

Fecha de entrega

Profesor: *Nombre del profesor responsable*

Informática, Multimedia y Telecomunicación

Memoria del Proyecto Final de Máster

**Máster Universitario en Desarrollo de sitios y aplicaciones web**

Mesta:

Formulación de productos

Consultor: *Nombre del consultor responsable*

**Autor: Ángela Redondo Rodríguez**

  
Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](http://creativecommons.org/licenses/by-nc-nd/3.0/es/)

© (el autor/a)

Reservados todos los derechos. Está prohibido la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendidos la impresión, la reprografía, el microfilme, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante alquiler y préstamo, sin la autorización escrita del autor o de los límites que autorice la Ley de Propiedad Intelectual.

*Dedicatoria*

Resumen

Abstract

Convenciones

TITULO 1: VERDANA 18 VERSALITAS

Título 2: Verdana 14 regular

Normal: Arial 10 regular

*Palabras en otro idioma: Arial 10 cursiva*

Títulos de figuras y tablas: Arial 9 cursiva

Índice de contenidos

[CAPÍTULO I. Introducción 10](#_Toc99465788)

[1. Descripción 11](#_Toc99465789)

[2. Objetivos 12](#_Toc99465790)

[2.1. Principales 12](#_Toc99465791)

[2.2. Secundarios 12](#_Toc99465792)

[3. Metodología 13](#_Toc99465793)

[4. Planificación 14](#_Toc99465794)

[4.1. Diagrama de Gantt 14](#_Toc99465795)

[4.2. Hítos y entregas 16](#_Toc99465796)

[CAPÍTULO II. Definición y arquitectura 19](#_Toc99465797)

[1. Marco teórico 19](#_Toc99465798)

[1.1. Base teórica: formulación de un champú sólido 19](#_Toc99465799)

[2. Marco tecnológico 21](#_Toc99465800)

[2.1. API Rest 21](#_Toc99465801)

[2.2. Redux 21](#_Toc99465802)

[*2.3. Progressive Web Apps* 22](#_Toc99465803)

[3. Contenidos 23](#_Toc99465804)

[4. Diagramas de clases 26](#_Toc99465805)

[5. Arquitectura de la aplicación 27](#_Toc99465806)

[5.1. Backend 28](#_Toc99465807)

[5.2. Frontend 29](#_Toc99465808)

[5.3. Gestion de versión 29](#_Toc99465809)

[6. Plataformas de desarrollo 30](#_Toc99465810)

[6.1. Software 30](#_Toc99465811)

[6.2. Hardware 30](#_Toc99465812)

[CAPÍTULO III. Diseño y Usabilidad 31](#_Toc99465813)

[1. Perfiles de usuario 31](#_Toc99465814)

[2. Sitemap 33](#_Toc99465815)

[3. Prototipos 35](#_Toc99465816)

[3.1. Prototipos de baja calidad 35](#_Toc99465817)

[3.2. Prototipos de alta calidad 42](#_Toc99465818)

[4. Diagrama de flujo 52](#_Toc99465819)

[4.1. Autenticación de usuario 52](#_Toc99465820)

[4.2. Configuración de cuenta 53](#_Toc99465821)

[4.3. Accesos relativo a los componentes 54](#_Toc99465822)

[4.4. Accesos relativos a la recetas o formulaciones 54](#_Toc99465823)

[CAPÍTULO IV. Desarrollo de la aplicación 55](#_Toc99465824)

[1. Guiones **¡Error! Marcador no definido.**](#_Toc99465825)

[2. Proceso de desarrollo 55](#_Toc99465826)

[2.1. PEC 2 – Sprint 1 56](#_Toc99465827)

[3. APIs utilizadas 59](#_Toc99465828)

[4. Seguridad 60](#_Toc99465829)

[5. Tests 61](#_Toc99465830)

[6. Versiones de la aplicación 62](#_Toc99465831)

[7. Requisitos de instalación y uso 63](#_Toc99465832)

[8. Instrucciones de instalación 64](#_Toc99465833)

[9. Instrucciones de uso 67](#_Toc99465834)

[10. Bugs 68](#_Toc99465835)

[11. Proyección a futuro 69](#_Toc99465836)

[CAPÍTULO V. Presupuesto y mercado 70](#_Toc99465837)

[1. Presupuesto 70](#_Toc99465838)

[2. Análisis de mercado 70](#_Toc99465839)

[3. Marketing y ventas 71](#_Toc99465840)

[CAPÍTULO VI. Conclusiones 72](#_Toc99465841)

[Anexo I. Entregables del proyecto 73](#_Toc99465842)

[Anexo II. Prototipo de baja calidad 74](#_Toc99465843)

[Anexo III. Prototipos de alta calidad 85](#_Toc99465844)

[Anexo IV. Código fuente 101](#_Toc99465845)

[Anexo V. Librerías externas utilizadas 101](#_Toc99465846)

[Anexo VI. Captura de pantalla 102](#_Toc99465847)

[Anexo VII. Guía de usuario 102](#_Toc99465848)

[Anexo VIII. Libro de estilo 102](#_Toc99465849)

[Anexo IX. Resumen ejecutivo 102](#_Toc99465850)

[Anexo X. Glosario 103](#_Toc99465851)

[Anexo XI. Bibliografía 103](#_Toc99465852)

[Anexo XII. Vita 104](#_Toc99465853)

Figuras y tablas

Índice de figuras

[Figura 1 Metodología SCRUM. Ref: http://www.pmoinformatica.com/ 13](#_Toc97032190)

[Figura 2 Arquitectura cliente-servidor **¡Error! Marcador no definido.**](file:///C:\Users\Angela%20Redondo\Documents\Anxela\UOC\Memoria%20TFM.docx#_Toc97032191)

[Figura 3 Herramientas utilizadas en el desarrollo del proyecto **¡Error! Marcador no definido.**](file:///C:\Users\Angela%20Redondo\Documents\Anxela\UOC\Memoria%20TFM.docx#_Toc97032192)

[Figura 4 Diagrama de Gantt períodos de los sprints 14](file:///C:\Users\Angela%20Redondo\Documents\Anxela\UOC\Memoria%20TFM.docx#_Toc97032193)

[Figura 5 Diagrama de Gantt **¡Error! Marcador no definido.**](file:///C:\Users\Angela%20Redondo\Documents\Anxela\UOC\Memoria%20TFM.docx#_Toc97032194)

Índice de tablas

[Tabla 1 Contenidos de la aplicación web 24](#_Toc97032183)

[Tabla 2 Herramientas de software utilizadas en el proyecto 30](#_Toc97032184)

[Tabla 3 Herramientas de hardware utilizadas en el proyecto 30](#_Toc97032185)

[Tabla 4 Planificación objetivos sprint PEC 1 16](#_Toc97032186)

[Tabla 5 Planificación objetivos sprint PEC 2 16](#_Toc97032187)

[Tabla 6 Planificación objetivos sprint PEC 3 17](#_Toc97032188)

[Tabla 7 Planificación objetivos sprint Entrega final 17](#_Toc97032189)

# Introducción

*Mesta* nace con el propósito de organizar, administrar y filtrar datos para aquellas personas que, como yo, les gusta crear los productos que necesita, tanto para consumo propio como para la venta. La formulación es un proceso que se lleva a cabo mezclando distintos componentes para conseguir un producto final. El proceso varía en función del producto al qui quieres llegar, también los componentes que lo forman y las fases en las que está dividido este proceso.

La formulación se puede aplicar en ámbitos muy diversos, desde la realización de pasta para moldear hasta la creación de una colonia. *Mesta* pretende abarcar todos estos distintos ámbitos, dejando al usuario personalizar todos las formulaciones y procesos. Esta aplicación podrá servir para llevar un inventario de los productos que tienes disponibles, las características y propiedades de cada uno de ellos y el recetario de las formulaciones realizadas.

La idea nace de la necesidad personal. Hace cinco años comencé a indagar poco a poco en el mundo de la cosmética natural, especialmente los champús sólidos. Al principio los hacía solo para mí, pero fue cuestión de tiempo que la gente que me rodea comenzará a pedirme que también les hiciera a ellos uno específico para su tipo de cabello. Fue cuestión de tiempo que el pequeño Excel que tenía para mí creciera hasta llegar a unas dimensiones muy tediosas de manejar. Esta aplicación está creada para administrar, gestionar y formular de una manera sencilla y cómoda.

Aunque esta idea nace de algo tan pequeño como la creación de champú en casa, el uso de esta aplicación en otros campos es potencialmente de gran utilidad. Una base de datos personificada con un diseño y usabilidad amigable y sencillo.

## Descripción

La formulación consiste en la combinación de distintos componentes para generar un producto final. La formulación se puede aplicar en numerosos campos, podemos separarlas en dos grandes campos en función de si los componentes que se utilizan con abstractos o materiales. En el primer caso, encontramos en las disciplinas de la matemática, la lógica, lingüística, etc.; en el caso de ser componentes materiales podemos hablar de disciplinas como física, ingeniería, química, biológica, industria, medicina o farmacéutica. *Mesta* estará orientado a este último caso, donde los componentes son materiales. Este tipo de formulación genera productos que están permanentemente presentes en el día a día como por ejemplo los medicamentos, cualquier producto de cosmética, los distintos tipos de recubrimientos, tintes, aleaciones, agentes de limpieza, alimentos, combustibles, fertilizantes, pesticidas y un largo etcétera.

*Mesta* es una herramienta creada para facilitar el trabajo a la hora de elaborar la formulación de un producto, creando un entorno fácil de utilizar y accesible para todas las personas. La finalidad de esta aplicación será el almacenaje y búsqueda rápida y sencilla de las distintas formulaciones ya realizadas. Facilitar la realización de la formulación a través de la búsqueda selectiva y cómoda de los distintos componentes.

Un usuario que esté utilizando la aplicación de *Mesta* podrá disponer de varios perfiles. Cada uno de ellos estará vinculado a un tipo de producto final, por ejemplo, si un usuario realiza productos relacionados con la cosmética, podrá tener un perfil para la realización de champús, otro para la realización de gel corporal y otro para mascarillas faciales. De esta forma se facilita el acceso a los distintos tipos de producción del usuario, el almacenaje de las formulaciones y de los componentes.

Esta aplicación se realizará siguiendo la tecnología de PWA (*Progressive web app*), de esta forma conseguiremos que el usuario pueda descargarse la aplicación en el móvil. Se utilizará el *framework* Angular para el *frontend* y se creará en el *backend*, con Laravel y PHP, una arquitectura API Rest de consulta.

## Objetivos

El objetivo principal de este proyecto es crear una aplicación web sencilla y usable que permita la formulación de productos, disponer de un historial de productos ya realizados y un inventario de los productos base disponibles.

### Principales

. Los objetivos principales a la hora de realizar la aplicación serán los siguientes:

* Acceso de un usuario a la aplicación donde están almacenados sus productos.
* Acceso al histórico de productos realizados.
* Acceso al inventario de productos disponibles.
* Creación de productos utilizados para la formulación.
* Creación de productos formulados.

Para conseguir el objetivo de usabilidad y amigabilidad de la aplicación procuraremos hacer lo más sencillo posible el proceso de formulación poniendo la meta en los siguientes objetivos:

* Creación de productos con *tags* de propiedades creadas por el usuario
* Filtro de productos por estas propiedades
* Creación de categorías para separar las fases del proceso de formulación.
* Productos asignados a categorías en función de la fase del proceso donde se utilicen.
* Filtro de productos por categoría.

### Secundarios

Como objetivos segundarios se valorará:

* Creación de distintos perfiles para un usuario, donde se puedan diferencias distintos tipos de productos, ya que los productos base no son los mismos ni las fases del proceso de formulación.
* Cálculo del precio del producto formulado en función del porcentaje de los componentes utilizados.
* Exportación e importación de plantillas Excel de la formulación resultante.

## Metodología

La metodología que se va a llevar a cabo estará basada en la metodología *SCRUM.* Para ello desglosaremos el proyecto en 4 *sprints*, los correspondientes a cada una de las *PECs* definidas en la guía de la asignatura.

Comenzaremos reutilizando las tareas definidas en el apartado de planificación como tareas del backlog del proyecto. Definiremos la estimación de las tareas que irán en cada sprint, sin embargo al inicio de cada uno de estos se revisará y se reajustarán las tareas necesarias (*sprint planning*).

En cada uno de estos *sprints* estarán definidas las tareas necesarias para llevar a cabo los objetivos propuestos. Al final de cada uno se reservarán un día para la revisión del código y el funcionamiento de las tareas realizadas, así como la creación de los bugs si hubiera para dejarlos definidos para el sprint siguiente, es decir, para realizar una revisión global del trabajo realizado (*sprint review*),

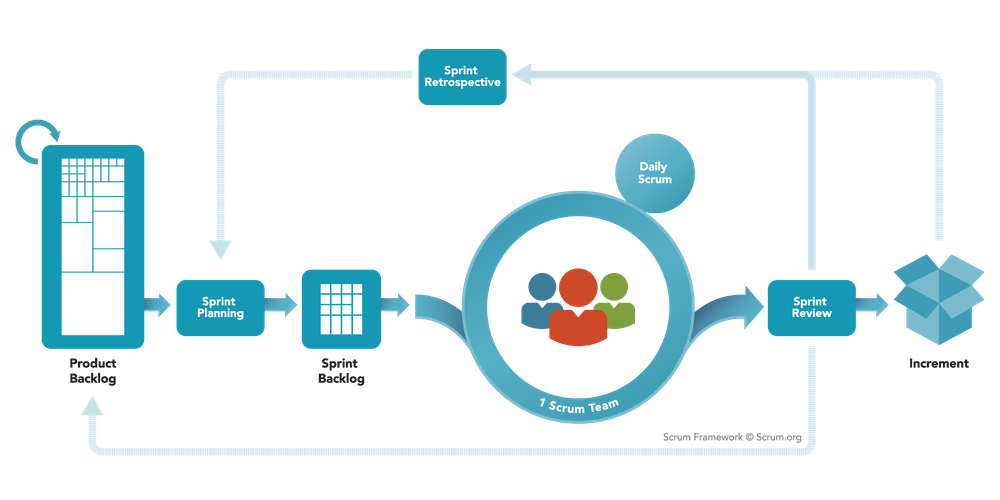


Figura Metodología SCRUM. Ref: <http://www.pmoinformatica.com/>

## Planificación

### Diagrama de Gantt

El proyecto está desglosado en 4 *sprints*, los correspondientes a las PECs definidas en la guía de la asignatura (Figura 2). A continuación, se definirán los objetivos para cada uno de estos *sprints,* al inicio de cada una de estos se definirán las tareas a realizar para llegar a cumplir los asignados a ese período.

Figura Diagrama de Gantt períodos de los sprints

A continuación se definirán los objetivos de cada uno de los *sprints*, como se puede observar, todos comienzan con un *sprint planning* y un *sprint review*, en el primero definiremos las tareas a realizar en el período correspondiente respecto a la realización en sí de la aplicación. Estas tareas se desglosarán en el siguiente apartado y se irán incorporando a la memoria a medida que se vayan definiendo. En el *sprint review* reorganizaremos las tareas que no se pudieron realizar, si se da el caso, en el *sprint* siguiente, y se hará una breve revisión del estado actual del proyecto.

Figura 3 Diagrama de Gantt

### Hítos y entregas

A continuación, las tareas con una pequeña descripción del objetivo a realizar con las horas correspondientes, teniendo en cuenta que 1 crédito universitario equivale a 25 horas y el Trabajo Final es de 12 créditos.

#### PEC 1

El objetivo de este *sprint* es tener claro la definición y el alcance del proyecto. La redacción de distintos puntos de la memoria.

A nivel tecnológico no se comenzará con el proyecto pero se asentarán las bases de qué tecnologías y herramientas se van a utilizar.

Tabla Planificación objetivos sprint PEC 1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Tareas** | **Fecha inicio** | **Días** | **Horas** | **Fecha fin** | **Descripción** |
| PEC 1 | Presentación proyecto | 16-feb. | 3 | 8 | 19-feb. | Presentación y consolidación del proyecto con el profesor |
| Estructura memoria | 19-feb. | 2 | 5 | 21-feb. | Realización de la estructura de la memoria |
| Redacción memoria | 21-feb. | 5 | 14 | 26-feb. | Realización de introducción, descripción, objetivos, métodos y herramientas del proyecto. |
| Planificación | 26-feb. | 2 | 5 | 28-feb. | Planificación del proyecto |
| **Sprint review** | **28-feb.** | **1** | **3** | **1-mar.** | **Revisión del sprint** |

#### PEC 2

Durante el segundo *sprint* se van a abordar tres puntos clave del proyecto. El primero será realizar el estudio de viabilidad; el segundo un estudio de usabilidad de la aplicación, realizaremos los primeros diseños; y el último definir la arquitectura de nuestra aplicación y comenzar a desarrollar el proyecto.

Cuando este sprint termine el objetivo es tener el tanto el estudio de viabilidad como de usabilidad terminados y el esqueleto de nuestra aplicación realizado.

