



26/03/2021

RAYCO MARTÍN CORUJO - 2º Daw

Índice

→ Apartado 1. Introducción	Página 2
• 1.1. Introducción y Presentación	Página 2
→ Apartado 2. Análisis de requerimientos	Página 2-4
• 2.1. Requisitos Técnicos	Página 3
• 2.2. Requisitos Operativos	Página 3
• 2.3. Factibilidad del Proyecto	Página 4
• 2.4. Desglose de Horas y Coste Total	Página 4
→ Apartado 3. Diseño del Sistema	Página 5-9
• 3.1. Introducción	Página
• 3.2. Modelo relacional de base de datos	Página
• 3.3. Patrón MVC	Página
• 3.4. Jerarquía de Privilegios	Página
• 3.5. Esquema de la Interfaz	Página
→ Apartado 4. Implementación	Página
• Framework	Página
• Versiones	Página
• Alojamiento de la web	Página
→ Apartado 5. Validación y Pruebas	Página
• Objetivos	Página
• Tipos	Página
• Pruebas con el cliente	Página
→ Apartado 6. Mantenimiento	Página
• Tipos	Página
• Copias de seguridad	Página
• Política de actualizaciones	Página

A medida que crece un negocio, este requiere de un mejor manejo de los datos que posee, parte del éxito de estos comercios radica en una buena organización de la información que controlan, con el fin de una mejor optimización del trabajo.

Librería Papelería Volcán es una librería situada en Tinajo, con una apertura de comercio hace pocos meses, los cuales son cruciales para la atracción de clientes. Durante la época de campaña ha experimentado un crecimiento inesperado en la reserva de libros de texto, por estos motivos se da la necesidad de crear e implementar un software que satisfaga todas las reservas que reciben en época de campaña.

Yo, Rayco Martín Corujo, estudiante del segundo curso de Desarrollo de Aplicaciones Web, soy el encargado de que esta empresa consiga con este software su objetivo, complacer todas las necesidades que requieran y proporcionarles un ambiente de completa organización y seguridad.

Análisis de requerimientos

La funcionalidad principal de esta aplicación web es la gestión de reservas de libros de texto para la librería, se caracterizará ya que tendrá una página principal de bienvenida donde todos los clientes podrán acceder sin el requisito de estar registrado.

Habrà en la parte superior de la web un apartado similar al de "Reserva ya" donde los clientes accederán a un espacio donde podrán seleccionar todos los libros que desean y hacer la reserva de la misma, al finalizar la reserva se redireccionará a una pasarela de pago donde deberán aportar el 15% del precio de la reserva con el fin de que los clientes se comprometan con la misma.

Respecto al administrador de la página, tendrá un espacio dedicado donde podrá tener acceso a todas las reservas realizadas por los clientes así como sus datos, además podrá añadir y eliminar libros y en general tener el control de todo lo que ofrecen y de todo lo que realizan sus clientes.

Requisitos técnicos

Arquitectura	El sitio web será compatible con los principales navegadores de Internet (Mozilla Firefox, Internet Explorer, Opera, Safari, Google Chrome).
Arquitectura	Los contenidos se almacenarán en un sistema gestor de bases de datos relacional.
Normativas y/o estándares	El portal web cumplirá las condiciones de accesibilidad descritas por la WAI (Web Accessibility Initiative) de nivel A.

Requisitos operativos

Operativa	El sitio web ofrecerá un diseño atractivo y tendrá en cuenta la imagen corporativa de la librería
Operativa	La estructura de contenidos debe ser clara
Operativa	La gestión de los contenidos podrá ser realizada por personal no técnico sin conocimientos de programación, de forma fácil e intuitiva.
Operativa	El sitio web permitirá la inscripción online a las reservas que oferta la librería
Operativa	El sitio web tendrá un apartado para que los clientes puedan consultar dudas
Operativa	El sitio web tendrá un espacio dedicado

	para la reserva de libros de texto
Operativa	El sitio web ofrecerá al administrador un espacio dedicado para la gestión de la página, donde entre otros datos podrá ver un CRUD de las reservas con sus clientes

Con estos requisitos llegamos a la conclusión de que el proyecto es un proyecto factible, ya que para el problema que nos plantea el cliente disponemos tanto de tiempo como de recursos para solucionarlo, además podemos decir que no es común encontrar aplicaciones con objetivos similares en el mercado las cuales se puedan llegar a comprar, todos estos sistemas de reservas que nos encontramos son propios y van integrados en la página web que dispone ese comercio o establecimiento, por lo que no podemos recomendarle al cliente una aplicación donde el costo que le supondrá sea menor.

