

Memoria

Sistema de enlace entre Sap Business One y una herramienta de gestión de proyectos

18 de Abril de 2022

INGENIERÍA DEL SOFTWARE

ADRIÁN JAÉN ZAFRA

DIRECTOR: GABRIEL QUINCOCES

PONENTE: JOAN ANTONI PASTOR

ÍNDICE

1. CONTEXTO	4
1.1. INTRODUCCIÓN	4
1.2. FORMULACIÓN DEL PROBLEMA	4
1.3. STAKEHOLDERS	5
2. JUSTIFICACIÓN	7
3. ALCANCE DEL PROYECTO	8
3.1. NECESIDADES DEL CLIENTE	8
3.1.2 SUBOBJETIVOS	8
3.2. OBSTÁCULOS Y RIESGOS	9
4. METODOLOGÍA DE TRABAJO	11
5. TAREAS	12
5.1 Preparación	12
5.2 Business BluePrint	12
5.3 Realización	13
5.4 Preparación final	13
5.5 Entrada en productivo y soporte	14
6. RECURSOS	14
6.1 Recursos humanos	14
6.2 Recursos tecnológicos	14
7. ESTIMACIONES	15
8. DIAGRAMA DE GANTT	16
9. GESTIÓN DE RIESGO	17
10. CONTROL Y SEGUIMIENTO	17
10.2 Estimaciones	18
10.3 Diagrama de Gantt	20
10. PRESUPUESTO	20
10.1 Identificación y estimación de costes	20
10.1.1 Recursos humanos	20
10.1.2 Recursos materiales	22
10.1.3 Recursos indirectos	23
10.1.4 Contingencia	23
10.1.5 Imprevistos	23
10.1.6 Presupuesto final	24
10.2 Control de seguimiento	25
10.2.1 Costes humanos	25
10.2.2 Contingencia	26

10.2.3 Presupuesto final	26
10.2 Control de gestión	26
11. INFORME DE SOSTENIBILIDAD	27
11.1 Dimensión ambiental	27
11.2 Dimensión económica	27
11.3 Dimensión social	27
12. LEYES Y REGULACIÓN	28
13. ESPECIFICACIÓN DEL SISTEMA	28
13.1 Diagrama de casos de uso	28
13.2 Especificación de casos de uso	29
13.3 Requisitos no funcionales [8]	31
14. ARQUITECTURA DEL SISTEMA	35
14.1 Arquitectura de SAP Business One	36
14.1.1 Base de datos	37
14.2 Arquitectura de Redmine	39
14.3 Arquitectura del sistema de enlace	40
14.3.1 Diagrama UML	42
14.3.2 Restricciones textuales	43
14.3.3 Patrones implicados	43
14.4 Diagramas de secuencia	44
14.4.1 Subir proyectos	44
14.4.2 Bajar datos de proyectos	45
15. EJEMPLO DE USO REAL	47
15.1 Configuración del sistema	48
15.2 Subida de proyectos	50
15.2 Bajar datos de proyectos	53
16. TESTING	54
16. CONCLUSIONES	55
17. BIBLIOGRAFIA	57

1. CONTEXTO

1.1. INTRODUCCIÓN

El trabajo de final de carrera (TFG) tiene como objetivo mostrar la adquisición de las competencias del grado en un ejercicio que permita combinar los contenidos prácticos y teóricos. En este caso se ha optado por un trabajo de modalidad B, debido a que permite al estudiante realizar este trabajo en un entorno empresarial y profesional que exige una alta calidad tanto por parte de la empresa para la que se trabaja, como por parte del cliente que contrata el proyecto. Tanto la temática como la forma de trabajar han sido propuestas por parte de la empresa.

En este trabajo se pretende crear una integración entre Sap Business One (Un *ERP*¹ pensado para pequeñas y medianas empresas) y Redmine (Una herramienta de gestión de proyectos).

Con esta integración se pretende optimizar la creación de los proyectos en Redmine en el momento de gestionar los contratos en SAP con el objetivo de facilitar la operativa del responsable de área.

1.2. FORMULACIÓN DEL PROBLEMA

Hoy en día, cada vez es más común trabajar con herramientas de gestión de proyectos. Esto se debe a que, este tipo de empresas que trabajan en base a proyectos, se enfrentan a múltiples retos diariamente para que el proyecto pueda seguir adelante con total normalidad. Comúnmente los proyectos se dividen en tareas, que tienen asignadas un tiempo y un orden, si cualquiera de estas dos variables se ve afectada, es posible que se vea afectado todo el proyecto.

Todo esto hace que la gestión del proyecto pase a ser un asunto fundamental, y en este caso en concreto, es conveniente que esté directamente comunicada con el ERP que utiliza la empresa, para de esta forma agilizar y automatizar una gran cantidad de procesos que anteriormente se realizaban de forma manual, y en algunos casos, poco efectiva.

En este proyecto en cuestión, nos centraremos principalmente en la asignación de proyectos, la gestión de las horas dedicadas a cada proyecto, y la facturación del proyecto.

Además, en este proyecto este tipo de gestión que hemos comentado anteriormente será en algunos casos un poco más compleja, ya que afecta a dos empresas, en primer lugar Soliguer, una empresa dedicada a proyectos de ingeniería agrícola, y en segundo lugar Hesman, filial de Soliguer, que se dedica a la gestión de proyectos energéticos renovables.

¹ Enterprise Resource Planning, es decir, sistema de planificación de recursos empresariales. [3]

De forma general, se trabajará con Soliguer, que es la empresa encargada de realizar los proyectos que se asignan en SAP, pero es probable que algunos de estos proyectos los realice Hesman, por lo tanto, se deberá encontrar una solución que permita asignar a Hesman un proyecto de Soliguer, pero que posteriormente la facturación se realice a la esta última empresa, ya que será la que contrate el cliente directamente.

Además, esta solución mencionada anteriormente, se tiene que poder ejecutar de diferentes opciones. Como opción principal, se creará un *Servicio de Windows* que se ejecutará una vez cada cierto tiempo, y realizará tanto la carga como la descarga de los datos sobre los proyectos, pero además, deberá existir la posibilidad de poder ejecutar este programa a través de un punto de menú en *SAP Business One*.

1.3. STAKEHOLDERS

- **Cliente:**

Soliguer. Empresa de Ingeniería, Arquitectura y Consultoría especializada en el diseño, ejecución y puesta en marcha de proyectos agroalimentarios.

Su rol es hacer uso del sistema para poder gestionar de una forma más sencilla los proyectos que genera en sap, las horas asignadas a cada proyecto en Redmine, y la facturación de los proyectos.

Sus objetivos son:

- Digitalizar y automatizar el proceso de creación de proyectos en Redmine.
- Digitalizar y automatizar la asignación y el traspaso de las horas dedicadas a un proceso desde Redmine hacia SAP.
- Controlar la facturación de los proyectos asignados.
- Delegar proyectos a la empresa Hesman, automatizando las acciones comentadas en los puntos anteriores.

- **Empresa filial:**

Hesman. Empresa dedicada a la gestión de proyectos energéticos renovables.

Su rol es hacer uso del sistema para gestionar los proyectos asignados por parte de Soliguer.

Sus objetivos son:

- Poder visualizar y gestionar a través de Redmine los proyectos que les han sido asignados.
- Tener el control desde Redmine de las horas realizadas en cada proyecto.

- **Trabajadores del cliente:**

Su rol es reportar las horas trabajadas en cada proyecto que tienen asignado.

Sus objetivos son:

- Simplificar el proceso de reportar las horas trabajadas.
- Poder visualizar los proyectos que tienen asignados y toda la información relevante sobre estos.

- **Empresa desarrolladora:**

MSS Seidor S.L. Empresa dentro del grupo Seidor especializada en soluciones informáticas para pequeñas y medianas empresas. *Partner*² de SAP.

Su rol es ofrecer una solución informática para las necesidades del cliente.



Imagen 1: Logotipo de Seidor

Sus objetivos son:

- Satisfacer las necesidades del cliente.
- Obtener un beneficio económico.

- **Director del proyecto:**

Su rol es organizar y supervisar el desarrollo y la implantación del proyecto.

Sus objetivos son:

- Satisfacer las necesidades del cliente.
- Proporcionar metodologías para el correcto desarrollo del proyecto.
- Repartir la carga de trabajo entre el equipo de desarrollo.

- **Consultor del proyecto:**

Su rol es analizar las necesidades del cliente y el tipo de empresa, para de esta forma utilizar sus conocimientos sobre SAP Business One y presentar una solución que cubra dichas necesidades.

Sus objetivos son:

- Satisfacer las necesidades del cliente.
- Optimizar los recursos.
- Elaborar una propuesta convincente.
- Encontrar soluciones a futuros problemas relacionados con los procesos empresariales que se pretenden automatizar.

- **Equipo de desarrollo:**

En este caso, el equipo de desarrollo lo formo principalmente yo, con la ayuda de dos desarrolladores más experimentados en este tipo de desarrollos.

² Relación entre empresas para un beneficio mutuo

El rol del equipo de desarrollo es desarrollar el sistema.

Los objetivos son:

- Programar de forma clara con la nomenclatura estándar de la empresa.
- Optimizar los recursos.
- Buscar nuevas soluciones.
- Generar un software que cumpla todos los requerimientos presentados tanto por el cliente, como por el consultor. [2]

2. JUSTIFICACIÓN

Este proyecto que se presenta como TFG, está hecho completamente a medida y tiene el objetivo de satisfacer unas necesidades muy completas del cliente. Antes de presentar la solución actual, se realizó un estudio tanto de las herramientas que se podían utilizar, cómo de la forma en la que se podría realizar la implementación de la interacción entre dichas herramientas.

En primer lugar debemos tener claro qué Seidor es un partner de SAP, por lo tanto, es obvio que una vez se contactó con el cliente, el ERP que se propondría para dicho proyecto sería una versión de SAP. En este caso se escogió SAP Business One ya que está enfocado a pequeñas y medianas empresas.

En segundo lugar tenemos la elección de la herramienta de gestión de proyectos. Para esta selección se tuvieron en cuenta diferentes herramientas, pero la que se escogió finalmente fue Redmine por los siguientes motivos:

- Herramienta muy flexible y potente, permite adaptarse a todas las necesidades de la empresa.
- Con Redmine se pueden gestionar varios proyectos utilizando la misma interfaz, pero con distinta configuración para cada uno de ellos. La transición de un proyecto a otro es muy sencilla y rápida.
- Las tareas y procesos pueden ser gestionados de forma eficiente con Redmine, a través de las “peticiones”.
- Dispone de una API³ muy completa que nos permite acceder a todos los valores necesarios para esta implementación. [1]

Una vez tenemos claras las herramientas principales con las que vamos a trabajar, procedemos a investigar si existe una solución que se pueda adaptar a lo que el cliente busca. En este caso el cliente necesita una solución bastante adaptada, puesto que es necesario crear ciertos formularios en SAP con los campos que ellos necesitan, y además, no encontramos soluciones públicas que realicen esta integración entre SAP y Redmine, por lo tanto debemos implementar esta solución desde cero.

³ Conjunto de protocolos y definiciones que se usan para integrar y desarrollar el software de las apps [5]

3. ALCANCE DEL PROYECTO

En este apartado nos centraremos principalmente en descubrir cuál es el alcance del proyecto. Para ello, expondremos todas aquellas necesidades del cliente que deberemos satisfacer, las cuales serán nuestros objetivos a cumplir, y los obstáculos y riesgos que podemos encontrar a lo largo del proceso.

3.1. NECESIDADES DEL CLIENTE

Como he mencionado anteriormente, un punto muy importante en el desarrollo de este proyecto, era identificar cuáles eran las necesidades principales del cliente, para tomarlas como objetivos durante el desarrollo. Una vez tenemos claras estas necesidades, debemos desglosarlas en subobjetivos, para poder ser mucho más concretos a la hora de identificar los desarrollos y las labores que se llevarán a cabo, además de los obstáculos y riesgos que podrán surgir posteriormente.

OBJETIVOS

- Automatizar el proceso de creación de proyectos en Redmine:
De esta manera solo es necesario crear los proyectos en SAP Business One y automáticamente se crearán en la herramienta de gestión de proyectos.
- Permitir que la empresa filial Hesman pueda trabajar en proyectos de Soliguer:
De esta manera los trabajadores de la empresa Hesman podrán imputar las horas trabajadas en cada proyecto y podrán acceder a la información relevante de estos.
- Acceder desde SAP Business One a las horas imputadas en cada proyecto:
Con esta funcionalidad se podrá guardar dicha información y utilizarse en el correcto reparto del asiento nómina.

Una vez tenemos claros los objetivos, podemos centrarnos en hilar un poco más fino, y buscar todos aquellos subobjetivos que debemos cumplir para que el desarrollo pueda satisfacer a la perfección estas necesidades del cliente.

3.1.2 SUBOBJETIVOS

Para poder estructurar nuestros subobjetivos de la forma más adecuada posible, he decidido que los dividiremos según la herramienta en la que se encuentren, es decir, tendremos algunos subobjetivos que deberemos cumplir durante los desarrollos y las adaptaciones realizadas en SAP Business One y otros que se realizarán directamente en el *Ejecutable/ Servicio de Windows* a través de llamadas a la API.

SAP Business One:

- Desarrollar un AddOn⁴ que realice automáticamente la creación de los campos de usuario y formularios necesarios para el proyecto.
- Tener un punto de menú para ejecutar la creación de dichos campos y formularios.
- Disponer de un formulario para introducir los datos de conexión a Redmine.

Ejecutable/ Servicio de Windows:

- Permitir ejecutar el programa tanto de forma automática como de forma casual.
- Subir a Redmine los proyectos indicados en SAP Business One, con la estructura marcada por la plantilla indicada, y asignando en SAP Business One el identificador que se recibe por parte de Redmine.
- Descargar de Redmine todos aquellos datos relacionados con las horas imputadas en un proyecto, crear un cronograma para almacenar dichos datos en Sap Business One en caso de que sea necesario, o actualizar un cronograma existente en caso de que coincidan tanto el empleado como el intervalo de tiempo.

3.2. OBSTÁCULOS Y RIESGOS

- Escalabilidad del sistema: El número de usuarios que puede llegar a usar este en principio no es muy elevado, pero puede crecer con el paso del tiempo y eso podría ocasionar problemas a la hora de transmitir la información. Además, no sabemos si en el futuro puede haber alguna otra empresa a la que se le deban asignar proyectos de Soliguer, cosa que también podría ser un inconveniente.
- Seguridad: El sistema es solo para usuarios de las empresas nombradas anteriormente, pero de todos modos se debe establecer controles de seguridad para que el usuario no pueda enviar información que pueda dañar el sistema o lo pueda corromper.
- Mantenimiento o caída del servidor: puede haber muchos motivos por los que pudiera caer el servidor y por lo tanto se debe vigilar siempre y tener la información a salvo para no tener riesgo de perderla.
- Problemas de desarrollar alguna funcionalidad: muchas veces funcionalidades que aparentemente parecen fáciles de programar, acaban siendo difíciles. Por estos problemas es muy importante la colaboración entre los miembros del equipo.
- Problemas de protección y mantenimiento de códigos fuente: es importante que los códigos fuente estén protegidos para evitar robos o plagios. También hay que mantener un sistema para que no haya problemas en versiones de códigos. En todo momento debe poder encontrar la versión correcta del programa.

⁴ Software adicional que se añade a tu ERP ayudando a potencializar al máximo el uso del mismo en tu empresa. [11]

4. METODOLOGÍA DE TRABAJO

Tal y como se ha comentado anteriormente, Seidor es un partner muy importante de Sap Business One. Es por este motivo que en esta empresa se han adquirido ciertas metodologías y formas de trabajar que propone SAP. Desde esta empresa se propone una metodología que se rige por las siglas **ASAP** y recibe el nombre de AcceleratedSAP. Tal y como podemos comprobar, se juega con el concepto de “As Soon As Possible”, ya que, como norma general, los proyectos requieren unos tiempos de entrega relativamente cortos. Por lo tanto, en este proyecto se utilizará esta metodología de trabajo que consta de las siguientes fases:

1. Preparación:

En esta fase inicial se planifica el proyecto con detalle. Ésta planificación recoge tanto los procedimientos del proyecto como la constitución del equipo de trabajo. El equipo de trabajo está formado por el director del proyecto, el consultor y los diferentes desarrolladores. Todo esto se formaliza con una reunión de *kick-off*⁵.

2. Business Blueprint:

Es una de las fases más importantes del proyecto, en ella se definen los procesos de gestión que se llevarán a cabo y se identifican los desarrollos que se harán. También se debe hacer un inventario de formularios e informes que se necesitarán. Por último se busca la mejor forma de hacer la conversión inicial de datos y la carga de estos.

