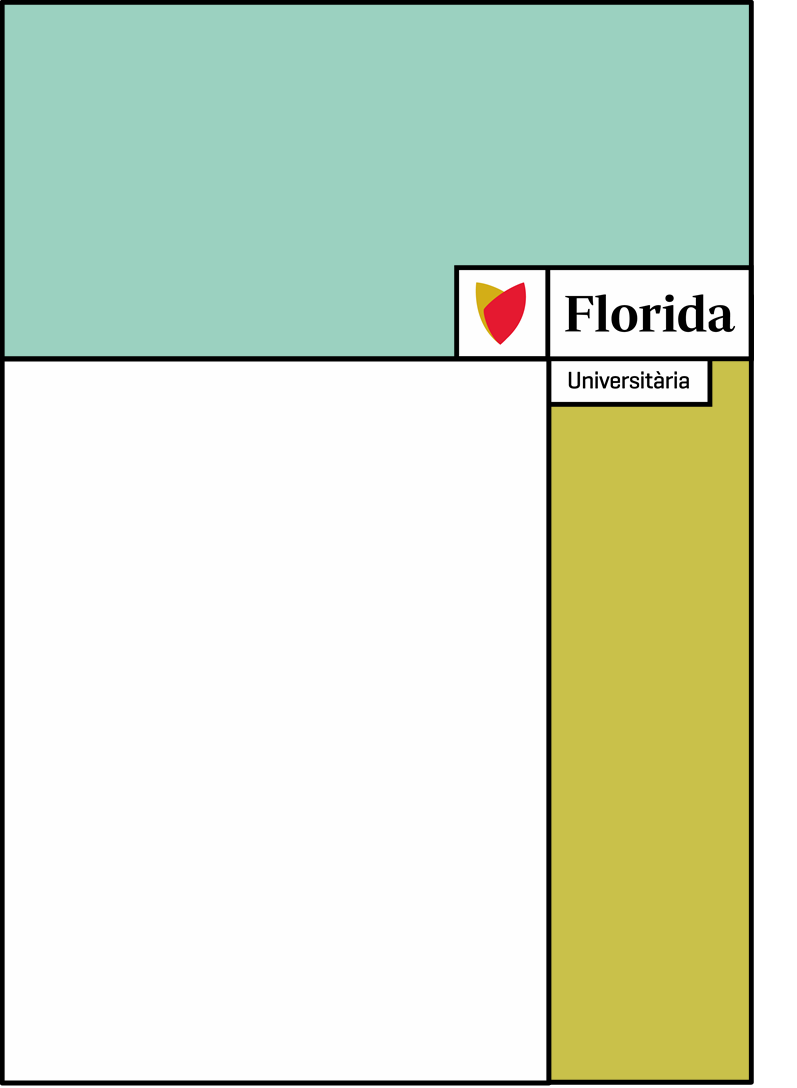
**Memoria P.F.C.**



**D.A.W**

**Proyecto Final de Ciclo**

**Tutor: Nacho Muñoz**

**Apellidos y nombre del autor/a:**

**Coviza Del Moral, Pablo**

**Desarrollo de Aplicaciones Web**

**Índice**

1. Portada……………………………………………………………………...0
2. Índice………………………………………………………………………..1
3. Resumen……………………………………………………………………2
4. Justificación y objetivos……………………………………………………3
5. Desarrollo…………………………………………………………………..4

5.a Análisis del mercado……………………………………………………...5

5.b Metodologías utilizadas…………………………………………………..6

5.c Descripción de componentes…………………………………………….7

5.d Estructura del proyecto……………………………………………………8

5.e Resultados obtenidos…………………………………………………….9

1. Conclusiones……………………………………………………………..10
2. Líneas futuras de trabajo………………………………………………..11
3. Glosario……………………………………………………………………12

9 Bibliografía………………………………………………………………….14

**Resumen**

Este Word es el documento de la memoria para el proyecto final de ciclo en Desarrollo de Aplicaciones Web. En él encontraremos varias secciones donde se detalla cada apartado.