Estos requisitos reflejados en el documento nos llevarán varias horas de trabajo, las cuales podemos ver desglosadas a continuación.

Desglose de horas

HORAS EMPLEADAS PARA EL FRONT-END: 25 Horas

HORAS EMPLEADAS PARA EL BACK-END (CONEXIÓN CON LA BBDD Y GESTIÓN CON EL FRAMEWORK): 45 Horas

HORAS EMPLEADAS PARA LA CREACIÓN DE LAS TABLAS Y SUS RELACIONES: 10 Horas

HORAS EMPLEADAS PARA LAS VALIDACIONES Y PRUEBAS PREVIAS A LA ENTREGA: 8 Horas

HORAS EMPLEADAS PARA LAS PRUEBAS CON EL CLIENTE EN EL COMERCIO: 10 Horas

HORAS EMPLEADAS PARA LOS PRIMEROS 6 MESES DE MANTENIMIENTO INCLUIDO: 12 Horas

HORAS TOTALES PARA LA REALIZACIÓN Y DESPLIEGUE DEL PROYECTO: 105 Horas

Precio Total del Proyecto: 1260 euros
--

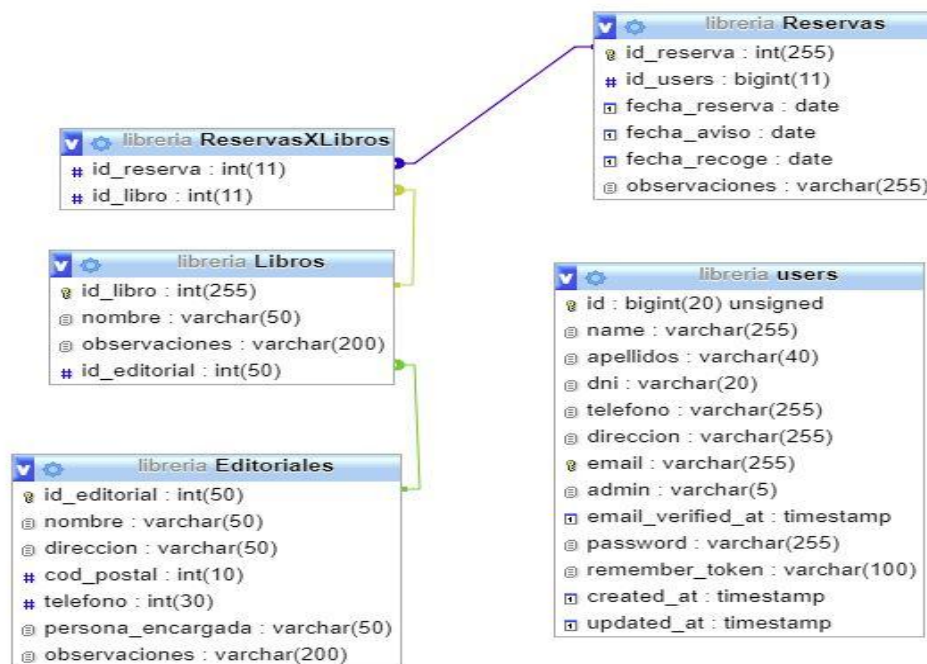
Diseño del sistema

El diseño del software se encuentra en el núcleo técnico de la ingeniería del software. Una vez que se han analizado los requisitos del software, el diseño, la implementación de código y las pruebas son las tres técnicas siguientes. Cada actividad transforma la información de manera que dé lugar en última instancia a un software validado.

En general, la actividad del diseño se refiere al establecimiento de las estructuras de datos, la arquitectura general del software, las representaciones de la interfaz y de los algoritmos. Por tanto, el diseño debe contemplar todos los requisitos explícitos obtenidos en la fase de análisis, y debe ser una guía que puedan leer y entender tanto los que construyen el código como los que prueban y mantienen el software.

