicación permite al usuario introducir su experiencia laboral de forma completa, dando bastante información útil al empleador.
* Búsqueda de trabajos: El buscador está muy logrado con la inclusión de geolocalización, categorías y diversos filtros permitiendo al usuario buscar trabajo de forma rápida, eficaz y con incorporación inmediata.
* Personalización de perfil: La creación de un perfil es sencilla, con posibilidad de agilizar el proceso con una cuenta de Apple o Google. La descripción del perfil se puede rellenar de forma simple gracias a que la aplicación pide unos determinados datos del usuario.
* Accesibilidad: El diseño de la interfaz está muy logrado, con una buena combinación de colores junto a unas barras de navegación que facilitan al usuario desplazarse por toda la aplicación con facilidad.
* Claridad de la oferta: La aplicación obliga al empleador a proporcionar información acerca de la ubicación, experiencia requerida, tipo de jornada y cuando se necesita la incorporación.
* Comunicación: Job Today presenta un sistema de mensajería interna que permite la comunicación entre el usuario y el empleador.

Una vez analizadas estas plataformas, Instajobs se diferencia de la competencia centrándose en un nicho de mercado concreto: estudiantes, personas con necesidad de un pequeño apoyo económico.

Estos usuarios se caracterizan por no querer compromisos, y por querer de forma inmediata trabajo. Mediante nuestra aplicación, hacemos una selección de unas ofertas en base a los ajustes del usuario y junto a un canal de comunicación interna agilizamos el proceso en el que un usuario encuentra trabajo.

El modelo económico se basará simplemente en cobrar a las empresas que deseen publicar más de dos ofertas y el cobro de tasas por destacar ofertas dentro de la plataforma.

## Alcance del software

Los productos software necesarios para poner en funcionamiento ***Instajobs***son:

* Servidores y dominio para la plataforma
* Diseño de una Web y App móvil.
* Base de datos de usuarios y ofertas.
* Sistema de registro con soporte para Linkedin, Facebook, Google Accounts.
* Sistema inicio de sesión.
* Sistema de gestión de ofertas que incluya publicación y valoración.
* Canal de comunicación empleador-usuario: chat y correo electrónico.
* Motor de búsqueda interna con filtros.
* Procesador de pago para aceptar distintos métodos de pago como Paypal y tarjetas bancarias.
* Sistema de perfil que incluya CV, historial de trabajos del usuario y formación del usuario.
* Un sistema de Newsletter.

El objetivo es la creación de una aplicación en la cual las empresas puedan publicar ofertas para trabajos concretos, y en la que, los usuarios puedan ver estas ofertas y, si alguno estuviera interesado en alguna de ellas, poder contactar con dicha empresa, creándose así un canal de comunicación rápido y sencillo entre la empresa y el usuario interesado.

La meta de este proyecto es la creación de una aplicación para trabajos concretos, de esta forma las empresas que necesiten algún trabajador para una tarea en específico, podrán describir en la aplicación el perfil del trabajador que están buscando y, así poder encontrarlo de forma rápida y fácil, sin tener que esperar que alguien vaya de manera física a la propia empresa a entregar el currículum.

Este proyecto, también beneficiará al usuario, ya que este no tendrá que ir de forma física hasta una empresa a entregar un currículum sin saber tan siquiera qué perfil de trabajador están buscando, sino que le será tan sencillo como registrarse en la aplicación y ver todas las ofertas de trabajo disponibles, hasta encontrar una que le interese y contactar con la empresa que publicó la oferta.

Deberemos asegurar que nuestra aplicación debe cumplir la legislación actual sobre la protección de datos [9] y entorno laboral en el estado español, por lo que tendremos que filtrar las ofertas que se publiquen siguiendo estos criterios, si se cumplen, la oferta podrá aparecer en la web sin problema alguno.

En este aplicación se pondrá especial atención a la ética, por lo que los datos que se consigan de los usuarios, como pueden ser los currículum, se mantendrán de forma privada para el resto de usuarios. Por otro lado, buscamos el buen posicionamiento del usuario en el mercado laboral, por lo que ofertas que intenten explotar al usuario laboralmente y no aseguren unas condiciones de trabajo dignas no serán publicadas en la app. También se mantendrá de forma privada, las estadísticas de cada empresa, como pudieran ser las estadísticas de los trabajos más demandados por los usuarios. Capacidades generales

Uno de los objetivos principales de esta aplicación es ofrecer un sistema de minitrabajos para estudiantes, donde podrán encontrarlos de manera sencilla teniendo en cuenta en qué zona residen, y sus estudios. Es importante recalcar que buscamos aportar un ambiente profesional, donde los usuarios se sientan confiados y seguros a la hora de buscar trabajos. También buscamos cierta implicación por parte de las empresas, para que ellas hagan más fácil la relación con los usuarios, ya que aporta un elemento social a nuestra aplicación.

Por otro lado, debemos tener en cuenta que el proceso de búsqueda de trabajo es algo no constante y fluctuable, es decir, un usuario puede estar activo durante mucho tiempo e inactivo durante otro tanto, por lo que debemos aportar un modelo que busque cierta consistencia para que los usuarios puedan volver a usarla sin ningún tipo de problema.

En definitiva, buscamos crear una aplicación con unas capacidades que puedan abarcar la búsqueda de minitrabajos de manera familiar mediante la aportación de menús con una amplia variedad de filtros y opciones, además de poder comunicarse con el empleador de manera directa y sin muchas complicaciones.

## Capacidades generales

El principal atractivo de nuestra aplicación es el aspecto temporal de los trabajos, la gente puede tener un trabajo extra durante media jornada o incluso menos y seguir con su trabajo habitual, o también tener múltiples mini-trabajos y adquirir más experiencia y aumentar el volumen de su CV.

También se ha pensado en el factor social de la aplicación, influenciados por webs como LinkedIn, ya que si podemos poner ‘cara’ a nuestro futuro trabajo se nos hará más familiar contactar con los empleadores y buscar trabajo.

El objetivo de esta aplicación es por tanto servir de salvoconducto entre estos tipos de trámites de manera rápida y efectiva. Si alguien busca un minitrabajo que le guste, se pone en contacto con la empresa y esta puede acceder a toda su información simplemente accediendo a su perfil, de esta manera, en pocos días el usuario podrá saber si ha sido aceptado o no, y, en el caso de serlo, empezar a trabajar en cuanto pueda.