Este proyecto tiene el objetivo de plasmar todo lo aprendido en los dos cursos de D.A.W. de manera sencilla y muy práctica.

Se ha desarrollado basado en un patrón de Modelo – Vista - Controlador puesto que de esta manera convierte el desarrollo de aplicaciones complejas en un proceso mucho más manejable. Permite a varios desarrolladores trabajar simultáneamente en la aplicación.

Las tecnologías utilizadas han sido PHP, Doctrine ORM, HTML5, TWIG, CSS3, JavaScript, Visual Studio Code, Bootstrap y PHPMyAdmin.

El nombre comercial escogido para el proyecto es “Coviza Gateway”.

Coviza Gateway es un prototipo de ERP de empresa orientado a la automatización de procesos y consulta de datos administrativos. Se pueden consultar pedidos, clientes, empleados y demás datos con facilidad. A parte de

hacer uso de CRUD para almacenamiento persistente en bases de datos y así

poder insertar, leer, modificar y/o eliminar dichos datos mediante formularios muy intuitivos.

**Justificación y objetivos**

* Optimizar los procesos legislativos de la empresa como, por ejemplo, la expedición de nóminas para que se pueda automatizar el proceso y realizar de forma ágil y sin errores.
* Favorecer el acceso a la información
* Tomar decisiones de forma más rápida al disponer de datos veraces actualizados o Golden Records.
* Tener un mayor control sobre los datos, pudiendo identificar posibles fallos y obtener su trazabilidad.
* Aumentar la seguridad
* Reducción de costes empresariales puesto que el ERP está desarrollado para reducir el margen de errores, eliminar duplicidades, etc… y así repercutir en el ahorro monetario para la empresa.

**Desarrollo**

El desarrollo de este proyecto tiene una propuesta de valor clara de integrar las bbdd de la empresa/cliente en una misma plataforma.

Tratar de diferenciarse del resto de productos del mercado con funcionalidades sencillas a la vez que prácticas.

El coste material será el mínimo gracias a que se trata de un proyecto sin ánimo de lucro, con voluntad de convertirse en opensource.

Las características principales serán las pipelines aplicadas a través de formularios, haciendo el uso del mismo apto para todo tipo de públicos.

La idea es conseguir tener un producto final funcional y responsive.

En los siguientes apartados detallaremos cada parte de este y ahondaremos más en profundidad.

**Análisis del mercado**

No se puede negar que existen propuestas similares a este proyecto en el mercado, aún así nos trataremos de diferenciar del resto en algo tan sencillo como la posible escalabilidad del producto final aprovechándonos del patrón MVC y de la inclusión del Framework Bootstrap para resaltar la interfaz y hacer del uso del producto una experiencia agradable al usuario.

Por todo lo anterior, considero que existe una gran oportunidad de negocio con este producto porque la gran mayoría de productos similares están bastante desfasados a los tiempos actuales, ya que, hacen uso de tecnologías más antiguas y arcaicas.

El valor añadido con este producto puede ser un gran atractivo para el consumidor.

Forma

Descripción generada automáticamente

**Metodologías utilizadas**

En este proyecto hemos seguido una metodología SCRUM adaptada al centro La Florida, ya que hemos sido manejados en pequeños grupos con un tutor a cargo del equipo, pero

en grandes rasgos la metodología utilizada ha sido esta. También personalmente me he apoyado en una herramienta llamada Trello que incorpora un sistema de tableros

para ir trazando mini objetivos a corto plazo.

SCRUM al estar enmarcada dentro de las metodologías ágiles, se basa en los siguientes aspectos:

- Flexibilidad en la adopción de cambios y nuevos requisitos durante la elaboración del proyecto.

