

Aplicación web para la gestión de consultas médicas

Memoria de Proyecto Final de Máster
Desarrollo de Sitios y Aplicaciones Web
Itinerario profesional

Autor: Alejandro Carmona Martos

Consultor: Anna Ferry Mestres
Profesor: César Pablo Córcoles Briongos
Profesor: Julià Minguillón Alfonso

Junio de 2019

Créditos

Licencia



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

Dedicatoria

Gracias a mi familia, en especial a mis padres que siempre han sido un gran apoyo y un punto de referencia.

Gracias a Guillermo por su paciencia, ayuda, comprensión y motivación para poder terminar este proyecto.

Alejandro Carmona Martos

Abstract

El objetivo del presente proyecto es el desarrollo y documentación de una aplicación web para gestionar las consultas médicas privadas. De forma autónoma, podrán organizar su propia consulta donde encontrarán la posibilidad de crear un historial clínico, anotar la evolución del paciente, datar los datos analíticos, gestionar informes e impresión de orientaciones médicas para el paciente.

Esta aplicación web cuenta con un diseño responsive o adaptable a cualquier tamaño de pantalla, de esta manera los usuarios se podrán conectar con diferentes dispositivos, ya sean ordenadores, móviles o tabletas.

En el desarrollo de la aplicación se han utilizado principalmente los frameworks, Laravel para el back-end y Angular para el front-end, además de realizar la maquetación en HTML5 y CSS3.

Palabras clave:

Aplicación web, médicos, consulta médica, gestión de consultas, informes clínicos, Laravel, Angular

Abstract (English version)

The objective of this project is the development and documentation of a web application to manage private medical consultations. Autonomously, they can organize their own consultation where they will find the possibility of creating a clinical record, record the evolution of the patient, date the analytical data, manage reports and print medical guidelines for the patient.

This web application is made with responsive design. It fits any screen size, so users can connect with different devices, like computers, mobile phones or tablets.

In the development of the application. We have used the framework Laravel for the back-end and Angular for the front-end, in addition to the layout in HTML5 and CSS3.

Keywords:

Web application, doctors, medical consultation, medical consultation management, clinical reports, Laravel, Angular

Índice

1. Introducción	8
1.1 Contexto y justificación del trabajo	8
2. Descripción	9
3. Objetivos.....	10
3.1 Principales.....	10
3.2 Secundarios.....	10
4. Contenidos.....	11
5. Metodología	12
6. Arquitectura de la aplicación.....	13
6.1 Listado de carpetas propuesta para el desarrollo	13
6.2 Modelo de datos	14
7. Plataforma de desarrollo	15
8. Planificación.....	16
9. Prototipos.....	18
9.1 Lo-Fi.....	18
9.2 Hi-Fi.....	32
10. Usabilidad/UX	41
10.1 Justificación patrones de diseño comunes en la aplicación.....	41
10.2 Justificación patrones de diseño de la estructura.....	42
10.3 Justificación patrones de diseño del flujo de acceso / registro de usuario.	43
10.4 Justificación patrones de diseño de flujo para añadir nuevo paciente.	43
10.5 Árbol de navegación	44
11. Seguridad.....	45
12. Requisitos de implantación y uso.....	46
13. Instrucciones de implantación.....	47
14. Instrucciones de uso de la aplicación.....	48
15. Proyección a futuro	51
16. Presupuesto.....	52
17. Análisis de mercado	53
18. Conclusiones	55
Anexo 1. Entregables del proyecto	56
Anexo 2. Código fuente (extractos)	58
Anexo 3. Librerías/Código externo utilizado.....	61
Anexo 4. Libro de estilo.....	62
Anexo 5. Bibliografía	64

Anexo 6. Vita	65
---------------------	----

Figuras y tablas

Lista de imágenes, tablas, gráficos, diagramas, etc., numeradas, con títulos y las páginas en las que aparecen.

Índice de figuras

Figura 1: Modelo de Base de datos de la aplicación	14
Figura 2: Diagrama de Gant.....	17
Figura 3: Wireframe de la página de inicio en versión escritorio	18
Figura 4: Wireframe acceso a la aplicación	19
Figura 5: Wireframe de registro de usuario a la aplicación.	20
Figura 6: Wireframe del flujo de registro de usuario versión escritorio.	21
Figura 7: Wireframe de la página de dashboard versión escritorio.....	22
Figura 8: Wireframe de la página de inicio versión tablet.....	23
Figura 9: Wireframe del flujo de registro de usuario versión tablet.....	24
Figura 10: Wireframe de la página de dashboard versión tablet	25
Figura 11: Wireframe de la página de inicio versión móvil	26
Figura 12: Wireframe del flujo de registro de usuario versión móvil	27
Figura 13: Wireframe de la página de dashboard versión móvil.....	28
Figura 14: Wireframe del flujo de añadir nuevo paciente versión escritorio	29
Figura 15: Wireframe del flujo de añadir nuevo paciente versión tablet.....	30
Figura 16: Wireframe del flujo de añadir nuevo paciente versión móvil	31
Figura 17: Mockup página de inicio versión escritorio	32
Figura 18: Mockup página de inicio versión tablet.....	33
Figura 19: Mockup página de inicio versión móvil	34
Figura 20: Mockup página de añadir paciente versión escritorio.....	35
Figura 21: Mockup página de añadir paciente versión tablet	36
Figura 22: Mockup página de añadir paciente versión móvil.....	37
Figura 23: Mockup página de paciente versión escritorio	38
Figura 24: Mockup página de paciente versión tablet.....	39
Figura 25: Mockup página de paciente versión móvil	40
Figura 26: Árbol de navegación de la aplicación web.....	44

Índice de tablas

Tabla 1: Software utilizado en el desarrollo del proyecto.....	15
Tabla 2: Web app utilizadas en el desarrollo del proyecto	15
Tabla 3: Hardware utilizado para el desarrollo del proyecto.....	15
Tabla 4: Planificación del trabajo.....	16
Tabla 5: Presupuesto equipo humano	52
Tabla 6: Presupuesto equipamiento técnico	52

1. Introducción

1.1 Contexto y justificación del trabajo

En esta memoria se ha querido plasmar, a modo de guía, el proyecto de una aplicación web que solucione un problema que se pueden encontrar los médicos cuando quieren saltar al mundo privado de las consultas médicas y necesitan una herramienta para gestionar los pacientes en su consulta.

Los médicos en España suelen trabajar en el ámbito de la sanidad pública, aunque a menudo compatibilizan esa función con una práctica privada, o bien en hospitales, compañías o empresas, o bien como autónomos creando su propia consulta. Este proyecto se ha centrado en este último grupo, los autónomos.

A pesar de que se puede encontrar en el mercado de consumo diferentes aplicaciones web relacionadas con la gestión de pacientes, estas aplicaciones traen demasiadas soluciones que realmente no llegan a utilizarse. Es por ello que nos hemos querido enfocar en un aspecto más reducido de la gestión de las consultas.

En estas páginas se encontrarán los objetivos, metodologías y herramientas para el desarrollo de nuestra aplicación web de gestión de consultas médicas.

2. Descripción

Se propone como Trabajo Fin de Master, en adelante TFM, el diseño y desarrollo de una aplicación web sobre gestión de consultas médicas. Dicha aplicación estará destinada a profesionales del sector de la medicina con el fin de organizar la consulta médica privada, proporcionando contenidos organizativos e intercalando textos con elementos visuales que dotarán a la aplicación de mayor interés y atractivo para el usuario.

De esta forma, la idea principal de la aplicación será la creación de una aplicación web de gestión y organización de consultas médicas, donde el médico, tendrá la posibilidad de gestionar a sus pacientes teniendo elementos que le permitan gestionar las consultas diarias, historia clínica o generar informes médicos.

Al ser un proyecto totalmente escalable, es fantástico para poder comenzar con una primera versión centrándonos en los elementos principales. Cabe destacar las funcionalidades en las que se quiere centrar esta versión, ordenadas por orden de importancia.

- Crear la historia clínica del paciente.
- Recoger los datos del paciente en cada consulta.
- Crear informes del paciente.
- Gestionar los datos analíticos del paciente.
- Imprimir hoja para paciente con la información del tratamiento.

El uso de internet puede aportar multitud de beneficios para la sociedad, pero al tener datos tan críticos y privados de los pacientes tenemos que tener siempre un cierto grado de prudencia en la seguridad, por ello, se creará con los frameworks de Laravel y Angular, que, dotarán de una seguridad extra, haciéndola confiable y segura.

El diseño de la aplicación está orientado al uso en multidispositivos, con el fin de poder usarse en cualquier parte a través de ordenadores, Tablet o Smartphone, y con una interfaz amigable, sencilla e intuitiva que no requiera de un alto nivel de conocimientos de nuevas tecnologías.

El TFM también aportará el desarrollo de la aplicación, junto con la implementación en un servidor web y las instrucciones de uso de la misma.

Por concluir, se pretende realizar el diseño y desarrollo de una aplicación web que contemple la organización y gestión de consultas clínicas para personal médico que trabaje de forma autónoma en consultas privadas. Realizando una versión beta de su interfaz y desarrollo. El nombre comercial elegido es: “*skipu*”, organizado en islandés. Se buscaba un nombre corto y que tuviera un nexo con la aplicación.

3. Objetivos

3.1 Principales

Los objetivos generales del proyecto se estructuran en los siguientes puntos:

- Analizar y desarrollar una aplicación web que permita facilitar la gestión de pacientes en una consulta médica.
- Documentar cada una de las fases del desarrollo de una aplicación web, desde la idea del proyecto, pasando por las distintas fases de análisis del mercado, diseño, desarrollo y publicación de la misma.

3.2 Secundarios

Objetivos adicionales que enriquecen el TF y que pueden sufrir variaciones.

- Profundizar en el conocimiento de las tecnologías para el desarrollo de un proyecto web, Laravel en el apartado de back-end y Angular en el apartado front-end.
- Diseñar una interfaz que se adapte a todo tipo de dispositivos.
- Adecuación de la planificación en el tiempo empleado.
- Presupuesto empleado en la creación de la aplicación de esta envergadura.

4. Contenidos

La aplicación web tendrá una composición clásica en cuanto a su **estructura web** formada por cabecera, cuerpo y pie de página.

- **Cabecera:** Contendrá el Logo de la aplicación web, menú de información y zona de acceso y registro.
- **Cuerpo:** Será donde recaiga todo el peso de la aplicación. La pantalla se dividirá en dos zonas de interacción, menú administración de la aplicación y zona de visualización de datos, formularios, etc.
- **Pie de página:** Dispondrá de todos los enlaces de información legal y el logotipo de la aplicación.

Esta estructura se respetará durante todo el proyecto en las distintas páginas. A continuación, se podrán analizar los contenidos de dichas páginas principales.

- **Inicio:** Página donde se podrá acceder antes del registro o acceso del usuario. Contendrá información comercial referente a la aplicación y la zona tanto de acceso, como de registro del usuario.
- **Dashboard:** Panel de administración del usuario, contendrá un menú en la parte izquierda, donde, se podrán visualizar todas las opciones para interactuar con la aplicación. En la parte central aparecerá un resumen con la información más importante como los últimos pacientes.
- **Perfil usuario:** Página donde el usuario podrá editar su información.
- **Consulta médica:** Aparecerá la información de la consulta y los pacientes que la componen.
- **Paciente:** Una de las páginas más importantes de la aplicación. Se puede encontrar el menú de la aplicación en la parte izquierda. En la parte central se podrán visualizar todos los datos del paciente, además, se podrá acceder a las distintas secciones de historia clínica, evolución del paciente, tratamientos e informes.
- **Historia clínica:** Encontraremos la información referente a la historia clínica del paciente. Antecedentes, listado de exploraciones, juicio clínico, diagnóstico, etc.
- **Evolución:** Será la ficha donde se incorporan los datos cuando se pase consulta al paciente.
- **Informes:** Página donde se listarán todos los informes del paciente, incluyendo la posibilidad de añadir un nuevo informe.

5. Metodología

A partir de la idea inicial se han estudiado los requisitos necesarios para poder desarrollar la aplicación. Se ha efectuado tanto el inicio de la documentación como la planificación de todo el proyecto, está será nuestra primera fase del proyecto.

Con la información obtenida, comenzaremos la segunda fase en la que efectuaremos el mapa web, donde se definirán las pantallas y el flujo que tendrá nuestra aplicación. Posteriormente, se crearán diferentes bocetos y prototipos para que podamos realizar el diseño final de nuestra aplicación. En esta fase, por último se definirá nuestro modelo de base de datos que nos ayudará en la parte de desarrollo.

A partir de entonces se comenzará a trabajar con las tecnologías escogidas. Se creará el espacio de trabajo de modo local. Además, se hará uso de GitHub para poder tener un control de versiones y mantener a salvo el código del proyecto.

Una vez finalizada la preparación del espacio, se pasará a desarrollar el gestor de consultas médicas. Incluyendo test de funcionamiento y la documentación de todos los procesos ejecutados.

En la última parte del proyecto se corregirán todos los errores que pueda tener la aplicación, así dejaremos una versión 1.0, eso sí, con posibilidad de mejora. Además de terminar la documentación y memoria del proyecto, se creará la presentación del trabajo.

6. Arquitectura de la aplicación

La arquitectura de la aplicación estará fundamentada en el desarrollo de una app con Angular y Laravel como frameworks de desarrollo.

- Angular nos ayudará para el front-end, parte de la aplicación que interactúa con el usuario, donde nos facilitará la interacción del usuario con el navegador.
- Laravel como base para servir una API REST.

La arquitectura de la aplicación se desarrollará a través de una API REST, donde la interfaz mediante peticiones HTTP obtendrá o generarán datos en cualquier formato como XML o JSON.

Características de REST para el proyecto:

- **Protocolo sin estado:** donde se trata cada petición HTTP de forma independiente y contiene toda la información necesaria para ejecutarla, lo que permite que ni servidor ni cliente necesiten recordar ningún estado previo.
- Las **operaciones** que se van a emplear con los datos del sistema REST y la especificación HTTP son cuatro: POST (crear), GET (leer y consultar), PUT (editar) y DELETE (eliminar).
- Los objetos siempre se manipulan a partir de la **URI (Identificador de recursos uniforme)**. Es la URI el identificador único de cada recurso del sistema REST. Nos facilitará acceder a la información para la modificación, borrado, etc.
- **Interfaz uniforme:** Necesaria para la transferencia de datos en el sistema REST, se aplicarán acciones concretas (POST, GET, PUT y DELETE) sobre los recursos, siempre y cuando estén identificados con una URI. Esto facilitará la existencia de una interfaz uniforme que sistematiza el proceso con la información.
- **Sistema de capas:** arquitectura jerárquica entre los componentes. Cada una de estas capas lleva a cabo una funcionalidad dentro del sistema REST.
- **Uso de hipermedias:** El concepto de hipermedia proporcionará a la interfaz de desarrollo la capacidad de proporcionar al cliente y al usuario los enlaces adecuados para ejecutar acciones concretas sobre los datos.