En definitiva, el trabajo temporal puede ser algo que pueda cambiar el entorno laboral, sobre todo con los tiempos de pandemia que corren y la recesión económica. Por tanto, la mejor manera de que sea accesible para un mayor número de personas es transformar dichos procesos y volcarlos en la aplicación que estamos creando, este tipo de servicio es algo que nuestros competidores no han aportado al público.

## Restricciones generales

Las restricciones generales que debe presentar nuestra aplicación son:

**Privacidad:** se protegerá los datos de los usuarios cumpliendo la LOPD 03/2018[9]. Se almacenarán las conversaciones entre las empresas y los usuarios con el fin de velar por el cumplimiento de los acuerdos en caso de incidencia.

**Legales:** no se admitirá trabajos que incumplan el Estatuto de los trabajadores[11].

**Tecnológicas:** con respecto a las restricciones tecnológicas, el sistema no permitirá el acceso a la web, ni a la aplicación si no se dispone de acceso a internet.

**Comerciales:** el sistema no debe permitir acceder al registro a los usuarios menores de edad y a los extranjeros sin permiso de trabajo.

**Mercados:** nuestra aplicación se limitará al mercado español.

**Idiomas:** inglés y español y sus dialectos: catalán, gallego, vasco.

## Características de los usuarios: roles y capacidades

|  |  |  |
| --- | --- | --- |
| **Roles Stakeholders** | **Función / Capacidad** | **Estimación relativa de usuarios** |
| Técnicos IT | Mantenimiento, Seguridad, Verificación y Desarrollo | 1.98% |
| Empleadores | Publicación de ofertas de empleo. | 45% |
| Usuarios (expertos) | Buscar trabajo. Gran capacidad para buscar trabajos de su interés. | 18% |
| Usuarios (ocasionales) | Buscar trabajo. Nuevos en web/app | 34% |
| Inversores | Invierten dinero en la compañía. | 0.01% |
| Dueños | Fundadores de la compañía. Dirigen, planifican y controlan el desarrollo de los productos y gestionan la empresa. | 0.01% |
| Gestores | Solucionan incidencias y se aseguran de que las ofertas sigan todas las políticas de la empresa, también de suspender cuentas si fuera necesario y de mantener un buen entorno en la app. | 1% |

Descripción de la interfaz de usuario



Nuestro producto se ha pensado como un producto software que será desarrollado como una web y como una aplicación móvil. La interfaz ha sido pensada de tal forma que independientemente de la habilidad del usuario, pueda ser aprendida de forma fácil, y que resulte intuitiva a la hora de utilizarse.

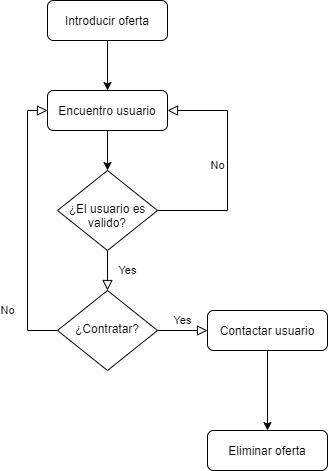
Por estas razones, se ha pensado poner un buscador que incorpora filtros para poder buscar ofertas de trabajo de forma sencilla.

En el centro de la página aparecerán las ofertas de trabajo que puedan interesar al usuario; y en la parte de la izquierda se incorporará un menú de inicio, junto con una opción para ver las ofertas que puedan haber sido marcadas como favoritas, y por último, el apartado “Mensajes”, desde donde el usuario podrá acceder a los chats de los que dispondrá para ponerse en contacto con las diferentes empresas.

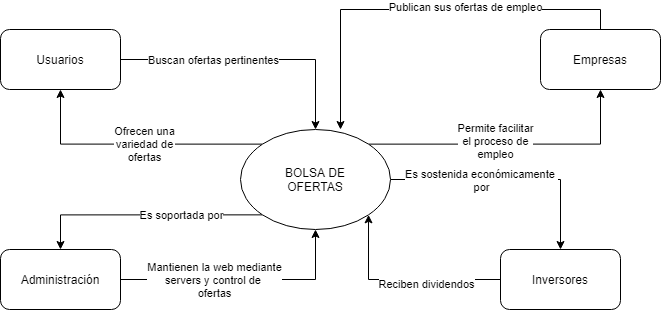
Esta interfaz ha sido adaptada en la versión móvil mediante una app, habiendo en esta una reorganización de los elementos, pero teniendo las mismas funcionalidades que en la versión web,

## Entorno operacional

* Despliegue: Nuestra aplicación y web deben estar disponibles en las plataformas más habituales:
  + En la store de Android y iPhone.
  + Tabletas.
  + Móviles.
  + Windows.
  + App Store de Mac.
  + Chrome.
  + Mozilla.
* Servidores: Nuestra aplicación móvil y web almacenarán sus datos en servidores no centralizados distribuidos por España para dar un mejor servicio. La información se encriptará para proteger los datos de las empresas y usuarios.
* Android Studio (Android): Entorno de desarrollo integrado oficial de la plataforma Android.
* Xcode (IOS): Herramienta para desarrollar nuestras aplicaciones en los sistemas operativos de Apple, el acceso es abierto para cualquier usuario y se puede programar en los principales lenguajes de programación.
* Empresas: Será esencial llegar a las organizaciones y empresas para que puedan acceder a nuestro servicio para ofertar sus trabajos más específicos. Se les facilitará la incorporación para que no suponga una pérdida de tiempo entrar en nuestro proyecto.



* Usuarios: Para llegar a los clientes se ofrecerá en entornos de personas cualificadas y dispuestas a entrar en un “minitrabajo”. Las campañas de marketing están orientadas a este perfil de usuario.



# Requisitos

## Justificación de la clasificación de requisitos

Para el desarrollo de la aplicación se ha estimado conveniente la clasificación de los requisitos funcionales por funciones: registro, perfil, publicaciones, búsqueda y comunicación.

El registro es el primer paso que debe hacer cualquier usuario para poder utilizar la aplicación exigiendo la información obligatoria.

El perfil es un elemento esencial, ya que será donde cada usuario muestre información acerca de él, su formación, experiencia, etc. Es decir, información indispensable para el empleador.

Publicaciones agrupa los requisitos necesarios para publicar ofertas, requiriendo información para para facilitar la búsqueda de estas.

En búsqueda se encuentran las funcionalidades necesarias para proporcionar al usuario una experiencia eficiente y óptima a la hora de buscar ofertas de su interés.

En métodos de pago se encontrará todo lo relativo a cómo los empleadores podrán pagar la tarifa que se les pide por poner publicaciones extra.