- Factor humano

- Colaboración e interacción con el cliente

- Desarrollo iterativo como forma de asegurar buenos resultados

La metodología SCRUM contiene 3 caracteristicas fundamentales:

1. Transpariencia

- Todos los miembros del equipo implicados en la tarea tienen conocimiento de qué ocurre en el proyecto y cómo ocurre.

Esto facilita una visión global del mismo.

2. Inspección

- Los miembros del equipo inspeccionan con asiduidad el progreso del proyecto para detectar posibles problemas.

Esto permite saber que el trabajo fluye y que el equipo funciona de manera auto organizada.

3. Adaptación

- Cuando existe algo que cambiar, el equipo se ajusta para conseguir el objetivo dentro del sprint.

Esta es la clave para el éxito en proyectos complejos y cambiantes donde se precisa de adaptabilidad y flexibilidad.

Para lograr resultados de calidad el equipo tiene como foco aportar valor y ofrecer resultados óptimos que permitan cumplir los objetivos

del cliente. Para ello los equipos son auto organizados y multifuncionales, es decir, cada uno se encarga de unas tareas específicas y de cumplir

los plazos que se hayan establecido. Esto garantiza la entrega de valor del equipo completo y de una manera auto suficiente.

Existen varios roles en este tipo de metodología:

1. Product Owner

- Es el responsable de hablar con el cliente y de maximizar el trabajo del equipo de desarrollo

2. SCRUM Master

- Es el encargado de que las técnicas SCRUM sean comprendidas y ejecutadas por el equipo, un líder que se encarga de eliminar

impedimentos que tenga el equipo dentro de un sprint.

3. Developer Team

- Son los encargados de desarrollar las tareas marcadas por el Product Owner de una manera auto organizada.

Los hitos de esta metodología de trabajo SCRUM son los siguientes:

1. Sprint

- El sprint es el corazón de Scrum, es el contenedor de los demás hitos del proceso. Todo lo que ocurre en una

iteración para entregar valor está dentro de un sprint y la duración máxima es de un mes ya que los sprints largos

pueden hacer que se pierda feedback valioso del cliente y poner en peligro el proyecto.

2. Sprint planning

- En esta reunión todo el equipo Scrum define qué tareas se van a abordar y cuál será el objetivo del sprint. La definición de qué se va

a hacer implica que el equipo tenga un objetivo y se encuentre comprometido con la entrega de valor que se hará al cliente al final

del sprint. A esto se le llama sprint goal.

3. Daily meeting

- Es una reunión diaria dentro del sprint que tiene como máximo 15 minutos de duración. En ella debe participar, sí o sí, el equipo de

desarrollo y el Scrum Master. En esta reunción se hacen varias preguntas como... ¿Qué hice ayer?, ¿Qué voy a hacer hoy?, ¿Tengo algún

impedimento que necesito que me solucionen?

4. Sprint review

- La review del valor que vamos a entregar al cliente se hace en esta reunión, al final de cada sprint. En ella el Product Owner presenta

lo desarrollado al cliente y el equipo de desarrollo muestra su funcionamiento. El cliente valida los cambios realizados y además brinda feedback

sobre nuevas tareas.

5. Sprint retrospective

- Es la reunión del equipo en la que se hace una evaluación de cómo se ha implementado la metodología Scrum en el último sprint.

Es una gran oportunidad para el equipo Scrum de inspeccionarse a sí mismo, proponiendo mejoras para el siguiente sprint.

Por último quiero destacar cuáles son las ventajas y desventajas de la metodología SCRUM:

Las ventajas que aporta esta metodología de trabajo es la facilidad de aprender SCRUM, los roles, hitos y herramientas son claros y tienen un

objetivo muy bien definido.

El cliente puede empezar a usar el producto muy rápidamente.

Se agiliza el proceso, ya que la aportación de valor es muy frecuente.

Existe menor probabilidad de sorpresas o imprevistos, porque el cliente está viendo asiduamente el proyecto.