6.1 Listado de carpetas propuesta para el desarrollo

Shared

--- models

--- services

Views

--- Singin

- Singup
- Dashboard
- Profile
- Paciente
 - Consulta
 - Historia
 - Tratamiento
 - Evolución

6.2 Modelo de datos

En cuanto al modelo de datos usado inicialmente será el propuesto en la siguiente figura.

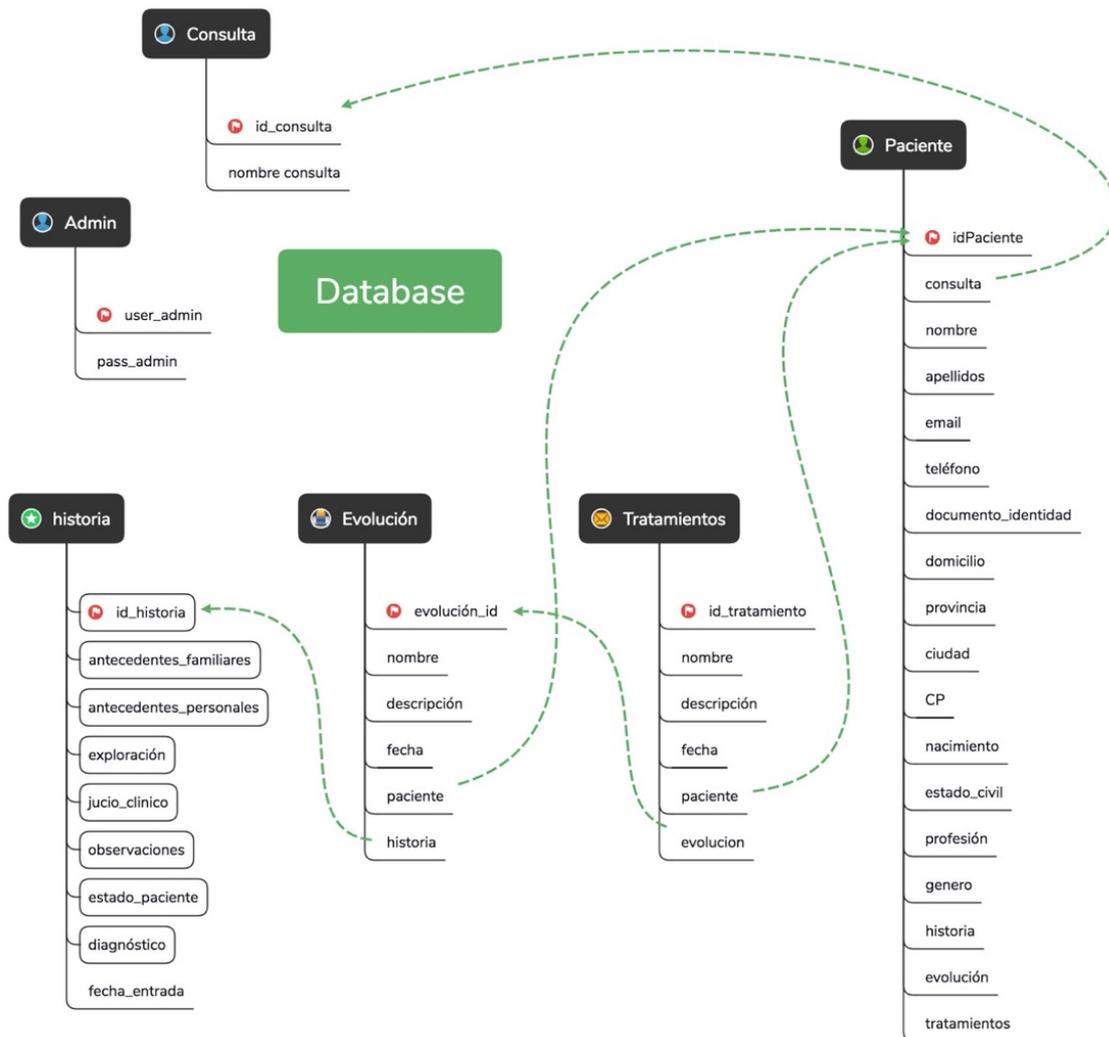


Figura 1: Modelo de Base de datos de la aplicación

7. Plataforma de desarrollo

Para el desarrollo del proyecto se han utilizado los siguientes recursos tecnológicos:

Software	
IDE Desarrollo	WebStorm
	PhpStorm
Servidor Local	MAMP
Servidor Web	Servidor compartido Ionos
Maquetado documentación	Microsoft Word
Diseño Gráfico	Photoshop
Creación de Wireframes	Balsamiq
Navegadores testeo	Chrome
	Firefox

Tabla 1: Software utilizado en el desarrollo del proyecto

Web app	
Control de versiones	GitHub
Respaldo de documentación	Google Drive

Tabla 2: Web app utilizadas en el desarrollo del proyecto

Hardware	
Ordenadores	iMac 21,5' - End 2013
	Macbook Pro - Mid 2010
Pantalla auxiliar	LG Ultrawide

Tabla 3: Hardware utilizado para el desarrollo del proyecto

8. Planificación

Cada día está estipulado una jornada de 4 horas.

Nombre	Duración	Inicio	Final	Horas
Inicio del proyecto	15 días	20/02/2019	06/03/2019	58 h
Búsqueda y organización de información	7 días			28 h
Introducción, contexto y justificación	3 días			12 h
Enfoque y método seguido	1 día			4 h
Creación de índice	1/2 día			2 h
Planificación del trabajo	3 días			12 h
Diseño de la aplicación	20 días	07/03/2019	26/03/2019	80 h
Árbol de navegación	1 día			4 h
Diseño de Wireframes	3 días			12 h
Diseño de Prototipos	6 días			24 h
Modelo de Base de datos	2 días			8 h
Redacción y corrección de memoria	8 días			32 h
Desarrollo de la aplicación	47 días	27/03/2019	12/05/2019	188 h
Configuración del espacio de trabajo	1 día			4 h
Programación de la aplicación	34 días			136 h
Maquetación y estilos de la aplicación	6 días			24 h
Revisión errores y testeo de la aplicación	2 días			8 h
Creación video funcionamiento aplicación	1 días			4h
Documentar desarrollo aplicación	3 días			12 h
Fase final del proyecto	29 días	13/05/2019	10/06/2019	116 h
Programación de aplicación y corrección errores	15 días			60 h
Documentación y revisión del proyecto	10 días			40 h
Elaboración presentación final	4 días			16 h
			Total	442 h

Tabla 4: Planificación del trabajo

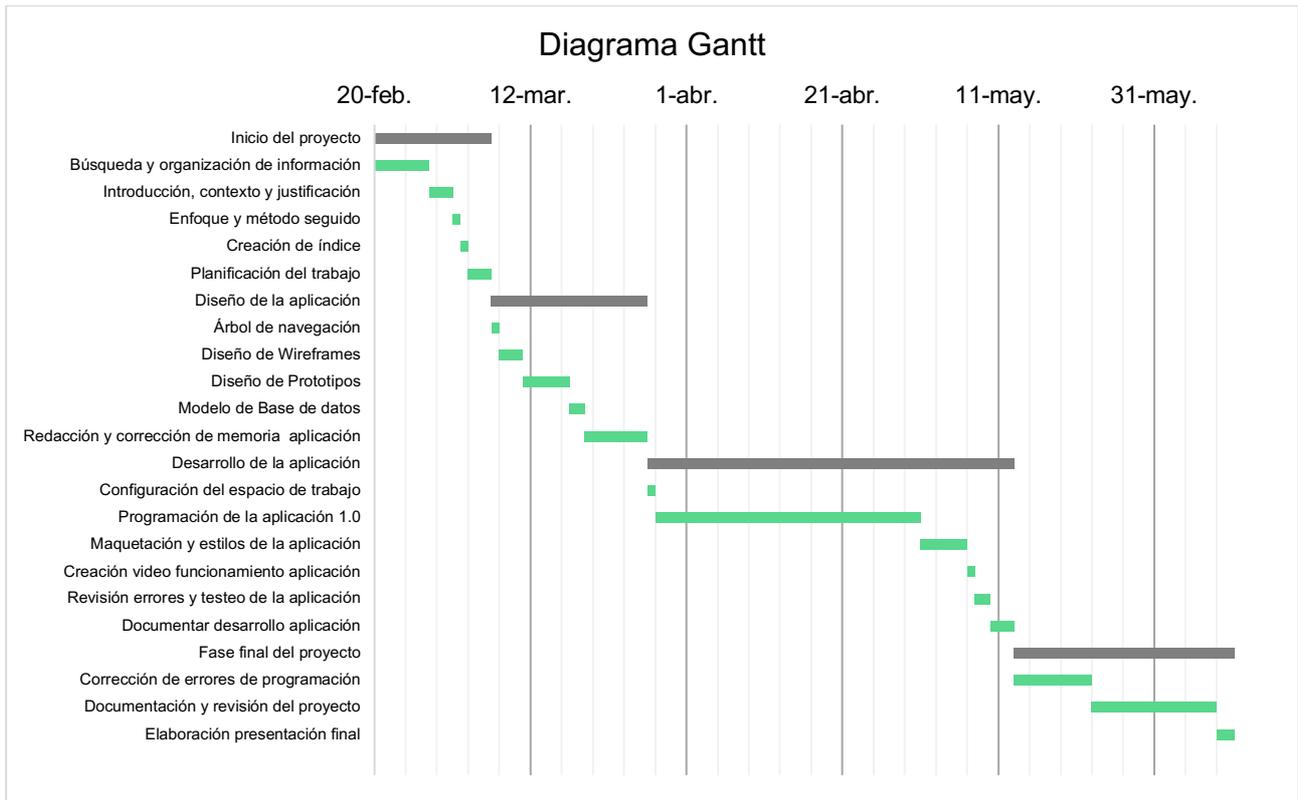


Figura 2: Diagrama de Gant

9. Prototipos

Previo al desarrollo de la aplicación, se han creado los prototipos tanto wireframes (Lo-Fi), como mockups (Hi-Fi). Con estas versiones de la aplicación se intenta aproximar al diseño final sin tener que modificar en la parte de desarrollo, ahorrando tiempo en el desarrollo del proyecto.

Ya que la aplicación consta de bastantes pantallas, se han querido plasmar algunos modelos como resumen, como por ejemplo el inicio de la aplicación, dashboard, añadir paciente (formulario), y vista de paciente.

9.1 Lo-Fi

VERSIÓN ESCRITORIO

Página de inicio y registro del usuario en la aplicación

Página de inicio Escritorio y Tablet landscape (1024px)



Figura 3: Wireframe de la página de inicio en versión escritorio

Página de acceso de usuario - Escritorio y tablet landscape (1024px)



Figura 4: Wireframe acceso a la aplicación

Página de registro de usuario - Escritorio y tablet landscape (1024px)

A Web Page

Logo

Sobre Nosotros Ayuda Contacto

Acceder Primeros Pasos

REGISTRO DE USUARIO

Nombre*

Apellidos*

Email*

Contraseña*

Marque este casilla si su centro no existe

He leído y acepto la Política de privacidad

[he olvidado mi contraseña](#)

[aviso legal](#) [política de privacidad](#) [condiciones de uso](#) [cookies](#) [mapa web](#)

Logo

Figura 5: Wireframe de registro de usuario a la aplicación.

Aplicación web para la gestión de consultas médicas. Alejandro Carmona Martos

Flujo de acceso y registro de usuario - Escritorio y tablet landscape (1024px)

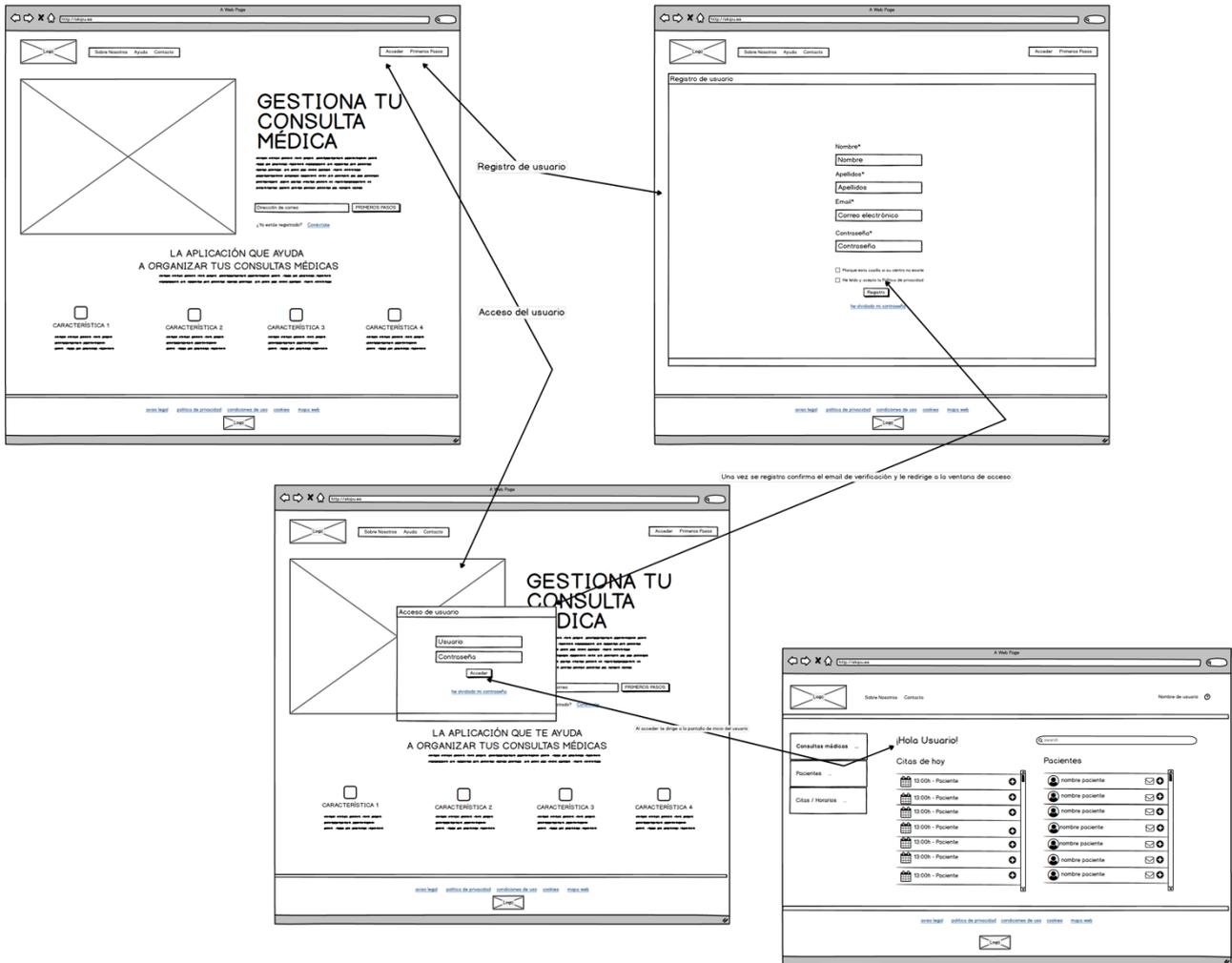


Figura 6: Wireframe del flujo de registro de usuario versión escritorio.