3. Configuración:

Se realiza una guía de parametrización dónde aparecen todas las especificaciones técnicas. También se incluye el diseño de los prototipos y se proponen diferentes escenarios de sistema que se utilizará. Se crea un plan de pruebas de integración y carga de datos.

4. Preparación final:

Se elaboran todos los manuales de usuario de la aplicación, adaptándose a los diferentes perfiles de usuario. Es importante hacer un plan de formación para evitar problemas de usabilidad. Finalmente, también se hace un plan de corte y de puesta en productivo.

5. Entrada en productivo y soporte:

Es la fase final, contempla el arranque real del sistema completo. Se hace un seguimiento y control de todas las incidencias que puedan haber. También será necesario soporte especial a algunos usuarios clave. [4]

Para asegurar el cumplimiento de todos los tiempos y fases, se realizan reuniones semanalmente donde cada miembro del equipo expone el progreso y las decisiones tomadas. Si hay algún tema de vital importancia se programa una reunión urgente.

⁵ Iniciación

5.TAREAS

En este apartado me dedicaré a explicar, principalmente, cómo está dividido el proyecto, las tareas que se deben realizar, y el tiempo estimado para estas teniendo en cuenta los diferentes factores que se explicaron en los diferentes subapartados de “Alcance del proyecto”.

En este caso, es un proyecto que se realiza en un periodo de tiempo reducido, puesto que empieza a principios de Septiembre, concretamente el día 6, y es necesario que esté finalizado y recibiendo soporte para el inicio de 2022.

Tal y como se mencionó en el apartado “Metodología”, utilizaremos ASAP como metodología de trabajo para este proyecto, debido a que es la que se utiliza en Seidor. Dicha metodología se compone por 5 fases que se realizan de forma secuencial, es decir, se realizan en orden y cada fase empieza cuando acaba la anterior, nunca antes.

Para ordenar las numerosas tareas que se deben realizar a lo largo del proyecto, situaremos cada tarea dentro de la fase a la que corresponde.

5.1 Preparación

- **P1. Planificación inicial y preparación del proyecto:** Consiste en la realización, por parte del director del proyecto, de una estructura básica y una solución inicial. Todo esto se hace a través de una recolección de información fundamental sobre las necesidades del cliente, el tipo de empresa, su forma de trabajar, los requerimientos...El tiempo estimado es de aproximadamente 20 horas.
- **P2. Identificar y garantizar los miembros del equipo:** Consiste en que el director del proyecto analice qué empleados son los más adecuados para el proyecto, tanto a nivel de consultoría como a nivel de desarrollo. El tiempo estimado es de aproximadamente 8 horas y depende de P1.
- **P3. Reunión de kick-off:** En esta reunión se le presenta al cliente la planificación inicial y el equipo de trabajo. El tiempo estimado es de aproximadamente 4 horas y depende de P2.

5.2 Business BluePrint

- **BBP1. Validar Alcance del proyecto:** Se documenta tanto la estructura organizacional actual, la distribución de información, y la situación de la empresa. El tiempo estimado es de aproximadamente 24 horas.
- **BBP2. Estructura organizacional:** Consiste en la elaboración de Modelos de Negocio y Técnicos que se realizan a partir de los requerimientos identificados. El tiempo estimado es de aproximadamente 32 horas y depende de BBP1.
- **BBP3. Definición de los requerimientos funcionales:** Se elabora un informe en el que se analiza la cobertura de los requerimientos funcionales y técnicos por parte de

la solución SAP y la aparición de posibles *gaps*⁶ tanto funcionales como técnicos. El tiempo estimado es de aproximadamente 32 horas y depende de BBP2.

- **BBP4. Detalle de configuración del sistema:** Se realiza una guía de parametrización dónde aparecen todas las especificaciones técnicas. También se incluye el diseño de los prototipos y se proponen diferentes escenarios de sistema que se utilizará. Se crea un plan de pruebas de integración y carga de datos. El tiempo estimado es de aproximadamente 32 horas y depende de BBP3.

5.3 Realización

- **R1. Configuración del entorno:** Consiste en la instalación y configuración de todas aquellas herramientas que serán necesarias tanto para el desarrollo como para el correcto funcionamiento del proyecto. El tiempo aproximado es de 24 horas.
- **R2. Configuración de la herramienta Redmine:** Consiste en configurar los aspectos necesarios de Redmine para que se pueda empezar a realizar el desarrollo del proyecto, pudiendo realizar y recibir llamadas por parte de Redmine. El tiempo aproximado es de 8 horas.
- **R3. Desarrollo de las llamadas a la API:** Desarrollo de las funciones encargadas tanto de obtener información de Redmine para almacenarla en la base de datos, como de obtener información de la base de datos para enviarla a Redmine. El tiempo aproximado es de 64 horas. Depende de R1 y R2.
- **R4. Definición de entidades y conexión con el web service:** Se definen las entidades con los datos que las forman y se implementan las funciones necesarias para conectar la aplicación con SAP. El tiempo aproximado es de 24h y depende de R3.
- **R5. Programación de formularios en SAP:** Consiste en realizar el desarrollo de los diferentes formularios que serán necesarios en SAP, con las correspondientes validaciones y los enlaces con las funciones que se comunican con Redmine. El tiempo estimado es de 96 horas y depende de R4.
- **R6. Cargas iniciales de datos:** Antes de poder empezar a probar que todo funcione correctamente, es necesario que el cliente nos proporcione datos reales para que las pruebas sean lo más fieles a la realidad posibles y así asegurar que todo funciona como debe. El tiempo estimado es de 24 horas y depende de R5.
- **R7. User testing:** Tanto el equipo de desarrollo como el cliente deben intentar realizar el máximo de pruebas posibles para comprobar que el funcionamiento es correcto y se puede pasar a productivo. El tiempo estimado es de 40 horas y depende de R5 y R6.

5.4 Preparación final

- **PF1. Instalación en productivo:** Una vez todas las pruebas en la base de datos de test han tenido resultados satisfactorios, se instala el producto en la base de datos operativa, aunque solo para algunos usuarios, como podrían ser el usuario “manager” y el usuario de soporte, para asegurar que la migración ha sido correcta y todo sigue funcionando bien. El tiempo estimado es de 8 horas.

⁶ Diferencias entre el desempeño real y el desempeño esperado en una organización o negocio.

- **PF2. Elaboración de manuales y formación:** Para que los usuarios del sistema puedan utilizarlo correctamente, se elaboran manuales en los que se explican al detalle todas las funcionalidades dentro de SAP. Además, se realizan una serie de cursos formativos sobre SAP y Redmine. El tiempo estimado es de 48 horas.
- **PF3. Plan de corte:** Se elabora un informe con los pasos a seguir en la puesta en marcha. Además, se elabora una previsión con los pasos a seguir en caso de que aparezcan errores o el sistema no funcione como debería. El tiempo estimado es de 8 horas.

5.5 Entrada en productivo y soporte

- **PS1. Entrada en productivo y soporte:** El sistema pasa a estar operativo al 100% en la base de datos productiva. Siguiendo el plan de corte, durante las dos primeras semanas habrá al menos dos desarrolladores disponibles para tareas urgentes de soporte. Después se podrá solicitar soporte vía mail y un desarrollador se pondrá en contacto para solucionar los problemas. El tiempo estimado es de 80 horas.

En cuanto al tiempo dedicado a la realización de reuniones, en este caso, esta metodología da por hecho que el equipo se reúne semanalmente para que cada miembro vaya comentando los avances realizados, es por esto que no se ha creado ninguna tarea de reunión, únicamente la reunión inicial. Además, es muy común que se vayan realizando reuniones entre diferentes miembros del equipo, e incluso, con el cliente, para aclarar diferentes tipos de dudas o buscar soluciones a problemas que vayan surgiendo. Estas reuniones se suelen concretar de forma relativamente improvisada vía mail, y el tiempo estimado está incluido en el tiempo aproximado de cara tarea.

6. RECURSOS

6.1 Recursos humanos

Dentro de los recursos humanos, tendremos en cuenta tanto los diferentes empleados de Seidor que participan en el proyecto, como el personal que el cliente pone a disposición de la empresa.

Por parte de Seidor, encontramos el director del proyecto (PM), el consultor (C), y finalmente el equipo de desarrollo (D).

Por parte del cliente, encontramos los responsables del proyecto (PR), y los empleados que utilizarán el sistema (E).

6.2 Recursos tecnológicos

Para poder realizar este desarrollo será imprescindible que el cliente ponga a nuestra disposición un servidor, al que nos conectaremos a través de OpenVPN. Por lo tanto, a nivel software, necesitaremos la herramienta "Conexión a escritorio remoto" (CER) que nos

ofrece Windows, la VPN mencionada anteriormente(OVPN), y luego los diferentes programas necesarios para realizar el desarrollo, que son: Visual Studio 2019 (VS), SAP Business One (SBO), Sap Business One Studio (SBOS), Google Chrome (GC), Postman (P), Office 365 (OF) y HANA Studio (HS).

A nivel de hardware, necesitaremos lo básico, un ordenador por empleado, un monitor auxiliar, un teclado y ratón, y conexión a internet.

7.ESTIMACIONES

ID	TAREA	HORAS	DEPENDENCIAS	RECURSOS
P1	Planificación inicial y preparación del proyecto	20		PM,PR,GC,OF
P2	Identificar y garantizar los miembros del equipo	8	P1	PM,OF
P3	Reunión de kick-off	4	P2	PM,C,D,PR,OF
BBP1	Validar Alcance del proyecto	24		PM, C,PR, OF
BBP2	Estructura organizacional	32	BBP1	PM,C,PR, OF
BBP3	Definición de los requerimientos funcionales	32	BBP2	C,PR, OF
BBP4	Detalle de configuración del sistema	32	BBP3	C, OF
R1	Configuración del entorno	24		C,D,CER,OVPN, SBO,SBOS,P,VS
R2	Configuración de la herramienta Redmine	8		C,PR,GC
R3	Desarrollo de las llamadas a la API	64	R1, R2	D,CER,OVPN,VS ,P,GC
R4	Definición de entidades y conexión con el web service	24	R3	D,CER,OVPN, SBO, VS
R5	Programación de formularios en SAP	96	R4	D,CER,OVPN, SBO,SBOS,VS
R6	Cargas iniciales de datos	24		C,CER,OVPN, SBO,HS,OF
R7	User testing	40	R5, R6	C,D,PR,OVPN, SBO,P,GC
PF1	Instalación en productivo	8		D,CER,OVPN, SBO,VS
PF2	Elaboración de manuales y formación	48		D,C,OF
PF3	Plan de corte	8		D,C,OF
PS1	Entrada en productivo y soporte	80		D,CER,OVPN,VS ,OF
TOTAL		576		

Tabla 1: Tabla de resumen de las tareas

8. DIAGRAMA DE GANTT

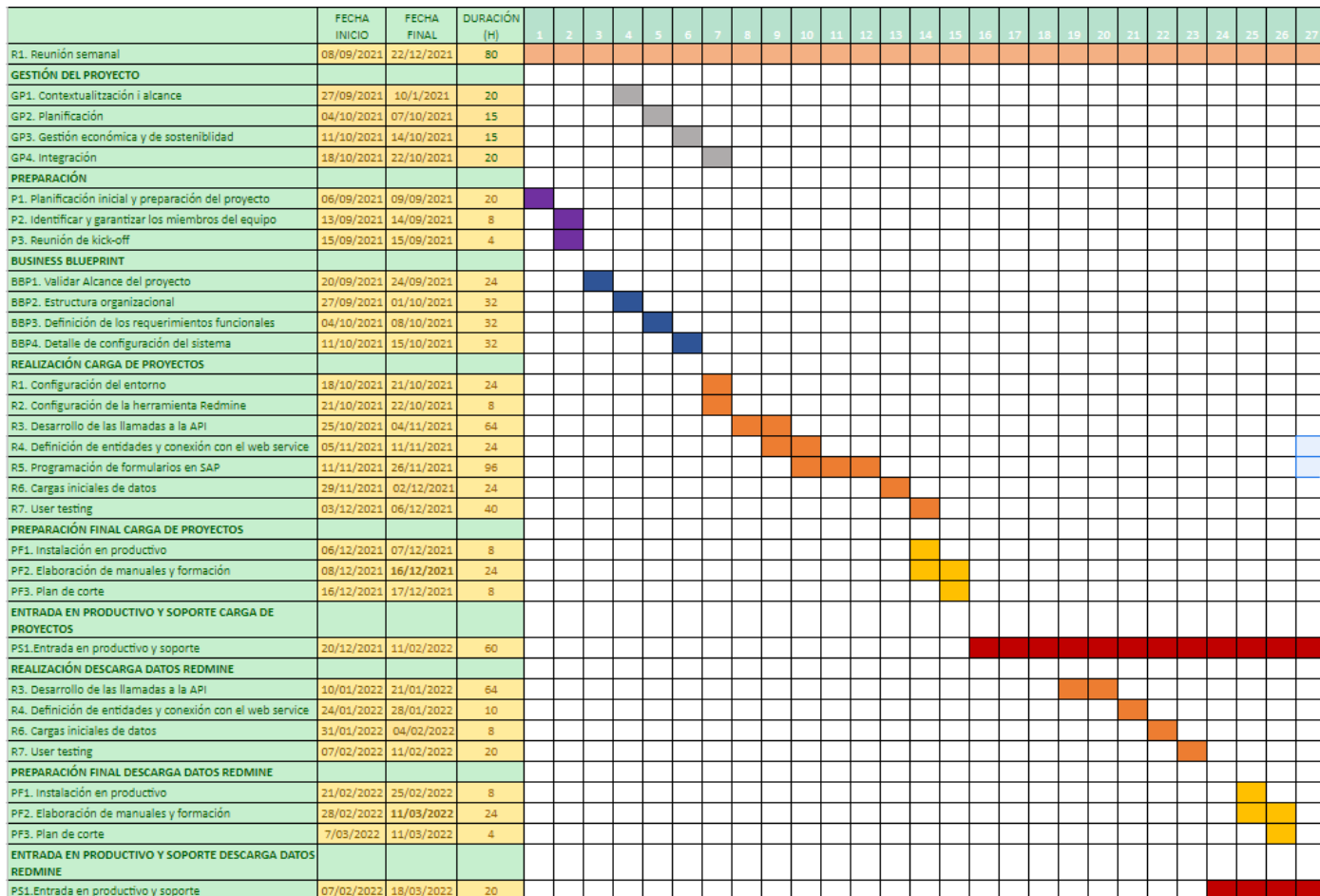


Imagen 2: Diagrama de Gantt

9. GESTIÓN DE RIESGO

Un aspecto muy importante a tener en cuenta para que el proyecto se pueda finalizar en el tiempo estipulado, es controlar los obstáculos que se puedan presentar y los errores que puedan haber a lo largo del desarrollo. Es por esto que hemos tenido en cuenta algunos de los problemas más frecuentes en este tipo de desarrollos, y la forma en la que los podemos evitar y/o solucionar.

1. Escalabilidad del sistema: El número de usuarios que puede llegar a usar este en principio no es muy elevado, pero puede crecer con el paso del tiempo y eso podría ocasionar problemas a la hora de transmitir la información. Por esto mismo, deberemos estar pendientes de que el cliente no introduzca muchos usuarios de golpe.

2. Seguridad: Se deben establecer controles de seguridad para que el usuario no pueda enviar información que pueda dañar el sistema o lo pueda corromper. Además, es importante que nadie pueda acceder a los datos privados de los proyectos de la empresa, por lo tanto deberemos dificultar al máximo el acceso a estos datos.
3. Mantenimiento o caída del servidor: Es imprescindible tener copias de seguridad de toda la información relevante del servidor, aunque la información de la base de datos y de SAP Business One en este caso no corren peligro, ya que se ejecutan de forma independiente en cada máquina, únicamente corre peligro el desarrollo realizado.
4. Problemas de desarrollo: Teniendo en cuenta que es un proyecto con unos plazos de entrega reducidos, debemos evitar a toda costa alargar los tiempos de desarrollo más de la cuenta, es por ello que en caso de que alguna funcionalidad suponga un problema para el desarrollador, deberá ponerse en contacto con otros desarrolladores e incluso con el consultor de forma prácticamente inmediata, para recibir ayuda y así poder seguir avanzando en el proyecto.
5. Problemas de protección y mantenimiento de código fuente: Se utilizará Git como sistema de gestión de versiones, perfecto para no perder cambios en el proyecto y poder realizar modificaciones en el código sin afectar directamente a la rama principal.