Las desventajas de SCRUM aunque sea fácil de aprender, es muy difícil de implementar eficientemente. Estos cambios deben ser adoptados desde los

altos mandos hasta los clientes.

Existen problemas a la hora de tener equipos multidisciplinares por la complicación que existe de encontrar personas capaces de hacer

todo el trabajo de un equipo.

El equipo puede optar por realizar el camino más corto hacia el objetivo del sprint y esto puede resultar en una pérdida de calidad sustancial.

**Descripción de componentes**

La aplicación se compone de una herramienta para la gestión de bases de datos MySQL. Esta base de datos es imprescindible a la hora de acceder a sus datos en crudo, consultar el estado de la bbdd, hacer modificaciones en la estructura de las tablas o modificar los registros. Para todo ello necesitamos un software de cliente para el acceso al sistema gestor.

Utilizaremos PHP como lenguaje de programación de código abierto. Es muy popular para el desarrollo web y que puede ser incrustrado en HTML5. Lo mejor de utilizar PHP es su extrema simplicidad para programadores Junior, pero a su vez ofrece muchas características avanzadas para los programadores Senior.

Para el Frontend haremos uso de tecnologías como CSS3 para aplicar estilos a los elementos de la web, HTML5 para presentar contenido en la web, Bootstrap para desarrollar un producto ‘responsive’ orientado a todo tipo de dispositivos y TWIG para crear plantillas más complejas de forma muy sencilla.

Imagen que contiene Texto

Descripción generada automáticamente

Icono

Descripción generada automáticamente

**Estructura del proyecto:**

* config: contiene todos los directorios necesarios para la web como hojas de estilos, imágenes...

dbConfig: contiene el archivo con extensión json de configuración de la aplicación

* + Routes: contiene el archivo json con las rutas a las distintas vistas
* public: contiene los archivos referentes a los comandos ejecutables desde la terminal.
* Css: contiene las hojas de estilo
* Images: contiene las imágenes usadas en la aplicación
* src: contiene el código fuente de la aplicación
  + Entity: contiene los distintos ficheros donde se definen las entidades
  + Controllers: contiene los controladores que dan funcionalidad a la aplicación.
  + Core:
  + Repository: contiene las clases que albergan todas las acciones de consulta en base de datos.
* templates: contiene las plantillas TWIG de las diferentes vistas
* vendor: contiene el core de Symfony y las librerías externas.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

**Resultados obtenidos**

Aunque la idea inicial era de crear un ERP completo de empresa, la realidad es que no he llegado a cumplir mis expectativas en cuanto a toda la funcionalidad que me hubiera gustado implementar al producto final.

El producto final tiene unos tiempos de respuesta bastante bajos y adecuados a la experiencia óptima de usuario.

La experiencia de usuario debería de ser agradable a la hora de navegar por la aplicación.

Se ha tratado de que sea intuitiva y fácil de consumir por el usuario.

Estoy muy contento con el resultado obtenido teniendo en cuenta el poco tiempo que he dispuesto para desarrollarlo.

**Conclusiones**

**Conclusiones generales:**

Como todo proyecto final de ciclo, el objetivo de este es satisfacer las necesidades del cliente, en este caso de la Universidad La Florida de Catarroja, la cual pedía el desarrollo de una aplicación propuesta por el propio alumnado a partir de las líneas establecidas por el equipo de profesorado y más concretamente previa aprobación del tutor encargado del PFC. Para lograr dicho objetivo, se ha recurrido a una planificación validada por el tutor del PFC Nacho Muñoz. Los objetivos marcados en cada SPRINT realizado cada aproximadamente 15 días se han cumplido en el plazo previsto y por lo tanto concluimos en que no ha habido retrasos a la hora de entregar el proyecto finalizado.