Página de inicio Escritorio y Tablet landscape (1024px)

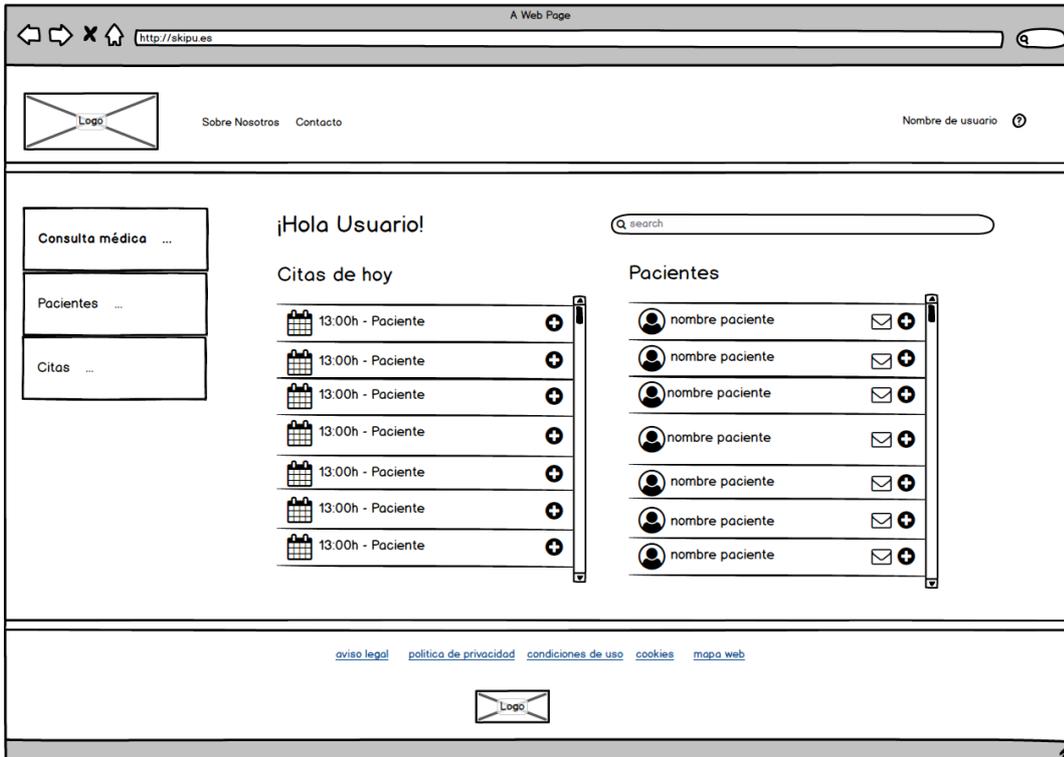


Figura 7: Wireframe de la página de dashboard versión escritorio

VERSIÓN TABLET PORTRAIT

Página de inicio y registro del usuario en la aplicación

Página de inicio tablet portrait (768px)

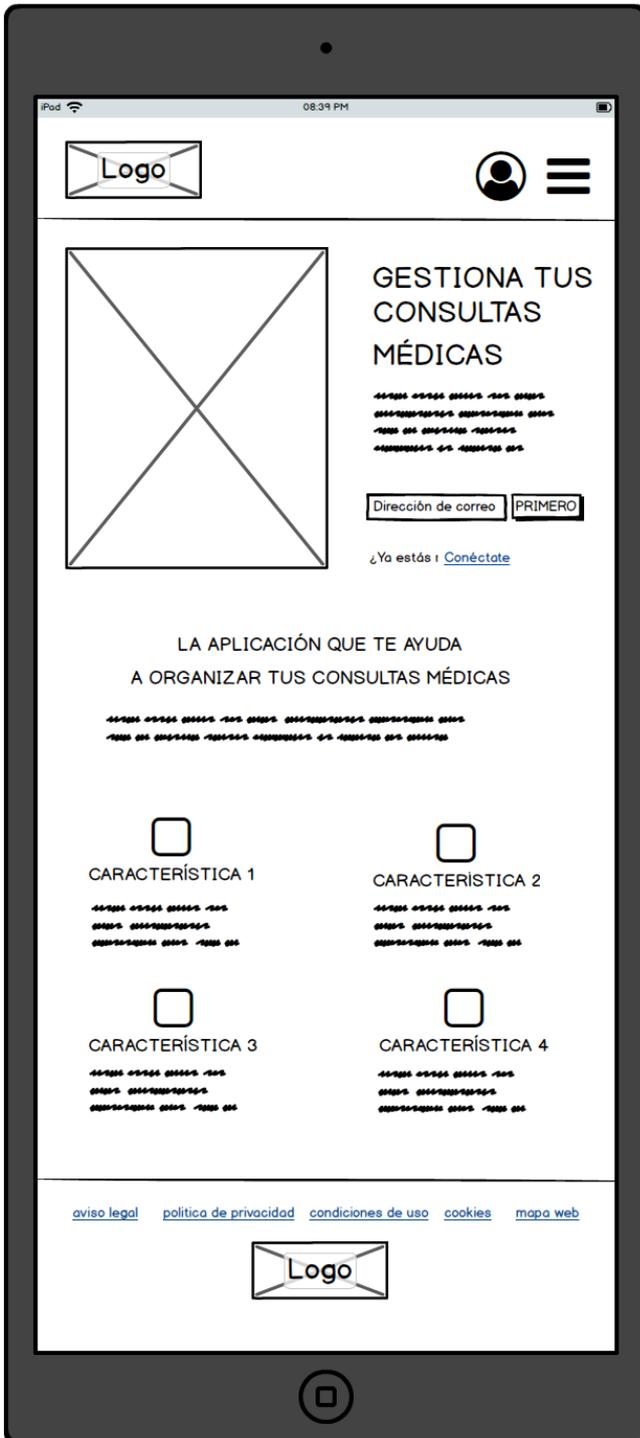


Figura 8: Wireframe de la página de inicio versión tablet

Flujo de acceso y registro de usuario - Tablet Portrait (768 px)

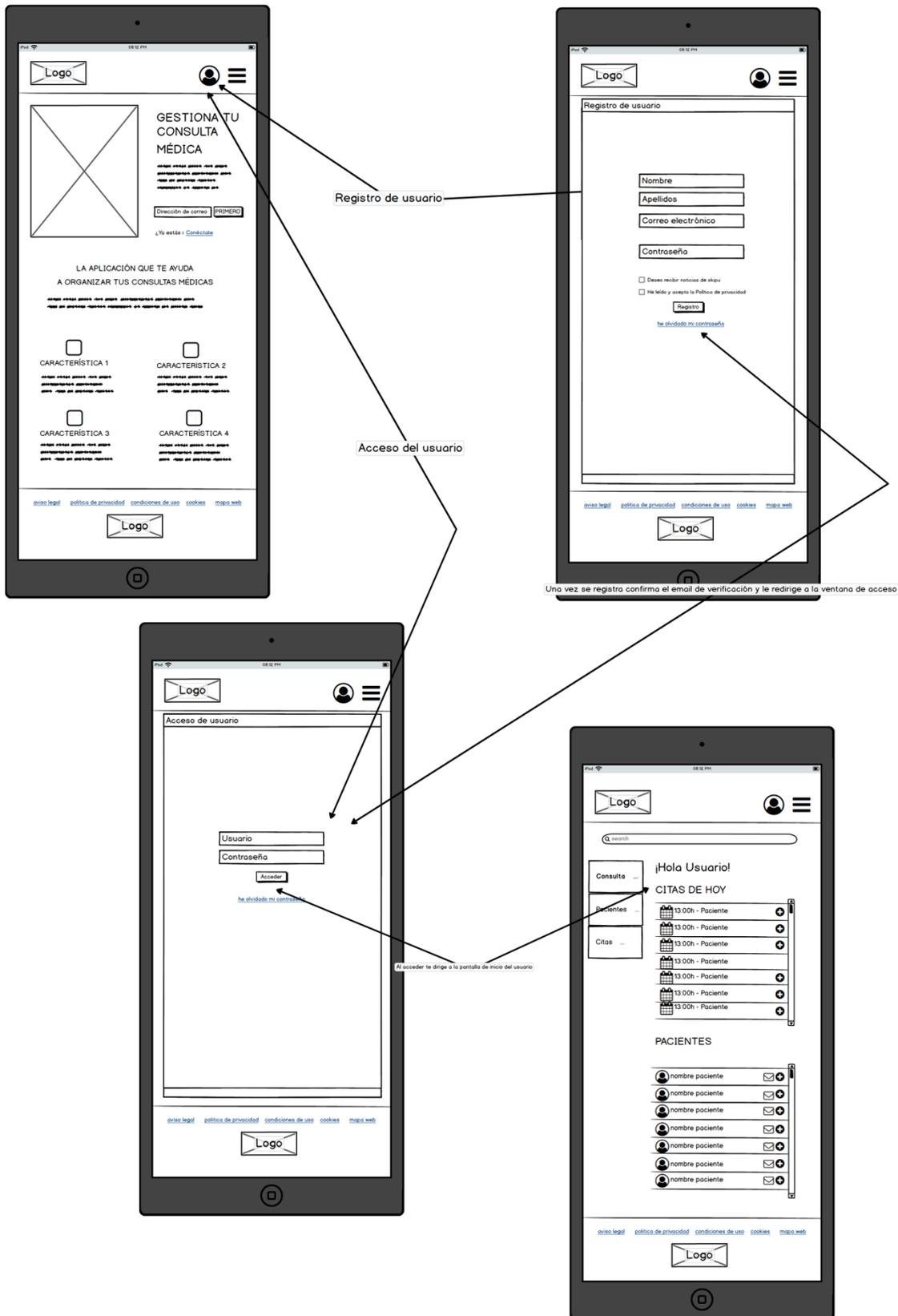


Figura 9: Wireframe del flujo de registro de usuario versión tablet

Página de inicio con registro tablet portrait (768px)

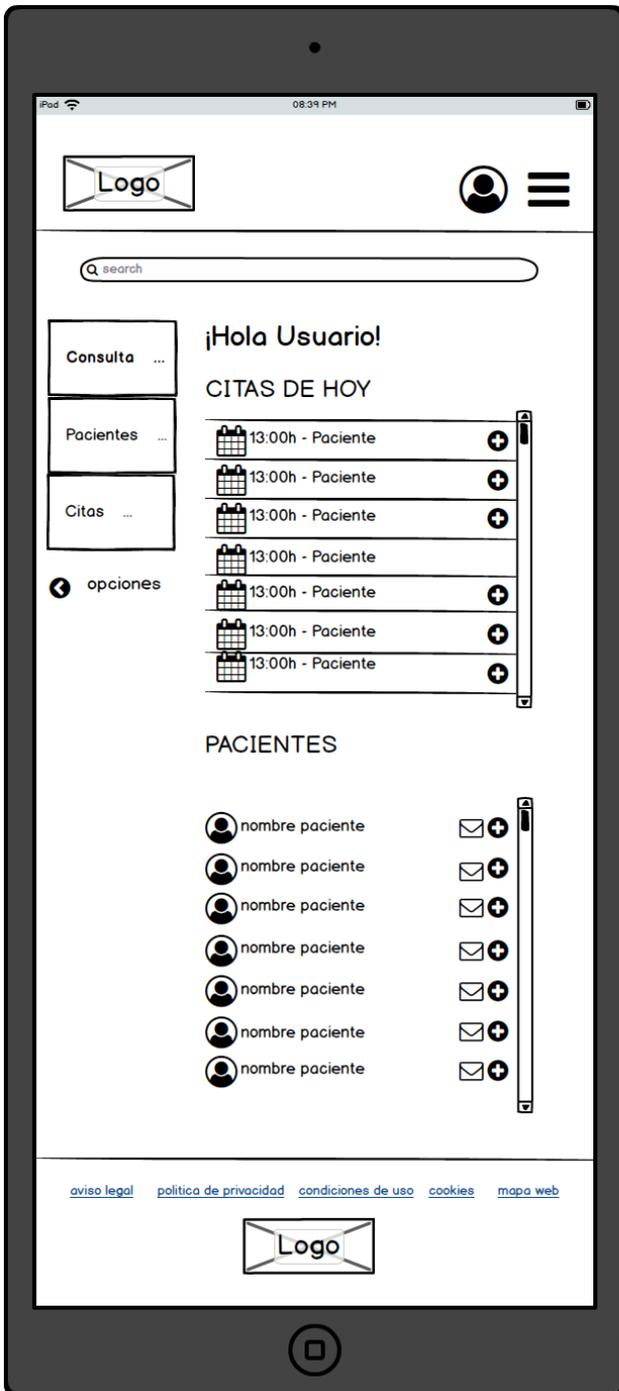


Figura 10: Wireframe de la página de dashboard versión tablet

VERSIÓN MÓVIL

Página de inicio y registro del usuario en la aplicación

Página de inicio con registro · móvil (468px)