10.CONTROL Y SEGUIMIENTO

Este apartado recoge todos los cambios que han habido hasta el momento actual en la planificación del proyecto.

Respecto a las primeras tareas, no hubo ningún percance y se pudieron resolver en el tiempo establecido inicialmente. Estas tareas son las que se incluyen en las fases de preparación y de *Bussines BluePrint*.

Una vez iniciada la fase de Realización, se empezaron a producir algunas pequeñas modificaciones respecto a lo que estaba previsto en un inicio. Esto principalmente se debe a que, por las exigencias y las necesidades del cliente, era necesario tener disponible en fase productiva el apartado del desarrollo relacionado con la subida de los proyectos de SAP a Redmine. Por lo tanto, una vez se terminó ese apartado del desarrollo, se empezó directamente la fase de User Testing y la posterior entrada en productivo de ese apartado, con el soporte posterior que eso conlleva.

El hecho de que este apartado mencionado anteriormente pasara a productivo de una forma más temprana a lo esperado inicialmente, implicó dedicar una cierta cantidad de tiempo relativamente elevada a resolver incidencias tanto en la fase de User Testing, como en la fase de puesta en productivo. Estas incidencias se resolvieron de forma efectiva en todos los casos, pero se retrasaron mucho sus soluciones, debido a que requerían una cierta comunicación con el cliente, la cual en algunos casos no fué todo lo fluida que habíamos deseado.

Al margen de lo comentado anteriormente, también nos hemos ido encontrando con diferentes inconvenientes durante la realización de este proyecto. Estos inconvenientes han estado generalmente relacionados con problemas de conectividad tanto con el servidor del cliente en el que debíamos trabajar, como con el Servidor de Redmine al que se conecta nuestro software para poder realizar sus funcionalidades.

Estos errores han producido que la puesta en productivo del desarrollo final, se retrase bastante más de lo esperado, aunque esto no ha supuesto ningún problema para el cliente, ya que en la mayoría de las ocasiones era totalmente consciente de que el proyecto estaba sufriendo retrasos por aspectos totalmente ajenos a la empresa desarrolladora.

También cabe mencionar, que personalmente he tenido que dedicar mucho más tiempo de desarrollo del esperado para el correcto funcionamiento de las diferentes partes del proyecto. Esto es debido a que, aunque el *Software* en sí no sea extremadamente complejo, el hecho de tener que adaptarme por una parte a la forma en la que trata los datos la herramienta de *Redmine*, y posteriormente, hacer lo propio con Sap Business One, ha implicado tener que dedicar una cantidad de tiempo extremadamente elevada resolviendo pequeños errores que se iban sucediendo en el código, y que me impedían continuar con el correcto desarrollo del proyecto. Por eso mismo, podremos ver próximamente, como el tiempo total dedicado a la implementación de las llamadas a la API, se ha multiplicado por dos.

A continuación se muestran los cambios mencionados anteriormente, reflejados en las tablas utilizadas en GEP.

10.2 Estimaciones

ID	TAREA	HORAS	DEPENDENCIAS	RECURSOS
P1	Planificación inicial y preparación del proyecto	20		PM,PR,GC,OF
P2	Identificar y garantizar los miembros del equipo	8	P1	PM,OF
P3	Reunión de kick-off	4	P2	PM,C,D,PR,OF
BBP1	Validar Alcance del proyecto	24		PM, C,PR, OF
BBP2	Estructura organizacional	32	BBP1	PM,C,PR, OF
BBP3	Definición de los requerimientos funcionales	32	BBP2	C,PR, OF
BBP4	Detalle de configuración del sistema	32	BBP3	C, OF
R1	Configuración del entorno	24		C,D,CER,OVPN, SBO,SBOS,P,VS
R2	Configuración de la herramienta Redmine	8		C,PR,GC
R3	Desarrollo de las llamadas a la API	128	R1, R2	D,CER,OVPN,VS ,P,GC
R4	Definición de entidades y conexión con el web service	34	R3	D,CER,OVPN, SBO, VS
R5	Programación de formularios en SAP	96	R4	D,CER,OVPN, SBO,SBOS,VS
R6	Cargas iniciales de datos	32		C,CER,OVPN, SBO,HS,OF
R7	User testing	60	R5, R6	C,D,PR,OVPN, SBO,P,GC
PF1	Instalación en productivo	16		D,CER,OVPN, SBO,VS
PF2	Elaboración de manuales y formación	48		D,C,OF
PF3	Plan de corte	12		D,C,OF
PS1	Entrada en productivo y soporte	80		D,CER,OVPN,VS ,OF
TOTAL		670		

Tabla 2: Tabla de resumen de las tareas actualizada

10.3 Diagrama de Gantt

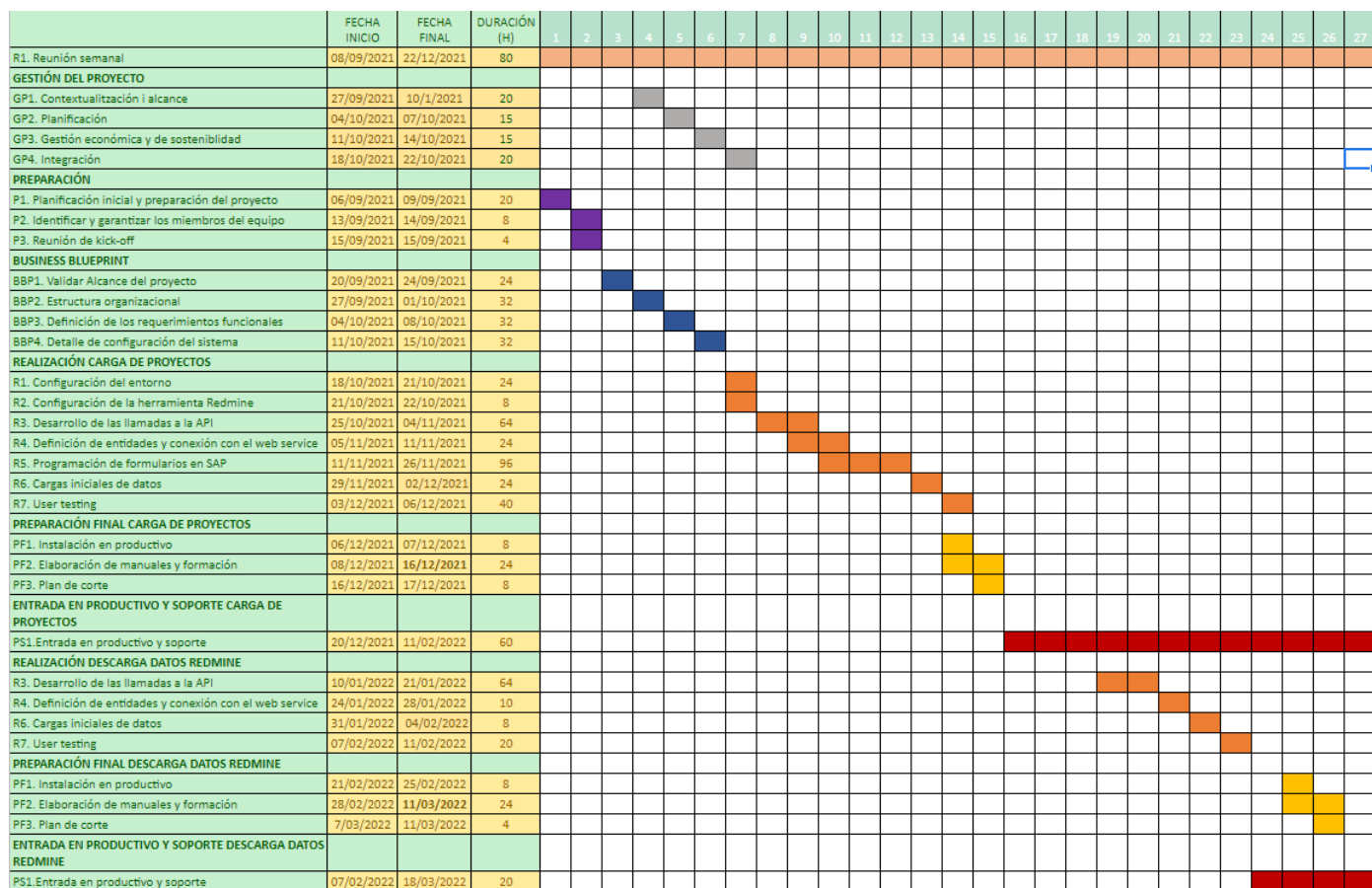


Imagen 3:Diagrama de Gantt

10.PRESUPUESTO

10.1 Identificación y estimación de costes

En el momento de identificar los costes que conlleva un proyecto, se deben detallar todos los recursos que se utilizarán, que se dividen en recursos humanos y recursos materiales. Los recursos humanos son todas aquellas personas que participan en el proyecto, desarrollando alguna de las tareas mencionadas anteriormente. En cambio, los recursos materiales, son las herramientas utilizadas para desarrollar el proyecto, en este caso software y hardware. Además, tenemos que tener en cuenta los recursos indirectos y también los costes de contingencia e imprevistos.

10.1.1 Recursos humanos

Lo primero que debemos hacer para calcular el coste que supondrán los recursos humanos, es identificar el precio/hora que cuesta cada trabajador a la empresa. En la siguiente tabla podemos ver dos precios diferentes, el bruto y el neto, el primero corresponde al precio sin impuestos, y el segundo al coste final con impuestos incluidos.

Persona	Precio/hora (€) bruto	Precio/hora (€) neto
Director del proyecto (PM)	23,1	33
Consultor (C)	17,5	25
Desarrollador (D)	10,5	15

Tabla 3: Sueldos y costes de los trabajadores del proyecto

Teniendo en cuenta los costes anteriores, procedemos a calcular el coste de cada tarea del proyecto.

Tarea	Precio por persona / tarea (€)			Total (€)
	PM	C	D	
P1	660			660
P2	264			264
P3	132	100	60	292
BBP1	396	300		696
BBP2	264	600		864
BBP3		800		800
BBP4		800		800
R1		300	180	480
R2		200		200
R3			960	960
R4			360	612
R5			1440	1440
R6		600		600
R7		400	240	640
PF1			120	120
PF2		600	360	960
PF3		100	60	160
PS1			1200	1200
TOTAL	1.716	4.800	4.980	11.828

Tabla 4: Costes humanos de cada tarea

10.1.2 Recursos materiales

En este tipo de proyectos, los costes materiales son relativamente reducidos, debido a que únicamente hay que tener en cuenta el coste de los ordenadores que utilizarán los trabajadores. Para calcular dicho coste durante el tiempo de duración del proyecto, se utiliza la siguiente fórmula:

$$\text{Amortización de un ordenador} = \frac{\text{costePC}}{\text{vida útil} * \text{días laborables} * \text{horas diarias}} * \text{horas dedicadas}.$$

Ordenador trabajador	costePC (€)	Años de vida útil	Horas diarias	Horas dedicadas	Amortización (€)
PM	900	4	8	52	6,65
C	900	4	8	192	24,54
D	900	4	8	332	42,44
TOTAL					73,63

Tabla 5: Amortizaciones ordenadores

10.1.3 Recursos indirectos

En Seidor, actualmente existe mucha flexibilidad a la hora de realizar teletrabajo, debido a la pandemia que hemos estado sufriendo durante el último año y medio. Por eso mismo podemos considerar que este proyecto se desarrolla prácticamente en su totalidad de forma telemática, y los costes a considerar son los que tiene cada persona por trabajar desde casa. En este caso, se ha considerado que esos costes son aproximadamente de unos 15€ mensuales por jornada laboral de 8h. Por lo tanto, si tenemos en cuenta el número de horas que cada persona dedica al proyecto, el resultado es:

$$\text{Coste total de recursos indirectos} = 4 \text{ meses de proyecto} * (3 \text{ personas} * 15) = 180€$$

10.1.4 Contingencia

Como hemos visto en apartados anteriores, siempre debemos estar preparados para posibles imprevistos que puedan ir surgiendo a lo largo del desarrollo del proyecto, por eso mismo, debemos calcular en nuestro presupuesto unos ciertos gastos de contingencia. Estos gastos se calculan a través de un porcentaje, y como norma general, en los proyectos informáticos suele ser de entre un 10% y un 20%. En este caso hemos considerado que un 10% será suficiente para cubrir dichos imprevistos.

Tipo	Coste (€)	Contingencia (%)	Coste final (€)
Personal	11.828	10	13.010,8
Material	73,63	10	81
Indirecto	180	10	198
TOTAL			13.289,8

Tabla 6: Costes de contingencia

10.1.5 Imprevistos

Al margen de los gastos mencionados anteriormente, también debemos tener en cuenta que, todos aquellos riesgos que en su momento definimos para este proyecto, pueden generar gastos extra. Realmente la probabilidad de que surja alguno de los problemas comentados en este proyecto es relativamente baja, pero siempre debe tenerse en cuenta. Como este tipo de incidencias las resolverá el desarrollador, calcularemos el coste a partir de su salario por hora.

Riesgo	Probabilidad (%)	Tiempo (h)	Coste (€)
R1. Escalabilidad	25	80	300
R2. Seguridad	15	32	72
R3. Servidor	10	24	36
TOTAL			408

Tabla 7: Gastos imprevistos

10.1.6 Presupuesto final

Tipo de coste	Coste (€)
Personal	11.828
Material	73,63
Indirectos	180
Total + contingencia	13.289,8
Imprevistos	408
TOTAL PROYECTO	13.697,8

Tabla 8: Presupuesto final del proyecto

10.2 Control de seguimiento

A continuación, podemos ver las diferentes tablas que han sufrido alguna variación durante el transcurso del proyecto respecto a la información expuesta anteriormente. En general, podemos decir que las únicas variaciones que ha sufrido el proyecto han estado relacionadas con el tiempo de desarrollo, por lo tanto, los cambios en el presupuesto se deberán exclusivamente al coste de dicho tiempo de desarrollo del proyecto.

10.2.1 Costes humanos

Tarea	Precio por persona / tarea (€)			Total (€)
	PM	C	D	
P1	660			660
P2	264			264
P3	132	100	60	292
BBP1	396	300		696
BBP2	264	600		864
BBP3		800		800
BBP4		800		800
R1		300	180	480
R2		200		200
R3			1920	1920
R4			510	510
R5			1440	1440
R6		600	120	720
R7		400	240	640
PF1			240	240
PF2		600	360	960
PF3		100	120	160
PS1			1200	1200
TOTAL	1.716	4.800	6.390	13.128

Tabla 9: Costes humanos de cada tarea

10.2.2 Contingencia

Tipo	Coste (€)	Contingencia (%)	Coste final (€)
Personal	13.128	10	14.440,8
Material	73,63	10	81
Indirecto	180	10	198
TOTAL			14.719,8

Tabla 10: Costes de contingencia

10.2.3 Presupuesto final

Tipo de coste	Coste (€)
Personal	13.128
Material	73,63
Indirectos	180
Total + contingencia	14.719,8
Imprevistos	408
TOTAL PROYECTO	15.127,8

Tabla 11: Presupuesto final del proyecto

10.2 Control de gestión

Para poder realizar una correcta gestión económica, es necesario controlar si se cumplen los gastos previstos. Por este motivo, se deben tener en cuenta unos ciertos mecanismos para poder realizar dicho seguimiento. A continuación mostramos las fórmulas que se utilizan para llevar a cabo este control.

- Desviación total dels costos de personal:

$$\text{coste_personal_estimado} - \text{coste_personal_real} = 11.828 - 13.128 = -1.300\text{€}$$

- Desviación total de los costes materiales:

$$\text{coste_material_estimado} - \text{coste_material_real} = 73,63 - 73,63 = 0$$

- Desviación total coste indirecto:

$$\text{coste_indirecto_estimado} - \text{coste_indirecto_real} = 180 - 180 = 0$$

- Desviación total de los imprevistos:

$$\text{coste_imprevisto_estimado} - \text{coste_imprevisto_real} = 408 - 408 = 0$$

- Desviación total de las horas:

$$(\text{horas_estimadas} - \text{horas_reales}) = 192 - 286 = -94$$

- Desviación total de los costes:

$$(\text{coste_total_estimado} - \text{coste_total_real}) = 13.697 - 15.127 = -1.430\text{€}$$

11. INFORME DE SOSTENIBILIDAD

Para realizar un buen proyecto informático, es necesario intentar descubrir el impacto social, económico y ambiental para la sociedad, que conlleva la implementación de dicho proyecto. De esta forma, podremos saber con más precisión si es necesario o no implementar la solución planteada, o si por lo contrario, se debe plantear una alternativa.

