UNIVERSIDAD SAN PABLO - CEU

ESCUELA POLITÉCNICA SUPERIOR

GRADO EN INGENIERÍA DE SISTEMAS DE INFORMACIÓN



TRABAJO FIN DE GRADO

**JOBS : diseño e implementación de una aplicación RESTful para búsqueda de empleo para estudiantes/graduados**

Autor: Carlos Soler Martín

Tutores: Sergio Saugar Garcia y Alex Rayón Jerez

Junio 2022

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Datos del alumno   |  | | --- | | Nombre: Carlos Soler Martín |   Datos del Trabajo   |  | | --- | | TÍTULO DEL PROYECTO: JOBS - diseño e implementación de una aplicación RESTful para búsqueda de empleo para estudiantes/graduados |   Tribunal calificador   |  |  | | --- | --- | | Presidente: | Fdo.: |  |  |  | | --- | --- | | Secretario: | Fdo.: |  |  |  | | --- | --- | | Vocal: | Fdo.: |  |  | | --- | | Reunido este tribunal el \_\_\_ /\_\_\_ /\_\_\_\_\_\_\_, acuerda otorgar al Trabajo Fin de Grado presentado por D./Dña. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ la calificación de \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |

Resumen

Texto con, a lo sumo, 200 palabras.

Palabras Clave

Abstract

Resumen anterior, en inglés.

Keywords

Índice de contenidos

[Capítulo 1 Introducción 1](#_Toc104540048)

[Capítulo 2 Gestión del proyecto 5](#_Toc104540049)

[2.1 Modelo de ciclo de vida 5](#_Toc104540050)

[2.2 Papeles desempeñados en el proyecto 6](#_Toc104540051)

[2.3 Planificación 6](#_Toc104540052)

[2.4 Presupuesto 7](#_Toc104540053)

[2.5 Ejecución 7](#_Toc104540054)

[Capítulo 3 Análisis 9](#_Toc104540055)

[3.1 Especificación de requisitos 9](#_Toc104540056)

[3.2 Análisis de los casos de uso y de las clases de análisis 11](#_Toc104540057)

[3.3 Análisis de seguridad 12](#_Toc104540058)

[3.4 Análisis desde la perspectiva del RGPD (si procede) 12](#_Toc104540059)

[Capítulo 4 Diseño e implementación 13](#_Toc104540060)

[4.1 Arquitectura del sistema 13](#_Toc104540061)

[4.2 Modelo de clases de diseño 13](#_Toc104540062)

[4.3 Diseño físico de datos 13](#_Toc104540063)

[4.4 Migración y carga inicial de datos (si procede) 13](#_Toc104540064)

[4.5 Diseño de la interfaz de usuario 14](#_Toc104540065)

[4.6 Entorno de construcción 14](#_Toc104540066)

[4.7 Plan de pruebas 14](#_Toc104540067)

[4.8 Diagrama de infraestructuras de nivel 3 14](#_Toc104540068)

[4.9 Diagrama de infraestructuras de nivel 2 (si procede) 14](#_Toc104540069)

[Capítulo 5 Construcción 16](#_Toc104540070)

[5.1 Referencia al repositorio de software 16](#_Toc104540071)

[5.2 Manuales 16](#_Toc104540072)

[Capítulo 6 Algoritmo de Recomendación 17](#_Toc104540073)

[6.1 Objetivo 17](#_Toc104540074)

[6.2 Introducción a la ciencia de los datos 17](#_Toc104540075)

[6.3 Machine Learning: métodos de aprendizaje 18](#_Toc104540076)

[6.4 Técnicas de análisis de datos prescriptivas: sistemas de recomendación 20](#_Toc104540077)

[6.5 Medidas de similitud o distancia 22](#_Toc104540078)

[6.6 Lenguaje utilizado: R 23](#_Toc104540079)

[6.7 Sistema de recomendación de JOBS 24](#_Toc104540080)

[6.7.1 Estructura del sistema de recomendación 24](#_Toc104540081)

[6.7.2 API “ofertas\_nuevas.R” 24](#_Toc104540082)

[6.7.3 API “alumno\_nuevo.R” 26](#_Toc104540083)

[6.7.4 Llamada de las APIs 27](#_Toc104540084)

[6.7.5 Intercambio de ficheros Web-APIs 27](#_Toc104540085)

[Capítulo 7 Conclusiones y líneas futuras 29](#_Toc104540086)

[Bibliografía 31](#_Toc104540087)

[Anexo I 33](#_Toc104540088)

[Otras posibilidades para realizar el análisis y el diseño 33](#_Toc104540089)

[Anexo II 34](#_Toc104540090)

[Fuente de inspiración del presente documento 34](#_Toc104540091)

[Glosario de términos 35](#_Toc104540092)

[Anexos 37](#_Toc104540093)

Índice de ilustraciones

Índice de tablas

# Introducción

Se explica en qué consiste el proyecto y se enumeran los capítulos. Debe incluir sus objetivos.

Desde el inicio de la pandemia, la situación de desempleo en España ha ido a peor y esto perjudica a todos los jóvenes estudiantes a la hora de encontrar su primer empleo o las prácticas que son necesarias para poder conseguir el título universitario. Este proyecto tiene como objetivo contribuir a solventar la dificultad que tienen los graduados universitarios españoles para encontrar un primer empleo. Algunos datos que explican esta problemática:

* Según un estudio realizado por Uniplaces, el 82% de adolescentes españoles se plantean ir a trabajar fuera de España, un porcentaje significativamente mayos comparado con el 63% de los franceses y el 66% de los alemanes.
* “Frustración, miedo e incertidumbre” son las mejores palabras para definir cómo se siente un adolescente que al acabar su grado o máster y no tener perspectiva de encontrar trabajo. Una de las principales razones por la que miles de jóvenes ‘hiperformados’ no optan a un puesto de trabajo es la falta de experiencia. ¿Cómo puedo empezar a trabajar si casi todas las empresas me piden experiencia? ¿Y las empresas que no me piden experiencia como las encuentro? Al final muchos jóvenes se conforman con un trabajo que les permita mantenerse y no consiguen trabajar en algo que realmente les atraiga.
* Cómo consecuencia de la pandemia los universitarios han tenido dificultades para encontrar prácticas curriculares. Miles de estudiantes o no han podido encontrar prácticas o se les han retrasado, por tanto, no han podido terminar el grado hasta que contarán con todos los créditos de las prácticas. El proceso de encontrar trabajo/practicas puede producir una sensación de desesperación/depresión para el estudiante.

El departamento de carreras profesionales de la Universidad CEU San Pablo es el encargado de ayudar al estudiante y graduado a encontrar practicas curriculares/primer trabajo. Cuentan con 5 aplicaciones para la búsqueda de empleo a nivel nacional e internacional. De esas 5 aplicaciones solamente se hace el uso de dos, que son:

* GetHighered: aplicación para buscar trabajo a nivel internacional
* JobTeaser: aplicación para buscar trabajo a nivel nacional.

La funcionalidad en ambas aplicaciones es la misma: el estudiante crea su perfil y adjunta su CV. Una vez realizado lo anterior, el estudiante debe de buscar de manera manual las empresas a las que le gustaría aplicar, una vez encuentra una empresa de trabajo que le interesa debe de enviar a través de la aplicación una pequeña carta de motivación o a veces la empresa te redirige a la sección de recruiting de su web. Por parte de las empresas también deben de registrarse en la aplicación y publicar su oferta de trabajo, o normalmente el equipo de carreras profesionales reciben ofertas de las empresas por correo y las postean en las plataformas de empleo. Cuando son las propias empresas las que publican la oferta de empleo es necesario que alguien de carreras profesionales valide la solicitud.

También cuentan con un portal de prácticas para el alumnado dentro del campus virtual, en el cual se puede: subir el CV del alumno, ver una lista de ofertas de trabajo y poder ponerse en contacto con la empresa, firmar el anexo de las practicas, realizar una memoria sobre las practicas realizadas, realizar un seguimiento del trámite de las prácticas y realizar una encuesta.

La situación descrita es común en el sistema universitario español y presenta una serie de ineficiencias que este proyecto trata de paliar:

* No existe una base de datos integrada de alumnos universitarios que buscan prácticas o un primer empleo.
* Tampoco existe un sistema de información eficiente de las prácticas o primeros empleos ofertados por las empresas. Las relaciones universidad-empresa son bilaterales y no son eficientes.
* Los departamentos de carreras profesionales de las universidades realizan de forma manual el proceso de matching entre alumnos (en base CV) y ofertas de prácticas/primeros empleos.
* Los departamentos de carreras profesionales de las universidades carecen de aplicaciones que les permitan realizar el matching anteriormente mencionado de forma automática, así cómo gestionar de forma eficiente todo el proceso administrativo asociado (contratos, anexos, memorias, etc.).