En comunicación detallamos cómo debe ser el canal mediante el cual usuario y empleador interactúan.

La clasificación de los requisitos no funcionales, o de calidad, se han descrito mediante las posibilidades de implementación de la aplicación, y es la siguiente: Consumo de recursos, Rendimiento, Fiabilidad y disponibilidad, Manejo de errores, Requisitos de interfaz, Restricciones y Seguridad.

En lo referente a consumo de recursos, se hablará sobre qué podrá utilizarse nuestra aplicación y web, que necesitan los usuarios.

En cuanto al rendimiento, indicamos los requisitos que abordan la velocidad de acceso a la web, y cómo es soportada mediante hardware.

Fiabilidad y disponibilidad, se hablará sobre cómo la aplicación será accesible en cualquier momento.

Manejo de errores, explicamos cómo resolvemos algunos errores que puede experimentar el sistema.

Restricciones, describe los límites de la aplicación en sus diseño e implementación.

Refiriéndose a la Seguridad, se agrupan todos los requisitos que tratan sobre encriptación de los datos de usuario, y su manejo en la aplicación.

En requisitos de interfaz accesibilidad se juntan los requisitos relacionados con los elementos de diseño de la web y cómo están predispuestos en diferentes dispositivos.

## Justificación de la plantilla de requisitos

Se ha considerado representar cada uno de los distintos requisitos mediante el uso de tablas, y en cada una de estas hemos incluido varios elementos que hemos considerado necesarios para poder explicar cada uno de estos de una manera precisa.

Para esto, hemos incluido un ID para poder identificar cada requisito, una clase que sirva para poder separar los distintos requisitos en áreas temáticas, un título y una descripción breve para cada uno, necesarios para el entendimiento de cada uno y, finalmente, las pruebas que se usarán para comprobar que estos requisitos se han implementado y funcionan correctamente.

También se ha considerado añadir un apartado para definir la prioridad de cada requisito, pero debido a que los requisitos definidos son principales y tienen la misma prioridad no hemos añadido esto.

## Requisitos funcionales

Añadidos en el Anexo: Requisitos.

## Requisitos no funcionales

Añadidos en el Anexo: Requisitos.

## Vocabulario del dominio

### Vocabulario del modelo conceptual

Chat: Método de comunicación privado bidireccional entre empleador y el usuario solicitante.

CIF: Código de identificación fiscal.

Curriculum: Documento donde se representan las capacidades, estudios, experiencia laboral, habilidades y datos de contacto del estudiantes.

Empleador: Creador de la publicación, y por tanto administrador del trabajo y la oferta.

Minitrabajo, mini-trabajo: Contrato de baja remuneración y máximo 15 horas de trabajo a la semana.

Ofertas: Publicaciones en la app de puestos de minitrabajos.

Perfil: Página personal del usuario que recoge sus datos para poder consultarlos.

Usuario: Cliente de la app, busca ofertas creadas por los empleadores.

Valoración: Método para evaluar las ofertas en la app mediante la opinión de usuarios.

### Vocabulario técnico

AES: Estándar de cifrado simétrico de datos.

APP: Aplicación móvil.

Chromium: Navegador web open-source sobre la que se construyen distintos navegadores.

Congestión: Es el fenómeno producido cuando a la [red](https://es.wikipedia.org/wiki/Red_de_computadoras), o parte de ella, se le ofrece más tráfico del que puede cursar.

CSS: Cascade Sheet Style.

HTML: Hypertext Markup Language:

JavaScript: Lenguaje programación ligero implementando por muchos navegadores.

Newsletter: Un boletín informativo es una publicación distribuida de forma regular, centrada en un tema principal.

PDF: Portable Document Format.

Resolución de pantalla: Cantidad de píxeles por pantalla.

Responsive: La app debe mostrar un buen formato independientemente del dispositivo mediante el que se acceda.

RSA: Estándar de cifrado asimétrico de datos.

TB: Terabyte. Unidad de almacenamiento de datos.

Teclado adaptativo: Teclado usado por discapacitados visuales.

W3C: World Wide Web Consortium.

LOPD: Ley Orgánica de Protección de Datos.

# Arquitectura

## Modelo conceptual

Esta sección debe contener un modelo cuyo propósito es especificar la *arquitectura de información* del sistema que se desea construir, mediante un conjunto de diagramas de clases adecuadamente explicados; se pueden completar con diagramas de objetos si se juzga pertinente.

El modelo de información, o modelo conceptual, debe estar adecuadamente *justificado* a partir de los requisitos. No tiene sentido que en él aparezcan clases, atributos, operaciones y otros elementos que no hayan aparecido anteriormente en los requisitos. Igualmente, no tiene sentido que en los requisitos se mencionen conceptos importantes que no aparezcan reflejados de ninguna manera en el modelo conceptual. La referencia al *vocabulario del dominio* (ver sección 3.4) es nuevamente un puente importante que vincula los requisitos con el modelo conceptual.

Es importante notar que la mera yuxtaposición de imágenes (diagramas) resulta muy poco explicativa e insuficiente; por otra parte, si el modelo se intentase explicar sólo mediante palabras, la explicación resultaría compleja y difícil de seguir. En otras palabras, hay que encontrar la combinación adecuada de diagramas y texto que se explican y complementan mutuamente. Analogía: imagina que quieres contar una excursión con fotografías; no te conformarías con poner una tras otra las fotografías, sino que cada una de ellas requiere una explicación textual, que la pone en relación con otras fotografías, llama la atención sobre sus aspectos más relevantes, etc.

Los diagramas de clases deben ayudar a entender el sistema. En consecuencia, a estos diagramas no se les debe añadir tanta información que sean difíciles de comprender, ni tan poca información que no añadan nada a lo que ya se dice en los requisitos. Estos diagramas deben tener el grado de detalle suficiente para conservar su carácter comunicativo, en tanto que constituyen una vista gráfica de los requisitos. Por tanto, deberán contener todas las clases conceptuales (es decir, las que aparecen mencionadas en los requisitos), pero no las que sean exigencia exclusivamente del diseño posterior. Así mismo, estas clases aparecerán con los atributos que sean relevantes desde el punto de vista conceptual. Para aumentar la claridad y potencia expresiva, puede ser conveniente mostrar un diagrama global en el que sólo aparezcan las clases, y varios diagramas parciales en los que aparezcan clases con atributos.

