
Especificación de requisitos de software

**Proyecto:
Librería Virtual
Grupo 47**

Instrucciones para el uso de este formato

Este formato es una plantilla tipo para documentos de requisitos del software.

Está basado y es conforme con el estándar IEEE Std 830-1998.

Los textos en color azul son indicaciones que deben eliminarse y, en su caso, sustituirse por los contenidos descritos en cada apartado.

Ficha del documento

Fecha	Revisión	Autor	Verificado dep. Calidad.
SEP 2022	01	Juan Manuel Sanchez Fin Fiorella Cristina Melgarejo Franco Agustin Trivini De Elejalde Andrea Inés Reyna Alejandro Celen Gonzalo Jesús Ezequiel Cuello Nicolas Angel Emiliano Gaitan Pamela Suarez	

Fecha	Revisión	Autor	Verificado dep. Calidad.
APR 2023	02	Juan Manuel Sanchez Fin Fiorella Cristina Melgarejo Andrea Inés Reyna	

Fecha	Revisión	Autor	Verificado dep. Calidad.
MAY 2023	03	Juan Manuel Sanchez Fin Fiorella Cristina Melgarejo Andrea Inés Reyna	

Fecha	Revisión	Autor	Verificado dep. Calidad.
JUN 2023	04	Juan Manuel Sanchez Fin Fiorella Cristina Melgarejo Andrea Inés Reyna	

Contenido

<u>FICHA DEL DOCUMENTO</u>	3
<u>CONTENIDO</u>	4
1 <u>INTRODUCCIÓN</u>	6
<u>1.1 Propósito</u>	6
<u>1.2 Alcance</u>	6
<u>1.3 Personal involucrado</u>	6
<u>1.4 Definiciones, acrónimos y abreviaturas</u>	7
<u>1.5 Referencias</u>	7
<u>1.6 Resumen</u>	7
2 <u>DESCRIPCIÓN GENERAL</u>	7
<u>2.1 Perspectiva del producto</u>	7
<u>2.3 Características de los usuarios</u>	8
<u>2.4 Restricciones</u>	9
3 <u>REQUISITOS ESPECÍFICOS</u>	9
<u>3.2 Product Backlog</u>	17
<u>3.2.1 Product Backlog</u>	17
<u>3.3 Sprints</u>	19
<u>3.3.1 Sprint 1</u>	19
<u>3.3.2 Spint 2</u>	20

1 Introducción

Nuestro proyecto se basa en un portal de gestión de venta de libros online que muestra los libros por título o portada, como también permite navegar libros por categorías, autores o editoriales. Contiene también un buscador avanzado con el uso de filtros, una sección de novedades, de más vendidos, de redes sociales y un formulario de contacto. De esta manera brindamos un portal claro y de fácil acceso para todo tipo de usuarios.

1.1 Propósito

El presente documento tiene como propósito definir las especificaciones funcionales, no funcionales para desarrollar un sistema de información web que permita a una librería gestionar la venta y envío de libros online, y a una persona encontrar un libro deseado, comprarlo y recibirlo en su domicilio.

1.2 Alcance

La Librería Virtual es un sistema que se encarga de digitalizar el negocio de las librerías físicas, facilitando el acceso a los usuarios para la compra online de libros, entregando información de los libros, novedades, ofertas y descuentos.

1.3 Personal involucrado

Nombre	Juan Manuel Sanchez Fin
Rol	Product Owner
Categoría Profesional	Full Stack Developer
Responsabilidad	Nexo entre Stakeholders y DevTeam
Información de contacto	jmsanchezfin@gmail.com

Nombre	Andrea Inés Reyna
Rol	ScrumMaster
Categoría Profesional	Full Stack Developer
Responsabilidad	Líder del equipo de desarrollo Ejecuta los sprints
Información de contacto	andree.reyna@gmail.com

Nombre	Fiorella Cristina Melgarejo
Rol	Developer
Categoría Profesional	Full Stack Developer
Responsabilidad	Diseño Frontend
Información de contacto	fiore9713@gmail.com