11.1 Dimensión ambiental

En este caso concreto, teniendo en cuenta que es un proyecto que no genera ningún producto físico, el impacto ambiental no es demasiado grande. Además, las condiciones actuales, han provocado que, como he comentado anteriormente, la gran mayoría del desarrollo del proyecto se realice de forma telemática, lo que supone un gran ahorro de recursos sobre todo a nivel de transporte. Dicho esto, llegamos a la conclusión de que el único impacto ambiental que tiene nuestro proyecto es el del consumo energético que supone el uso de los ordenadores, y el desgaste sobre estos que acabará haciendo que se deban renovar en un futuro.

11.2 Dimensión económica

A nivel económico, hemos podido ver en los apartados anteriores el presupuesto total del proyecto, teniendo en cuenta los costes humanos, materiales e indirectos.

Esta solución beneficia a la empresa ahorrando a los empleados una gran cantidad de tiempo a la hora de tratar las tareas que realizan, de crear los proyectos y de calcular los salarios. Por lo tanto, consideramos que a nivel económico la inversión está más que justificada ya que recortan el gasto en tareas que se podían automatizar.

11.3 Dimensión social

Por lo que hace a la dimensión social, el hecho de que, como se ha comentado anteriormente, el proyecto sirva para ahorrar a los empleados una gran cantidad de tiempo a la hora de tratar las tareas que realizan, de crear los proyectos y de calcular los salarios. Esto puede tener como resultado una mayor satisfacción por parte de los empleados, además, tenemos en cuenta de que se trata de una empresa de ingeniería, por lo tanto, como norma general estará formada por empleados que se vean motivados por la búsqueda de nuevos desafíos, y las tareas tan monótonas que hemos mencionado anteriormente está claro que no lo son.

Por último, este proyecto es una gran oportunidad para mí, ya que me permite adquirir responsabilidades y conocimientos dentro de la empresa que, de otra manera habría tardado mucho más tiempo en conseguir.

12. LEYES Y REGULACIÓN

Al tratarse de un TFG en modalidad “B”, es decir, realizado en una empresa. Debemos cumplir las normas de la empresa en relación a la seguridad de la información. Dichas normas, se gestiona en el registro “RE27.02_NORMATIVA_SEGURIDAD_INFORMACION”, en el que se referencian entre otras las siguientes normativas y leyes:

- Reglamento (UE) 2016/679 del Parlamento Europeo y del Consejo de 27 de abril de 2016 relativo a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos (RGPD).
- Ley Orgánica de Protección de Datos y garantías de los derechos digitales Ley 03/2018 (LOPDGDD).
- Prevención de Riesgos Laborales Ley 31/1995 de 8 de noviembre y Real Decreto 39/1997 de 17 de enero, por el que se aprueba el Reglamento de los Servicios de Prevención.
- Ley 34/2002, de 11 de julio, de Servicios de la Sociedad de la Información y Comercio Electrónico (LSSI-CE).
- Real Decreto 3/2010, de 8 de enero, por el que se regula el Esquema Nacional de Seguridad en el ámbito de la administración electrónica. BOE de 29 de enero de 2010.
- EL CÓDIGO PENAL. LEY ORGÁNICA 10/1995, DE 23 DE NOVIEMBRE, del Código Penal.

Es importante destacar que, estas leyes se deben cumplir tanto para la información confidencial de la empresa desarrolladora, como para dicha información de la empresa cliente.

13. ESPECIFICACIÓN DEL SISTEMA

El objetivo de esta etapa de especificación, es explicar de forma completa cómo se comportará nuestro sistema. Dentro de esta etapa se definirán los casos de uso con sus correspondientes especificaciones y los requerimientos no funcionales.

13.1 Diagrama de casos de uso

El diagrama de casos de uso, cómo veremos a continuación, en este caso es extremadamente sencillo. Eso es debido a que, como hemos ido mencionando a lo largo del documento, nuestro sistema tiene la función de enviar y recibir datos a Redmine, por lo tanto en él solo encontraremos estas dos funcionalidades, aunque es cierto que para que esto se pueda realizar, se ha tenido que crear un desarrollo en SAP, el cual tiene otras funcionalidades que veremos a continuación.

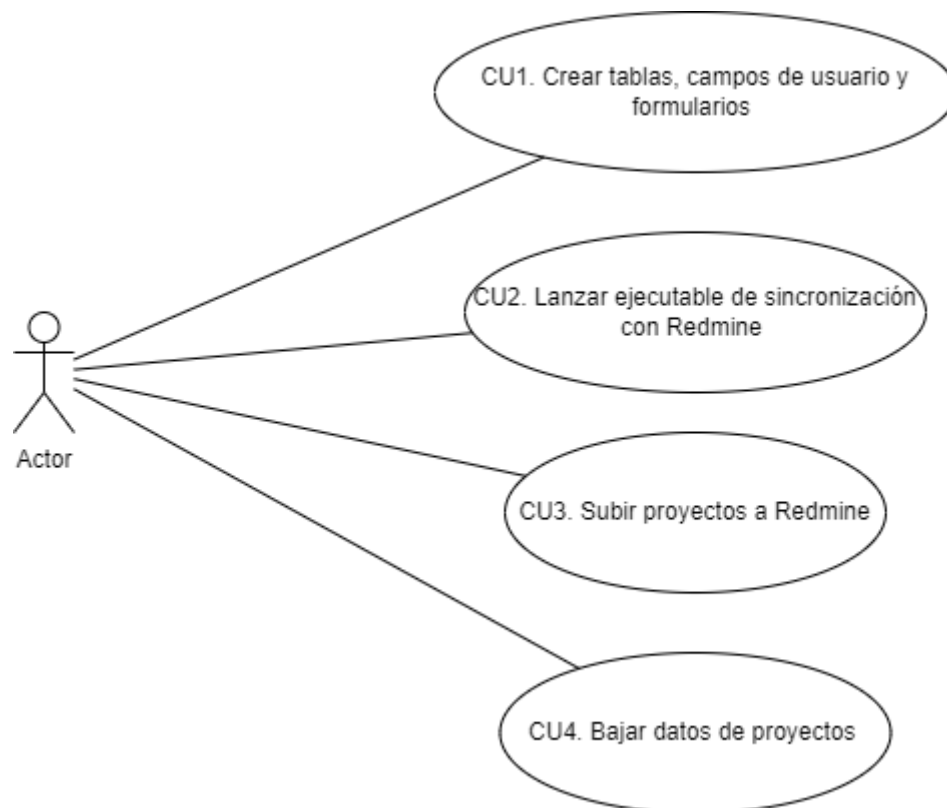


Tabla 7: Costes humanos de cada tarea

13.2 Especificación de casos de uso

Caso de uso	CU1. Crear tablas, campos de usuario y formularios
Actores	Usuario
Precondición	El usuario ha iniciado sesión en SAP Business One con permisos de "manager"
Disparador	El usuario despliega el menú de "Gestión -> Inicialización del sistema", y pulsa el botón de "Crear tablas, campos de usuario y formularios."
Escenario de éxito	
<ol style="list-style-type: none"> 1. El sistema muestra un aviso para que el usuario confirme la acción. 2. El sistema realiza la creación de los datos especificados. 3. El sistema muestra por pantalla el éxito de la creación. 	
Extensiones	

Tabla 12: Caso de uso número 1

Caso de uso	CU2. Lanzar ejecutable de sincronización con <i>Redmine</i>
Actores	Usuario
Precondición	El usuario ha iniciado sesión en SAP Business One, posteriormente, ha accedido al menú de “Gestión -> Inicialización del sistema”, ha pulsado el botón de “Parametrizaciones Add-on” y ha especificado la ubicación del ejecutable en el campo “Nombre Fichero”.
Disparador	El usuario despliega el menú de “Gestión” y pulsa el botón de “Iniciar integración Redmine.”
Escenario de éxito	
1. El sistema lanza el ejecutable, y este se despliega en pantalla.	
Extensiones	
El ejecutable no se encuentra en la ubicación indicada, por lo tanto, este no se lanzará.	

Tabla 13: Caso de uso número 2

Caso de uso	CU3. Subir proyectos a Redmine
Actores	Usuario
Precondición	El usuario ha iniciado sesión en SAP Business One, posteriormente, ha accedido al menú de “Gestión -> Inicialización del sistema”, ha pulsado el botón de “Parametrizaciones Add-on” y ha especificado los datos de conexión de Redmine. Posteriormente, el usuario ha iniciado la aplicación y se ha realizado correctamente la conexión.
Disparador	El usuario selecciona la opción de “Subir proyectos a redmine” en la aplicación.
Escenario de éxito	
<ol style="list-style-type: none"> 1. El sistema muestra que se ha conectado correctamente a SAP. 2. El sistema realiza la subida de datos a Redmine. 3. El sistema muestra por pantalla el éxito de la subida de los datos. 	
Extensiones	
<ol style="list-style-type: none"> 1. El sistema no se ha podido conectar correctamente a SAP. <ol style="list-style-type: none"> 1.a) El sistema muestra al usuario que no se ha podido conectar correctamente. 2. El sistema no se puede conectar correctamente con Redmine. <ol style="list-style-type: none"> 2.a) El usuario debe revisar los datos de conexión introducidos previamente en SAP Business One. 2.b) El caso de uso retorna al disparador. 	

Tabla 14: Caso de uso número 3

Caso de uso	CU4. Bajar datos de Redmine
Actores	Usuario
Precondición	El usuario ha iniciado sesión en SAP Business One, posteriormente, ha accedido al menú de “Gestión -> Inicialización del sistema”, ha pulsado el botón de “Parametrizaciones Add-on” y ha especificado los datos de conexión de Redmine. Posteriormente, el usuario ha iniciado la aplicación y se ha realizado correctamente la conexión.
Disparador	El usuario selecciona la opción de “Bajar datos de Redmine” en la aplicación.
Escenario de éxito	
<ol style="list-style-type: none"> 1. El sistema muestra que se ha conectado correctamente a SAP. 2. El sistema realiza la carga de datos en SAP. 3. El sistema muestra por pantalla el éxito de la carga de datos. 	
Extensiones	
<ol style="list-style-type: none"> 1. El sistema no se ha podido conectar correctamente a SAP. <ol style="list-style-type: none"> 1.a) El sistema muestra al usuario que no se ha podido conectar correctamente. 2. El sistema no se puede conectar correctamente con Redmine. <ol style="list-style-type: none"> 2.a) El usuario debe revisar los datos de conexión introducidos previamente en SAP Business One. 2.b) El caso de uso retorna al disparador. 	

Tabla 15: Caso de uso número 4

13.3 Requisitos no funcionales [8]

Req. no funcional 1	Apariencia
Tipo	10A
Descripción	El sistema debe tener un diseño acorde con las preferencias del cliente.
Justificación	En este caso, al ser un sistema contratado por un cliente, es fundamental que quede satisfecho en todos los aspectos, incluida la apariencia.
Condición de satisfacción	El cliente debe dar su aprobación sobre la apariencia del sistema.

Tabla 16: Requisito no funcional número 1

Req. no funcional 2	Facilidad de uso
Tipo	11a
Descripción	El sistema será fácil de utilizar para cualquier tipo de usuario, intuitivo y se evitará cualquier error en su uso.
Justificación	Este sistema puede ser utilizado por cualquiera de los empleados del cliente, por lo tanto, es necesario que sea muy intuitivo y sencillo de utilizar.
Condición de satisfacción	Se presentará el sistema ante el cliente, y este debe estar de acuerdo con su diseño y utilidad.

Tabla 17: Requisito no funcional número 2

Req. no funcional 3	Aprendizaje
Tipo	11c
Descripción	El sistema tiene un aprendizaje muy sencillo, además, se proporcionará un tutorial completo por escrito para que no haya ningún tipo de dudas de cómo se utiliza.
Justificación	Este sistema está diseñado para facilitar el trabajo a los empleados de la empresa que ha contratado nuestro servicio, por lo tanto, debe tener un aprendizaje fácil y cómodo, para que se pueda utilizar desde el momento de la instalación.
Condición de satisfacción	Durante la presentación del sistema y la etapa de testing, el cliente nos irá comentando todo tipo de dudas que le surjan alrededor del sistema. Con este tipo de dudas podremos evaluar hasta qué punto hemos conseguido cumplir con este requisito.

Tabla 18: Requisito no funcional número 3

Req. no funcional 4	Comprensión
---------------------	-------------

Tipo	11d
Descripción	Los usuarios al ejecutar el sistema, sabrán perfectamente cuál es la función de cada botón y que es lo que deben hacer.
Justificación	Este sistema tiene unas funcionalidades básicas de entender y cada botón las describe perfectamente.
Condición de satisfacción	Los símbolos y palabras utilizadas en el sistema se entenderán a la perfección.

Tabla 19: Requisito no funcional número 4

Req. no funcional 5	Velocidad y latencia
Tipo	12a
Descripción	El sistema se debe ejecutar con una velocidad acorde a los datos a tratar, y siempre mostrará al usuario que está pasando.
Justificación	El sistema tratará los datos con gran velocidad, además, no estará más de 3 o 5 segundos sin mostrar al usuario que está sucediendo.
Condición de satisfacción	El usuario debe ser consciente en todo momento de que el sistema está funcionando correctamente, y este no se debe demorar más de la cuenta en tratar los datos.

Tabla 20: Requisito no funcional número 5

Req. no funcional 6	Precisión y exactitud
---------------------	-----------------------

Tipo	12c
Descripción	Los datos tanto exportados como importados deben ser exactos y precisos.
Justificación	Para poder realizar las facturaciones de forma correcta, es necesario que se especifiquen con precisión las horas y minutos dedicados a cada tarea, además de las fechas.
Condición de satisfacción	En todos los campos de tiempo se mostrarán tanto las horas como los minutos, y en los campos de fecha se mostrará la fecha completa.

Tabla 21: Requisito no funcional número 6

Req. no funcional 7	Fiabilidad y disponibilidad
Tipo	12d
Descripción	El sistema debe estar operativo en cualquier momento.
Justificación	Los usuarios pueden necesitar utilizar las funcionalidades del sistema en cualquier momento.
Condición de satisfacción	El sistema debe estar siempre disponible salvo que haya un error en Sap Business One o en Redmine.

Tabla 22: Requisito no funcional número 7

Req. no funcional 8	Escalabilidad
Tipo	12g
Descripción	El sistema debe de ser escalable.
Justificación	Es posible que el cliente incorpore nuevos usuarios a su sistema, que se genere una gran cantidad de nuevos proyectos o incluso que se quieran añadir nuevas funcionalidades.
Condición de satisfacción	Se intentará hacer el software lo más dinámico posible para soportar futuros cambios de manera sencilla.

Tabla 23: Requisito no funcional número 8

Req. no funcional 9	1
---------------------	---

Tipo	15a
Descripción	Solo personal autorizado tendrá acceso a a los diferentes menús que ofrece el sistema dentro de <i>SAP Business One</i> .
Justificación	Para asegurar la privacidad de los datos, y el correcto tratamiento de estos, el cliente decidirá qué usuarios pueden acceder a los diferentes menús del sistema.
Condición de satisfacción	Se añadirá en <i>SAP Business One</i> una condición para cada menú, que indicará qué usuarios pueden verlo y ejecutarlo.

Tabla 22: Requisito no funcional número 9

Req. no funcional 10	Privacidad
Tipo	15c
Descripción	El sistema debe asegurar la privacidad de los datos proporcionados por las empresas.
Justificación	Las respuestas, los resultados y cualquier otro dato que las empresas nos proporcionen deben de ser accesibles solo para personas autorizadas.
Condición de satisfacción	Se comprobará que este tipo de datos son accesibles solo para personas autorizadas.

Tabla 24: Requisito no funcional número 10

14. ARQUITECTURA DEL SISTEMA

A nivel de arquitectura del sistema, debemos tener en cuenta que en este proyecto se trabaja con tres sistemas diferentes, y cada uno de ellos tendrá su arquitectura propia. Por este motivo, inicialmente haremos un repaso por las arquitecturas que ya están previamente diseñadas, para poder entender plenamente su funcionamiento y de esta manera encontrar la mejor solución para el sistema que estamos diseñando.

Pero antes de centrarnos en cada arquitectura por separado, vamos a ver una imagen que nos resume cómo es la interacción entre el usuario y los diferentes sistemas.