Con el fin de resaltar que este modelo representa la arquitectura de información del sistema, y no el diseño de la implementación, en esta práctica *queda prohibido incluir operaciones en las clases*. En otros contextos es perfectamente legítimo y útil representar las operaciones de las clases, pero cuando el objetivo es representar la arquitectura de información, las operaciones resultan contraproducentes.

Debe mencionarse la *herramienta de modelado* utilizada para crear los diagramas (ver 1.3 Referencias).

MODELADO DE DIAGRAMA DE CLASES.

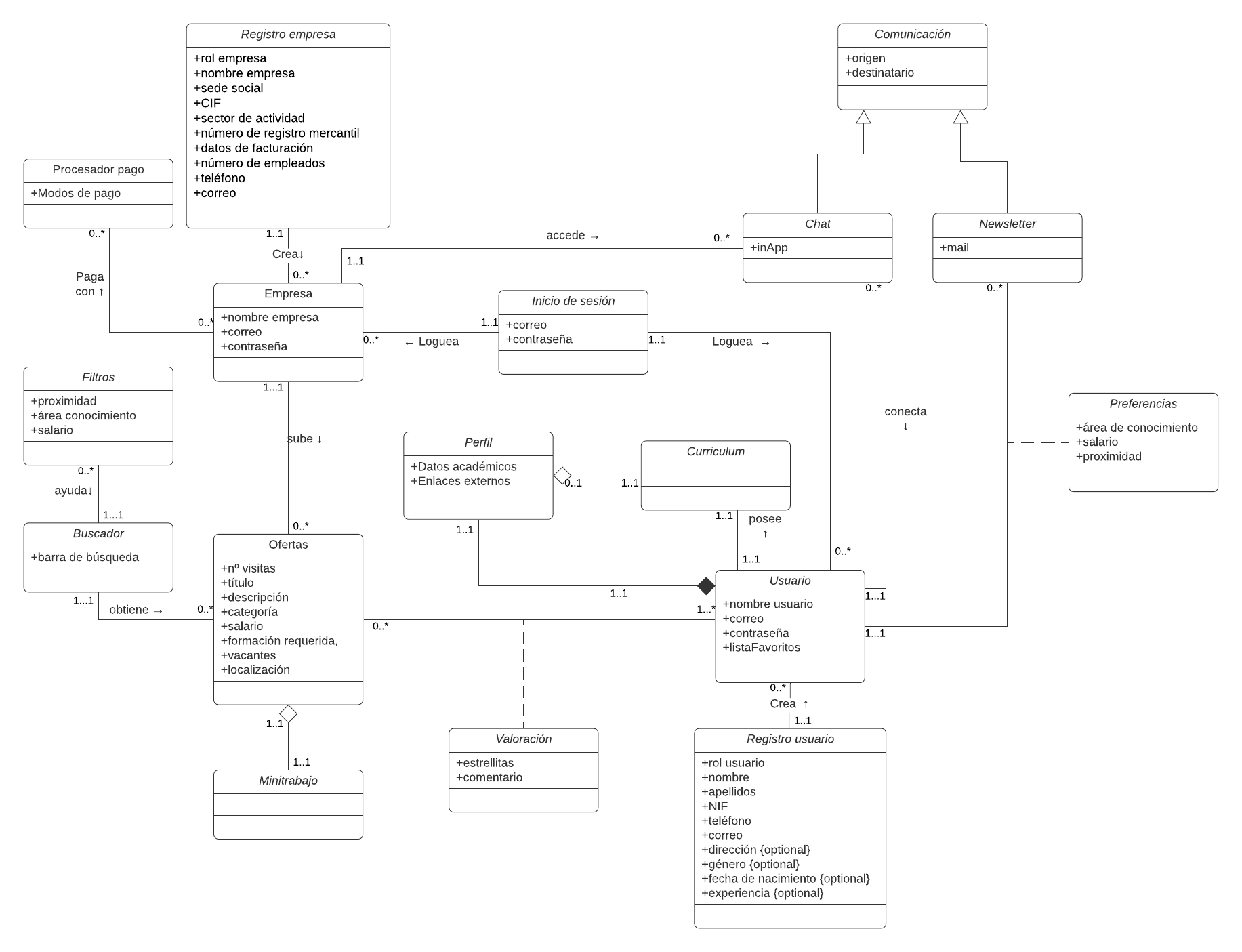
Para realizar el modelado de este diagrama es necesario conocer en extensión las funcionalidades de la aplicación para poder conocer las clases que la componen. Como este ha sido nuestro primer paso, hemos conseguido obtener las clases que se explican a continuación, con sus respectivas relaciones y atributos.

El apartado de gestión de usuarios se ha cubierto con Usuario, Registro Usuario, Empresa, Registro Empresa e Inicio Sesión. Las dos clases de registros representan los formularios de cada tipo de cuenta, que van asociados a sus respectivas clases, Usuario y Empresa, que se usan como las cuentas existentes en la aplicación, y sus atributos son correo electrónico, nombre de usuario y contraseña, aunque a la clase Usuario se ha añadido el atributo lista de favoritos para satisfacer la funcionalidad de ‘añadir ofertas a favoritos’. La clase inicio de sesión está conectada a los dos tipos de cuenta, y maneja el formulario de inicio de sesión de las cuentas, como su propio nombre indica.

En cuanto a la información que presta el usuario se maneja con las clases Currículum y Perfil. Currículum está ligado a Usuario mediante una asociación 1 a 1 (un usuario posee un único currículum y este sólo tiene un dueño). A su vez, Currículum se asocia a Perfil mediante una agregación, si se elimina el perfil, su currículum ligado no se destruye, ya que el currículum es propio del usuario y no del perfil. Para completar la funcionalidad, Perfil tiene como atributos los datos académicos y enlaces externos del usuario, y se asocia a Usuario por composición, ya que al borrar la cuenta, obviamente su perfil también desaparece, respetando los derechos de uso de la información personal.