Nombre	Gonzalo Jesús Ezequiel Cuello
Rol	QA Testing
Categoría Profesional	Full Stack Developer
Responsabilidad	Analista de calidad del producto
Información de contacto	ecuello19@gmail.com

Nombre	
--------	--

Rol	
Categoría Profesional	
Responsabilidad	
Información de contacto	

Nombre	
Rol	
Categoría Profesional	
Responsabilidad	
Información de contacto	

1.4 Definiciones, acrónimos y abreviaturas

Nombre	Descripción
Usuario	Persona que usará el sistema para gestionar procesos
SIS-I	Sistema de Información Web para la Gestión de Procesos Administrativos y Académicos
ERS	Especificación de Requisitos Software
RF	Requerimiento Funcional
RNF	Requerimiento No Funcional
FTP	Protocolo de Transferencia de Archivos
PWA	Aplicación Web Progresiva
Moodle	Aula Virtual

1.5 Referencias

Título del Documento	Referencia
Standard IEEE 830 - 1998	IEEE

1.6 Resumen

Este proyecto proporcionará un sistema que se podrá implementar de una manera rápida y sencilla, pudiendo así transformar un negocio físico en una tienda online.

Los usuarios encontrarán en esta plataforma una manera sencilla de comprar online sus libros y recibirlos en la comodidad de su hogar. En tanto que las empresas logran una rápida implementación, siendo compatible tanto con una pequeña tienda como en una gran cadena.

2 Descripción general

2.1 Perspectiva del producto

El producto diseñado está hecho para trabajar en entornos WEB, lo que permitirá a todo tipo de usuario iniciar una búsqueda sobre un libro cualquiera que sea, desplegar su información y también poder adquirir dicho libro y a que precio. También se podrá utilizar cómo una PWA.

2.2 Características de los usuarios

Tipo de usuario	Vendedor
Formación	Basica
Actividades	Control y manejo del sistema en general. Conocimiento del producto que publica, manejo básico de computación.

Tipo de usuario	Visitante
Formación	Manejo de entornos web
Actividades	Busca libros e indaga información. Compra, paga y recibe en un domicilio la compra.

2.3 Restricciones

- Esta interfaz debe ser utilizada con Internet.
- Podrá ser accedido desde computadoras, cómo también desde celulares y tablets (responsive design)
- Tecnologías a utilizar: HTML5 y CSS3, Bootstrap, Python, JavaScript, MySQL, Angular, Django
- El sistema deberá tener un diseño e implementación sencilla, independiente de la plataforma o del lenguaje de programación.
- Metodología de desarrollo: Metodología Agile y Scrum

3 Requisitos específicos

- La web debe contar con un panel para que el administrador pueda subir los libros (con su título y subtítulo, autor, editorial, portada, categoría, precio, reseña, sección de comentarios de usuario, y detalles cómo cantidad de páginas, peso, formato, año de edición, idioma, ISBN)
- un formulario para que el usuario se registre (correo electrónico, nombre y apellido, contraseña)
- un panel para que el usuario registre datos adicionales (fecha de nacimiento, sexo, documento de identidad), direcciones de entrega (provincia, localidad, nombre y apellido de destinatario, calle, número, piso, depto, prefijo telefonico, número de

telefono, código postal), dirección de facturación provincia, localidad, nombre y apellido de destinatario, calle, número, piso, depto, prefijo telefonico, número de telefono, código postal).

