
Especificación de requisitos de software

Proyecto:
Web de animales - Ecommerce

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.
2023			

Contenido

<u>FICHA DEL DOCUMENTO</u>	3
<u>CONTENIDO</u>	4
<u>1 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
<u>2 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
<u>3 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

Este documento es una Especificación de Requisitos Software (ERS) para el Sistema de información para la gestión de procesos y control basándose en las directrices dadas por el estándar IEEE para IRS ANSI/IEEE 830, 1998. Desarrollando formalmente el documento necesario para fomentar el aprendizaje, desarrollo y organización del grupo para poder crear la página web ("WebDeAnimales") que sea innovadora y funcional enriqueciendo de experiencia a nosotros, sus desarrolladores.

1.1 Propósito

El presente documento tiene como propósito dar a conocer el funcionamiento general o, más específicamente, definir las especificaciones funcionales, para el desarrollo de un sistema de información web institucional destinado al proyecto Web Animal que está dirigido tanto a los administrador del negocio (y sus respectivos empleados), como a los clientes del mismo.

1.2 Alcance

Esta especificación de requisitos está dirigida para el cliente del sitio "Web de animales"

1.3 Personal involucrado

Nombre	Montivero Axel Ezequiel
Rol	Desarrollador
Categoría Profesional	Estudiante
Responsabilidad	Diseño y programación del sistema
Información de contacto	ezmant14@gmail.com

Nombre	Amato Fabiana Jazmín
Rol	Desarrollador
Categoría Profesional	Estudiante
Responsabilidad	Diseño y programación del sistema
Información de contacto	proyectos.amato@gmail.com

Nombre	Gómez Juan Ezequiel
Rol	Desarrollador
Categoría Profesional	Estudiante
Responsabilidad	Diseño y programación del sistema
Información de contacto	juanezequielgomez95@gmail.com

Nombre	Reyes Melanie
Rol	Desarrollador
Categoría Profesional	Estudiante
Responsabilidad	Diseño y programación del sistema
Información de contacto	melanie.reyes.tecnicos2017@gmail.com

Nombre	Alejandro Moreno
Rol	Desarrollador
Categoría Profesional	Estudiante
Responsabilidad	Diseño y programación del sistema
Información de contacto	morenocabreroalejandro@gmail.com

Nombre	Nicolas Mattos
Rol	Desarrollador
Categoría Profesional	Estudiante
Responsabilidad	Diseño y programación del sistema
Información de contacto	nicomattos2014@gmail.com

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
Moodle	Aula Virtual

1.5 Referencias

Título del Documento	Referencia
Standard IEEE 830 - 1998	IEEE

1.6 Resumen

El presente documento consta de 3 secciones:

- La primera sección describe el producto que se está desarrollando, su propósito, audiencia objetivo, uso previsto y alcance.
- La segunda sección proporciona más información sobre las necesidades de los usuarios y los factores que podrían impedir que se cumplan los requisitos establecidos en la SRS.
- La última sección principal está dedicada a los requisitos específicos, tanto funcionales como no funcionales.

2 Descripción general

2.1 Perspectiva del producto

El sistema Web de animales será un producto diseñado para trabajar en entornos WEB, lo que permitirá su utilización de forma rápida y eficaz, además cuenta con las funcionalidades para que los usuarios puedan adoptar, hacer seguimiento, donar y contactar veterinarios. Para este proyecto, nos enfocaremos en el sistema de adopción.

2.2 Características de los usuarios

Tipo de usuario	Entidad protectora de animales
Formación	Cuidado y atención de animales en situación de vulnerabilidad
Actividades	Brindar información, fotos y/o videos sobre los animales rescatados. Realizar seguimiento de reportes de animales extraviados.

Tipo de usuario	Visitante
Formación	Manejo de entornos web
Actividades	Realizar seguimiento de los animales rescatados para adoptar y/o realizar una donación a la entidad. Reportar animales perdidos, abandonados o maltratados. Contactar veterinarios.