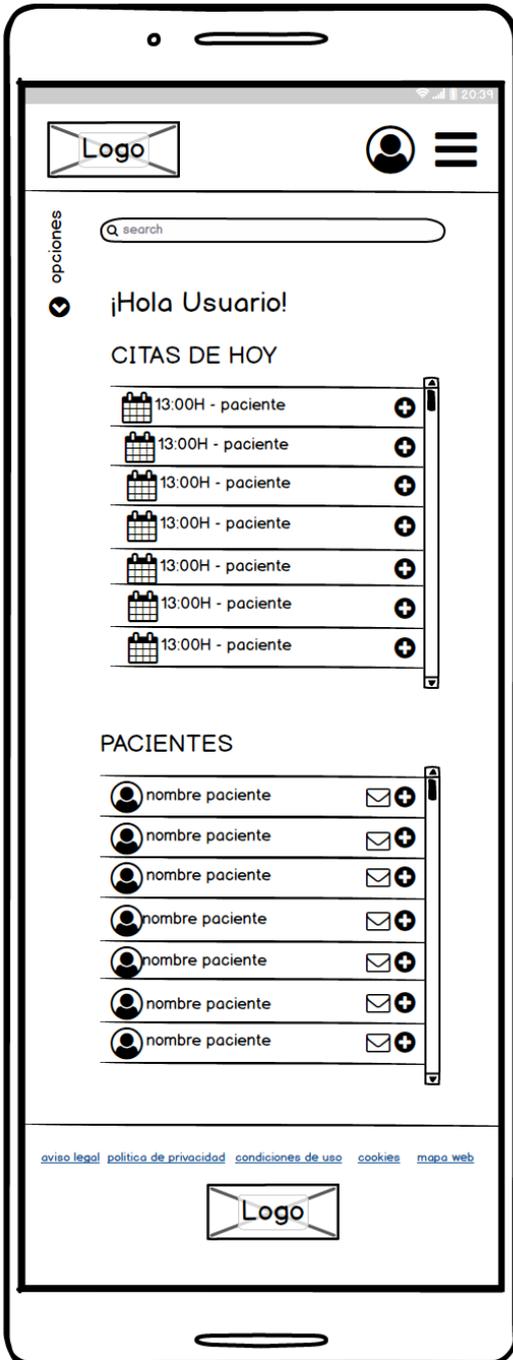


Figura 11: Wireframe de la página de inicio versión móvil

Flujo de acceso y registro de usuario - Móvil (480px)

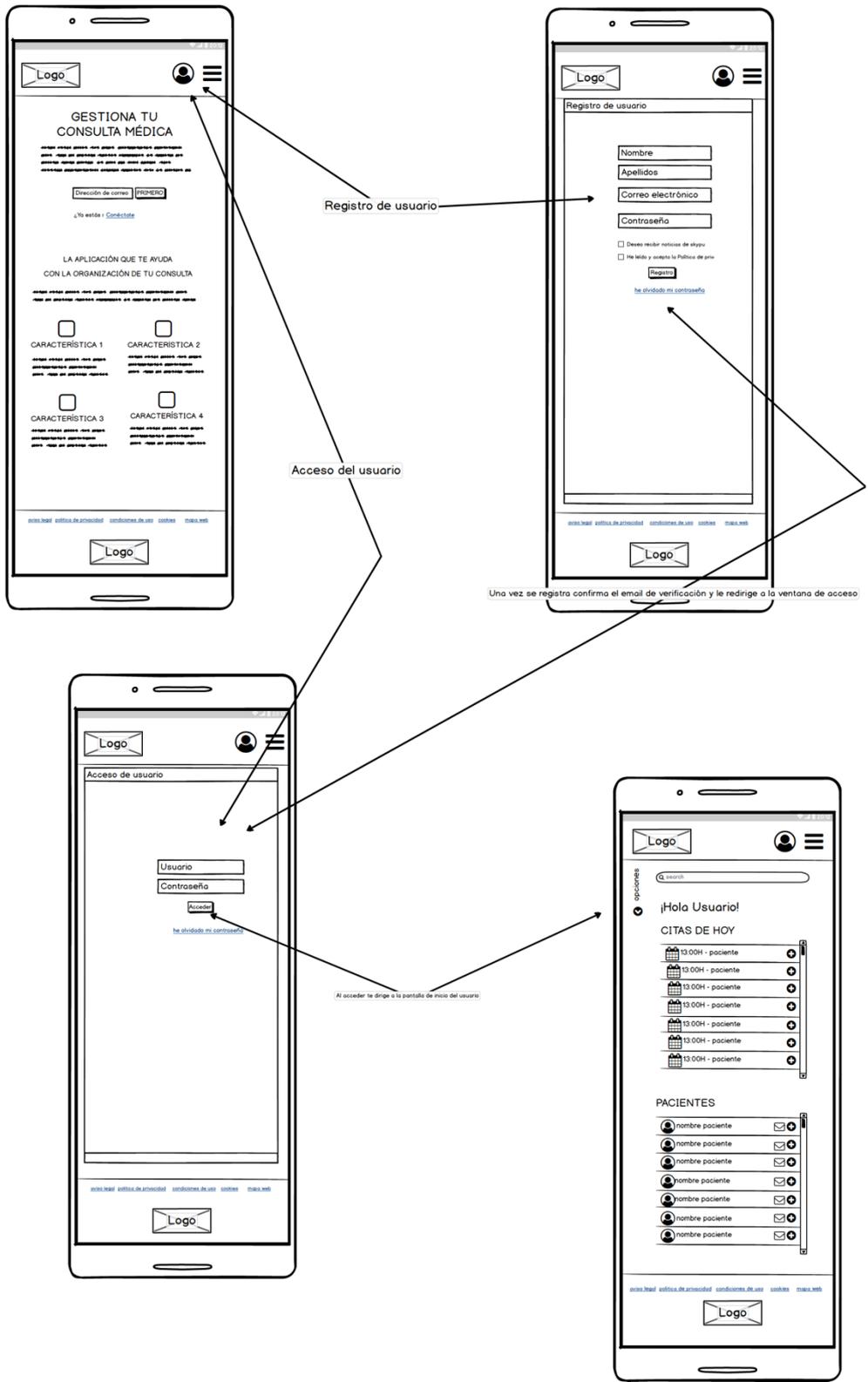


Figura 12: Wireframe del flujo de registro de usuario versión móvil

Página de inicio sin registro · móvil (480px)

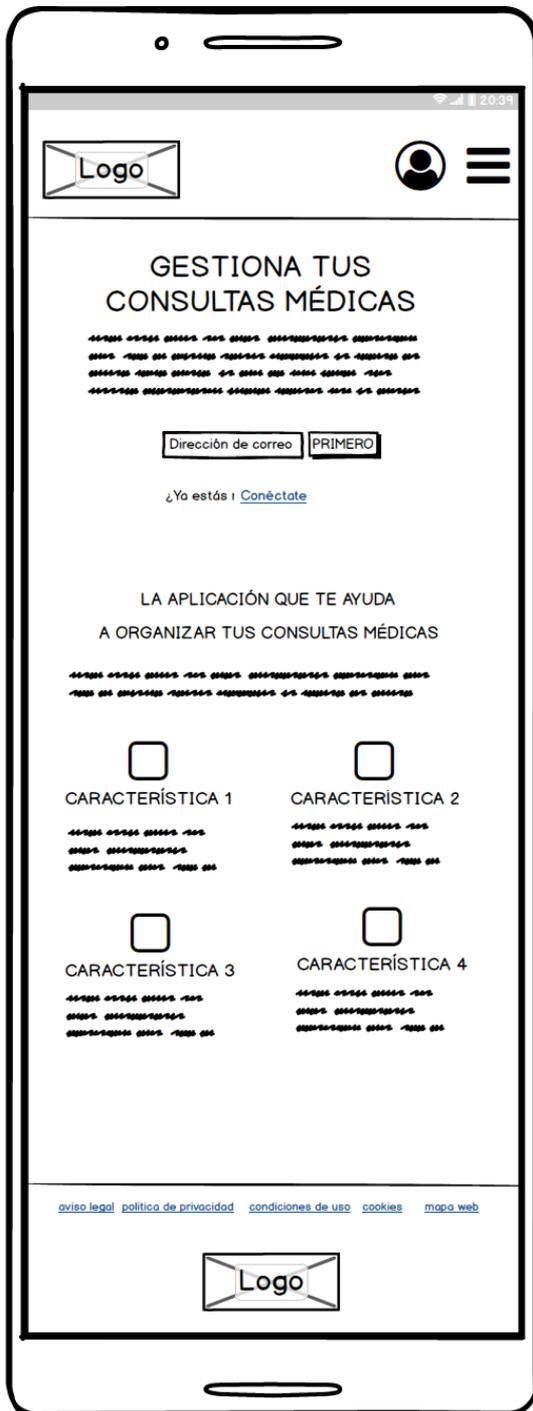


Figura 13: Wireframe de la página de dashboard versión móvil

FLUJO DE AÑADIR PACIENTE - VERSIÓN ESCRITORIO

Añadir un nuevo paciente a la consulta médica.

Flujo de añadir paciente - Escritorio y tablet landscape (1024px)

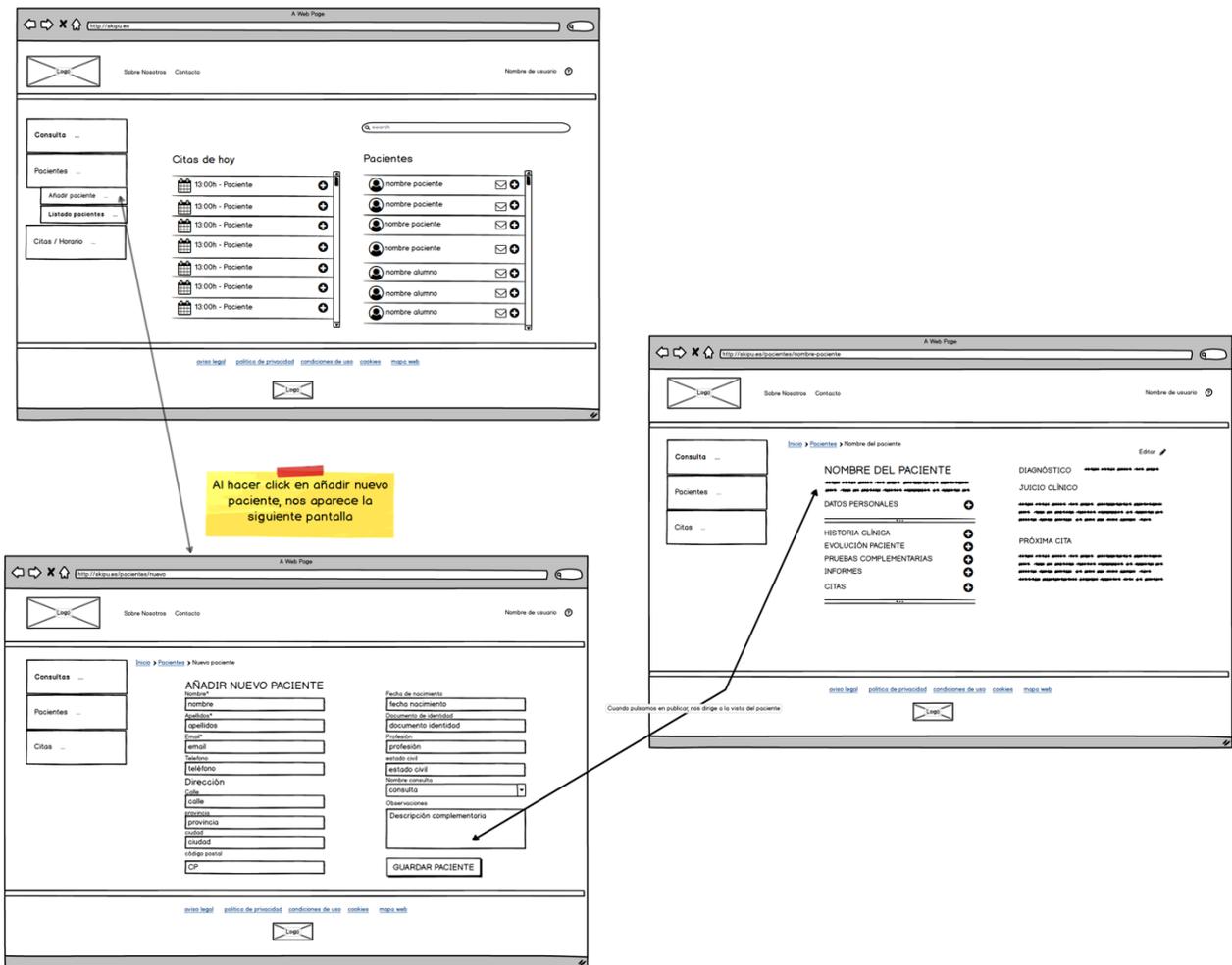


Figura 14: Wireframe del flujo de añadir nuevo paciente versión escritorio

FLUJO DE AÑADIR PACIENTE - VERSIÓN TABLET PROTRAIT

Añadir un nuevo paciente a la consulta médica.

Flujo de añadir paciente - Tablet Portrait (768 px)

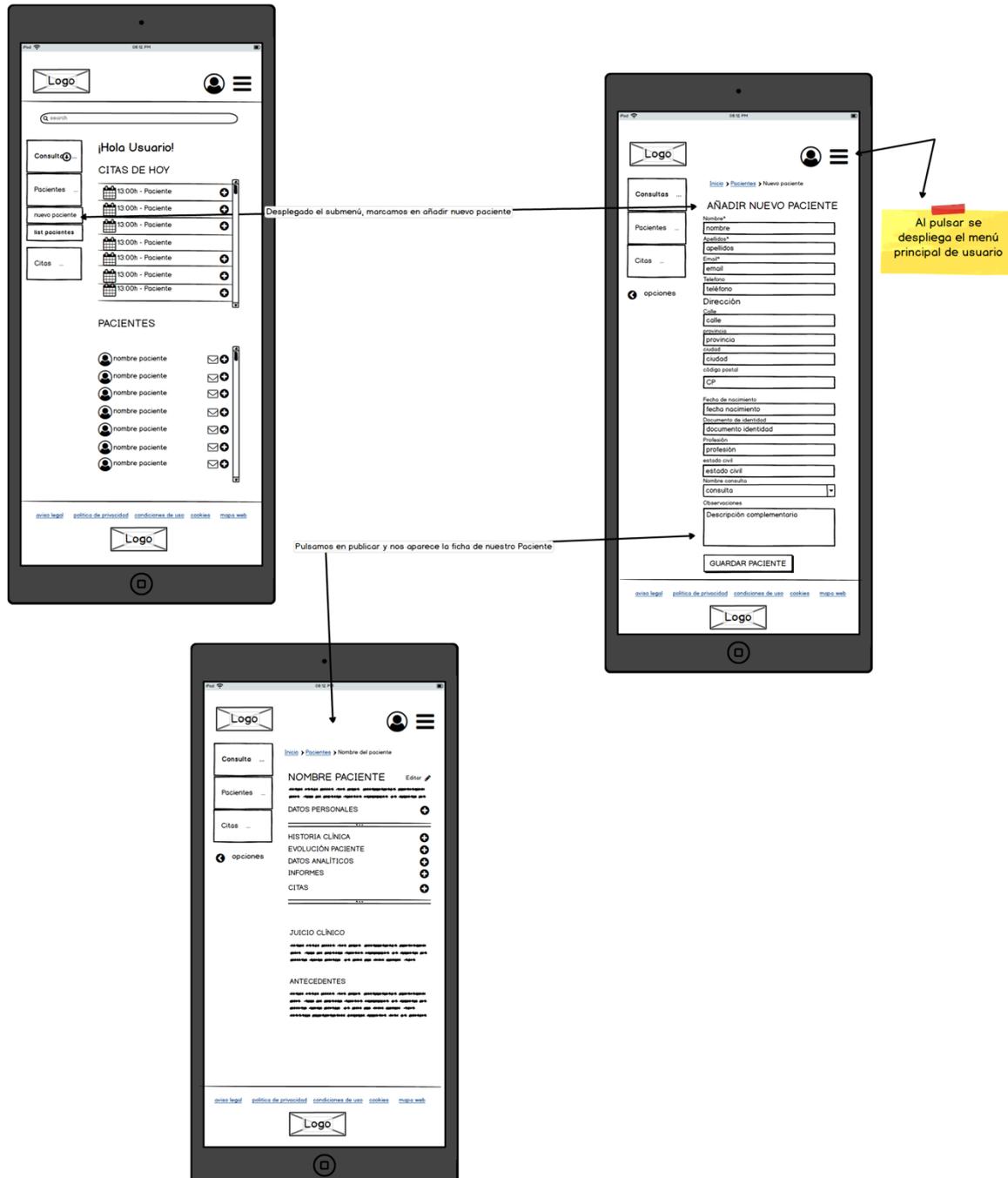


Figura 15: Wireframe del flujo de añadir nuevo paciente versión tablet

FLUJO DE AÑADIR PACIENTE - VERSIÓN MÓVIL

Añadir un nuevo paciente a la consulta médica.