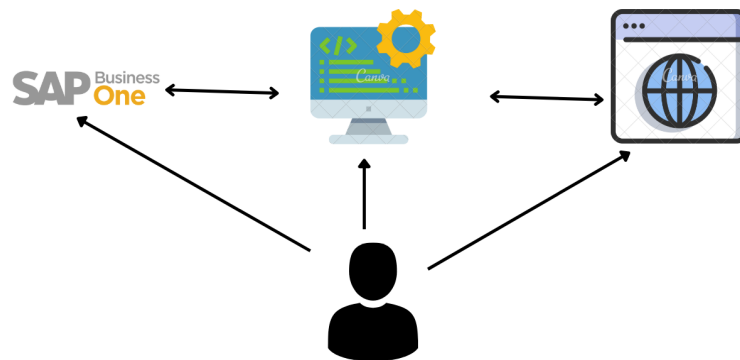


Imagen 4: Resumen del sistema

Cómo podemos observar, el usuario se puede llegar a comunicar con los tres sistemas, en primer lugar, interactúa con Sap Business One, dónde puede crear proyectos, consultar todo tipo de información, puede configurar todas las parametrizaciones necesarias para la integración y puede iniciar dicha integración. En este punto, el usuario interactúa con el sistema de enlace para decidir qué acción desea realizar. Además, el usuario interactúa con *Redmine* para añadir registros de tiempo, actualizar datos sobre proyectos, crear plantillas etc.

Respecto a la interacción entre sistemas, en la imagen podemos ver cómo el sistema de enlace se comunica con los dos sistemas restantes tanto para enviar cómo para recibir información.

14.1 Arquitectura de SAP Business One

Este sistema está basado en la arquitectura cliente-servidor de dos niveles [7]. Esto implica que tenemos una interfaz de usuario instalada y ejecutándose en un ordenador y envía

solicitudes a un servidor para ejecutar operaciones complejas. En esta arquitectura encontramos dos niveles diferenciados:

- Cliente (fat client): En este caso el cliente tiene inmersa toda la lógica de negocio dentro de la aplicación que trabaja como interfaz de usuario. El cliente es el encargado de iniciar todas las solicitudes o peticiones, espera y recibe las respuestas del servidor y se comunica directamente con el usuario a través de la interfaz gráfica de usuario mencionada anteriormente.
- Servidor: Este nivel funciona como almacenamiento de datos. Aquí se almacena toda la información del sistema. El servidor espera recibir solicitudes por parte del cliente, para procesarlas y posteriormente enviar una respuesta. [6]

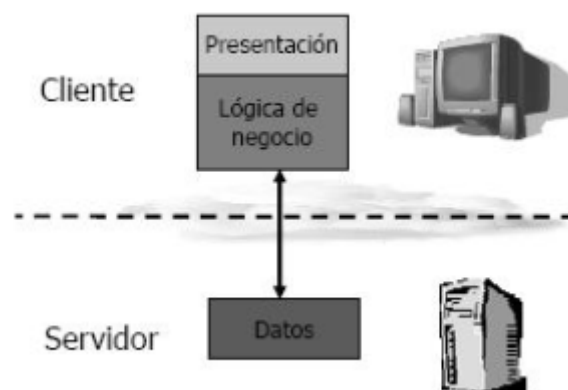


Imagen 5: Arquitectura cliente servidor

El hecho de que esta arquitectura sea de dos niveles, permite al servidor responder todas las solicitudes y recursos del cliente.

Este modelo en concreto tiene diferentes ventajas y desventajas:

- Ventajas:
 - Control centralizado
 - Escalabilidad
 - Estructura de desarrollo simple
- Desventajas:
 - Alta carga de red
 - Funcionalidad limitada

14.1.1 Base de datos

En este apartado, el objetivo es entender cuáles son las tablas de la base de datos de SAP Business One con las que vamos a trabajar en nuestro sistema, que campos vamos a utilizar de estas tablas y cómo se relacionan entre ellas en caso de que haya algún tipo de relación.

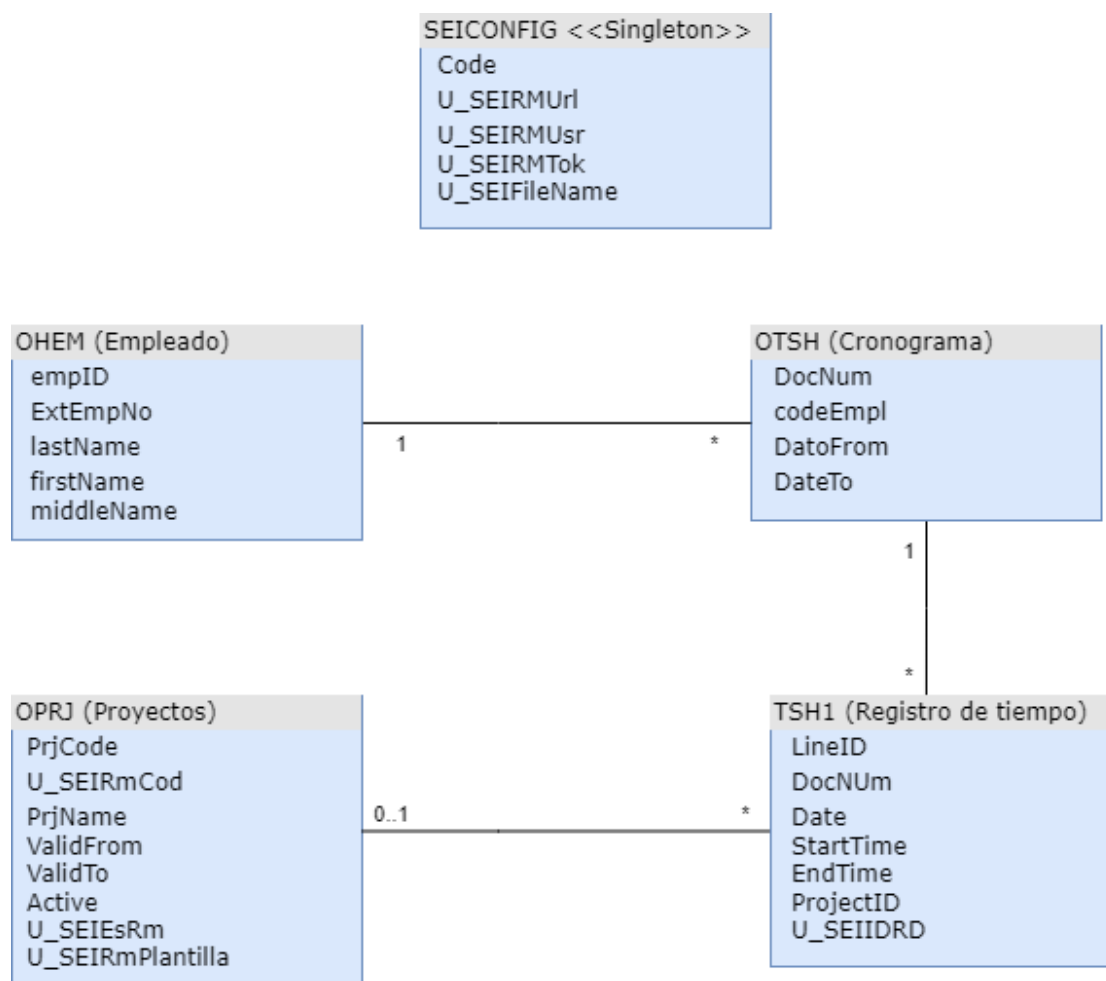


Imagen 6: Tablas de la base de datos

En la imagen anterior podemos observar cómo trabajaremos con cinco tablas diferentes, una de ellas, en concreto la tabla “SEICONFIG”, la crea nuestro *Addon* a través de su funcionalidad “Crear tablas, campos de usuario y formularios”, y se utiliza para almacenar la información necesaria para la conexión con Redmine y la ejecución de nuestro sistema de enlace. Por otro lado, tenemos otras cuatro tablas, que son propias de SAP Business One, y que utilizaremos para recoger o almacenar la información correspondiente en cada caso.

Vamos a proceder a explicar cuál es la utilidad de cada una de estas tablas, y explicaremos qué campos se han creado en ellas y porque.

- OHEM: Esta tabla contiene los datos de los empleados registrados en el sistema. Es necesario interactuar con esta tabla puesto que los registros de horas van asignados a un empleado. En este caso, no ha sido necesario crear ningún campo nuevo en esta tabla, puesto que en vez de ello, hemos utilizado el campo “ExtEmpNo” para introducir el identificador que tiene dicho empleado en Redmine, esto es estrictamente necesario para el correcto funcionamiento de la aplicación, pues que si no tenemos esta información, no se puede establecer una relación entre la información del empleado que nos devuelve *Redmine* y la información que tenemos en Sap Business One.

- OTSH: Esta se utiliza para almacenar la información sobre los cronogramas en Sap Business One, que es el formulario en el que se guardan los registros de tiempo. En este caso, se utiliza un cronograma diferente para cada Empleado, y se establece una fecha de inicio pero no una fecha de finalización, ya que este cronograma almacenará todos los registros de tiempo del usuario independientemente de la fecha en la que se produzcan.
- TSH1: Es la tabla que utiliza SAP Business One para almacenar cada registro de tiempo que contiene un cronograma. En esta tabla se ha creado el campo U_SEIIDRD, que corresponde al identificador que le asigna Redmine a este registro de tiempo. Es un campo necesario para identificar si un registro de tiempo ya se ha asignado y así evitar errores y repeticiones.
- OPRJ: En esta tabla se almacena toda la información acerca de los proyectos en SAP Business One. En esta tabla se han creado tres campos para poder realizar la integración con Redmine, primero tenemos el campo U_SEIRmCod, que corresponde el identificador que le asigna *Redmine* a este proyecto, posteriormente tenemos el campo U_SEIEsRm, un booleano que nos indica si este proyecto se debe subir a Redmine, y por último, U_SEIRmPlantilla, que contiene la plantilla que hay que asignar a este proyecto en *Redmine*.

Cabe destacar que, en ningún caso se muestran todos los campos que contiene cada tabla, ya que todas ellas tienen una gran cantidad de campos que no son útiles para el enlace con Redmine, y por lo tanto, mostrarlos únicamente afectaría a la comprensión del diagrama.

14.2 Arquitectura de Redmine

En nuestra solución, la parte que utilizaremos de Redmine es su API, por lo tanto, en este apartado nos vamos a centrar en la arquitectura de la API.

La API de Redmine utiliza una arquitectura REST o RESTful, que es un estilo de arquitectura a la hora de realizar una comunicación entre cliente y servidor, la cual corresponde a cualquier interfaz entre sistemas que use *HTTP* para obtener datos o generar operaciones sobre esos datos en todos los formatos posibles, como *XML* y *JSON*. Esta arquitectura es muy utilizada debido a que ofrece una solución más sencilla para manipular los datos respecto a otras como SOAP(Simple Object Access Protocol), que tiene más capacidad pero es más compleja.

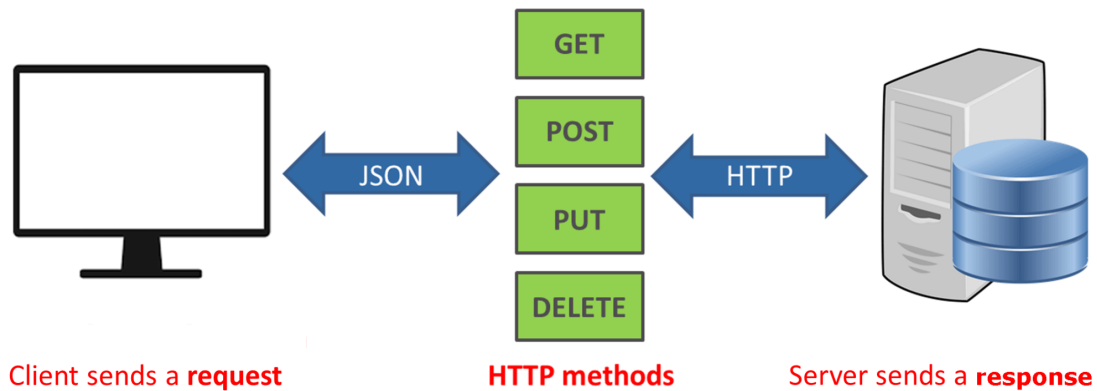


Imagen 7: Resumen arquitectura REST

Para que se considere que una API utiliza la arquitectura REST, debe cumplir con una serie de características:

- Protocolo cliente/servidor sin estado
- Arquitectura cliente-servidor compuesta de clientes, servidores y recursos con la gestión de solicitudes a través de *HTTP*.
- Interfaz uniforme entre los elementos.
- Utiliza hipertextos como pueden ser *HTML*, *XML* o *JSON* para representar la información.
- Utiliza *URIs* siguiendo una sintaxis universal.

Las ventajas que ofrece esta arquitectura son:

- Separación entre cliente y servidor.
- Visibilidad, fiabilidad y escalabilidad.
- Es independiente del tipo de plataformas y lenguajes.

14.3 Arquitectura del sistema de enlace

Una vez hemos podido entender de forma algo superficial como funcionan los dos sistemas que pretendemos enlazar, vamos a proceder a explicar en profundidad la arquitectura completa de este enlace.

Antes de nada debemos entender que este sistema está separado en dos proyectos, el *Add-on* y el ejecutable. En cuanto al *Add-on*, nos encontramos con una estructura ya definida por la empresa desarrolladora, por lo tanto, no nos vamos a centrar en dicha arquitectura ya que en este proyecto simplemente se han modificado algunas de sus clases para poder crear los campos y formularios necesarios.

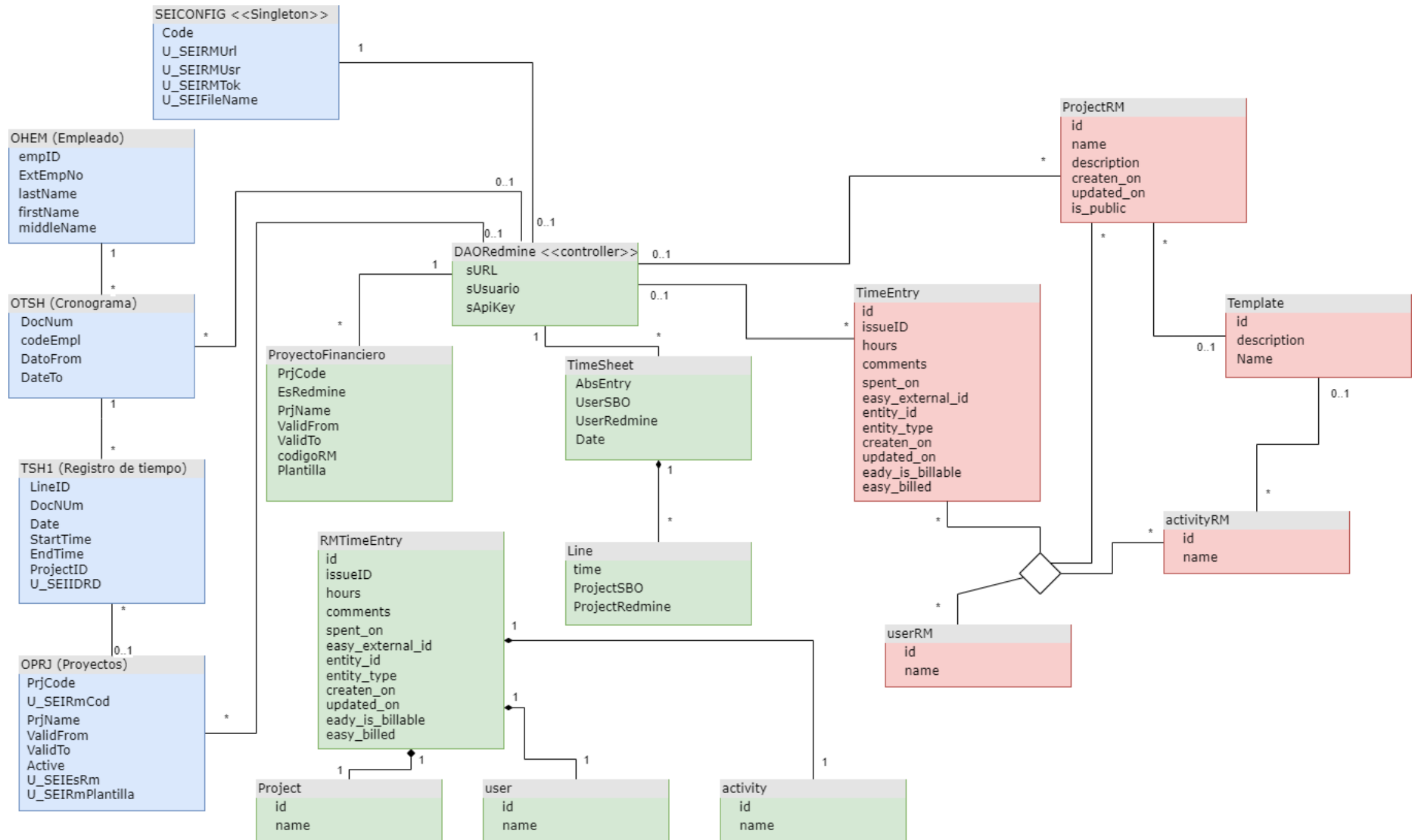
En este caso, nos vamos a centrar en la arquitectura del proyecto que se encarga de realizar el enlace. Para ello lo primero que vamos a ver es un diagrama UML que incluye todas las clases que se van a utilizar, tanto de SAP Business One, como de Redmine y del nuestro sistema de enlace. En este caso, para diferenciar los sistemas, veremos que cada

Sistema de enlace entre SAP Business One y una herramienta de gestión de proyectos

clase tiene el fondo de un color, el azul está destinado a las clases que corresponden a SAP Business One, el rojo a las clases de *Redmine* y el verde a las clases correspondientes al sistema de enlace.