2.3 Restricciones

- Las restricciones del desarrollo del sistema son las siguientes:
 - Utilización de repositorio con control de versiones GitHub
 - Documentación del código con Microsoft Word
 - Utilización exclusiva del lenguaje de programación Python 3.
 - Utilización de JavaScript, Angular y Django.
 - Acceso a una base de datos.
- Para el acceso al sistema se contemplará dos tipos de usuarios:
 - El administrador tendrá acceso a todas las funcionalidades.
 - El empleado que solo tendrá acceso limitado al sistema según las indicaciones del administrador.

3 Requisitos específicos

3.1 Requisitos Funcionales

1. Reportar animales perdidos.
2. Adoptar animales que han sido rescatados.
3. Hacer donaciones.
4. Agradecer a los donadores.
5. Ver animales rescatados.
6. Contacto con Veterinarios
7. Lugares PetFriendly para pasear.
8. Creación de usuarios para poder realizar las funciones principales del sistema.

3.2 Requisitos No Funcionales

1. Hosting para alojar el sitio Web
2. Dominio
3. Rendimiento.
4. Servicio 24hs disponible.
5. Protección contra ataques.

Product Backlog de este año (2023)

#US00 Como asociación quiero actualizar las tecnologías de acuerdo a las nuevas tendencias en angular
#US01 Como asociación quiero contar con una tienda para poder ofrecer adopción
#US02 Como usuario quiero crearme una cuenta en el sitio
#US03 Como usuario quiero poder ver una lista de animales para adoptar
#US04 Como usuario quiero ingresar al carrito para poder adoptar
#US05 Como usuario quiero darle un nombre a la mascota que adoptar
#US06 Como usuario quiero poder elegir el tipo de retiro de la mascota
#TK07 Adaptar el Mapa de sitio a los nuevos requerimientos del sitio

Tareas:

#TK00 Actualizar documento IEEE830
#TK01 Mejorar diseño de home
#TK02 Mejorar diseño de nosotros
#TK03 Actualizar planilla del proyecto
#TK04 Actualizar documentación de la Wiki
#TK05 Adaptar el Modelo Relacional a los nuevos requerimientos del sitio

- #TK06 Adaptar el DER a los nuevos requerimientos del sitio
- #TK07 Adaptar el Mapa de sitio a los nuevos requerimientos del sitio
- #TK08 Adaptar home a angular
- #TK09 Adaptar Nosotros a angular
- #TK10 Adaptar Registrarse a angular
- #TK11 Adaptar Mi cuenta a angular
- #TK12 Adaptar Loguearse a angular
- #TK13 Adaptar Reportar a angular
- #TK14 Adaptar Seguimiento a angular
- #TK15 Modificar página de adopción de acuerdo al ecommerce en angular
- #TK16 Crear página de carrito
- #TK17 Crear página de retiro y de nombrar animal.
- #TK18 Adaptar Veterinarios a angular
- #TK19 Adaptar Donar a angular
- #TK20 Adaptar Agradecimientos a angular
- #TK21 Hacer la conexión según el MVC en Django
- #TK22 Adaptar PetFriendly a angular
- #TK23 Subir archivo SQL
- #TK24 Crear pagina de detalles del producto en angular
- #TK25 Adaptar el mapa de sitio a los nuevos requerimientos del sitio
- #TK26 Adaptar Registrarse a Angular
- #TK27 Adaptar Mi cuenta a angular
- #TK28 Crear servicio para el componente Veterinarios
- #TK29 Crear servicio para el componente Agradecimientos
- #TK30 Crear servicios para el componente PetFriendly
- #TK31 Crear servicio para el componente Contacto en angular
- #TK32 Crear ABM (CRUD) de Donaciones con Django.
- #TK33 Crear servicio para el componente lista-adopcion en angular
- #TK34 Crear ABM (CRUD) de Veterinarios con Django
- #TK35 Crear ABM (CRUD) de Usuario-Animales con Django
- #TK36 Crear ABM(CRUD) de reportes con Django
- #TK37 corregir validaciones del formulario en "Reportar"
- #TK38 Implementar Validaciones en componente Contacto
- #TK39 Crear ABM (CRUD) de Contacto con Django
- #TK40 Crear ABM (CRUD) de Refugios con Django
- #TK41 Modificar componente "LugaresPetFriendly" por "Refugios" en Angular
- #TK42 Corregir validaciones del formulario en "Registrarse"
- #TK43 Crear ABM (CRUD) de Animales con Django
- #TK44 Crear ABM (CRUD) de TipoAnimales con Django
- #TK45 Crear API de Login con Django
- #TK46 Crear API de Registro con Django
- #TK47 Crear API de Animales con Django
- #TK48 Crear servicio para componente miCuenta
- #TK49 Crear API de Contacto con Django
- #TK50 Renombrar el componente Agradecimientos por "Donaciones"
- #TK51 Conectar a una API el componente donaciones