Flujo de añadir paciente - Móvil (480 px)

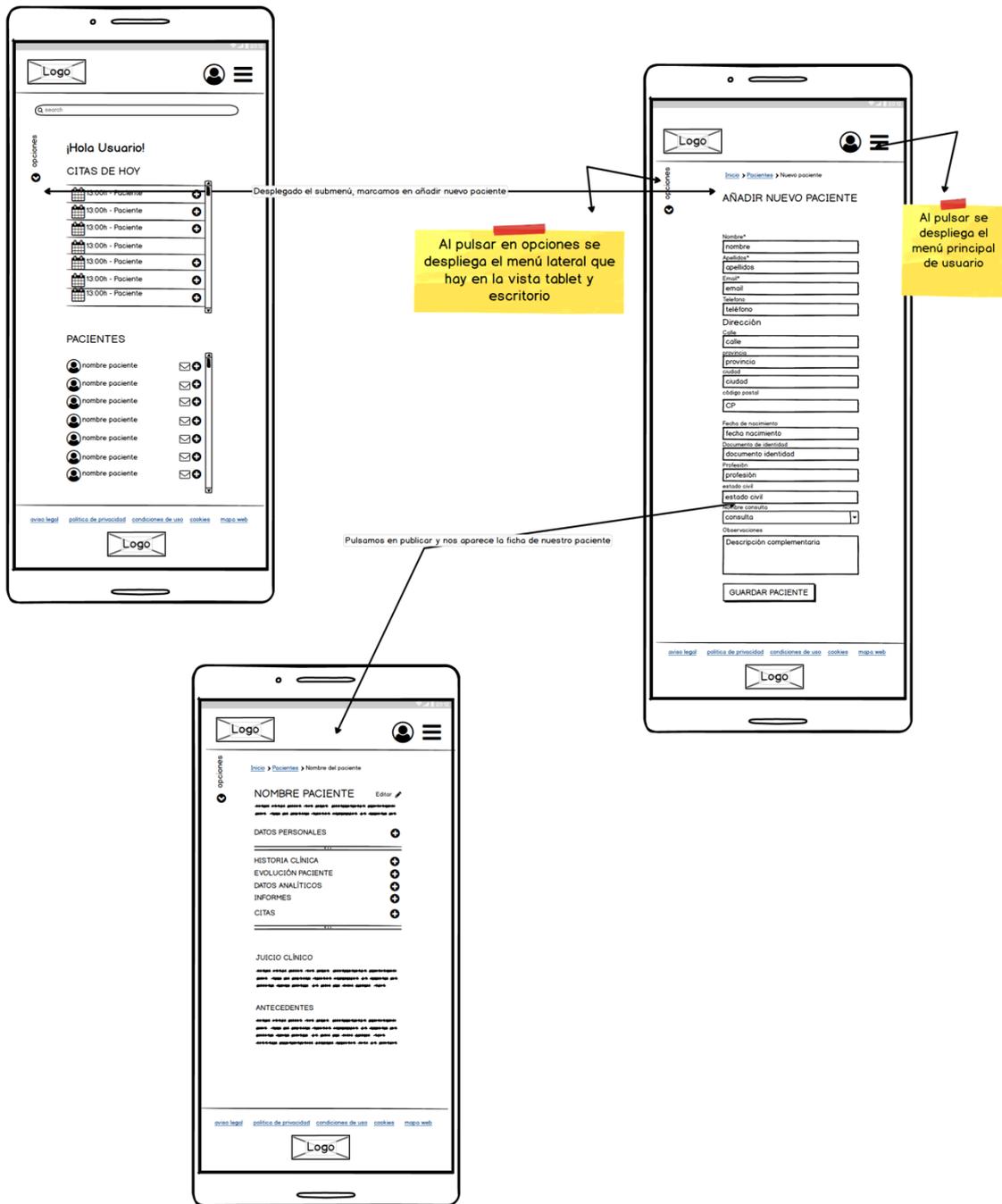


Figura 16: Wireframe del flujo de añadir nuevo paciente versión móvil

9.2 Hi-Fi

MOCKUP VERSIÓN ESCRITORIO

Página de inicio de la aplicación



Figura 17: Mockup página de inicio versión escritorio

MOCKUP VERSIÓN TABLET PORTRAIT

Página de inicio de la aplicación



Figura 18: Mockup página de inicio versión tablet

MOCKUP VERSIÓN MÓVIL

Página de inicio de la aplicación



Figura 19: Mockup página de inicio versión móvil

MOCKUP VERSIÓN ESCRITORIO

Página añadir paciente de la aplicación

skipu Sobre Nosotros ▼ Alejandro Carmona

inicio ► pacientes ► nuevo paciente

Añadir nuevo paciente

<input type="text" value="nombre"/>	<input type="text" value="fecha de nacimiento"/>
<input type="text" value="apellidos"/>	<input type="text" value="estado civil"/>
<input type="text" value="email"/>	<input type="text" value="profesión"/>
<input type="text" value="teléfono"/>	<input style="border-bottom: 1px solid #ccc;" type="text" value="género"/>
Dirección	<input type="text" value="documento identidad"/>
<input type="text" value="calle"/>	<input style="border-bottom: 1px solid #ccc;" type="text" value="consulta"/>
<input type="text" value="provincia"/>	<input type="text" value="Otros datos de interés"/>
<input type="text" value="ciudad"/>	
<input type="text" value="CP"/>	<input type="button" value="GUARDAR PACIENTE"/>

[Aviso legal](#) · [Política de privacidad](#) · [Condiciones de uso](#) · [Cookies](#) · [Mapa web](#)

skipu
Copyright 2019

Figura 20: Mockup página de añadir paciente versión escritorio

MOCKUP VERSIÓN TABLET

Página añadir paciente de la aplicación

skipu Sobre Nosotros ▼ Alejandro Carmona 

inicio ► pacientes ► **nuevo paciente**

Consultas médicas

- editar
- añadir paciente
- listado pacientes

Pacientes

- nuevo paciente
- listado pacientes

Citas / horario

- nueva cita
- calendario citas

Añadir nuevo paciente

nombre

apellidos

email

teléfono

Dirección

calle

provincia

ciudad

CP

fecha de nacimiento

estado civil

profesión

género

documento identidad

consulta

Otros datos de interés

GUARDAR PACIENTE

[Aviso legal](#) · [Política de privacidad](#) · [Condiciones de uso](#) · [Cookies](#) · [Mapa web](#)

skipu
Copyright 2019

Figura 21: Mockup página de añadir paciente versión tablet

MOCKUP VERSIÓN MÓVIL

Página añadir paciente de la aplicación

skipu ▼ Alejandro Carmona

inicio ► pacientes ► **nuevo paciente**

Añadir nuevo paciente

nombre

apellidos

email

teléfono

Dirección

calle

provincia

ciudad

CP

fecha de nacimiento

estado civil

profesión

género ▼

documento identidad

consulta ▼

Otros datos de interés

GUARDAR PACIENTE

[Aviso legal](#) · [Política de privacidad](#) · [Condiciones de uso](#) · [Cookies](#) · [Mapa web](#)