Tabla Planificación objetivos sprint PEC 2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Tareas** | **Fecha inicio** | **Días** | **Horas** | **Fecha fin** | **Descripción** |
| PEC 2 | **Sprint planning** | 1-mar. | 1 | 3 | 2-mar. | Definición del desglose de las tareas a realizar durante el sprint |
| Mejoras memoria | 2-mar. | 2 | 5 | 4-mar. | Continuación y mejoras de los capítulos entregados en la PEC1 |
| Diagramas | 4-mar. | 3 | 8 | 7-mar. | Definición de diagramas y detalle de casos de usos |
| Usabilidad | 7-mar. | 4 | 11 | 11-mar. | Estudio de usabilidad |
| Viabilidad | 11-mar. | 4 | 11 | 15-mar. | Análisis de mercado y viabilidad |
| Arquitectura | 15-mar. | 4 | 11 | 19-mar. | Estudio de la arquitectura de la aplicación |
| Desarrollo proyecto | 19-mar. | 10 | 27 | 29-mar. | Inicio del proyecto y autenticación |
| **Sprint review** | 29-mar. | **1** | 3 | **30-mar.** | **Revisión del sprint** |

#### PEC 3

El tercer *sprint* se dedicará únicamente a la realización del proyecto y los avances paralelamente de la memoria. En este *sprint* se intentará alcanzar los objetivos principales de la aplicación. Teniendo entonces una aplicación testeada y funcional.

Tabla Planificación objetivos sprint PEC 3

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Tareas** | **Fecha inicio** | **Días** | **Horas** | **Fecha fin** | **Descripción** |
| PEC 3 | **Sprint planning** | 30-mar. | 1 | 3 | 31-mar. | Definición del desglose de las tareas a realizar durante el sprint |
| Mejoras memoria | 31-mar. | 2 | 5 | 2-abr. | Continuación y mejoras de los capítulos entregados en la PEC2 |
| Desarrollo proyecto | 2-abr. | 26 | 71 | 28-abr. | Dearrollar el backend y frontend de la aplicación |
| Documentación memoria | 28-abr. | 6 | 16 | 4-may. | Documentar en la memoria los avances que se han llevado a cabo |
| Vídeo | 4-may. | 3 | 8 | 7-may. | Realización del vídeo que se demuestre el estado actual de la aplicación |
| **Sprint review** | 7-may. | **1** | 3 | **8-may.** | **Revisión del sprint** |
| **Sprint planning** | 30-mar. | 1 | 3 | 31-mar. | Definición del desglose de las tareas a realizar durante el sprint |

#### Entrega final

En la entrega final se finalizará el proyecto. Si fuera posible se valorarían añadir alguno de los objetivos secundarios a la aplicación.

Tabla Planificación objetivos sprint Entrega final

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Tareas** | **Fecha inicio** | **Días** | **Horas** | **Fecha fin** | **Descripción** |
| Entrega final | **Sprint planning** | 8-may. | 1 | 3 | 9-may. | Definición del desglose de las tareas a realizar durante el sprint |
| Finalizar el proyecto | 9-may. | 11 | 30 | 20-may. | Finalizar el proyecto |
| Finalizar la memoria | 20-may. | 9 | 25 | 29-may. | Finalizar la memoria |
| Presentación proyecto | 29-may. | 4 | 11 | 2-jun. | Realización de la presentación del proyecto escrita-visual |
| Video presentación | 2-jun. | 3 | 8 | 5-jun. | Realización de una vídeo presentación del proyecto |
| Autoinforme | 5-jun. | 1 | 3 | 6-jun. | Realización de un autoinfome de evaluación sobre el uso de las competencias transversales |

# Definición y arquitectura

## Marco teórico

La creciente preocupación por el medio ambiente ha conseguido que parte de la sociedad se centre en la necesidad de crear un entorno más sostenible y habitable. Una de las corrientes que nacen es el uso de productos orgánicos de origen sostenibles, producto que las grandes industrias han tardado en comercializar y que fueron los negocios locales y pequeños empresarios los que comenzaron a ponerlos al alcance de las personas.

De esta forma nace también la corriente *DIY, Do it yourself*, “hazlo tu mismo”, una manera de contrarrestar el consumismo y disminuir los desechos que genera cada persona. Esta corriente enseñó la posibilidad de poder realizar muchos de los productos que se usan en la rutina diaria de la gran mayoría de las personas desde casa y con conocimiento de la procedencia de los productos.

Tras estas preocupaciones y con la intención de poner a disposición una herramienta que facilite la realización de estos productos, la administración, gestión y formulación de componentes, se crea esta aplicación.

### Base teórica: formulación de un champú sólido

La formulación se aplica a multitud de disciplinas. En el capítulo anterior hablamos de los campos en los que esta aplicación web iba a tener cavidad. Para estudiar las bases de teóricas de esta aplicación estudiaremos la técnica que se utiliza en algunos de estos casos, de esta forma determinar las necesidades que tiene que cubrir la aplicación. El caso en particular que vamos a tratar es el del champú sólido.

#### La composición

La formulación de un champú suele estar compuesta por 5 tipos de componentes:

* Tensioactivo o surfactante: son sustancias sintéticas que se utilizan en aplicación doméstica para el lavado. Cuando se mezcla con agua permiten mantener una emulsión, provocando una espuma que permite el mejor enjuague del champú y mantiene limpio el cabello.
* Hidrolatos: también llamadas aguas florales.
* Aceites vegetales y mantecas.
* Activos: aquellos componentes que ayudan a conseguir las características que quieres que tenga el champú.
* Aceites esenciales

La composición es el porcentaje de cada uno do de estos tipos de componentes en el champú, normalmente suelen diferenciarse en función del tiempo de cabello. Por ejemplo, si queremos realizar un champú para un cabello graso, el porcentaje de aceites vegetales ha de ser menor que en el caso de un cabello seco, compensando este porcentaje añadiendo más activos. Mesta tendrá la opción de definir distintos tipos de composiciones y reutilizarlas en distintas recetas que se realicen.

#### Los ingredientes o componentes

Una vez seleccionada la composición del champú pasamos a escoger los componentes que queremos que formen parte de ello. Para eso tenemos que seleccionar aquellos que cumplan con los requisitos del tipo de champú que estamos creando. La cantidad de componentes que se disponen es ilimitada, una buena forma de hacer una distribución de estos es separarlos por tipo de ingredientes en función de la fase a la que pertenecen. En cada una de las fases tendremos que distribuir el porcentaje entre los distintos ingredientes o componentes que queramos incluir en la receta. Por ejemplo, si en la composición los aceites vegetales ocupan un 15% del producto final y nosotros queremos utilizar tres ingredientes distintos, asignamos el porcentaje deseado a cada uno de ellos hasta llegar a cubrir el 15%.

En Mesta los ingredientes o componentes se presentarán en la página agrupado por los tipos de componentes que se hayan definido a la hora de crear el perfil, se indicará cuál es el porcentaje que se quiere dar a cada ingrediente. Indicando en la fase la parte que se ha cubierto del porcentaje asignado y la que falta por asignar del porcentaje asignado en la composición.

Los componentes o ingredientes se podrán se podrán crear desde Mesta añadiendo un nombre, una descripción, una imagen, la fase o el tipo de ingrediente en función de la composición y las propiedades de este ingrediente.

#### Las propiedades

Las propiedades de los componentes vienen dadas por la naturaleza del propio ingrediente que se esté utilizando. En la aplicación web se permitirá crear las propiedades que el usuario necesite, funcionarán como *tags* y se permitirá una búsqueda ágil de componente o receta por este. Se asignarán a los ingredientes a la hora de su definición o importación; la receta los heredará en función de los componentes que lleve su formulación.

#### La receta

La receta es el resultado final de la formulación. Obtenemos una formulación con una composición específica para las características que se quieren conseguir, cada uno de los elementos de esta composición estarán formados por componentes que disponen de propiedades que heredará la receta a la hora de almacenar en la aplicación web, que permitirá el filtrado por estas.

## Marco tecnológico

A continuación, explicamos las bases de la arquitectura de la aplicación. Explicaremos el concepto de API Rest, el patrón de arquitectura Redux y las aplicaciones web progresivas (PWA).

### API Rest

Una API (Application Programming Interfaces) es un servidor que crea un interfaz HTTP que con un conjunto de definiciones y protocolos crear una comunicación entre otro servidor o una aplicación externa con información a la que tienen acceso de alguna forma, normalmente una base de datos. Las operaciones básicas de una API son las que se llaman CRUD ( *Create-Read-Update-Delete*) que indica tanto la posibilidad de obtener la información como de crearla, modificarla o eliminarla.

La arquitectura de API más habitual es la que se conoce como REST (*Representational State transfer)*, los fundamentos de este diseño incluyen:

* Protocolo cliente/servidor sin estado: Cada una de las peticiones que se realiza en la API, tanto por la parte de cliente y servidor, contiene la información suficiente para trabajar independientemente, no necesitan guardar ningún tipo de estado adicional.
* Dispone de un conjunto de acciones muy bien definidas: GET, POST, PUT y DELETE. Estas acciones se corresponden con las operaciones CRUD definidas anteriormente.
* Dispone de una sintaxis universal para identificar los recursos, cada uno de estos es direccionable a través de su URI\*
* Uso de hipermedios, permitiendo la navegación entre recursos REST.
* Habilitación y uso de la caché , indicando si se puede o no guardar en la caché en la cabecera HTTP.

### Redux

Redux es un patrón de arquitectura para gestionar la información de una aplicación. Permite tener un estado global con variables fijas e inmutables, un control sobre el flujo de datos que permite una baja interacción entre componentes y una arquitectura de datos escalables.

La arquitectura Redux almacena la información de la aplicación centralizada en un solo lugar, en el *Store*. Este nunca se manipula directamente, sino que se hace a través de las acciones y con ello se modifica el *state* de la aplicación. Este nuevo *state* es una combinación entre el estado anterior y la consecuencia de la acción realizada, esta combinación se realiza con las funciones *reducer*.

Dispatch

Subscribe

### *Progressive Web Apps*

Las *Progressive web apps* (PWA) es una aplicación web que utiliza HTML, CSS y Javascript y puede ser consultada desde el dispositivo móvil como una aplicación nativa. Las PWA se basan en las siguientes propiedades según los desarrolladores de Google:

* Progresiva: debe funcionar en todos los navegadores.
* Adaptable: diseño responsivo, que funcione tanto en dispositivos móviles como en escritorio.
* Independiente de la conectividad: utilizar los *service workers* para que la calidad de conectividad no afecte a la experiencia de usuario en la aplicación.
* Más rápido después de la primera carga, en la cual guarda información de la vista para no tener que volver a cargarla.
* El estilo de la app debe tener un estilo propio de aplicación de móvil.
* Siempre actualizada debido al proceso del *service workers.*
* Detectable como una aplicación a través del *manifest*.
* Reconectable: capacidad para utilizar notificaciones *push*.
* Instalable en el dispositivo desde la pantalla de inicio.
* Capacidad de compartirse a partir de una URL.

## Contenidos

La aplicación estará formada por distintas páginas en función del estado de autenticación del usuario. En el caso de no estar autenticado la aplicación nos llevara a una página informativa desde la que podremos logarnos o registrarnos si es la primera vez que accedes a la aplicación. Además, de la página de inicio, la de *login* y la de registro, también se crearon dos páginas más, una para recuperar la contraseña en caso de haberla olvidado un usuario ya registrado y otra para cambiar dicha contraseña.

Una vez has iniciado sesión en la aplicación nos encontramos con tres ramas distintas: configuración de cuenta, las páginas relacionadas con los componentes y páginas relacionadas con las recetas. La configuración de cuenta contendrá el nombre de usuario y el email al que está asociado la cuenta, permitirá modificarlos, la opción de eliminar la cuenta y de cambiar la contraseña. También habrá un formulario de contacto y la visualización de los perfiles creados y la opción de crear uno nuevo. Cuando creas un nuevo perfil te llevará a otra página con un formulario a rellenar con la información necesaria: nombre del perfil, breve descripción y definición de fases del proceso que se lleva a cabo en ese perfil.

Respecto a los componentes nos encontramos con tres páginas. La primera de ellas es la vista en lista de todos los componentes definidos agrupados por la fase del proceso al que pertenecen. También estarán situados los filtros de nombre, propiedades y fase del proceso al que pertenecen. Desde esta vista, seleccionando un componente podemos ver el detalle de este, que contendrá información como el nombre, descripción, imagen si hubiera, fase a la que pertenece y propiedades del componente. Desde aquí podremos acceder a la edición o eliminación de este componente. Además, desde la vista de lista de componente podremos acceder a la creación de uno nuevo donde habrá que rellenar un formulario con información como el nombre, la descripción, una imagen, la fase a la que pertenecen y las propiedades del componente, en caso de que no esté definida la propiedad que se necesita se permitirá la creación, edición o eliminación de propiedades desde esta vista.

Respecto a las recetas o formulaciones encontramos unas vistas muy parecidas a la de los componentes: una lista, una vista de detalle y otra de edición o creación. En la primera vista tenemos la lista de recetas o formulaciones realizadas, estos ítems se podrán filtrar por nombre y/o propiedades. Desde aquí podemos acceder a la vista de formulación donde encontraremos un formulario con nombre, descripción, composición escogida o la opción de añadir una nueva composición, y los componentes que forman esta formulación. Volviendo a la vista de lista podemos acceder a la vista de detalle de una receta, donde estará la información del formulario y proceder a la edición de la formulación o eliminación.

En resumen:

* Los componentes tendrán asignado una fase del proceso de creación, así como todas aquellas propiedades necesarias.
* Las formulaciones o recetas tendrán indicado qué componentes tiene asignados en cada una de las fases para su formulación.
* Los formulaciones o recetas heredarán las propiedades de los componentes.

Respecto a los filtros de componentes:Nombre

* Fase de proceso a la que pertenecen
* Propiedades

Respecto a los filtros de los productos formulados:

* Nombre
* Propiedades

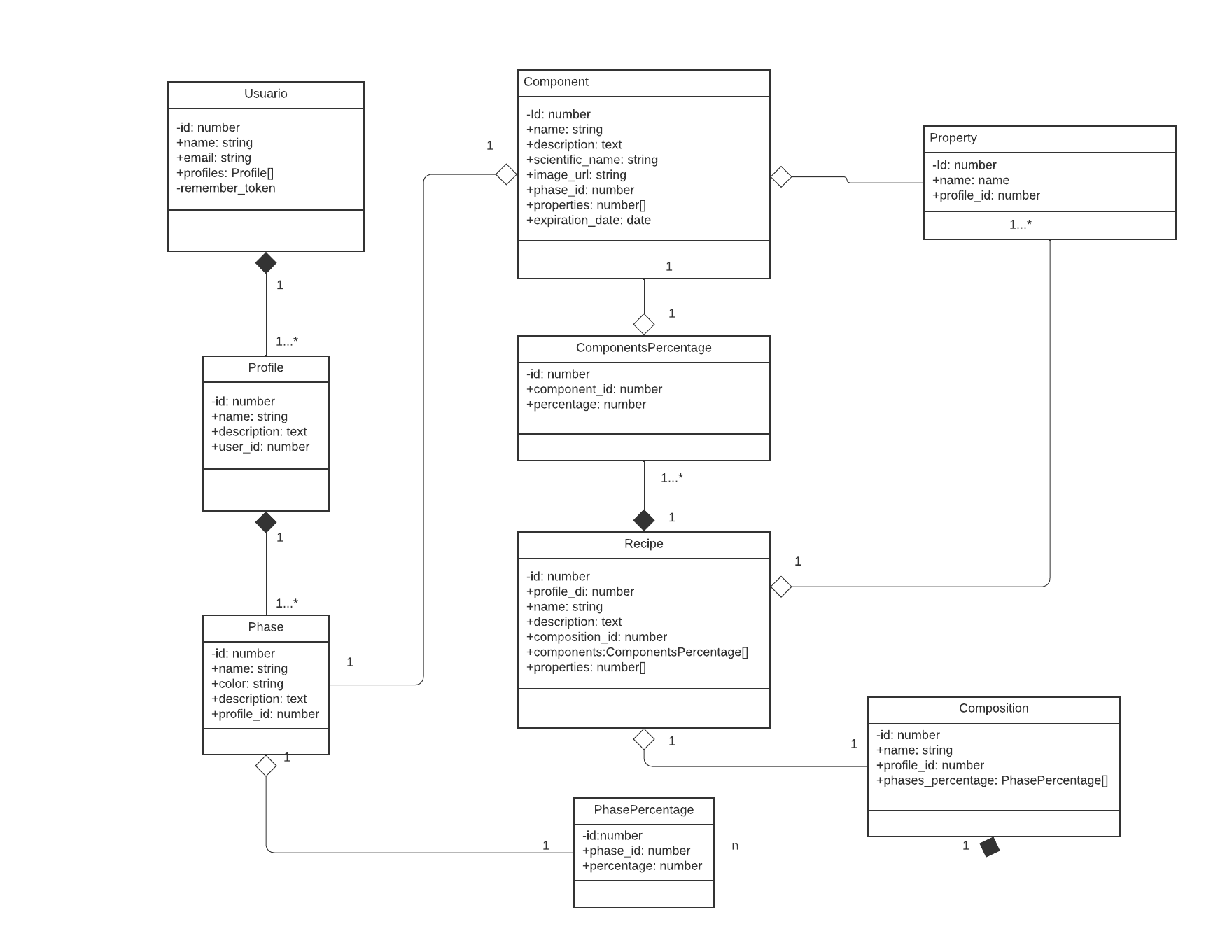
La aplicación web tendrá un diseño totalmente responsive para que sea cómodo utilizarla desde dispositivos más pequeños, como un *smartphone* o una *Tablet,* aplicando las tecnologías de las aplicaciones web progresivas.