La plataforma JOBS ayudará a solventar alguna de las deficiencias anteriormente mencionadas ya que tiene como objetivo:

* Constituirse en una base de datos unificada de CVs de alumnos y ofertas de prácticas/primeros empleos.
* Homogeneizar los CVs de los alumnos a través de un modelo de estandarizado de capacidades (“skills”)
* Facilitar un matching automático entre alumnos (en base CV) y ofertas de prácticas/primeros empleos.
* Desarrollar un modelo (basado en un algoritmo de machine learning) que recomiende a los alumnos las ofertas de prácticas/empleo mas adecuadas a su CV y qué también les recomiende ofertas menos evidentes, pero que son adecuadas al perfil del alumno.

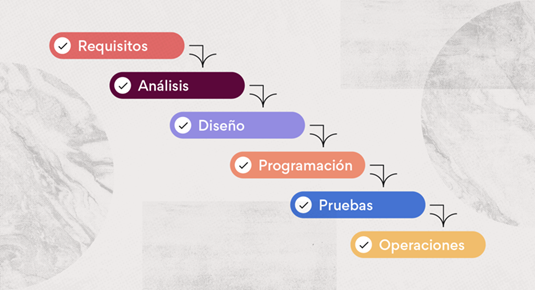
# Gestión del proyecto

## Modelo de ciclo de vida

Ha de indicarse si se lleva a cabo un modelo de ciclo de vida en cascada, incremental, etc.

El ciclo de vida de un software es una secuencia estructurada y bien definida de las etapas necesarias para desarrollar el software deseado. Para el desarrollo del proyecto se ha decidido utilizar un Modelo de Cascada ya que es una metodología simple y las actividades (requisitos) están divididas en fases secuenciales contando con bucles por si es necesario volver hacia atrás para modificar/mejorar una fase. Dicho modelo se divide en las siguientes fases:

* Requisitos: consiste en la planificación detallada del proyecto donde: hay que estimar el tiempo y los recursos necesarios, evaluar cuidadosamente los riesgos, realizar una lista detallada de tareas, gráficos de dependencia, establecer hitos, ir mejorando los planes a medida que vamos sabiendo más, gráficos de GANTT y PERT
* Análisis: consiste en la especificación del producto, describir el sistema desde la perspectiva del usuario, descripción detallada de los datos de entrada/salida, como serán manejados los errores, todo debe de ser preciso para no tener que volver a fases anteriores.
* Diseño: está dividida en el diseño de la arquitectura (decidir el lenguaje de programación, diseñar las interfaces, muy importante que todas las decisiones de diseño estén especificadas claramente) y diseño detallado (diseñar estructura de datos y los algoritmos).
* Programación: traducir el diseño en código.
* Testing: probar que funciona como se esperaba el código
* Mantenimiento (Operaciones): mantenimiento del programa y refactorización del código.



*Ilustración 1: Fases del modelo en cascada*

## Papeles desempeñados en el proyecto

Además del tutor del trabajo fin de grado (TFG), que será el *dueño de producto*, y del estudiante, en el que colapsarán los papeles de director del proyecto y analista-programador (en caso de que el proyecto esté orientado a desarrollo), se indicará si también participan *stakeholders* y usuarios expertos y quiénes van a ser.

Los dos tutores de este TFG, Sergio Saugar Garcia y Alex Rayón Jerez, son los promotores y propietarios de este proyecto y el alumno, Carlos Soler Martín es el director del proyecto, siendo también el analista-programador que ha diseñado y desarrollado la aplicación JOBS.

## Planificación

Se identificarán las tareas a realizar, su descripción, sus fechas estimadas de inicio y de fin, sus precedencias, los recursos que requieren, etc. Se podrán utilizar diagramas de Gantt, PERT / CPM, etc. Deberán incluirse también los diferentes hitos a lo largo del proyecto. Para que el estudiante tenga la experiencia real de planificar y controlar su cumplimiento, es **importante** que la planificación se realice con anterioridad a la finalización del proyecto.

Se pueden encontrar explicaciones tanto sobre los diagramas que se indican en esta sección como los que se mencionan en los siguientes apartados en la [documentación sobre técnicas de la metodología Métrica 3](https://administracionelectronica.gob.es/pae_Home/dam/jcr:da7d91fa-d6bd-467c-be32-a72e27c603b3/METRICA_V3_Tecnicas.pdf).

## Presupuesto

En caso de que el desarrollo del TFG en sí mismo implique costes, o tenga sentido un trabajo con proyección a futuro, se indicará qué personal (con sus perfiles, dedicación, etc.), licencias de software, infraestructuras, etc. serán necesarios, así como sus precios y costes correspondientes.

El desarrollo del proyecto de este TFG no ha supuesto incurrir en ningún coste, ya que el software utilizado no ha implicado pagar ninguna licencia y todas las horas de desarrollo han sido asumidas por el alumno. Estando cursando el alumno el doble grado de ISI y ADE, su intención es continuar con este proyecto en el TFG de ADE del curso que viene y realizar una propuesta de emprendimiento empresarial con la intención que a futuro pueda ser la base de una empresa emergente (“startup”).

## Ejecución

Se mostrará cómo ha transcurrido realmente el proyecto. Se podrán utilizar los mismos diagramas que en apartado anterior (diagramas de Gantt, PERT / CPM, etc.).

Xxx

# Análisis

## Especificación de requisitos

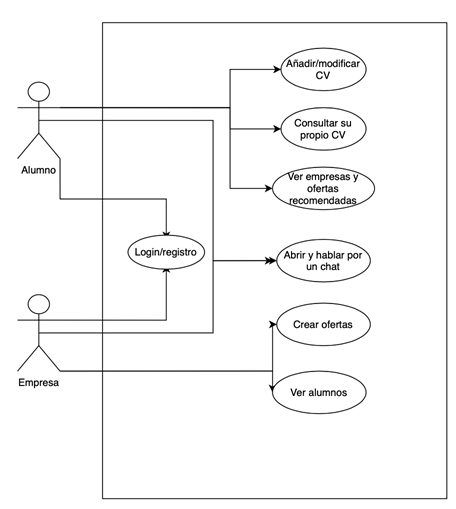
En un escenario habitual, se deberán especificar, al menos, los siguientes tipos de requisitos:

* *Funcionalidades del sistema,* que podrán detallarse, por ejemplo, mediante diagramas de casos de uso con su especificación. Alternativamente se pueden mostrar historias de usuario.
* *Interfaz de usuario* (si procede). Puede incluirse, por ejemplo, el nivel de cumplimiento de la norma de accesibilidad [WCAG](https://www.w3.org/WAI/standards-guidelines/wcag/).
* *Rendimiento*, incluyendo tiempos de respuesta.
* *Capacidad*, incluyendo el número máximo de peticiones que se pueden procesar por unidad de tiempo.
* *Seguridad*. Se indicará el nivel de seguridad que ha de satisfacer el sistema, por ejemplo, según el Esquema Nacional de Seguridad (RD 3/2010) o según alguna norma de la familia ISO 27000. Este nivel de seguridad se establecerá a partir de un análisis de riesgos (véase la sección de análisis de seguridad).
* *Interoperabilidad con otros sistemas* (si procede).
* *Protección de datos* (si procede), en concreto, el cumplimiento del Reglamento General de Protección de Datos ([UE] 2016/679) y de la Ley Orgánica 3/2018 de Protección de Datos Personales y Garantía de los Derechos Digitales (véase la sección de análisis desde la perspectiva del RGPD).
* *Requisitos sobre entorno tecnológico y de comunicaciones*, es decir, requisitos sobre gestores de bases de datos, sistema o los sistemas operativos, hardware, protocolos de red, etc.

Al comienzo del desarrollo del proyecto se definieron una serie de requisitos que para facilitar el desarrollo de la aplicación:

Requisitos funcionales

* Los alumnos deberán de ser capaces de registrarse, hacer login en la aplicación y cerrar sesión.
* Los alumnos deberán añadir su “CV” (skill, capacidades), que posteriormente podrá ser modificado por dicho alumno si lo desea
* Los alumnos podrán consultar su CV
* Los alumnos podrán ver las ofertas de empresas registradas en la aplicación
* Los alumnos deberán de ser capaces de ver las ofertas recomendadas en función del CV introducido
* Las empresas deberán de ser capaces de registrarse y hacer login en la aplicación
* Las empresas serán capaces de crear nuevas ofertas
* Las empresas tendrán acceso a los CV de los alumnos.
* **Chat**

****

*Ilustración 2: Diagrama de casos de uso*

Requisitos de interfaces externas

* Para la gestión del proyecto se utilizará un sistema de control de versiones

Requisitos de rendimiento

Requisitos tecnológicos

* Se utilizará la base de datos PostGreSQL para la gestión de todos los datos de la aplicación

## Análisis de los casos de uso y de las clases de análisis

A partir de los casos de uso, se obtendrá el modelo de clases de análisis, así como la realización de los casos de uso (diagramas de interacción entre objetos). En el diagrama de clases, para cada una de ellas, se identificarán sus atributos, responsabilidades (funcionalidades), asociaciones, agregaciones y generalizaciones. El comportamiento de las clases podrá mostrarse mediante diagramas de transición de estados.

Los diagramas elaborados no estarán condicionados por la tecnología utilizada, sino que estarán centrados en el problema en sí a resolver.