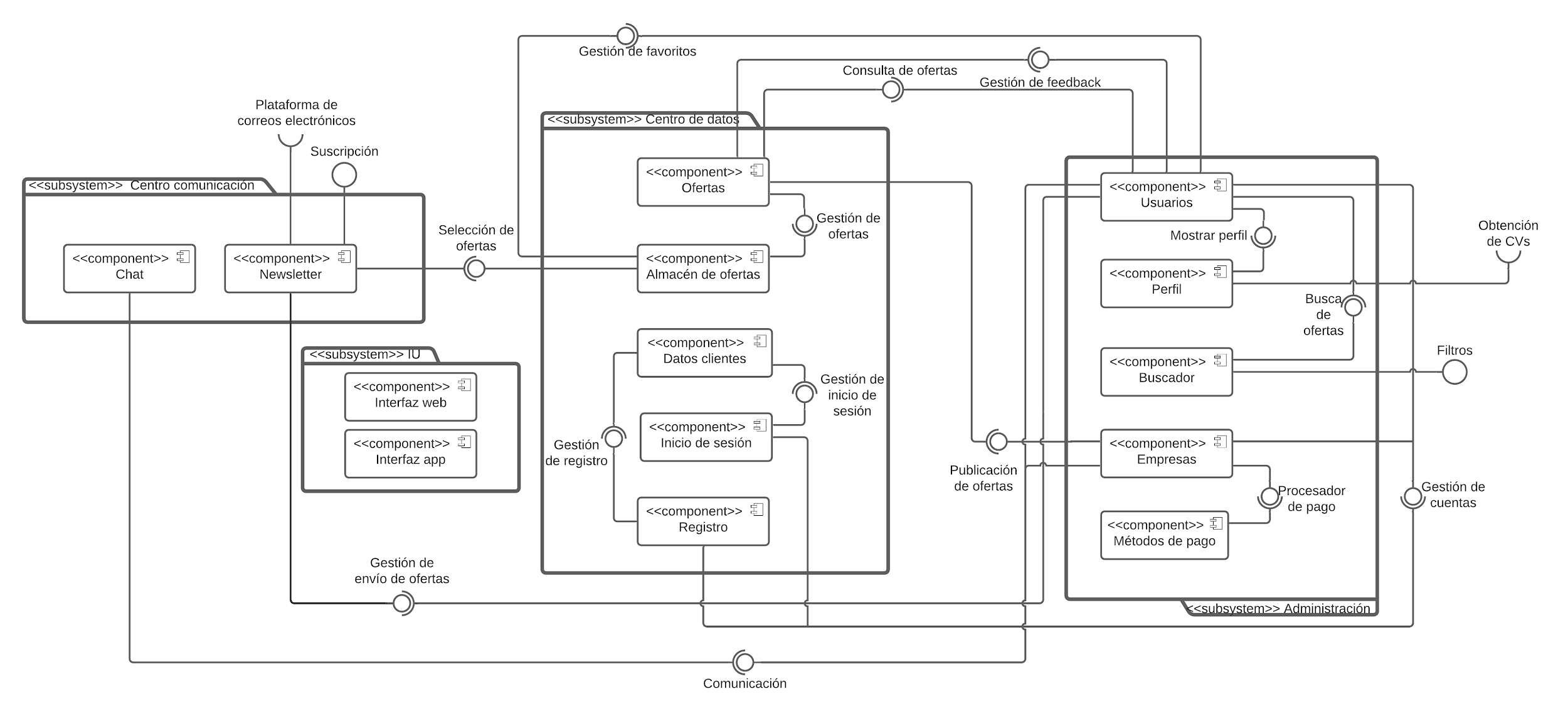
Para cubrir la búsqueda y publicación de ofertas asociamos a la clase Empresa con la clase Ofertas, con una relación de 1 a n. La clase ofertas tiene como atributos todos los elementos mencionados en los requisitos, y está ligada a la subclase Minitrabajo mediante una asociación 1 a 1. Por otro lado, el filtrado de ofertas se explica con las clases Filtros y Buscador. Filtros posee como atributos todos los filtros explicados anteriormente y se asocia a Buscador y este, a su vez va ligado a Ofertas. Para la valoración de ofertas se crea una clase asociación entre Ofertas y Usuario llamada Valoración, para satisfacer la relación n a n entre las dos clases anteriores. Un pequeño apunte que se debe hacer en esta parte del diagrama es la clase Procesador Pago, que explica las formas de pago que se pueden aplicar a las ofertas (de nuevo, ya especificado en el anexo de requisitos).

Por último hemos abarcado el factor ‘social’ de la aplicación con Chat (asociada a Empresa y Usuario) y con Newsletter (asociada a Usuario mediante la clase asociación Preferencias, para tener en cuenta la ‘personalización’ de las newsletter para cada usuario). Ambas clases son hijas de la clase Comunicación, que contiene toda la interacción que puede ocurrir entre usuarios.



*Acceso:* [*IS Final Modelo Conceptual.png*](https://drive.google.com/file/d/1nTQc6wCoR6_Nso7ipG2H5oUSnwsU2Bfh/view?usp=sharing)

## Modelo de implementación



*Acceso:* [*IS Final Componentes.png*](https://drive.google.com/file/d/1uMqrSp6AwGfwOuideXfpcTJLJJysFoYn/view?usp=sharing)

En esta sección se presenta y justifica la *arquitectura de desarrollo* elegida (descomposición del sistema en subsistemas y componentes), y se especifican las dependencias entre los distintos componentes que hayan resultado de la descomposición. La descomposición se hace habitualmente a varios niveles (componentes que a su vez se descomponen en subcomponentes, etc.), pero en esta práctica sólo se trata la descomposición de primer nivel, centrada en la vista externa (interfaces) de los componentes, y las relaciones entre ellos.

Se deben utilizar los diagramas y vistas que sean necesarios y las oportunas explicaciones textuales. Vale lo dicho en la sección 4.1 sobre la adecuada combinación de imágenes y palabras

La descomposición arquitectónica elegida debe *justificarse* teniendo en cuenta los criterios vistos en clase: simplicidad, extensibilidad, modificabilidad, eficiencia, y tal vez otros que se consideren relevantes. También deben considerarse los requisitos no funcionales a la hora de justificar la arquitectura. La clasificación de requisitos por áreas temáticas (ver 3.1) puede también proporcionar pistas importantes para esta descomposición. Si existen varias opciones arquitectónicas, entonces hay que compararlas y escoger de modo razonado la que mejor se adapte a los requisitos del sistema.

No hay obligación de seguir una arquitectura predefinida, ni hay una descomposición que *a priori* sea mejor que las demás (por ejemplo, MVC, modelo-vista-controlador), sino que la decisión depende del contexto. Lo que hay que hacer es justificar la elección de la arquitectura, razonando por qué es adecuada para resolver el problema planteado. La arquitectura empleada puede ser un estilo arquitectónico estándar, pero también puede ser una variante o un híbrido de estilos, o puede estar sólo vagamente inspirada en uno de ellos. Por lo mismo, tampoco basta que la arquitectura elegida se corresponda con un estilo estándar: lo que hay que hacer es justificar que es adecuada para resolver el problema.