Finalmente, las páginas que vamos a crear serán las siguientes:

Tabla Contenidos de la aplicación web

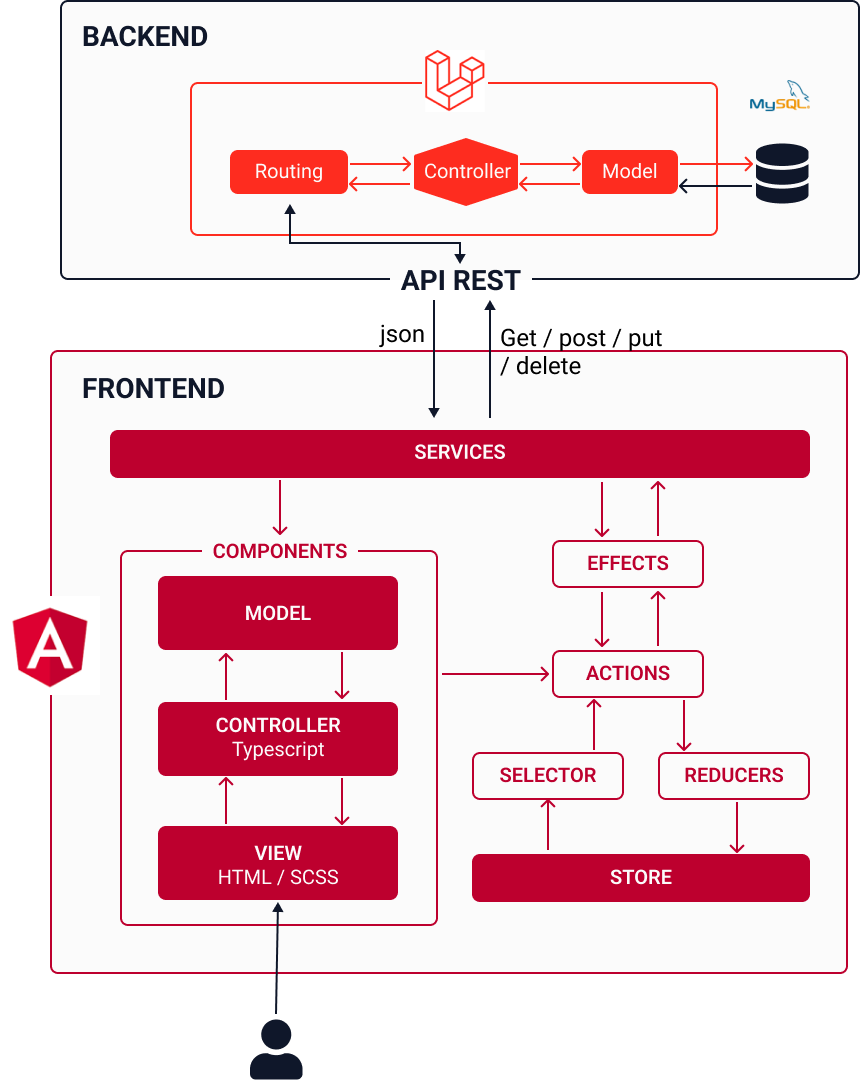
|  |  |
| --- | --- |
| *Landing page* | Página de inicio contará con una breve presentación de la aplicación web, así como lo accesos o registros del usuario. |
| *Login* | Página que permite el inicio de sesión en la aplicación |
| Registro | Página que permite el registro de un usuario por primera vez en la aplicación |
| Contraseña olvidada | Formulario en el cual se puede insertar el correo electrónico para recuperar la contraseña |
| Generar contraseña | Formulario para cambiar la contraseña de usuario |
| Configuración de usuario | Página donde se podrán realizar los ajustes del usuario, así como gestionar los perfiles. |
| Creación de perfil | Desde la pantalla de configuración de usuario se podrá crear, editar o eliminar perfiles. La gestión de cada perfil será nombre, descripción y fases de proceso de ese perfil con asignación de colores. |
| Vista recetas | Se selecciona el primer perfil por defecto y se muestra una vista de los productos realizados. En esta vista habrá un buscador con filtros para buscar los productos finales. |
| Vista componentes | A partir de la vista anterior, podremos acceder a la vista en lista de los componentes. En esta página se situará un buscador con filtros para buscar los componentes. |
| Formulario componentes | A partir de la vista anterior, se podrá tanto crear como editar nuevos componentes. Este formulario estará compuesto por: nombre, descripción, propiedades y fase asignada. |
| Detalle componentes | A partir de la página de **vista de componentes**se podrá acceder a los detalles de un componente. |
| Formulación | A partir de la página de **vista de productos** se podrá crear o editar un producto ya realizado y se accederá a la formulación del producto. |
| Detalle receta | A partir de la página de **vista de productos** se podrá acceder a la vista de detalles del producto formulado. |

## Diagramas de clases

A continuación, se presenta el diagrama de clases de la aplicación. En este se incluye la clase de los usuarios, perfiles, componentes, propiedades, composiciones y recetas o formulaciones y cómo se relacionan entre ellas.

## Arquitectura de la aplicación

Vamos a diferenciar el proyecto en dos partes: *backend* y *frontend*. El backend se encargará de comunicarse con la base de datos y proporcionar al *frontend* una API de servicios REST. Este *backend* estará realizado con el *framework* Laravel que utiliza el lenguaje PHP. Para la base de datos se utilizará un sistema de gestión relacional MySQL. Por otro parte, el *frontend* se va a realizar utilizando el framework Angular. Para facilitar la gestión del usuario y perfiles se va a utilizar el tipo de arquitectura REDUX. En resumen, contaremos con una arquitectura cliente-servidor sencilla, donde el cliente consume los datos que el servidor le proporciona.

Para la obtención de los datos que se usarán en la parte del cliente se utilizará el protocolo HTTP así como para la autenticación de usuario. La comunicación entre las partes se hará con el uso común del flujo de HTTP: *get*, *post*, *put¸*y *delete*. Los datos se devolverán al cliente con formato JSON.

### Backend

El *backend* se realizó con el *framework* Laravel, como la gran mayoría de los *frameworks* tiene una arquitectura de *software* MVC (Modelo-vista-controlador). Este tipo de arquitectura consiste en que el modelo gestione los datos de la aplicación y las acciones que se puede hacer sobre ellos. La vista proporciona una vista para poder realizar las acciones y el controlador recibe las acciones y las ejecuta sobre el modelo.

Las *frameworks* MVC han de disponer herramientas para gestionar las operaciones básicas CRUD y que permita desarrollas un servicio web que siga la arquitectura REST. Las funcionalidades básicas de un *framework* MVC son:

* Un mapeo de URL claro y limpio.
* Acceso a la base de datos
* *Templates* o plantilla.
* El cacheo de páginas para obtener un mayor rendimiento
* Autenticación de usuarios
* Seguridad

El *framework* Laravel, basado en PHP, cumple con todos estos requisitos. Como gestos de paquetes del *backend* se ha escogido Composer, ya que es más apropiado para la gestión de paquetes en entornos de PHP.

### Frontend

El desarrollo del *frontend* lo hemos llevado a cabo utilizando el *framwork* de Angular. A grandes rasgos, Angular permite:

* Creación y uso de tus propios componentes, lo cual permites reducir la complejidad del proyecto y facilita la realización de tests.
* Utiliza lo que se conoce como *data binding* que permite la relación directa de los elementos de la vista con el controlador.
* Importación de dependencias específica para cada componente, se puede definir servicios a distintos componentes.
* Realización de test.

El gesto de paquetes que utiliza Angular es el de node.js, npm, este permite utilizar herramientas de forma ágil como preprocesadores de CSS. Respecto al estilo de la aplicación se utilizará el *framework* de CSS Tailwind. Para la normalización del código se utilizará la herramienta de prettier y ESLint. Angular está creado acorde al lenguaje tipado TypeScript. Para el testing se utilizará la librería Jest. Respecto a la gestión de la asincronía utilizaremos observables implementados con la ayuda de la librería RxJS que permite una programación reactiva. Finalmente para la gestión del estado de la aplicación utilizaremos la arquitectura Redux con la implementación de NgRX en el proyecto.

### Gestion de versión

Para mantener un control de versionado de nuestro proyecto utilizaremos el software Git.

## Plataformas de desarrollo

### Software

Las herramientas utilizadas para el desarrollo del proyecto a nivel de software son:

Tabla Herramientas de software utilizadas en el proyecto

|  |  |
| --- | --- |
| **Visual Studio Code** | Editor de código para la creación del *frontend* de la aplicación. |
| **PhpMyAdmin** | Gestión y administración de la base de datos MySQL |
| **Figma** | Editor de gráficos en línea, utilizado para generar prototipos y diseñar la aplicación |
| **GitHub** | Plataforma para almacenar y gestionar las versiones del proyecto. |
| **PostMan** | Realización de las pruebas del servicio API Rest proporcionado por el backend |
| **Microsoft Office** | Paquete de Microsoft Office para la realización de la documentación del proyecto. |
| **FlowMapp** | Plataforma en línea que permite la realización de *sitemaps,* flujos de usuario, descripción de páginas, etc. |
| **LucidChart** | Plataforma en línea que permite la realización de diagramas UML |
| **Xampp** | Paquete de software libre que ofrece la gestión de bases de datos MySQL y el servidor Web |

### Hardware

A nivel de hardware se han empleado:

Tabla 7 Herramientas de hardware utilizadas en el proyecto

|  |  |
| --- | --- |
| **ASUS** | Portátil con sistema operativo Windows 10 |
| **IPhone 7 Plus** | Sistema operativo iOS 15.2 |

# Diseño y Usabilidad

## Perfiles de usuario

A continuación, se presentan dos de los tipos de perfil a los que va a ir dirigida la aplicación.

Ester Domínguez



**26**

**Barcelona**

**Graduada**

**Periodismo**

**30h semanales**

**18.000 € anuales**

Edad

Residencia

Nivel educacional

Profesión

Jornada laboral

Salario

**BIOGRAFÍA**

Ester tiene 26 años, nacida en Vigo vive desde hace dos años en Barcelona y, hasta la pandemia, trabajaba en una compañía creando contenido audiovisual para la página de la empresa y las redes sociales. Además, estudia un máster en desarrollo web en la Universitat Oberta de Catalunya. Ester lleva años modificando sus hábitos para adecuarlos a un estilo de vida cada vez más Residuo cero, empezó comprando a granel y cambiando sus productos de aseo por aquellos reutilizables. Durante la pandemia, tras perder el empleo, Ester comenzó a vender por Instagram los champús que hasta ahora había fabricado para sus familiares y amigos, como tuvo bastante éxito decidió crear su propia marca y llevar el negocio al siguiente nivel y vende productos de higiene corporal, capilar y productos de limpieza a granel.

**MOTIVACIONES**

Lo que hace especial el negocio de Ester y la razón por la que funciona tan bien es porque hace productos en función de las necesidades del cliente. Sin embargo, a medida que aumenta su cartera de cliente también lo hace el Excel donde hasta ahora llevaba apuntadas los componentes y formulaciones que iba realizando.

**OBJETIVOS**

Poder gestionar los datos de formulaciones, componentes y propiedades de cada uno de ellos, poder acceder a la información de forma cómoda y sencilla.

Alexandre Romero



**42**

**Pontevedra**

**Licenciado**

**Profesor**

**40h semanales**

**38.000 € anuales**

Edad

Residencia

Nivel educacional

Profesión

Jornada laboral

Salario

**BIOGRAFÍA**

Alexandre tiene 42 años, está divorciado desde hace dos, en su matrimonio tuvo un hijo y una hija que ahora tienen 8 y 10 años respectivamente. Alexandre tiene una licenciatura en biología y actualmente es profesor en el instituto Rosalía de Castro situado a las afueras de la ciudad donde reside, Pontevedra. Normalmente Alexandre se mueve andando por la ciudad, para ir hasta su trabajo le gusta ir en bicicleta si el tiempo lo permite, si no comparte coche con otro profesor del instituto. Alexandre disfruta enseñándole a sus hijos los beneficios de tener un huerto en el pequeño jardín situado detrás de su casa. A Alexandre le gusta asistir a talleres que imparten en su ciudad y alrededores. Hace unos años fue a uno donde aprendió a realizar productos de limpieza e higiene y desde aquella se convirtió en los principales regalos para amigos y familiares. Disfruta enseñando a sus hijos como aprovechar al máximo posible todo lo que le rodea.

**MOTIVACIONES**

La lista de productos que Alexandre realizar ha crecido y con ello la gestión de los componentes y la recetas que realiza. Le gustaría poder tenerlo todo a mano en lugar de necesitar el uso de libretas donde tener todo apuntado y cada vez se le hace más difícil dar con la información.

**OBJETIVOS**

Poder gestionar y crear las formulaciones que necesita aunque en cualquier lugar desde el dispositivo móvil, con la información accesible y organizada de forma cómoda.

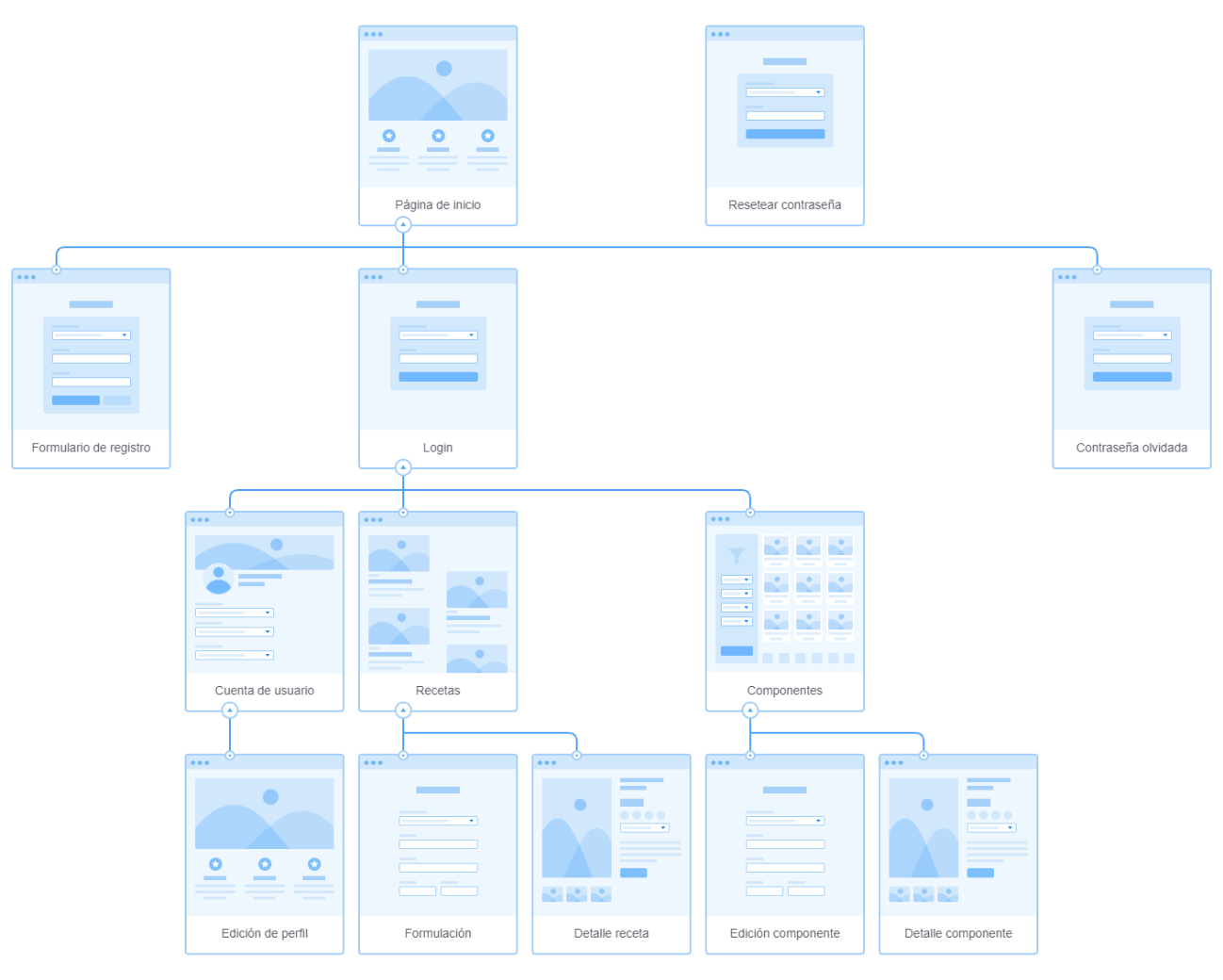
## Sitemap

En la figura se muestra el mapa web de la aplicación. Desde la página inicial podemos ver la información explicativa del contenido de la aplicación. De aquí nos vincula tanto a la página para iniciar sesión como a la de registro. Así mismo, como a la página de restaurar la contraseña en el caso de ser un usuario ya registrado que ha olvidado la contraseña. Al mismo nivel que la página de inicio, está la página de resetear la contraseña mediando un enlace del correo electrónico.