Xxx

## Análisis de seguridad

Se llevará a cabo un análisis de riesgos de acuerdo con las dimensiones de autenticidad, confidencialidad, integridad, disponibilidad y trazabilidad.

Xxx

## Análisis desde la perspectiva del RGPD (si procede)

En caso de que sea necesario, se llevará a cabo una gestión del riesgo y evaluación de impacto en tratamientos de datos personales.

Xxx

# Diseño e implementación

## Arquitectura del sistema

Se elaborará un diagrama donde se indiquen las interfaces del sistema, los diferentes servicios que proporciona, los componentes en que se apoyan tales servicios, capa de acceso a datos, etc.

Xxx

## Modelo de clases de diseño

El diagrama de clases de análisis se transformará en otro en el que se incluyan aspectos más tecnológicos (accesos a bibliotecas, API de persistencia, etc.).

Xxx

## Diseño físico de datos

Se llevará un proceso de normalización y de optimización para obtener las tablas de la base de datos.

Xxx

## Migración y carga inicial de datos (si procede)

En caso de que sea necesario, se elaborará un plan de migración y carga inicial de datos.

Xxx

## Diseño de la interfaz de usuario

Se mostrará cómo serán las ventanas, la navegación, etc. En caso de que sea necesario, podrán utilizarse diagramas de transición de estados.

Xxx

## Entorno de construcción

Se indicará el IDE, *frameworks*, simuladores, etc. que se han utilizado.

Xxx

## Plan de pruebas

Se establecerán los criterios para codificar las pruebas realizadas a través de *test driven development* (TDD). Asimismo, se diseñarán el resto de pruebas que sean necesarias: estrés, carga, evaluación de la accesibilidad, a través de *scripting* (p.ej., con Selenium), con usuarios reales, etc.

Xxx

## Diagrama de infraestructuras de nivel 3

Se mostrarán las diferentes zonas de seguridad, *firewalls*, VLANs, servidores, etc. Los componentes de la infraestructura podrán ser tanto materiales como virtuales.

Xxx

## Diagrama de infraestructuras de nivel 2 (si procede)

Se mostrarán los *routers*, *switches*, conexiones entre ellos, etc. Para cada uno de los dispositivos se indicarán sus características técnicas.

Xxx

# Construcción

## Referencia al repositorio de software

Se indicará la URL del repositorio en que está el software para, si procede, que el tribunal pueda probarlo en sus propias máquinas.

Xxx

## Manuales

En caso de que no tenga sentido incluir en el fichero README.md del repositorio alguno de los siguientes manuales: instalación, operación o usuario, se incluirá en la memoria del TFG.

Xxx

# Algoritmo de Recomendación

## Objetivo

El objetivo es desarrollar un algoritmo de recomendación que permita dotar a la plataforma de una funcionalidad que recomiende a los alumnos aplicar a puestos de trabajo/prácticas que sean apropiados con su CV. Cuando un nuevo alumno se da de alta en la plataforma e introduce su CV (resumido en la valoración de una serie de cualidades o “skills”) el algoritmo calcula la similitud del CV del nuevo alumno con el CV de alumnos que ya están en la base de datos y los cuales han aplicado con éxito a puestos de trabajo/prácticas. El algoritmo recomienda al nuevo alumno puestos de trabajo/prácticas similares a los que han optado alumnos que tienen un CV similar.

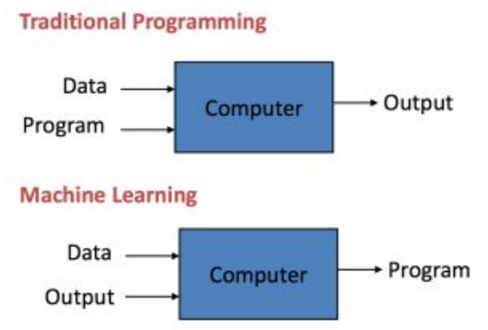
## Introducción a la ciencia de los datos

La automatización de tareas y la capacidad de extraer valor de los datos es el presente y el futuro cercano de cualquier tipo de actividad/función/negocio que implique análisis y toma de decisiones. Otras eras de la historia de la humanidad se han caracterizado por manipular materias primas físicas: cobre, hierro, bronce, etc. Nosotros nos ha tocado vivir la “era de los datos”, y es el dato el nuevo material del que tenemos que extraer valor.

La ciencia de datos es un campo apasionante que combina habilidades estadísticas y cuantitativas avanzadas con programación con el objetivo de sacar valor a la inmensa cantidad de datos con los que nos encontramos en nuestro entorno académico o profesional, que es a lo que nos referimos cuando hablamos de “Big Data”. En general estas técnicas de análisis y explotación de datos es lo que popularmente se ha llamado “Inteligencia Artificial” y, más en concreto, “Machine Learning”, que son algoritmos con la habilidad de aprender sin haber sido explícitamente programados.

## Machine Learning: métodos de aprendizaje

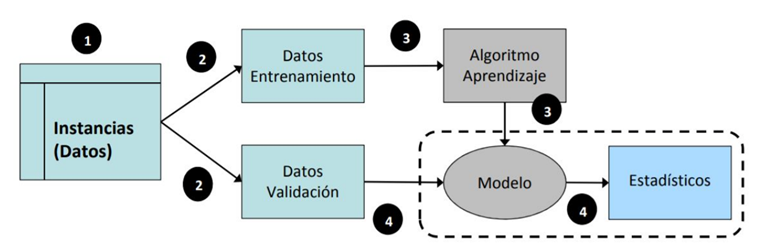
Machine Learning son un conjunto de métodos capaces de detectar automáticamente patrones en los datos con el objetivo de definir modelos que permitan representar de una forma simplificada la realidad. Machine Learning implica definir algoritmos que se encarga de que el modelo aprenda de los datos (se ajuste a los datos), es decir, el algoritmo se encarga de crear una función matemática para representar la realidad a partir de los datos. La diferencia fundamental entre la programación tradicional y la de Machine Learning radica en los elementos de entrada y de salida que tiene un modelo y el otro:



*Ilustración 2: diferencia entre la programación tradicional y el ML*

Los métodos de aprendizaje implican la consecución de una serie de pasos:

* Los datos iniciales son preprocesados, se presentan al sistema de aprendizaje.
* Los datos se dividen en dos grupos. Los datos de entrenamiento servirán para generar el modelo. Los datos de validación servirán para valorar la calidad del modelo.
* El algoritmo de aprendizaje, genera el patrón que mejor se adecua a los datos.
* Se valora la calidad del modelo utilizando los datos validación, para lo cual se utilizan diferentes estadísticos/métricas.



*Ilustración 3: pasos de un modelo de aprendizaje*

Las técnicas y algoritmos de Machine Learning analizan datos con dos objetivos:

* Predecir: intentan responder a la pregunta, ¿Qué podría ocurrir? Implica ser capaz de predecir qué valores de salida se darán para unos valores de entrada determinados.
* Informar: intentan responder a la pregunta, ¿Qué ha ocurrido? Implica extraer información acerca de cómo el modelo, para unas variables de entrada, produce unas variables de salida.

En función del objetivo que persiguen hay dos familias de métodos de aprendizaje:

* Métodos de aprendizaje predictivos o supervisados: el objetivo es estimar una variable a futuro, en función del tipo de variable distinguimos entre:
  + Clasificación: la variable es categórica pudiendo ser binaria (ej. sí/no, paga/no paga…), múltiple (ej. compra a, b o c) o ordenada (ej. riesgo alto, medio, bajo)
  + Regresión: la variable es numérica (ej. precio, cantidad, tiempo…)
* Métodos de aprendizaje descriptivos o no supervisados: el objetivo es explorar los datos
  + Agrupamiento o “Clustering”: el objetivo es agrupar los datos en conjuntos (“clusters”) que se parezcan dentro de un mismo conjunto y que se diferencien de otros conjuntos (Ej. clientes con hábitos de compra similares)
  + Reglas de asociación: buscan reglas (reglas de co-ocurrencia) que describen la mayor parte posible de los datos (ej. productos que se compran juntos)
  + Análisis de Componentes Principales (ACP)

## Técnicas de análisis de datos prescriptivas: sistemas de recomendación

Como acabamos de ver hay dos tipos de técnicas de análisis de datos, las predictivas que nos permiten estimar lo que no conocemos y las descriptivas que nos permiten explorar, analizar y entender mejor los datos. Pero existe un tercer nivel de análisis de datos como extensión natural a los dos anteriores, la analítica prescriptiva que tiene como objetivo realizar recomendaciones sobre las acciones que se han de seguir, por ejemplo, para reducir costes o mejorar los beneficios. Algunos casos de uso típicos de la analítica prescriptiva son: generar planificaciones, establecer estrategias de precios, anticipar la afluencia de público a comercios, etc.

La gran cantidad de sitios especializados en Internet, ofertando millones de productos y/o servicios para su consumo, se ha convertido en un caos de información. Los sistemas de recomendación surgen como solución a este problema. Un sistema de recomendación recibe información del usuario acerca de productos y/o servicios en los que el usuario se encuentra interesado y le recomienda aquéllos cercanos a sus necesidades. Los sistemas de recomendación utilizan métodos matemáticos y estadísticos para explorar los datos y crear recomendaciones adaptadas a cada usuario. Los algoritmos de recomendación son utilizados por Netflix para remendarnos películas, por Spotify para recomendarnos canciones y por Amazon para recomendarnos productos.