Como en el caso anterior, debe mencionarse la *herramienta de modelado* utilizada para crear los diagramas (ver 1.3 Referencias).

DISEÑO DIAGRAMA COMPONENTES.

Para el diseño del diagramas de componentes se ha decidido que existiesen tres subsistemas distintos que son: el Centro de comunicación, el Centro de datos y la Administración. Ahora se especificarán los componentes que se encuentran en cada uno de los subsistemas, detallando las relaciones que tienen cada uno de estos componentes con las diferentes interfaces.

El otro interfaz se tratará de “Selección de ofertas” que irá conectado a Newsletter, para que se puedan seleccionar las ofertas en las que cada usuario está interesado y estas se puedan mandar en forma de newsletter a cada usuario.

Otro componente del segundo subsistema es Datos clientes que se encuentra conectada a una interfaz requerida “Obtención de CVs”, está conexión se encuentra así definida, ya que en los datos de

## Especificación del diseño de componentes

En esta sección se especifican las propiedades de cada uno de los componentes identificados en la sección anterior. Cada componente deberá tener un nombre identificador único. El identificador deberá reflejar el propósito y la funcionalidad del componente, siendo a la vez breve y significativo. Si es necesario utilizar abreviaturas, deben utilizarse coherentemente y sin ambigüedad, y deben quedar documentadas. Los componentes deberán tener identificadores mutuamente consistentes: por ejemplo, si un componente se llama Publicador\_Registros, entonces es posible que exista uno que se llame Editor\_Registros, pero no Edición\_Registros, ni Editor-Registros, ni EditorRegistros.

El elemento esencial de la especificación de cada componente son sus interfaces, que obviamente deben coincidir perfectamente con las interfaces identificadas en el modelo de implementación (sección 4.2).

### Componente Chat (repetir para cada componente)

Por cada componente se repite la misma estructura con los siguientes apartados:

**Tipo**

Debe indicarse esencialmente si es ejecutable o no ejecutable. Componentes no ejecutables serían archivos de configuración, plantillas de páginas web, etc. El contenido de los demás apartados de la descripción del componente depende del tipo de componente: si es ejecutable, habrá que hacer énfasis en el Procesamiento (por ejemplo, métodos ofrecidos por las clases de implementación); si no es ejecutable, en los Datos (por ejemplo, información contenida en una página web). Los apartados de la descripción no son fijos, sino que deben adaptarse al tipo de componente.

**Propósito [Función, Procesamiento, Datos, Recursos]**

El propósito del componente se define esencialmente trazándolo contra los requisitos que el componente implementa. Esta trazabilidad hacia atrás justifica la existencia del componente y explica su propósito. Este apartado contiene una breve descripción textual del propósito del componente, basada en esos requisitos. El detalle de la trazabilidad se especifica en la sección 5.3.

**Dependencias (interfaces requeridas)**

Se definen las dependencias respecto a interfaces proporcionadas por otros componentes del sistema, o por sistemas externos. En el primer caso, dado que toda *interfaz requerida* tiene su contrapartida en una *interfaz proporcionada*, no es necesario repetir la especificación de la interfaz en ambos lugares; basta con especificarla como interfaz proporcionada y referenciarla adecuadamente en el otro lugar donde es requerida. En el segundo caso, sí es necesario especificar completamente la interfaz requerida, puesto que la interfaz proporcionada queda fuera del sistema.

**Interfaces (Interfaces proporcionadas)**

De modo general una interfaz es un conjunto de operaciones que ofrecen un servicio coherente. En diseño orientado a objetos se va más allá: una interfaz *define un tipo*, que proporciona un conjunto coherente de operaciones sobre las instancias compatibles con ese tipo.

La definición de interfaces permite aislar componentes y lograr un diseño modular y mantenible mediante interacciones bien definidas. Para definir correctamente el tipo especificado por la interfaz, se enumeran las operaciones pertenecientes a la interfaz, así como los contratos que deben satisfacer cada una de las operaciones. Para especificar los contratos debe utilizarse la técnica de *diseño por contratos* vista en el curso.

### Componente Chat

**Tipo**

Es de tipo ejecutable.

**Propósito [Función, Procesamiento, Datos, Recursos]**

El propósito del componente como ya ha sido mencionado, será el de servir como medio de comunicación entre el usuario interesado en una oferta y la empresa que haya publicado dicha oferta.

**Dependencias (interfaces requeridas)**

La interfaz requerida de este componente llamada “Comunicación” se especificará en el apartado de interfaz proporcionada del componente Usuarios.

**Interfaces (Interfaces proporcionadas)**

Este componente no tiene interfaces proporcionadas.

### Componente Newsletter

**Tipo**

Es de tipo no ejecutable.

**Propósito [Función, Procesamiento, Datos, Recursos]**

El propósito de este componente será englobar todas las ofertas en las que un usuario pudiese estar interesado, para después poder mandar una newsletter personalizada para cada usuario.

**Dependencias (interfaces requeridas)**

Este componente tiene dos interfaces requeridas:

-”Plataforma de correos electrónicos”: que será necesaria para poder mandar las newsletters una vez finalizadas.

-”Selección de ofertas”: se definirá en la interfaz proporcionada del componente Almacén de ofertas.

**Interfaces (Interfaces proporcionadas)**

-”Suscripción”: se ofrecerá la posibilidad de suscribirse al enviarse la newsletter.

### Componente Ofertas

**Tipo**

Es de tipo no ejecutable.

**Propósito [Función, Procesamiento, Datos, Recursos]**

El propósito del componente será contener los datos de las ofertas, los cuales podrán ser consultados y gestionados.

**Dependencias (interfaces requeridas)**

La interfaz requerida de este componente llamada “Gestión de feedback” se especificará en el apartado de interfaz proporcionada del componente Usuarios.

La otra interfaz es “Publicación de ofertas”, será explicado en el apartado de interfaz proporcionada del componente Empresas.

**Interfaces (Interfaces proporcionadas)**

-”Consulta de ofertas”: se encuentra conectada al interfaz requerida procedente del componente Usuarios, estas interfaces servirán para que el componente Ofertas proporcione los datos necesarios para que el usuario consulte los datos de las ofertas.

-”Gestión de ofertas”: se encuentra conectada al interfaz requerida procedente del componente Almacén de ofertas, estas interfaces servirán para que el componente Ofertas proporcione los datos necesarios para que las ofertas sean almacenadas.

### Componente Almacén de ofertas

**Tipo**

Es de tipo no ejecutable.