skipu
Copyright 2019

Figura 22: Mockup página de añadir paciente versión móvil

MOCKUP VERSIÓN ESCRITORIO

Página paciente de la aplicación

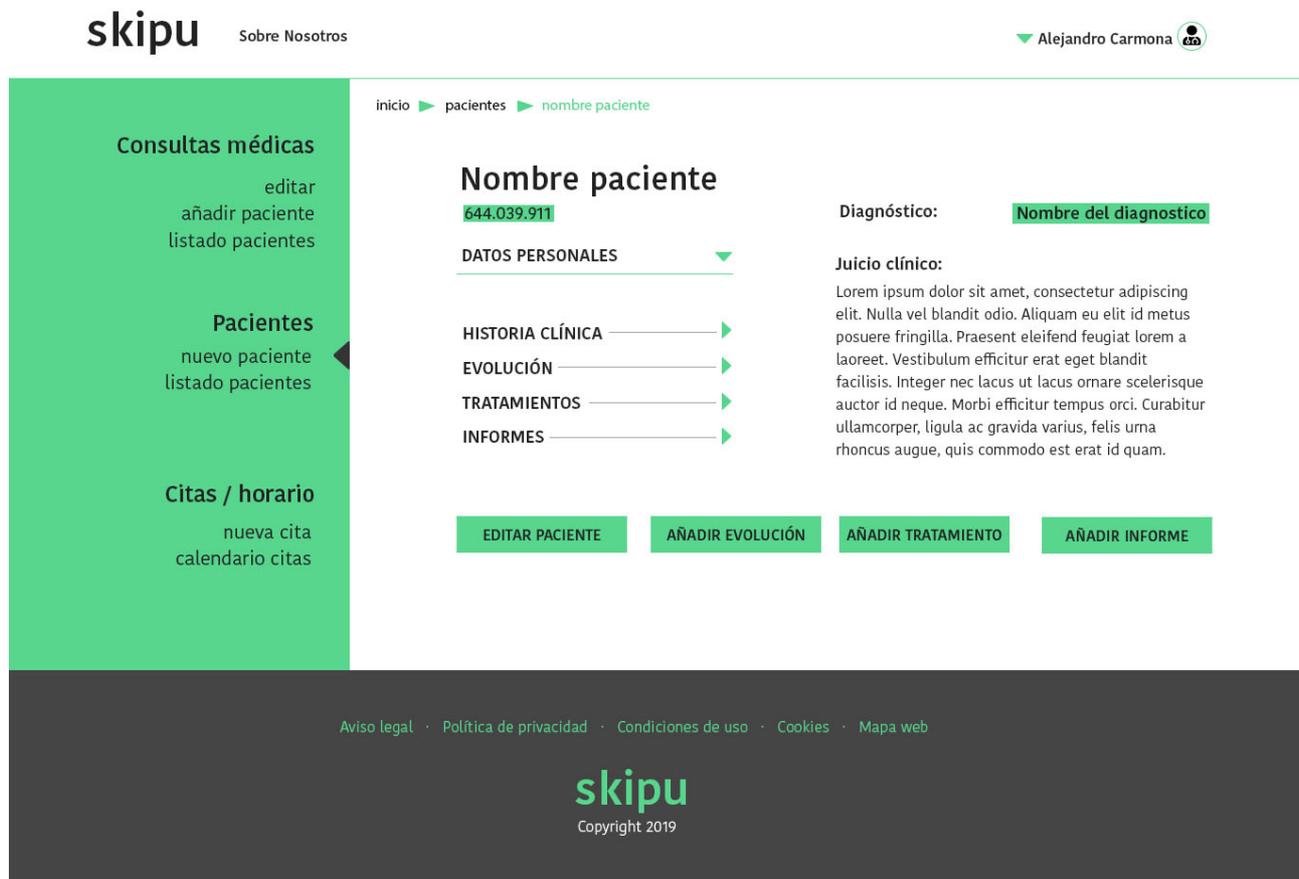


Figura 23: Mockup página de paciente versión escritorio

MOCKUP VERSIÓN TABLET

Página paciente de la aplicación

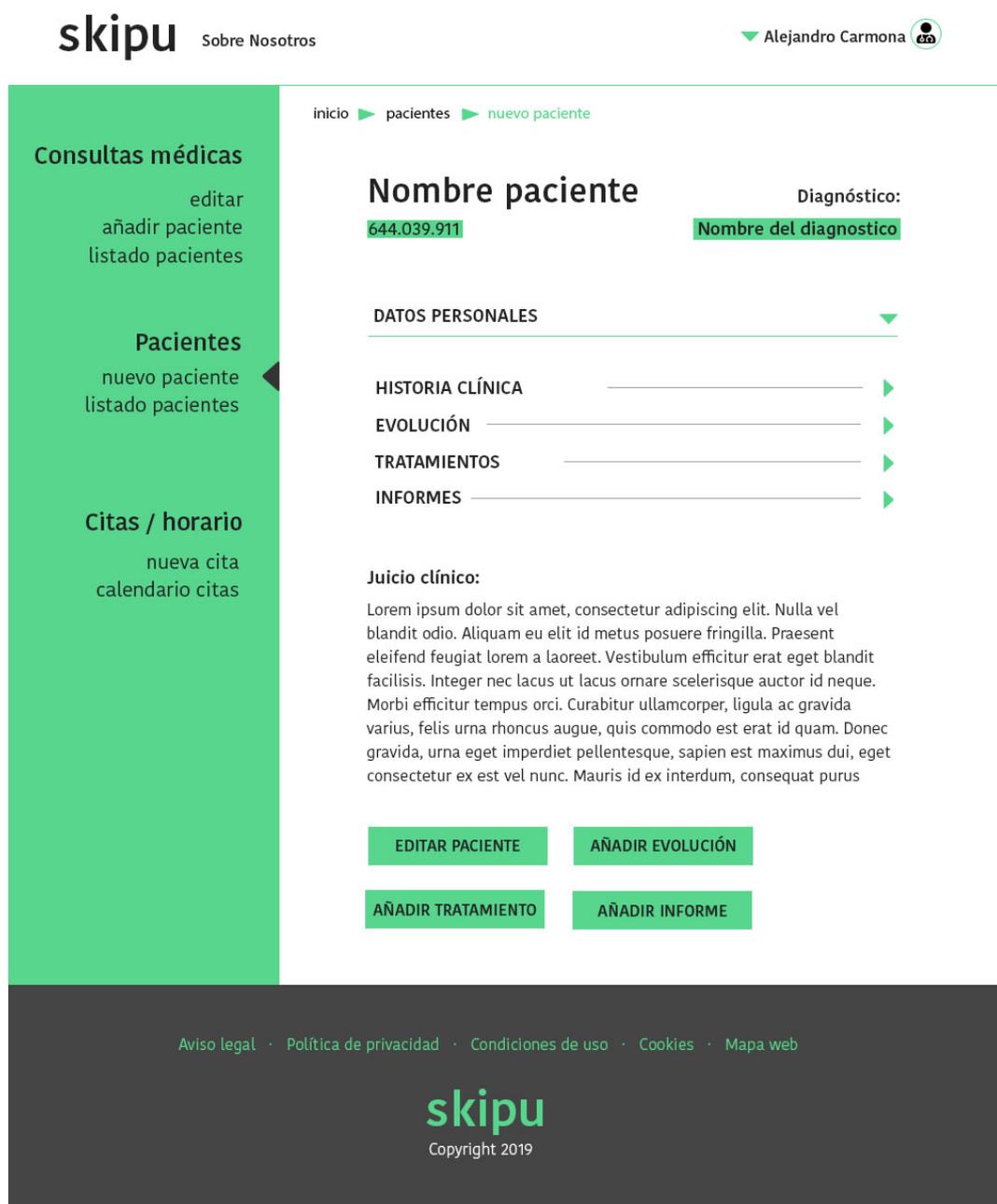


Figura 24: Mockup página de paciente versión tablet

MOCKUP VERSIÓN MÓVIL

Página añadir paciente de la aplicación

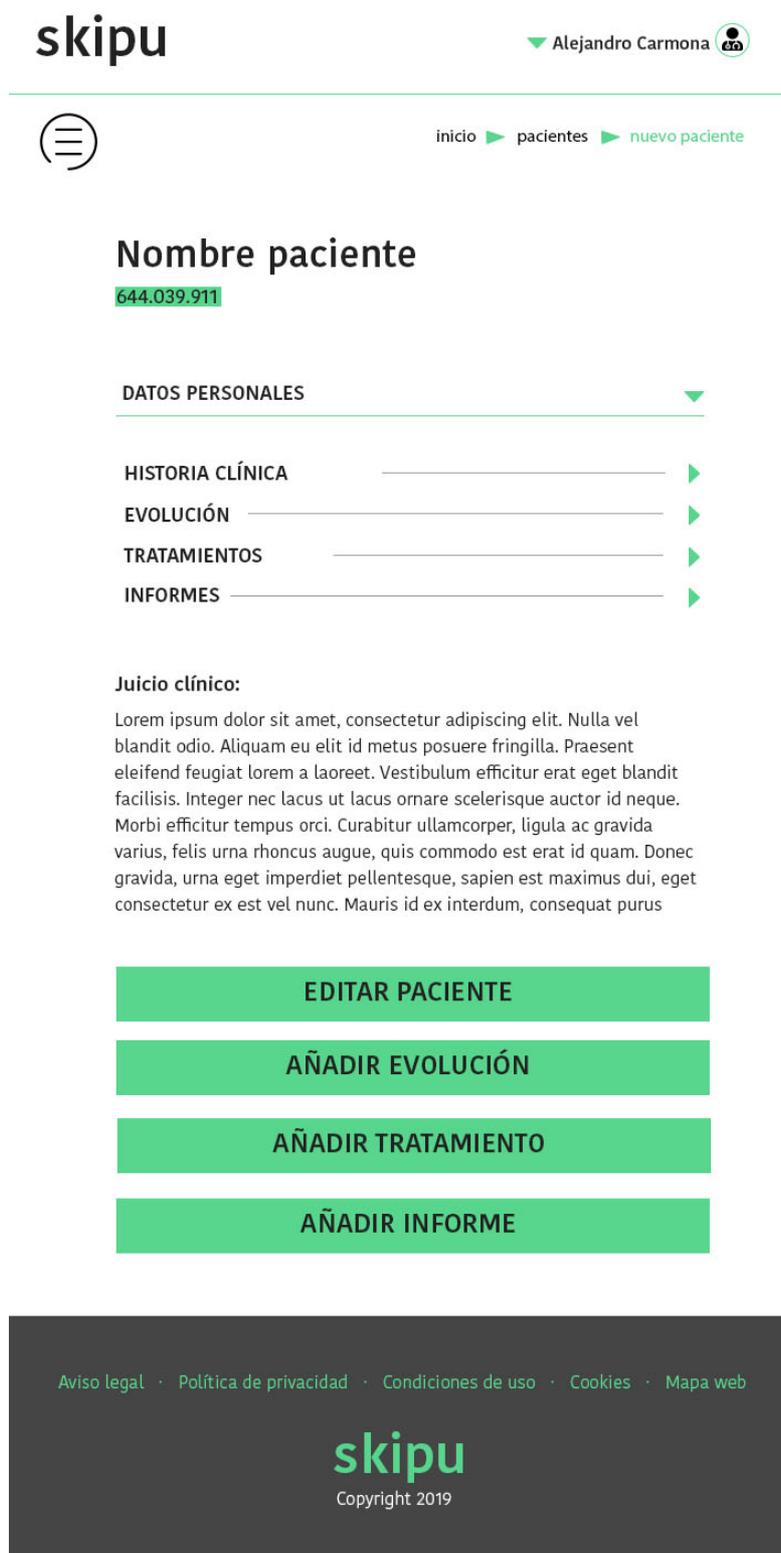


Figura 25: Mockup página de paciente versión móvil

10. Usabilidad/UX

Para comenzar se ha de realizar la justificación de los patrones de diseño comunes comenzando con los niveles de escala, límites y repetición. Después, se analizará un muestreo dos de los diseños concretos que se usarán durante toda la aplicación, de una forma simplificada a modo de ejemplo.

10.1 Justificación patrones de diseño comunes en la aplicación.

Niveles de escala:

El tamaño de los elementos en el diseño hace que se perciba la importancia de ellos mismos, y con respecto al conjunto. Por ello, los puntos más importantes en la aplicación están enfocados a la pregunta, ¿de qué va la aplicación web? además se tiene que crear la necesidad de registrarse para su uso. Por ello, se ha destacado tanto el mensaje, como el formulario inicial de registro en la página de inicio.

Límites:

Para crear una organización y orden en el diseño se apoyará en los límites, donde se diferencia cabecera, cuerpo y pie de página. Dentro del cuerpo, se irán mostrando en subsecciones los elementos de cada pantalla, como, por ejemplo menú de usuario y contenido a mostrar.

Repetición:

Los elementos que se repetirán para conseguir la uniformidad en el diseño de la aplicación van a ser: la posición de la cabecera con logo y menú de usuario, el área central con menú del panel de administración y contenido, y el pie de página con enlaces legales, mapa web y logo.

Menú del panel de administración:

En la aplicación hay un menú interior que aparecerá una vez el usuario esté registrado y acceda a través de la zona de acceso con usuario y contraseña, para ello se ha elegido un menú vertical con dos referencias:

Para escritorio se basa en un menú vertical que al pasar por encima el ratón aparece el submenú.

Se utiliza cuando hay entre 2 y 9 secciones de contenido que necesiten una estructura de navegación jerárquica.

Su funcionalidad se asemeja a una de una aplicación de escritorio. Imita la metáfora.

Referencia.: <http://ui-patterns.com/patterns/HorizontalDropdownMenu>

Para el menú tablet y móvil se utilizará un menú de acordeón, donde se desplegarán los submenús. Esta solución se ha tomado porque, el efecto de pasar el ratón por encima en estos dispositivos no está presente, además, la necesidad de optimizar el espacio en la pantalla ya que se reduce considerablemente.

Este patrón lo usamos cuando:

- Queremos los beneficios de un menú de barra lateral normal, pero que puede no tener el espacio para enumerar todas las opciones.
- Como hay más de 2 secciones principales en un sitio web, cada una con 2 o más subsecciones.
- Tenemos menos de 10 secciones principales
- Sólo tenemos dos niveles para mostrar en la navegación principal.

Referencia: <http://ui-patterns.com/patterns/AccordionMenu>

10.2 Justificación patrones de diseño de la estructura.

Debido a la necesidad de la vista flexible en multidispositivos, se plantea esta aplicación web con una perspectiva de Responsive Web Design, en adelante, RWD, por lo que tenemos que establecer los puntos de ruptura del diseño. Los puntos de ruptura para RWD en nuestro caso son:

1. Escritorio: Cualquier resolución mayor de 1024px y tablet landscape: 1024px
2. Tablet portrait: 768px
3. Teléfono pantalla grande: 480px

Se cogerá la página de inicio como ejemplo para la justificación del diseño de estructura.

Layout:

Mostly Fluid para el contenido de la aplicación de escritorio a móvil.

La versión de escritorio del sitio cuenta con un diseño de varias columnas en una cuadrícula de ancho fijo de 1024px - con grandes márgenes en ambos lados en pantallas más grandes como por ejemplo en 1280px. El contenido, mediante una cuadrícula fluida, se va redistribuyendo y las columnas se apilan verticalmente. Conforme vamos reduciendo el tamaño se apilan una encima de otra siendo de izquierda a derecha (izquierda la que queda más arriba).

Off Canvas para el menú en versión tablet portrait y móvil.

En el tamaño de pantalla menor de 768px, el diseño cambia a una sola columna con contenido, para evitar que el menú de usuario estorbe a la vista hemos optado un off canvas y que desaparezca este menú y solamente aparezca cuando pulsemos en el botón adecuado de opciones. Ventajas de estos patrones:

- Generalmente no se suelen utilizar más un/dos puntos de ruptura entre las pantallas grandes y las pequeñas.
- El layout fluye naturalmente con las dimensiones del contenedor padre y se necesitan hacer pocos ajustes para varias resoluciones de pantalla y dispositivos.

- Con off canvas, aumentamos el espacio al usuario en la versiones más reducidas.

Referencias: <https://carlosazaustre.es/blog/los-5-patrones-del-responsive-design/>

10.3 Justificación patrones de diseño del flujo de acceso / registro de usuario.

Patrones de diseño de interfaces:

Formulario de acceso: Este patrón proporciona el acceso a la aplicación una vez estamos registrados.

Su funcionamiento será el siguiente, al pulsar en la pestaña de acceso aparecerá la opción de cumplimentar un formulario con usuario y contraseña, además de añadir un enlace que nos llevaría al flujo de recordar contraseña al usuario. Se contemplará la opción de almacenar la contraseña para que el usuario pueda acceder la siguiente vez de forma más rápida.

Automáticamente al acceder correctamente aparecerá el dashboard inicial de la aplicación, así el usuario sabrá que ha accedido correctamente. Saldrá un mensaje de error si no ha realizado correctamente.

Referencias: <http://www.welie.com/patterns/showPattern.php?patternID=login>

Formulario de registro: Este patrón proporciona el registro de un nuevo usuario a la aplicación.

Cómo la aplicación se basa en el registro de usuarios, se ha creado un formulario para que se puedan registrar. Si el usuario no está registrado deberá cumplimentar este formulario. Son datos básicos apoyados de ayuda para la introducción de los mismos. Uno de los campos que tienen que rellenar es el de la consulta médica a la que pertenecen. Posteriormente deberá loguearse con esos datos como medida doble verificación. Las formas de uso de este patrón de diseño son:

- Cuando se quiere restringir el acceso a cierto contenido.
- Cuando se desea controlar qué usuarios tienen acceso al contenido.

Referencias: <http://ui-patterns.com/patterns/AccountRegistration>

<http://www.welie.com/patterns/showPattern.php?patternID=registration>

Dashboard: Este patrón se utilizará para mostrar datos de múltiples fuentes de la aplicación. Aparecerá una vez el usuario accede correctamente a la aplicación web con su usuario.

<http://ui-patterns.com/patterns/dashboard>

10.4 Justificación patrones de diseño de flujo para añadir nuevo paciente.

Formulario: Con este patrón de diseño agrupamos los datos del elemento paciente teniendo en cuenta los siguientes puntos:

- Campos obligatorios y opcionales.
- Usando el elemento de entrada correcto
- Buena información para informar al usuario
- Prevención de errores de entrada
- Navegación por teclado

<http://www.welie.com/patterns/showPattern.php?patternID=forms>

Agregar personas con búsqueda de autocompletar: Este patrón de diseño nos ayudará:

- Con problemas de ambigüedad, cuando un elemento se puede introducir de varias maneras.
- Con el tipo de información ingresada, ya que se puede asociar fácilmente con una información específica en el sistema. Por ejemplo, un usuario puede no recordar cómo se escribe "Albayzín", pero puede ingresar las primeras 3 letras y encontrar la palabra en la lista de autocompletar.
- Cuando las sugerencias de autocompletado puedan obtenerse de un conjunto de datos que sea manejable en tamaño.
- Cuando la velocidad de entrada es un objetivo importante
- La precisión de entrada es nuestro objetivo importante
- Cuando el número de elementos, como es este caso, es demasiado grande o inconveniente para mostrar en un cuadro desplegable estándar.

Referencia: <http://ui-patterns.com/patterns/Autocomplete>

10.5 Árbol de navegación

El siguiente esquema muestra una representación resumida de las pantallas que existen en nuestra aplicación y los nexos entre ellas.

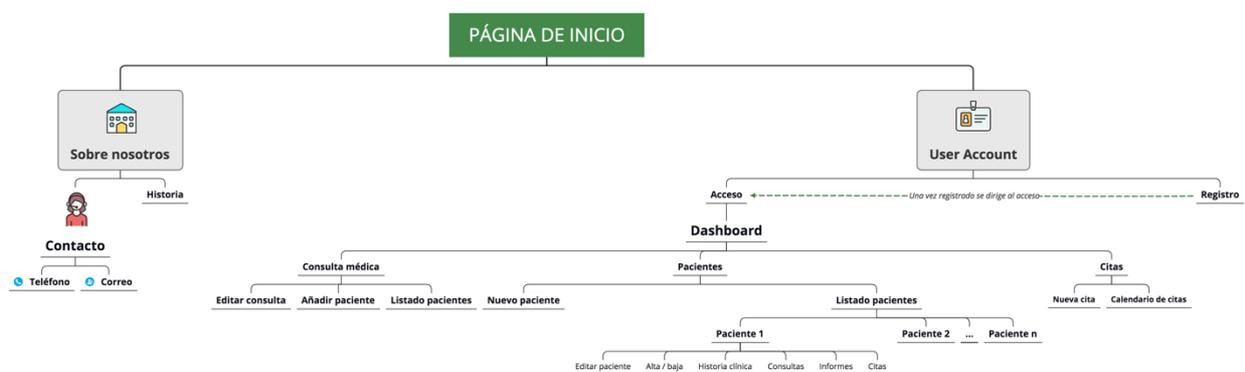


Figura 26: Árbol de navegación de la aplicación web.