La finalidad de un sistema de recomendación es predecir la valoración que un usuario va a hacer de un ítem (libros, películas, páginas, productos, anuncios, etc.) que todavía no ha evaluado. Esta valoración se genera al analizar o las características de cada ítem o las valoraciones de cada usuario a cada ítem. Los sistemas de recomendación se utilizan para recomendar contenido personalizado a los usuarios en base a su perfil, preferencias o intereses.

A la hora de definir y diseñar un sistema de recomendación podemos optar por tres enfoques:

* Filtrado colaborativo: las recomendaciones a un usuario se basan en las preferencias de “usuarios similares”
* Filtrado basado en contenido: las recomendaciones se basan en sus propias preferencias, buscando elementos similares por los que ha mostrado interés
* Filtrado basado en reglas: las recomendaciones se basan en reglas previamente definidas como, por ejemplo, en características sociodemográficas.

Hay dos tipos de filtrados colaborativos:

* Basados en usuario (user-based): para predecir la valoración que un usuario A hará de un ítem X que todavía no ha visto, se buscan usuarios con perfiles similares a A y se utilizan las valoraciones de estos otros usuarios sobre el ítem X como estimación de la valoración de A.
* Basados en ítems (item-based): para predecir la valoración que un usuario A hará de un ítem X que todavía no ha visto, se buscan otros ítems similares (en función del perfil de valoraciones que han recibido) y que el usuario A también haya valorado. Se utilizan las valoraciones que el propio usuario A ha hecho de los ítems similares como predicción de su valoración sobre el ítem X. Este sistema puede parecer similar al basado en contenido, la diferencia se encuentra en que cada ítem no está definido por sus atributos sino por el perfil de valoraciones que ha recibido.

He optado para diseñar el sistema de recomendación de Jobs por un enfoque de filtrado colaborativo. Mi objetivo es recomendar a los usuarios de Jobs ofertas de trabajo o puestos en prácticas a las que son más afines en función de su perfil, es decir, su CV es afín a los requerimientos de la oferta.

## Medidas de similitud o distancia

Todos los sistemas de recomendación tienen una cosa en común, para poder llevar a cabo las predicciones, necesitan definir y cuantificar la similitud entre ítems o usuarios.

Los sistemas de recomendación utilizan el término distancia como cuantificación de la similitud o diferencia entre observaciones. Si se representan las observaciones en un espacio p dimensional, siendo p el número de variables asociadas a cada observación (ítem o usuario), cuando más se asemejen dos observaciones, más próximas estarán, de ahí que se emplee el término distancia. Hay diferentes medidas de similitud:

* Correlación de Pearson: se emplea para calcular la correlación lineal entre dos vectores. Es la medida más apropiada si hay datos nulos.
* Correlación restringida de Pearson: utilizada valores medianos en lugar de medias de rating de dos usuarios.
* Coseno: es una de las medidas de similitud más utilizada. Trata de encontrar cómo dos vectores están relacionados entre ellos midiendo el coseno del ángulo entre los dos vectores. Considera los datos nulos cómo válidos.
* Jaccard: considera el número de preferencias compartidas entre dos usuarios y asume que dos usuarios serán más parecidos cuando tengan valores informados (ratings informados) similares.
* Diferencias de medias cuadradas: considera los ratings absolutos, en lugar de qué ítems han sido valorados. Normalmente se suelen combinar Jaccard.
* PIP: Proximity calcula la diferencia aritmética entre dos ratings. Impact calcula si un ítem le gusta o no al usuario. Popularity calcula la media aritmética del rating de ítems del usuario.
* Correlación de Spearman: Considera el ranking en lugar del rating para calcular la similitud.
* Correlación Tau de Kendall: considera los rankings relativos en lugar del rating para calcular la similitud.

Para el sistema de recomendación de Jobs en optado por utilizar el coseno como medida de similitud. El coseno del ángulo que forman dos vectores puede interpretarse como una medida de similitud de sus orientaciones, independientemente de sus magnitudes. Si dos vectores tienen exactamente la misma orientación (el ángulo que forman es 0º) su coseno toma el valor de 1, si son perpendiculares (forman un ángulo de 90º) su coseno es 0 y si tienen orientaciones opuestas (ángulo de 180º) su coseno es de -1.

## Lenguaje utilizado: R

De entre los diferentes lenguajes de programación que se pueden utilizar para el análisis de datos he decidido utilizar R para programar el algoritmo de selección.

He elegido R ya que he considerado que presenta ventajas significativas: excelente gama de paquetes de código abierto y de alta calidad, dispone funciones y métodos estadísticos integrales muy completos y permite una muy buena visualización de datos con el uso de bibliotecas como ggplot2. También he considerado que R tiene como principal desventaja el rendimiento, que no es el mejor, es decir, R es lento en comparación con otros lenguajes de programación, pero para programar el algoritmo de recomendación es lo suficientemente rápido.

## Sistema de recomendación de JOBS

### Estructura del sistema de recomendación

El sistema de recomendación se estructura en dos APIs:

* API“ofertas\_nuevas.R”: calcula la similitud, en función de los requerimientos, entre las ofertas nuevas que las empresas han imputado en el sistema y las ofertas que ya han sido asignadas a alumnos.
* API“alumno\_nuevo.R”: calcula la similitud, en función del CV, del alumno nuevo con alumnos a los que ya se les ha asignado una oferta. Recomienda al alumno nuevo la/s oferta/s nueva/s que son similares a las ofertas asignadas a alumnos similares a él. Se considera que si dos alumnos son similares pueden recibir ofertas similares, es decir, tienen afinidad a ofertas similares.

### API “ofertas\_nuevas.R”

Este API hace post a una función que tiene dos parámetros: “a” es el valor inferior del código de oferta nueva (job\_id) y “b” es el valor superior del código de oferta nueva. Este modelo parte de dos imputs: la matriz de datos de ofertas asignadas (70 ofertas x 18 variables) y la matriz de ofertas nuevas (10 ofertas x 18 variables), que contienen información que no es relevante para el análisis (5 variables) e información de los requerimientos de cada oferta (13 variables) que es la base del análisis. Se genera la matriz ofertas (80x18) que es la unión de la matriz de ofertas asignadas y la matriz de ofertas nuevas. Se eliminan las variables (columnas) no relevantes y se obtiene una matriz de ofertas (80x13) que contiene los requerimientos de cada oferta.

Se transpone la matriz de ofertas (13x80) y, utilizando el método del coseno, se calcula la similitud (por columnas) entre las 80 ofertas. Cada oferta es un vector de 13 variables (requerimientos), si dos vectores son iguales el ángulo que forman es 0º y su coseno toma el valor de 1 (máxima similitud), mientras que si son totalmente diferentes forman un ángulo de 90º y su coseno es 0 (similitud nula).

La matriz de similitud entre ofertas (80x80) muestra la similitud de cada oferta con las otras 79, evidentemente cada oferta tiene similitud 1 consigo misma. Por ejemplo, si una oferta tiene una similitud de 0,9348 con otra, implica que los vectores de dichas ofertas forman un ángulo de 20,80º y su coseno es 0,9348. Lo cual implica qué de los 13 requerimientos de dichas ofertas, 10 son iguales y 3 (20,80º/90º x 13) son diferentes.

Se genera la matriz de vecinos (80x6) que contiene las 6 ofertas más similares a cada oferta (6 ofertas). La primera oferta más similar a una determinada oferta es ella misma con una similitud de 1. Una vez eliminada la columna del primer vecino se obtiene una matriz de vecinos (80x5) sobre la cual se genera la matriz vecinos ofertas nuevas (10x5), para lo cual se define un rango de códigos de ofertas (job\_id) que tiene como valor inferior el parámetro “a” (a = 71) y como valor superior el parámetro “b” (b =80). Finalmente se salva el archivo "vecinos\_ofertas\_nuevas.xlsx" que se convierte en un input del API “alumno\_nuevo.R”. El output de este API es un array que muestra para cada una de las 10 ofertas nuevas las 5 ofertas más similares de entre las ofertas asignadas.

### API “alumno\_nuevo.R”

Este API hace post a una función que tiene un parámetro, “alumnonuevo”, que corresponde con el código del alumno (alumno\_id) nuevo. Este modelo parte de dos imputs nuevos: la matriz de alumnos antiguos (70 alumnos x 43 variables) y el vector del alumno nuevo (1 alumno x 43 variables), que contienen información que no es relevante para el análisis (7 variables) e información del CV de los alumnos estructurado en skills (36 variables). Este modelo también tiene cómo input la matriz de ofertas nuevas (también es input del API “ofertas\_nuevas.R”) y la matriz de vecinos ofertas nuevas (output del API “ofertas\_nuevas.R”).