Una vez iniciada la sesión en la aplicación podemos acceder a tres apartados distintos: configuración de cuenta, los componentes y las recetas. Desde la página de configuración de cuenta podemos cambiar nuestro correo electrónico, el nombre o la contraseña, así como, eliminar la cuenta. Además, nos permite el acceso a la página de creación, edición y eliminación de perfiles.

Relacionado con los componentes, tenemos como vista principal la lista de estos, desde donde podemos crear uno nuevo o seleccionando un componente podemos acceder a los detalles de este y a la edición o eliminación de ducho componente.

Relacionado con las recetas o formulaciones accedemos desde la lista de recetas a realizar una nueva formulación o seleccionando una de las recetas acceder a los detalles de la receta y desde aquí editar y/o eliminar la receta.



## Prototipos

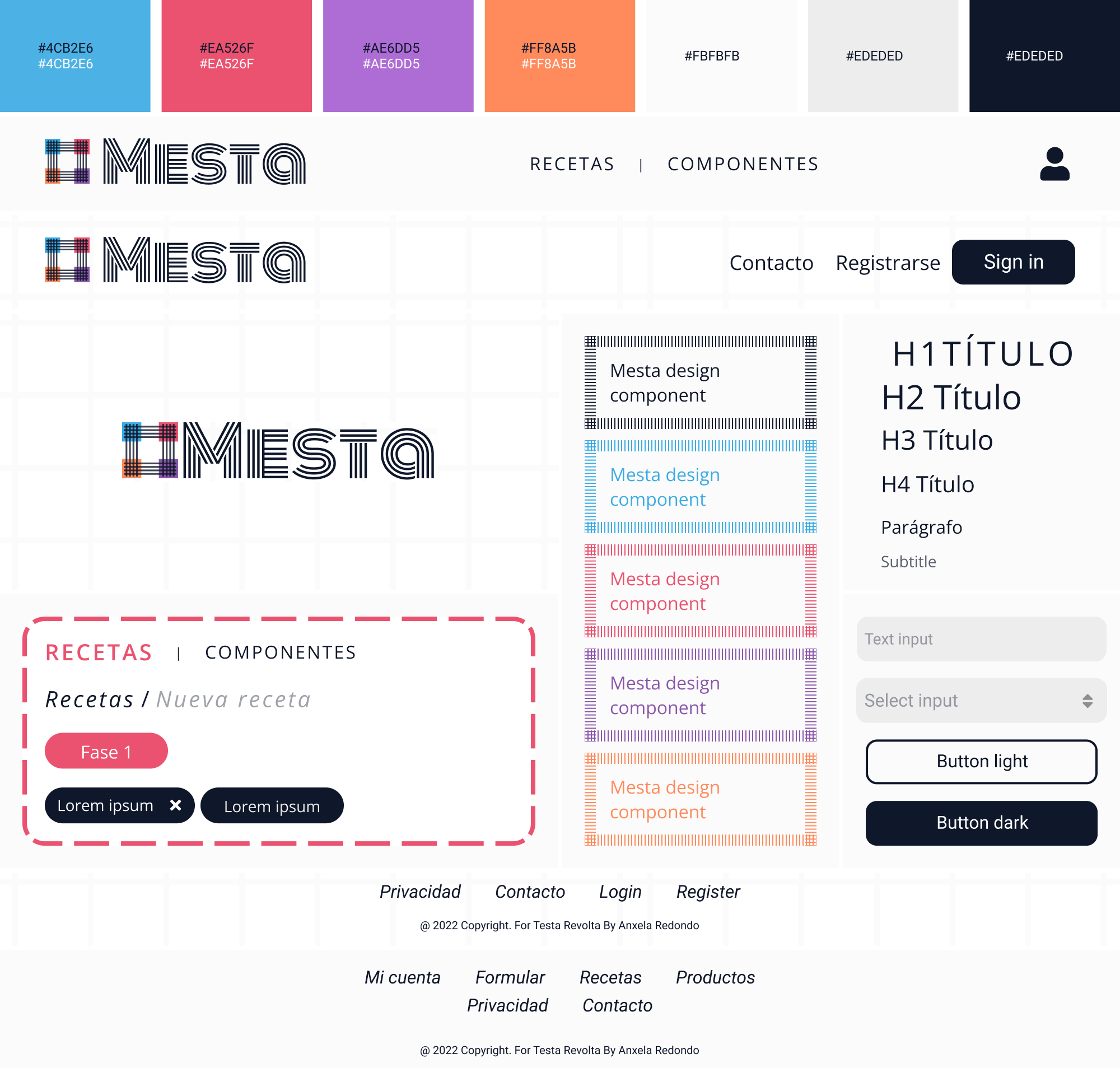
A continuación, se presentan los prototipos de baja calidad para un dispositivo de escritorio y en alta calidad para dispositivos móvil y escritorio.

### Prototipos de baja calidad

Se realizaron los siguientes *Wireframes* como prototipo de baja calidad para la versión de escritorio. Se muestra a continuación la vista inicial de cada pantalla, en el Anexo II Prototipo de baja calidad se encuentran los *wireframes* completos.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

### Prototipos de alta calidad

Recogidos en la siguiente figura encontramos los puntos clave principales del diseño para la aplicación.

En la parte superior vemos los colores principales de la aplicación. Los dos siguientes componentes son el *header* de la aplicación, el primero para cuando el usuario ha iniciados sesión en la aplicación y el segundo para cuando está en la página principal. A continuación, podemos ver el título con la tipografía *Monoton* de *Google Fonts* y el logo diseñado para la aplicación, este está colocado sobre un fondo diseñado para la página principal de la aplicación. Debajo de este encontramos el contenedor que se va a utilizar para envolver ciertos elementos de la página, dentro de este contenedor tenemos, primero el menú principal de la aplicación; el *breadcrumb*; y el componente que será para fase y para indicar las propiedades. A la derecha vemos la estética de cinco componentes que corresponderán al estilo utilizado para la representación de componentes y de recetas realizadas. A la derecha de la imagen, están dispuestos los distintos niveles de títulos, parágrafo y subtítulos, donde se ha usado la tipografía Open Sans. Debajo de la tipografía están los componentes propios de formularios, *inputs* y botones. Finalmente, cerrando la figura, nos encontramos con el *footer*, el primero sería el correspondiente a la página inicial y el último cuando se ha iniciado sesión en la aplicación.

A continuación, presentamos los *mockups* de la versión móvil y a continuación de la versión de ordenador. Al igual que en los prototipos de baja calidad se mostrará la vista inicial al entrar en la página y con todo detalle en el Anexo III Prototipos de alta calidad.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

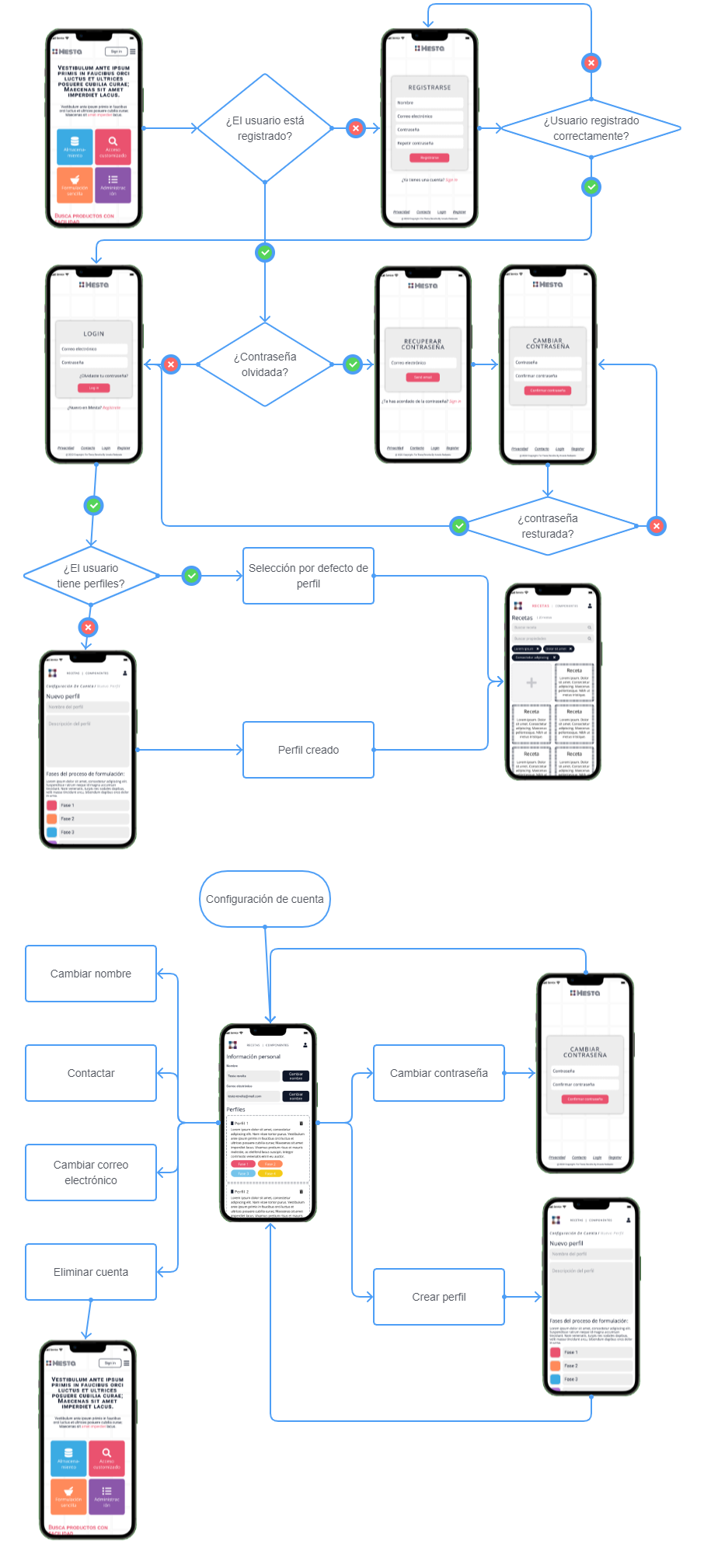
|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

## Diagrama de flujo

A continuación se representa los diagramas de flujos de uso de la aplicación. Empezando por la autenticación, siguiendo con las acciones relacionadas con la configuración de cuenta, seguimos con las acciones relacionadas con los componentes y terminamos con las relacionadas con las recetas.

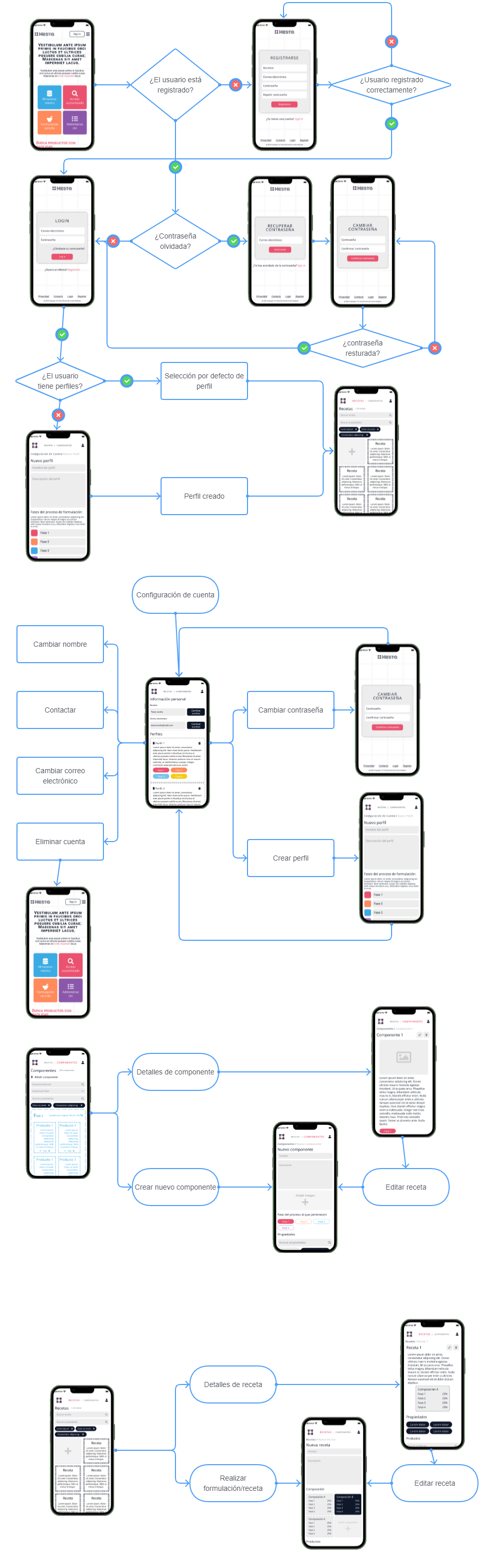
### Autenticación de usuario

### Configuración de cuenta

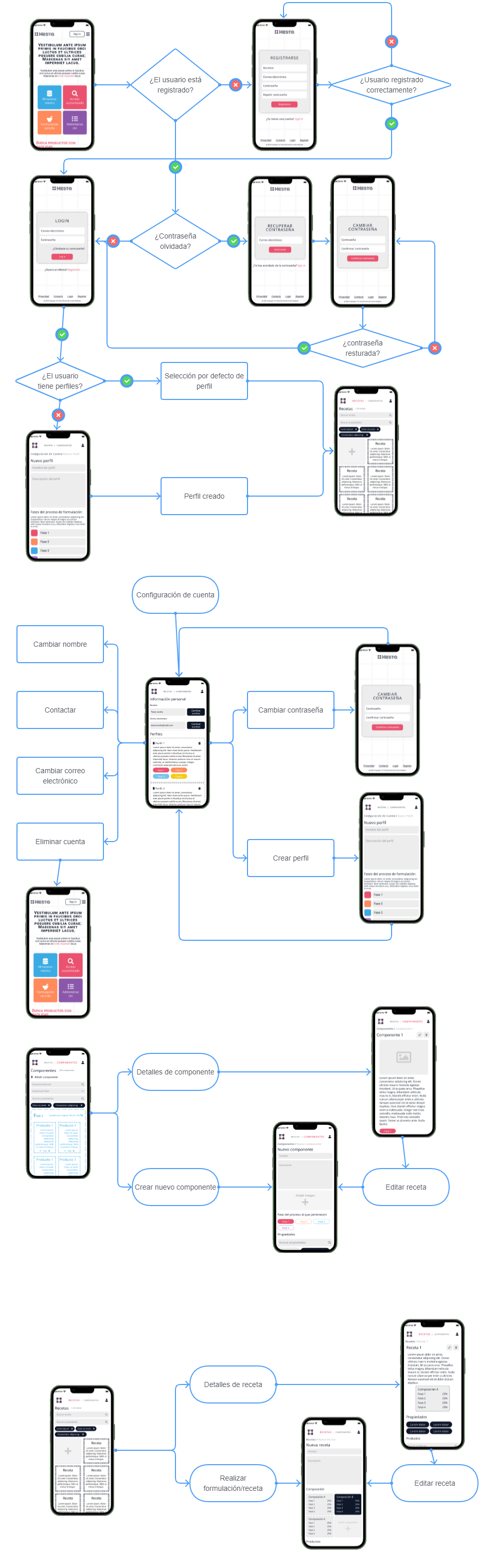
La configuración de cuenta se accede desde la cabecera y las acciones que se pueden realizar son cambiar cuenta, cambiar correo electrónico, contactar, crear nuevo perfil o editar existente, cambiar contraseña y eliminar cuenta. Tanto el cambio de correo como contraseña se realizan desde la propia página de configuración de cuenta, esta también dispone de un formulario de contacto.

### Accesos relativo a los componentes

Desde la lista de componentes podemos acceder a los detalles de un componente, así como, editar y crear nuevo componente.



### Accesos relativos a la recetas o formulaciones

Desde la página de recetas se puede ver el detalle de una receta y crear o editar recetas ya existentes.