#TK52 Crear API de Agregar Donaciones con Django
#TK53 Crear API de Mostrar Donaciones con Django

Sprints.

N° de sprint	00
Sprint Backlog	<p>#TK00 Actualizar documento IEEE830</p> <p>#TK01 Mejorar diseño de home</p> <p>#TK02 Mejorar diseño de nosotros</p> <p>#TK03 Actualizar planilla del proyecto</p> <p>#TK04 Actualizar documentación de la Wiki</p> <p>#TK05 Adaptar el Modelo Relacional a los nuevos requerimientos del sitio</p> <p>#TK06 Adaptar el DER a los nuevos requerimientos del sitio</p> <p>#TK07 Adaptar el Mapa de sitio a los nuevos requerimientos del sitio</p>
Responsabilidades	<p><i>La “consultora ISPC”, solicita en esta oportunidad, incorporar un módulo de e-commerce para comercializar sus productos y/o servicios online.</i></p> <p>Antes de comenzar</p> <ul style="list-style-type: none"> Definir Scrum Master inicial y registrarse cada integrante faltante dentro del repositorio a trabajar. Revisar documentación IEEE830 si está completa y retomar desde allí para esta nueva etapa seguir documentando los avances. Revisar la estructura web Semántica, responsive (RWD) <p><i>Pasos a seguir de acuerdo a las Fases del Ciclo de Vida de un Software:</i></p> <p>ANÁLISIS</p> <ol style="list-style-type: none"> Definir requerimientos para el nuevo módulo a desarrollar e-commerce (colocarlos en el <i>Product Backlog del Project</i>), a su vez revisar si han cumplimentado todos los requerimientos previos, realizando mejoras del mismo Plantear Historias de Usuarios y Tareas dependientes de las US para incorporarlas en el repositorio remoto

Proyecto Integrador
Especificación de requisitos de software

	<p>GitHub. (<i>Issues y Milestones</i>) - Tener en cuenta la redacción adecuada para las US y nomenclatura, ej “#US01 Como usuario quiero ingresar al carrito para poder comprar”</p> <p>3. Definir tareas dentro de las Historias de Usuario (GITHUB) ej dentro de las ISSUES #TK01 <i>revisar IEEE830.</i></p>
Calendario	<i>Fecha Inicio = 10/04/2023 - Fecha de Fin = 21/04/2023</i>
Inconvenientes:	