Se genera la matriz de alumnos (71x43) que es la unión de la matriz de alumnos antiguos y el vector del alumno nuevo. Se eliminan las variables (columnas) no relevantes y se obtiene una matriz de alumnos (71x36) que contiene el CV de cada alumno estructurado en 36 skills.

Se transpone la matriz de alumnos (36x71) y, utilizando el método del coseno, se calcula la similitud (por columnas) entre los 71 alumnos. Cada alumno es un vector de 36 variables (skills). La matriz de similitud entre alumnos (71x71) muestra la similitud de cada alumno con los otros 70, evidentemente cada alumno tiene similitud 1 consigo mismo. Por ejemplo, si un alumno tiene una similitud de 0,9059 con otro alumno, implica que los vectores de dichos alumnos forman un ángulo de 25,05º y su coseno es 0,9059. Lo cual implica que de las 36 skills, 26 son iguales y 10 (25,05º/90º x 36) son diferentes.

Se genera la matriz de vecinos (71x6) que contiene los 6 alumnos más similares a cada alumno. El primer alumno más similar a un determinado alumno es él mismo con una similitud de 1. Una vez eliminada la columna del primer vecino se obtiene una matriz de vecinos (71x5) sobre la cual se identifican los 5 vecinos del nuevo alumno (1x5), para lo cual se iguala el código de alumno (alumno\_id) al parámetro “alumnonuevo” (alumnonuevo =71).

Se identifican los vecinos comunes entre el vector de los 5 vecinos del nuevo alumno y la matriz de los 5 vecinos de las 10 ofertas nuevas. Dichos vecinos comunes permiten identificar los códigos (job\_id) de las ofertas nuevas a recomendar al alumno nuevo (rec\_alumno71). Se unen los data frame rec\_alumno71 y ofertas\_nuevas para generar las ofertas nuevas que se recomienda considerar al nuevo alumno. Finalmente se salva el archivo en dos formatos: "REC\_alumno71.xlsx" y “REC\_alumno71.json". El output de este API es un array que muestra al alumno nuevo la/s oferta/s nueva/s que son similares a las ofertas asignadas a alumnos similares a él.

### Llamada de las APIs

Xxx

### Intercambio de ficheros Web-APIs

Xxx

# Conclusiones y líneas futuras

Ideas a las que se llega después del desarrollo del proyecto, así como las líneas posibles de trabajo posterior.

Xxx

# Bibliografía

Bibliografía citada en la memoria. Seguirá el [formato APA](https://apastyle.apa.org/).

Ejemplos de referencias:

1. Grady, J. S., Her, M., Moreno, G., Perez, C., & Yelinek, J. (2019). Emotions in storybooks: A comparison of storybooks that represent ethnic and racial groups in the United States. Psychology of Popular Media Culture, 8(3), 207–217. <https://doi.org/10.1037/ppm0000185>
2. Jerrentrup, A., Mueller, T., Glowalla, U., Herder, M., Henrichs, N., Neubauer, A., & Schaefer, J. R. (2018). Teaching medicine with the help of “Dr. House”. PLoS ONE, 13(3), Article e0193972. <https://doi.org/10.1371/journal.pone.0193972>
3. Schaefer, N. K., & Shapiro, B. (2019, September 6). New middle chapter in the story of human evolution. Science, 365(6457), 981–982. <https://doi.org/10.1126/science.aay3550>
4. Carey, B. (2019, March 22). Can we get better at forgetting? The New York Times. <https://www.nytimes.com/2019/03/22/health/memory-forgetting-psychology.html>
5. Rabinowitz, F. E. (2019). Deepening group psychotherapy with men: Stories and insights for the journey. American Psychological Association. <https://doi.org/10.1037/0000132-000>
6. Aron, L., Botella, M., & Lubart, T. (2019). Culinary arts: Talent and their development. In R. F. Subotnik, P. Olszewski-Kubilius, & F. C. Worrell (Eds.), The psychology of high performance: Developing human potential into domain-specific talent (pp. 345–359). American Psychological Association. <https://doi.org/10.1037/0000120-016>
7. Harvard University. (2019, August 28). Soft robotic gripper for jellyfish [Video]. YouTube. <https://www.youtube.com/watch?v=guRoWTYfxMs>
8. APA Databases [@APA\_Databases]. (2019, September 5). Help students avoid plagiarismWeb emoji of crossing hands and researchers navigate the publication process. More details available in the 7th edition @APA\_Style table [Tweet]. Twitter. <https://twitter.com/APA_Databases/status/1169644365452578823>
9. Fagan, J. (2019, March 25). Nursing clinical brain. OER Commons. Retrieved September 17, 2019, from <https://www.oercommons.org/authoring/53029-nursing-clinical-brain/view>
10. National Institute of Mental Health. (2018, July). Anxiety disorders. U.S. Department of Health and Human Services, National Institutes of Health. <https://www.nimh.nih.gov/health/topics/anxiety-disorders/index.shtml>
11. Woodyatt, A. (2019, September 10). Daytime naps once or twice a week may be linked to a healthy heart, researchers say. CNN. <https://www.cnn.com/2019/09/10/health/nap-heart-health-wellness-intl-scli/index.html>
12. World Health Organization. (2018, May 24). The top 10 causes of death. <https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death>

# Anexo I

# Otras posibilidades para realizar el análisis y el diseño

A lo largo de este documento se ha asumido un desarrollo orientado a objetos. No obstante, se pueden seguir otros enfoques, por ejemplo, un desarrollo estructurado con diagrama de contexto, diagramas de flujos de datos, diagrama entidad-relación, etc.

Asimismo, es posible utilizar el diagrama entidad-relación de forma complementaria a los diagramas de clases.

Por otra parte, el tutor del proyecto puede dar sus propias orientaciones si lo considera oportuno.

# Anexo II

# Fuente de inspiración del presente documento

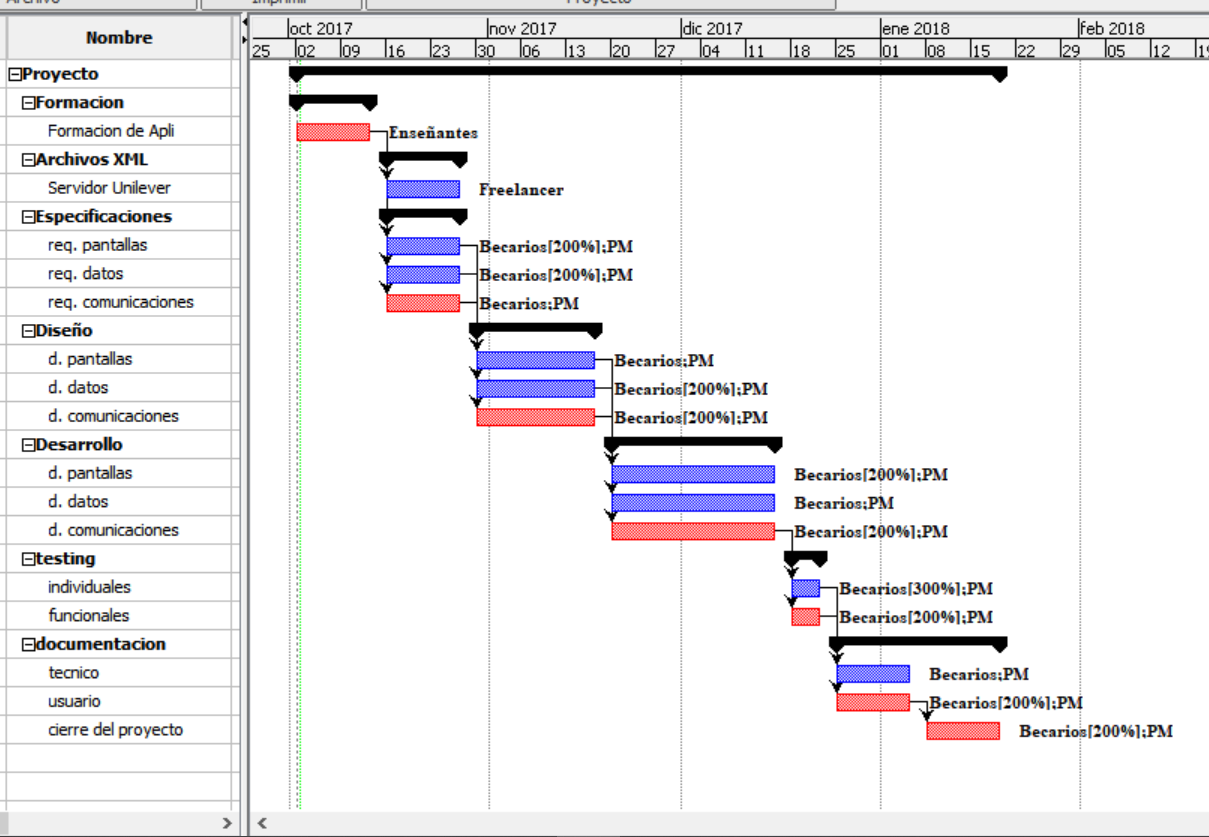
Independiente de que su uso que esté actualmente más o menos extendido, se ha considerado que la metodología [Métrica 3](https://administracionelectronica.gob.es/pae_Home/pae_Documentacion/pae_Metodolog/pae_Metrica_v3.html) es adecuada como marco de referencia en la elaboración de la documentación. En consecuencia, ha sido una fuente de inspiración.