11. Seguridad

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

Para proporcionar mayor seguridad a nuestra aplicación hemos dotado a nuestro backend de JWT. Es un estándar abierto de uso libre y para varios lenguajes de programación. Se utiliza para proteger la transferencia de información al consumir una API por medio de un token.

JWT (JSON Web Tokens)

La autenticación JWT es usada cuando se trabaja con una API y permite representar los datos del usuario de forma segura. Cuando el usuario inicia sesión en una aplicación web usando su email y contraseña, se devuelve automáticamente un token. Este token es usado en las posteriores llamadas a la API y es enviado en las cabeceras de las peticiones protegidas de la API.

12. Requisitos de implantación y uso

Información detallada acerca de los recursos necesarios para implantar y usar nuestra aplicación web en un servidor.

SERVIDOR

SOFTWARE - Recursos necesarios para la implantación de la aplicación en un servidor web	
Apache	Version mínima
PHP	Version 7.3.2
MySQL	5.0.12
Dominio	
SSL	
phpMyAdmin	4.8.5

HARDWARE - Recursos necesarios para la implantación de la aplicación en un servidor web	
Hosting compartido	Requisito mínimo.

FORMACIÓN – Requisitos técnicos para la implantación de la aplicación en un servidor web	
Desarrollador web con experiencia en implantación de proyectos en servidores web.	

CLIENTE

SOFTWARE - Para el uso de la aplicación	
Navegadores	Chrome
	Safari
	Edge
	Firefox

HARDWARE - Para el uso de la aplicación	
PC, MAC	Para uso ofimático
Móvil	Para uso ofimático
iPad o Tablet	Para uso ofimático

FORMACIÓN – Para el uso de la aplicación	
No necesaria experiencia previa en aplicaciones web. Usuario estándar.	

13. Instrucciones de implantación

Información con pasos detallados acerca de cómo se debe implantar la aplicación. Las presentes instrucciones también se acompañan a la aplicación en un archivo contenido en sus directorios.

Partimos de un punto en el que el administrador web dispone de un servidor web y dominio.

1. Subida de la carpeta del contenido de la API (skipu-backend) al servidor web por SFTP
2. Creación y subida de la BBDD
3. Configuración archivo .env con la conexión BBDD
 - a. Servidor
 - b. Puerto
 - c. Nombre BBDD
 - d. Usuario BBDD
 - e. Contraseña
4. Obtener ruta API mostrada en navegador
5. Configurar el archivo GLOBAL.ts que se encuentre en (skipu-frontend/app/src/shared/servicios)
6. Subida del proyecto en Angular compilado (carpeta Dist)
7. Probar que la aplicación funciona en el navegador.

14. Instrucciones de uso de la aplicación

Información con pasos detallados acerca de cómo se debe utilizar la aplicación. Las presentes instrucciones deben también acompañar a la aplicación en un archivo contenido en sus directorios.

La aplicación web está alojada en <http://skipu.es/>

1. Registro en la aplicación:
 - 1.1. Para registrarse en la aplicación se debe acceder al formulario de registro, se encuentra al pulsar en el botón de registro de la cabecera de la aplicación web.
 - 1.2. A continuación, solicitará **nombre, apellidos, email y contraseña**.
 - 1.3. Si se ha realizado con éxito, aparece un mensaje en color gris para acceder al formulario de acceso de la aplicación. En caso contrario mostrará un error de color rojo con la información.

2. Acceso al panel de administración:
 - 2.1. Para acceder a la aplicación se debe ir al formulario de acceso, que se encuentra al pulsar en el botón de acceso de la cabecera de la aplicación web. También se puede acceder justo después del registro, en el mensaje de éxito de registro de usuario.
 - 2.2. Se introduce **email y contraseña**.
 - 2.3. Si se ha realizado con éxito, se dirigirá automáticamente al Dashboard o Panel de administración. En caso contrario mostrará un error de color rojo con la información

3. Dashboard o Panel de administración
 - 3.1. Lo primero que se tiene que realizar es **crear una nueva consulta**, que es donde se incluirá a los pacientes.
 - 3.2. Una vez se haya creado la consulta, nos permitirá incluir a los pacientes.
 - 3.3. Habitualmente si tenemos ya consultas y pacientes creados nos aparecerá una lista donde podemos editar y borrar los elementos de las dos listas.

4. Creación, edición y borrado de una consulta
 - 4.1. **Crear una consulta:** Desde el Dashboard o Panel de administración se pulsa en nueva consulta, se cumplimenta el formulario que aparece y marca en guardar.
 - 4.2. **Editar una consulta:** Desde el Dashboard aparece una lista con las consultas que el usuario tiene. Junto a los nombres aparece el botón de editar. Si se pulsa, se dirige al formulario con los datos de la consulta (primera versión, solo nombre) y se puede modificar el nombre y a continuación guardar el cambio.
 - 4.3. **Borrar una consulta:** Desde el Dashboard aparece una lista con las consultas que el usuario tiene. Junto a los nombres aparece el botón de borrar. Al pulsar en el botón automáticamente se borra la consulta indicada.

5. Creación, edición y borrado de un paciente

5.1. **Crear una paciente:** Desde el Dashboard o Panel de administración o desde el detalle de la consulta, se pulsa en nuevo paciente, se cumplimenta el formulario que aparece y se finaliza la acción en guardar.

5.2. **Editar una consulta:** Desde el Dashboard o detalle de consulta aparece una lista con los pacientes que el usuario tiene. Junto a los nombres aparece el botón de editar. Si se pulsa, se dirige al formulario con los datos del paciente y se puede modificar todos los datos que se quiera. Para finalizar el cambio se pulsa en guardar.

5.3. **Borrar una consulta:** Desde el Dashboard o detalle de consulta aparece una lista con los pacientes que el usuario tiene. Junto a los nombres aparece el botón de borrar. Al pulsar en el botón automáticamente se borra el paciente indicado y toda su información.

6. Creación, edición de un historial

6.1. **Crear un historial:** Desde el detalle de paciente, si aún no dispone de historia, nos aparecerán varios puntos donde añadirla como el menú, al pulsar en historia clínica o justamente debajo de juicio clínico. Se pulsa en cualquiera de los mencionados anteriormente y se cumplimenta el formulario que aparece. Para finaliza la acción, marcamos en guardar.

6.2. **Editar una consulta:** Desde el detalle de paciente o detalle de historia aparece la posibilidad de editar historia clínica. Si se pulsa, se dirige al formulario con los datos de la historia y se puede modificar todos los datos que se quiera. Para finalizar el cambio se pulsa en guardar.

7. Creación, edición de una evolución

7.1. **Crear una evolución:** Desde el detalle de paciente, si aún no dispone de evoluciones, nos aparecerán varios puntos donde añadirla como el menú, al pulsar en evolución. Se pulsa en cualquiera de los mencionados anteriormente y se cumplimenta el formulario que aparece. Para finaliza la acción, marcamos en guardar.

7.2. **Editar una evolución:** Desde el listado de evoluciones aparece una lista con las evoluciones del paciente. Junto a los nombres aparece el botón de editar. Si se pulsa, se dirige al formulario con los datos de la evolución y se puede modificar todos los datos que se quiera. Para finalizar el cambio se pulsa en guardar.

7.3. **Borrar una evolución:** Desde el listado de evoluciones aparece una lista con las evoluciones del paciente. Junto a los nombres aparece el botón de borrar. Al pulsar en el botón automáticamente se borra la evolución indicada y toda su información.

8. Creación, edición de un tratamiento

8.1. **Crear un tratamiento:** Desde el detalle de evolución, aparecerá un apartado de tratamientos para añadirlo a dicha evolución. Se pulsa en añadir tratamiento y se cumplimenta el formulario que aparece. Para finaliza la acción, marcamos en guardar.

8.2. **Editar un tratamiento:** Desde el listado de tratamientos o dentro del detalle de evolución, aparece una lista con los tratamientos del paciente. Junto a los nombres aparece el botón de editar. Si se pulsa, se dirige al formulario con los datos del tratamiento y se puede modificar todos los datos que se quiera. Para finalizar el cambio se pulsa en guardar.

8.3. **Borrar una tratamiento:** Desde el listado de tratamientos o dentro del detalle de evolución, aparece una lista con los tratamientos del paciente. Junto a los nombres aparece el botón de borrar. Al pulsar en el botón automáticamente se borra el tratamiento indicado y toda su información.

9. Creación, edición de un informe

9.1. **Crear un informe:** Desde el detalle de paciente, si aún no dispone de informes, nos aparecerán varios puntos donde añadirla como el menú, al pulsar en informes. Se pulsa en cualquiera de los mencionados anteriormente y se cumplimenta el formulario que aparece. Para finaliza la acción, marcamos en guardar.

9.2. **Editar un informe:** Desde el listado de informes aparece una lista con los informes del paciente. Junto a los nombres aparece el botón de editar. Si se pulsa, se dirige al formulario con los datos del informe y se puede modificar todos los datos que se quiera. Para finalizar el cambio se pulsa en guardar.

9.3. **Borrar un informe:** Desde el listado de informes aparece una lista con los informes del paciente. Junto a los nombres aparece el botón de borrar. Al pulsar en el botón automáticamente se borra el informe indicado y toda su información.

15. 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 de la aplicación.

- **Gestión de citas médicas:** Se incluirá un gestor donde se implementará un calendario con las citas de los pacientes. Se ha reservado según los wireframes y los prototipos de alto nivel un espacio para incluir este apartado en una nueva versión.
- **Exploraciones complementarias:** Se incluirá la información ajena a nuestra consulta, como por ejemplo pruebas médicas en hospitales, o informes de otros médicos.
- **SSL:** Se incluirá para mejorar la seguridad y posicionamiento de la aplicación.
- **Mayor seguridad en API REST:** Se implementarán mejoras en el apartado del backend. Ya que se ha realizado acciones de seguridad sencillas, aunque primordiales.

16. Presupuesto

A continuación, se detallará el presupuesto estimado para el diseño y desarrollo de la aplicación web. Para el equipo humano se ha estimado que se trabajará durante 4 horas cada día, por lo que tenemos en total 860 horas. Si a esto lo multiplicamos por el importe por hora que cobra un ingeniero, estimado en 35€/hora. Obtenemos el importe total de **30.100€**.

EQUIPO HUMANO			
Acciones	Tiempo	Horas	Importe
Definición de la idea principal	7 días	28 h	980 €
Planificación y análisis de requisitos	4 días	16 h	560 €
Elaboración de propuestas y solución final	4 días	16 h	560 €
Diseño de la aplicación	20 días	80 h	2.800 €
Desarrollo de la aplicación	180 días	720 h	25.200 €
TOTAL			30.100€

Tabla 5: Presupuesto equipo humano

EQUIPAMIENTO TÉCNICO			
Recursos	Tipo de pago	Importe por unidad	Importe anual
Servidor web	mensual	80 €	960 €
Dominio (.es y .com)	anual	14 €	28 €
WebStorm	anual	129 €	129 €
PhpStorm	anual	199 €	199 €
Balsamiq	anual	90 €	90 €
Photoshop	mensual	29,99 €	359,88 €
TOTAL			1.765,88 €

Tabla 6: Presupuesto equipamiento técnico

17. Análisis de mercado

Antes de detallar los objetivos del proyecto y el desarrollo del mismo, es de vital importancia estudiar las aplicaciones existentes en el mercado actual que tiene la misma temática que la presentada en este trabajo fin de máster. Gracias a ello, se tendrá una visión global de las soluciones que aportan y si han resuelto los problemas de los usuarios que las demandan.

El segmento que tenemos como objetivo es el de médico autónomo que gestiona una consulta individual.

Las palabras clave que se han empleado en la búsqueda “aplicación web consultas médicas”, “aplicaciones web para organizar consultas médicas”. Como característica principal hemos observado que las aplicaciones están basadas para clínicas, hospitales y también contienen muchas variables que descentralizan el foco principal de la gestión completa del paciente.

Ahora, se pasará a analizar las distintas aplicaciones existentes.

Salucenter: Software de gestión de pacientes para profesionales de la salud.

- Tipo aplicación: web
- Características: Calendario, notificaciones, archivos de pacientes, historial, gestión de mutuas, facturas.
- Precios: desde 19,95€ a 69,95 mes

Mediconta: Se trata de un programa o software médico para gestión de Clínicas Médicas que le permitirá gestionar su Clínica o Gabinete Médico de una manera cómoda y fácil gracias a su intuitivo diseño.

- Tipo aplicación: software
- Características: Funciones administrativas, Gestión de pacientes, Herramientas, Contabilidad, Gestión de almacén.
- Precios: 64€ + IVA al año

Clinics sft, Software para la gestión de clínicas médicas, permite el trabajo en red de su centro, envíe SMS automáticamente, simplifique la contabilidad, todo ello para mejorar la calidad en la gestión diaria.

- Tipo aplicación: web
- Características: Citas, pacientes contabilidad, administración, CMI, especialidad sector dental.
- Precios: Alta 1175€ + Mensual desde 23,33€ a 65€.

Ofimedic, Ofimedic.Sistemas informáticos de gestión de consultas médicas. Agendas, recepción, consultorios, pacientes, multimedia, informes, tratamientos, exploraciones, diagnósticos, recetas...

- Tipo aplicación: software + aplicación web
- Características: agenda, historias médicas, administración, CRM, estadística y LOPD.

- Precios: Diferentes precios con permanencia.

Q soft, SALUS – Líder en soluciones de Software Médico para la Gestión Integral de Clínicas, Consultas, Centros Médicos y Hospitales.

- Tipo aplicación: software
- Características: agenda, historias médicas, administración, facturación...
- Precios: No hacen referencia.

18. Conclusiones

General

En diferentes partes del proyecto se ha comentado que el propósito final de este trabajo fin de máster consistía en realizar una aplicación web para la gestión de consultas médicas.