## Arquitectura REDUX: Estados de la aplicación

### Estado: *Auth*

### Estado: *User*

### Estado: *Profiles*

### Estado: *Phases*

### Estado: *Properties*

### Estado: *Compositions*

# Desarrollo de la aplicación

## Proceso de desarrollo

Se segmento el proyecto en distintas tareas, diferenciándolas entre aquellas específicas del *backend¸frontend* y diseño (UI/UX).

|  |  |
| --- | --- |
| **BACKLOG** | |
| BACK | Iniciar proyecto Laravel |
| FRONT | Iniciar proyecto Angular |
| BACK/FRONT | Registro de usuarios |
| BACK/FRONT | Login usuarios |
| BACK/FRONT | Logout usuarios |
| BACK/FRONT | Contraseña olvidada |
| BACK/FRONT | Recuperar contraseña |
| UI/UX | Diseño Lo-Fi |
| UI/UX | Diseño Hi-Fi |
| FRONT | Redux refactor |
| FRONT | Aplicar diseño páginas de autenticación |
| FRONT | Página de inicio |
| FRONT | Configuración de usuario |
| BACK | Catálogo perfiles y peticiones GET - POST- PUT-DELETE |
| FRONT | Edición de perfiles |
| BACK | Catálogo fases de proceso y peticiones GET - POST- PUT-DELETE |
| BACK | Catálogo fases de proceso y peticiones GET - POST- PUT-DELETE |
| BACK | Catálogo componentes y peticiones GET - POST- PUT-DELETE |
| BACK | Catálogo propiedades y peticiones GET - POST- PUT-DELETE |
| FRONT | Formulario creación de componentes |
| FRONT | Lista de componentes |
| FRONT | Detalle de componentes |
| BACK | Catálogo recetas y peticiones GET - POST -PUT -DELETE |
| FRONT | Creación de formulación |
| FRONT | Lista de recetas |
| FRONT | Detalle de recetas |
| FRONT | Filtro por nombre de componentes |
| FRONT | Filtro por nombre de recetas |
| FRONT | Creación de propiedades |
| FRONT | Edición de propiedades |
| FRONT | Eliminación de propiedades |
| FRONT | Creación de composiciones |
| FRONT | Edición de composiciones |
| FRONT | Eliminación de composiciones |
| FRONT | Edición de formulación |
| BACK/FRONT | Filtro componentes por propiedades, fase de proceso nombre |
| BACK/FRONT | Filtro de receta por nombre, componentes y propiedades |

### PEC 2 – Sprint 1

Durante el primer *sprint* se inició el proyecto tanto en *backend* como en *frontend* se creó el entorno de desarrollo y se realizó la autenticación del usuario, el registro de este, el cambio de contraseña y la posibilidad de recuperar la contraseña.

|  |  |  |
| --- | --- | --- |
| **PEC 2 – Sprint 1** | | **Horas** |
| BACK | Iniciar proyecto Laravel | 1,5 |
| FRONT | Iniciar proyecto Angular | 0,5 |
| BACK/FRONT | Registro de usuarios | 2 |
| BACK/FRONT | Login usuarios | 2 |
| BACK/FRONT | Logout usuarios | 1 |
| BACK/FRONT | Contraseña olvidada | 1 |
| BACK/FRONT | Recuperar contraseña | 1 |
| UI/UX | Diseño Lo-Fi | 6 |
| UI/UX | Diseño Hi-Fi | 12 |
| **Horas totales** | | **27** |

### PEC 3 – Sprint 2

Durante el segundo sprint se llevó a cabo el groso del proyecto. Se realizó:

* Edición y eliminación de cuenta nueva.
* Creación, edición y eliminación de perfiles de usuario.
* Creación, edición y eliminación de fases asociadas a perfiles.
* Creación, edición y eliminación de propiedades.
* Creación, edición y eliminación de composiciones.
* Creación, edición y eliminación de componentes.
* Eliminación de recetas.
* Formulación.

|  |  |  |
| --- | --- | --- |
| **PEC 3 – Sprint 2** | | **Horas** |
| FRONT | Redux refactor | 5 |
| FRONT | Aplicar diseño páginas de autenticación | 4 |
| FRONT | Página de inicio | 6 |
| FRONT | Configuración de usuario | 6 |
| BACK | Catálogo perfiles y peticiones GET - POST- PUT-DELETE | 4 |
| FRONT | Edición de perfiles | 6 |
| BACK | Catálogo fases de proceso y peticiones GET - POST- PUT-DELETE | 4 |
| BACK | Catálogo fases de proceso y peticiones GET - POST- PUT-DELETE | 4 |
| BACK | Catálogo componentes y peticiones GET - POST- PUT-DELETE | 4 |
| BACK | Catálogo propiedades y peticiones GET - POST- PUT-DELETE | 4 |
| FRONT | Formulario creación de componentes | 6 |
| FRONT | Lista de componentes | 6 |
| FRONT | Detalle de componentes | 6 |
| BACK | Catálogo recetas y peticiones GET - POST -PUT -DELETE | 4 |
| FRONT | Creación de formulación | 16 |
| FRONT | Lista de recetas | 6 |
| FRONT | Detalle de recetas | 6 |
| FRONT | Filtro por nombre de componentes | 1 |
| FRONT | Filtro por nombre de recetas | 1 |
| FRONT | Creación de propiedades | 1 |
| FRONT | Edición de propiedades | 1 |
| FRONT | Eliminación de propiedades | 1 |
| FRONT | Creación de composiciones | 2 |
| FRONT | Edición de composiciones | 1 |
| FRONT | Eliminación de composiciones | 1 |
| **Horas totales** | | **106** |

## APIs utilizadas

Indicar las APIs de terceros que se hayan utilizado para el trabajo, y detallar su uso.

## Seguridad

Bearer access token Passport laravel

Información acerca de cómo se han enfrentado los riesgos de seguridad informática inherentes al trabajo por su tipología, temática, etc.

## Tests

Descripción de los tests y pruebas realizadas para poner a prueba el trabajo respecto a su funcionalidad, rendimiento, utilidad, etc., indicando, si ha lugar, qué herramientas, personas y procedimientos/protocolos se han aplicado. Incluir, por ejemplo, tests de:

* Usuario
* Usabilidad
* Seguridad

## Versiones de la aplicación

Versiones compiladas/preparadas para las distintas entregas parciales de proyectos de desarrollo o, incluso, de contenidos multimedia.

Para desarrollo de software habitualmente son:

* Alpha
* Beta
* 1.0

## Requisitos de instalación y uso

Además de los requisitos que se explicaran en los siguientes apartados, será necesario tener instalago Git para el control de versiones y clonación del proyecto.

### Requisitos back-end

Para la gestión del backend se ha de tener instalado el manejador de paquetes Composer.

|  |
| --- |
| **Requisitos backend - Laravel** |
| Versión de PHP mayor o igual que la 7.2.5 |
| BCMath PHP Extension |
| Ctype PHP Extension |
| Fileinfo PHP extension |
| JSON PHP Extension |
| Mbstring PHP Extension |
| OpenSSL PHP Extension |
| PDO PHP Extension |
| Tokenizer PHP Extension |
| XML PHP Extension |

### Requisitos front-end

|  |
| --- |
| **Requisitos front-end - Angular** |
| Node.js |
| Npm |
|  |

### Requisitos hardware

Obtener una instancia de MySQL y PhpMyAdmin.

|  |  |
| --- | --- |
| **Dispositivo de desarrollo – Requisito MySQL Database** | |
| *Sistema operativo* | *Arquitectura* |
| Oracle Linux 8 / Red Hat Enterprise Linux 8 / CentOS 8 | x86\_64, ARM 64 |
| Oracle Linux 7 / Red Hat Enterprise Linux 7 / CentOS 7 | ARM 64 |
| Oracle Linux 7 / Red Hat Enterprise Linux 7 / CentOS 7 | x86\_64 |
| Oracle Linux 6 / Red Hat Enterprise Linux 6 / CentOS 6 | x86\_32, x86\_64 |
| Solaris 11 (Update 4+) | SPARC\_64 |
| Ubuntu 20.04 LTS | x86\_64 |
| Ubuntu 18.04 LTS | x86\_32, x86\_64 |
| SUSE Enterprise Linux 15 / OpenSUSE 15 (15.3) | x86\_64 |
| SUSE Enterprise Linux 12 (12.5+) | x86\_64 |
| Debian GNU/Linux 11 | x86\_64 |
| Debian GNU/Linux 10 | x86\_64 |
| Microsoft Windows 2022 Server | x86\_64 |
| Microsoft Windows 2019 Server | x86\_64 |
| Microsoft Windows 2016 Server | x86\_64 |
| Microsoft Windows 2012 Server R2 | x86\_64 |
| Microsoft Windows 11 | x86\_64 |
| Microsoft Windows 10 | x86\_64 |
| macOS 12 | x86\_64, ARM\_64 |
| macOS 11 | x86\_64, ARM\_64 |
| Generic Linux (tar format) | x86\_32, x86\_64, glibc 2.12, libstdc++ 4.4 |

## Instrucciones de instalación

El repositorio del proyecto se encuentra alojado en la siguiente dirección <https://github.com/anxela-re/mesta>. Dentro del repositorio nos encontramos con dos carpetas frontend y backend, donde se aloja cada una de las partes del proyecto a la que hacen referencia con su nombre. Las instrucciones de instalación para cada una de ellas se explican a continuación y se encuentran dentro del proyecto en el archivo *README* ubicado en la raíz de cada una de estas carpetas. Para clonar este repositorio ejecutar:

git clone <https://github.com/anxela-re/mesta.git>

### Back-end

Para la instalación del backend, una vez tenemos descargado el repositorio y estamos situados en la carpeta de “backend”, instalamos las dependencias del proyecto ejecutando:

composer install

Se tiene que crear el archivo en la carpeta raíz de “backend” el archivo .env con el nombre que se le haya dado a instancia de la base de datos en donde se quiere almacenar los catálogos de la aplicación:

DB\_CONNECTION=mysql

DB\_HOST=127.0.0.1

DB\_PORT=3306

DB\_DATABASE=mesta

DB\_USERNAME=root

DB\_PASSWORD=

Por ejemplo, en este caso: Mesta. Para la gestión de base de datos puedes utilizar cualquier software que trabaje con datos MySQL e iniciar los servidores Apache y MySQL. Nosotros estamos utilizando XAMPP, y como herramienta para administrar los datos phpMyAdmin.

Una vez tenemos levantado el servidor. La primera vez que levantamos el servidor del proyecto se va a necesitar ejecutar el comando

php artisan Passport:keys

Este comando genera las llaves de encriptación que la librería *Passport* necesita para generar los *access tokens* de los usuarios. A continuación se realizan las migraciones necesarias para la aplicación, ejecutando:

php artisan migrate

Para terminar, se arranca el servidor local ejecutando:

php artisan serve

### Front-end

Tras la clonación del proyecto se accede al directorio de “frontend”. Para realizar la instalación del proyecto se ejecuta:

npm install

Una vez terminado de instalar todas las dependencias del front-end, levantamos el front-end ejecutando:

npm start

Una vez haya finalizado la ejecución se abrirá automáticamente una ventana con la aplicación. En caso de no ser así, si todo ha ido bien, la aplicación debería estar levantada en la dirección <https://localhost:4200>

## Manual de usuario

Información con pasos detallados acerca de cómo se debe utilizar el servicio/aplicación. Las presentes instrucciones deben también acompañar al servicio/aplicación en un archivo contenido en sus directorios o como una función más del servicio/aplicación.

## Bugs

Lista de errores detectados en la funcionalidad del servicio/aplicación.

## Proyección a futuro

Información, predicciones y sugerencias acerca de ampliaciones a futuro del trabajo, y/o lista de mejoras a realizar en hipotéticas futuras versiones del servicio/aplicación.

# Presupuesto y mercado

## Presupuesto

Costes detallados del proyecto. Incluyendo:

* Equipo humano
* Equipamiento técnico
* Otros recursos
* etc.

## Análisis de mercado

Estudio de mercado que incluya su audiencia potencial, segmentación, competencia, márgenes de precios, etc.

## Marketing y ventas

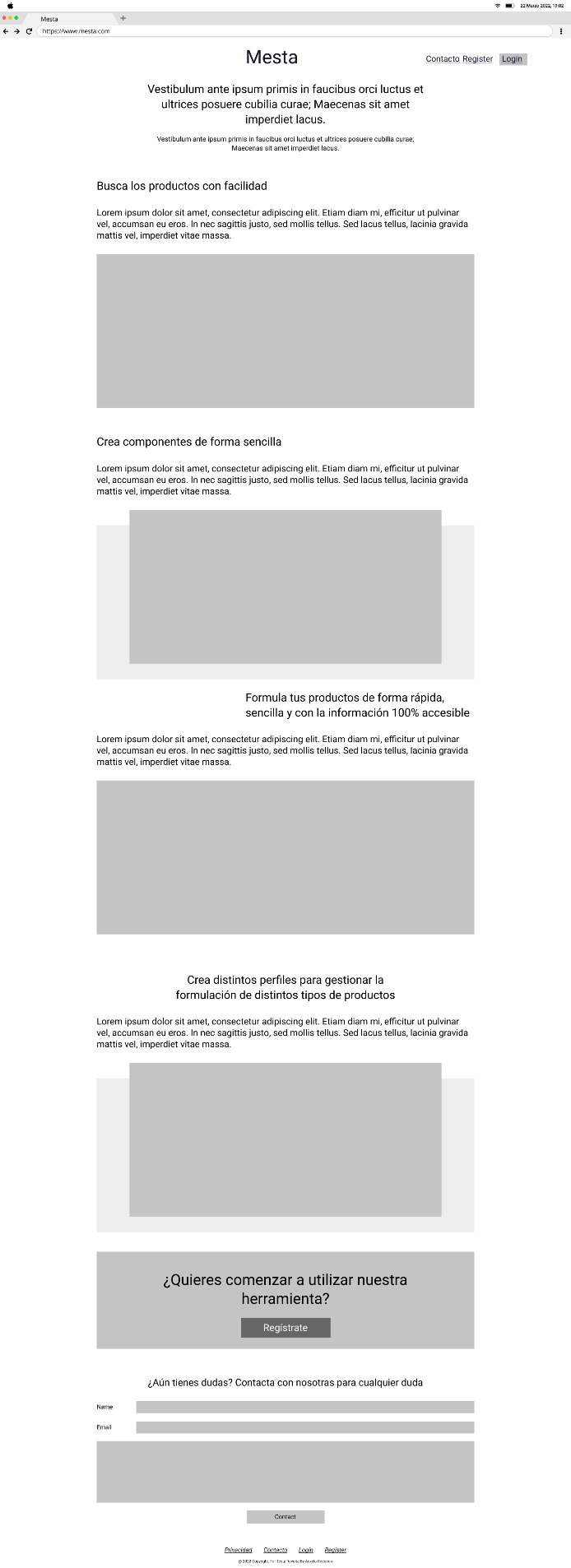
Enfoque del trabajo en el plano de su branding y plan de promoción así como su política de precios y estrategia de venta.