Modelo Relacional de la Base de Datos

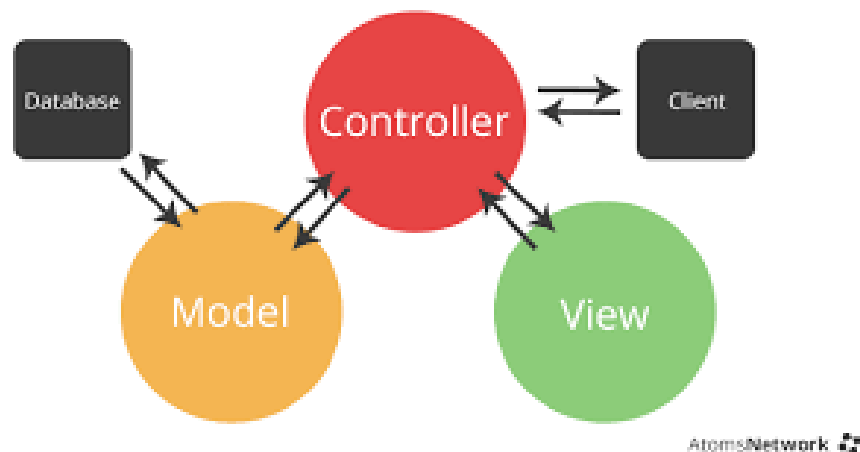
A continuación, pasamos a tablas los datos que serán necesarios para nuestro proyecto. Este modelo es algo más técnico porque está orientado al personal informático y generalmente tiene traducción directa al modelo físico que entiende el SGBD.



Patrón MVC

Vamos a utilizar el patrón Modelo - Vista - Controlador para la implementación de nuestro sistema. MVC es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos:

- La vista presentará los contenidos obtenidos del modelo en un formato claro y accesible para que el usuario del sistema de información pueda interactuar con la misma.
- Cuando un usuario interactúa con la vista haciendo clic en algún botón o realizando alguna acción, el controlador recibirá la petición solicitada y generará una acción para invocar peticiones al modelo e incluso a la propia vista.
- El modelo es la representación específica de la información con la que opera el sistema. Representa la lógica de negocio y el sistema de gestión de base de datos que almacena todos los datos del proyecto.



Jerarquía de privilegios

Un punto crítico en el desarrollo del software es el de los permisos y es importante disponer de algún mecanismo que permita establecer qué operaciones podrá realizar cada usuario.

Para nuestro proyecto software es conveniente utilizar un sistema de roles donde tendremos a los usuarios que serán clientes del establecimiento y al administrador del sistema, que puede ser uno o varios.

Por lo general se suelen definir los siguientes roles:

- **Usuario:** usuario que simplemente podrá realizar las acciones propias de la página web para clientes.