**Propósito [Función, Procesamiento, Datos, Recursos]**

El propósito del componente es el de almacenar las ofertas publicadas en el sistema.

**Dependencias (interfaces requeridas)**

La interfaz requerida de este componente llamada ”Gestión de ofertas”, ha sido explicada junto con la interfaz ofrecida del componente Ofertas.

**Interfaces (Interfaces proporcionadas)**

-”Gestión de favoritos”: se encuentra conectada al interfaz requerida procedente del componente Usuarios, servirá para poder gestionar las ofertas favoritas de cada usuario del sistema.

-”Selección de ofertas”: se encuentra conectada al interfaz requerida procedente del componente Newsletter, servirá para que se puedan seleccionar las ofertas en las que cada usuario está interesado y estas se puedan mandar en forma de newsletter a cada usuario.

### Componente Datos clientes

**Tipo**

Es de tipo no ejecutable.

**Propósito [Función, Procesamiento, Datos, Recursos]**

El propósito del componente es almacenar los datos de los clientes, tanto de usuarios como de empresas.

**Dependencias (interfaces requeridas)**

La interfaz requerida de este componente llamada ”Gestión de inicio de sesión”, será explicada junto con la interfaz ofrecida del componente Ofertas.

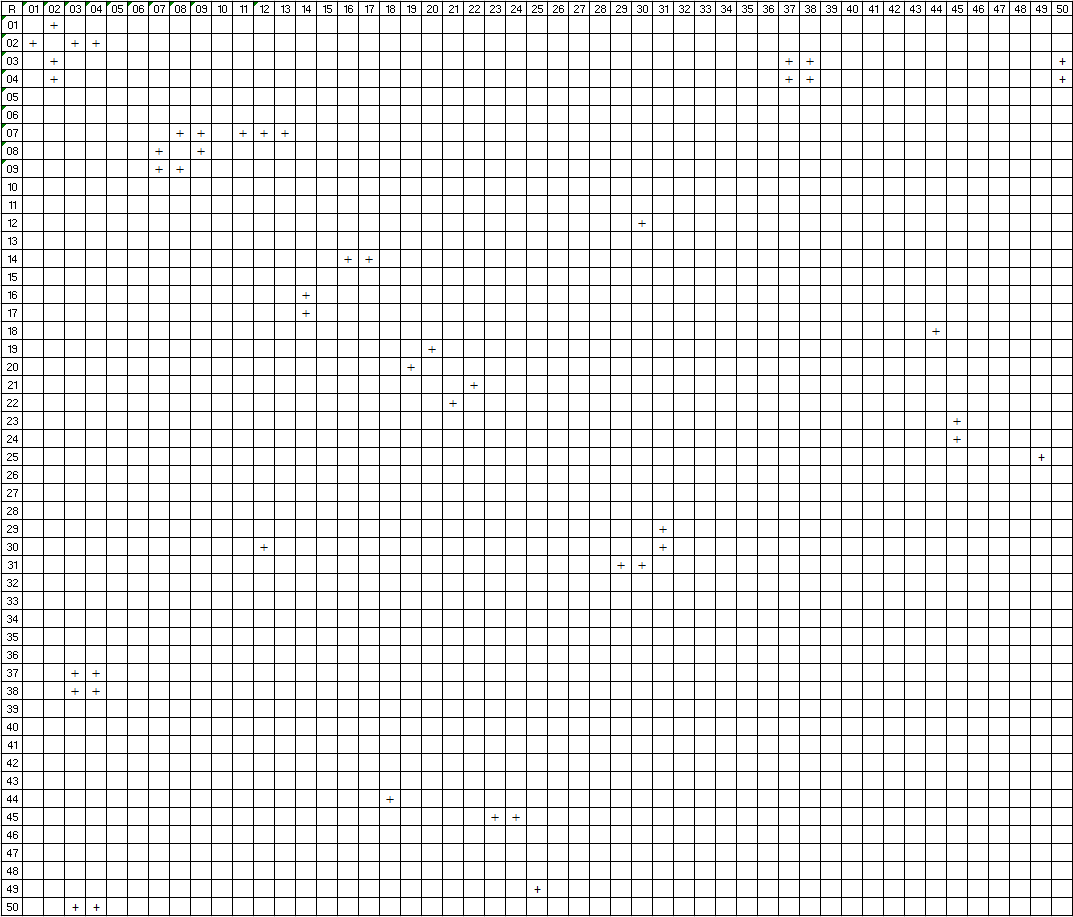
**Interfaces (Interfaces proporcionadas)**

-”Gestión de favoritos”: se encuentra conectada al interfaz requerida procedente del componente Usuarios, servirá para poder gestionar las ofertas favoritas de cada usuario del sistema.

-”Selección de ofertas”: se encuentra conectada al interfaz requerida procedente del componente Newsletter, servirá para que se puedan seleccionar las ofertas en las que cada usuario está interesado y estas se puedan mandar en forma de newsletter a cada usuario.