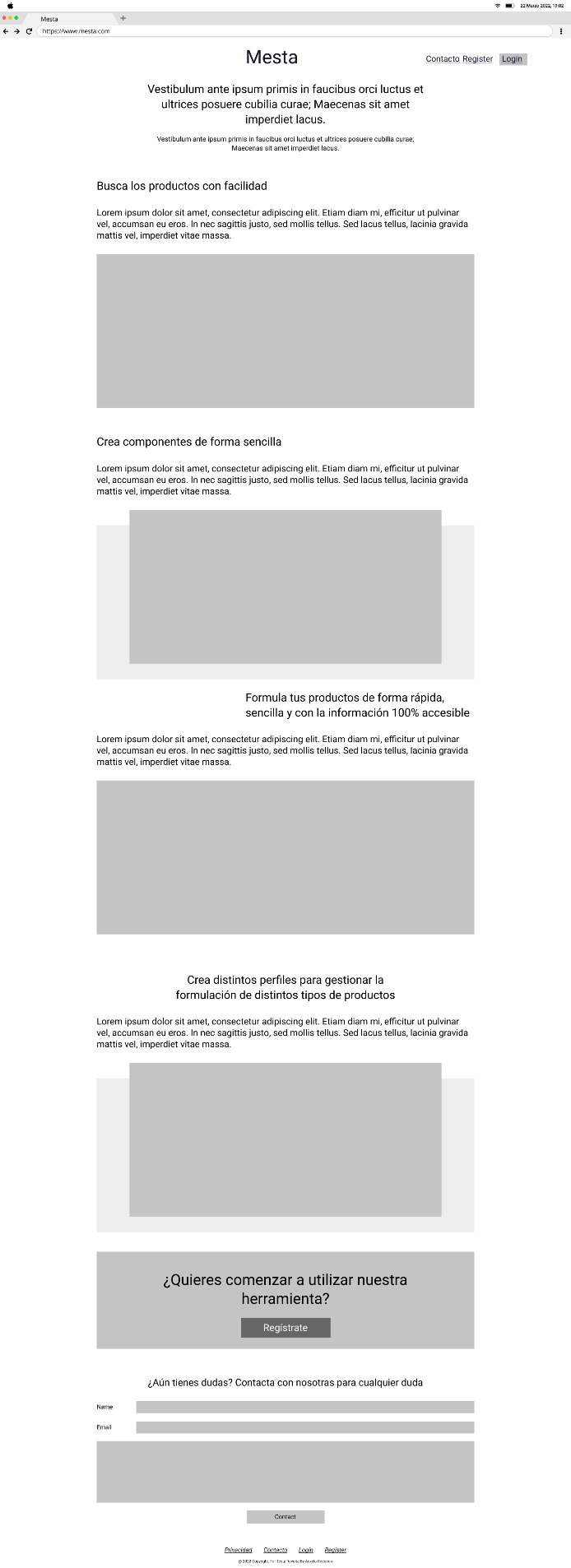
# Conclusiones

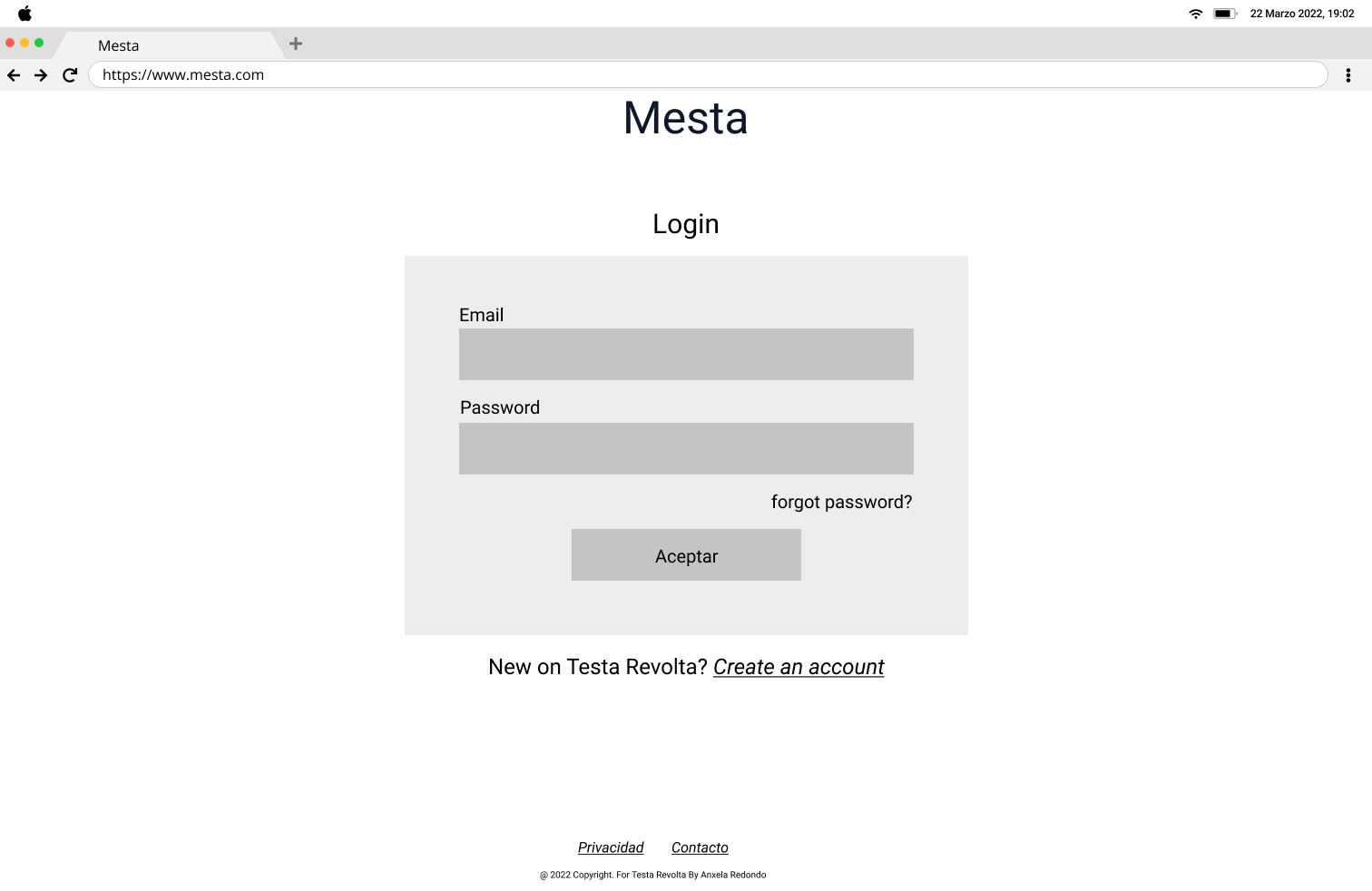
1. Entregables del proyecto

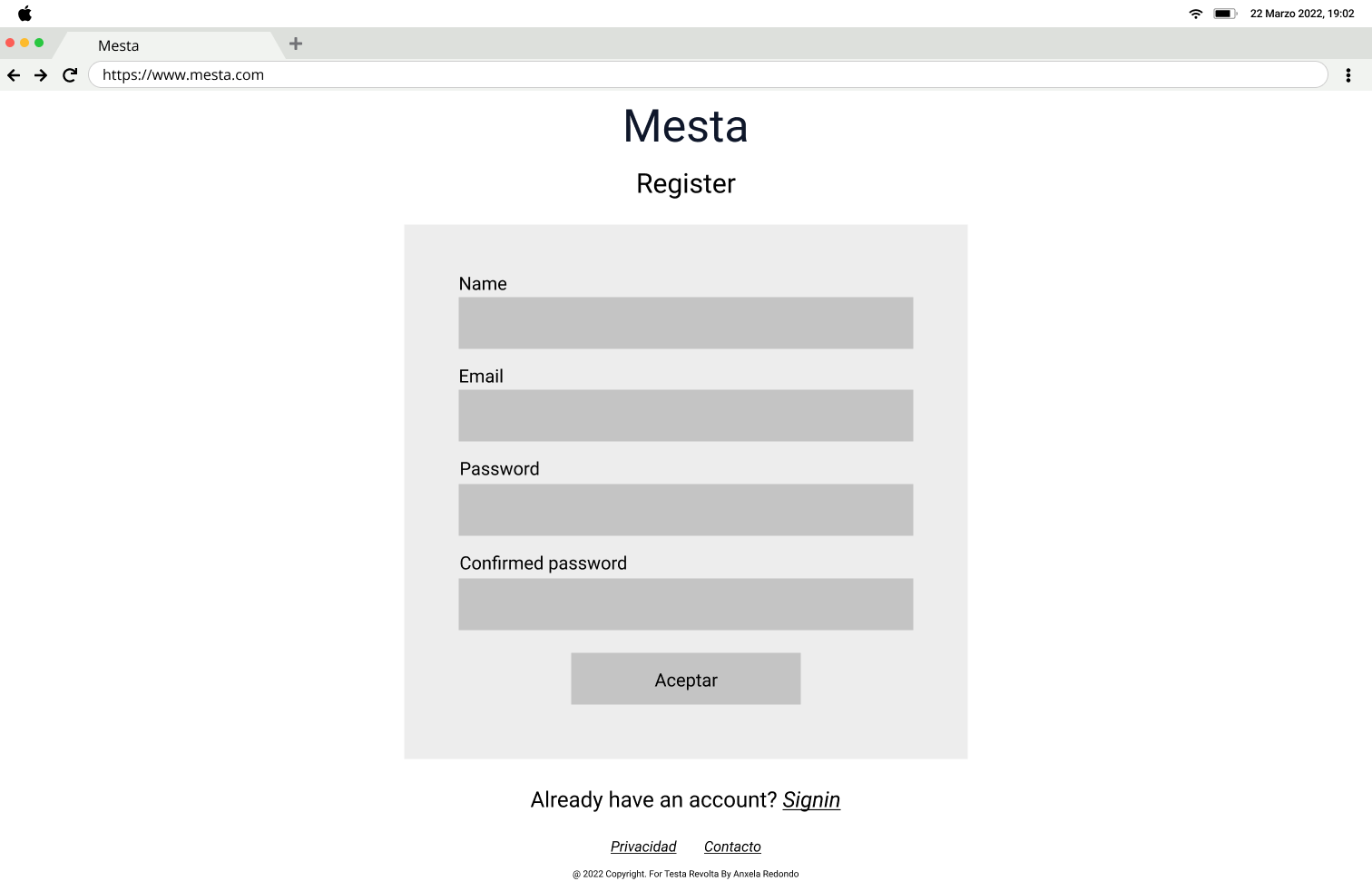
Lista de archivos entregados y su descripción.

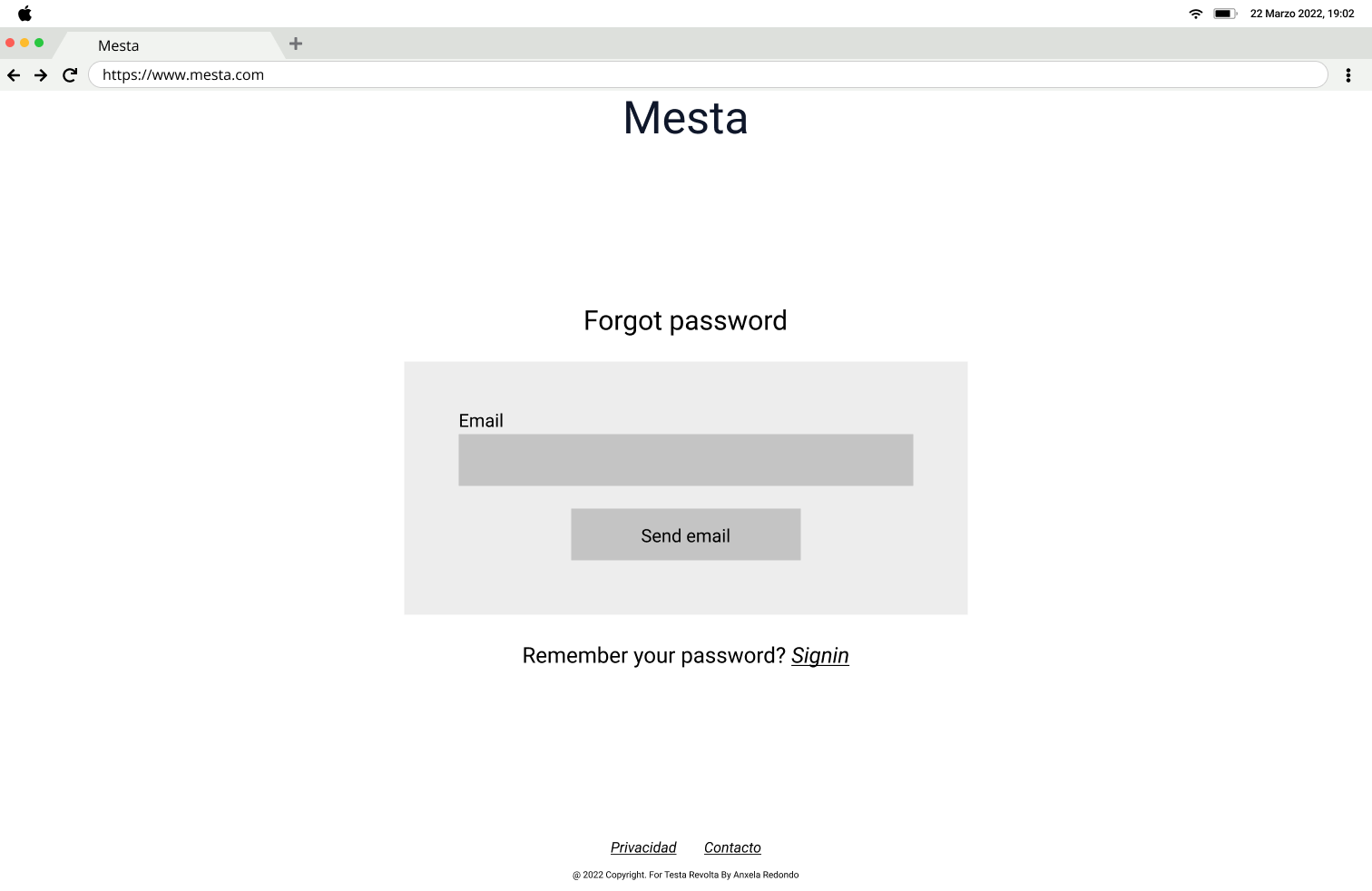
Repositorio GitHub <https://github.com/anxela-re/mesta>

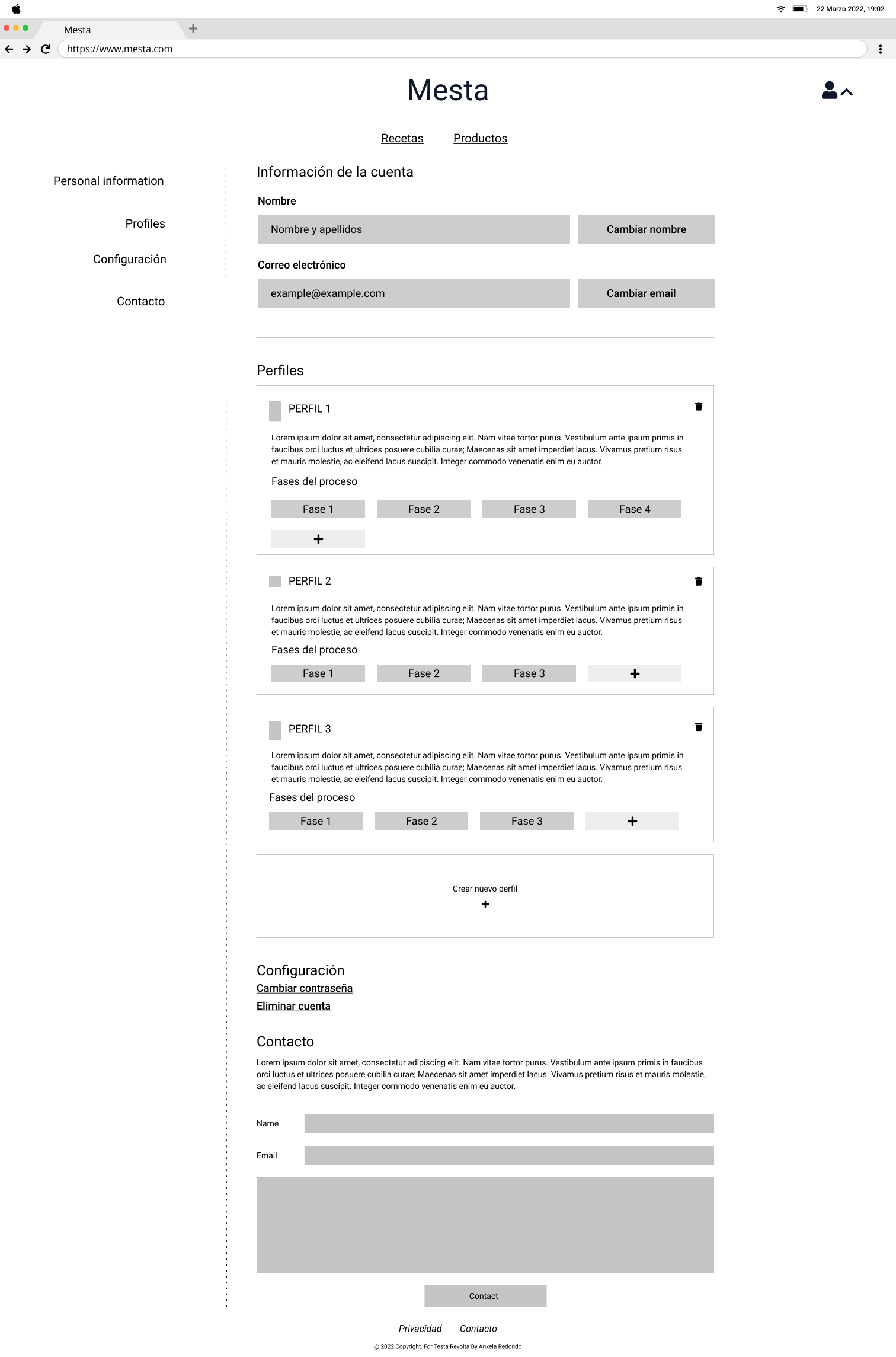
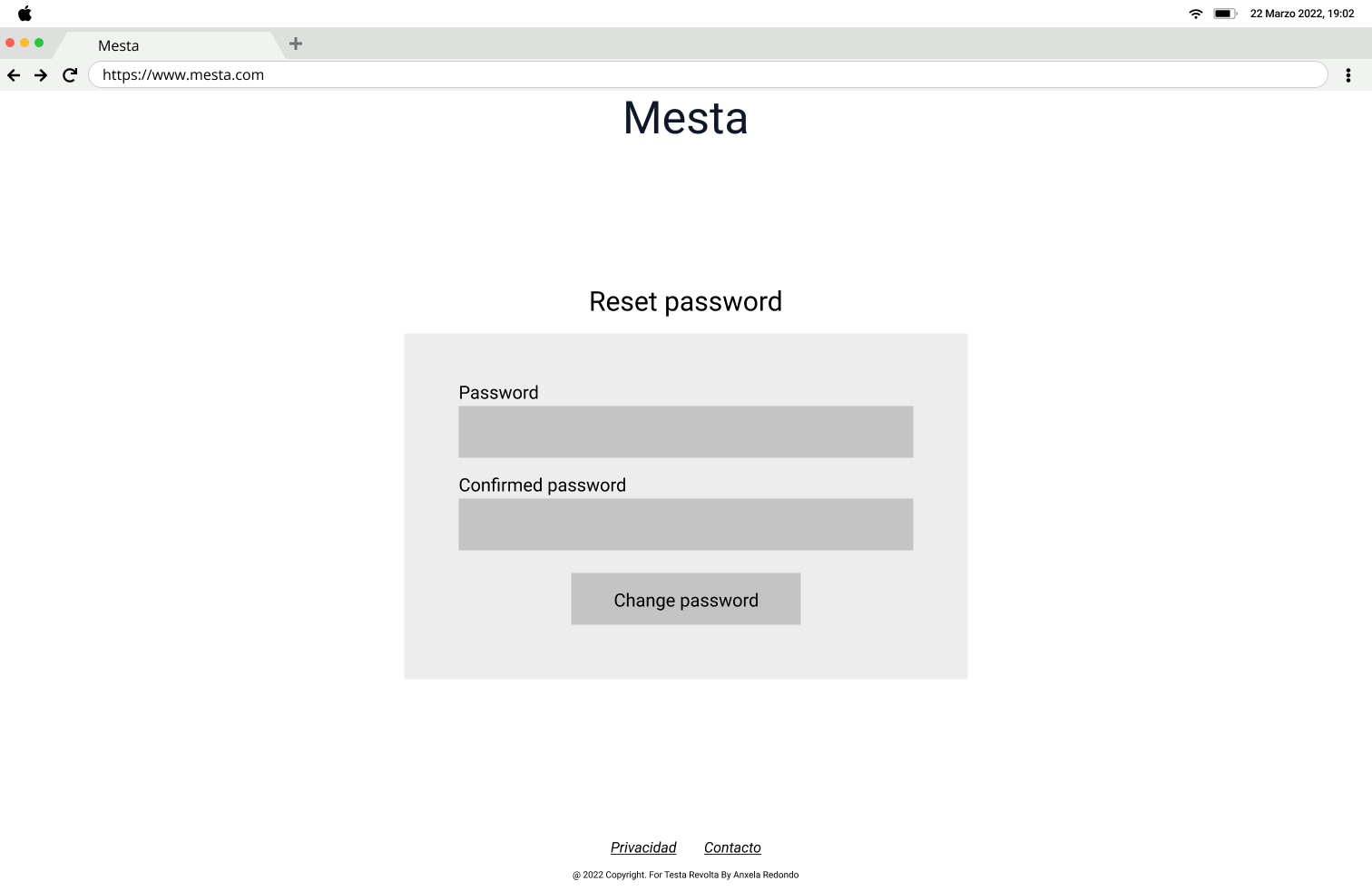
1. Prototipo de baja calidad