# Glosario de términos

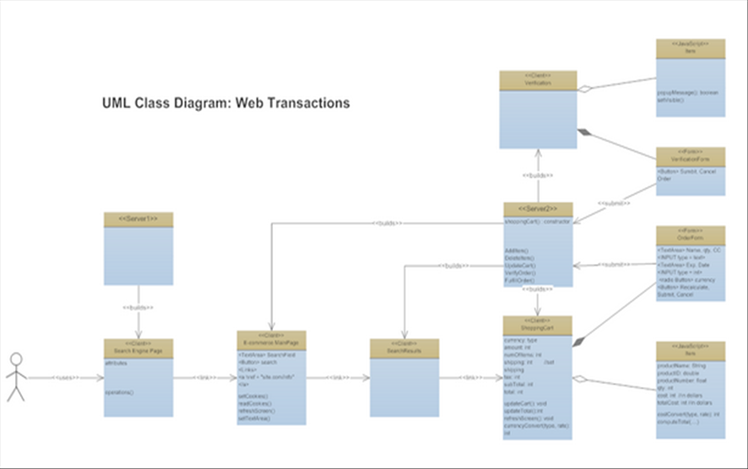
Si es necesario.

# Anexos

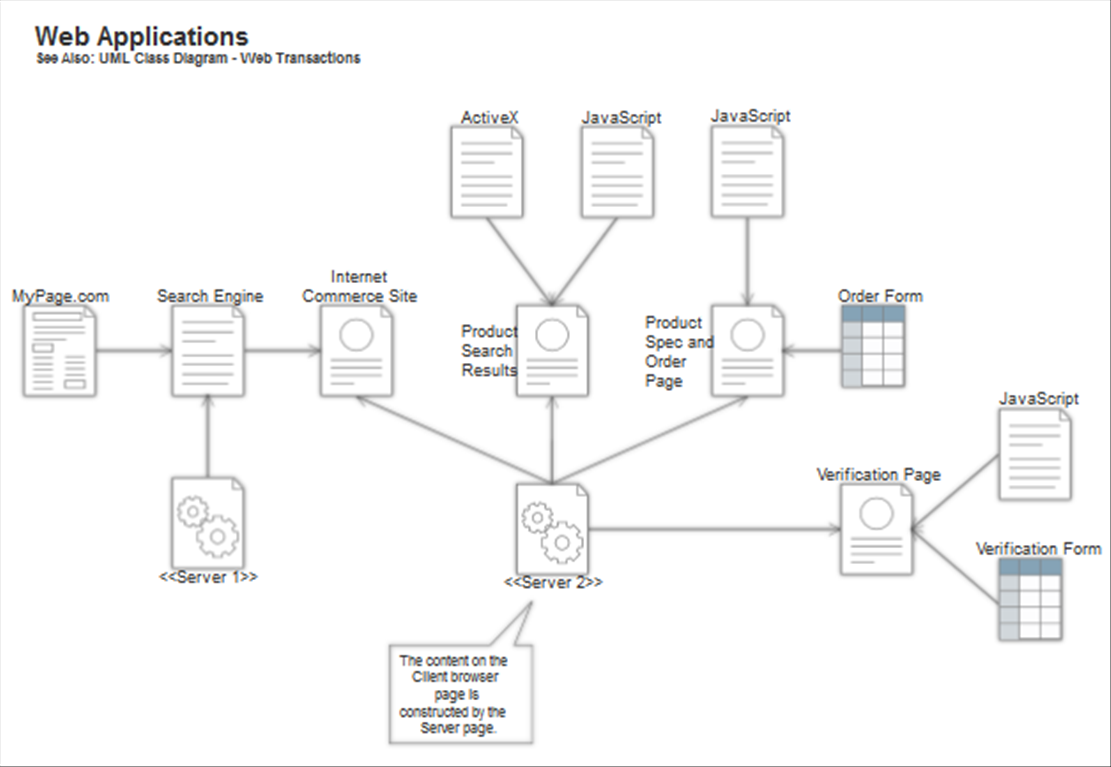
Ejemplo de planificación de un TFG



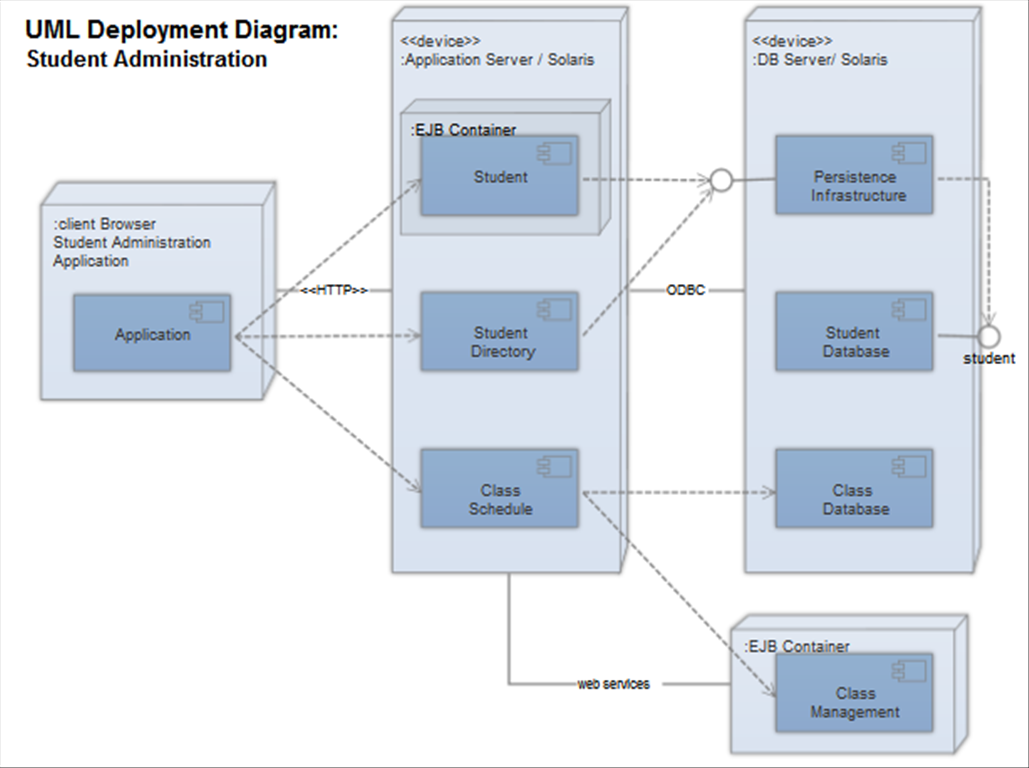
Ejemplo de diagrama de clases



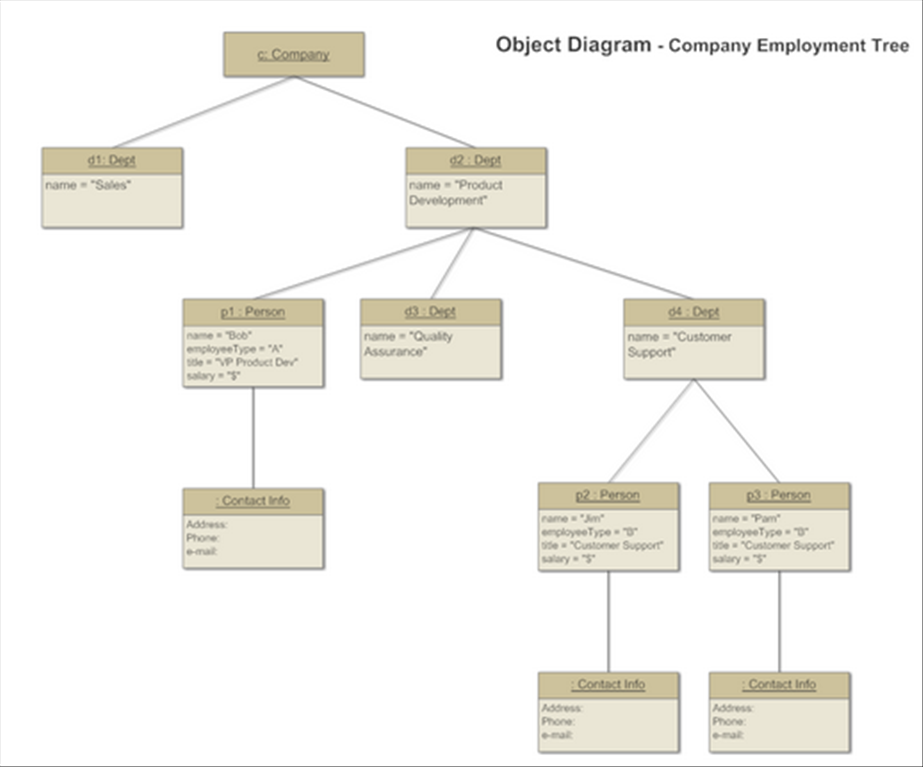
Ejemplo de diagrama de componentes



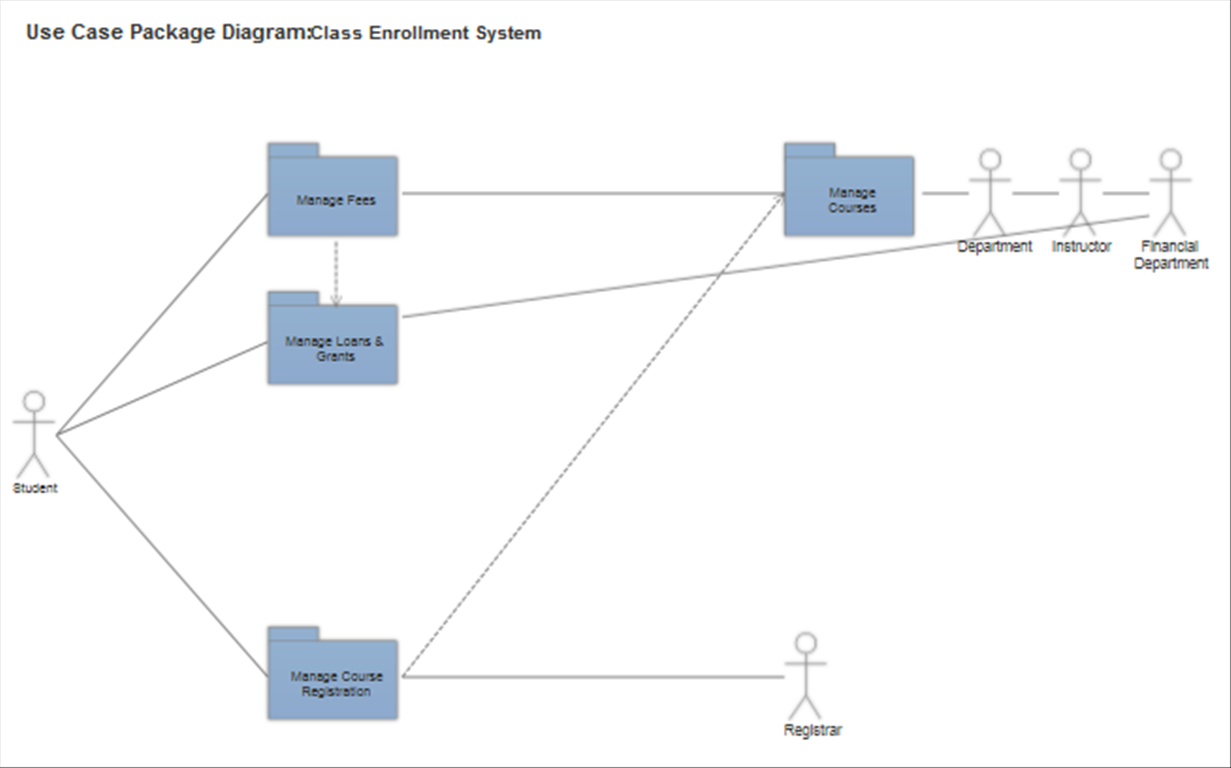
Ejemplo de diagrama de despliegue



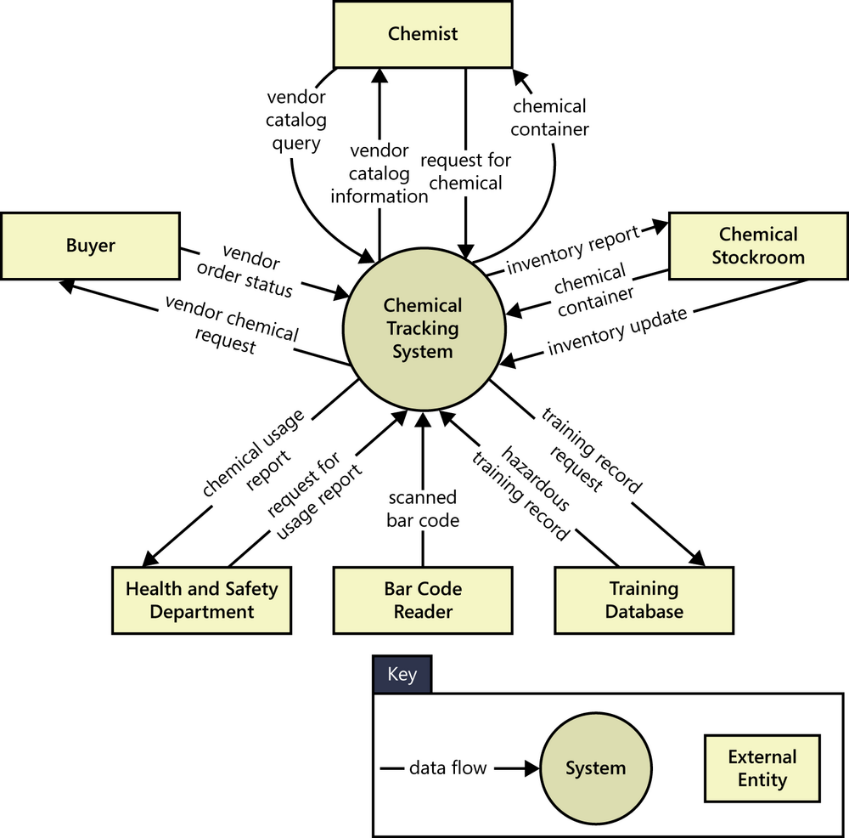
Ejemplo de diagrama de objetos



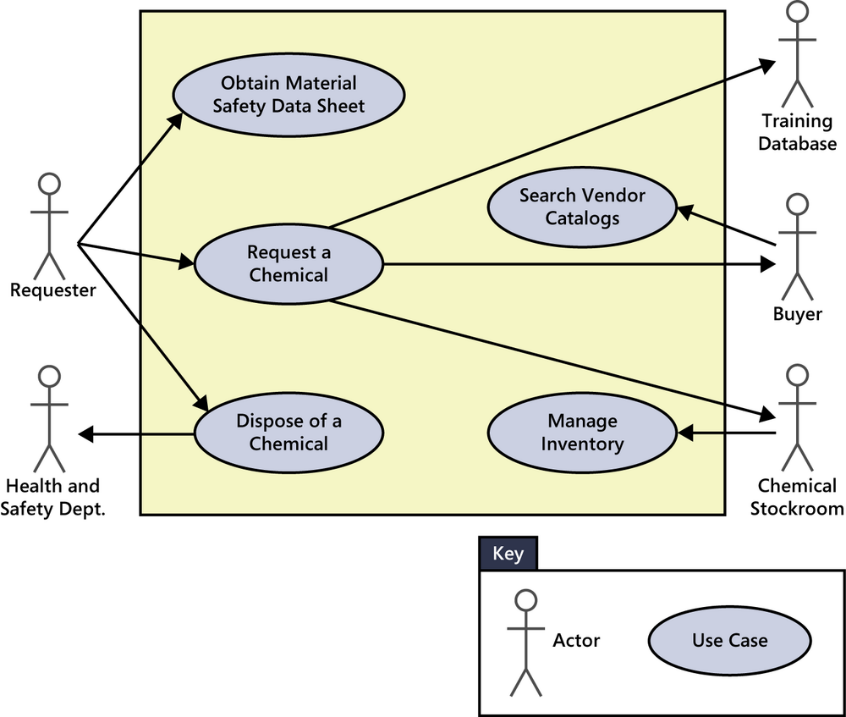
Ejemplo de diagrama de paquetes



Ejemplo de diagrama de contexto