Para su desarrollo se ha partido de un proceso comercial de una aplicación desde cero, ya que se ha trabajado en aspectos tan importantes como el presupuesto, planificación, metodología y objetivos para alcanzar en el trabajo. Además se ha realizado un trabajo metódico incluyendo tanto los bocetos y prototipos de la aplicación, identidad corporativa, arquitectura de la aplicación, composición de la base de datos, tecnologías para el desarrollo, herramientas para el trabajo profesional y hasta una proyección de futuro para una nueva versión.

Ante estas premisas marcadas durante la elaboración del proyecto, se puede considerar que todas ellas se han logrado recopilar, evaluar y entender a lo largo de todo el documento. A título personal, considero que tras obtener una aplicación que cumple los requisitos propuestos, el producto puede ser lanzado como una versión BETA, para ver su funcionamiento con usuarios profesionales que pueda realizar un nuevo test a la aplicación.

Planificación realizada

Durante el desarrollo del proyecto, se han ido cumpliendo los objetivos designados en la planificación propuesta, además en el plazo marcado y sin contratiempos graves. La única variación que se produjo dentro del proceso interno, fue el incremento de horas de programación para la puesta a punto de la aplicación web en un servidor online, debido a que los servidores compartidos tenían aspectos capados, que no nos permitían disfrutar plenamente de todas las características de la aplicación. Igualmente, no se han visto modificada ninguna fecha de entrega.

Anexo 1. Entregables del proyecto

Lista de archivos entregados y su descripción.

Archivo: PAC_FINAL_prj_Carmona_Martos_Alejandro.zip

Contiene tanto la carpeta del frontend y el backend de nuestro proyecto, como la carpeta de manuales de uso e instalación.

- **Proyecto desarrollado**
 - / skipu / **skipu-frontend y skipu-backend**
- **Manuales de uso e instalación de la aplicación web.**
 - Usuario: / skipu / manuales / **Instrucciones de uso de la aplicación.pdf**
 - Instalación: / skipu / manuales / **Instrucciones de instalación de la aplicación.pdf**

Archivo: PAC_FINAL_mem_Carmona_Martos_Alejandro.zip

Contiene tanto la memoria final del proyecto, como los anexos y autoevaluación.

- **Memoria completa del trabajo fin de máster**
- **TFM_mem_CarmonaMartos_Alejandro.pdf**
- **Anexos**
 - anexos / **Figura 26.jpg** Árbol de navegación de la aplicación web.
 - anexos / **competencias_transversales_del_proyecto.pdf**
 - anexos / LO-FI /
 - Figura 3.jpg: Wireframe de la página de inicio en versión escritorio
 - Figura 4.jpg: Wireframe acceso a la aplicación
 - Figura 5.jpg: Wireframe de registro de usuario a la aplicación.
 - Figura 6.jpg: Wireframe del flujo de registro de usuario versión escritorio.
 - Figura 7.jpg: Wireframe de la página de dashboard versión escritorio
 - Figura 8.jpg: Wireframe de la página de inicio versión tablet
 - Figura 9.jpg: Wireframe del flujo de registro de usuario versión tablet
 - Figura 10.jpg: Wireframe de la página de dashboard versión tablet
 - Figura 11.jpg: Wireframe de la página de inicio versión móvil
 - Figura 12.jpg: Wireframe del flujo de registro de usuario versión móvil
 - Figura 13.jpg: Wireframe de la página de dashboard versión móvil
 - Figura 14.jpg: Wireframe del flujo de añadir nuevo paciente versión escritorio
 - Figura 15.jpg: Wireframe del flujo de añadir nuevo paciente versión tablet
 - Figura 16.jpg: Wireframe del flujo de añadir nuevo paciente versión móvil
- **Autoinforme de evaluación**
 - autoinforme / **Informe_Autoevaluacion_TFM_Alejandro_Carmona_Martos_es**

Archivo: PAC_FINAL_vid_Carmona_Martos_Alejandro.zip

Incluye un archivo con el video de presentación de la aplicación skipu en .mp4

- **PAC_FINAL_vid_Carmona_Martos_Alejandro.mp4**

Archivo: PAC_FINAL_prs_Carmona_Martos_Alejandro.zip

Incluye dos archivos con la presentación comercial tanto en PDF como en Powerpoint.

- **PAC_FINAL_prs_Carmona_Martos_Alejandro.pdf**
- **PAC_FINAL_prs_Carmona_Martos_Alejandro.pptx**

Anexo 2. Código fuente (extractos)

Selección de partes relevantes del código fuente de la aplicación creada.

Backend: Ejemplo de CRUD (create, read, update and delete) en la parte del backend de nuestra API Rest
PacienteController.php

```
<?php
namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Http\Requests;
use App\Helpers\JwtAuth;
use App\Paciente;
use App\Tratamiento;
use App\Historia;
use App\Evolucion;
use App\Informe;

class PacienteController extends Controller
{
    //INDICE
    public function index(Request $request){
        // carga todos los pacientes con todos los datos del usuario
        $pacientes = Paciente::all();
        return response()->json(array(
            'pacientes' => $pacientes,
            'status' => 'success',
        ),200);
    }

    // MOSTRAR PACIENTE
    public function show($id){
        // busca por id un paciente
        $paciente = Paciente::find($id);
        if(is_object($paciente)){
            $paciente = Paciente::find($id);
            return response()->json(array(
                'paciente' => $paciente,
                'status' => 'success',
            ),200);
        }else{
            return response()->json(array(
                'paciente' => 'El paciente no existe',
                'status' => 'error',
            ),200);
        }
    }

    //GUARDAR CONSULTA
    public function store(Request $request){
        // Comprobación para poder interactuar con nuestra API a través del token del usuario y JWT.
        $hash = $request->header('Authorization',null);
        $jwtAuth = new JwtAuth();
        $checkToken = $jwtAuth->checkToken($hash);

        if($checkToken){
            // RECOGER DATOS POR POST
            $json = $request->input('json',null);
            $params = json_decode($json);
            $paramsArray = json_decode($json,true);

            // CONSEGUIR USUARIO IDENTIFICADO
            $user = $jwtAuth->checkToken($hash, true);

            // VALIDACION
            $request->merge($paramsArray);
            try{
                $validate = $this->validate($request,[
                    'name' => 'required'
                ]);
            }catch(\Illuminate\Validation\ValidationException $e){
                return $e->getReponse();
            }
        }

        // GUARDAR CONSULTA

        $paciente = new Paciente();
        $paciente->consulta_id = $params->consulta_id;
        $paciente->name = $params->name;
        $paciente->surname = $params->surname;
        $paciente->email = $params->email;
        $paciente->telefono = $params->telefono;
        $paciente->documento = $params->documento;
        $paciente->domicilio = $params->domicilio;
```



```

        $historia->delete();

        /* Comprobamos si el paciente tiene informes
        para eliminarlos antes de borrar el paciente */
        $informe = Informe::where('paciente_id', $id);
        $informe->delete();

        $paciente = Paciente::find($id);

        // Borrar paciente
        $paciente->delete();

        // Devolverlo
        $data = array(
            'paciente' => $paciente,
            'status' => 'success',
            'code' => 200
        );
    }else{
        $data = array(
            'paciente' => 'Error al borrar el paciente',
            'status' => 'success',
            'code' => 300
        );
    }

    return response()->json($data, 200);
}
}
}

```

Frontend: Ejemplo de servicio CRUD (create, read, update and delete) de paciente en la parte del frontend (Angular) de nuestra API Rest.

PacienteService.ts

```

import { Injectable } from "@angular/core";
import { HttpClient, HttpHeaders } from "@angular/common/http";

import { Observable } from "rxjs";
import { GLOBAL } from "../global";
import { Paciente } from "../models/paciente";

@Injectable()
export class PacienteService {
    public url: string;

    constructor(
        public _http: HttpClient
    ){
        this.url = GLOBAL.url;
    }

    pruebas(){
        return "CONSULTA SEVICIO FUNCIONA"
    }

    create(token, paciente: Paciente): Observable<any>{
        let json = JSON.stringify(paciente);
        let params = 'json='+json;

        let headers = new HttpHeaders().set('Content-Type', 'application/x-www-form-urlencoded')
            .set('Authorization', token);

        return this._http.post(this.url+'pacientes', params, {headers: headers});
    }

    getPacientes(): Observable<any>{
        let headers = new HttpHeaders().set('Content-Type', 'application/x-www-form-urlencoded') ;
        return this._http.get(this.url+'pacientes',{headers: headers});
    }

    getPaciente(id):Observable<any>{
        return this._http.get(this.url+'pacientes/'+id);
    }

    update(token, paciente, id):Observable<any>{
        let json = JSON.stringify(paciente);
        let params = 'json='+json;

        let headers = new HttpHeaders().set('Content-Type', 'application/x-www-form-urlencoded')
            .set('Authorization', token);
        return this._http.put(this.url+'pacientes/'+id, params, {headers: headers});
    }

    delete(token, id): Observable<any>{
        let headers = new HttpHeaders().set('Content-Type', 'application/x-www-form-urlencoded')
            .set('Authorization', token);
        return this._http.delete(this.url+'pacientes/'+id, {headers: headers});
    }
}

```

Anexo 3. Librerías/Código externo utilizado

Información detallada acerca de qué librerías, código, archivos, y cualquier otra herramienta tecnológica desarrollada por terceros utilizada en el trabajo, y qué partes de los mismos han sido usadas y cómo.

Bootstrap

<https://getbootstrap.com/docs/4.3/getting-started/introduction/>

Aunque se ha realizado un diseño desde cero se ha utilizado Bootstrap como base para elementos como el Grid de la aplicación, facilitándonos en este aspecto la composición rápida de la estructura y del responsive.

JWT

Firebase/php-jwt

Con esta librería hemos podido crear el sistema de tokens, ya que brinda control absoluto de la autenticación, ya que permite autenticar usuarios o dispositivos mediante tokens web JSON (JWT) seguros.

jQuery

<https://jquery.com/>

Es una biblioteca multiplataforma de JavaScript, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. En nuestro se ha empleado para manejar el menú vertical en el responsive.

Anexo 4. Libro de estilo

Libro de estilo que define la línea gráfica del trabajo.

- Logotipo.

skipu

- Paleta de colores.
 - ■ #82E0AA color principal
 - ■ #222222 color secundario y para texto de lectura.
- Tipografía.
 - Nombre: Roboto

Rr ABCČĆDĐEFGHIJKLMNOPQRSŠT
UVWXYZŽabcčćdđefghijklmnopq
rsštuvwxyzžАБВГГДЂЕЁЄЖЗСИІ
ЇЙЈКЛЉМНЊОПРСТЋУЎФХЦЧЏ
ШЩЪЫЬЭЮЯабвггдђеєжзсииій
јкљмњопрстћуўфхцчџшщъыь
эюяАВГДЕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦ
ΧΨΩαβγδεζηθικλμνξοπρστυφχψ
ωάΆέΈεΉίϊΊόΌύϋΎΏΆÂÊÔσϜă
âêôσϜ1234567890'?'!"(%)[#]{@}
/&\<-+÷×=>®©\$€£¥ç;:,.*

- Estilos

Thin
Thin Italic
Light
Light Italic
Regular
Regular Italic
Medium
Medium Italic
Bold
Bold Italic
Black
Black Italic

- Iconos y otros elementos gráficos.
 - Iconos tomados de Fontawesome relacionados con consultas médicas.

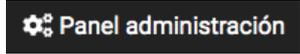


- Botones.

- Botón principal



- Botón principal efecto hover



- Botón secundario



- Botón secundario con efecto hover



Anexo 5. Bibliografía

- Angular. (2019). Documentación oficial de Angular: <https://angular.io/docs>
- Laravel. (2019). Documentación oficial de Laravel: <https://laravel.com/docs/5.8>
- Bootstrap. (2019). Grid system de Bootstrap: <https://getbootstrap.com/docs/4.0/layout/grid/>
- ExpertTyCE. (1 de 11 de 2018). Cómo crear un API Rest con Laravel 5.7 y JWT Token. Obtenido de Medium: <https://medium.com/@experttyce/c%C3%B3mo-crear-un-api-rest-con-laravel-5-7-y-jwt-token-94b79c533c6d>
- Javier Cobos Sanz. (4 de 5 de 2017). Iniciando un proyecto_ Laravel 5.4 y Angular 4: <https://javiercobossanz.com/2017/05/04/iniciando-un-proyecto-laravel-5-4-y-angular-4/#.XP07INMzYWp>
- UOC (Sin fecha). Flat design (diseño plano) <http://multimedia.uoc.edu/blogs/dii/es/tendencias/flat-design-disseny-pla/>
- UOC (Sin fecha). Web adaptativa: <http://multimedia.uoc.edu/blogs/dii/es/tendencias/web-adaptatiu/>
- UOC (Sin fecha). Diseñar para varias plataformas: <http://multimedia.uoc.edu/blogs/dii/es/disseny/dissenyar-per-a-diverses-plataformes/>
- UOC (Sin fecha). Pautas de diseño de páginas web: <http://multimedia.uoc.edu/blogs/dii/es/disseny/pautes-de-disseny/pautes-de-disseny-de-pagines-web/>
- UOC (Sin fecha). Pautas de diseño de dispositivos móviles. <http://multimedia.uoc.edu/blogs/dii/es/disseny/pautes-de-disseny/pautes-de-disseny-de-dispositius-mobils/>
- UOC (Sin fecha). Prototipado: <http://multimedia.uoc.edu/blogs/dii/es/prototipatge/>
- UOC(Sin fecha). Prototipado Herramientas Recomendadas: <http://multimedia.uoc.edu/blogs/dii/es/prototipatge/eines-recomanades/>.

Anexo 6. Vita

Alejandro Carmona Martos, natural de Córdoba y residente en Cádiz. Es Ingeniero Técnico en Informática de Gestión por la Universidad de Almería finalizando sus estudios en 2009. Posteriormente realizó un Máster Propio en Administración, Comunicación y Seguridad Informática por la misma universidad, enfocado en la administración de sistemas operativos, seguridad informática y redes, finalizando los estudios en 2013 y presentando como trabajo final de máster una guía para el desarrollo de videojuegos para Android.

Gracias a esta formación, comenzó a trabajar como desarrollador web en la empresa Eco Eco Proyectos, creando un proyecto en Drupal hasta 2015.

En el curso 2015 / 2016 comienza el Máster del Profesorado en la Universidad de Córdoba, compatibilizando la asistencia con el trabajo en la agencia de comunicación Terceto en Granada. En Noviembre de 2017 en Granada, da el salto al mundo del freelance o autónomo, colaborando con una nueva agencia de Comunicación No Solo Una Idea, y dando clases de diseño y desarrollo web en la escuela de arte Sboza2.

Por motivos personales se traslada en septiembre de 2018 a Cádiz, donde reside actualmente y ha realizado este Máster, además de compatibilizar tu trabajo como freelance en diferentes proyectos.