Quiero destacar que en este diagrama tan solo se muestran las clases y los campos que se utilizan para realizar el enlace.

14.3.1 Diagrama UML



En esta imagen se muestra la separación de los 3 sistemas. En este caso, para realizar la comunicación entre sistemas se ha optado por utilizar un controlador, que es el encargado de realizar las llamadas necesarias, recolectar la información recibida, adaptarla mediante las clases que se muestran en color verde, y a partir de ahí, enviar la información en el formato correspondiente al sistema que la demanda y realizar con ella las operaciones necesarias para que se almacene de forma correcta.

14.3.2 Restricciones textuales

- Claves externas: (SEICONFIG, Code), (OHEM, empID), (OTSH,DocNum), (TSH1, {LineID, DocNum}), (OPRJ, PrjCode), (ProyectoFinanciero, PrjCode), (TimeSheet, AbsEntry), (RMTimeEntry, id), (Project, id), (TimeEntry, id), (ProjectRM, id), (Template, id), (userRM, id), (activityRM, id)
- Para que exista un TimeEntry en Redmine, es necesario que exista una relación entre un ProjectRM, una activityRM y un userRM.
- Una activityRM siempre debe estar asociada a un ProjectRM.

14.3.3 Patrones implicados

En este caso, el único patrón que se utiliza es el patrón Singleton, un patrón de diseño utilizado para restringir las instancias que se pueden crear de un objeto, mientras que proporciona un punto de acceso global a esta instancia.

Este patrón se utiliza para la clase "SEICONFIG", puesto que solo se puede tener una instancia, y nos interesa que se pueda acceder de forma global a dicha instancia, para así siempre poder disponer de los datos necesarios para la integración.

14.4 Diagramas de secuencia

14.4.1 Subir proyectos

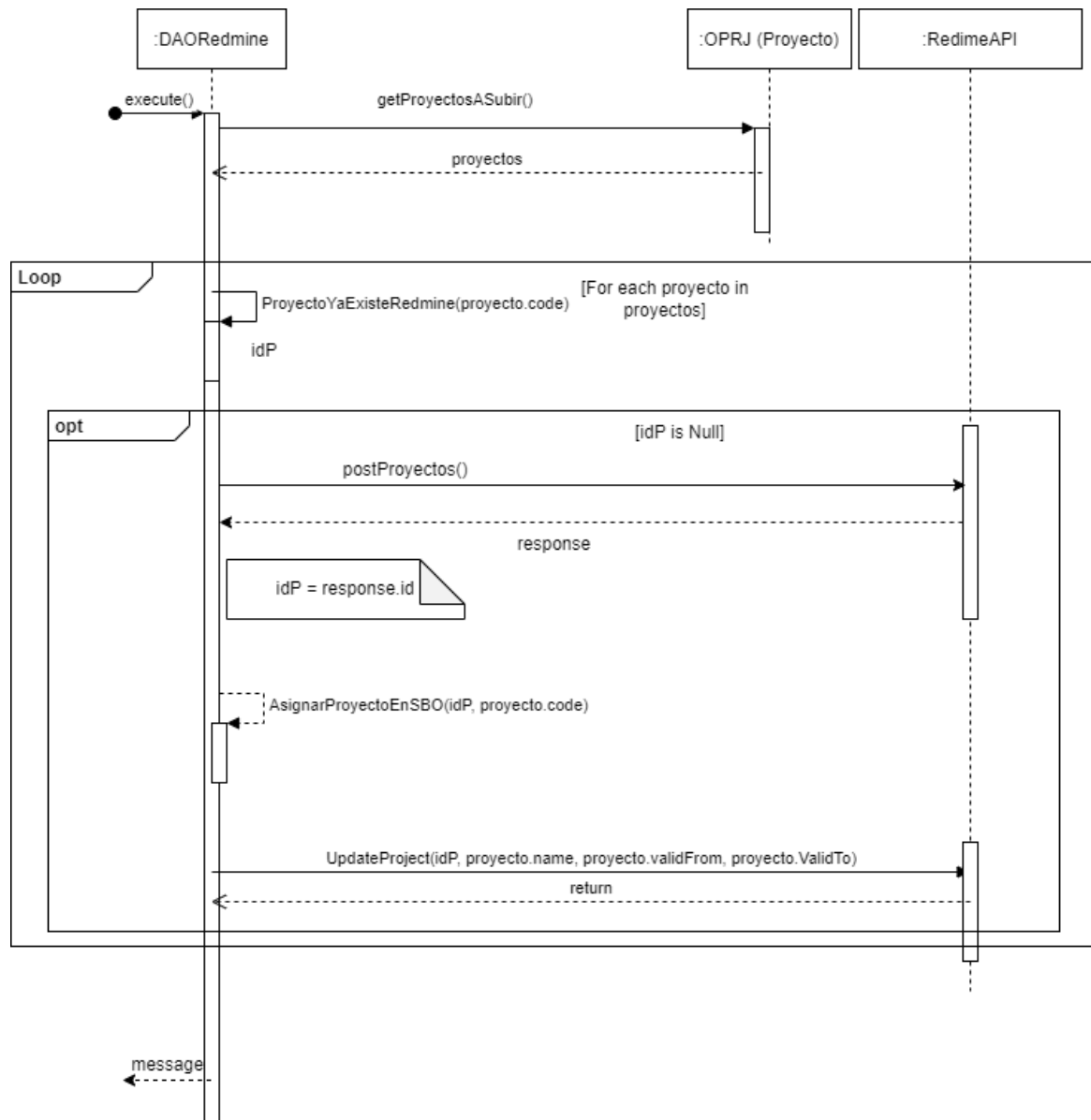


Imagen 8: Diagrama de secuencia de la operación subir proyectos

En esta imagen, podemos ver el diagrama de secuencia para la operación de subir proyectos a Redmine.

Como podemos observar en la imagen, el primer paso es realizar una consulta a la base de datos de SBO, en concreto a la tabla OPRJ que hace referencia a los proyectos, para de esta forma conseguir toda la información necesaria sobre los proyectos que se desean subir a Redmine.

A continuación, se realiza una iteración sobre todos estos proyectos, revisando para cada iteración, si ese proyecto ya está publicado en Redmine, y por lo tanto no se debe volver a subir.

En caso de que la anterior operación nos haya retornado un valor nulo, significa que ese proyecto no está publicado en Redmine, por lo tanto, se cumplirá la condición y se realizará la subida del proyecto a Redmine.

Una vez el proyecto está publicado en Redmine, utilizaremos la información proporcionada por la respuesta de la API de Redmine, para actualizar la tabla de proyectos en SBO, y así poder tener una relación entre el proyecto en SBO y en Redmine.

Por último, se hace una actualización del proyecto publicado en Redmine, para así poder asignar la plantilla indicada en SBO, y de esta manera el proyecto tendrá la estructura que desea el cliente.

14.4.2 Bajar datos de proyectos

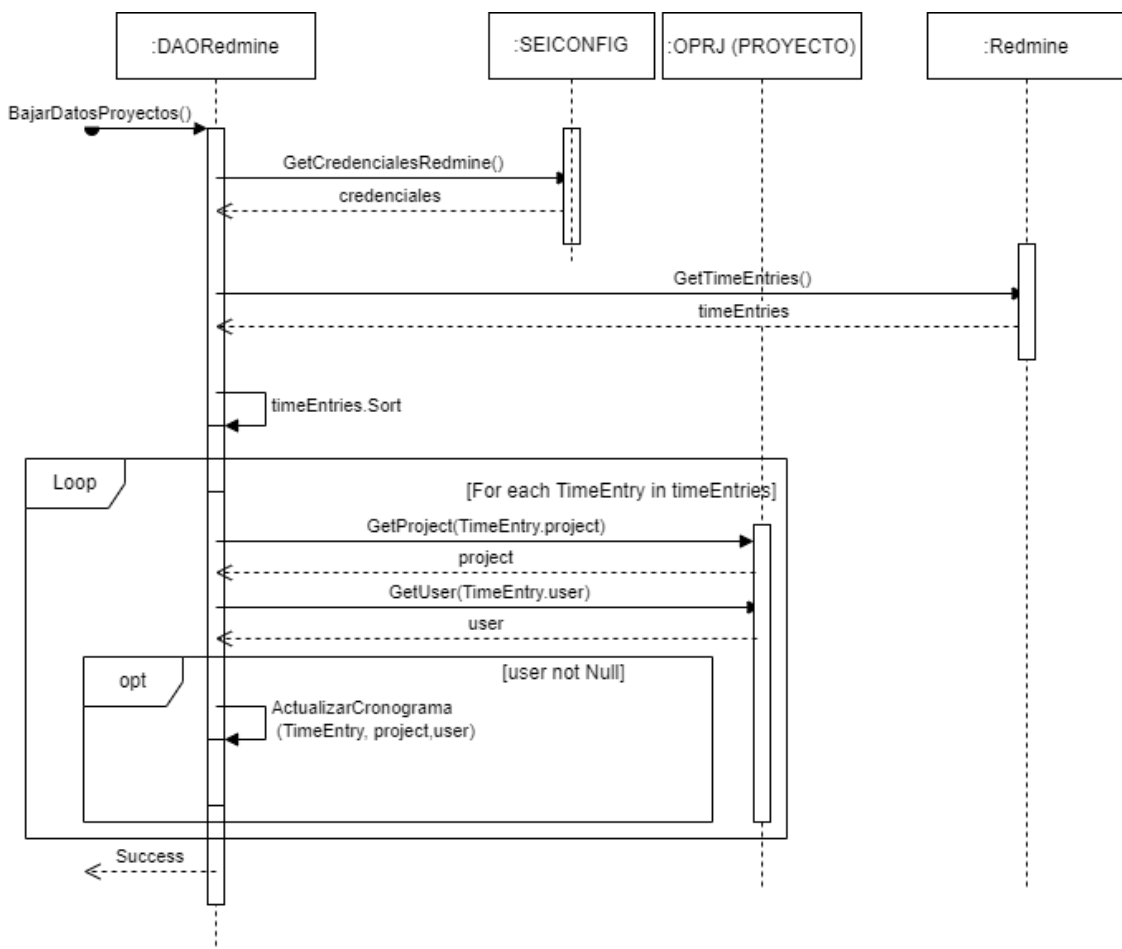


Imagen 9: Diagrama de secuencia de la operación bajar datos de proyectos

Esta imagen nos muestra el diagrama de secuencia de la funcionalidad que baja los datos de los proyectos desde *Redmine* a SAP Business One.

El primer paso, es acceder a la tabla “SEICONFIG” que contiene la información de acceso a la API de *Redmine*.

Una vez tenemos dicha información, procedemos a realizar una llamada a la API para conseguir todos los registros de tiempo, posteriormente los ordenamos por fecha.

Realizamos un bucle, para cada registro de tiempo, *Redmine* nos devuelve los identificadores de usuario y proyecto que tiene en su sistema pero estos identificadores no són los mismos que se utilizan en SAP Business One, por lo tanto realizamos las consultas necesarias para conseguir los identificadores que necesitamos.

Una vez tenemos estos identificadores, y en el caso que el usuario exista en SAP Business One, es decir, que su identificador no sea nulo, procedemos a actualizar los cronogramas.

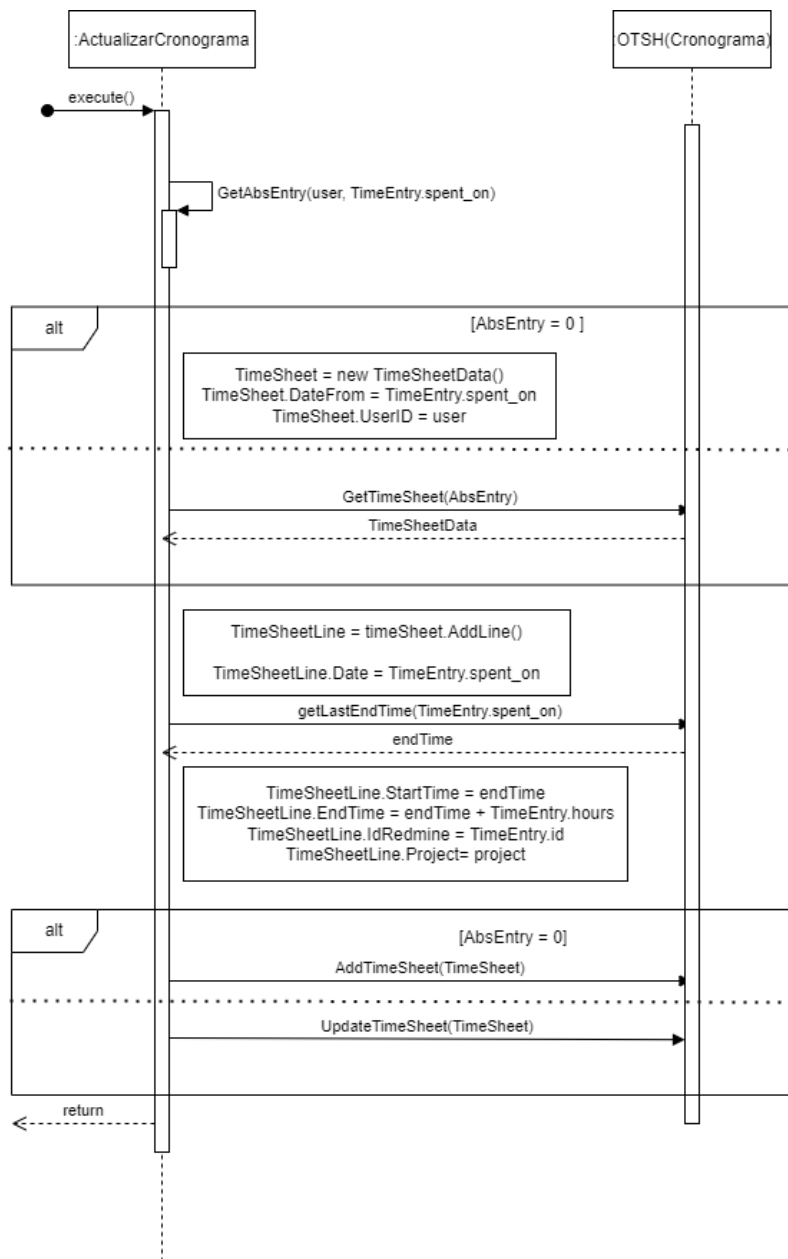


Imagen 10: Diagrama de secuencia de la función Actualizar Cronograma

Para actualizar los cronogramas, lo primero que hacemos es consultar si en SAP Business One ya existe un cronograma para ese usuario y cuyo intervalo de tiempo incluya la fecha del registro de tiempo que deseamos añadir.

Si no existe, lo creamos y le asignamos el usuario y la fecha. Si por el contrario ya existe, lo rescatamos de la tabla de cronogramas.

Una vez hecho esto, es momento de añadir la línea que corresponderá al registro de tiempo. Para ello debemos asignarle a esta línea los valores correspondientes, y posteriormente, añadiremos o actualizaremos el cronograma dependiendo de si ya existía en SAP Business One o no.

15. EJEMPLO DE USO REAL

Para poder entender mejor cómo funciona el sistema, vamos a ver un ejemplo del que podría ser un uso completo de dicho sistema.

En este caso, primero mostraremos los pasos que hay que realizar en SAP Business One, para configurar los diferentes campos y formularios que necesita la integración para poder funcionar, este paso solo se tiene que realizar una vez cuándo se instala el Add-on en SAP Business One.

Posteriormente, mostraremos el que podría ser un uso real por uno de los trabajadores del cliente.