**Conclusiones personales:**

Desde mi punto de vista el proyecto ha sido un gran reto. He tratado de plasmar todo lo aprendido durante estos 2 años de aprendizaje en las diferentes asignaturas cursadas de la mejor manera posible y siempre tratando de aplicar buenas prácticas de programación.

Dentro de todas las posibilidades que se me ofrecieron he escogido estas tecnologías para desarrollar el PFC porque siento que son las idóneas para trasladar todo mi aprendizaje de la mejor manera posible.

Estos 2 años han requerido que aportara mi 200% en cada semana de trabajo, compaginando trabajo a jornada completa con los estudios en modalidad semipresencial, y la verdad que no ha sido nada fácil. Largas noches de trabajo con mucho café, codo a codo con mis compañeros de curso, webinars interminables de programación y un largo etcétera que han hecho que me sienta muy orgulloso de lo que estoy logrando.

Así que solo queda seguir disfrutando de este maravilloso mundo de constante formación y agradecer la dedicación de los profesores implicados.

**Líneas futuras de trabajo**

* Convertir el proyecto en un proyecto Opensource
* Migrar la aplicación al Framework Symfony
* Mejorar la interfaz de la página web
* Incluir pagos recurrentes a modo de suscripción mensual

**Un letrero de color blanco

Descripción generada automáticamente con confianza baja**

**Glosario**

**API:** Acrónimodeinterfaz de programación de aplicaciones (Application Programming Interface). Es el conjunto de funciones y procedimientos para ser utilizados por otro software.

**Framework:** Conjunto de librerías que ayudan al desarrollo rápido de aplicaciones.

**PFC:** Proyecto Final de Ciclo

**CRUD:** Acrónimo deCrear, Leer, Actualizar y Borrar (Create, Read, Update, Delete)

**MVC:** Acrónimo de Modelo, Vista y Controlador. Es un patrón que se encarga de dividir el código Frontend y Backend en componentes separados. De esta manera es mucho más fácil administrar y hacer cambios a cualquiera de los lados sin que interfieran entre sí.

**BBDD:** Abreviatura de Base de Datos. Conjunto de datos comunes almacenados según las necesidades de la empresa/usuario.

**PHP:** Acrónimo recursivo de Hypertext Preprocessor, es un linguaje de programación de código abierto.

**HTML5:** Acrónimo de Lenguaje de Marcado de Hypertexto (HyperText markup Language). Es el estándar con el que están programadas todas las páginas web.

**JavaScript:** Lenguaje de programación orientado a objetos. Se utiliza mayormente para dotar de mayor interactividad y dinamismo a las webs.

**Frontend:** Es la parte de un sitio web que interactúa con los usuarios.

**Backend:** Es la parte que se conecta con la bbdd y el servidor que utiliza dicho sitio web.

**Responsive:** Diseño web adaptativo, es una técnica de diseño web que busca la correcta visualización de una misma página web en distintos dispositivos.

**TWIG:** Motor de plantilla para PHP y es de código abierto.

**Bootstrap:** Framework multiplataforma o conjunto de herramientas de código abierto (opensource) para desarrollos web responsive con HTML, CSS y JavaScript.

**BIBLIOGRAFÍA:**

**Stackoverflow**

<https://stackoverflow.com/>

**FreeCodeCamp**

<https://www.freecodecamp.org/>

**Wikipedia**

<https://www.wikipedia.org/>

**TWIG**

<https://twig.symfony.com/>

**Bootstrap**

<https://getbootstrap.com/>

**CSS3**

<https://en.wikipedia.org/wiki/CSS>

**HTML5**

<https://en.wikipedia.org/wiki/HTML5>

**Doctrine ORM**

<https://www.doctrine-project.org/projects/orm.html>

**Visual Studio Code Plugins**

<https://code.visualstudio.com/docs/editor/extension-marketplace>

**w3schools**

<https://www.w3schools.com/>

**mdn**

<https://developer.mozilla.org/es/>