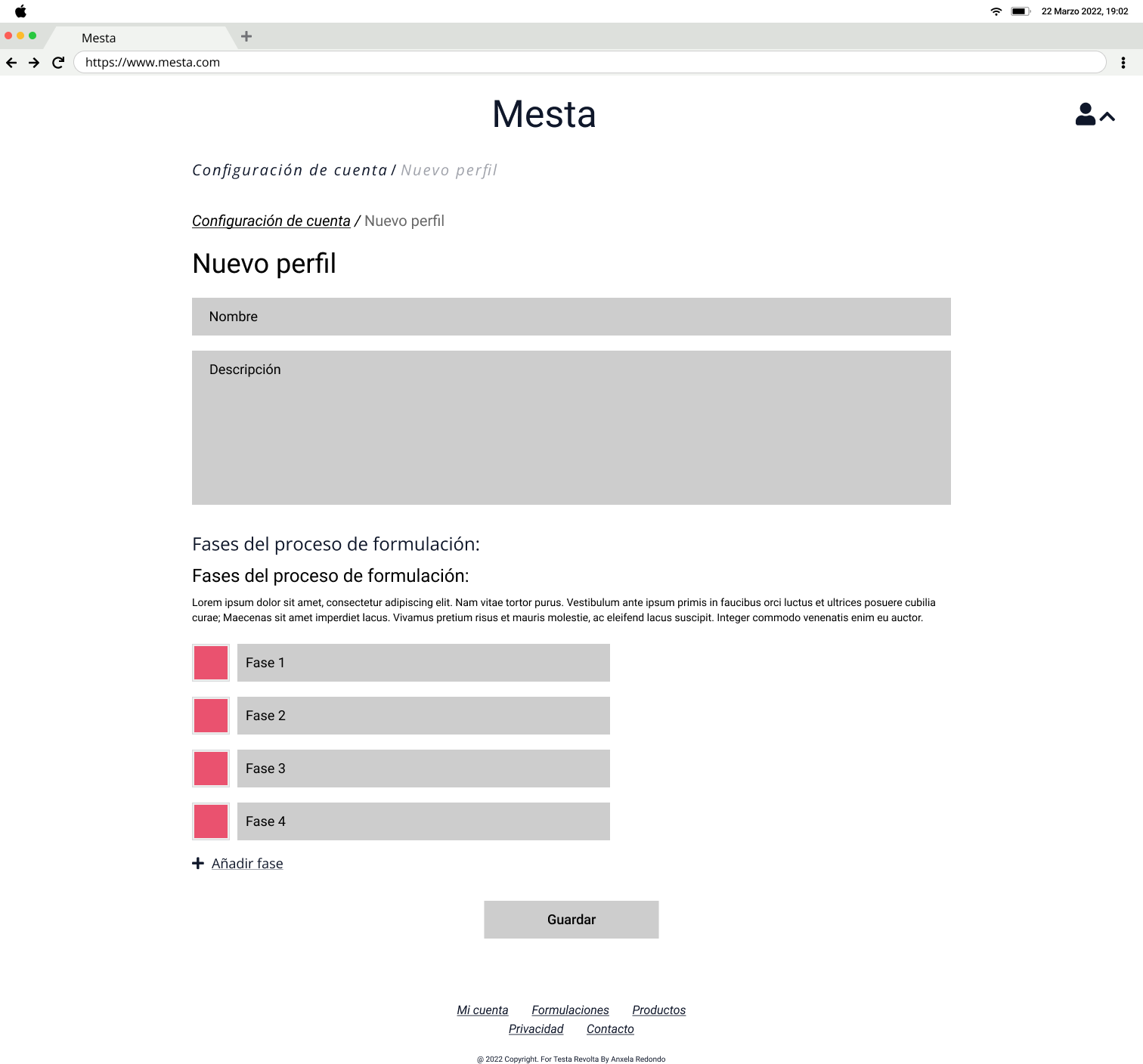


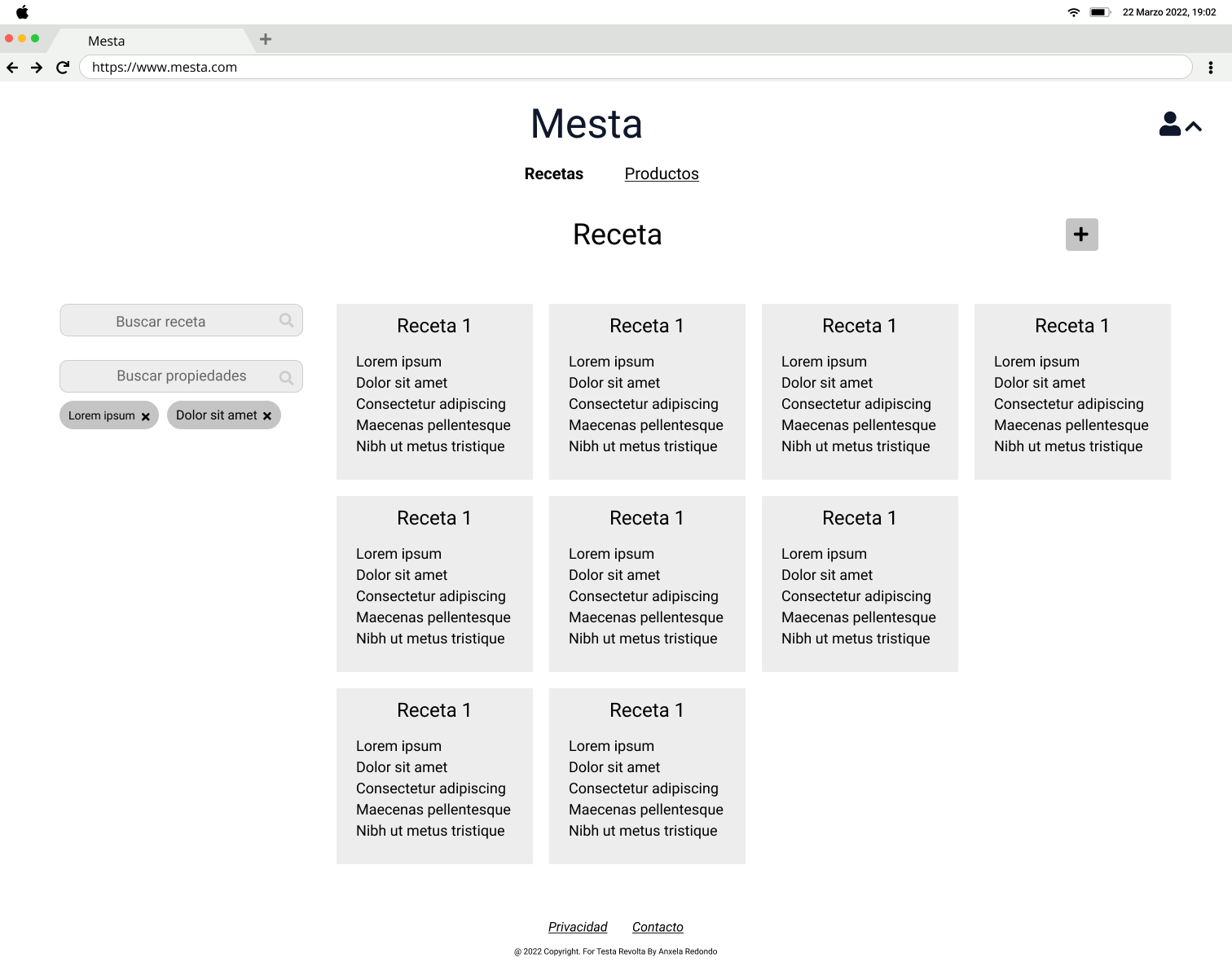


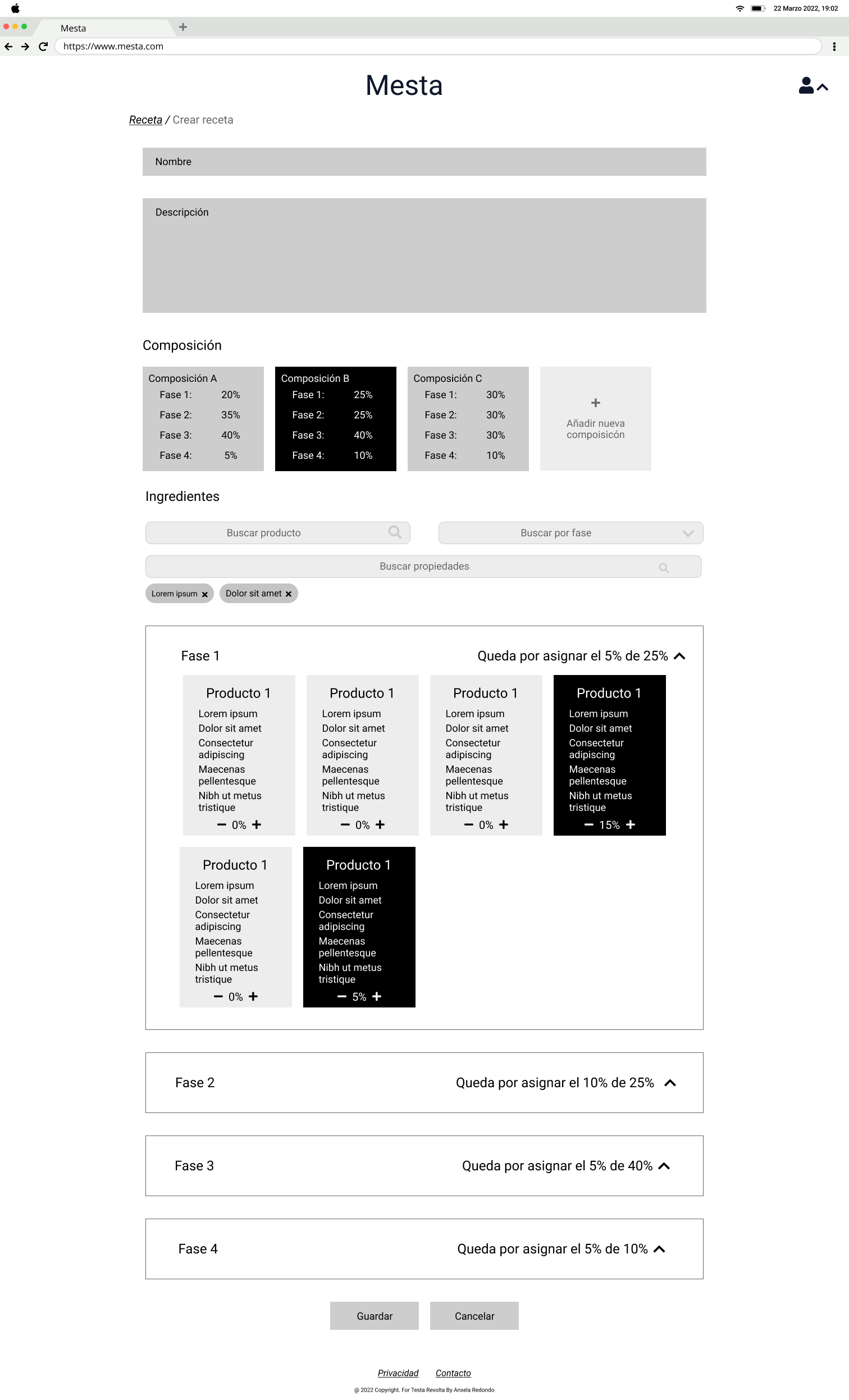


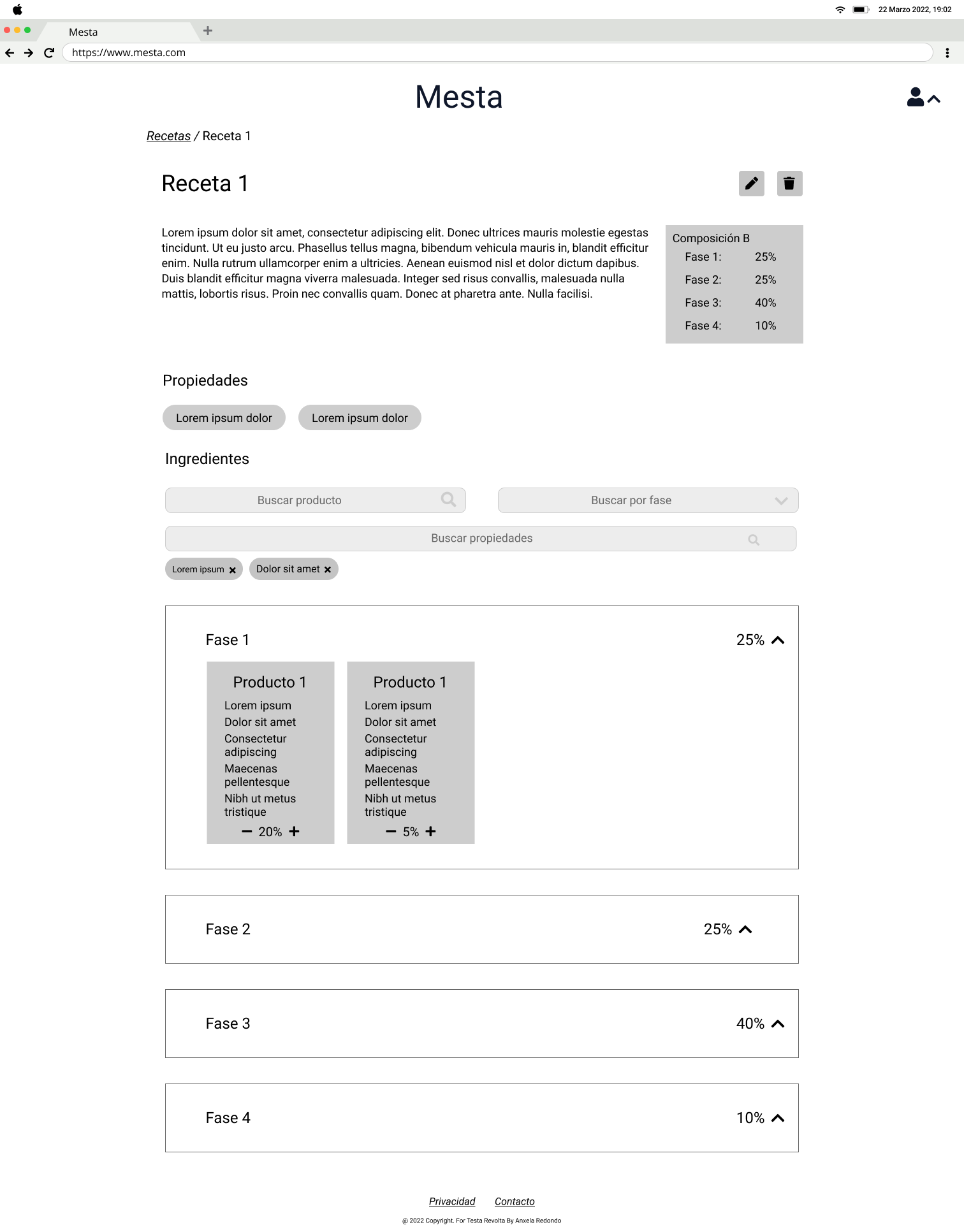


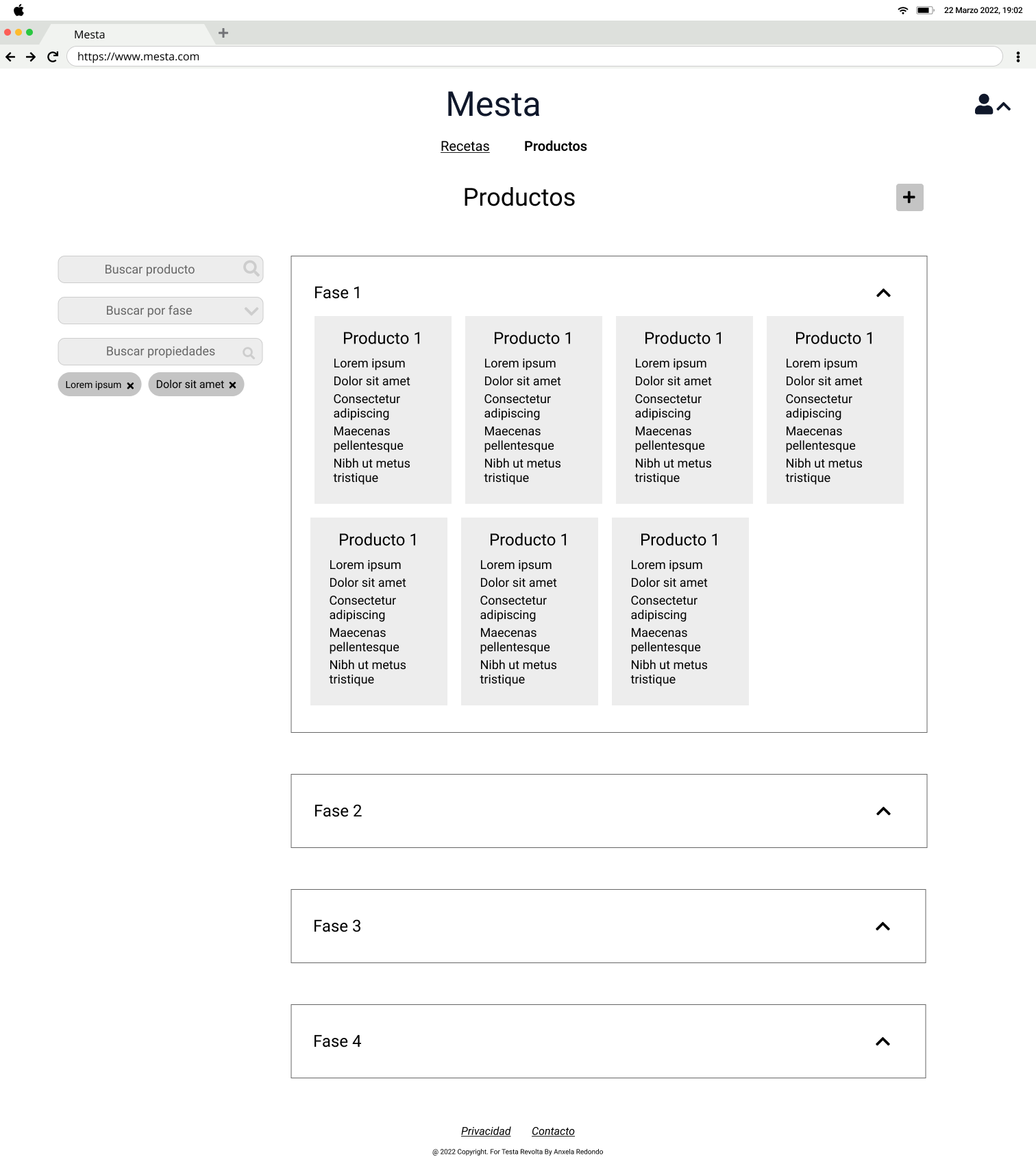


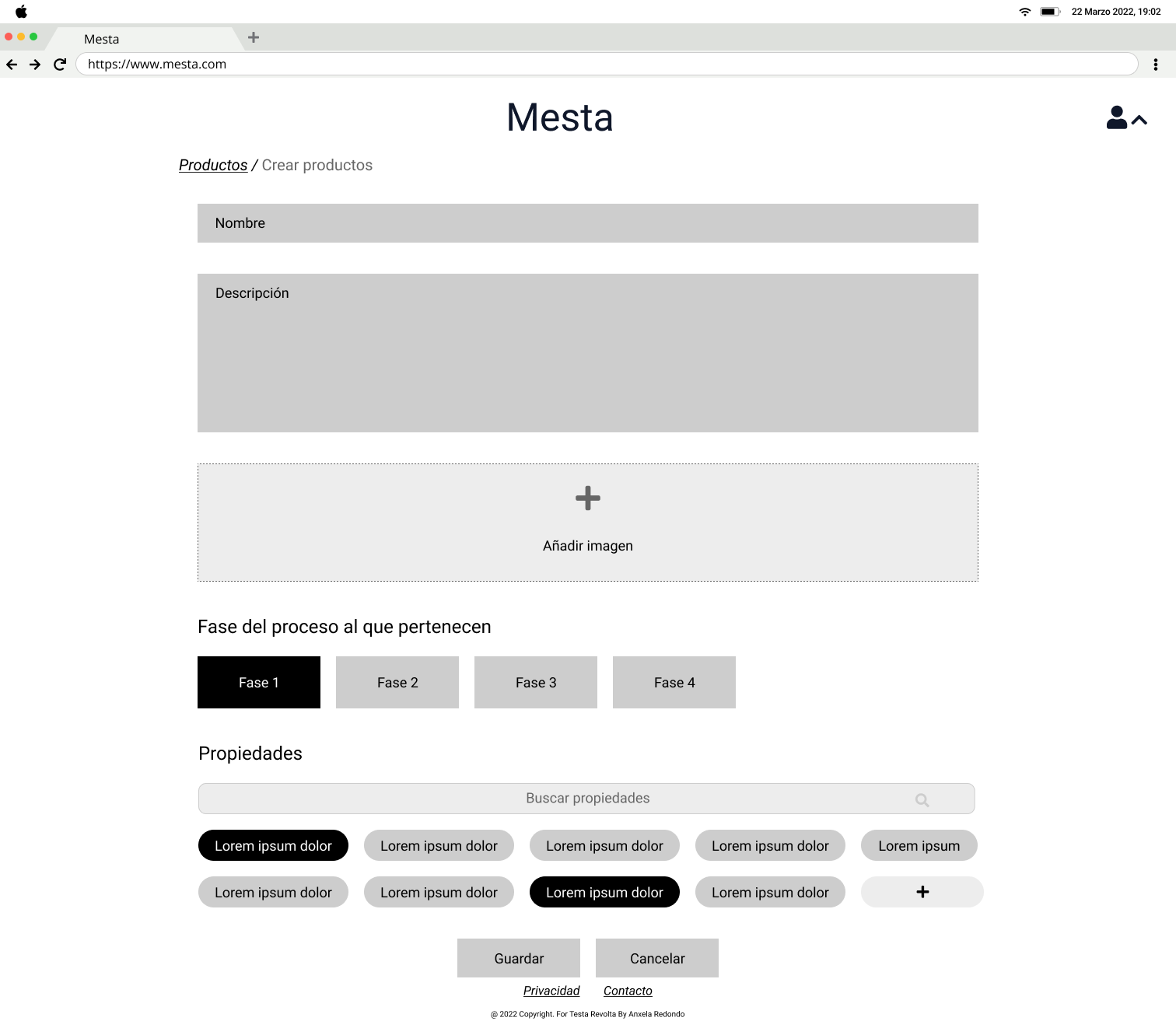


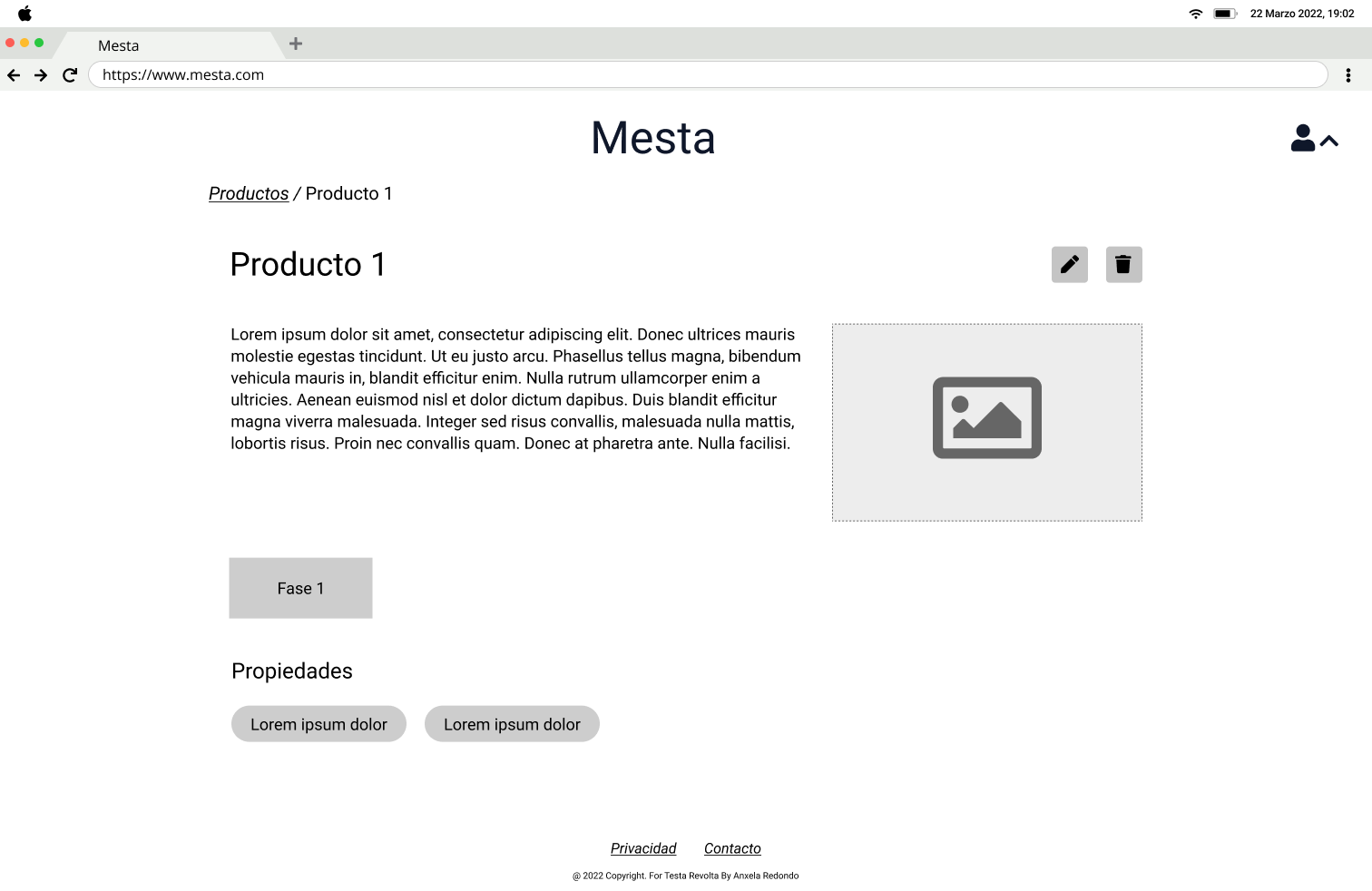












1. Prototipos de alta calidad

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

1. Código fuente frontend
2. Código fuente backend

Selección de partes relevantes del código fuente del servicio/aplicación creado, descripción y comentarios.

1. Librerías externas utilizadas

Las librerías utilizadas para el desarrollo del front-end de la aplicación fueron:

|  |  |
| --- | --- |
| **FRONT-END** | |
| **Fontawesome** | <https://fontawesome.com/> |
| Librería de iconos que te ofrece imágenes de vectores escalables con posibilidad de customizarlas. | |
| **NgRx** | <https://ngrx.io/> |
| Librería que permite implementar la arquitectura Redux en proyectos desarrollados con el framework de Angular. Esta librería incluye: la librearía ngrx/effects, ngrx/store y ngrx/devtools | |
| **Ngx-colors** | <https://ngx-colors.web.app/> |
| Componente de angular que permite la selección de un color con distintas posibilidades. | |
|  | |

|  |  |
| --- | --- |
| **BACK-END** | |
| **Passport** | [https://laravel.com/docs/9.x/passport/](https://fontawesome.com/) |
| Laravel Passport es una librería que permite la implementación de un servicio de autenticación para proyectos realizados con Laravel. | |
|  | |

1. Captura de pantalla

Capturas de pantalla tanto del trabajo/servicio/aplicación realizado como del proceso de trabajo. Este anexo también puede utilizarse para recopilar las capturas mostradas en otras secciones, en mayor tamaño para su mejor visualización, o no ser necesario su uso por el tipo de trabajo realizado.

1. Guía de usuario

Información detallada acerca de cómo utilizar el servicio/aplicación/trabajo realizado. Esta guía debe estar a disposición del usuario.

1. Libro de estilo

|  |
| --- |
| **Paleta de colores** |
|  |
| **Tipografías y fuentes** |
| Logotipos y anagramas  Arquitectura y proyecto |
| **Iconos** |
| |  |  |  |  | | --- | --- | --- | --- | |  |  |  |  | |

En la parte derecha la temática oscura y la izquierda la clara

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logotipos** | | | | |
|  | | |  | |
| **Anagramas** | | | | |
|  | | |  | |
| **Fondos** | | | |
|  |  | | |
| **Botones habilitados** | | | |
|  | |  | |
| **Botones deshabilitados** | | | |
|  | |  | |
| **Item** **no seleccionado** | | | |
|  | |  | |
| **Ítem seleccionado** | | | |
|  | |  | |
| **Input buscador** | | | |
|  | |  | |
| **Botones añadir, edita y eliminar** | | | |
|  | |  | |
| **Menu navegación (link actual: componentes)** | | | |
|  | |  | |
| **Ítem aplicación** | | | |
|  | |  | |
| **Modal** | | | |