15.1 Configuración del sistema

El primer paso para configurar el sistema, es la creación de los campos, las tablas y los formularios que se utilizan en esta integración. Para ello debemos asistir al menú y buscar el punto de menú que se encuentra en “Gestión -> Inicialización sistema -> Crear Tablas, Campos y UDO de Usuario REDMINE”.

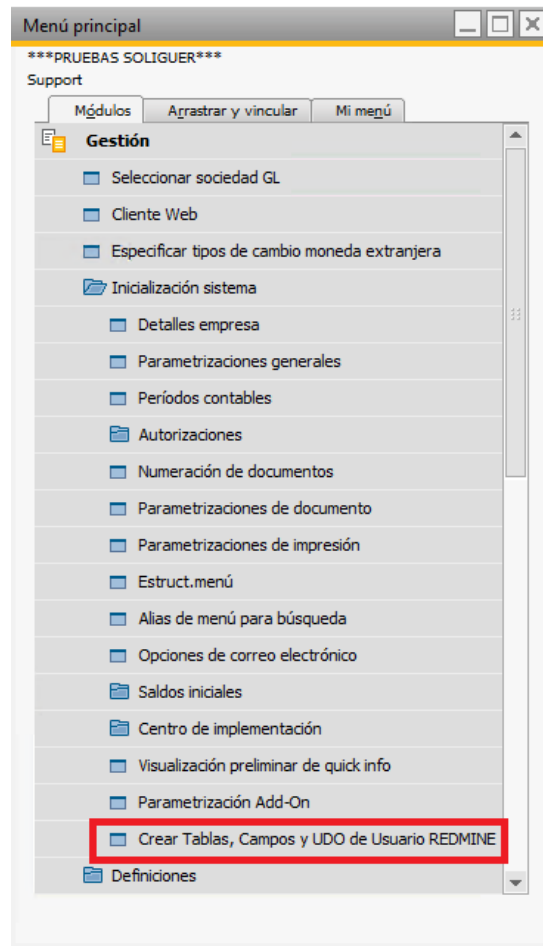


Imagen 11: Menú creación de campos

Una vez pulsemos, nos aparecerá el aviso que encontramos en la imagen siguiente. En este caso debemos marcar la opción “Sí” y empezará la creación.

Mientras se realiza la creación, aparece en la parte inferior izquierda de la pantalla una barra de progreso y finalmente nos indicará cuando se ha creado todo correctamente. En este caso, como los campos ya estaban creados, no hemos realizado esta acción para evitar problemas.

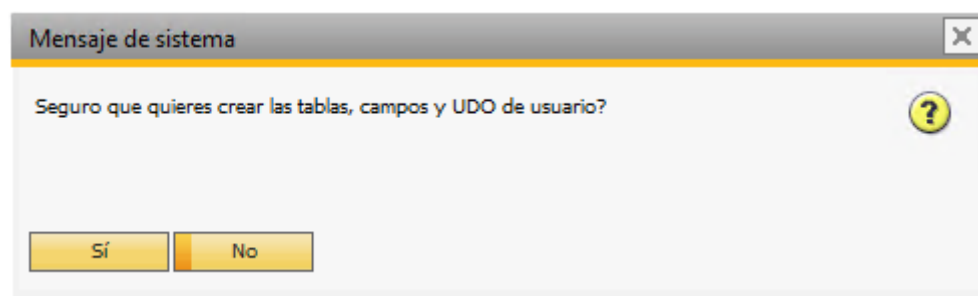


Imagen 12: Confirmar creación de campos

Si el paso anterior se ha realizado correctamente, debería aparecer un nuevo punto de menú justo encima del que hemos utilizado anteriormente, con el nombre “Parametrización Add-On”. A través de este punto de menú, accedemos a los parámetros que debemos rellenar para que nuestra integración pueda acceder al servidor de Redmine.

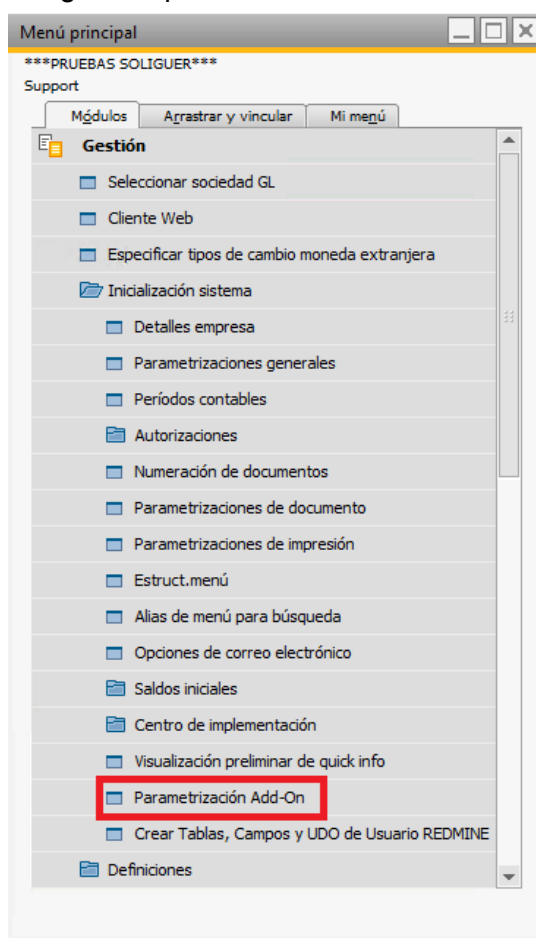


Imagen 13: Menú parametrizaciones

En la imagen siguiente, podemos ver como hemos llenado estos parámetros, indicando el nombre del usuario que realiza las acciones en Redmine, la dirección del servidor, el Token que podemos encontrar en Redmine y necesitamos para realizar las llamadas a la API, y por último, la ubicación dónde se encuentra el ejecutable de la integración.

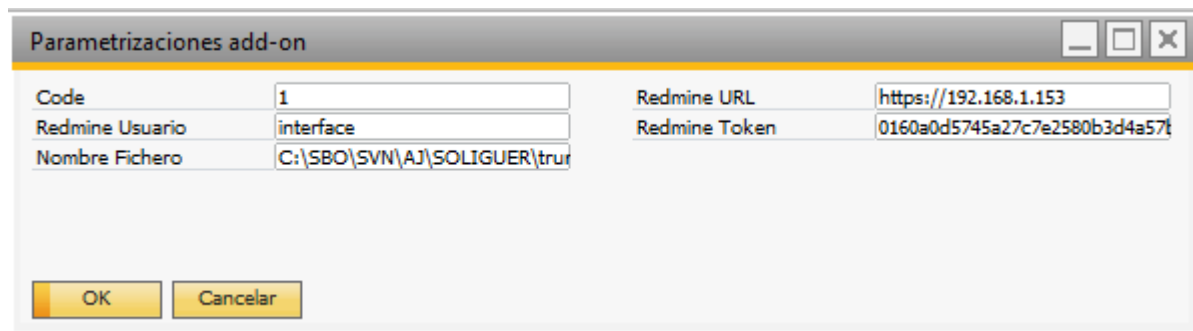


Imagen 14: Pantalla de parametrizaciones

Una vez tenemos estos datos completos, podemos proceder a utilizar la integración.

15.2 Subida de proyectos

Para realizar este ejemplo, vamos a crear dos nuevos proyectos en el apartado de “Proyectos - Definición” de Sap Business One, al crear los proyectos, debemos indicar la plantilla que le queremos asociar en el campo “Plantilla Proyecto Redmine”, y debemos marcar la opción “Es proyecto Redmine?”.

Como podemos comprobar, hay un gran número de proyectos definidos, por eso mismo hemos optado en este caso por introducir un nombre que muestre claramente que son proyectos de prueba, ya que estos proyectos se van a publicar en el servidor de Redmine y así se evitarán problemas de confusión.

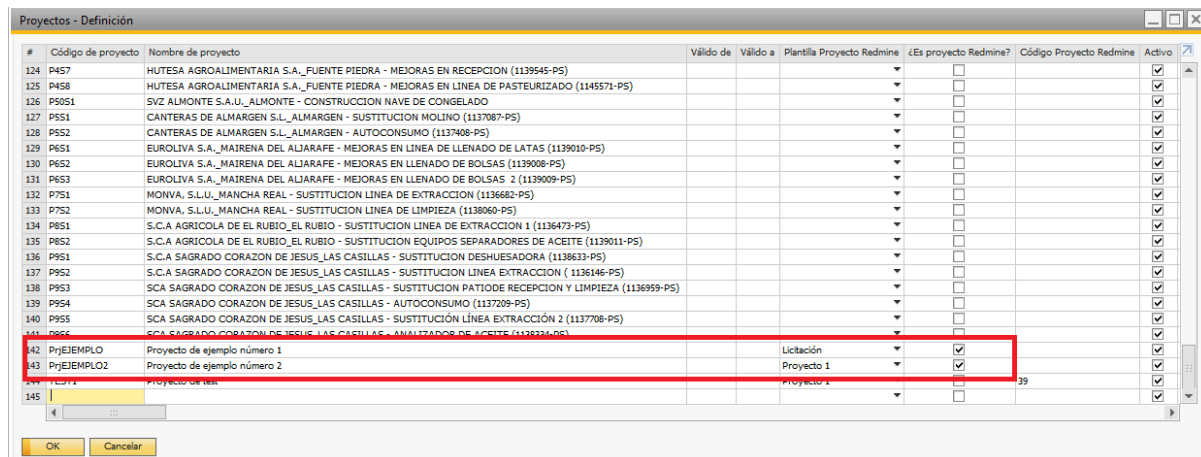


Imagen 15: Proyectos creados

Una vez creados los proyectos, ya podemos proceder a iniciar la integración, que se lanza a través del punto de menú que vemos en la siguiente imagen.

Sistema de enlace entre SAP Business One y una herramienta de gestión de proyectos

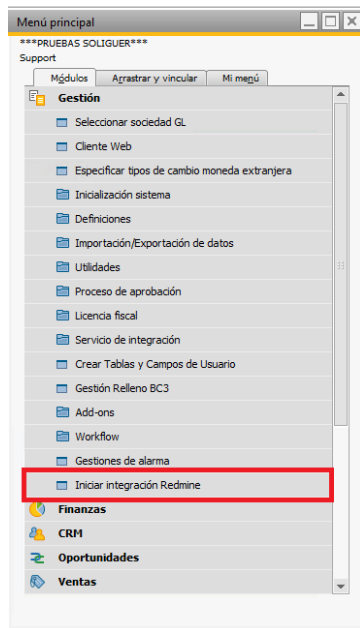


Imagen 16: Menú para iniciar la integración con Redmine

Cuándo se abre el ejecutable, simplemente debemos corroborar que aparece el mensaje “CONECTADO A LA SOCIEDAD [Nombre de la base de datos]”, y podemos proceder a iniciar la subida de proyectos.

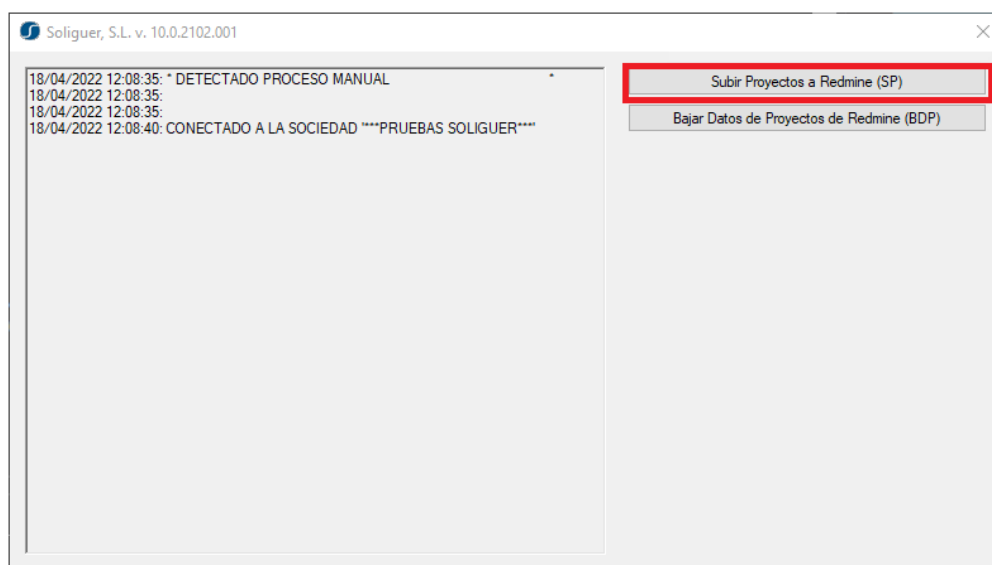


Imagen 17: Botón de subir proyectos en la integración con Redmine

Una vez los proyectos se hayan subido correctamente, veremos que nos indica con un mensaje que el proceso ha terminado correctamente. Llegados a este punto, nos movemos a *Redmine* para comprobar si es correcto, y si la información se corresponde con la indicada.

Sistema de enlace entre SAP Business One y una herramienta de gestión de proyectos

En las imágenes siguientes, se puede ver cómo, efectivamente, los proyectos se han publicado de forma correcta, además, vemos que cada uno tiene unas tareas diferentes, esto se debe a que cada proyecto ha adoptado la estructura de la plantilla indicada anteriormente.

Proyecto de ejemplo número 2

Vistazo WBS Gantt Tiempo empleado Tareas Gestión de recursos Documentos Calendario Archivos ...

Descripción del proyecto

Tareas (99)

✓	ID	TIPO DE TAREA	ESTADO	PRIORIDAD	ASUNTO	ASIGNATARIO	ACTUALIZADO POR ÚLTIMA VEZ
<input type="checkbox"/>	2994	Redacción proy. -delineación	No iniciada	Media	Anteproyecto - plano situación actual / situación futura		18/04/2022 12:12
<input type="checkbox"/>	2995	Redacción proy. -memoria/ppto	No iniciada	Media	Anteproyecto - valoración		18/04/2022 12:12
<input type="checkbox"/>	3036	Admon comercial	No iniciada	Media	Ventas - Realización de contrato		18/04/2022 12:12
<input type="checkbox"/>	2987	Ayudas - Estudio	No iniciada	Media	Ayudas - Análisis ayudas aplicables		18/04/2022 12:12
<input type="checkbox"/>	2993	Redacción proy. -memoria/ppto	No iniciada	Media	Elaboración de Anteproyecto		18/04/2022 12:12
<input type="checkbox"/>	2996	Redacción proy. -memoria/ppto	No iniciada	Media	Anteproyecto - visita técnica		18/04/2022 12:12
<input type="checkbox"/>	2988	Ayudas - Solicitud	No iniciada	Media	Ayudas - Solicitud subvención básica		18/04/2022 12:12
<input type="checkbox"/>	3006	Redacción proy. -memoria/ppto	No iniciada	Media	Proyectos - Redacción proyecto básico		18/04/2022 12:12
<input type="checkbox"/>	3001	Trámite Administrativo	No iniciada	Media	Trámites ad - Ayuntamiento Innecesaridad		18/04/2022 12:12
<input type="checkbox"/>	3008	Redacción proy. -memoria/ppto	No iniciada	Media	Proyectos - Calificación ambiental Ampliación		18/04/2022 12:12

Proyecto de ejemplo número 1

Vistazo WBS Gantt Tiempo empleado Tareas Gestión de recursos Documentos Calendario Archivos ...