- una barra de búsqueda rápida
- una sección de novedades
- una sección de más vendidos
- un formulario de contacto
- una sección de redes sociales
- un carrito de compras
- una pasarela de pagos
- un panel para que el usuario realice el seguimiento de las compras
- un panel para que el administrador realice el seguimiento de los cobros y las ventas
- una sección para que el administrador agregue las facturas y los números de tracking de los envíos

Sprints 2022

N° de sprint	00
Sprint Backlog	<ol style="list-style-type: none">1.- Requerimientos IEEE380<ul style="list-style-type: none">- subirlo en una carpeta de GitHub en la rama main2.- Historias de los Usuarios y Tareas Dependientes de las Historias<ul style="list-style-type: none">- #US01, #US02, etc- #TK02, #TK02, etc- Idear Mapa del sitio3.- Crear Project y pasarlas al product backlog4.- Agregar a la Wiki los registros del Sprint

Proyecto Integrador

Especificación de requisitos de software

Responsabilidades	Identificamos y definimos los requerimientos Planteamos las Historias de Usuarios y las Tareas Creamos el Product Backlog Documentamos en la Wiki las meetings y los Sprint Completamos y subimos el documento IEEE380 Retrospectiva del Sprint
Calendario	Fecha Inicio = 17/09/2022 - Fecha de Fin = 03/10/2022
Inconvenientes: Es difícil trabajar bajo Scrum informalmente. Primera vez para todos que debemos trabajar en un proyecto de forma grupal, con Scrum y documentando el proceso. Dejamos las Task / Issues para el próximo Sprint	

N° de sprint	01
Sprint Backlog	1.- Estructura HTML, semántica y estilos CSS. Links funcionales. Diseño Responsive y utilización de Bootstrap. Presentación del proyecto, quienes somos y contacto. 2.- Modelo de ER, Relacional, Casos de Uso y Clases. 3.- Abstracción y modularización patrón MVC.
Responsabilidades	Creamos la estructura de HTML con semántica, estilos CSS, usando Bootstrap y diseño Responsive. Creamos los modelos de entidad relación y relacionales de la base de datos. Creamos los modelos en UML de Casos de Uso y Clases. Generamos la estructura MVC modularizado y abstrayendo las clases.
Calendario	Fecha Inicio = 03/10/2022 - Fecha de Fin = 17/10/2022
Inconvenientes: No tuvimos respuesta alguna del tutor en nuestro foro privado. Hace falta consignas más claras y precisas, y sobre todo respuestas más rápidas. Los sprint son cortos y las dudas no se aclaran. Se dijo que sólo se debía crear tres html y después otra cosa.	

Proyecto Integrador
Especificación de requisitos de software

N° de sprint	02
Sprint Backlog	1.- Sitio funcional en hosting remoto de Sitio Institucional en WordPress según lo solicitado en SGC. 2.- Funcionalidad con JavaScript en formularios. Validaciones. 3.- Sitio disponible online. 4.- Script de la BD 5.- Creación de CRUD 6.- Documentación
Responsabilidades	Creamos el sitio institucional en hosting remoto, utilizando WordPress. Creamos funcionalidades en JavaScript, validando datos y accionando en eventos. Puesta online del sitio usando GitPages. Realizamos el script para la creación de la base de datos. Creamos un CRUD de login y un CRUD de registro Documentamos todo el proyecto.
Calendario	Fecha Inicio = 17/10/2022 - Fecha de Fin = 14/11/2022
Inconvenientes: No tuvimos respuesta alguna del tutor a cargo, ni tampoco devolución de los sprints, así que trabajamos solos con los resultados que nos parecieron correctos. Es muy difícil lograr el compromiso de tantos participantes sin tener un tutor a cargo. No quedó del todo claro usar el patrón MVC sin un framework; no pudimos hacerlo funcional. Lo dejamos ejemplificado de todas maneras.	

Sprints 2023

N° de sprint	00
Sprint Backlog	1.- Requerimientos IEEE380 - Actualizado v2 - subirlo en una carpeta de GitHub en la rama main 2.- Historias de los Usuarios y Tareas Dependientes de las Historias - #US01, #US02, etc - #TK02, #TK02, etc 3.- Crear Project y pasarlas al product backlog 4.- Agregar a la Wiki los registros del Sprint 5.- Revisar la estructura web Semántica, responsive (RWD)
Responsabilidades	Identificamos y definimos los requerimientos Planteamos las Historias de Usuarios y las Tareas Creamos el Product Backlog Documentamos en la Wiki las meetings y los Sprint Completamos y subimos el documento IEEE380 v2 Retrospectiva del Sprint
Calendario	Fecha Inicio = 10/04/2023 - Fecha de Fin = 21/04/2023
Inconvenientes:	Perdimos un compañero del grupo. Se les dificultó a algunos la instalación de Node.js para utilizar Typescript. No todos pudieron asistir a las clases pactadas.