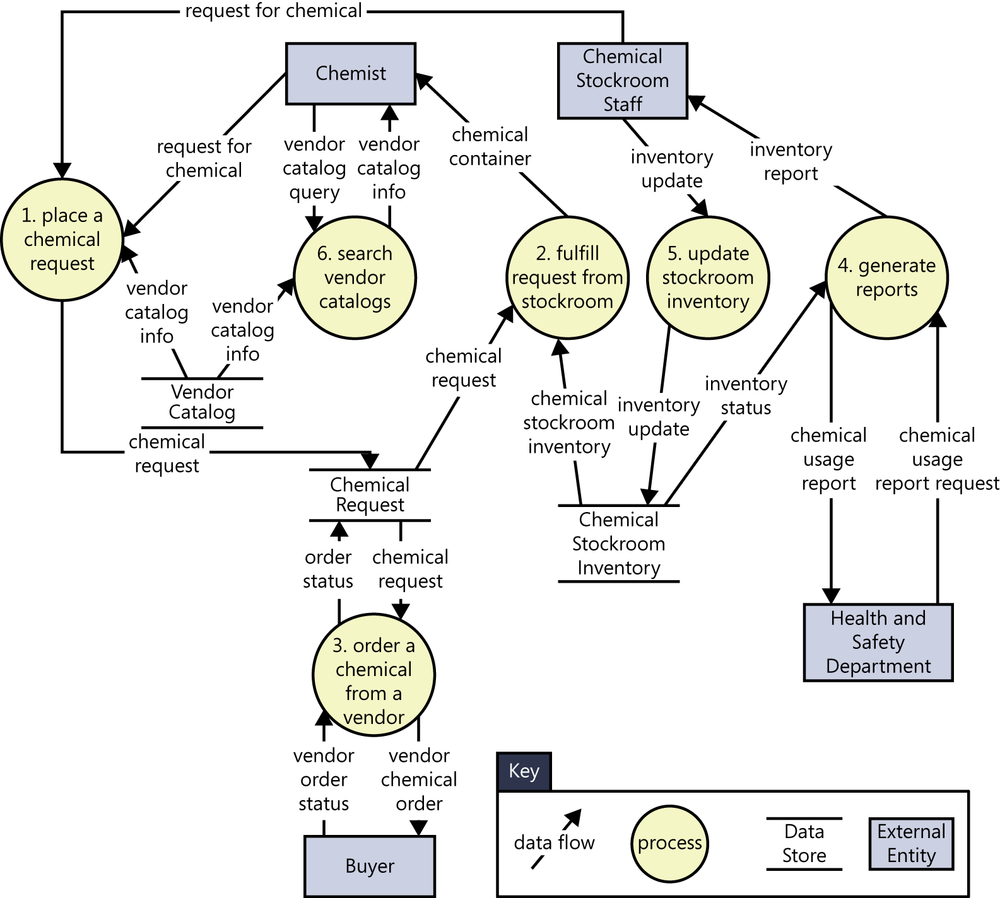
Ejemplo de caso de uso



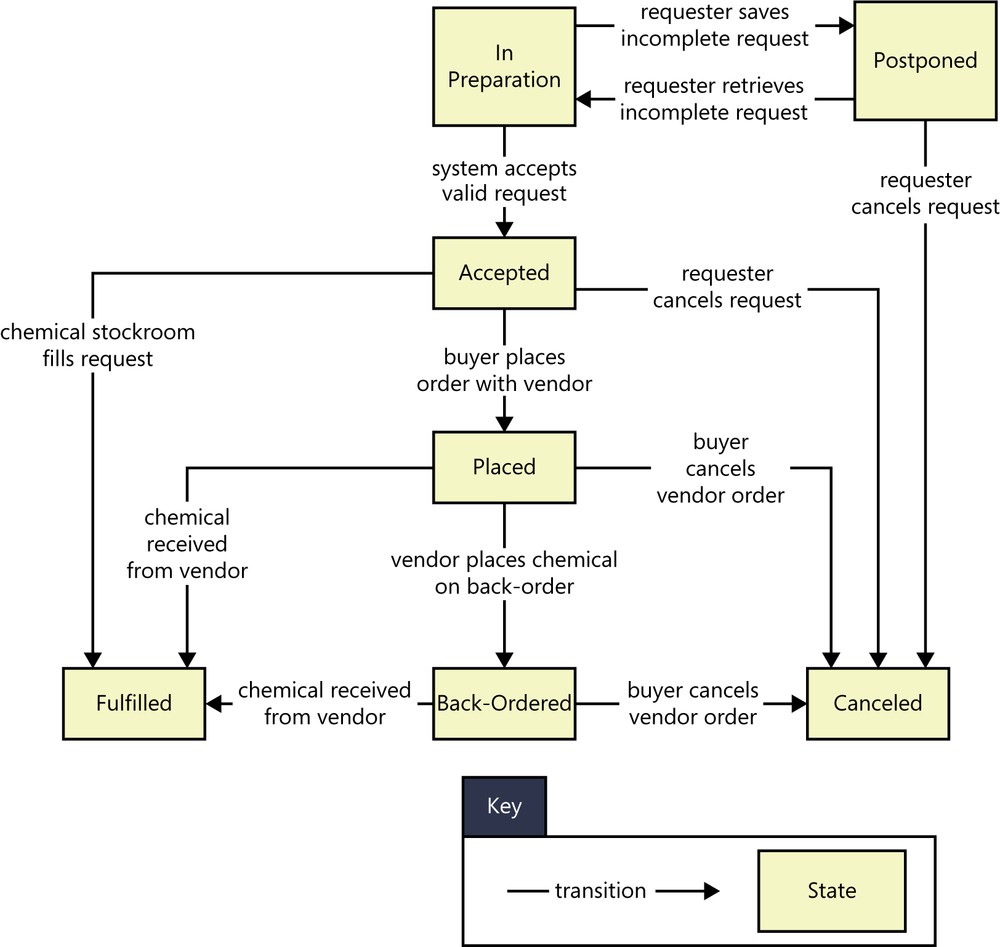
Ejemplo de diagrama de actividades

An illustration showing the normal flow of a use
                  case in the left part of the figure. Starting with a solid
                  dot that represents the use case preconditions, arrows lead
                  downward through a series of steps in rectangles to the use
                  case postconditions. In the middle of the stack of
                  rectangles is a diamond representing a branch point
                  condition. An arrow goes from the diamond to the right,