Descripción del proyecto

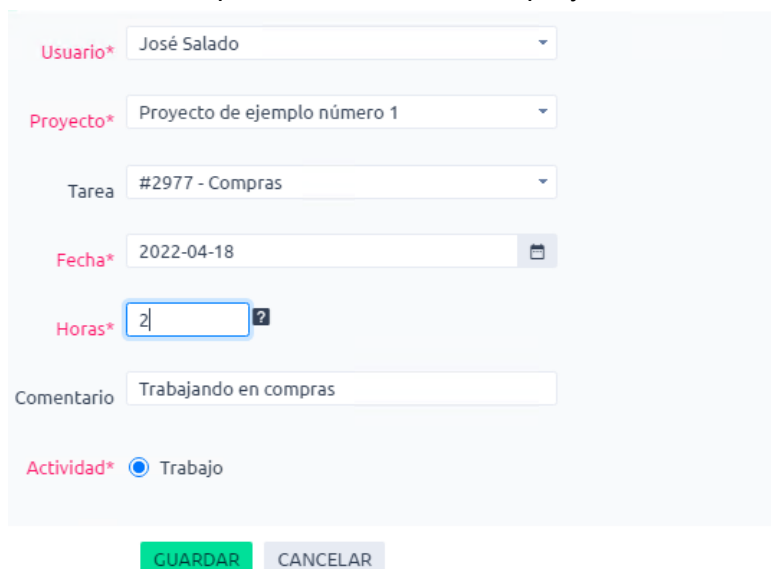
Tareas (28)

✓	ID	TIPO DE TAREA	ESTADO	PRIORIDAD	ASUNTO	ASIGNATARIO	ACTUALIZADO POR ÚLTIMA VEZ
<input type="checkbox"/>	2957	Licitaciones - Estudio	No iniciada	Media	Filtrado y estudio licitación		18/04/2022 12:12
<input type="checkbox"/>	2958	Licitaciones - Estudio	No iniciada	Media	Filtrado de licitaciones		18/04/2022 12:12
<input type="checkbox"/>	2959	Licitaciones - Estudio	No iniciada	Media	Estudio y resumen pliego de condiciones		18/04/2022 12:12
<input type="checkbox"/>	2961	Licitaciones - Pres. oferta	No iniciada	Media	Presentación oferta criterios automáticos		18/04/2022 12:12
<input type="checkbox"/>	2960	Licitaciones - Pres. oferta	No iniciada	Media	Presentación de oferta		18/04/2022 12:12
<input type="checkbox"/>	2962	Licitaciones - Pres. oferta	No iniciada	Media	Presentación oferta criterios juicio valor		18/04/2022 12:12
<input type="checkbox"/>	2964	Licitaciones - Adjudicación	No iniciada	Media	Presentación de documentación administrativa		18/04/2022 12:12
<input type="checkbox"/>	2965	Licitaciones - Adjudicación	No iniciada	Media	Presentación de plan seguridad y salud		18/04/2022 12:12
<input type="checkbox"/>	2966	Licitaciones - Adjudicación	No iniciada	Media	Presentación de Gestión de Residuos		18/04/2022 12:12
<input type="checkbox"/>	2967	Licitaciones - Adjudicación	No iniciada	Media	Presentación de Control de Calidad		18/04/2022 12:12

Imágenes 18 y 19: Proyectos creados en Redmine

15.2 Bajar datos de proyectos

Una vez los proyectos están publicados en Redmine correctamente, los usuarios que tengan este proyecto asignado pueden empezar a registrar el tiempo que le dedican a cada tarea. En la imagen siguiente se muestra un ejemplo de un usuario que registra un tiempo de 2 horas a una de las tareas que contiene uno de los proyectos creados anteriormente.



Formulario de registro de horas en Redmine. Campos:

- Usuario*: José Salado
- Proyecto*: Proyecto de ejemplo número 1
- Tarea: #2977 - Compras
- Fecha*: 2022-04-18
- Horas*: 2
- Comentario: Trabajando en compras
- Actividad*: Trabajo

Botones: GUARDAR, CANCELAR

Imagen 20: Registro de horas en Redmine

En este caso, se han creado tres registros de tiempo diferentes dos para un proyecto y uno para el otro, y a continuación, procederemos a bajar estos datos e introducirlos en *SAP Business One*. Para ello, acudimos nuevamente a la integración, y seleccionamos el botón "Bajar Datos de Proyectos de Redmine". Una vez pulsado este botón, veremos por pantalla el identificador del empleado que ha registrado la entrada de tiempo que se está consultando en ese momento, y finalmente, nos informa de que el proceso ha terminado, por lo tanto podemos volver a *SAP Business One* a consultar la información.

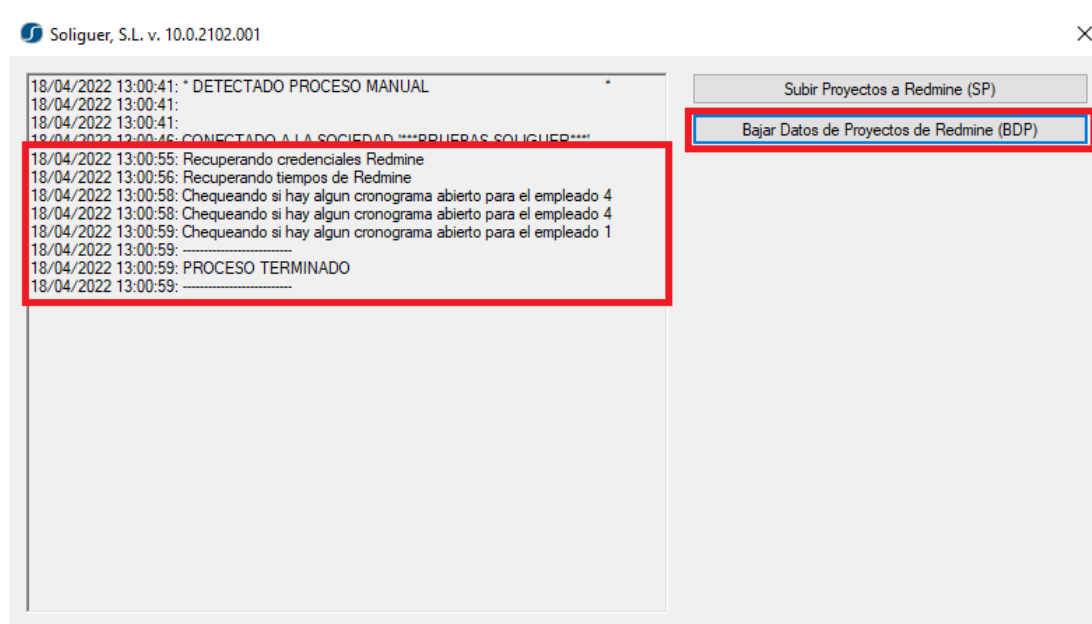


Imagen 21: Botón de bajar proyectos y resultado de la ejecución

En este caso, para consultar dicha información debemos acceder a la ventana de “Cronograma”. Una vez ahí, buscamos los empleados que han realizado el registro de tiempo. Como podemos ver en las siguientes imágenes, se establece como fecha de inicio del cronograma, la fecha en la que se añadió el primer registro de tiempo, posteriormente, si consultamos la información que se encuentra en las líneas, podemos ver como coinciden el tiempo dedicado y el proyecto con la información introducida en Redmine. Además, como SAP Business One nos obliga a establecer una hora de inicio y una hora de fin, pero en Redmine no se registra esta información, el código se ha diseñado para que el primer registro de cada día empiece a las 8 AM y acabe después de tantas horas como se haya trabajado, y los posteriores registros de ese mismo empleado y día empezarán a la última hora de finalización registrada, de esta forma se evitan solapamientos que impedirían añadir los registros.

The screenshot shows the 'Cronograma' window for employee 'SANCHEZ ROMERO'. The header information includes: Tipo: Empleado, Código: 1, Nombre: SANCHEZ ROMERO, Nombre: JOSE ANTONIO, Departamento: AREA PROYECTOS. The main table displays the following data:

#	Fecha	Hora de...	Hora de...	Clase actividad	Número de orden...	Proy. Financi...	Centro de coste	Posición de tr...	Nº llamada de ser...	Salto	Tiempo no facturable	Tiempo efectivo	Tiempo facturable	Id. Redmine
1	18/04/2022	08:00	10:00			PjEJEMPLO2						02:00	02:00	20
2														

The screenshot shows the 'Cronograma' window for employee 'SALADO ALONSO'. The header information includes: Tipo: Empleado, Código: 4, Nombre: SALADO ALONSO, Nombre: JOSE, Departamento: AREA PROYECTOS. The main table displays the following data:

#	Fecha	Hora de...	Hora de...	Clase actividad	Número de orden...	Proy. Financi...	Centro de coste	Posición de tr...	Nº llamada de ser...	Salto	Tiempo no facturable	Tiempo efectivo	Tiempo facturable	Id. Redmine
1	18/04/2022	08:00	10:00			PjEJEMPLO						02:00	02:00	18
2	18/04/2022	10:00	12:00			PjEJEMPLO						02:00	02:00	19
3														

Imágenes 22 y 23: Resultado final de los cronogramas

16. TESTING

En todo proyecto de software, es necesario que se realicen un gran número de pruebas y se analice a la perfección el funcionamiento del sistema antes de ser instalado en un entorno productivo. En este caso no iba a ser diferente, todo lo contrario, es especialmente importante la realización de estas pruebas mencionadas anteriormente, debido a que este sistema trabaja directamente con información importante del cliente, por lo tanto, si el sistema tiene errores cuando se empieza a utilizar de forma productiva, puede suponer grandes problemas tanto para el cliente como para la empresa desarrolladora.

Para poder realizar todos estas pruebas, inicialmente el consultor del proyecto se encargó de generar una base de datos en SAP Business One, con datos idénticos a la base productiva, para que se pudiera trabajar con la información más real posible, sin afectar a los datos que se utilizan diariamente en la empresa.

Las pruebas que se han realizado a este sistema han pasado por tres fases diferentes, dependiendo del estado del sistema y de quien realiza dichas pruebas. Durante el proceso de desarrollo, yo como desarrollador me he encargado de hacer innumerables pruebas, para ir comprobando que cada proceso estaba funcionando como estaba acordado. Posteriormente, una vez el desarrollo estaba finalizado, el consultor se encargaba de realizar pruebas más complejas, debido a que sus conocimientos sobre SAP Business One y los datos del cliente son más elevados. Por último, tenemos las pruebas realizadas por el propio equipo de la empresa cliente, este equipo está formado por diferentes personas que se van a encargar en el futuro de utilizar el sistema.

Durante el transcurso del proyecto, hubo un pequeño cambio en la planificación de estas pruebas mencionadas anteriormente, debido a que la idea inicial, era desarrollar el proyecto por completo, y posteriormente iniciar una etapa de testing junto con el cliente, antes de que el sistema se instale en la base de datos productiva. Esto se modificó debido a que el cliente necesitaba poder subir sus proyectos a Redmine con relativa urgencia, por lo tanto, primero se desarrolló esa funcionalidad y se realizaron las tres etapas de testing sobre ella, se instaló en la base de datos productiva, y posteriormente se realizó lo propio con el resto del proyecto.

16. CONCLUSIONES

Por último, voy a valorar cómo ha sido el desarrollo de este proyecto, enfocándome tanto en lo que he podido aportar al proyecto a nivel profesional, cómo en lo que me ha aportado a mi este proyecto a nivel profesional, formativo y personal.

Inicialmente, vamos a valorar si se han cumplido los objetivos marcados en un inicio. Dichos objetivos se han marcado de una forma bastante clara a mi parecer, puedo llegar a esta conclusión principalmente por la satisfacción del cliente con el resultado final, pero también porque si nos fijamos en los objetivos marcados en un inicio, el sistema desarrollado cumple perfectamente con cada uno de ellos, y con la solvencia requerida por los requisitos no funcionales que se han indicado anteriormente.

A nivel de conocimientos, este proyecto me ha enriquecido de una forma muy considerable. Realizar este proyecto me ha obligado a aprender a manejar perfectamente por el ambiente de SAP Business One, inicialmente a nivel de interfaz, y posteriormente a nivel de desarrollo, probablemente este último punto haya sido el que más me ha exigido, ya que aunque el código no sea el más complejo y extenso del mundo, he tenido que aprender inicialmente a desarrollar un *Add-on* para poder tener el ambiente listo para trabajar, posteriormente aprender a realizar la instalación de dicho *Add-on*, y finalmente tener batallas extensas con el código para conseguir tratar correctamente los datos de los dos sistemas que estaba enlazando. Además, ha sido la primera vez que, como desarrollador, he sido el encargado de diseñar por completo todo lo relacionado con el código, la programación y el diseño del sistema, por lo tanto he tenido que tomar responsabilidades que nunca antes había tomado, y lo agradezco profundamente.

Respecto a las competencias transversales marcadas en un inicio al inscribir el proyecto, vamos a realizar un breve análisis para comprobar si se han cumplido y en que nivel de profundidad:

CES1.2: Dar solución a problemas de integración en función de las estrategias, de los estándares y de las tecnologías disponibles. [En profundidad]

Al tratarse de una integración entre dos sistemas, esta competencia se ha cumplido durante todo el proyecto, ya que se han tenido que buscar alternativas y soluciones constantemente para lograr integrar la información de un sistema en el otro y viceversa.

CES1.5: Especificar, diseñar, implementar y evaluar bases de datos. [Un poco]

Para poder iniciar el proyecto, fue necesario crear inicialmente una base de datos en SAP Business One donde se almacena toda la información del cliente, y posteriormente, crear una nueva base de datos de test para poder realizar todas las pruebas necesarias sin modificar la base de datos productiva.

CES1.6: Administrar bases de datos (CIS4.3). [Un poco]

Para poder realizar la integración, se han tenido que crear diversos campos en las tablas de las bases de datos que utiliza SAP Business One, además de crear una tabla con diferentes campos para almacenar las credenciales. Además, constantemente se realizan queries en el código para poder acceder a la información que contiene SAP Business One.

CES1.7: Controlar la calidad y diseñar pruebas en la producción de software. [Un poco]

En mi caso particular, he sido el encargado de realizar la mayor parte de pruebas al código para asegurarme de que cada funcionalidad cumplía con su objetivo y no aparecían errores durante su ejecución.

CES2.1: Definir y gestionar los requisitos de un sistema software. [Bastante]

Como hemos podido comprobar en el documento, se han definido tanto los requisitos funcionales como no funcionales del sistema.

CES1.9: Demostrar comprensión en la gestión y gobierno de los sistemas software. [En profundidad]

Para poder realizar un proyecto de software, es necesario diseñar su arquitectura y de esta forma entender cómo se realizará la gestión de este proyecto. Éste caso no podía ser diferente, personalmente he sido el encargado de realizar todo el diseño y la gestión del sistema.

17. BIBLIOGRAFIA

[1] Aguilar, Q. (2021, 29 julio). *Redmine como herramienta de gestión de proyectos*. Ilimit. Recuperado 18 de octubre de 2021, de <https://www.ilimit.com/blog/redmine-como-herramienta-de-gestion-de-proyectos/>

[2] Author, G. (2021, 18 mayo). *Qué son los stakeholders, qué tipos existen y de qué manera impactan a una empresa*. Rock Content - ES. Recuperado 20 de septiembre de 2021, de <https://rockcontent.com/es/blog/que-es-un-stakeholder/>

[3] D. (2019, 26 junio). *¿Qué es un sistema ERP y para qué sirve?* Defontana. Recuperado 24 de septiembre de 2021, de <https://www.defontana.com/cl/que-es-un-sistema-erp-y-para-que-sirve/>

[4] SAP ASAP. (2017, 12 junio). Consultoría SAP. Recuperado 26 de septiembre de 2021, de <https://www.consultoria-sap.com/2012/07/sap-asap.html>

[5] *API REST: qué es y cuáles son sus ventajas en el desarrollo de proyectos*. (1970, 1 enero). BBVA API_Market. Recuperado 10 de abril de 2022, de <https://www.bbvaapimarket.com/es/mundo-api/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos/>

[6] *Arquitectura cliente servidor*. (2016, 17 octubre). OposicionesTic. Recuperado 8 de abril de 2022, de <https://oposicionestic.blogspot.com/2011/06/arquitectura-cliente-servidor.html>

[7] Balagueró, T. (2021, 17 diciembre). *Como funciona la arquitectura del sistema SAP.*

Deusto. Recuperado 5 de abril de 2022, de <https://www.deustoformacion.com/blog/gestion-empresas/funciona-arquitectura-sistema-sap>

[8] EcuRed. (2013, 17 julio). *Requisitos no funcionales - EcuRed.* Recuperado 5 de abril de 2022, de https://www.ecured.cu/Requisitos_no_funcionales#Orientados_al_Usuario

[9] Huaman, W. C. (2018, 7 septiembre). *Los 10 patrones comunes de arquitectura de software.* Medium. Recuperado 8 de abril de 2022, de <https://medium.com/@maniakhitoccori/los-10-patrones-comunes-de-arquitectura-de-software-d8b9047edf0b>

[10] pmoinformatica.com. (2015, 13 abril). *Requerimientos no funcionales: Una clasificación.* La Oficina de Proyectos de Informática. Recuperado 5 de abril de 2022, de <http://www.pmoinformatica.com/2015/04/requerimientos-no-funcionales-una.html>

[11] *¿Que es un AddOn para SAP Business One?* (2017, 1 junio). ApiData SAP. Recuperado 4 de abril de 2022, de <https://www.apidata.com.mx/single-post/2017/06/01/-que-es-un-addon-para-sap-business-one>

[12] Refactoring.Guru. (2014, 10 enero). *Refactoring and Design Patterns.* Recuperado 8 de abril de 2022, de <https://refactoring.guru/>