N° de sprint	01
Sprint Backlog	<p>#TK08 Adaptar home a angular</p> <p>#TK09 Adaptar Nosotros a angular</p> <p>#TK10 Adaptar Registrarse a angular</p> <p>#TK11 Adaptar Mi cuenta a angular</p> <p>#TK12 Adaptar Loguearse a angular</p> <p>#TK13 Adaptar Reportar a angular</p> <p>#TK14 Adaptar Seguimiento a angular</p> <p>#TK15 Modificar página de adopción de acuerdo al ecommerce en angular</p> <p>#TK16 Crear página de carrito</p> <p>#TK17 Crear página de retiro y de nombrar animal.</p> <p>#TK18 Adaptar Veterinarios a angular</p> <p>#TK19 Adaptar Donar a angular</p> <p>#TK20 Adaptar Agradecimientos a angular</p> <p>#TK21 Hacer la conexión según el MVC en Django</p> <p>#TK22 Adaptar PetFriendly a angular</p> <p>#TK23 Subir archivo SQL</p> <p>#TK24 Crear pagina de detalles del producto en angular</p> <p>#TK25 Adaptar el mapa de sitio a los nuevos requerimientos del sitio</p> <p>#TK26 Adaptar Registrarse a Angular</p> <p>#TK27 Adaptar Mi cuenta a angular</p>

Responsabilidades	<p>Fecha Inicio = 24/04/2023 - Fecha de Fin = 07/05/2023</p> <p>Punto inicial de Implementación FRONTEND 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)</p> <p>Crear módulos y componentes para la tienda virtual o ecommerce. (PRODUCTO o SERVICIOS, PRODUCTO INDIVIDUAL, COMPRA).</p> <p>Incorporar la navegabilidad de la aplicación mediante Routing con Angular.</p> <p>Diseño del BACKEND Subir en la carpeta documentación las tablas actualizadas de la base de datos:</p> <p>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.</p> <p>GESTION DE PROYECTO Subir al repo grupal en GitHub, mediante comandos GIT desde el local, pueden crearla en una branch por desarrollador para que cada uno tenga una copia, o directamente crear una branch FEATURE, para actualizar los cambios que no se encuentran aún en condiciones de incorporarse a la branch DEVELOP.</p> <p>En GitHub, mediante la sección WIKI deben reflejar los avances individuales y de equipo en relación a Scrum (Ceremonias, roles, historias de usuarios, etc). Actualizar IEEE830.</p> <p>“FEATURE” son las branches en las que se trabaja la mayor parte del tiempo. Cuando no se utilizan más se borran.</p>
-------------------	---

Proyecto Integrador
Especificación de requisitos de software

	<p>"DEVELOP" es la branch donde se mantiene la versión de desarrollo más estable y reciente. La que es candidata para pasar a ser la principal (MAIN).</p> <p>La branch "MAIN" o "MASTER" es la que suele usarse para mantener las versiones estables (productivas) de nuestros proyectos. Listas para salir a producción.</p> <p>Estas son algunas CONVENCIONES más usadas para trabajar con branches en proyectos de desarrollo de software.</p>
Calendario	Fecha Inicio = 24/04/2023 - Fecha de Fin = 07/05/2023
Inconvenientes:	

N° de sprint	02
Sprint Backlog	<p>#US06 Como usuario quiero poder elegir el tipo de retiro de la mascota</p> <p>#TK28 Crear servicio para el componente Veterinarios</p> <p>#TK29 Crear servicio para el componente Agradecimientos</p> <p>#TK30 Crear servicios para el componente PetFriendly</p> <p>#TK31 Crear servicio para el componente Contacto en angular</p> <p>#TK32 Crear ABM (CRUD) de Donaciones con Django.</p> <p>#TK33 Crear servicio para el componente lista-adopcion en angular</p> <p>#TK34 Crear ABM (CRUD) de Veterinarios con Django</p> <p>#TK35 Crear ABM (CRUD) de Usuario-Animales con Django</p> <p>#TK36 Crear ABM(CRUD) de reportes con Django</p> <p>#TK40 Crear ABM (CRUD) de Refugios con Django</p>
Responsabilidades	<p>Frontend</p> <p>El goal de este sprint es construir un frontend dinámico. Por ello, se requiere crear servicios que provean los datos a los componentes. Para lograrlo, los servicios deben consumir una API Rest de prueba (Ej. https://www.npmjs.com/package/json-server).</p> <p>Se requiere:</p> <p>Crear servicios que consuman un API Rest de prueba. Para realizar las peticiones http, utilizar 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.</p> <p>Incluye enlace de datos (Data binding).</p> <p>Incorporar la programación reactiva (asincrónica) mediante observables para la comunicación entre servicios y componentes.</p> <p>Backend</p> <p>Establecer el modelo de base de datos del primer sprint en Django usando clases.</p>