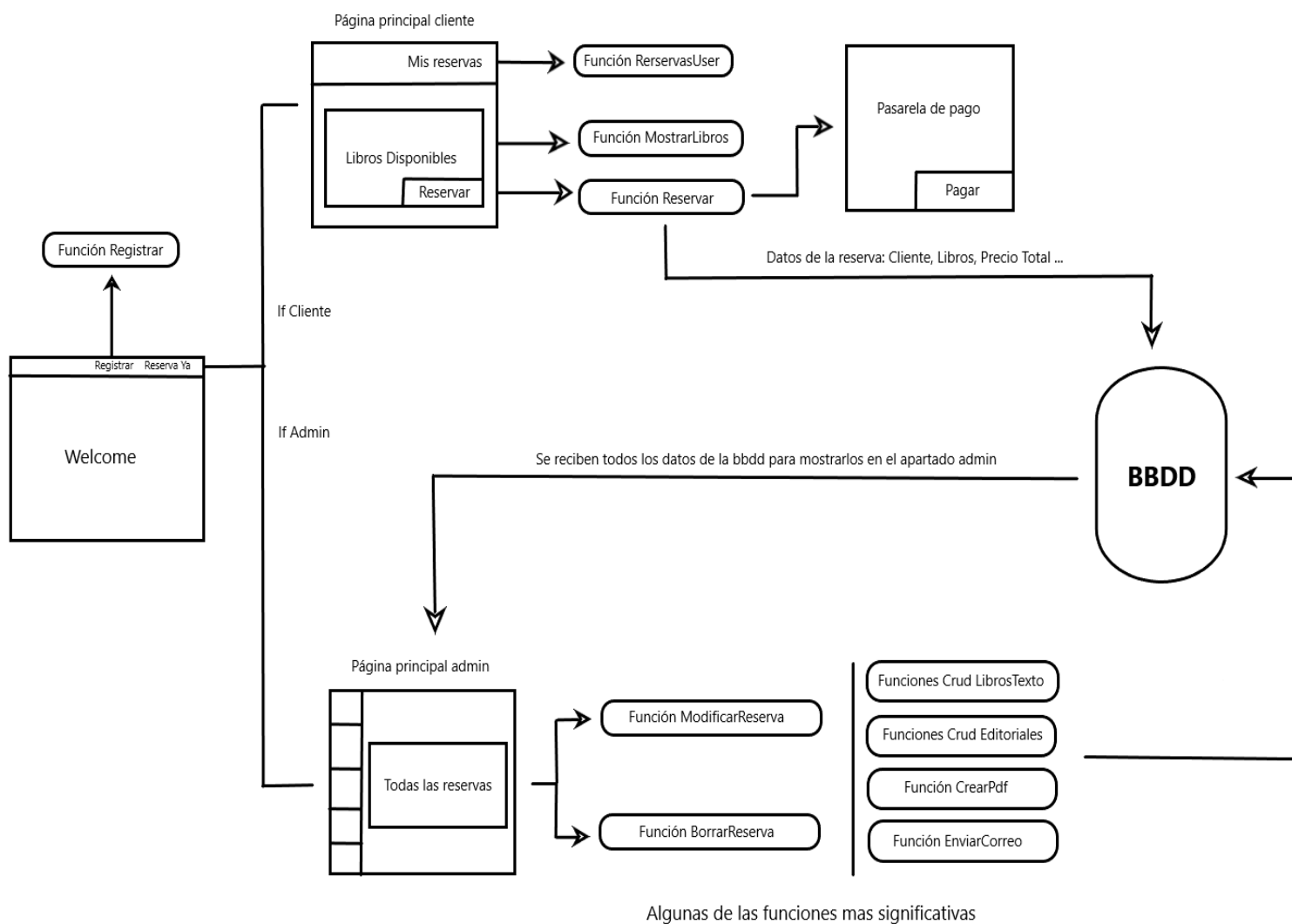
- **Administrador:** es el responsable técnico y puede acceder a cualquiera de los aspectos del sistema, configurando o modificando cualquier parámetro de éste. Este usuario tiene control total sobre el sitio web, por lo que puede realizar cualquier operación de configuración o mantenimiento sobre este.

Esquema de la interfaz

Los elementos más importantes de nuestro sitio web son:

- El sitio debe explicar a quién pertenece y que permite hacer a quienes lo visitan de manera simple y rápida.
- Debe contar con un sistema de navegación que sea visible y completo.
- El contenido se debe mostrar de manera clara, con títulos comprensibles y con enlaces hacia las secciones más usadas.

- Los elementos gráficos del sistema deben estar preparados para ayudar a cumplir los objetivos del sitio y no sólo como adornos utilizados para rellenar el espacio existente.



Implementación

Framework

Los frameworks de desarrollo nos entregan una estructura, armazón o marco de trabajo para desarrollar nuestros proyectos. En otras palabras, nos entregan una serie de bibliotecas de funciones que junto a convenciones comunes, se organizan bajo una estructura sobre la cual desarrollamos nuestros proyectos. Entre las ventajas de usar frameworks, podemos encontrar las siguientes:

- Facilidad para integrar a otras personas a tus proyectos, ya que se comparten convenciones de desarrollo comunes.
- No tenemos que preocuparnos de mantener actualizadas muchas de las funcionalidades de nuestro sistema de información.
- De esta forma evitamos reinventar la rueda, ya que aprovechamos los componentes existentes con la finalidad de ahorrar tiempo de desarrollo.

En este caso utilizaré el framework LARAVEL, es uno de los frameworks de código abierto más fáciles de asimilar para PHP. Es simple, muy potente y tiene una interfaz elegante y divertida de usar. Fue creado en 2011 y tiene una gran influencia de frameworks como Ruby on Rails, Sinatra y ASP.NET MVC.