N° de sprint	01
Sprint Backlog	1.- Convertir los archivos .html del proyecto previo, en una SPA en Angular con módulos (Layouts/Shared y Pages son un ejemplo) con sus components correspondientes. (HOME, REGISTRO, LOGIN, DASHBOARD) 2.- Crear módulos y componentes para la tienda virtual o ecommerce. (PRODUCTO o SERVICIOS, PRODUCTO INDIVIDUAL, COMPRA). 3.- Incorporar la navegabilidad de la aplicación mediante Routing con Angular. 4.- Subir en la carpeta documentación las tablas actualizadas de la base de datos: -Diseño conceptual al que ya hemos visto como DER: Diagrama Entidad-Relación. -Diseño Lógico que consta de la Normalización y el Modelo Relacional. -Diseño Físico que es el script legible de la DB en MySQL. -Crear un Diagrama de Clases y Casos de Uso para facilitar el modelado en POO.
Responsabilidades	Dejamos funcional el MVP de la web con Angular con sus correspondientes módulos y componentes. Incorporamos navegabilidad de la app mediante routing con Angular. Actualizamos documentación de los diagramas de diseño conceptual DER y Modelo Relacional. Incorporamos el nuevo script de SQL para la DB. Nos quedó pendiente el diagrama de clases.
Calendario	Fecha Inicio = 24/04/2023 - Fecha de Fin = 05/05/2023
Inconvenientes:	Se nos bajó otro compañero. No todos pudieron dejar funcionando Angular ni tenerlo del todo claro. Hace falta algo más de material de apoyo / videos.

Proyecto Integrador

Especificación de requisitos de software

N° de sprint	02
Sprint Backlog	<p>Crear servicios que consuman un API Rest de prueba utilizando el servicio de HttpClient.</p> <p>Injectar a los componentes los correspondientes servicios a fin de que los componentes puedan acceder a los datos y mostrarlos en pantalla. Incluye enlace de datos (Data binding).</p> <p>Incorporar la programación reactiva (asíncronica) mediante observables para la comunicación entre servicios y componentes.</p> <p>Establecer el modelo de base de datos del primer sprint en Django usando clases.</p> <p>Establecer el sistema de usuarios en Django-Admin (registro, login y roles)</p> <p>Crear un ABM de productos/servicios utilizando el panel de control de Django.</p>
Responsabilidades	<p>Actualizamos el pendiente del diagrama de clases.</p> <p>Creamos un servicio que consume una API Rest de prueba, para dejar funcionando el home (listado de productos disponibles que toma de un JSON con datos cómo los links, las imágenes, los títulos, etc) y también el detalle de los productos (otro servicio).</p> <p>Establecimos la nueva base de datos, esta vez creando con un modelo desde Django.</p> <p>Sistema de login para Django funcionando, al igual que el CRUD de todas las tablas de la DB.</p>
Calendario	Fecha Inicio = 08/05/2023 - Fecha de Fin = 19/05/2023
Inconvenientes: Sprint complicado ya que muchos compañeros no pudieron participar. Hace falta algo más de material de apoyo / videos sobre todo de la parte de programación reactiva, que es algo totalmente nuevo.	