|  | |  | |
| **Toast errór y toast éxito** | | | |
|  | | | |
| **Input por defecto, input *focus* e input erróneo** | | | |
|  | | | |

1. Resumen ejecutivo

Micro-plan de empresa de una página de extensión, sin contar la titulación de capítulo y con el formato de texto que el alumno considere. En su realización ha de tenerse en cuenta que debe poder presentarse como documento individual. Debe resumir, como mínimo, los siguientes aspectos, propios de un plan de empresa completo:

* Nombre comercial
* Resumen comercial
* Modelo de negocio
* Expertise
* Productos y servicios
* Mercado
* Competencia
* Plan de marketing
* Inversión inicial y costes a corto y medio plazos
* Proyección económica corto y medio plazos y ROI
* DAFO

1. Glosario

Glosario de términos y acrónimos utilizados en el trabajo (sólo aquellos mencionados en el presente documento) con breves definiciones de cada uno de ellos, o un índice analítico con la lista de términos, nombres y palabras clave en el texto y las páginas donde se pueden encontrar.

1. Bibliografía

<http://js-programadors.aula.uoc.edu/es/5-2-1-que-es-una-api/>

<https://es.wikipedia.org/wiki/Transferencia_de_Estado_Representacional>

<https://www.institutodermocosmetica.com/como-se-hace-un-champu-solido-natural/>

<https://es.wikipedia.org/wiki/Identificador_de_recursos_uniforme>

<https://www.arsys.es/blog/redux-datos-aplicaciones>

<https://web.dev/progressive-web-apps/>

<https://developers.google.com/web/fundamentals?hl=en>

<https://infrequently.org/2015/06/progressive-apps-escaping-tabs-without-losing-our-soul/>

<http://cvapp.uoc.edu/autors/MostraPDFMaterialAction.do?id=231265>

<https://es.wikipedia.org/wiki/XAMPP>

<https://laravel.com/docs/7.x#server-requirements>

<https://www.mysql.com/support/supportedplatforms/database.html>

Bibliografía de aquellas publicaciones mencionadas en el documento. Las referencias en el texto pueden apuntar directamente a los ítems de este anexo. Si el alumno prefiere anotar las referencias bibliográficas al pie de la página en las que se usan, deberá igualmente referenciar todas las publicaciones en este anexo.

A continuación, se ha añadido una base bibliográfica al documento y se han insertado dos registros, un libro y un artículo online, como ejemplos. Ambos están referenciados en el capítulo Marco teórico/Escenario del presente documento. En caso de utilizar el editor Word de Microsoft, pueden verse las funciones de Referencia del mismo y la documentación de ayuda de dicho programa para conocer su uso.

Anderson, C. (24 de 4 de 2012). The Man Who Makes the Future: Wired Icon Marc Andreessen.Obtenido de Wired: http://www.wired.com/epicenter/2012/04/ff\_andreessen/

Manovich, L. (2011). The Language of New Media.Cambridge: MIT Press.

1. Vita

Sección opcional aunque recomendable.

Breve nota biográfica del autor del TF. Máximo 700 caracteres.