# Listas y tablas

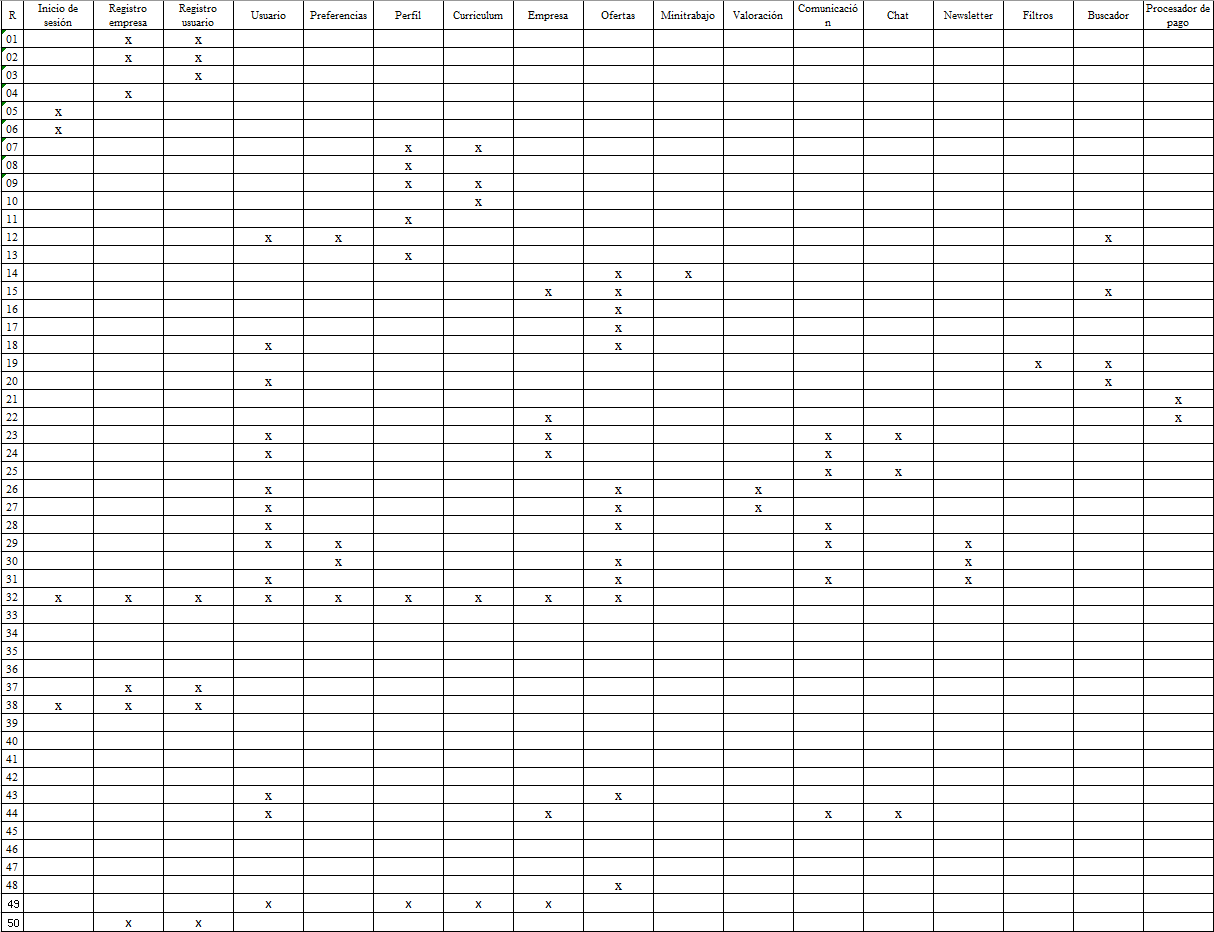
## Consistencia entre requisitos: conflictos, redundancias, acoplamientos



*Acceso:* [*consistencia.png*](https://drive.google.com/file/d/1QdKd0xJ2KDC2aU71TdIsJ3eegMf_MT3e/view?usp=sharing)

Leyenda: x: Conflicto. +: Acoplamiento. o: Redundancia.

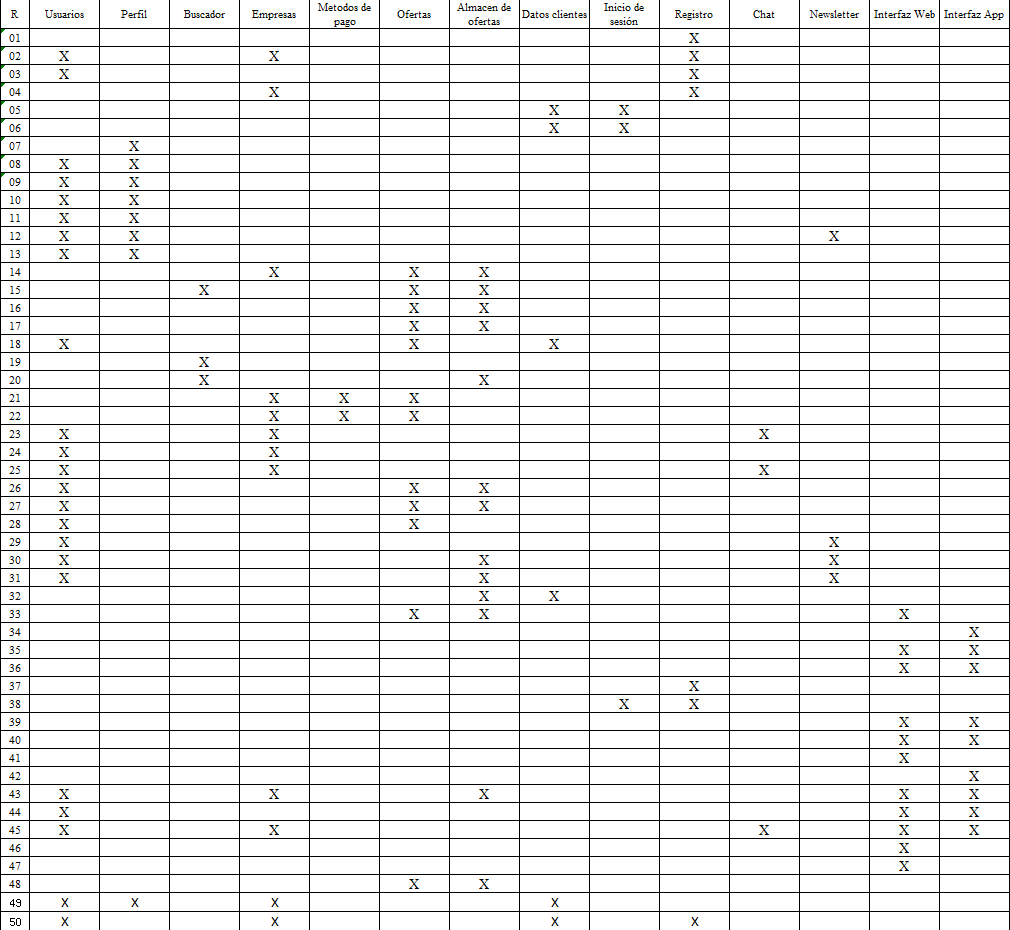
## Trazabilidad requisitos – modelo conceptual (clases)



## *Acceso:* [*trazabilidad-conceptual.png*](https://drive.google.com/file/d/1Esn6PU8pwYiGn_gH_jx3oebKexHZpjMD/view?usp=sharing)

## Trazabilidad requisitos – modelo de implementación (componentes)

En estas tres secciones pueden utilizarse tablas de doble entrada o tablas de 2-3 columnas. La tabla de doble entrada tiene el peligro de ser excesivamente dispersa, por lo que a menudo las tablas simplificadas de pocas columnas son una solución más conveniente.



*Acceso:* [*trazabilidad-componentes.png*](https://drive.google.com/file/d/1Qd2mAGf0iGtWbKAUP4AArB_fYbQf4kBR/view?usp=sharing)