N° de sprint	03
Sprint Backlog	<p>Incorporar formularios reactivos de Registro y Login para la autenticación y formularios para la manipulación (Alta) de servicios/productos/suscripción, según el caso con sus respectivas validaciones.</p> <p>Conexión con el backend para el registro e inicio de sesión.</p> <p>Conexión con backend para almacenar los datos de los productos/servicios y detalles de producto/servicios en la DB a través del modelo en Django.</p> <p>Opcional: Agregar rutas protegidas a la navegación en base al usuario autenticado/autorizado. Ej. para acceder al tablero de control o bien, para acceder al registro de productos/servicios.</p> <p>Definición de la Api Rest y los endpoints necesarios para las funcionalidades de:</p> <p>Registro de usuarios.</p> <p>Login de usuarios.</p> <p>Registro / Alta de al menos 3 productos/servicios/suscripción (solo desde el rol Administrador, mediante su panel de control).</p> <p>Tablero de control (Gestión) o Dashboard con módulos planteados en la maquetación del FrontEnd según proyecto:</p> <p>El rol ADMIN pueda ver la lista de usuarios registrados y el listado de los productos/servicios/suscripción según corresponda.</p>

Proyecto Integrador

Especificación de requisitos de software

	<p>El rol USUARIO registrado solo que pueda ver su Dashboard para editar su perfil.</p> <p>Los datos serán consumidos desde la Base de datos MySQL.</p>
Responsabilidades	<p>Incorporamos formularios reactivos de Registro y Login para la autenticación y formularios para la manipulación (Alta) de servicios/productos/suscripción, según el caso con sus respectivas validaciones. Logramos conexión con el backend para el registro e inicio de sesión y para almacenar datos de los productos.</p> <p>Definimos la Api Rest y los endpoints necesarios para las funcionalidades de: Registro de usuarios, Login de usuarios, Registro / Alta de productos (solo desde el rol Administrador, mediante su panel de control). Tablero de control (Gestión) o Dashboard con módulos planteados en la maquetación del FrontEnd según proyecto:</p> <p>El rol ADMIN puede ver la lista de usuarios registrados y el listado de los productos/servicios/suscripción según corresponda.</p> <p>El rol USUARIO registrado solo que pueda ver su Dashboard para editar su perfil. Todos los datos son consumidos desde la DB MySQL.</p>
Calendario	Fecha Inicio = 22/05/2023 - Fecha de Fin = 04/06/2023
Inconvenientes: Sprint complicado ya que muchos compañeros no pudieron participar, sólo pudimos aportar tres de nosotros. Queda pendiente las rutas protegidas de los usuarios si lo suben a la plataforma.	

N° de sprint	04
Sprint Backlog	<p>Tablero de control para usuarios: Mostrar catálogo (mínimo 3 productos o servicios o suscripción). Seleccionar para comprar. Simular Pago con API Mercado Pago o similar</p> <p>Tablero de control administrador CRUD básico funcional de producto y cliente.</p> <p>Actualizar documentación IEEE830 - tablero Kanban (project) y Wiki.</p> <p>Actualizado todo el proyecto en GitHub como respaldo</p> <p>Demo final en video con participación de todo el equipo describiendo todas las funcionalidades.</p>
Responsabilidades	<p>Mostramos el catálogo de productos. Login y registro de usuario.</p> <p>Simulamos agregar al carrito y procesar la orden con una API Rest propia. Actualizamos documentación y diagramas. Completamos Wiki.</p> <p>Adjuntamos video con feedback de los integrantes.</p>
Calendario	Fecha Inicio = 05/06/2023 - Fecha de Fin = 18/06/2023
Inconvenientes: No recibimos ninguna devolución de ningún sprint, por lo tanto no sabemos si estamos en el camino correcto o necesitamos alguna verificación. Tuvimos muchos problemas para conectar el frontend con el backend. Se nos hizo imposible lograr una conexión con la API de mercadopago, pero hicimos el ejemplo de mercadobricks. Pudimos presentar un MVP: El usuario se registra, loguea, agrega artículos al carrito y procesa la orden. El backend registra toda la operación, generando un Pedido con su Detalle de Pedido, una Orden con su Pago, y gestionando el stock.	