Proyecto Integrador

Especificación de requisitos de software

	<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>
Calendario	<p>Apertura: lunes, 8 mayo 2023</p> <p>Cierre: domingo, 21 mayo 2023</p>
Inconvenientes:	

N° de sprint	03
Sprint Backlog	<p>#TK38 Implementar Validaciones en componente Contacto</p> <p>#TK39 Crear ABM (CRUD) de Contacto con Django</p> <p>#TK41 Modificar componente "LugaresPetFriendly" por "Refugios" en Angular</p> <p>#TK42 Corregir validaciones del formulario en "Registrarse"</p> <p>#TK43 Crear ABM (CRUD) de Animales con Django</p> <p>#TK44 Crear ABM (CRUD) de TipoAnimales con Django</p> <p>#TK45 Crear API de Login con Django</p> <p>#TK46 Crear API de Registro con Django</p> <p>#TK47 Crear API de Animales con Django</p> <p>#TK48 Crear servicio para componente miCuenta</p> <p>#TK49 Crear API de Contacto con Django</p>
Responsabilidades	<p>Frontend:</p> <ul style="list-style-type: none"> 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><i>Diseño de los formularios:</i></p> <ul style="list-style-type: none"> Deberán estar conformados por input types acorde al dato que se va a ingresar (https://developer.mozilla.org/es/docs/Web/HTML/Element/input) Deberán informar al usuario mediante mensajes personalizados (Validaciones) Deberán respetar los colores semánticos que propone Bootstrap, https://getbootstrap.com/docs/5.0/utilities/borders/ Por ej. <ol style="list-style-type: none"> <code></code> <code></code>

Proyecto Integrador

Especificación de requisitos de software

	<ul style="list-style-type: none"> • Conexión con el backend para el registro e inicio de sesión. • 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. • 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>Backend</p> <p>Definición de la Api Rest y los endpoints necesarios para las funcionalidades de:</p> <ul style="list-style-type: none"> • Registro de usuarios. • Login de usuarios. • Registro / Alta de al menos 3 productos/servicios/suscripción (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: <ul style="list-style-type: none"> ○ El rol ADMIN pueda ver la lista de usuarios registrados y el listado de los productos/servicios/suscripción según corresponda. ○ El rol USUARIO registrado solo que pueda ver su Dashboard para editar su perfil. • Los datos serán consumidos desde la Base de datos MySQL. <p><i>OPCIONAL:</i> Token validación del login con JWT.</p>
Calendario	Fecha Inicio = 22/05/2023 - Fecha de Fin = 04/06/2023
Inconvenientes:	

N° de sprint	04
Sprint Backlog	<p>#TK50 Renombrar el componente Agradecimientos por "Donaciones"</p> <p>#TK51 Conectar a una API el componente donaciones</p> <p>#TK52 Crear API de Agregar Donaciones con Django</p> <p>#TK53 Crear API de Mostrar Donaciones con Django</p> <p>#TK37 corregir validaciones del formulario en "Reportar"</p>
Responsabilidades	<p>Requerimientos Incremento del producto: Funcionalidades completas (front + back integrado)</p> <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 https://www.mercadopago.com.ar/developers/es/docs#online-payments</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. (quien no explique en el mismo alguna de las funcionalidades/código/gestión del proyecto, no será tenido en cuenta para calificación)</p> <p>OPCIONAL: Sitio publicado en Servidor</p>
Calendario	<p>Fecha Inicio = 05/06/2023 - Fecha de Fin = 18/06/2023</p>
Inconvenientes:	

1 Apéndice

1.1 Mapa del Sitio