Versiones

- Php: 7.4.3
- Laravel: 8

Además este proyecto estará alojado en goormIDE, un potente servicio IDE en la nube que nos servirá para maximizar nuestra productividad. Una de las características principales es que al estar en la nube podremos acceder a nuestra web de una manera rápida y sencilla, donde cada web que tengamos alojada tendrá una URL específica

Validación y pruebas

Durante las fases anteriores de análisis, diseño e implementación hemos intentado construir un sistema software partiendo de un concepto abstracto para llegar a una implementación real. Esta nueva fase tiene como finalidad garantizar la calidad del software creando una serie de casos de pruebas para descubrir errores en el sistema construido.

Objetivo

El objetivo principal de la ejecución de las pruebas se dirige a:

- Descubrir tantos errores como sea posible.
- Notificar acerca de los riesgos percibidos del Proyecto.
- Identificar errores funcionales de la aplicación, enmarcadas en grados de usabilidad ya definidos.
- Evaluar la calidad del producto y señalar un indicador de aceptación del mismo
- Evaluar la calidad técnica del producto y resolver los errores identificados en las pruebas de tipo técnico.
- Cumplir con los requerimientos específicos del cliente, en cuanto a la ejecución de las pruebas.

Tipos

De acuerdo con el enfoque de desarrollo de nuestro sistema software y con la finalidad de garantizar el cumplimiento de los requisitos planteados, hemos escogido un plan de pruebas basado en la metodología de Rational Unified Process (RUP).

A continuación detallamos cada una de las pruebas que llevaremos a cabo sobre el sistema de información:

Tipo de prueba	Descripción
Pruebas de IU codificadas	Prueba la interfaz de usuario realizando acciones de IU.
Pruebas unitarias	Prueba el código en el nivel de método.
Pruebas de carga	Prueba el rendimiento y el esfuerzo de una aplicación mediante pruebas unitarias, pruebas de rendimiento web o pruebas de IU codificadas.
Pruebas de rendimiento web	Se usa como parte de las pruebas de carga para probar las respuestas del servidor para aplicaciones web enviando y recibiendo solicitudes http.
Pruebas genéricas	Prueba la funcionalidad usando llamadas API o herramientas de línea de comandos para la aplicación en pruebas.

Estas pruebas consistirán en la práctica de introducir datos de prueba para comprobar la funcionalidad del sistema y que todas las gestiones de errores sean correctas, este tipo de pruebas las realizaré antes de entregarla al cliente.

Una vez todo esté funcionando, la aplicación se le entregará al cliente donde pasará unos días con él comprobando el funcionamiento de la aplicación ya de cara al público y con datos reales, con el fin de que el cliente sepa utilizar la web y resolverle las posibles dudas que le surjan.

Mantenimiento

El mantenimiento de un proyecto es una parte muy importante del mismo ya que debe presentar una imagen profesional, sin errores y constantemente actualizada, para ello proporcionaremos en este proyecto un mantenimiento evolutivo y correctivo. donde actualizaremos la web cada vez que exista una nueva mejora disponible y corregiremos posibles errores que le surjan al cliente.

Mantenimiento correctivo

- Un mantenimiento correctivo sucede cuando existe un fallo que es necesario atender para garantizar la seguridad, rendimiento o funcionalidad del sitio WEB,
Es un mantenimiento más complejo y en ocasiones puede hacer que tengamos que colocar la web en mantenimiento por un periodo de tiempo dependiendo de las modificaciones a realizar.

Mantenimiento evolutivo

- Un mantenimiento evolutivo es aquel en el que modificamos algo que funcionaba o era correcto, con el fin de aumentar, disminuir o cambiar las funcionalidades del sistema, ya sea por las necesidades del usuario o por otras razones como, por ejemplo, cambios normativos.

Este mantenimiento nos llevará unas 12 horas durante los primeros 6 meses, donde incluimos las actualizaciones, copias de seguridad y posibles errores, en caso de que estas horas de mantenimiento se amplíen por culpa nuestra, el coste correrá a nuestro cargo.

Durante los primeros 6 meses el precio de este ya está incluido en el precio de la web, una vez superado este período ofrecemos unas tarifas de mantenimiento que parten desde los 30 euros mensuales hasta la tarifa premium de 80 euros/mes.