                  where there is another series of vertical boxes and arrows
                  representing steps in the alternative flow. The last step in
                  the alternative flow has an arrow going back to the left to
                  rejoin the last step in the normal flow.

Ejemplo de DFD Diagrama de Flujo de datos



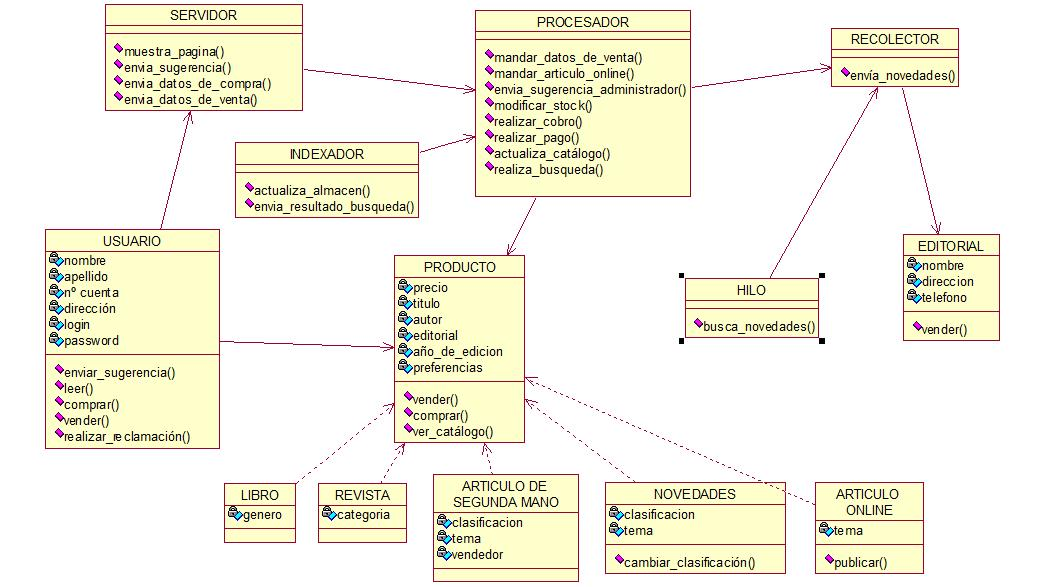
Ejemplo de diagrama de transición



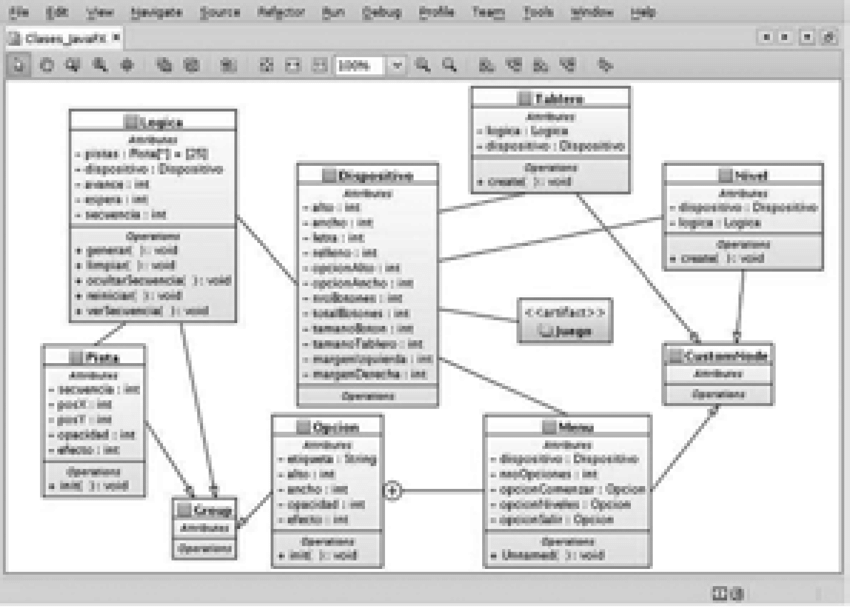
Ejemplo de diagrama lógico de una vpn



Ejemplo de Diagrama E/R



Ejemplo de diagrama de clases de usuarios



Ejemplo de interfaz HMI

